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.

A diagram of a computer program

Description automatically generated