
urlquick Documentation

Release 0.9.4

William Forde

Mar 10, 2021

Contents

1	Urlquick	3
2	Features	5
3	Install	7
4	Usage	9
5	API	11
5.1	Session Class	14
5.2	Response Object	17
5.3	Request Object	19
6	External Links	21
	Index	23

CHAPTER 1

Urlquick

A light-weight http client with requests like interface. Featuring persistent connections and caching support. This project was originally created for use by Kodi add-ons, but has grown into something more. I found, that while requests has a very nice interface, there was a noticeable lag when importing the library. The other option available is to use urllib2, but then you loose the benefit of persistent connections that requests have. Hence the reason for this project.

All GET, HEAD and POST requests are cached locally for a period of 4 hours. When the cache expires, conditional headers are added to a new request e.g. “Etag” and “Last-modified”. Then if the server returns a 304 Not-Modified response, the cache is reused, saving having to re-download the content body.

CHAPTER 2

Features

- Simple Keep-Alive & Connection Pooling
- Sessions with limited Cookie Controls
- International Domains and URLs
- Automatic Content Decoding
- Elegant Key/Value Cookies
- Automatic Decompression
- Unicode Response Bodies
- Basic Authentication
- Connection Timeouts
- Resource Caching

CHAPTER 3

Install

Run

```
pip install urlquick
```

-or-

```
pip install git+https://github.com/willforde/urlquick.git
```


CHAPTER 4

Usage

Urlquick is similar to the requests library but it only implements most top level methods like GET, POST and PUT. The Session class is also implemented in a more limited form. The response object is fully comparable with the ‘requests’ response object. # link request object

```
>>> import urlquick
>>> r = urlquick.get('https://api.github.com/events')
>>> r.status_code
200
>>> r.headers['content-type']
'text/html; charset=utf-8'
>>> r.encoding
'utf-8'
>>> r.content
b'[{"repository":{"open_issues":0,"url":"https://github.com/...
>>> r.text
u'[{"repository":{"open_issues":0,"url":"https://github.com/...
>>> r.json()
[{"repository": {"open_issues": 0, "url": "https://github.com/...
```


`urlquick.MAX_AGE = 14400`

The default max age of the cache in seconds is used when no max age is given in request.

`urlquick.CACHE_LOCATION`

Location of the cache files. Defaults to the current working directory. It Will create a “.cache” subdirectory.

`urlquick.request` (*method*, *url*, *params=None*, *data=None*, *headers=None*, *cookies=None*, *auth=None*,
timeout=10, *allow_redirects=None*, *verify=True*, *json=None*, *raise_for_status=None*,
max_age=None)

Make request for remote resource.

Parameters

- **method** (*str*) – HTTP request method, GET, HEAD, POST.
- **url** (*str*) – Url of the remote resource.
- **params** (*dict*) – [opt] Dictionary of url query key/value pairs.
- **data** – [opt] Dictionary (will be form-encoded) or bytes sent in the body of the Request.
- **headers** (*dict*) – [opt] HTTP request headers.
- **cookies** (*dict*) – [opt] Dictionary of cookies to send with the request.
- **auth** (*tuple*) – [opt] (username, password) for basic authentication.
- **timeout** (*int*) – [opt] Connection timeout in seconds.
- **allow_redirects** (*bool*) – [opt] Enable/disable redirection. Defaults to `True`.
- **verify** (*bool*) – [opt] Controls whether to verify the server’s TLS certificate. Defaults to `True`
- **json** – [opt] Json data sent in the body of the Request.
- **raise_for_status** (*bool*) – [opt] Raise’s `HTTPError` if status code is > 400. Defaults to `False`.

- **max_age** (*int*) – [opt] Age the ‘cache’ can be, before it’s considered stale. -1 will disable caching. Defaults to *MAX_AGE*

Returns A requests like Response object.

