
requests-html Documentation

Release v0.3.4

Kenneth Reitz

May 10, 2020

Contents:

1	Installation	3
2	Tutorial & Usage	5
3	JavaScript Support	9
4	Pagination	11
5	Using without Requests	13
6	API Documentation	15
6.1	Main Classes	15
6.2	Utility Functions	19
6.3	HTML Sessions	19
7	Indices and tables	27
	Python Module Index	29
	Index	31

This library intends to make parsing HTML (e.g. scraping the web) as simple and intuitive as possible.

When using this library you automatically get:

- **Full JavaScript support!**
- *CSS Selectors* (a.k.a jQuery-style, thanks to PyQuery).
- *XPath Selectors*, for the faint of heart.
- Mocked user-agent (like a real web browser).
- Automatic following of redirects.
- Connection-pooling and cookie persistence.
- The Requests experience you know and love, with magical parsing abilities.
- **Async Support**

CHAPTER 1

Installation

```
$ pipenv install requests-html
```

Only **Python 3.6** is supported.

CHAPTER 2

Tutorial & Usage

Make a GET request to `python.org`, using `Requests`:

```
>>> from requests_html import HTMLSession
>>> session = HTMLSession()

>>> r = session.get('https://python.org/')
(continues on next page)
```

Or want to try our async session:

```
>>> from requests_html import AsyncHTMLSession
>>> asession = AsyncHTMLSession()

>>> r = await asession.get('https://python.org/')
```

But async is fun when fetching some sites at the same time:

```
>>> from requests_html import AsyncHTMLSession
>>> asession = AsyncHTMLSession()

>>> async def get_pythonorg():
...     r = await asession.get('https://python.org/')

>>> async def get_reddit():
...     r = await asession.get('https://reddit.com/')

>>> async def get_google():
...     r = await asession.get('https://google.com/')

>>> session.run(get_pythonorg, get_reddit, get_google)
```

Grab a list of all links on the page, as-is (anchors excluded):

```
>>> r.html.links
{'//docs.python.org/3/tutorial/', '/about/apps/', 'https://github.com/python/
pythonondotorg/issues', '/accounts/login/', '/dev/peps/', '/about/legal/', '//docs.
python.org/3/tutorial/introduction.html#lists', '/download/alternatives', 'http://
feedproxy.google.com/~r/PythonInsider/~3/kihd2DW98YY/python-370a4-is-available-for-
testing.html', '/download/other/', '/downloads/windows/', 'https://mail.python.org/
mailman/listinfo/python-dev', '/doc/av', 'https://devguide.python.org/', '/about/
success/#engineering', 'https://wiki.python.org/moin/PythonEventsCalendar
#Submitting_an_Event', 'https://www.openstack.org', '/about/gettingstarted/',
'http://feedproxy.google.com/~r/PythonInsider/~3/AMoBel8b8Mc/python-3.html', '/
5
```

