

59 I want to dynamically query Google Maps through the Google Directions API. As an example, this request calculates the route from Chicago, IL to Los Angeles, CA via two waypoints in Joplin, MO and Oklahoma City, OK:

<http://maps.googleapis.com/maps/api/directions/json?origin=Chicago,IL&destination=Los+Angeles,CA&waypoints=Joplin,MO|Oklahoma+City,OK&sensor=false>

37 It returns a result [in the JSON format](#).

How can I do this in Python? I want to send such a request, receive the result and parse it.

`python` `json`

share improve this question

edited Oct 8 '15 at 15:42

asked Jun 17 '11 at 13:17

[approxiblue](#)



[Arun](#)

4,611 12 28 45

504 2 7 10

add a comment

5 Answers

active

oldest

votes

I recommend using the awesome [requests](#) library:

117 `import json, requests`

```
url = 'http://maps.googleapis.com/maps/api/directions/json'
```

```
params = dict(
    origin='Chicago,IL',
    destination='Los+Angeles,CA',
    waypoints='Joplin,MO|Oklahoma+City,OK',
    sensor='false'
)
```

```
resp = requests.get(url=url, params=params)
data = json.loads(resp.text)
```

share improve this answer

edited Jun 10 '14 at 20:58

answered Jun 17 '11 at 13:22

[nyuszika7h](#)

[zeekay](#)



4,091 2 31 45



31.5k 5 74 92

6 This doesn't work (i'm assuming because there is no longer a read() function in requests). The last line should be: `data = json.loads(resp.content)` – [slykat Sep 20 '13 at 3:24](#)

1 I needed to change the last line to `data = json.loads(resp.text)` – [user2601995 Apr 11 '14 at 3:33](#)

@SHernandez That's not portable, as they made an API breaking change in requests v1 (`json` is a method rather than an attribute now), and Debian's current stable version (wheezy) is still using v0. (Although v2 is now available in backports.) – [nyuszika7h Jun 10 '14 at 20:50](#)

ok sorry for the confusion. I am deleting my older comments to clean this up – [SHernandez Jun 10 '14 at 20:56](#)

24 `resp.json()` also works. – [Cees Timmerman Mar 23 '15 at 9:07](#)

This might not be obvious to some but I'll just say that you also need to print the data afterwards :P – [nmuJan 21 '16 at 21:57](#)

```
output of r = requests.get(url, verify=False)
print(r.json()) simplejson.scanner.JSONDecodeError: Expecting value: line 1 column 1
(char 0) output of data = json.load(r.text) ValueError: No JSON object could be
decoded so it did not work for me – Umesh Kaushik Apr 13 at 10:45
```

[add a comment](#)

The `requests` Python module takes care of both retrieving JSON data and decoding it, due to its builtin JSON decoder. Here is an example taken from [the module's documentation](#):

61

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

So there is no use of having to use some separate module for decoding JSON.

[share](#) [improve this answer](#)

[edited Nov 2 '13 at 8:50](#)

[answered Jul 8 '13 at 0:06](#)



Pierre

3,860 2 22 43



Andrei Horak

5,005 6 56 115

-
- 4 If you need to be compatible with requests 0.x (Debian wheezy), you should use `json.load()` or `json.loads()` instead, as in 0.x, `json` is a property rather than a function. – [nyuszika7h Dec 9 '13 at 14:56](#)
-
- 2 @nyuszika If you are using debian, if somehow possible, use pip to get newer python libraries. You don't want to code with old python libraries, unless there is an important reason to use what debian has in the apt repositories. – [SHernandez Jun 10 '14 at 12:46](#)
- @SHernandez That's a valid point, but some packages might depend on the `python-requests` (or `python3-requests`) package, so you will need to install somewhere else than `/usr/local` to avoid breaking those packages. On the other hand, when portability/compatibility is trivial, in my opinion it's worth it.– [nyuszika7h Jun 10 '14 at 20:53](#)
-
- 2 How to extract only a particular name-value pair from the json response 'r' ? – [Nikhil George Jan 17 '15 at 16:37](#)
-
- 1 In `r.json()` (from my answer) you have the actual response, JSON-decoded. You can access it like a normal `list / dict`; `print r.json()` to see how it looks like. Or refer to the API docs of the service you've made the request for. – [Andrei Horak Jan 17 '15 at 18:16](#)
-

[show 2 more comments](#)requests has built-in `.json()` method

13

```
import requests
requests.get(url).json()
```

share improve this answer

answered Nov 5 '13 at 9:19

[maow](#)

513 4 13

[add a comment](#)

12

```
import urllib
import json

url = 'http://maps.googleapis.com/maps/api/directions/json?origin=Chicago,IL&destination=Los
result = json.load(urllib.urlopen(url))
```

share improve this answer

answered Jun 17 '11 at 13:22



[clyfish](#)

5,439 1 21 23

2 Thanks for your help, however the following is to be noted : The urllib.urlopen() function has been removed in Python 3.0 in favor of urllib2.urlopen(). – [Arun Jun 17 '11 at 14:00](#)

2 Arun, yes but it's no longer named urllib2 – [Corey Goldberg Jun 17 '11 at 15:09](#)

It's urllib.request in Python 3. – [nyuszika7h Jun 10 '14 at 20:55](#)

It does not work. json.loads gives "TypeError: the JSON object must be str, not 'HTTPResponse'" and json.load gives "TypeError: the JSON object must be str, not 'bytes'" – [M Hornbacher Dec 23 '15 at 17:46](#)

[add a comment](#)

8 Use the requests library, pretty print the results so you can better locate the keys/values you want to extract, and then use nested for loops to parse the data. In the example I extract step by step driving directions.

```
import json, requests, pprint

url = 'http://maps.googleapis.com/maps/api/directions/json?'

params = dict(
    origin='Chicago,IL',
    destination='Los+Angeles,CA',
    waypoints='Joplin,MO|Oklahoma+City,OK',
    sensor='false'
)

data = requests.get(url=url, params=params)
binary = data.content
output = json.loads(binary)

# test to see if the request was valid
#print output['status']

# output all of the results
#pprint.pprint(output)

# step-by-step directions
```

```
for route in output['routes']:
    for leg in route['legs']:
        for step in leg['steps']:
            print step['html_instructions']
```

share improve this answer

edited Mar 22 '13 at 5:56

answered Mar 22 '13 at 1:23



Michael

607 10 37

Michael, how can I make some sense out of this once I got the data? How do I display it in "classic" json visual format (like the one you get in your browser)? Here is what I get in my terminal:

[\[link\]s13.postimg.org/3r55jajk7/terminal.png](https://s13.postimg.org/3r55jajk7/terminal.png) – AlexStarbuck Feb 16 '16 at 9:16

1 @AlexStarbuck import pprint then -> pprint.pprint(step['html_instructions']) – Michael Feb 16 '16 at 14:20