Return type *urlquick.Response*

Raises

- **MaxRedirects** – If too many redirects was detected.
- **ConnError** – If connection to server failed.
- **HTTPError** – If response status is greater or equal to 400 and raise_for_status is True.
- **SSLError** – If an SSL error occurs while sending the request.
- **Timeout** – If the connection to server timed out.

`urlquick.get(url, params=None, **kwargs)`

Sends a GET request.

Requests data from a specified resource.

Parameters

- **url** (*str*) – Url of the remote resource.
- **params** (*dict*) – [opt] Dictionary of url query key/value pairs.
- **kwargs** – Optional arguments that *request* takes.

Returns A requests like Response object.

Return type *urlquick.Response*

`urlquick.head(url, **kwargs)`

Sends a HEAD request.

Same as GET but returns only HTTP headers and no document body.

Parameters

- **url** (*str*) – Url of the remote resource.
- **kwargs** – Optional arguments that *request* takes.

Returns A requests like Response object.

Return type *urlquick.Response*

`urlquick.post(url, data=None, json=None, **kwargs)`

Sends a POST request.

Submits data to be processed to a specified resource.

Parameters

- **url** (*str*) – Url of the remote resource.
- **data** – [opt] Dictionary (will be form-encoded) or bytes sent in the body of the Request.
- **json** – [opt] Json data sent in the body of the Request.
- **kwargs** – Optional arguments that *request* takes.

Returns A requests like Response object.

Return type *urlquick.Response*

`urlquick.put(url, data=None, **kwargs)`

Sends a PUT request.

Uploads a representation of the specified URI.

Parameters

- **url** (*str*) – Url of the remote resource.
- **data** – [opt] Dictionary (will be form-encoded) or bytes sent in the body of the Request.
- **kwargs** – Optional arguments that *request* takes.

Returns A requests like Response object.

Return type *urlquick.Response*

`urlquick.patch(url, data=None, **kwargs)`

Sends a PATCH request.

Parameters

- **url** (*str*) – Url of the remote resource.
- **data** – [opt] Dictionary (will be form-encoded) or bytes sent in the body of the Request.
- **kwargs** – Optional arguments that *request* takes.

Returns A requests like Response object.

Return type *urlquick.Response*

`urlquick.delete(url, **kwargs)`

Sends a DELETE request.

Parameters

- **url** (*str*) – Url of the remote resource.
- **kwargs** – Optional arguments that *request* takes.

Returns A requests like Response object.

Return type *urlquick.Response*

`urlquick.cache_cleanup(max_age=None)`

Remove all stale cache files.

Parameters **max_age** (*int*) – [opt] The max age the cache can be before removal. defaults => *MAX_AGE*

exception `urlquick.UrlError`

Base exception. All exceptions and errors will subclass from this.

exception `urlquick.Timeout`

Request timed out.

exception `urlquick.MaxRedirects`

Too many redirects.

exception `urlquick.ContentError`

Failed to decode content.

exception `urlquick.ConnError`

A Connection error occurred.

exception `urlquick.SSLError`

An SSL error occurred.

exception `urlquick.HTTPError (url, code, msg, hdrs)`
Raised when HTTP error occurs.

5.1 Session Class

class `urlquick.Session (**kwargs)`

Provides cookie persistence and connection-pooling plus configuration.

Parameters `kwargs` – Default configuration for session variables.

Variables

- **max_repeats** (*int*) – Max number of repeat redirects. Defaults to 4
- **max_redirects** (*int*) – Max number of redirects. Defaults to 10
- **allow_redirects** (*bool*) – Enable/disable redirection. Defaults to True
- **raise_for_status** (*bool*) – Raise HTTPError if status code is > 400. Defaults to False
- **max_age** (*int*) – Max age the cache can be, before it's considered stale. -1 will disable caching. Defaults to `MAX_AGE`

auth

Default Authentication tuple to attach to Request.

Returns Default authentication tuple.

Return type `tuple`

cookies

Dictionary of cookies to attach to each request.

