Backend Tests

These tests use unittest python framework.

These tests are intended to ensure the core functionality of the backend API is working. While these tests are not extensive, they will give you a starting point to begin debugging. For example all the test ensure a 200 status code they do not ensure the data returned is valid.

Please add more tests as needed!

1. `test_hashRoute_content()`

   Tests if the API base route is running `200 status code` else fail

   Author: Jalen Hardy

2. `test_hashRoute_content()`

   Sends a post request to the /hashes route with a valid hash
   `Passes if`
   - *200 status code*
   - count = 1
   - hash returned is the same as the one sent
   Else fail

   ** I was careful to rollback the changes to the DB after running this test so I can test if the API is making a requests to the backend. If rolling back the changes is not an option you can get unique hashes from https://virusshare.com/hashes

   Author: Jalen Hardy

3. `test_hashRoute_content_invalid_hash()`

   Sends a post request with an invalid hash 'invalid', this will force the API to make a request to VirusTotal. VirusTotal will throw a '`vt.error.APIError`' the backend catches this error and returns an error message.

   `Passes if`
   - 200 status code
   - status_msg = 'INVALID Hash'

Else fail

Author: Jalen Hardy

4. `test_hash_popularity()`

This test sends a valid request to /hashes and records the hashes popularity. An identical request is immediately sent to /hashes and the second request hash's popularity is recorded

Passes if
- 200 status code
- Popularity $_2$ = Popularity $_1$ + 1

Else fail

Author: Jalen Hardy

5. `test_hash_limit()`

This test sends multiple requests (18) requests sequentially to the API.

Passes if EVERY request
- 200 status code
- count = 1
- Hash sent is the same as the one received
- Length of hash report = 1

Else fail

** I was careful to rollback the changes to the DB after running this test so I can test if the API is making a requests to the backend. If rolling back the changes is not an option you can get unique hashes from https://virusshare.com/hashes

Author: Jalen Hardy

6. `test_extra_route()`

Blackbox tests

Passes if
- 200 status code
- Returns json

- Length of the Date array > 0
- Length of the Popular array > 0

Else fail

Author: Jalen Hardy

Bug Report

1. There were certain hashes which caused the API to crash. The bug was caused by an attribute error. We used an attribute called type_tag but sometime VirusTotal did not send type_tag.

   Resolved: switch type_tag to type

2. Sending invalid hashes causes the API to crash. VirusTotal attempts to lookup the hash but its invalid so VirusTotal throws '`vt.error.APIError`'
   RESOLVED: catch error