

Forum:[Hack Forums](#)

Thread: [Sociopaths Tutorials - Starting with Requests](#)

Vassilios - March 19, 2019, 4:01 pm

Sociopaths Tutorials - Starting with Requests | By Nachzehrer

What is the Requests library?

Requests is an Apache2 HTTP library, written for python. Requests allow you to send HTTP/1.1 requests to websites and you can also add headers, form data and parameters. It also allows you to access the contents of the website in a quick and friendly way.

How do I install Requests?

If you have pip installed then simply go to your command prompt and type:

Code pip install requests

Different Types of Requests:

GET: Usually when you make a GET request, the website will return a response to you with the content of the website, unless it has anti-bot protection.

POST: POST requests are made in order to send data on the server, we can log into websites using POST requests and much more depending on the website.

What are headers?

Headers are used between you to pass additional information with each-other such as what kind of browser are you using, what kind of language do you want your content presented in, where did you come from and much more!

Code print(r.headers)

This piece of code should show you what kind of headers your request used, if you want to access a specific one you can do:

Code print(r.headers['name'])

Making GET Requests:

Code r = requests.get('website_url')r: The response from the website, wrapped in a requests object. You might need another library to parse it to a more friendly format such as beautiful soup. Or you can use r.json() or r.content.

requests: The name of the library, we need to use it before any operations that uses its functions.

GET: The operation that we are currently doing, it uses a tuple and it tries to retrieve the contents of the string inside it.

Making POST Requests:

Code payload = {'key1': 'value1', 'key2': 'value2'}

r = requests.post("website", data=payload)

print(r.status_code)r: The response from the website, wrapped in a requests object

requests: The name of the library, we need to use it before any operations that uses its functions.
POST: Here we are doing a POST request so we use post instead of GET.

Now you might see something weird which wasn't seen before. The payload and the data=.

Payload is what we want to send to the server, for example if we tried to log into FaceBook our browser would send the following POST request which we can see from Inspect Element > Network:

Our payload is the "Request Body". We could technically just switch everything and make something such as:

```
Code {'jazoest':'2711', 'lsd':'AVq7w8rg' ... }
```

Then send that to the URL up top together with the headers and we should be able to log in as long as there is no serious bot detection. To send the data we just save our dictionary to a variable such as payload and then data=payload in the post request!

This is all you need to start with Requests! Experiment by yourself and make anything you wish, when you get comfortable you could read on other requests such as PUT.

lockhead - March 20, 2019, 10:23 am

Pretty good tutorial, just a few things..

```
Code r = requests.get("http://mozilla.org")
```

r is not not the content that is returned. r is the response, wrapped in a requests object. Which means if you wanna scrape a website you need r.text or r.content (for binary data) to get the actualy website content.

AeroBlue - March 20, 2019, 10:37 am

(Mar 20, 2019 10:23 AM)lockhead Wrote: [CITE](#) Pretty good tutorial, just a few things..

```
Code r = requests.get("http://mozilla.org")
```

r is not not the content that is returned. r is the response, wrapped in a requests object. Which means if you wanna scrape a website you need r.text or r.content (for binary data) to get the actualy website content.

yea you can also use commands like r.headers or r.json() depending on the situation

Vassilios - March 20, 2019, 10:52 am

(Mar 20, 2019 10:23 AM)lockhead Wrote: [CITE](#) Pretty good tutorial, just a few things..

```
Code r = requests.get("http://mozilla.org")
```

r is not not the content that is returned. r is the response, wrapped in a requests object. Which means if you wanna scrape a website you need r.text or r.content (for binary data) to get the actualy website content.

Thanks for that! I didn't phrase it correctly.

datguy.dev - March 20, 2019, 2:59 pm

I don't know what to request. ?

Vassilios - March 20, 2019, 3:05 pm

(Mar 20, 2019 02:59 PM)datguy.dev Wrote: [\[CITE\]](#) I don't know what to request. ?
Whatever your heart desires.

S2K - March 22, 2019, 4:34 pm

(Mar 19, 2019 04:01 PM)Nachzehirer Wrote: [\[CITE\]](#)
May I ask what is the name of this program?

Vassilios - March 22, 2019, 5:07 pm

(Mar 22, 2019 04:34 PM)Ms-DOs Wrote: [\[CITE\]](#) May I ask what is the name of this program?
Open Inspect Element, go to Networking, find your POST or GET request then click Edit & Resend.