Returns Session cookies

Return type `dict`

delete (*url*, ***kwargs*)

Sends a DELETE request.

Parameters

- **url** (*str*) – Url of the remote resource.
- **kwargs** – Optional arguments that `request` takes.

Returns A requests like Response object.

Return type `urlquick.Response`

get (*url*, *params=None*, ***kwargs*)

Sends a GET request.

Requests data from a specified resource.

Parameters

- **url** (*str*) – Url of the remote resource.
- **params** (*dict*) – [opt] Dictionary of url query key/value pairs.
- **kwargs** – Optional arguments that `request` takes.

Returns A requests like Response object.

Return type *urlquick.Response*

head (*url*, ***kwargs*)

Sends a HEAD request.

Same as GET but returns only HTTP headers and no document body.

Parameters

- **url** (*str*) – Url of the remote resource.
- **kwargs** – Optional arguments that *request* takes.

Returns A requests like Response object.

Return type *urlquick.Response*

headers

Dictionary of headers to attach to each request.

Returns Session headers

Return type *dict*

params

Dictionary of querystrings to attach to each Request. The dictionary values may be lists for representing multivalued query parameters.

Returns Session params

Return type *dict*

patch (*url*, *data=None*, ***kwargs*)

Sends a PATCH request.

Parameters

- **url** (*str*) – Url of the remote resource.
- **data** – [opt] Dictionary (will be form-encoded) or bytes sent in the body of the Request.
- **kwargs** – Optional arguments that *request* takes.

Returns A requests like Response object.

Return type *urlquick.Response*

post (*url*, *data=None*, *json=None*, ***kwargs*)

Sends a POST request.

Submits data to be processed to a specified resource.

Parameters

- **url** (*str*) – Url of the remote resource.
- **data** – [opt] Dictionary (will be form-encoded) or bytes sent in the body of the Request.
- **json** – [opt] Json data sent in the body of the Request.
- **kwargs** – Optional arguments that *request* takes.

Returns A requests like Response object.

Return type *urlquick.Response*

put (*url*, *data=None*, ***kwargs*)

Sends a PUT request.

Uploads a representation of the specified URI.

Parameters

- **url** (*str*) – Url of the remote resource.
- **data** – [opt] Dictionary (will be form-encoded) or bytes sent in the body of the Request.
- **kwargs** – Optional arguments that *request* takes.

Returns A requests like Response object.

Return type *urlquick.Response*

request (*method*, *url*, *params=None*, *data=None*, *headers=None*, *cookies=None*, *auth=None*, *timeout=10*, *allow_redirects=None*, *verify=True*, *json=None*, *raise_for_status=None*, *max_age=None*)

Make request for remote resource.

Parameters

- **method** (*str*) – HTTP request method, GET, HEAD, POST.
- **url** (*str*) – Url of the remote resource.
- **params** (*dict*) – [opt] Dictionary of url query key/value pairs.
- **data** – [opt] Dictionary (will be form-encoded) or bytes sent in the body of the Request.
- **headers** (*dict*) – [opt] HTTP request headers.
- **cookies** (*dict*) – [opt] Dictionary of cookies to send with the request.
- **auth** (*tuple*) – [opt] (username, password) for basic authentication.
- **timeout** (*int*) – [opt] Connection timeout in seconds.
- **allow_redirects** (*bool*) – [opt] Enable/disable redirection. Defaults to `True`.
- **verify** (*bool*) – [opt] Controls whether to verify the server's TLS certificate. Defaults to `True`.
- **json** – [opt] Json data sent in the body of the Request.
- **raise_for_status** (*bool*) – [opt] Raise's `HTTPError` if status code is > 400. Defaults to `False`.
- **max_age** (*int*) – [opt] Age the 'cache' can be, before it's considered stale. -1 will disable caching. Defaults to `MAX_AGE`.

Returns A requests like Response object.