Grab a list of all links on the page, in *absolute form* (anchors excluded):

```
>>> r.html.absolute_links
{'https://github.com/python/pythondotorg/issues', 'https://docs.python.org/3/tutorial/
↳', 'https://www.python.org/about/success/', 'http://feedproxy.google.com/~r/
↳PythonInsider/~3/kih2DW98YY/python-370a4-is-available-for-testing.html', 'https://
↳www.python.org/dev/peps/', 'https://mail.python.org/mailman/listinfo/python-dev',
↳'https://www.python.org/doc/', 'https://www.python.org/', 'https://www.python.org/
↳about/', 'https://www.python.org/events/python-events/past/', 'https://devguide.
↳python.org/', 'https://wiki.python.org/moin/PythonEventsCalendar#Submitting_an_Event
↳', 'https://www.openstack.org', 'http://feedproxy.google.com/~r/PythonInsider/~3/
↳AMoBel8b8Mc/python-3.html', 'https://docs.python.org/3/tutorial/introduction.html
↳#lists', 'http://docs.python.org/3/tutorial/introduction.html#using-python-as-a-
↳calculator', 'http://pyfound.blogspot.com/', 'https://wiki.python.org/moin/
↳PythonBooks', 'http://plus.google.com/+Python', 'https://wiki.python.org/moin/',
↳'https://www.python.org/events/python-events', 'https://status.python.org/',
↳'https://www.python.org/about/apps', 'https://www.python.org/downloads/release/
↳python-2714/', 'https://www.python.org/psf/donations/', 'http://buildbot.net/',
↳'http://twitter.com/ThePSF', 'https://docs.python.org/3/license.html', 'http://wiki.
↳python.org/moin/Languages', 'https://docs.python.org/faq/', 'https://jobs.python.org
↳', 'https://www.python.org/about/success/#software-development', 'https://www.
↳python.org/about/success/#education', 'https://www.python.org/community/logos/',
↳'https://www.python.org/doc/av', 'https://wiki.qt.io/PySide', 'https://www.python.
↳org/events/python-user-group/660/', 'https://wiki.gnome.org/Projects/PyGObject',
↳'http://www.ansible.com', 'http://www.saltstack.com', 'https://www.python.org/dev/
↳peps/peps.rss', 'http://planetpython.org/', 'https://www.python.org/events/python-
↳user-group/past/', 'https://docs.python.org/3/tutorial/controlflow.html#defining-
↳functions', 'https://www.python.org/community/diversity/', 'https://docs.python.org/
↳3/tutorial/controlflow.html', 'https://www.python.org/community/awards', 'https://
↳www.python.org/events/python-user-group/638/', 'https://www.python.org/about/legal/
↳', 'https://www.python.org/dev/', 'https://www.python.org/download/alternatives',
↳'https://www.python.org/downloads/', 'https://www.python.org/community/lists/',
↳'http://www.wxpython.org/', 'https://www.python.org/about/success/#government',
↳'https://www.python.org/psf/', 'https://www.python.org/psf/codeofconduct/', 'http://
↳bottlepy.org', 'http://roundup.sourceforge.net/', 'http://pandas.pydata.org/',
↳'http://brochure.getpython.info/', 'https://www.python.org/downloads/source/',
↳'https://bugs.python.org/', 'https://www.python.org/downloads/mac-osx/', 'https://
↳www.python.org/about/help/', 'http://tornadoweb.org', 'http://flask.pocoo.org/',
↳'https://www.python.org/users/membership/', 'http://blog.python.org', 'https://www.
↳python.org/privacy/', 'https://www.python.org/about/gettingstarted/', 'http://
↳python.org/dev/peps/', 'https://www.python.org/about/apps/', 'https://docs.python.
↳org', 'https://www.python.org/success-stories/', 'https://www.python.org/community/
↳forums/', 'http://feedproxy.google.com/~r/PythonInsider/~3/zVC80sq9s00/python-364-
↳is-now-available.html', 'https://www.python.org/community/merchandise/', 'https://
↳www.python.org/about/success/#arts', 'https://wiki.python.org/moin/Python2orPython3
↳', 'http://trac.edgewall.org/', 'http://feedproxy.google.com/~r/PythonInsider/~3/
↳wh73_1A-N7Q/python-355rc1-and-python-348rc1-are-now.html', 'https://pypi.python.org/
↳', 'https://www.python.org/events/python-user-group/650/', 'http://www.
↳riverbankcomputing.co.uk/software/pyqt/intro', 'https://www.python.org/about/quotes/
↳', 'https://www.python.org/downloads/windows/', 'https://www.python.org/events/
↳calendars/', 'http://www.scipy.org', 'https://www.python.org/community/workshops/',
↳'https://www.python.org/blogs/', 'https://www.python.org/accounts/signup/', 'https://
↳www.python.org/events/', 'https://kivy.org/', 'http://www.facebook.com/pythonlang?
↳fref=ts', 'http://www.web2py.com/', 'https://www.python.org/psf/sponsorship/
↳sponsors/', 'https://www.python.org/community/', 'https://www.python.org/download/
↳other/', 'https://www.python.org/psf/landing/', 'https://www.python.org/events/
↳python-user-group/665/', 'https://wiki.python.org/moin/BeginnersGuide', 'https://
↳www.python.org/accounts/login/', 'https://www.python.org/downloads/release/python-
↳364/', 'https://www.python.org/dev/core-mentorship/', 'https://www.python.org/
↳success/#business', 'https://www.python.org/community/sigs/', 'https://www.python.
↳org/events/python-user-group/', 'http://ipython.org', 'https://www.python.org/shell/
↳', 'https://www.python.org/community/irc/', 'https://www.python.org/about/success/
↳#engineering', 'http://www.pylonsproject.org/', 'http://pycon.blogspot.com/',
```

(continues on next page)

(continued from previous page)

Select an Element with a CSS Selector ([learn more](#)):

```
>>> about = r.html.find('#about', first=True)
```

Grab an Element's text contents:

```
>>> print (about.text)
About
Applications
Quotes
Getting Started
Help
Python Brochure
```

Introspect an Element's attributes ([learn more](#)):

```
>>> about.attrs
{'id': 'about', 'class': ('tier-1', 'element-1'), 'aria-haspopup': 'true'}
```

Render out an Element's HTML:

```
>>> about.html
'<li aria-haspopup="true" class="tier-1 element-1 " id="about">\n<a class="" href="/
↳ about/" title="">About</a>\n<ul aria-hidden="true" class="subnav menu" role="menu">
↳ \n<li class="tier-2 element-1" role="treeitem"><a href="/about/apps/" title="">
↳ Applications</a></li>\n<li class="tier-2 element-2" role="treeitem"><a href="/about/
↳ quotes/" title="">Quotes</a></li>\n<li class="tier-2 element-3" role="treeitem"><a
↳ href="/about/gettingstarted/" title="">Getting Started</a></li>\n<li class="tier-2
↳ element-4" role="treeitem"><a href="/about/help/" title="">Help</a></li>\n<li class=
↳ "tier-2 element-5" role="treeitem"><a href="http://brochure.getpython.info/" title=""
↳ ">Python Brochure</a></li>\n</ul>\n</li>'
```

