
CCR Documentation

Release 0.3

CCR Tools Team

January 03, 2016

1	CCR Documentation	1
1.1	CCR Documentation	1
1.2	Indices and tables	417
1.3	Description	417
1.4	Functions	417
1.5	Session Management	417
1.6	Authentication	418
1.7	Custom Exceptions	419
2	Indices and tables	421
3	Description	423
4	Functions	425
5	Session Management	427
6	Authentication	429
7	Custom Exceptions	431
8	Indices and tables	433
	Python Module Index	435

CCR Documentation

1.1 CCR Documentation

1.1.1 CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

CCR Documentation

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)
search for some keywords - returns results as a list

`ccr.info` (*package*)
get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)
submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

`class ccr.Session(username=None, password=None, rememberme=False)`
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)
submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)
unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)
unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)
unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)
store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

ccr.search (*keywords*)
search for some keywords - returns results as a list

ccr.info (*package*)
get information for a specific package - returns results as a list

ccr.msearch (*maintainer*)
search for packages owned by 'maintainer' - returns results as a list

ccr.list_orphans ()
search for orphaned packages - returns results as a list

Session Management

class ccr.Session (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)
disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)
flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)
set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f, category*)
submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)
unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search (keywords)`
search for some keywords - returns results as a list

`ccr.info (package)`
get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails**authenticate** (*username, password, rememberme=False*)authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails**check_vote** (*package, return_id=False*)check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur**close** ()

end the session

delete (*package*)delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure**disown** (*package*)disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure**flag** (*package*)flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure**notify** (*package*)set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure**setcategory** (*package, category*)change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.**submit** (*f, category*)submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur**unflag** (*package*)unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure**unnotify** (*package*)unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure**unvote** (*package*)unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote**vote** (*package*)vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class ccrauth.AuthFile

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class ccrauth.AuthDB

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`

search for some keywords - returns results as a list

`ccr.info(package)`

get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session(username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exit

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search (keywords)`
search for some keywords - returns results as a list

`ccr.info (package)`
get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exit

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class ccrauth.AuthFile

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class ccrauth.AuthDB

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session(username=None, password=None, rememberme=False)`
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exit

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class ccrauth.AuthFile

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class ccrauth.AuthDB

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`

search for some keywords - returns results as a list

`ccr.info(package)`

get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session(username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)
 check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()
 end the session

delete (*package*)
 delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)
 disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)
 flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)
 set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)
 change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)
 submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)
 unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)
 unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)
 unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)
 vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class ccrauth.AuthFile

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class ccrauth.AuthDB

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session(username=None, password=None, rememberme=False)`
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)
 submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
 raises a `ConnectionError` if a network error occur

unflag (*package*)
 unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
 raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)
 unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
 a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)
 unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)
 vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
 if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails**authenticate** (*username, password, rememberme=False*)authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails**check_vote** (*package, return_id=False*)check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur**close** ()

end the session

delete (*package*)delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure**disown** (*package*)disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure**flag** (*package*)flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure**notify** (*package*)set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure**setcategory** (*package, category*)change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.**submit** (*f, category*)submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur**unflag** (*package*)unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure**unnotify** (*package*)unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure**unvote** (*package*)unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote**vote** (*package*)vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`

search for some keywords - returns results as a list

`ccr.info(package)`

get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session(username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails**authenticate** (*username, password, rememberme=False*)authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails**check_vote** (*package, return_id=False*)check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur**close** ()

end the session

delete (*package*)delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure**disown** (*package*)disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure**flag** (*package*)flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure**notify** (*package*)set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure**setcategory** (*package, category*)change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.**submit** (*f, category*)submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur**unflag** (*package*)unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure**unnotify** (*package*)unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure**unvote** (*package*)unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote**vote** (*package*)vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class ccrauth.AuthFile

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class ccrauth.AuthDB

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`

search for some keywords - returns results as a list

`ccr.info(package)`

get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session(username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search (keywords)`
search for some keywords - returns results as a list

`ccr.info (package)`
get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)
 submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
 raises a `ConnectionError` if a network error occur

unflag (*package*)
 unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
 raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)
 unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
 a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)
 unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)
 vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
 if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails**authenticate** (*username, password, rememberme=False*)authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails**check_vote** (*package, return_id=False*)check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur**close** ()

end the session

delete (*package*)delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure**disown** (*package*)disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure**flag** (*package*)flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure**notify** (*package*)set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure**setcategory** (*package, category*)change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.**submit** (*f, category*)submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur**unflag** (*package*)unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure**unnotify** (*package*)unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure**unvote** (*package*)unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote**vote** (*package*)vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search (keywords)`
search for some keywords - returns results as a list

`ccr.info (package)`
get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exit

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)
 submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
 raises a `ConnectionError` if a network error occur

unflag (*package*)
 unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
 raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)
 unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
 a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)
 unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)
 vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
 if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails**authenticate** (*username, password, rememberme=False*)authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails**check_vote** (*package, return_id=False*)check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur**close** ()

end the session

delete (*package*)delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure**disown** (*package*)disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure**flag** (*package*)flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure**notify** (*package*)set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure**setcategory** (*package, category*)change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.**submit** (*f, category*)submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur**unflag** (*package*)unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure**unnotify** (*package*)unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure**unvote** (*package*)unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote**vote** (*package*)vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`

search for some keywords - returns results as a list

`ccr.info(package)`

get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session(username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search (keywords)`
search for some keywords - returns results as a list

`ccr.info (package)`
get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class ccrauth.AuthFile

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class ccrauth.AuthDB

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`

search for some keywords - returns results as a list

`ccr.info(package)`

get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search (keywords)`
search for some keywords - returns results as a list

`ccr.info (package)`
get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails**authenticate** (*username, password, rememberme=False*)authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails**check_vote** (*package, return_id=False*)check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur**close** ()

end the session

delete (*package*)delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure**disown** (*package*)disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure**flag** (*package*)flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure**notify** (*package*)set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure**setcategory** (*package, category*)change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.**submit** (*f, category*)submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur**unflag** (*package*)unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure**unnotify** (*package*)unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure**unvote** (*package*)unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote**vote** (*package*)vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class ccrauth.AuthFile

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class ccrauth.AuthDB

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`

search for some keywords - returns results as a list

`ccr.info(package)`

get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session(username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)
submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)
unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)
unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)
unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)
vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exit

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)
 submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
 raises a `ConnectionError` if a network error occur

unflag (*package*)
 unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
 raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)
 unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
 a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)
 unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)
 vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
 if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class ccrauth.AuthFile

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class ccrauth.AuthDB

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`

search for some keywords - returns results as a list

`ccr.info(package)`

get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session(username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exit

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)
search for some keywords - returns results as a list

`ccr.info` (*package*)
get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exit

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (ccr.auth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)
search for some keywords - returns results as a list

`ccr.info` (*package*)
get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)
 submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
 raises a `ConnectionError` if a network error occur

unflag (*package*)
 unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
 raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)
 unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
 a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)
 unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)
 vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
 if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exit

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`

search for some keywords - returns results as a list

`ccr.info(package)`

get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session(username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)
search for some keywords - returns results as a list

`ccr.info` (*package*)
get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)
 submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
 raises a `ConnectionError` if a network error occur

unflag (*package*)
 unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
 raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)
 unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
 a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)
 unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)
 vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
 if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`

search for some keywords - returns results as a list

`ccr.info(package)`

get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session(username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exit

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)
search for some keywords - returns results as a list

`ccr.info` (*package*)
get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class ccrauth.AuthFile

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class ccrauth.AuthDB

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
 search for some keywords - returns results as a list

`ccr.info(package)`
 get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
 search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
 search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
 class for all CCR actions

adopt (*package*)
 adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
 authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
 check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
 end the session

delete (*package*)
 delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
 disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
 flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
 set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
 change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`

search for some keywords - returns results as a list

`ccr.info(package)`

get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)
 submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
 raises a `ConnectionError` if a network error occur

unflag (*package*)
 unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
 raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)
 unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
 a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)
 unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)
 vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
 if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exit

