

PYTHON REQUESTS MODULE

Reference

<https://requests.readthedocs.io/en/master/>

<https://docs.python-requests.org/en/master/user/quickstart/>

Definition and Usage

- The requests module allows you to send HTTP requests using Python.
- The HTTP request returns a [Response Object](#) with all the response data (content, encoding, status, etc).

Installation of Requests

- To install Requests, simply run this simple command in your terminal of choice:

```
$ pip3 install requests
```

Syntax

`requests.methodname(params)`

```
>>> r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
'{"type":"User"...}'
>>> r.json()
{'private_gists': 419, 'total_private_repos': 77, ...}
```

Method	Description
<code>delete(url, args)</code>	Sends a DELETE request to the specified url
<code>get(url, params, args)</code>	Sends a GET request to the specified url
<code>head(url, args)</code>	Sends a HEAD request to the specified url
<code>patch(url, data, args)</code>	Sends a PATCH request to the specified url
<code>post(url, data, json, args)</code>	Sends a POST request to the specified url
<code>put(url, data, args)</code>	Sends a PUT request to the specified url
<code>request(method, url, args)</code>	Sends a request of the specified method to the specified url

Make a Request

```
>>> import requests
```

```
>>> r = requests.get('https://api.github.com/events')
```

Passing Parameters In URLs

```
>>> payload = {'key1': 'value1', 'key2': ['value2', 'value3']}  
  
>>> r = requests.get('https://httpbin.org/get', params=payload)  
>>> print(r.url)  
https://httpbin.org/get?key1=value1&key2=value2&key2=value3
```

Custom Headers

```
>>> url = 'https://api.github.com/some/endpoint'  
>>> headers = {'user-agent': 'my-app/0.0.1'}  
  
>>> r = requests.get(url, headers=headers)
```


More complicated POST requests

```
>>> payload = {'key1': 'value1', 'key2': 'value2'}

>>> r = requests.post("https://httpbin.org/post", data=payload)
>>> print(r.text)
{
  ...
  "form": {
    "key2": "value2",
    "key1": "value1"
  },
  ...
}
```

More complicated POST requests

There are times that you may want to send data that is not form-encoded. If you pass in a string instead of a dict, that **data** will be posted directly.

```
>>> import json

>>> url = 'https://api.github.com/some/endpoint'
>>> payload = {'some': 'data'}

>>> r = requests.post(url, data=json.dumps(payload))
```

Instead of encoding the dict yourself, you can also pass it directly using the **json** parameter (added in version 2.4.2) and it will be encoded automatically:

```
>>> url = 'https://api.github.com/some/endpoint'
>>> payload = {'some': 'data'}

>>> r = requests.post(url, json=payload)
```

Response Status Codes

```
>>> r = requests.get('https://httpbin.org/get')  
>>> r.status_code  
200
```

Response Content

```
>>> import requests

>>> r = requests.get('https://api.github.com/events')
>>> r.text
' [{"repository":{"open_issues":0,"url":"https://github.com/...
```

JSON Response Content

```
>>> import requests  
  
>>> r = requests.get('https://api.github.com/events')  
>>> r.json()  
[{'repository': {'open_issues': 0, 'url': 'https://github.com/...
```

Response Headers

We can view the server's response headers using a Python dictionary:

```
>>> r.headers
{
  'content-encoding': 'gzip',
  'transfer-encoding': 'chunked',
  'connection': 'close',
  'server': 'nginx/1.0.4',
  'x-runtime': '148ms',
  'etag': '"e1ca502697e5c9317743dc078f67693f"',
  'content-type': 'application/json'
}
```

Cookies

If a response contains some Cookies, you can quickly access them:

```
>>> url = 'http://example.com/some/cookie/setting/url'
>>> r = requests.get(url)

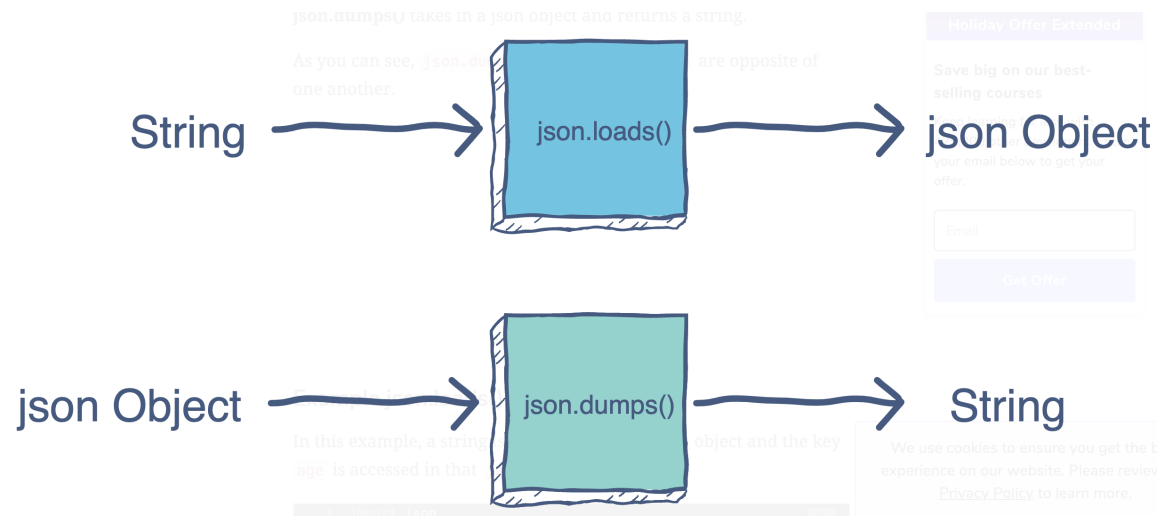
>>> r.cookies['example_cookie_name']
'example_cookie_value'
```

To send your own cookies to the server, you can use the cookies parameter:

```
>>> url = 'https://httpbin.org/cookies'
>>> cookies = dict(cookies_are='working')

>>> r = requests.get(url, cookies=cookies)
>>> r.text
'{"cookies": {"cookies_are": "working"}}'
```

JSON - Loads and Dumps



```
import requests
import json
tenants = requests.get("https://sandboxapicdc.cisco.com/api/node/class/fvTenant.json?rsp-subtree-include=health,faults", headers=header, verify=False)
json_response = json.loads(tenants.text)
print(json.dumps(json_response, sort_keys=True, indent=4))
```


Basic Authentication

```
import requests  
from requests.auth import HTTPBasicAuth
```

```
>>> requests.get('https://api.github.com/user', auth=('user', 'pass'))  
<Response [200]>
```

<https://docs.python-requests.org/en/master/user/authentication/>