Grab an Element's root tag name:

```
>>> about.tag
'li'
```

Show the line number that an Element's root tag located in:

```
>>> about.lineno
249
```

Select an Element list within an Element:

```
>>> about.find('a')
[<Element 'a' href='/about/' title='' class=''>, <Element 'a' href='/about/apps/'
↳ title=''>, <Element 'a' href='/about/quotes/' title=''>, <Element 'a' href='/about/
↳ gettingstarted/' title=''>, <Element 'a' href='/about/help/' title=''>, <Element 'a
↳ href='http://brochure.getpython.info/' title=''>]
```

Search for links within an element:

```
>>> about.absolute_links
{'http://brochure.getpython.info/', 'https://www.python.org/about/gettingstarted/',
↳ 'https://www.python.org/about/', 'https://www.python.org/about/quotes/', 'https://
↳ www.python.org/about/help/', 'https://www.python.org/about/apps/'}
```

(continues on next page)

(continued from previous page)

Search for text on the page:

```
>>> r.html.search('Python is a {} language')[0]
programming
```

More complex CSS Selector example (copied from Chrome dev tools):

```
>>> r = session.get('https://github.com/')
>>> sel = 'body > div.application-main > div.jumbotron.jumbotron-codelines > div >
↳div > div.col-md-7.text-center.text-md-left > p'

>>> print(r.html.find(sel, first=True).text)
GitHub is a development platform inspired by the way you work. From open source to
↳business, you can host and review code, manage projects, and build software.
↳alongside millions of other developers.
```

XPath is also supported ([learn more](#)):

```
>>> r.html.xpath('a')
[<Element 'a' class='btn' href='https://help.github.com/articles/supported-browsers'>]
```

You can also select only elements containing certain text:

```
>>> r = session.get('http://python-requests.org/')
>>> r.html.find('a', containing='kenneth')
[<Element 'a' href='http://kennethreitz.com/pages/open-projects.html'>, <Element 'a'
↳href='http://kennethreitz.org/'>, <Element 'a' href='https://twitter.com/
↳kennethreitz' class=('twitter-follow-button',) data-show-count='false'>, <Element 'a
↳' class=('reference', 'internal') href='dev/contributing/#kenneth-reitz-s-code-style
↳'>]
```

CHAPTER 3

JavaScript Support

Let's grab some text that's rendered by JavaScript:

```
>>> r = session.get('http://python-requests.org/')
>>> r.html.render()
>>> r.html.search('Python 2 will retire in only {months} months!')['months']
'<time>25</time>'
```

Or you can do this async also:

```
>>> r = aession.get('http://python-requests.org/')
>>> await r.html.arender()
>>> r.html.search('Python 2 will retire in only {months} months!')['months']
'<time>25</time>'
```

Note, the first time you ever run the `render()` method, it will download Chromium into your home directory (e.g. `~/.pypeteer/`). This only happens once. You may also need to install a few [Linux packages](#) to get pypeteer working.

There's also intelligent pagination support (always improving):

```
>>> r = session.get('https://reddit.com')
>>> for html in r.html:
...     print(html)
<HTML url='https://www.reddit.com/'>
<HTML url='https://www.reddit.com/?count=25&after=t3_81puu5'>
<HTML url='https://www.reddit.com/?count=50&after=t3_81nevg'>
<HTML url='https://www.reddit.com/?count=75&after=t3_81lqtp'>
<HTML url='https://www.reddit.com/?count=100&after=t3_81k1c8'>
<HTML url='https://www.reddit.com/?count=125&after=t3_81p438'>
<HTML url='https://www.reddit.com/?count=150&after=t3_81nrnd'>
...
```

For *async* pagination use the new *async for*:

```
>>> r = await asession.get('https://reddit.com')
>>> async for html in r.html:
...     print(html)
<HTML url='https://www.reddit.com/'>
<HTML url='https://www.reddit.com/?count=25&after=t3_81puu5'>
...
```

You can also just request the next URL easily:

```
>>> r = session.get('https://reddit.com')
>>> r.html.next()
'https://www.reddit.com/?count=25&after=t3_81pm82'
```


CHAPTER 5

Using without Requests

You can also use this library without Requests:

```
>>> from requests_html import HTML
>>> doc = """<a href='https://httpbin.org'>"""

>>> html = HTML(html=doc)
>>> html.links
{'https://httpbin.org'}
```