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class ccrauth.AuthFile

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class ccrauth.AuthDB

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session(username=None, password=None, rememberme=False)`
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exit

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)
search for some keywords - returns results as a list

`ccr.info` (*package*)
get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)
 submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
 raises a `ConnectionError` if a network error occur

unflag (*package*)
 unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
 raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)
 unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
 a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)
 unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)
 vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
 if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exit

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails**authenticate** (*username, password, rememberme=False*)authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails**check_vote** (*package, return_id=False*)check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur**close** ()

end the session

delete (*package*)delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure**disown** (*package*)disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure**flag** (*package*)flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure**notify** (*package*)set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure**setcategory** (*package, category*)change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.**submit** (*f, category*)submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur**unflag** (*package*)unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure**unnotify** (*package*)unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure**unvote** (*package*)unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote**vote** (*package*)vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`

search for some keywords - returns results as a list

`ccr.info(package)`

get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session(username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)
 check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()
 end the session

delete (*package*)
 delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)
 disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)
 flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)
 set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)
 change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)
 submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)
 unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)
 unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)
 unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)
 vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class ccrauth.AuthFile

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class ccrauth.AuthDB

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)
search for some keywords - returns results as a list

`ccr.info` (*package*)
get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)
 submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
 raises a `ConnectionError` if a network error occur

unflag (*package*)
 unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
 raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)
 unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
 a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)
 unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)
 vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
 if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exit

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)
search for some keywords - returns results as a list

`ccr.info` (*package*)
get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)
 submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
 raises a `ConnectionError` if a network error occur

unflag (*package*)
 unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
 raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)
 unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
 a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)
 unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)
 vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
 if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`

search for some keywords - returns results as a list

`ccr.info(package)`

get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session(username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session(username=None, password=None, rememberme=False)`
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)
 submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
 raises a `ConnectionError` if a network error occur

unflag (*package*)
 unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
 raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)
 unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
 a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)
 unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)
 vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
 if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`

search for some keywords - returns results as a list

`ccr.info(package)`

get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exit

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)
 check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()
 end the session

delete (*package*)
 delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)
 disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)
 flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)
 set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)
 change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)
 submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)
 unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)
 unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)
 unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)
 vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class ccrauth.AuthFile

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class ccrauth.AuthDB

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`

search for some keywords - returns results as a list

`ccr.info(package)`

get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session(username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session(username=None, password=None, rememberme=False)`
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
 raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
 raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
 a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
 if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- modindex
- search

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search(keywords)`
search for some keywords - returns results as a list

`ccr.info(package)`
get information for a specific package - returns results as a list

`ccr.msearch(maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exit

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package*, *return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication**class** `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions**exception** `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)

- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search (keywords)`
search for some keywords - returns results as a list

`ccr.info (package)`
get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)
disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)
flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)
set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f*, *category*)
 submit a package to CCR raises `KeyError` on bad category raises `IOError` [Errno 2] if 'f' does not exist
 raises a `ConnectionError` if a network error occur

unflag (*package*)
 unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
 raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)
 unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
 a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)
 unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)
 vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
 if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username*, *password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username*, *password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session (username=None, password=None, rememberme=False)`

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccr.auth.AuthFile`

A class to manage authentication information in a file (`ccr.auth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description `ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management**class** `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails**authenticate** (*username, password, rememberme=False*)authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails**check_vote** (*package, return_id=False*)check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur**close** ()

end the session

delete (*package*)delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure**disown** (*package*)disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure**flag** (*package*)flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure**notify** (*package*)set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure**setcategory** (*package, category*)change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.**submit** (*f, category*)submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur**unflag** (*package*)unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure**unnotify** (*package*)unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure**unvote** (*package*)unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote**vote** (*package*)vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description

`ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)
authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package, return_id=False*)
check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)
disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)
flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)
set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package, category*)
change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f, category*)
submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)
unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)
unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)
unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)
vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

Authentication

class ccrauth.**AuthFile**

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)
store authentication information in the file raises IOError if the disk is full

class `ccrauth.AuthDB`
A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)
store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`
Package does not exist

exception `ccr.InvalidPackage`
Invalid package or wrong file type

exception `ccr.CCRWarning`
Base class for all other warnings

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Description

`ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search` (*keywords*)
search for some keywords - returns results as a list

`ccr.info` (*package*)
get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()
search for orphaned packages - returns results as a list

Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_DeleteWarning` on failure

disown (*package*)

disown a CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` on failure

flag (*package*)

flag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

notify (*package*)

set the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `requests.ConnectionError` if a network error occur raises a `_CategoryWarning` for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises `KeyError` on bad category raises `IOError [Errno 2]` if 'f' does not exist raises a `ConnectionError` if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

1.1.2 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

1.1.3 Description

`ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

1.1.4 Functions

`ccr.search` (*keywords*)

search for some keywords - returns results as a list

`ccr.info` (*package*)

get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)

search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans` ()

search for orphaned packages - returns results as a list

1.1.5 Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)

class for all CCR actions

adopt (*package*)

adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn't exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a `ConnectionError` if a network error occur raises a `ValueError` if login fails

check_vote (*package*, *return_id=False*)
check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()
end the session

delete (*package*)
delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)
disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)
flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)
set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package*, *category*)
change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f*, *category*)
submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)
unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)
unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)
unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)
vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

1.1.6 Authentication

class ccrauth.**AuthFile**
A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username*, *password*)
store authentication information in the file raises IOError if the disk is full

class ccrauth.**AuthDB**
A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)
store/update authentication information in the database

1.1.7 Custom Exceptions

exception `ccr.PackageNotFound`
Package does not exist

exception `ccr.InvalidPackage`
Invalid package or wrong file type

exception `ccr.CCRWarning`
Base class for all other warnings

1.2 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

1.3 Description

`ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

1.4 Functions

`ccr.search` (*keywords*)
search for some keywords - returns results as a list

`ccr.info` (*package*)
get information for a specific package - returns results as a list

`ccr.msearch` (*maintainer*)
search for packages owned by ‘maintainer’ - returns results as a list

`ccr.list_orphans` ()
search for orphaned packages - returns results as a list

1.5 Session Management

class `ccr.Session` (*username=None, password=None, rememberme=False*)
class for all CCR actions

adopt (*package*)
adopt an orphaned CCR package raises a `PackageNotFound` exception if the package doesn’t exist raises a `ConnectionError` if a network error occur raises a `_OwnershipWarning` if the package is already maintained or if it fails

authenticate (*username, password, rememberme=False*)

authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

check_vote (*package, return_id=False*)

check to see if you have already voted for a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur

close ()

end the session

delete (*package*)

delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _DeleteWarning on failure

disown (*package*)

disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _OwnershipWarning on failure

flag (*package*)

flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

notify (*package*)

set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

setcategory (*package, category*)

change/set the category of a package already in the CCR raises a PackageNotFound exception if the package doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for an invalid category or if it fails.

submit (*f, category*)

submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist raises a ConnectionError if a network error occur

unflag (*package*)

unflag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _FlagWarning on failure

unnotify (*package*)

unset the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _NotifyWarning on failure

unvote (*package*)

unvote a package on CCR raises a PackageNotFound exception if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a PackageNotFound if the package doesn't exist raises a ConnectionError if a network error occur raises a _VoteWarning if it is already voted or if it couldn't vote

1.6 Authentication

class ccrauth.AuthFile

A class to manage authentication information in a file (ccrauth.txt)

store_auth_info (*username, password*)

store authentication information in the file raises IOError if the disk is full

class `ccr.auth.AuthDB`

A class to manage authentication information in a database (ccr.db)

store_auth_info (*username, password*)

store/update authentication information in the database

1.7 Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

Description

`ccr` is a Python library for interacting with the Chakra Community Repository, or CCR.

