sprockets.mixins.http

Release 2.6

Mar 07, 2022

Contents

HTTP Client Mixin for Tornado RequestHandlers. Automatically retries on errors, sleep when rate limited, and handles content encoding and decoding using MsgPack and JSON.

CHAPTER 1

Installation

`sprockets.mixins.http` is available on the Python Package Index and can be installed via `pip`:

```
pip install sprockets.mixins.http
```

If you would like to use *tornado.curl_httpclient.CurlAsyncHTTPClient*, you can install pycurl with:

```
pip install sprockets.mixins.http[curl]
```

CHAPTER 2

Documentation

https://sprocketsmixinshttp.readthedocs.io

CHAPTER 3

Requirements

- ietfparse >=1.5.1
- tornado >=5
- sprockets.mixins.mediatype[msgpack] >=3

CHAPTER 4

Example

This examples demonstrates the most basic usage of `sprockets.mixins.http`

```
from tornado import ioloop, web
from sprockets.mixins import http


class RequestHandler(http.HTTPClientMixin, web.RequestHandler):

    async def get(self, *args, **kwargs):
        response = await self.http_fetch('https://api.github.com')
        if not response.ok:
            self.set_status(response.code)
        self.write(response.body)


if __name__ == "__main__":
    app = web.Application([(r'/', RequestHandler)])
    app.listen(8000)
    ioloop.IOLoop.current().start()
```

As with Tornado, to use the curl client which has numerous benefits:

```
from tornado import httpclient, ioloop, web
from sprockets.mixins import http

httpclient.AsyncHTTPClient.configure(
    'tornado.curl_httpclient.CurlAsyncHTTPClient')


class RequestHandler(http.HTTPClientMixin, web.RequestHandler):

    async def get(self, *args, **kwargs):
        response = await self.http_fetch('https://api.github.com')
        if not response.ok:
```

(continues on next page)

(continued from previous page)

```
            self.set_status(response.code)
        self.write(response.body)


if __name__ == "__main__":
    app = web.Application([(r'/', RequestHandler)])
    app.listen(8000)
    ioloop.IOLoop.current().start()
```

CHAPTER 5

Error Response Body

For errors, i.e. a response with HTTP status code in the 400 range...

The HTTPResponse object's body is reduced down to just the error message. That is this mixin's default behavior.

For a JSON response body with Problem Details (RFC 7807), you may want more than just the error message. To gain access to the complete, deserialized response body; a class that uses this mixin can set:

```
self.simplify_error_response = False
```

CHAPTER 6

Environment Variables

HTTP_MAX_CLIENTS	An optional setting that specifies the maximum number of simultaneous asynchronous HTTP requests. If not specified, the default Tornado value of 10 will be used.

CHAPTER 7

License

`sprockets.mixins.http` is released under the 3-Clause BSD license.

CHAPTER 8

Issues

Please report any issues to the Github project at https://github.com/sprockets/sprockets.mixins.http/issues

8.1 API

class sprockets.mixins.http.**HTTPClientMixin**(**args*, ***kwargs*)

Mixin for making http requests using the asynchronous *http_fetch()* method.

_http_req_apply_default_headers(*request_headers*, *content_type*, *body*)

Set default values for common HTTP request headers

Parameters

- **request_headers** (*dict*) – The HTTP request headers
- **content_type** (ietfparse.datastructures.ContentType or str) – The mime-type used in the request/response
- **body** (*mixed*) – The request body

Return type dict

_http_req_body_serialize(*body*, *content_type*)

Conditionally serialize the request body value if mime_type is set and it's serializable.

Parameters

- **body** (*mixed*) – The request body
- **content_type** (*str*) – The content type for the request body

Raises ValueError

http_fetch(*url*, *method='GET'*, *request_headers=None*, *body=None*, *content_type=<ietfparse.datastructures.ContentType application/msgpack, 0 parameters>*, *follow_redirects=False*, *max_redirects=None*, *connect_timeout=None*, *request_timeout=None*, *retry_timeout=None*, *max_http_attempts=None*, *auth_username=None*, *auth_password=None*, *user_agent=None*, *validate_cert=True*, *allow_nonstandard_methods=False*, *dont_retry=None*, ***kwargs*)

Perform a HTTP request

Will retry up to *max_http_attempts* times with an exponentially increasing sleep time starting with *retry_timeout* seconds. If a `Retry-Header` is included in a response, then it will override the calculated sleep time.