You can also render JavaScript pages without Requests:

```
# ^^ proceeding from above ^^
>>> script = """
    () => {
        return {
            width: document.documentElement.clientWidth,
            height: document.documentElement.clientHeight,
            deviceScaleFactor: window.devicePixelRatio,
        }
    }
    """
>>> val = html.render(script=script, reload=False)

>>> print(val)
{'width': 800, 'height': 600, 'deviceScaleFactor': 1}

>>> print(html.html)
<html><head></head><body><a href="https://httpbin.org"></a></body></html>
```

For using *arender* just pass *async_=True* to HTML.

```
# ^^ using above script ^^
>>> html = HTML(html=doc, async_=True)
>>> val = await html.arender(script=script, reload=False)
```

(continues on next page)

(continued from previous page)

```
>>> print(val)
{'width': 800, 'height': 600, 'deviceScaleFactor': 1}
```

6.1 Main Classes

These classes are the main interface to `requests-html`:

```
class requests_html.HTML(*, session: Union[HTMLSession, AsyncHTMLSession] = None, url: str = 'https://example.org/', html: Union[str, bytes], default_encoding: str = 'utf-8', async_: bool = False)
```

An HTML document, ready for parsing.

Parameters

- **url** – The URL from which the HTML originated, used for `absolute_links`.
- **html** – HTML from which to base the parsing upon (optional).
- **default_encoding** – Which encoding to default to.

`absolute_links`

All found links on page, in absolute form ([learn more](#)).

```
arender(retries: int = 8, script: str = None, wait: float = 0.2, scrolldown=False, sleep: int = 0, reload: bool = True, timeout: Union[float, int] = 8.0, keep_page: bool = False, cookies: list = [], send_cookies_session: bool = False)
```

Async version of render. Takes same parameters.

`base_url`

The base URL for the page. Supports the `<base>` tag ([learn more](#)).

`encoding`

The encoding string to be used, extracted from the HTML and `HTMLResponse` headers.

```
find(selector: str = '*', *, containing: Union[str, List[str]] = None, clean: bool = False, first: bool = False, _encoding: str = None) → Union[List[requests_html.Element], requests_html.Element]
```

Given a CSS Selector, returns a list of *Element* objects or a single one.

Parameters

- **selector** – CSS Selector to use.

- **clean** – Whether or not to sanitize the found HTML of `<script>` and `<style>` tags.
- **containing** – If specified, only return elements that contain the provided text.
- **first** – Whether or not to return just the first result.
- **_encoding** – The encoding format.

Example CSS Selectors:

- `a`
- `a.someClass`
- `a#someID`
- `a[target=_blank]`

See W3School's [CSS Selectors Reference](#) for more details.

If `first` is `True`, only returns the first *Element* found.

full_text

The full text content (including links) of the *Element* or *HTML*.

html

Unicode representation of the HTML content ([learn more](#)).

links

All found links on page, in as-is form.

lxml

lxml representation of the *Element* or *HTML*.

next (*fetch*: *bool* = *False*, *next_symbol*: *List[str]* = [*'next'*, *'more'*, *'older'*]) → *Union[requests_html.HTML, List[str]]*

Attempts to find the next page, if there is one. If `fetch` is `True` (default), returns *HTML* object of next page. If `fetch` is `False`, simply returns the next URL.

pq

PyQuery representation of the *Element* or *HTML*.

raw_html

Bytes representation of the HTML content. ([learn more](#)).

render (*retries*: *int* = 8, *script*: *str* = *None*, *wait*: *float* = 0.2, *scrolldown*=*False*, *sleep*: *int* = 0, *reload*: *bool* = *True*, *timeout*: *Union[float, int]* = 8.0, *keep_page*: *bool* = *False*, *cookies*: *list* = *[{}]*, *send_cookies_session*: *bool* = *False*)

Reloads the response in Chromium, and replaces HTML content with an updated version, with JavaScript executed.

Parameters

- **retries** – The number of times to retry loading the page in Chromium.
- **script** – JavaScript to execute upon page load (optional).
- **wait** – The number of seconds to wait before loading the page, preventing timeouts (optional).
- **scrolldown** – Integer, if provided, of how many times to page down.
- **sleep** – Integer, if provided, of how many seconds to sleep after initial render.
- **reload** – If `False`, content will not be loaded from the browser, but will be provided from memory.

- **keep_page** – If True will allow you to interact with the browser page through `r.html.page`.
- **send_cookies_session** – If True send `HTMLSession.cookies` convert.
- **cookies** – If not empty send cookies.

If `scrolldown` is specified, the page will scroll down the specified number of times, after sleeping the specified amount of time (e.g. `scrolldown=10, sleep=1`).

If just `sleep` is provided, the rendering will wait *n* seconds, before returning.