Functions

`ccr.search (keywords)`
search for some keywords - returns results as a list

`ccr.info (package)`
get information for a specific package - returns results as a list

`ccr.msearch (maintainer)`
search for packages owned by 'maintainer' - returns results as a list

`ccr.list_orphans ()`
search for orphaned packages - returns results as a list

Session Management

```

class ccr.Session (username=None, password=None, rememberme=False)
    class for all CCR actions

    adopt (package)
        adopt an orphaned CCR package raises a PackageNotFound exception if the package doesn't exist raises a
        ConnectionError if a network error occur raises a _OwnershipWarning if the package is already maintained
        or if it fails

    authenticate (username, password, rememberme=False)
        authenticate on CCR raises a ConnectionError if a network error occur raises a ValueError if login fails

    check_vote (package, return_id=False)
        check to see if you have already voted for a package raises a PackageNotFound exception if the package
        doesn't exist raises a ConnectionError if a network error occur

    close ()
        end the session

    delete (package)
        delete a package from CCR raises a PackageNotFound exception if the package doesn't exist raises a
        ConnectionError if a network error occur raises a _DeleteWarning on failure

    disown (package)
        disown a CCR package raises a PackageNotFound exception if the package doesn't exist raises a Connec-
        tionError if a network error occur raises a _OwnershipWarning on failure

    flag (package)
        flag a CCR package as out of date raises a PackageNotFound exception if the package doesn't exist raises
        a ConnectionError if a network error occur raises a _FlagWarning on failure

    notify (package)
        set the notify flag on a package raises a PackageNotFound exception if the package doesn't exist raises a
        ConnectionError if a network error occur raises a _NotifyWarning on failure

    setcategory (package, category)
        change/set the category of a package already in the CCR raises a PackageNotFound exception if the pack-
        age doesn't exist raises a requests.ConnectionError if a network error occur raises _CategoryWarning for
        an invalid category or if it fails.

    submit (f, category)
        submit a package to CCR raises KeyError on bad category raises IOError [Errno 2] if 'f' does not exist
        raises a ConnectionError if a network error occur

```

unflag (*package*)

unflag a CCR package as out of date raises a `PackageNotFound` exception if the package doesn't exist
raises a `ConnectionError` if a network error occur raises a `_FlagWarning` on failure

unnotify (*package*)

unset the notify flag on a package raises a `PackageNotFound` exception if the package doesn't exist raises
a `ConnectionError` if a network error occur raises a `_NotifyWarning` on failure

unvote (*package*)

unvote a package on CCR raises a `PackageNotFound` exception if the package doesn't exist raises a `Con-`
`nectionError` if a network error occur raises a `_VoteWarning` if it is already unvoted or if it couldn't unvote

vote (*package*)

vote for a package on CCR raises a `PackageNotFound` if the package doesn't exist raises a `ConnectionError`
if a network error occur raises a `_VoteWarning` if it is already voted or if it couldn't vote

Authentication

class `ccrauth.AuthFile`

A class to manage authentication information in a file (`ccrauth.txt`)

store_auth_info (*username, password*)

store authentication information in the file raises `IOError` if the disk is full

class `ccrauth.AuthDB`

A class to manage authentication information in a database (`ccr.db`)

store_auth_info (*username, password*)

store/update authentication information in the database

Custom Exceptions

exception `ccr.PackageNotFound`

Package does not exist

exception `ccr.InvalidPackage`

Invalid package or wrong file type

exception `ccr.CCRWarning`

Base class for all other warnings

Indices and tables

- `genindex`
- `modindex`
- `search`

C

ccr, [11](#)

A

- adopt()** (ccr.Session method), 12, 14, 15, 17, 19, 21, 23, 24, 26, 28, 30, 32, 33, 35, 37, 39, 41, 42, 44, 46, 48, 50, 51, 53, 55, 57, 59, 60, 62, 64, 66, 68, 69, 71, 73, 75, 77, 78, 80, 82, 84, 86, 87, 89, 91, 93, 95, 96, 98, 100, 102, 104, 105, 107, 109, 111, 113, 114, 116, 118, 120, 122, 123, 125, 127, 129, 131, 132, 134, 136, 138, 140, 141, 143, 145, 147, 149, 150, 152, 154, 156, 158, 159, 161, 163, 165, 167, 168, 170, 172, 174, 176, 177, 179, 181, 183, 185, 186, 188, 190, 192, 194, 195, 197, 199, 201, 203, 204, 206, 208, 210, 212, 213, 215, 217, 219, 221, 222, 224, 226, 228, 230, 231, 233, 235, 237, 239, 240, 242, 244, 246, 248, 249, 251, 253, 255, 257, 258, 260, 262, 264, 266, 267, 269, 271, 273, 275, 276, 278, 280, 282, 284, 285, 287, 289, 291, 293, 294, 296, 298, 300, 302, 303, 305, 307, 309, 311, 312, 314, 316, 318, 320, 321, 323, 325, 327, 329, 330, 332, 334, 336, 338, 339, 341, 343, 345, 347, 348, 350, 352, 354, 356, 357, 359, 361, 363, 365, 366, 368, 370, 372, 374, 375, 377, 379, 381, 383, 384, 386, 388, 390, 392, 393, 395, 397, 399, 401, 402, 404, 406, 408, 410, 411, 413, 415, 417, 427
- AuthDB** (class in ccrauth), 13, 15, 16, 18, 20, 22, 24, 25, 27, 29, 31, 33, 34, 36, 38, 40, 42, 43, 45, 47, 49, 51, 52, 54, 56, 58, 60, 61, 63, 65, 67, 69, 70, 72, 74, 76, 78, 79, 81, 83, 85, 87, 88, 90, 92, 94, 96, 97, 99, 101, 103, 105, 106, 108, 110, 112, 114, 115, 117, 119, 121, 123, 124, 126, 128, 130, 132, 133, 135, 137, 139, 141, 142, 144, 146, 148, 150, 151, 153, 155, 157, 159, 160, 162, 164, 166, 168, 169, 171, 173, 175, 177, 178, 180, 182, 184, 186, 187, 189, 191, 193, 195, 196, 198, 200, 202, 204, 205, 207, 209, 211, 213, 214, 216, 218, 220, 222, 223, 225, 227, 229, 231, 232, 234, 236, 238, 240, 241, 243, 245, 247, 249, 250, 252, 254, 256, 258, 259, 261, 263, 265, 267, 268, 270, 272, 274, 276, 277, 279, 281, 283, 285, 286, 288, 290, 292, 294, 295, 297, 299, 301, 303, 304, 306, 308, 310, 312, 313, 315, 317, 319, 321, 322, 324, 326, 328, 330, 331, 333, 335, 337, 339, 340, 342, 344, 346, 348, 349, 351, 353, 355, 357, 358, 360, 362, 364, 366, 367, 369, 371, 373, 375, 376, 378, 380, 382, 384, 385, 387, 389, 391, 393, 394, 396, 398, 400, 402, 403, 405, 407, 409, 411, 413, 414, 416, 418, 429
- authenticate()** (ccr.Session method), 12, 14, 16, 17, 19, 21, 23, 24, 26, 28, 30, 32, 33, 35, 37, 39, 41, 42, 44, 46, 48, 50, 51, 53, 55, 57, 59, 60, 62, 64, 66, 68, 69, 71, 73, 75, 77, 78, 80, 82, 84, 86, 87, 89, 91, 93, 95, 96, 98, 100, 102, 104, 105, 107, 109, 111, 113, 114, 116, 118, 120, 122, 123, 125, 127, 129, 131, 132, 134, 136, 138, 140, 141, 143, 145, 147, 149, 150, 152, 154, 156, 158, 159, 161, 163, 165, 167, 168, 170, 172, 174, 176, 177, 179, 181, 183, 185, 186, 188, 190, 192, 194, 195, 197, 199, 201, 203, 204, 206, 208, 210, 212, 213, 215, 217, 219, 221, 222, 224, 226, 228, 230, 231, 233, 235, 237, 239, 240, 242, 244, 246, 248, 249, 251, 253, 255, 257, 258, 260, 262, 264, 266, 267, 269, 271, 273, 275, 276, 278, 280, 282, 284, 285, 287, 289, 291, 293, 294, 296, 298, 300, 302, 303, 305, 307, 309, 311, 312, 314, 316, 318, 320, 321, 323, 325, 327, 329, 330, 332, 334, 336, 338, 339, 341, 343, 345, 347, 348, 350, 352, 354, 356, 357, 359, 361, 363, 365, 366, 368, 370, 372, 374, 375, 377, 379, 381, 383, 384, 386, 388, 390, 392, 393, 395, 397, 399, 401, 402, 404, 406, 408, 410, 412, 413, 415, 417, 427
- AuthFile** (class in ccrauth), 13, 15, 16, 18, 20, 22, 24, 25, 27, 29, 31, 33, 34, 36, 38, 40, 42, 43, 45, 47, 49, 51, 52, 54, 56, 58, 60, 61, 63, 65, 67, 69, 70, 72, 74, 76, 78, 79, 81, 83, 85, 87, 88, 90, 92, 94, 96, 97, 99, 101, 103, 105, 106, 108, 110, 112, 114, 115, 117, 119, 121, 123, 124, 126, 128, 130, 132, 133, 135, 137, 139, 141, 142, 144, 146,

