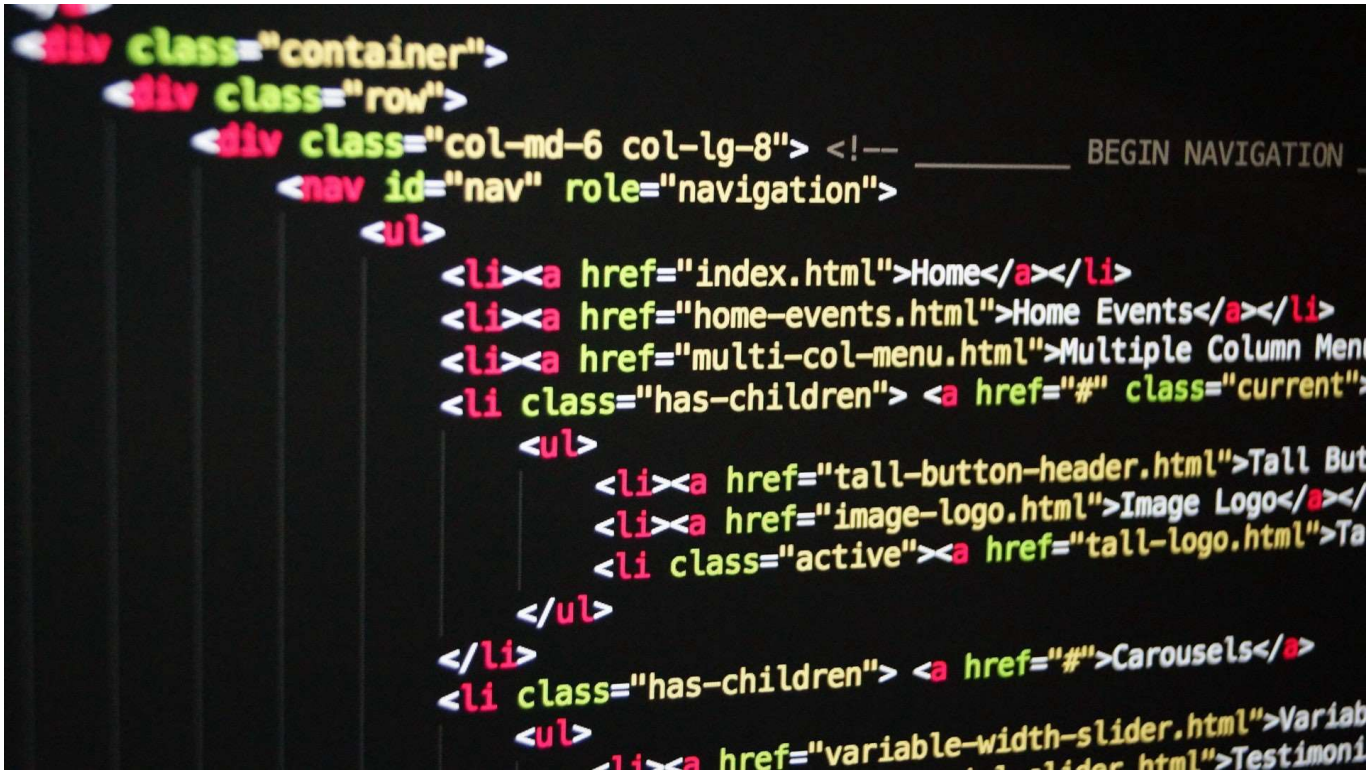
# Requests-Cache Library Tutorial

JASON RIGDEN Jason Rigden Jan 22, 2018 · 2 min read



The Requests library for Python is super popular. In this post, I'll talk about a very helpful companion library called: Request-cache

The Requests-cache library caches HTTP request made with the Request library. That mean that we send a request to a server and then save the response. Then the next time we want to request that resource, we just refer to our saved version. This can be very useful for initial coding and testing projects that use HTTP APIs.

Install

```
pip install requests-cache
```

There are two ways of using Request-cache. In this post, I will only explore the monkey patch way. Using the monkey patch method, allows us to use caching with only a couple lines of code added to you existing code.

Example before caching:

```
import requests
```

```
r = requests.get('http://jasonrigden.com')
```

Example after caching:

```
import requests
import requests_cache

requests_cache.install_cache()

r = requests.get('http://jasonrigden.com')
```

By default the cache is saved in a sqlite database. We could also use a python dict, redis, and mongodb. In this Post we will use the default backend.

We can name the database:

```
requests_cache.install_cache('test_cache')
```

And set the expire time in seconds:

```
requests_cache.install_cache(expire_after=1)
```

We can disable the cache:

```
requests_cache.disabled()
```

We can clear the cache:

```
requests_cache.clear()
```

And remove it:

```
requests_cache.uninstall_cache()
```

We can test if a request was from the cache:

```
r = requests.get('http://jasonrigden.com')
print(r.from_cache)
```

Python Tutorial Requests Library