

new hampshire



python

group



HTTP(S) Requests

HTTP Methods

GET **Read Resource**

HEAD

POST **Create Resource**

PUT Create or Update Entire Resource

DELETE Delete Resource

CONNECT

OPTIONS

TRACE

PATCH Partial Update Resource

HTTP Methods

	Safe	Idempotent	Cacheable	Accepts Body	Returns Body
GET					
POST			Sometimes		
PUT					
DELETE				May	May
PATCH					

HTTP Response and Error Codes

200 OK

301/307/308: Moved, temporarily redirected, permanently redirected

400 Bad request (it's your fault)

401 Unauthorized (you need to log in)

403 Forbidden (you're logged in, but you still can't do that)

404 Not found (usually a bad URL)

408 Timeout

500 Internal Server Error

501 Not Implemented

502 Bad Gateway

503 Service Unavailable

HTTP Error Codes - Handling Exceptions

Python exceptions:

In `requests` or `httpx`, bad responses are still valid

If you want to explicitly raise an exception based on the error code, use werkzeugs `Aborter` class and nice set of HTTP exceptions

HTTPS

- Please(!) default to HTTPS for clients (like `requests` and `httpx`)
- aka HTTP over TLS or HTTP over SSL
 - Transport Layer Security
 - Secure Sockets Layer (predecessor)
- Security concern is “man in the middle” or “packet sniffing”
- You want to see `TLSError` if the server you’re trying to connect to isn’t secure, you can build some logic or notification around this

```
requests.get('https://evil.org', verify=False) # nope
```

Threading

Requests:

- Python waits for response
- Code execution blocked
- Server status impacts script performance

Consider using threads to do the job of “waiting”

Threading Example 1: Stress test

Send out requests as fast as my laptop can manage

```
1  import requests
2  import threading
3
4  def make_request():
5
6      url="https://www.ruqqus.com"
7      response=requests.get(url)
8
9      print(response.status_code)
10
11 def test_the_server(n=100):
12
13     for i in range(n):
14         new_thread=threading.Thread(target=make_request)
15         new_thread.start()
```


Threading Example 2: Waiting

User submits post with a link

- Read data from submitted form
- Save data as row in “submissions” table in DB
- 301 Redirect user to the post that was just made

Make thumbnail without making user wait

```
#spin off thumbnail generation as new thread
if new_post.url and not new_post.url.endswith(("jpeg", ".jpg", ".png")):
    new_thread=threading.Thread(target=thumbnail_thread, args=(new_post,))
    new_thread.start()
```

```
#continue processing new post stuff
```

thumbs.py - /Users/adam/Documents/GitHub/ruqqus/ruqqus/helpers/thumbs.py (3.8.0)

```
import requests
from os import environ
from urllib.parse import urlparse
```

```
from .get import *
from ruqqus.__main__ import db
```

```
def thumbnail_thread(post):
```

```
    params={"access_key":environ.get("APIFLASH_KEY"),
            "url": post.url,
            "height":720,
            "width":1280,
            "format":"jpeg",
            "response_type":"json",
            "thumbnail_width":300
            }
```

```
x=requests.get("https://api.apiflash.com/v1/urltoimage", params=params)
```

```
post.thumb_id=urlparse(x.json()["url"]).path.split("/")[1].split(".")[0]
```

```
db.add(post)
db.commit()
```