Parameters

- **url** (*str*) – The URL for the request
- **method** (*str*) – The HTTP request method, defaults to `GET`
- **request_headers** (*dict*) – Headers to include in the HTTP request
- **body** (*mixed*) – The HTTP request body to send with the request
- **content_type** (`ContentType` or str) – The mime type to use for requests & responses. Defaults to `application/msgpack`
- **follow_redirects** (*bool*) – Follow HTTP redirects when received
- **max_redirects** (*int*) – Maximum number of redirects to follow, default is 5
- **connect_timeout** (*float*) – Timeout for initial connection in seconds, default 20 seconds
- **request_timeout** (*float*) – Timeout for entire request in seconds, default 20 seconds
- **retry_timeout** (*float*) – Time to sleep between retries, default 3 seconds
- **max_http_attempts** (*int*) – Maximum number of times to retry a request, default is 3 attempts
- **auth_username** (*str*) – Username for HTTP authentication
- **auth_password** (*str*) – Password for HTTP authentication
- **user_agent** (*str*) – The str used for the `User-Agent` header, default used if unspecified.
- **validate_cert** (*bool*) – For HTTPS requests, validate the server's certificate? Default is True
- **allow_nonstandard_methods** (*bool*) – Allow methods that don't adhere to the HTTP spec.
- **dont_retry** (*set*) – A list of status codes that will not be retried if an error is returned. Default: set({})
- **kwargs** – additional keyword parameters are passed to `tornado.httpclient.AsyncHTTPClient.fetch()`

Return type *HTTPResponse*

Raises `RuntimeError` if the `raise_error` keyword argument is specified

_http_req_user_agent()

Return the User-Agent value to specify in HTTP requests, defaulting to `service/version` if configured in the application settings, or if used in a consumer, it will attempt to obtain a user-agent from the

consumer's process. If it can not auto-set the User-Agent, it defaults to `sprockets.mixins.http/[VERSION]`.

Return type str

_http_req_modify_for_retry(*response: sprockets.mixins.http.HTTPResponse*, *attempt: int*, *url: str*, *headers: dict*, *body*)

Implement this method to modify the request on each attempt.

Parameters

- **response** – the current HTTP response which includes both response and exception history
- **attempt** – current attempt counter
- **url** – current request URL
- **headers** – current request headers
- **body** – serialized request body

The default behavior is to add the `X-Retry-Attempt` header. You will need to implement this for protocols that include a one-time-use value such as the OAuth 1 request nonce.

Returns a tuple containing the URL, headers, and body to use in the next request

class sprockets.mixins.http.**HTTPResponse**(*simplify_error_response=True*)

Encapsulate the response(s) for requests made using the *http_fetch()* method.

attempts

Return the number of HTTP attempts made by calculating the number of exceptions and responses the object contains.

Return type int

body

Returns the HTTP response body, deserialized if possible.

Return type mixed

code

Returns the HTTP status code of the response.

Return type int

duration

Return the calculated duration for the total amount of time across all retries.

Return type float

exceptions

Return the list of exceptions raised when making the request.

Return type list(Exception)

headers

Return the HTTP Response headers as a dict.

Return type dict

history

Return all of the HTTP responses for the request.

Return type list(tornado.httpclient.HTTPResponse)

links
Return the parsed link header if it was set, returning a list of the links as a dict.

Return type list(dict()) or None

ok
Returns *True* if the response status code was between 200 and 399. Returns *False* if no responses were received or the response status code was >= 400.