Return type *urlquick.Response*

Raises

- **MaxRedirects** – If too many redirects was detected.
- **ConnError** – If connection to server failed.
- **HTTPError** – If response status is greater or equal to 400 and `raise_for_status` is `True`.
- **SSLError** – If an SSL error occurs while sending the request.
- **Timeout** – If the connection to server timed out.

5.2 Response Object

class urlquick.**Response** (*response, org_request, start_time, history*)

A Response object containing all data returned from the server.

content

Content of the response in bytes.

Raises *ContentError* – If content failes to decompress.

cookies

A dictionary of Cookies the server sent back.

encoding

Encoding, to decode with, when accessing *resp.text*.

links

Dictionary of ‘parsed header links’ of the response, if any.

text

Content of the response in unicode.

The response content will be decoded using the best available encoding based on the response headers.
Will fallback to *apparent_encoding* if no encoding was given within headers.

apparent_encoding = None

The default encoding, used when no encoding is given.

content

Content of the response in bytes.

Raises *ContentError* – If content failes to decompress.

cookies

A dictionary of Cookies the server sent back.

elapsed = None

The amount of time elapsed, between sending the request and the arrival of the response (as a timedelta).

encoding

Encoding, to decode with, when accessing *resp.text*.

headers

Case-insensitive Dictionary of Response Headers.

history = None

A list of Response objects from the history of the Request. Any redirect responses will end up here.

is_permanent_redirect

True, if this Response is one of the permanent versions of redirect.

is_redirect

True, if this Response is a well-formed HTTP redirect, that could have been processed automatically.

iter_content (*chunk_size=512, decode_unicode=False*)

Iterates over the response data. The chunk size are the number of bytes it should read into memory. This is not necessarily the length of each item returned, as decoding can take place.

Parameters

- **chunk_size** (*int*) – [opt] The chunk size to use for each chunk. (default=512)
- **decode_unicode** (*bool*) – [opt] True to return unicode, else False to return bytes. (default=‘False’)

iter_lines (*chunk_size=None, decode_unicode=False, delimiter=b'\n'*)

Iterates over the response data, one line at a time.

Parameters

- **chunk_size** (*int*) – [opt] Unused, here for compatibility with requests.
- **decode_unicode** (*bool*) – [opt] True to return unicode, else False to return bytes. (default='False')
- **delimiter** (*bytes*) – [opt] Delimiter used as the end of line marker. (default=b'\n')

json (***kwargs*)

Returns the json-encoded content of a response.

Parameters **kwargs** – [opt] Arguments that `json.loads()` takes.

Raises **ValueError** – If the response body does not contain valid json.

links

Dictionary of 'parsed header links' of the response, if any.

ok

True, if status_code is less than 400.

This attribute checks if the status code of the response is between 400 and 600. To see if there was a client error or a server error. If the status code is between 200 and 400, this will return True. This is not a check to see if the response code is 200 "OK".

parse (*tag=", attrs=None*)

Parse's "HTML" document into a element tree using HTMLement.

See also:

The htmlement documentation can be found at.

<http://python-htmlement.readthedocs.io/en/stable/?badge=stable>

Parameters

- **tag** (*str*) – [opt] Name of 'element' which is used to filter tree to required section.
- **attrs** (*dict*) – [opt] Attributes of 'element', used when searching for required section. Attrs should be a dict of unicode key/value pairs.

Returns The root element of the element tree.

Return type `xml.etree.ElementTree.Element`

Raises **MissingDependency** – If the optional 'HTMLement' dependency is missing.

raise_for_status ()

Raises stored error, if one occurred.

Raises **HTTPError** – If response status code is greater or equal to 400

raw = None

File-like object representation of response (for advanced usage).

reason = None

Textual reason of response HTTP Status e.g. "Not Found" or "OK".

request = None

The *Request* object to which this is a response.

status_code = None

Integer Code of responded HTTP Status e.g. 404 or 200.

text

Content of the response in unicode.