If `script` is specified, it will execute the provided JavaScript at runtime. Example:

```
script = """
    () => {
        return {
            width: document.documentElement.clientWidth,
            height: document.documentElement.clientHeight,
            deviceScaleFactor: window.devicePixelRatio,
        }
    }
"""
```

Returns the return value of the executed script, if any is provided:

```
>>> r.html.render(script=script)
{'width': 800, 'height': 600, 'deviceScaleFactor': 1}
```

Warning: the first time you run this method, it will download Chromium into your home directory (~/.pypeteer).

search (*template: str*) → `parse.Result`

Search the `Element` for the given Parse template.

Parameters **template** – The Parse template to use.

search_all (*template: str*) → `Union[List[parse.Result], parse.Result]`

Search the `Element` (multiple times) for the given parse template.

Parameters **template** – The Parse template to use.

text

The text content of the `Element` or `HTML`.

xpath (*selector: str, *, clean: bool = False, first: bool = False, _encoding: str = None*) →

`Union[List[str], List[requests_html.Element], str, requests_html.Element]`

Given an XPath selector, returns a list of `Element` objects or a single one.

Parameters

- **selector** – XPath Selector to use.
- **clean** – Whether or not to sanitize the found HTML of `<script>` and `<style>` tags.
- **first** – Whether or not to return just the first result.
- **_encoding** – The encoding format.

If a sub-selector is specified (e.g. `//a/@href`), a simple list of results is returned.

See W3School's [XPath Examples](#) for more details.

If `first` is True, only returns the first `Element` found.

class requests_html.**Element** (*, *element*, *url*: str, *default_encoding*: str = None)

An element of HTML.

Parameters

- **element** – The element from which to base the parsing upon.
- **url** – The URL from which the HTML originated, used for `absolute_links`.
- **default_encoding** – Which encoding to default to.

absolute_links

All found links on page, in absolute form ([learn more](#)).

attrs

Returns a dictionary of the attributes of the *Element* ([learn more](#)).

base_url

The base URL for the page. Supports the `<base>` tag ([learn more](#)).

encoding

The encoding string to be used, extracted from the HTML and `HTMLResponse` headers.

find (*selector*: str = '*', *, *containing*: Union[str, List[str]] = None, *clean*: bool = False, *first*: bool = False, *_encoding*: str = None) → Union[List[requests_html.Element], requests_html.Element]
Given a CSS Selector, returns a list of *Element* objects or a single one.

Parameters

- **selector** – CSS Selector to use.
- **clean** – Whether or not to sanitize the found HTML of `<script>` and `<style>` tags.
- **containing** – If specified, only return elements that contain the provided text.
- **first** – Whether or not to return just the first result.
- **_encoding** – The encoding format.

Example CSS Selectors:

- `a`
- `a.someClass`
- `a#someID`
- `a[target=_blank]`

See W3School's [CSS Selectors Reference](#) for more details.

If `first` is `True`, only returns the first *Element* found.

full_text

The full text content (including links) of the *Element* or *HTML*.

html

Unicode representation of the HTML content ([learn more](#)).

links

All found links on page, in as-is form.

lxml

lxml representation of the *Element* or *HTML*.

pq

PyQuery representation of the *Element* or *HTML*.

raw_html

Bytes representation of the HTML content. ([learn more](#)).

search (*template: str*) → parse.Result

Search the *Element* for the given Parse template.

Parameters **template** – The Parse template to use.

search_all (*template: str*) → Union[List[parse.Result], parse.Result]

Search the *Element* (multiple times) for the given parse template.

Parameters **template** – The Parse template to use.

text

The text content of the *Element* or *HTML*.

xpath (*selector: str, *, clean: bool = False, first: bool = False, _encoding: str = None*) →

Union[List[str], List[requests_html.Element], str, requests_html.Element]

Given an XPath selector, returns a list of *Element* objects or a single one.

Parameters

- **selector** – XPath Selector to use.
- **clean** – Whether or not to sanitize the found HTML of <script> and <style> tags.
- **first** – Whether or not to return just the first result.
- **_encoding** – The encoding format.

If a sub-selector is specified (e.g. //a/@href), a simple list of results is returned.

See W3School's [XPath Examples](#) for more details.

If *first* is True, only returns the first *Element* found.

6.2 Utility Functions

`requests_html.user_agent` (*style=None*) → str

Returns an apparently legit user-agent, if not requested one of a specific style. Defaults to a Chrome-style User-Agent.

6.3 HTML Sessions

These sessions are for making HTTP requests:

class `requests_html.HTMLSession` (***kwargs*)

close ()

If a browser was created close it first.

delete (*url, **kwargs*)

Sends a DELETE request. Returns Response object.

Parameters

- **url** – URL for the new Request object.
- ****kwargs** – Optional arguments that *request* takes.

Return type requests.Response

get (*url*, ***kwargs*)

Sends a GET request. Returns Response object.

Parameters

- **url** – URL for the new Request object.
- ****kwargs** – Optional arguments that request takes.