148, 150, 151, 153, 155, 157, 159, 160, 162, 164, 166, 168, 169, 171, 173, 175, 177, 178, 180, 182, 184, 186, 187, 189, 191, 193, 195, 196, 198, 200, 202, 204, 205, 207, 209, 211, 213, 214, 216, 218, 220, 222, 223, 225, 227, 229, 231, 232, 234, 236, 238, 240, 241, 243, 245, 247, 249, 250, 252, 254, 256, 258, 259, 261, 263, 265, 267, 268, 270, 272, 274, 276, 277, 279, 281, 283, 285, 286, 288, 290, 292, 294, 295, 297, 299, 301, 303, 304, 306, 308, 310, 312, 313, 315, 317, 319, 321, 322, 324, 326, 328, 330, 331, 333, 335, 337, 339, 340, 342, 344, 346, 348, 349, 351, 353, 355, 357, 358, 360, 362, 364, 366, 367, 369, 371, 373, 375, 376, 378, 380, 382, 384, 385, 387, 389, 391, 393, 394, 396, 398, 400, 402, 403, 405, 407, 409, 411, 412, 414, 416, 418, 429

C

ccr (module), 1–11

CCRWarning, 13, 15, 17, 18, 20, 22, 24, 26, 27, 29, 31, 33, 35, 36, 38, 40, 42, 44, 45, 47, 49, 51, 53, 54, 56, 58, 60, 62, 63, 65, 67, 69, 71, 72, 74, 76, 78, 80, 81, 83, 85, 87, 89, 90, 92, 94, 96, 98, 99, 101, 103, 105, 107, 108, 110, 112, 114, 116, 117, 119, 121, 123, 125, 126, 128, 130, 132, 134, 135, 137, 139, 141, 143, 144, 146, 148, 150, 152, 153, 155, 157, 159, 161, 162, 164, 166, 168, 170, 171, 173, 175, 177, 179, 180, 182, 184, 186, 188, 189, 191, 193, 195, 197, 198, 200, 202, 204, 206, 207, 209, 211, 213, 215, 216, 218, 220, 222, 224, 225, 227, 229, 231, 233, 234, 236, 238, 240, 242, 243, 245, 247, 249, 251, 252, 254, 256, 258, 260, 261, 263, 265, 267, 269, 270, 272, 274, 276, 278, 279, 281, 283, 285, 287, 288, 290, 292, 294, 296, 297, 299, 301, 303, 305, 306, 308, 310, 312, 314, 315, 317, 319, 321, 323, 324, 326, 328, 330, 332, 333, 335, 337, 339, 341, 342, 344, 346, 348, 350, 351, 353, 355, 357, 359, 360, 362, 364, 366, 368, 369, 371, 373, 375, 377, 378, 380, 382, 384, 386, 387, 389, 391, 393, 395, 396, 398, 400, 402, 404, 405, 407, 409, 411, 413, 415, 417, 419, 431

check_vote() (ccr.Session method), 12, 14, 16, 17, 19, 21, 23, 24, 26, 28, 30, 32, 33, 35, 37, 39, 41, 42, 44, 46, 48, 50, 51, 53, 55, 57, 59, 60, 62, 64, 66, 68, 69, 71, 73, 75, 77, 78, 80, 82, 84, 86, 87, 89, 91, 93, 95, 96, 98, 100, 102, 104, 105, 107, 109, 111, 113, 114, 116, 118, 120, 122, 123, 125, 127, 129, 131, 132, 134, 136, 138, 140, 141, 143, 145, 147, 149, 150, 152, 154, 156, 158, 159, 161, 163, 165, 167, 168, 170, 172, 174, 176, 177, 179, 181, 183, 185, 186, 188,

190, 192, 194, 195, 197, 199, 201, 203, 204, 206, 208, 210, 212, 213, 215, 217, 219, 221, 222, 224, 226, 228, 230, 231, 233, 235, 237, 239, 240, 242, 244, 246, 248, 249, 251, 253, 255, 257, 258, 260, 262, 264, 266, 267, 269, 271, 273, 275, 276, 278, 280, 282, 284, 285, 287, 289, 291, 293, 294, 296, 298, 300, 302, 303, 305, 307, 309, 311, 312, 314, 316, 318, 320, 321, 323, 325, 327, 329, 330, 332, 334, 336, 338, 339, 341, 343, 345, 347, 348, 350, 352, 354, 356, 357, 359, 361, 363, 365, 366, 368, 370, 372, 374, 375, 377, 379, 381, 383, 384, 386, 388, 390, 392, 393, 395, 397, 399, 401, 402, 404, 406, 408, 410, 412, 414, 415, 418, 427

close() (ccr.Session method), 12, 14, 16, 17, 19, 21, 23, 25, 26, 28, 30, 32, 34, 35, 37, 39, 41, 43, 44, 46, 48, 50, 52, 53, 55, 57, 59, 61, 62, 64, 66, 68, 70, 71, 73, 75, 77, 79, 80, 82, 84, 86, 88, 89, 91, 93, 95, 97, 98, 100, 102, 104, 106, 107, 109, 111, 113, 115, 116, 118, 120, 122, 124, 125, 127, 129, 131, 133, 134, 136, 138, 140, 142, 143, 145, 147, 149, 151, 152, 154, 156, 158, 160, 161, 163, 165, 167, 169, 170, 172, 174, 176, 178, 179, 181, 183, 185, 187, 188, 190, 192, 194, 196, 197, 199, 201, 203, 205, 206, 208, 210, 212, 214, 215, 217, 219, 221, 223, 224, 226, 228, 230, 232, 233, 235, 237, 239, 241, 242, 244, 246, 248, 250, 251, 253, 255, 257, 259, 260, 262, 264, 266, 268, 269, 271, 273, 275, 277, 278, 280, 282, 284, 286, 287, 289, 291, 293, 295, 296, 298, 300, 302, 304, 305, 307, 309, 311, 313, 314, 316, 318, 320, 322, 323, 325, 327, 329, 331, 332, 334, 336, 338, 340, 341, 343, 345, 347, 349, 350, 352, 354, 356, 358, 359, 361, 363, 365, 367, 368, 370, 372, 374, 376, 377, 379, 381, 383, 385, 386, 388, 390, 392, 394, 395, 397, 399, 401, 403, 404, 406, 408, 410, 412, 414, 416, 418, 427

D

delete() (ccr.Session method), 12, 14, 16, 17, 19, 21, 23, 25, 26, 28, 30, 32, 34, 35, 37, 39, 41, 43, 44, 46, 48, 50, 52, 53, 55, 57, 59, 61, 62, 64, 66, 68, 70, 71, 73, 75, 77, 79, 80, 82, 84, 86, 88, 89, 91, 93, 95, 97, 98, 100, 102, 104, 106, 107, 109, 111, 113, 115, 116, 118, 120, 122, 124, 125, 127, 129, 131, 133, 134, 136, 138, 140, 142, 143, 145, 147, 149, 151, 152, 154, 156, 158, 160, 161, 163, 165, 167, 169, 170, 172, 174, 176, 178, 179, 181, 183, 185, 187, 188, 190, 192, 194, 196, 197, 199, 201, 203, 205, 206, 208, 210, 212, 214, 215, 217, 219, 221, 223, 224, 226, 228, 230, 232, 233, 235, 237, 239, 241,

