
NIWLittleUtils Documentation

Release v0.1.0.dev10

NIWyclin (a.k.a. birdhackor)

Dec 17, 2017

Contents

1	Intro	3
2	Contents	5
2.1	codec — A module that help handle codec.	5
2.2	filesystem — A module that contain some helpful function to deal with filesystem.	6
2.3	random_tool — A module to generate things randomly.	6
3	Indices and tables	9
	Python Module Index	11

State:

Documentation: <https://niwlittleutils.readthedocs.org/>

Repo: <https://bitbucket.org/birdhackor/niwlittleutils>

CHAPTER 1

Intro

NIWLittleUtils is a tool collection that contain some useful module and function.

This library is a work in progress. Please give feedback!

2.1 codec — A module that help handle codec.

`NIWLittleUtils.codec.b64decode(b)`

Decode a Base64 encoded byte string.

Decode a Base64 encoded byte string using a URL-safe alphabet, which substitutes `-` instead of `+` and `_` instead of `/` in the standard Base64 alphabet. The input `b` may also be a ASCII-only Unicode strings.

Return type bytes

`NIWLittleUtils.codec.b64encode(b)`

Base64 encodes a byte string.

Base64 encodes a byte string. The resulting bytestring is safe for putting into URLs. The input `b` should be a `bytes` or `bytearray` object in Python3.3 or any `bytes-like object` after Python3.4. This function will raise `TypeError` if `b` is not `bytes-like object` (For example, `str`).

Return type bytes

`NIWLittleUtils.codec.urlsafe_decode(s)`

Decode a string that encode by `urlsafe_encode()`.

Decode a string that encoded by `urlsafe_encode()`. If the input `s` which returned by `urlsafe_encode()` is encode from `bytes` or `bytearray` object, it will return `bytes` object. If `s` is encode from `str` object, it will return `str` object.

Return type str or bytes

`NIWLittleUtils.codec.urlsafe_encode(s)`

Encode a string or byte string to url-safe string.

Encode a string or byte string to url-safe string. The input can be a object of `str`, `bytes` or `bytearray`.

This function will make sure that return string contains characters that *RFC 3986 section 2.3* specified only, but it won't check the length. User should remember that browsers have limit for length of URLs. (For example, IE8's maximum URL length is 2083 chars)

Return type str

2.2 filesystem — A module that contain some helpful function to deal with filesystem.

`NIWLittleUtils.filesystem.scantree(path='.')`

Recursively yield `DirEntry` objects for given directory.

Python 3.5 and higher version has `os.scandir()` to get the files and folders and their file attribute information in folder given by *path*. But Python version won't `scandir` recursively, this function can scan folder recursively.

Notice that this function will not follow symlinks to avoid symlink loops.

Parameters *path* – Path to folder which you want to scan.

`NIWLittleUtils.filesystem.secure_filename(filename, codec='utf8')`

Pass it a filename and it will return a secure version of it.

This function is a modified version of `werkzeug.utils.secure_filename`.

The filename can then safely be stored on a regular file system and passed to `os.path.join()`.

You can use parameter *codec* to specify the codec which is used to encode the filename. The *codec* could only be *utf8* or *ascii*. If you need high portability, you should let *codec* to be *'ascii'*. It will be *'utf8'* by default.

On windows systems the function also makes sure that the file is not named after one of the special device files.

```
>>> secure_filename("My cool movie.mov")
'My_cool_movie.mov'
>>> secure_filename("../../etc/passwd")
'etc_passwd'
>>> secure_filename('i contain cool \xfcml\xe4uts.txt')
'i_contain_cool_umläuts.txt'
>>> secure_filename('i contain cool \xfcml\xe4uts.txt', 'ascii')
'i_contain_cool_umlauts.txt'
```

The function might return an empty filename. It's your responsibility to ensure that the filename is unique and that you generate random filename if the function returned an empty one.

User should remember, this function will make sure filename is secure, but it cannot make sure file itself is secure. For example, if a user upload a file named `test.sh`, this function will return same filename. If this file has security vulnerabilities script inside and you execute it... well, good luck.

Parameters *filename* – the filename to secure

2.3 random_tool — A module to generate things randomly.

`NIWLittleUtils.random_tool.get_random_string(length=36, allowed_chars='complex')`

Returns a securely generated random string.

Generate a securely random string with length or allowed characters you want.

If *allowed_chars* is ignored, it's `string.ascii_letters + string.digits + string.punctuation` by default (We use Python builtin library `string` to set value). You can set it to following string values or any *allowed_chars* else.

Value	Meaning
'complex'	string.ascii_letters + string.digits + string.punctuation
'middle'	string.ascii_letters + string.digits + '-._~'
'simple'	string.ascii_letters + string.digits
'lowercase'	string.ascii_lowercase
'uppercase'	string.ascii_uppercase
'digits'	string.digits

```
>>> get_random_string(8)
'Qi4Y_fA5'
```

```
>>> get_random_string(5, '0123456789abcdef')
'06aef'
```

```
>>> get_random_string(5, 'digits')
'05154'
```

Parameters

- **length** (*int*) – Random string length.
- **allowed_chars** (*str*) – Character allowed to use to generate. There has 6 default setting (case-insensitive).

Returns A securely generated random string.

Return type str

CHAPTER 3

Indices and tables

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

c

`codec`, 5

f

`filesystem`, 6

n

`NIWLittleUtils.codec`, 5

`NIWLittleUtils.filesystem`, 6

`NIWLittleUtils.random_tool`, 6

r

`random_tool`, 6

B

b64decode() (in module NIWLittleUtils.codec), 5

b64encode() (in module NIWLittleUtils.codec), 5

C

codec (module), 5

F

filesystem (module), 6

G

get_random_string() (in module NIWLittleUtils.random_tool), 6

N

NIWLittleUtils.codec (module), 5

NIWLittleUtils.filesystem (module), 6

NIWLittleUtils.random_tool (module), 6

R

random_tool (module), 6

S

scantree() (in module NIWLittleUtils.filesystem), 6

secure_filename() (in module NIWLittleUtils.filesystem),
6

U

urlsafe_decode() (in module NIWLittleUtils.codec), 5

urlsafe_encode() (in module NIWLittleUtils.codec), 5