Return type requests.Response

get_adapter (*url*)

Returns the appropriate connection adapter for the given URL.

Return type requests.adapters.BaseAdapter

get_redirect_target (*resp*)

Receives a Response. Returns a redirect URI or None

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

Sends a HEAD request. Returns Response object.

Parameters

- **url** – URL for the new Request object.
- ****kwargs** – Optional arguments that request takes.

Return type requests.Response

merge_environment_settings (*url*, *proxies*, *stream*, *verify*, *cert*)

Check the environment and merge it with some settings.

Return type dict

mount (*prefix*, *adapter*)

Registers a connection adapter to a prefix.

Adapters are sorted in descending order by prefix length.

options (*url*, ***kwargs*)

Sends a OPTIONS request. Returns Response object.

Parameters

- **url** – URL for the new Request object.
- ****kwargs** – Optional arguments that request takes.

Return type requests.Response

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

Sends a PATCH request. Returns Response object.

Parameters

- **url** – URL for the new Request object.
- **data** – (optional) Dictionary, list of tuples, bytes, or file-like object to send in the body of the Request.
- ****kwargs** – Optional arguments that request takes.

Return type requests.Response

post (*url*, *data=None*, *json=None*, ***kwargs*)
Sends a POST request. Returns Response object.

Parameters

- **url** – URL for the new Request object.
- **data** – (optional) Dictionary, list of tuples, bytes, or file-like object to send in the body of the Request.
- **json** – (optional) json to send in the body of the Request.
- ****kwargs** – Optional arguments that request takes.

Return type requests.Response

prepare_request (*request*)
Constructs a PreparedRequest for transmission and returns it. The PreparedRequest has settings merged from the Request instance and those of the Session.

Parameters **request** – Request instance to prepare with this session’s settings.

Return type requests.PreparedRequest

put (*url*, *data=None*, ***kwargs*)
Sends a PUT request. Returns Response object.

Parameters

- **url** – URL for the new Request object.
- **data** – (optional) Dictionary, list of tuples, bytes, or file-like object to send in the body of the Request.
- ****kwargs** – Optional arguments that request takes.

Return type requests.Response

rebuild_auth (*prepared_request*, *response*)
When being redirected we may want to strip authentication from the request to avoid leaking credentials. This method intelligently removes and reapplies authentication where possible to avoid credential loss.

rebuild_method (*prepared_request*, *response*)
When being redirected we may want to change the method of the request based on certain specs or browser behavior.

rebuild_proxies (*prepared_request*, *proxies*)
This method re-evaluates the proxy configuration by considering the environment variables. If we are redirected to a URL covered by NO_PROXY, we strip the proxy configuration. Otherwise, we set missing proxy keys for this URL (in case they were stripped by a previous redirect).

This method also replaces the Proxy-Authorization header where necessary.

Return type dict

request (*method*, *url*, *params=None*, *data=None*, *headers=None*, *cookies=None*, *files=None*, *auth=None*, *timeout=None*, *allow_redirects=True*, *proxies=None*, *hooks=None*, *stream=None*, *verify=None*, *cert=None*, *json=None*)
Constructs a Request, prepares it and sends it. Returns Response object.

Parameters

- **method** – method for the new Request object.
- **url** – URL for the new Request object.
- **params** – (optional) Dictionary or bytes to be sent in the query string for the Request.

- **data** – (optional) Dictionary, list of tuples, bytes, or file-like object to send in the body of the Request.
- **json** – (optional) json to send in the body of the Request.
- **headers** – (optional) Dictionary of HTTP Headers to send with the Request.
- **cookies** – (optional) Dict or CookieJar object to send with the Request.
- **files** – (optional) Dictionary of 'filename': file-like-objects for multi-part encoding upload.
- **auth** – (optional) Auth tuple or callable to enable Basic/Digest/Custom HTTP Auth.
- **timeout** (*float or tuple*) – (optional) How long to wait for the server to send data before giving up, as a float, or a (connect timeout, read timeout) tuple.
- **allow_redirects** (*bool*) – (optional) Set to True by default.
- **proxies** – (optional) Dictionary mapping protocol or protocol and hostname to the URL of the proxy.
- **stream** – (optional) whether to immediately download the response content. Defaults to False.
- **verify** – (optional) Either a boolean, in which case it controls whether we verify the server's TLS certificate, or a string, in which case it must be a path to a CA bundle to use. Defaults to True.
- **cert** – (optional) if String, path to ssl client cert file (.pem). If Tuple, ('cert', 'key') pair.

Return type requests.Response

resolve_redirects (*resp, req, stream=False, timeout=None, verify=True, cert=None, proxies=None, yield_requests=False, **adapter_kwargs*)
Receives a Response. Returns a generator of Responses or Requests.

response_hook (*response, **kwargs*) → requests_html.HTMLResponse
Change response encoding and replace it by a HTMLResponse.

send (*request, **kwargs*)
Send a given PreparedRequest.

