
dlmanager Documentation

Release 0.1.1

Julien Pagès

February 19, 2016

1	API	3
1.1	DownloadManager	3
1.2	Download	4
1.3	PersistLimit	5
2	Indices and tables	7

dlmanager is a Python 2 and 3 download manager library. It is hosted [on github](#).

Contents:

1.1 DownloadManager

class `dlmanager.DownloadManager` (*destdir*, *session=None*, *persist_limit=None*)

`DownloadManager` is responsible of starting and managing downloads inside a given directory. It will download a file only if a given filename is not already there.

Note that background downloads needs to be stopped. For example, if you have an exception while a download is occurring, python will only exit when the download will finish. To get rid of that, there is a possible idiom:

```
def download_things(manager):
    # do things with the manager
    manager.download(url1, f1)
    manager.download(url2, f2)
    ...

manager = DownloadManager(destdir)
try:
    download_things(manager)
finally:
    # ensure we cancel all background downloads to ask the end
    # of possible remainings threads
    manager.cancel()
```

Parameters

- **destdir** – a directory where files are downloaded. It will be created if it does not exists.
- **session** – a requests session. If `None`, one will be created for you.
- **persist_limit** – an instance of `PersistLimit`, to allow limiting the size of the download dir. Defaults to `None`, meaning no limit.

cancel (*cancel_if=None*)

Cancel downloads, if any.

if `cancel_if` is given, it must be a callable that take the download instance as parameter, and return `True` if the download needs to be canceled.

Note that download threads won't be stopped directly.

download (*url*, *fname=None*, *progress=None*)

Returns a started `Download` instance, or `None` if `fname` is already present in `destdir`.

if a download is already running for the given fname, it is just returned. Else the download is created, started and returned.

Parameters

- **url** – url of the file to download.
- **fname** – name to give for the downloaded file. If None, it will be the name extracted in the url.
- **progress** – a callable to report the download progress, or None. See `Download.set_progress()`.

wait (*raise_if_error=True*)

Wait for all downloads to be finished.

1.2 Download

class dlmanager.**Download**(*url, dest, finished_callback=None, chunk_size=16384, session=None, progress=None*)

Download is responsible of downloading one file in the background.

Example of use:

```
dl = Download(url, dest)
dl.start()
dl.wait() # this will block until completion / cancel / error
```

If a download fail or is canceled, the temporary dest is removed from the disk.

Usually, Downloads are created by using `DownloadManager.download()`.

Parameters

- **url** – the url of the file to download
- **dest** – the local file path destination
- **finished_callback** – a callback that will be called in the thread when the thread work is done. Takes the download instance as a parameter.
- **chunk_size** – size of the chunk that will be read. The thread can not be stopped while we are reading that chunk size.
- **session** – a `requests.Session` instance that will do the real downloading work. If None, `requests` module is used.
- **progress** – A callable to report the progress (default to None). see `set_progress()`.

cancel ()

Cancel a previously started download.

error ()

Returns None or a tuple of three values (type, value, traceback) that give information about the exception.

get_dest ()

Returns the dest.

get_url ()

Returns the url.

is_canceled ()

Returns True if we canceled this download.

is_running()

Returns True if the downloading thread is running.

raise_if_error()

Raise an error if any. If the download was canceled, raise *DownloadInterrupt*.

set_progress(*progress*)

set a callable to report the progress of the download, or None to disable any report.

The callable must take three parameters (download, current, total). Note that this method is thread safe, you can call it during a download.

start()

Start the thread that will do the download.

wait(*raise_if_error=True*)

Block until the downloading thread is finished.

Parameters **raise_if_error** – if True (the default), *raise_if_error()* will be called and raise an error if any.

class `dlmanager.DownloadInterrupt`

Raised when a download is interrupted.

1.3 PersistLimit

class `dlmanager.PersistLimit(size_limit, file_limit=5)`

Keep a list of files, removing the oldest ones when the *size_limit* is reached.

The access time of a file is used to determine the oldests, e.g. the last time a file was read.

Parameters

- **size_limit** – the size limit in bytes. A value of 0 means no limit.
- **file_limit** – even if the size limit is reached, this force to keep at least *file_limit* files.

register_dir_content(*directory*, *pattern*='*')

Register every files in a directory that match *pattern*.

register_file(*path*)

register a single file.

remove_old_files()

remove oldest registered files.

Indices and tables

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

C

cancel() (dlmanager.Download method), 4
cancel() (dlmanager.DownloadManager method), 3

D

Download (class in dlmanager), 4
download() (dlmanager.DownloadManager method), 3
DownloadInterrupt (class in dlmanager), 5
DownloadManager (class in dlmanager), 3

E

error() (dlmanager.Download method), 4

G

get_dest() (dlmanager.Download method), 4
get_url() (dlmanager.Download method), 4

I

is_canceled() (dlmanager.Download method), 4
is_running() (dlmanager.Download method), 4

P

PersistLimit (class in dlmanager), 5

R

raise_if_error() (dlmanager.Download method), 5
register_dir_content() (dlmanager.PersistLimit method),
5
register_file() (dlmanager.PersistLimit method), 5
remove_old_files() (dlmanager.PersistLimit method), 5

S

set_progress() (dlmanager.Download method), 5
start() (dlmanager.Download method), 5

W

wait() (dlmanager.Download method), 5
wait() (dlmanager.DownloadManager method), 4