The response content will be decoded using the best available encoding based on the response headers.
Will fallback to *apparent_encoding* if no encoding was given within headers.

url = None

Final URL location of Response.

xml ()

Parse's "XML" document into a element tree.

Returns The root element of the element tree.

Return type `xml.etree.ElementTree.Element`

5.3 Request Object

class `urlquick.Request` (*method*, *url*, *headers*, *data=None*, *json=None*, *params=None*, *referrer=None*)

A Request Object

auth = None

Tuple of (username, password) for basic authentication.

data = None

Request body, to send to the server.

header_items ()

Return list of tuples (header_name, header_value) of the Request headers, as native type of `str`.

host = None

The URI authority, typically a host, but may also contain a port separated by a colon.

method = None

The HTTP request method to use.

selector

Resource selector, with the url path and query parts.

type = None

The URI scheme.

url = None

Urlencoded url of the remote resource.

CHAPTER 6

External Links

Requests Quickstart: <http://docs.python-requests.org/en/master/user/quickstart/>

Bug Tracker: <https://github.com/willforde/urlquick/issues>

A

apparent_encoding (*urlquick.Response* attribute), 17

auth (*urlquick.Request* attribute), 19

auth (*urlquick.Session* attribute), 14

C

cache_cleanup() (*in module urlquick*), 13

ConnError, 13

content (*urlquick.Response* attribute), 17

ContentError, 13

cookies (*urlquick.Response* attribute), 17

cookies (*urlquick.Session* attribute), 14

D

data (*urlquick.Request* attribute), 19

delete() (*in module urlquick*), 13

delete() (*urlquick.Session* method), 14

E

elapsed (*urlquick.Response* attribute), 17

encoding (*urlquick.Response* attribute), 17

G

get() (*in module urlquick*), 12

get() (*urlquick.Session* method), 14

H

head() (*in module urlquick*), 12

head() (*urlquick.Session* method), 15

header_items() (*urlquick.Request* method), 19

headers (*urlquick.Response* attribute), 17

headers (*urlquick.Session* attribute), 15

history (*urlquick.Response* attribute), 17

host (*urlquick.Request* attribute), 19

HTTPError, 13

I

is_permanent_redirect (*urlquick.Response* attribute), 17

is_redirect (*urlquick.Response* attribute), 17

iter_content() (*urlquick.Response* method), 17

iter_lines() (*urlquick.Response* method), 17

J

json() (*urlquick.Response* method), 18

L

links (*urlquick.Response* attribute), 17, 18

M

MAX_AGE (*in module urlquick*), 11

MaxRedirects, 13

method (*urlquick.Request* attribute), 19

O

ok (*urlquick.Response* attribute), 18

P

params (*urlquick.Session* attribute), 15

parse() (*urlquick.Response* method), 18

patch() (*in module urlquick*), 13

patch() (*urlquick.Session* method), 15

post() (*in module urlquick*), 12

post() (*urlquick.Session* method), 15

put() (*in module urlquick*), 12

put() (*urlquick.Session* method), 15

R

raise_for_status() (*urlquick.Response* method), 18

raw (*urlquick.Response* attribute), 18

reason (*urlquick.Response* attribute), 18

Request (*class in urlquick*), 19

request (*urlquick.Response* attribute), 18

request() (*in module urlquick*), 11

request() (*urlquick.Session* method), 16

Response (*class in urlquick*), 17

S

`selector` (*urlquick.Request* attribute), 19
`Session` (class in *urlquick*), 14
`SSLError`, 13
`status_code` (*urlquick.Response* attribute), 18

T

`text` (*urlquick.Response* attribute), 17, 19
`Timeout`, 13
`type` (*urlquick.Request* attribute), 19

U

`url` (*urlquick.Request* attribute), 19
`url` (*urlquick.Response* attribute), 19
`UrlError`, 13
`urlquick.CACHE_LOCATION` (built-in variable), 11

X

`xml()` (*urlquick.Response* method), 19