Return type requests.Response

should_strip_auth (*old_url, new_url*)
Decide whether Authorization header should be removed when redirecting

class requests_html.AsyncHTMLSession (*loop=None, workers=None, mock_browser: bool = True, *args, **kwargs*)
An async consumable session.

close ()
If a browser was created close it first.

delete (*url, **kwargs*)
Sends a DELETE request. Returns Response object.

Parameters

- **url** – URL for the new Request object.
- ****kwargs** – Optional arguments that request takes.

Return type requests.Response

get (*url*, ***kwargs*)

Sends a GET request. Returns `Response` object.

Parameters

- **url** – URL for the new `Request` object.
- ****kwargs** – Optional arguments that `request` takes.

Return type `requests.Response`

get_adapter (*url*)

Returns the appropriate connection adapter for the given URL.

Return type `requests.adapters.BaseAdapter`

get_redirect_target (*resp*)

Receives a `Response`. Returns a redirect URI or `None`

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

Sends a HEAD request. Returns `Response` object.

Parameters

- **url** – URL for the new `Request` object.
- ****kwargs** – Optional arguments that `request` takes.

Return type `requests.Response`

merge_environment_settings (*url*, *proxies*, *stream*, *verify*, *cert*)

Check the environment and merge it with some settings.

Return type `dict`

mount (*prefix*, *adapter*)

Registers a connection adapter to a prefix.

Adapters are sorted in descending order by prefix length.

options (*url*, ***kwargs*)

Sends a OPTIONS request. Returns `Response` object.

Parameters

- **url** – URL for the new `Request` object.
- ****kwargs** – Optional arguments that `request` takes.

Return type `requests.Response`

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

Sends a PATCH request. Returns `Response` object.

Parameters

- **url** – URL for the new `Request` object.
- **data** – (optional) Dictionary, list of tuples, bytes, or file-like object to send in the body of the `Request`.
- ****kwargs** – Optional arguments that `request` takes.

Return type `requests.Response`

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

Sends a POST request. Returns `Response` object.

Parameters

- **url** – URL for the new `Request` object.
- **data** – (optional) Dictionary, list of tuples, bytes, or file-like object to send in the body of the `Request`.
- **json** – (optional) json to send in the body of the `Request`.
- ****kwargs** – Optional arguments that `request` takes.

Return type `requests.Response`**prepare_request** (*request*)

Constructs a `PreparedRequest` for transmission and returns it. The `PreparedRequest` has settings merged from the `Request` instance and those of the `Session`.

Parameters **request** – `Request` instance to prepare with this session's settings.**Return type** `requests.PreparedRequest`**put** (*url*, *data=None*, ***kwargs*)

Sends a PUT request. Returns `Response` object.

Parameters

- **url** – URL for the new `Request` object.
- **data** – (optional) Dictionary, list of tuples, bytes, or file-like object to send in the body of the `Request`.
- ****kwargs** – Optional arguments that `request` takes.

Return type `requests.Response`**rebuild_auth** (*prepared_request*, *response*)

When being redirected we may want to strip authentication from the request to avoid leaking credentials. This method intelligently removes and reapplies authentication where possible to avoid credential loss.

rebuild_method (*prepared_request*, *response*)

When being redirected we may want to change the method of the request based on certain specs or browser behavior.

rebuild_proxies (*prepared_request*, *proxies*)

This method re-evaluates the proxy configuration by considering the environment variables. If we are redirected to a URL covered by `NO_PROXY`, we strip the proxy configuration. Otherwise, we set missing proxy keys for this URL (in case they were stripped by a previous redirect).

This method also replaces the `Proxy-Authorization` header where necessary.

Return type `dict`**request** (**args*, ***kwargs*)

Partial original request func and run it in a thread.

resolve_redirects (*resp*, *req*, *stream=False*, *timeout=None*, *verify=True*, *cert=None*, *proxies=None*, *yield_requests=False*, ***adapter_kwargs*)

Receives a `Response`. Returns a generator of `Responses` or `Requests`.

response_hook (*response*, ***kwargs*) → `requests_html.HTMLResponse`

Change response encoding and replace it by a `HTMLResponse`.

run (**coros*)

Pass in all the coroutines you want to run, it will wrap each one in a task, run it and wait for the result. Return a list with all results, this is returned in the same order coros are passed in.

send (*request*, ***kwargs*)

Send a given PreparedRequest.

