

Urllib3 Library Retry()

We will discuss some useful urllib3 library methods we can utilize when working with this library and in cross collaboration with Python Requests library. The specific method we will discuss is the urllib3 retry method where we can build this out. This can be attached to the Requests library HTTP Adapter and mounted to a sessions object (connection pooling) to work with API's.

Urllib3 Retry Configuration Settings

retry()

- Each retry attempt will create a new Retry object with updated values, so they can be safely reused.
- Retries can be defined as a default for a pool.
- Retries can be disabled by passing False.
- Errors will be wrapped in MaxRetryError unless retries are disabled, in which case the causing exception will be raised

allowed_methods (*Collection*)

Set of uppercased HTTP method verbs that we should retry on. By default, we only retry on methods which are considered to be idempotent (multiple requests with the same parameters end with the same state).

DEFAULT_ALLOWED_METHODS = frozenset({'DELETE', 'GET', 'HEAD', 'OPTIONS', 'PUT', 'TRACE'})

status_forcelist (*Collection*)

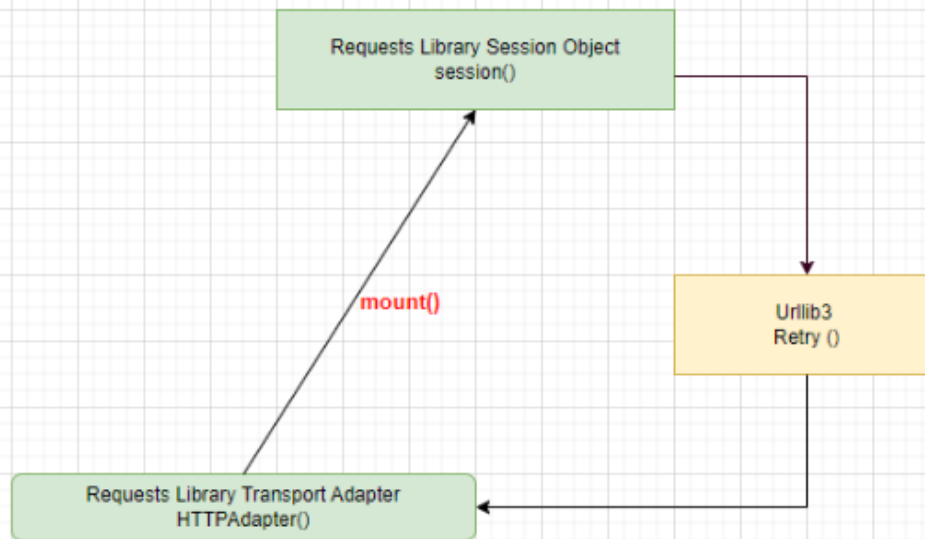
A set of integer HTTP status codes that we should force a retry on. A retry is initiated if the request method is in allowed_methods() and the response status code is in status_forcelist(). By default, this is disabled with None.

backoff_factor (*float*)

A backoff factor to apply between attempts after the second try (most errors are resolved immediately by a second try without a delay). urllib3 will sleep for x number of seconds before attempting the retry.

Discussion

If we build out our retry using Urllib3 library initiating our Requests Session object and mounting HTTP Adapter with pre-defined URllib3 retry configuration if we input status_forcelist does this define the status codes we want to retry on, meaning anything outside this collection will not be retried. This will require further research, testing and analysis.



Urllib3 Retry Workflow

When we want to implement a retry method when working with API's we must use the following workflow.

- Firstly, we need to initiate a session object from the Requests library in order to be able to mount a HTTP Adapter also from the Requests library.
- We can then build a retry method from Urllib3's library which has numerous methods we can utilize here which can be found on this modules open source documentation.
- Next we can build out HTTP Adapter method from the Requests library including our retry structure built out.
- Once, we have followed the above we can then mount() the pre-defined HTTP Adapter to our initiated sessions object.
- We can then run requests using the session object initiated.