242, 244, 246, 248, 250, 251, 253, 255, 257,
259, 260, 262, 264, 266, 268, 269, 271, 273,
275, 277, 278, 280, 282, 284, 286, 287, 289,
291, 293, 295, 296, 298, 300, 302, 304, 305,
307, 309, 311, 313, 314, 316, 318, 320, 322,
323, 325, 327, 329, 331, 332, 334, 336, 338,
340, 341, 343, 345, 347, 349, 350, 352, 354,
356, 358, 359, 361, 363, 365, 367, 368, 370,
372, 374, 376, 377, 379, 381, 383, 385, 386,
388, 390, 392, 394, 395, 397, 399, 401, 403,
404, 406, 408, 410, 412, 414, 416, 418, 427

`disown()` (`ccr.Session` method), 12, 14, 16, 17, 19, 21, 23,
25, 26, 28, 30, 32, 34, 35, 37, 39, 41, 43, 44, 46,
48, 50, 52, 53, 55, 57, 59, 61, 62, 64, 66, 68, 70,
71, 73, 75, 77, 79, 80, 82, 84, 86, 88, 89, 91, 93,
95, 97, 98, 100, 102, 104, 106, 107, 109, 111,
113, 115, 116, 118, 120, 122, 124, 125, 127,
129, 131, 133, 134, 136, 138, 140, 142, 143,
145, 147, 149, 151, 152, 154, 156, 158, 160,
161, 163, 165, 167, 169, 170, 172, 174, 176,
178, 179, 181, 183, 185, 187, 188, 190, 192,
194, 196, 197, 199, 201, 203, 205, 206, 208,
210, 212, 214, 215, 217, 219, 221, 223, 224,
226, 228, 230, 232, 233, 235, 237, 239, 241,
242, 244, 246, 248, 250, 251, 253, 255, 257,
259, 260, 262, 264, 266, 268, 269, 271, 273,
275, 277, 278, 280, 282, 284, 286, 287, 289,
291, 293, 295, 296, 298, 300, 302, 304, 305,
307, 309, 311, 313, 314, 316, 318, 320, 322,
323, 325, 327, 329, 331, 332, 334, 336, 338,
340, 341, 343, 345, 347, 349, 350, 352, 354,
356, 358, 359, 361, 363, 365, 367, 368, 370,
372, 374, 376, 377, 379, 381, 383, 385, 386,
388, 390, 392, 394, 395, 397, 399, 401, 403,
404, 406, 408, 410, 412, 414, 416, 418, 427

F

`flag()` (`ccr.Session` method), 12, 14, 16, 18, 19, 21, 23, 25,
26, 28, 30, 32, 34, 35, 37, 39, 41, 43, 44, 46, 48,
50, 52, 53, 55, 57, 59, 61, 62, 64, 66, 68, 70, 71,
73, 75, 77, 79, 80, 82, 84, 86, 88, 89, 91, 93, 95,
97, 98, 100, 102, 104, 106, 107, 109, 111, 113,
115, 116, 118, 120, 122, 124, 125, 127, 129,
131, 133, 134, 136, 138, 140, 142, 143, 145,
147, 149, 151, 152, 154, 156, 158, 160, 161,
163, 165, 167, 169, 170, 172, 174, 176, 178,
179, 181, 183, 185, 187, 188, 190, 192, 194,
196, 197, 199, 201, 203, 205, 206, 208, 210,
212, 214, 215, 217, 219, 221, 223, 224, 226,
228, 230, 232, 233, 235, 237, 239, 241, 242,
244, 246, 248, 250, 251, 253, 255, 257, 259,
260, 262, 264, 266, 268, 269, 271, 273, 275,
277, 278, 280, 282, 284, 286, 287, 289, 291,
293, 295, 296, 298, 300, 302, 304, 305, 307,

309, 311, 313, 314, 316, 318, 320, 322, 323,
325, 327, 329, 331, 332, 334, 336, 338, 340,
341, 343, 345, 347, 349, 350, 352, 354, 356,
358, 359, 361, 363, 365, 367, 368, 370, 372,
374, 376, 377, 379, 381, 383, 385, 386, 388,
390, 392, 394, 395, 397, 399, 401, 403, 404,
406, 408, 410, 412, 414, 416, 418, 427

I

`info()` (in module `ccr`), 12, 13, 15, 17, 19, 21, 22, 24, 26,
28, 29, 31, 33, 35, 37, 38, 40, 42, 44, 46, 47,
49, 51, 53, 55, 56, 58, 60, 62, 64, 65, 67, 69, 71,
73, 74, 76, 78, 80, 82, 83, 85, 87, 89, 91, 92, 94,
96, 98, 100, 101, 103, 105, 107, 109, 110, 112,
114, 116, 118, 119, 121, 123, 125, 127, 128,
130, 132, 134, 136, 137, 139, 141, 143, 145,
146, 148, 150, 152, 154, 155, 157, 159, 161,
163, 164, 166, 168, 170, 172, 173, 175, 177,
179, 181, 182, 184, 186, 188, 190, 191, 193,
195, 197, 199, 200, 202, 204, 206, 208, 209,
211, 213, 215, 217, 218, 220, 222, 224, 226,
227, 229, 231, 233, 235, 236, 238, 240, 242,
244, 245, 247, 249, 251, 253, 254, 256, 258,
260, 262, 263, 265, 267, 269, 271, 272, 274,
276, 278, 280, 281, 283, 285, 287, 289, 290,
292, 294, 296, 298, 299, 301, 303, 305, 307,
308, 310, 312, 314, 316, 317, 319, 321, 323,
325, 326, 328, 330, 332, 334, 335, 337, 339,
341, 343, 344, 346, 348, 350, 352, 353, 355,
357, 359, 361, 362, 364, 366, 368, 370, 371,
373, 375, 377, 379, 380, 382, 384, 386, 388,
389, 391, 393, 395, 397, 398, 400, 402, 404,
406, 407, 409, 411, 413, 415, 417, 425

`InvalidPackage`, 13, 15, 17, 18, 20, 22, 24, 26, 27, 29, 31,
33, 35, 36, 38, 40, 42, 44, 45, 47, 49, 51, 53,
54, 56, 58, 60, 62, 63, 65, 67, 69, 71, 72, 74,
76, 78, 80, 81, 83, 85, 87, 89, 90, 92, 94, 96,
98, 99, 101, 103, 105, 107, 108, 110, 112, 114,
116, 117, 119, 121, 123, 125, 126, 128, 130,
132, 134, 135, 137, 139, 141, 143, 144, 146,
148, 150, 152, 153, 155, 157, 159, 161, 162,
164, 166, 168, 170, 171, 173, 175, 177, 179,
180, 182, 184, 186, 188, 189, 191, 193, 195,
197, 198, 200, 202, 204, 206, 207, 209, 211,
213, 215, 216, 218, 220, 222, 224, 225, 227,
229, 231, 233, 234, 236, 238, 240, 242, 243,
245, 247, 249, 251, 252, 254, 256, 258, 260,
261, 263, 265, 267, 269, 270, 272, 274, 276,
278, 279, 281, 283, 285, 287, 288, 290, 292,
294, 296, 297, 299, 301, 303, 305, 306, 308,
310, 312, 314, 315, 317, 319, 321, 323, 324,
326, 328, 330, 332, 333, 335, 337, 339, 341,
342, 344, 346, 348, 350, 351, 353, 355, 357,
359, 360, 362, 364, 366, 368, 369, 371, 373,

375, 377, 378, 380, 382, 384, 386, 387, 389,
391, 393, 395, 396, 398, 400, 402, 404, 405,
407, 409, 411, 413, 415, 417, 419, 431