Return type requests.Response

should_strip_auth (*old_url*, *new_url*)

Decide whether Authorization header should be removed when redirecting

CHAPTER 7

Indices and tables

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

r

`requests_html`, [15](#)

A

`absolute_links` (*requests_html.Element* attribute), 18
`absolute_links` (*requests_html.HTML* attribute), 15
`arender()` (*requests_html.HTML* method), 15
`AsyncHTMLSession` (class in *requests_html*), 22
`attrs` (*requests_html.Element* attribute), 18

B

`base_url` (*requests_html.Element* attribute), 18
`base_url` (*requests_html.HTML* attribute), 15

C

`close()` (*requests_html.AsyncHTMLSession* method), 22
`close()` (*requests_html.HTMLSession* method), 19

D

`delete()` (*requests_html.AsyncHTMLSession* method), 22
`delete()` (*requests_html.HTMLSession* method), 19

E

`Element` (class in *requests_html*), 17
`encoding` (*requests_html.Element* attribute), 18
`encoding` (*requests_html.HTML* attribute), 15

F

`find()` (*requests_html.Element* method), 18
`find()` (*requests_html.HTML* method), 15
`full_text` (*requests_html.Element* attribute), 18
`full_text` (*requests_html.HTML* attribute), 16

G

`get()` (*requests_html.AsyncHTMLSession* method), 22
`get()` (*requests_html.HTMLSession* method), 20
`get_adapter()` (*requests_html.AsyncHTMLSession* method), 23

`get_adapter()` (*requests_html.HTMLSession* method), 20
`get_redirect_target()` (*requests_html.AsyncHTMLSession* method), 23
`get_redirect_target()` (*requests_html.HTMLSession* method), 20

H

`head()` (*requests_html.AsyncHTMLSession* method), 23
`head()` (*requests_html.HTMLSession* method), 20
`HTML` (class in *requests_html*), 15
`html` (*requests_html.Element* attribute), 18
`html` (*requests_html.HTML* attribute), 16
`HTMLSession` (class in *requests_html*), 19

L

`links` (*requests_html.Element* attribute), 18
`links` (*requests_html.HTML* attribute), 16
`lxml` (*requests_html.Element* attribute), 18
`lxml` (*requests_html.HTML* attribute), 16

M

`merge_environment_settings()` (*requests_html.AsyncHTMLSession* method), 23
`merge_environment_settings()` (*requests_html.HTMLSession* method), 20
`mount()` (*requests_html.AsyncHTMLSession* method), 23
`mount()` (*requests_html.HTMLSession* method), 20

N

`next()` (*requests_html.HTML* method), 16

O

`options()` (*requests_html.AsyncHTMLSession* method), 23
`options()` (*requests_html.HTMLSession* method), 20

P

`patch()` (*requests_html.AsyncHTMLSession method*), 23
`patch()` (*requests_html.HTMLSession method*), 20
`post()` (*requests_html.AsyncHTMLSession method*), 23
`post()` (*requests_html.HTMLSession method*), 20
`pq()` (*requests_html.Element attribute*), 18
`pq()` (*requests_html.HTML attribute*), 16
`prepare_request()` (*requests_html.AsyncHTMLSession method*), 24
`prepare_request()` (*requests_html.HTMLSession method*), 21
`put()` (*requests_html.AsyncHTMLSession method*), 24
`put()` (*requests_html.HTMLSession method*), 21

R

`raw_html()` (*requests_html.Element attribute*), 18
`raw_html()` (*requests_html.HTML attribute*), 16
`rebuild_auth()` (*requests_html.AsyncHTMLSession method*), 24
`rebuild_auth()` (*requests_html.HTMLSession method*), 21
`rebuild_method()` (*requests_html.AsyncHTMLSession method*), 24
`rebuild_method()` (*requests_html.HTMLSession method*), 21
`rebuild_proxies()` (*requests_html.AsyncHTMLSession method*), 24
`rebuild_proxies()` (*requests_html.HTMLSession method*), 21
`render()` (*requests_html.HTML method*), 16
`request()` (*requests_html.AsyncHTMLSession method*), 24
`request()` (*requests_html.HTMLSession method*), 21
`requests_html` (*module*), 15
`resolve_redirects()` (*requests_html.AsyncHTMLSession method*), 24
`resolve_redirects()` (*requests_html.HTMLSession method*), 22
`response_hook()` (*requests_html.AsyncHTMLSession method*), 24
`response_hook()` (*requests_html.HTMLSession method*), 22
`run()` (*requests_html.AsyncHTMLSession method*), 24

S

`search()` (*requests_html.Element method*), 19
`search()` (*requests_html.HTML method*), 17

`search_all()` (*requests_html.Element method*), 19
`search_all()` (*requests_html.HTML method*), 17
`send()` (*requests_html.AsyncHTMLSession method*), 24
`send()` (*requests_html.HTMLSession method*), 22
`should_strip_auth()` (*requests_html.AsyncHTMLSession method*), 25
`should_strip_auth()` (*requests_html.HTMLSession method*), 22

T

`text()` (*requests_html.Element attribute*), 19
`text()` (*requests_html.HTML attribute*), 17

U

`user_agent()` (*in module requests_html*), 19

X

`xpath()` (*requests_html.Element method*), 19
`xpath()` (*requests_html.HTML method*), 17