:rtype bool

raw
Return the raw tornado HTTP Response object

Return type tornado.httpclient.HTTPResponse

8.2 Version History

8.2.1 2.6.0 Mar 07 2022

- Add support for `Content-Type` suffixes (ex. `a/b+json`, `x/y+msgpack`)
- Fix exception when `self.correlation_id` is `None`

8.2.2 2.5.0 Sep 16 2021

- Change `HTTPResponse.links` to return empty list when `Link` header is not present
- Move `X-Retry-Attempt` header insertion into *`_http_req_modify_for_retry()`* hook. This is also needed for using the client with OAuth 1 servers.

8.2.3 2.4.1 Nov 30 2020

- Make request retry timeout configurable
- Apply retry sleeping to all retried attempts
- Use an exponential backoff if `Retry-After` header is absent
- Add `retry_timeout` parameter to *`http_fetch()`*

8.2.4 2.4.0 Nov 3 2020

- Fix serialization of empty request bodies.
- Rate limit 503s as well as 423s and 429s.
- Advertise & test support for Python 3.8 and 3.9.

8.2.5 2.3.3 Apr 8 2020

- Pass keyword parameters through to the underlying HTTPClient fetch method. This enables niceties like streaming callback support

8.2.6 2.3.1 Apr 7, 2020

- Address #27 by using the shortest appropriate timeout

8.2.7 2.3.0 Dec 9, 2019

- Added an option to control response body transformation for errors, i.e. HTTP status code >= 400. By default, a JSON or otherwise structured response body will be reduced down to its error `message`. That can be overridden by setting `simplify_error_response` to False.
- Fix compile-time setting of default argument values in `http_fetch`.

8.2.8 2.2.1 Sep 20, 2019

- Remove the lrucache on response body due to a bug in behavior

8.2.9 2.2.0 Aug 29, 2019

- Add handling of `tornado.httpclient.HTTPTimeoutError` and `tornado.httpclient.HTTPStreamClosedError` exceptions
- Fix documentation builds
- Update documentation links to readthedocs.io

8.2.10 2.1.0 May 7, 2019

- Cast the `url` parameter of `http_fetch` to a string. Allows for native use of URL abstractions like yarl.

8.2.11 2.0.1 Apr 1, 2019

- Fix a bug with the rejected consumer User-Agent behavior

8.2.12 2.0.0 Apr 1, 2019

- Refactor the HTTPResponse to a stand-alone class - Add `history` attribute of the response with all response objects - Add `links` attribute of the response with the parsed link header if set - Add `exceptions` attribute with stack of exceptions returned as responses
- Add `dont_retry` as argument to `http_fetch` method
- Change logging level in a few places to a more appropriate level
- Add support for rejected consumers when auto-creating the `User-Agent` header
- Add the netloc of a request to the log entry created when rate limited
- Use RequestHandler.settings instead of RequestHandler.application.settings when auto-creating the `User-Agent` header for a Tornado request handler
- Add test coverage of the Warning response header behavior

8.2.13 1.1.1 Jan 9, 2019

- Fix failure when response lacks Content-Type header

8.2.14 1.1.0 Oct 11, 2018

- Add logging of response `Warning` headers

8.2.15 1.0.9 Aug 7, 2018

- Add support for Python 3.6 and 3.7
- Add support for Tornado < 6

8.2.16 1.0.8 Feb 7, 2018

- Add `max_redirects` keyword param
- Add `validate_cert` keyword param
- Fix log records always using default number of attempts

8.2.17 1.0.7 Oct 19, 2017

- Change the hard pin requirement on umsgpack

8.2.18 1.0.6 Aug 16, 2017

- Add `max_http_attempts` keyword param

8.2.19 1.0.5 Aug 7, 2017

- Add support for allow_nonstandard_methods and max_clients

8.2.20 1.0.4 May 12, 2017

- Add support for passing the user_agent parameter per request

8.2.21 1.0.3 Apr 28, 2017

- Fix the installer

8.2.22 1.0.2 Apr 26, 2017

- Documentation Updates

8.2.23 1.0.1 Apr 26, 2017

- Default Accept headers include both msgpack and json

8.2.24 1.0.0 Apr 26, 2017

- Initial Version

Index

Symbols

A

B

C

D

E

H

L

O

R