L

`list_orphans()` (in module `ccr`), 12, 13, 15, 17, 19, 21, 22,
24, 26, 28, 30, 31, 33, 35, 37, 39, 40, 42, 44, 46,
48, 49, 51, 53, 55, 57, 58, 60, 62, 64, 66, 67, 69,
71, 73, 75, 76, 78, 80, 82, 84, 85, 87, 89, 91, 93,
94, 96, 98, 100, 102, 103, 105, 107, 109, 111,
112, 114, 116, 118, 120, 121, 123, 125, 127,
129, 130, 132, 134, 136, 138, 139, 141, 143,
145, 147, 148, 150, 152, 154, 156, 157, 159,
161, 163, 165, 166, 168, 170, 172, 174, 175,
177, 179, 181, 183, 184, 186, 188, 190, 192,
193, 195, 197, 199, 201, 202, 204, 206, 208,
210, 211, 213, 215, 217, 219, 220, 222, 224,
226, 228, 229, 231, 233, 235, 237, 238, 240,
242, 244, 246, 247, 249, 251, 253, 255, 256,
258, 260, 262, 264, 265, 267, 269, 271, 273,
274, 276, 278, 280, 282, 283, 285, 287, 289,
291, 292, 294, 296, 298, 300, 301, 303, 305,
307, 309, 310, 312, 314, 316, 318, 319, 321,
323, 325, 327, 328, 330, 332, 334, 336, 337,
339, 341, 343, 345, 346, 348, 350, 352, 354,
355, 357, 359, 361, 363, 364, 366, 368, 370,
372, 373, 375, 377, 379, 381, 382, 384, 386,
388, 390, 391, 393, 395, 397, 399, 400, 402,
404, 406, 408, 409, 411, 413, 415, 417, 425

M

`msearch()` (in module `ccr`), 12, 13, 15, 17, 19, 21, 22, 24,
26, 28, 30, 31, 33, 35, 37, 39, 40, 42, 44, 46, 48,
49, 51, 53, 55, 57, 58, 60, 62, 64, 66, 67, 69, 71,
73, 75, 76, 78, 80, 82, 84, 85, 87, 89, 91, 93, 94,
96, 98, 100, 102, 103, 105, 107, 109, 111, 112,
114, 116, 118, 120, 121, 123, 125, 127, 129,
130, 132, 134, 136, 138, 139, 141, 143, 145,
147, 148, 150, 152, 154, 156, 157, 159, 161,
163, 165, 166, 168, 170, 172, 174, 175, 177,
179, 181, 183, 184, 186, 188, 190, 192, 193,
195, 197, 199, 201, 202, 204, 206, 208, 210,
211, 213, 215, 217, 219, 220, 222, 224, 226,
228, 229, 231, 233, 235, 237, 238, 240, 242,
244, 246, 247, 249, 251, 253, 255, 256, 258,
260, 262, 264, 265, 267, 269, 271, 273, 274,
276, 278, 280, 282, 283, 285, 287, 289, 291,
292, 294, 296, 298, 300, 301, 303, 305, 307,
309, 310, 312, 314, 316, 318, 319, 321, 323,
325, 327, 328, 330, 332, 334, 336, 337, 339,
341, 343, 345, 346, 348, 350, 352, 354, 355,
357, 359, 361, 363, 364, 366, 368, 370, 372,
373, 375, 377, 379, 381, 382, 384, 386, 388,
390, 391, 393, 395, 397, 399, 400, 402, 404,

406, 408, 409, 411, 413, 415, 417, 425

N

`notify()` (`ccr.Session` method), 12, 14, 16, 18, 19, 21, 23,
25, 27, 28, 30, 32, 34, 36, 37, 39, 41, 43, 45, 46,
48, 50, 52, 54, 55, 57, 59, 61, 63, 64, 66, 68, 70,
72, 73, 75, 77, 79, 81, 82, 84, 86, 88, 90, 91, 93,
95, 97, 99, 100, 102, 104, 106, 108, 109, 111,
113, 115, 117, 118, 120, 122, 124, 126, 127,
129, 131, 133, 135, 136, 138, 140, 142, 144,
145, 147, 149, 151, 153, 154, 156, 158, 160,
162, 163, 165, 167, 169, 171, 172, 174, 176,
178, 180, 181, 183, 185, 187, 189, 190, 192,
194, 196, 198, 199, 201, 203, 205, 207, 208,
210, 212, 214, 216, 217, 219, 221, 223, 225,
226, 228, 230, 232, 234, 235, 237, 239, 241,
243, 244, 246, 248, 250, 252, 253, 255, 257,
259, 261, 262, 264, 266, 268, 270, 271, 273,
275, 277, 279, 280, 282, 284, 286, 288, 289,
291, 293, 295, 297, 298, 300, 302, 304, 306,
307, 309, 311, 313, 315, 316, 318, 320, 322,
324, 325, 327, 329, 331, 333, 334, 336, 338,
340, 342, 343, 345, 347, 349, 351, 352, 354,
356, 358, 360, 361, 363, 365, 367, 369, 370,
372, 374, 376, 378, 379, 381, 383, 385, 387,
388, 390, 392, 394, 396, 397, 399, 401, 403,
405, 406, 408, 410, 412, 414, 416, 418, 427

P

`PackageNotFound`, 13, 15, 17, 18, 20, 22, 24, 26, 27, 29,
31, 33, 35, 36, 38, 40, 42, 44, 45, 47, 49, 51,
53, 54, 56, 58, 60, 62, 63, 65, 67, 69, 71, 72,
74, 76, 78, 80, 81, 83, 85, 87, 89, 90, 92, 94, 96,
98, 99, 101, 103, 105, 107, 108, 110, 112, 114,
116, 117, 119, 121, 123, 125, 126, 128, 130,
132, 134, 135, 137, 139, 141, 143, 144, 146,
148, 150, 152, 153, 155, 157, 159, 161, 162,
164, 166, 168, 170, 171, 173, 175, 177, 179,
180, 182, 184, 186, 188, 189, 191, 193, 195,
197, 198, 200, 202, 204, 206, 207, 209, 211,
213, 215, 216, 218, 220, 222, 224, 225, 227,
229, 231, 233, 234, 236, 238, 240, 242, 243,
245, 247, 249, 251, 252, 254, 256, 258, 260,
261, 263, 265, 267, 269, 270, 272, 274, 276,
278, 279, 281, 283, 285, 287, 288, 290, 292,
294, 296, 297, 299, 301, 303, 305, 306, 308,
310, 312, 314, 315, 317, 319, 321, 323, 324,
326, 328, 330, 332, 333, 335, 337, 339, 341,
342, 344, 346, 348, 350, 351, 353, 355, 357,
359, 360, 362, 364, 366, 368, 369, 371, 373,
375, 377, 378, 380, 382, 384, 386, 387, 389,
391, 393, 395, 396, 398, 400, 402, 404, 405,
407, 409, 411, 413, 415, 417, 419, 431

S

`search()` (in module `ccr`), 12, 13, 15, 17, 19, 21, 22, 24, 26, 28, 29, 31, 33, 35, 37, 38, 40, 42, 44, 46, 47, 49, 51, 53, 55, 56, 58, 60, 62, 64, 65, 67, 69, 71, 73, 74, 76, 78, 80, 82, 83, 85, 87, 89, 91, 92, 94, 96, 98, 100, 101, 103, 105, 107, 109, 110, 112, 114, 116, 118, 119, 121, 123, 125, 127, 128, 130, 132, 134, 136, 137, 139, 141, 143, 145, 146, 148, 150, 152, 154, 155, 157, 159, 161, 163, 164, 166, 168, 170, 172, 173, 175, 177, 179, 181, 182, 184, 186, 188, 190, 191, 193, 195, 197, 199, 200, 202, 204, 206, 208, 209, 211, 213, 215, 217, 218, 220, 222, 224, 226, 227, 229, 231, 233, 235, 236, 238, 240, 242, 244, 245, 247, 249, 251, 253, 254, 256, 258, 260, 262, 263, 265, 267, 269, 271, 272, 274, 276, 278, 280, 281, 283, 285, 287, 289, 290, 292, 294, 296, 298, 299, 301, 303, 305, 307, 308, 310, 312, 314, 316, 317, 319, 321, 323, 325, 326, 328, 330, 332, 334, 335, 337, 339, 341, 343, 344, 346, 348, 350, 352, 353, 355, 357, 359, 361, 362, 364, 366, 368, 370, 371, 373, 375, 377, 379, 380, 382, 384, 386, 388, 389, 391, 393, 395, 397, 398, 400, 402, 404, 406, 407, 409, 411, 413, 415, 417, 425

