GET and POST requests using Python

geeksforgeeks.org/get-post-requests-using-python

December 7, 2016

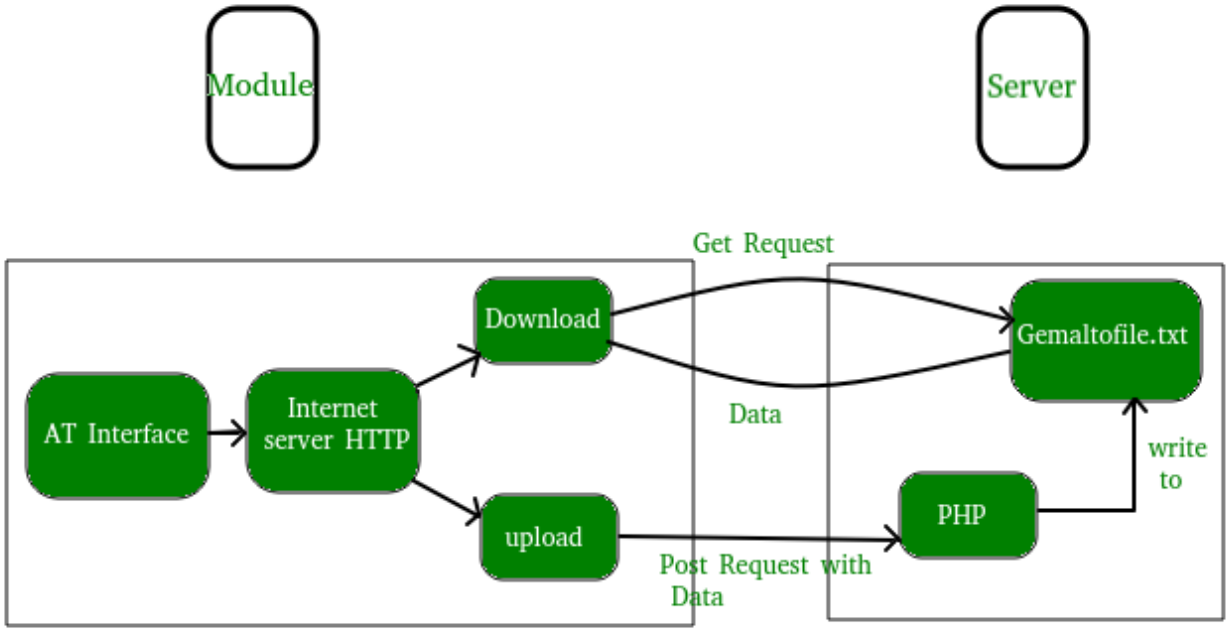
This post discusses two HTTP (Hypertext Transfer Protocol) request methods GET and POST requests in Python and their implementation in python.

What is HTTP?
HTTP is a set of protocols designed to enable communication between clients and servers. It works as a request-response protocol between a client and server.
A web browser may be the client, and an application on a computer that hosts a web site may be the server.

So, to request a response from the server, there are mainly two methods:

1. **GET** : to request data from the server.
2. **POST** : to submit data to be processed to the server.

Here is a simple diagram which explains the basic concept of GET and POST methods.



Now, to make HTTP requests in python, we can use several HTTP libraries like:

- httplib

- urllib
- requests

The most elegant and simplest of above listed libraries is Requests. We will be using requests library in this article. To download and install Requests library, use following command:

```
pip install requests
```

OR, download it from here and install manually.

Making a Get request

filter_none
edit

play_arrow

brightness_4

```
import requests
URL = "http://maps.googleapis.com/maps/api/geocode/json"
location = "delhi technological university"
PARAMS = {'address':location}
r = requests.get(url = URL, params = PARAMS)
data = r.json()
latitude = data['results'][0]['geometry']['location']['lat']
longitude = data['results'][0]['geometry']['location']['lng']
formatted_address = data['results'][0]['formatted_address']
print("Latitude:%s\nLongitude:%s\nFormatted Address:%s"
%(latitude, longitude,formatted_address))
```

Output:

```
Latitude:28.7499867
Longitude:77.1183137
Formatted Address:Delhi Technological University, Shahbad Daulatpur Village, Rohini, Delhi, 110042, India
```

The above example finds latitude, longitude and formatted address of a given location by sending a GET request to the Google Maps API. An API (Application Programming Interface) enables you to access the internal features of a program in a limited fashion. And in most cases, the data provided is in JSON(JavaScript Object Notation) format (which is

implemented as dictionary objects in Python!).

Important points to infer :

```
PARAMS = {'address':location}
```

The URL for a GET request generally carries some parameters with it. For requests library, parameters can be defined as a dictionary. These parameters are later parsed down and added to the base url or the api-endpoint.
To understand the parameters role, try to print **r.url** after the response object is created. You will see something like this:

```
http://maps.googleapis.com/maps/api/geocode/json?address=delhi+technological+university
```

This is the actual URL on which GET request is made

```
r = requests.get(url = URL, params = PARAMS)
```

Here we create a response object 'r' which will store the request-response. We use requests.get() method since we are sending a GET request. The two arguments we pass are url and the parameters dictionary.

```
data = r.json()
```

Now, in order to retrieve the data from the response object, we need to convert the raw response content into a JSON type data structure. This is achieved by using json() method. Finally, we extract the required information by parsing down the JSON type object.

Making a POST request

filter_none
edit

play_arrow

brightness_4

```
import requests
API_ENDPOINT = "http://pastebin.com/api/api_post.php"
API_KEY = "XXXXXXXXXXXXXXXXX"
source_code = 
data = {'api_dev_key':API_KEY,
'api_option':'paste',
'api_paste_code':source_code,
'api_paste_format':'python'}
r = requests.post(url = API_ENDPOINT, data = data)
pastebin_url = r.text
print("The pastebin URL is:%s"%pastebin_url)
```

This example explains how to paste your **source_code** to pastebin.com by sending POST request to the PASTEBIN API.
First of all, you will need to generate an API key by signing up here and then access your API key here.

Important features of this code:

```
data = {'api_dev_key':API_KEY,
        'api_option':'paste',
        'api_paste_code':source_code,
        'api_paste_format':'python'}
```

Here again, we will need to pass some data to API server. We store this data as a dictionary.

```
r = requests.post(url = API_ENDPOINT, data = data)
```

Here we create a response object 'r' which will store the request-response. We use requests.post() method since we are sending a POST request. The two arguments we pass are url and the data dictionary.

```
pastebin_url = r.text
```

In response, the server processes the data sent to it and sends the pastebin URL of your **source_code** which can be simply accessed by **r.text .**

requests.post method could be used for many other tasks as well like filling and submitting the web forms, posting on your FB timeline using the Facebook Graph API, etc.

Here are some important points to ponder upon:

- When the method is GET, all form data is encoded into the URL, appended to the **action** URL as query string parameters. With POST, form data appears within the **message body** of the HTTP request.
- In GET method, the parameter data is limited to what we can stuff into the request line (URL). Safest to use less than 2K of parameters, some servers handle up to 64K.No such problem in POST method since we send data in **message body** of the HTTP request, not the URL.
- Only ASCII characters are allowed for data to be sent in GET method.There is no such restriction in POST method.
- GET is less secure compared to POST because data sent is part of the URL. So, GET method should not be used when sending passwords or other sensitive information.

This blog is contributed by **Nikhil Kumar**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.