`Session` (class in `ccr`), 12, 14, 15, 17, 19, 21, 23, 24, 26, 28, 30, 32, 33, 35, 37, 39, 41, 42, 44, 46, 48, 50, 51, 53, 55, 57, 59, 60, 62, 64, 66, 68, 69, 71, 73, 75, 77, 78, 80, 82, 84, 86, 87, 89, 91, 93, 95, 96, 98, 100, 102, 104, 105, 107, 109, 111, 113, 114, 116, 118, 120, 122, 123, 125, 127, 129, 131, 132, 134, 136, 138, 140, 141, 143, 145, 147, 149, 150, 152, 154, 156, 158, 159, 161, 163, 165, 167, 168, 170, 172, 174, 176, 177, 179, 181, 183, 185, 186, 188, 190, 192, 194, 195, 197, 199, 201, 203, 204, 206, 208, 210, 212, 213, 215, 217, 219, 221, 222, 224, 226, 228, 230, 231, 233, 235, 237, 239, 240, 242, 244, 246, 248, 249, 251, 253, 255, 257, 258, 260, 262, 264, 266, 267, 269, 271, 273, 275, 276, 278, 280, 282, 284, 285, 287, 289, 291, 293, 294, 296, 298, 300, 302, 303, 305, 307, 309, 311, 312, 314, 316, 318, 320, 321, 323, 325, 327, 329, 330, 332, 334, 336, 338, 339, 341, 343, 345, 347, 348, 350, 352, 354, 356, 357, 359, 361, 363, 365, 366, 368, 370, 372, 374, 375, 377, 379, 381, 383, 384, 386, 388, 390, 392, 393, 395, 397, 399, 401, 402, 404, 406, 408, 410, 411, 413, 415, 417, 427

`setcategory()` (`ccr.Session` method), 12, 14, 16, 18, 19, 21, 23, 25, 27, 28, 30, 32, 34, 36, 37, 39, 41, 43, 45, 46, 48, 50, 52, 54, 55, 57, 59, 61, 63, 64, 66, 68, 70, 72, 73, 75, 77, 79, 81, 82, 84, 86, 88, 90, 91, 93, 95, 97, 99, 100, 102, 104, 106, 108,

109, 111, 113, 115, 117, 118, 120, 122, 124, 126, 127, 129, 131, 133, 135, 136, 138, 140, 142, 144, 145, 147, 149, 151, 153, 154, 156, 158, 160, 162, 163, 165, 167, 169, 171, 172, 174, 176, 178, 180, 181, 183, 185, 187, 189, 190, 192, 194, 196, 198, 199, 201, 203, 205, 207, 208, 210, 212, 214, 216, 217, 219, 221, 223, 225, 226, 228, 230, 232, 234, 235, 237, 239, 241, 243, 244, 246, 248, 250, 252, 253, 255, 257, 259, 261, 262, 264, 266, 268, 270, 271, 273, 275, 277, 279, 280, 282, 284, 286, 288, 289, 291, 293, 295, 297, 298, 300, 302, 304, 306, 307, 309, 311, 313, 315, 316, 318, 320, 322, 324, 325, 327, 329, 331, 333, 334, 336, 338, 340, 342, 343, 345, 347, 349, 351, 352, 354, 356, 358, 360, 361, 363, 365, 367, 369, 370, 372, 374, 376, 378, 379, 381, 383, 385, 387, 388, 390, 392, 394, 396, 397, 399, 401, 403, 405, 406, 408, 410, 412, 414, 416, 418, 427

`store_auth_info()` (`ccrauth.AuthDB` method), 13, 15, 16, 18, 20, 22, 24, 25, 27, 29, 31, 33, 34, 36, 38, 40, 42, 43, 45, 47, 49, 51, 52, 54, 56, 58, 60, 61, 63, 65, 67, 69, 70, 72, 74, 76, 78, 79, 81, 83, 85, 87, 88, 90, 92, 94, 96, 97, 99, 101, 103, 105, 106, 108, 110, 112, 114, 115, 117, 119, 121, 123, 124, 126, 128, 130, 132, 133, 135, 137, 139, 141, 142, 144, 146, 148, 150, 151, 153, 155, 157, 159, 160, 162, 164, 166, 168, 169, 171, 173, 175, 177, 178, 180, 182, 184, 186, 187, 189, 191, 193, 195, 196, 198, 200, 202, 204, 205, 207, 209, 211, 213, 214, 216, 218, 220, 222, 223, 225, 227, 229, 231, 232, 234, 236, 238, 240, 241, 243, 245, 247, 249, 250, 252, 254, 256, 258, 259, 261, 263, 265, 267, 268, 270, 272, 274, 276, 277, 279, 281, 283, 285, 286, 288, 290, 292, 294, 295, 297, 299, 301, 303, 304, 306, 308, 310, 312, 313, 315, 317, 319, 321, 322, 324, 326, 328, 330, 331, 333, 335, 337, 339, 340, 342, 344, 346, 348, 349, 351, 353, 355, 357, 358, 360, 362, 364, 366, 367, 369, 371, 373, 375, 376, 378, 380, 382, 384, 385, 387, 389, 391, 393, 394, 396, 398, 400, 402, 403, 405, 407, 409, 411, 413, 415, 416, 419, 429

`store_auth_info()` (`ccrauth.AuthFile` method), 13, 15, 16, 18, 20, 22, 24, 25, 27, 29, 31, 33, 34, 36, 38, 40, 42, 43, 45, 47, 49, 51, 52, 54, 56, 58, 60, 61, 63, 65, 67, 69, 70, 72, 74, 76, 78, 79, 81, 83, 85, 87, 88, 90, 92, 94, 96, 97, 99, 101, 103, 105, 106, 108, 110, 112, 114, 115, 117, 119, 121, 123, 124, 126, 128, 130, 132, 133, 135, 137, 139, 141, 142, 144, 146, 148, 150, 151, 153, 155, 157, 159, 160, 162, 164, 166, 168, 169, 171,

173, 175, 177, 178, 180, 182, 184, 186, 187, 189, 191, 193, 195, 196, 198, 200, 202, 204, 205, 207, 209, 211, 213, 214, 216, 218, 220, 222, 223, 225, 227, 229, 231, 232, 234, 236, 238, 240, 241, 243, 245, 247, 249, 250, 252, 254, 256, 258, 259, 261, 263, 265, 267, 268, 270, 272, 274, 276, 277, 279, 281, 283, 285, 286, 288, 290, 292, 294, 295, 297, 299, 301, 303, 304, 306, 308, 310, 312, 313, 315, 317, 319, 321, 322, 324, 326, 328, 330, 331, 333, 335, 337, 339, 340, 342, 344, 346, 348, 349, 351, 353, 355, 357, 358, 360, 362, 364, 366, 367, 369, 371, 373, 375, 376, 378, 380, 382, 384, 385, 387, 389, 391, 393, 394, 396, 398, 400, 402, 403, 405, 407, 409, 411, 412, 414, 416, 418, 429

`submit()` (`ccr.Session` method), 12, 14, 16, 18, 20, 21, 23, 25, 27, 28, 30, 32, 34, 36, 37, 39, 41, 43, 45, 46, 48, 50, 52, 54, 55, 57, 59, 61, 63, 64, 66, 68, 70, 72, 73, 75, 77, 79, 81, 82, 84, 86, 88, 90, 91, 93, 95, 97, 99, 100, 102, 104, 106, 108, 109, 111, 113, 115, 117, 118, 120, 122, 124, 126, 127, 129, 131, 133, 135, 136, 138, 140, 142, 144, 145, 147, 149, 151, 153, 154, 156, 158, 160, 162, 163, 165, 167, 169, 171, 172, 174, 176, 178, 180, 181, 183, 185, 187, 189, 190, 192, 194, 196, 198, 199, 201, 203, 205, 207, 208, 210, 212, 214, 216, 217, 219, 221, 223, 225, 226, 228, 230, 232, 234, 235, 237, 239, 241, 243, 244, 246, 248, 250, 252, 253, 255, 257, 259, 261, 262, 264, 266, 268, 270, 271, 273, 275, 277, 279, 280, 282, 284, 286, 288, 289, 291, 293, 295, 297, 298, 300, 302, 304, 306, 307, 309, 311, 313, 315, 316, 318, 320, 322, 324, 325, 327, 329, 331, 333, 334, 336, 338, 340, 342, 343, 345, 347, 349, 351, 352, 354, 356, 358, 360, 361, 363, 365, 367, 369, 370, 372, 374, 376, 378, 379, 381, 383, 385, 387, 388, 390, 392, 394, 396, 397, 399, 401, 403, 405, 406, 408, 410, 412, 414, 416, 418, 427

U

`unflag()` (`ccr.Session` method), 12, 14, 16, 18, 20, 21, 23, 25, 27, 29, 30, 32, 34, 36, 38, 39, 41, 43, 45, 47, 48, 50, 52, 54, 56, 57, 59, 61, 63, 65, 66, 68, 70, 72, 74, 75, 77, 79, 81, 83, 84, 86, 88, 90, 92, 93, 95, 97, 99, 101, 102, 104, 106, 108, 110, 111, 113, 115, 117, 119, 120, 122, 124, 126, 128, 129, 131, 133, 135, 137, 138, 140, 142, 144, 146, 147, 149, 151, 153, 155, 156, 158, 160, 162, 164, 165, 167, 169, 171, 173, 174, 176, 178, 180, 182, 183, 185, 187, 189, 191, 192, 194, 196, 198, 200, 201, 203, 205, 207, 209, 210, 212, 214, 216, 218, 219, 221, 223, 225,

227, 228, 230, 232, 234, 236, 237, 239, 241, 243, 245, 246, 248, 250, 252, 254, 255, 257, 259, 261, 263, 264, 266, 268, 270, 272, 273, 275, 277, 279, 281, 282, 284, 286, 288, 290, 291, 293, 295, 297, 299, 300, 302, 304, 306, 308, 309, 311, 313, 315, 317, 318, 320, 322, 324, 326, 327, 329, 331, 333, 335, 336, 338, 340, 342, 344, 345, 347, 349, 351, 353, 354, 356, 358, 360, 362, 363, 365, 367, 369, 371, 372, 374, 376, 378, 380, 381, 383, 385, 387, 389, 390, 392, 394, 396, 398, 399, 401, 403, 405, 407, 408, 410, 412, 414, 416, 418, 427

`unnotify()` (`ccr.Session` method), 13, 14, 16, 18, 20, 21, 23, 25, 27, 29, 30, 32, 34, 36, 38, 39, 41, 43, 45, 47, 48, 50, 52, 54, 56, 57, 59, 61, 63, 65, 66, 68, 70, 72, 74, 75, 77, 79, 81, 83, 84, 86, 88, 90, 92, 93, 95, 97, 99, 101, 102, 104, 106, 108, 110, 111, 113, 115, 117, 119, 120, 122, 124, 126, 128, 129, 131, 133, 135, 137, 138, 140, 142, 144, 146, 147, 149, 151, 153, 155, 156, 158, 160, 162, 164, 165, 167, 169, 171, 173, 174, 176, 178, 180, 182, 183, 185, 187, 189, 191, 192, 194, 196, 198, 200, 201, 203, 205, 207, 209, 210, 212, 214, 216, 218, 219, 221, 223, 225, 227, 228, 230, 232, 234, 236, 237, 239, 241, 243, 245, 246, 248, 250, 252, 254, 255, 257, 259, 261, 263, 264, 266, 268, 270, 272, 273, 275, 277, 279, 281, 282, 284, 286, 288, 290, 291, 293, 295, 297, 299, 300, 302, 304, 306, 308, 309, 311, 313, 315, 317, 318, 320, 322, 324, 326, 327, 329, 331, 333, 335, 336, 338, 340, 342, 344, 345, 347, 349, 351, 353, 354, 356, 358, 360, 362, 363, 365, 367, 369, 371, 372, 374, 376, 378, 380, 381, 383, 385, 387, 389, 390, 392, 394, 396, 398, 399, 401, 403, 405, 407, 408, 410, 412, 414, 416, 418, 428

`unvote()` (`ccr.Session` method), 13, 14, 16, 18, 20, 22, 23, 25, 27, 29, 30, 32, 34, 36, 38, 39, 41, 43, 45, 47, 48, 50, 52, 54, 56, 57, 59, 61, 63, 65, 66, 68, 70, 72, 74, 75, 77, 79, 81, 83, 84, 86, 88, 90, 92, 93, 95, 97, 99, 101, 102, 104, 106, 108, 110, 111, 113, 115, 117, 119, 120, 122, 124, 126, 128, 129, 131, 133, 135, 137, 138, 140, 142, 144, 146, 147, 149, 151, 153, 155, 156, 158, 160, 162, 164, 165, 167, 169, 171, 173, 174, 176, 178, 180, 182, 183, 185, 187, 189, 191, 192, 194, 196, 198, 200, 201, 203, 205, 207, 209, 210, 212, 214, 216, 218, 219, 221, 223, 225, 227, 228, 230, 232, 234, 236, 237, 239, 241, 243, 245, 246, 248, 250, 252, 254, 255, 257, 259, 261, 263, 264, 266, 268, 270, 272, 273, 275, 277, 279, 281, 282, 284, 286, 288, 290, 291, 293, 295, 297, 299, 300, 302, 304, 306,

308, 309, 311, 313, 315, 317, 318, 320, 322,
324, 326, 327, 329, 331, 333, 335, 336, 338,
340, 342, 344, 345, 347, 349, 351, 353, 354,
356, 358, 360, 362, 363, 365, 367, 369, 371,
372, 374, 376, 378, 380, 381, 383, 385, 387,
389, 390, 392, 394, 396, 398, 399, 401, 403,
405, 407, 408, 410, 412, 414, 416, 418, 428

V

vote() (ccr.Session method), 13, 14, 16, 18, 20, 22, 23, 25,
27, 29, 31, 32, 34, 36, 38, 40, 41, 43, 45, 47, 49,
50, 52, 54, 56, 58, 59, 61, 63, 65, 67, 68, 70, 72,
74, 76, 77, 79, 81, 83, 85, 86, 88, 90, 92, 94, 95,
97, 99, 101, 103, 104, 106, 108, 110, 112, 113,
115, 117, 119, 121, 122, 124, 126, 128, 130,
131, 133, 135, 137, 139, 140, 142, 144, 146,
148, 149, 151, 153, 155, 157, 158, 160, 162,
164, 166, 167, 169, 171, 173, 175, 176, 178,
180, 182, 184, 185, 187, 189, 191, 193, 194,
196, 198, 200, 202, 203, 205, 207, 209, 211,
212, 214, 216, 218, 220, 221, 223, 225, 227,
229, 230, 232, 234, 236, 238, 239, 241, 243,
245, 247, 248, 250, 252, 254, 256, 257, 259,
261, 263, 265, 266, 268, 270, 272, 274, 275,
277, 279, 281, 283, 284, 286, 288, 290, 292,
293, 295, 297, 299, 301, 302, 304, 306, 308,
310, 311, 313, 315, 317, 319, 320, 322, 324,
326, 328, 329, 331, 333, 335, 337, 338, 340,
342, 344, 346, 347, 349, 351, 353, 355, 356,
358, 360, 362, 364, 365, 367, 369, 371, 373,
374, 376, 378, 380, 382, 383, 385, 387, 389,
391, 392, 394, 396, 398, 400, 401, 403, 405,
407, 409, 410, 412, 414, 416, 418, 428