

Daily Assessment format

Date: 06/06/2020

Course: python

Topic: geocoding

Github

Repository: python-course

Name: Jyoti & Donns

USN: 4AL17E1037

Report

Geocoding request & response (latitude/longitude lookup)
the following example requests the latitude & longitude of "1600 Amphitheatre parkway, Mountain view, CA" & specifies the output must being JSON format. you can test this by entering the url into your web browser (besure to use your API-key with your actual API key). there sponsor includes the latitude & longitude of the address. view the developer's guide for more information about building geocoding request url & available parameters & understanding the response.
Below is a sample geocoding response, in JSON:

```
{
  "results": [
    {
      "address_components": [
        {
          "long_name": "1600",
          "short_name": "1600",
          "types": ["street_number"]
        },
        {
          "long_name": "Amphitheatre parkway",
          "short_name": "Amphitheatre pkwy",
          "types": ["route"]
        }
      ]
    }
  ]
}
```

```

{
  "long_name": "mountainview",
  "short_name": "mountainview",
  "types": ["locality", "political"]
},
{
  "long_name": "Santa Clara County",
  "short_name": "Santa Clara County",
  "types": ["administrative_area_level_2", "political"]
},
{
  "long_name": "California",
  "short_name": "CA",
  "types": ["administrative_area_level_1", "political"]
},
{
  "long_name": "United States",
  "short_name": "US",
  "types": ["Country", "political"]
},
{
  "long_name": "94043",
  "short_name": "94043",
  "types": ["postal_code"]
},
{
  "formatted_address": "1600 Amphitheatre Pkwy, Mountain  
View, CA 94043, USA",
  "geometry": {
    "location": {
      "lat": 37.4267861,
      "lng": 122.0806032
    }
  }
},

```



```

"location_type": "RoofTop",
"viewport": {
  "northeast": {
    "lat": 37.4281350802915
    "lng": -122.0792542197085
  },
  "southeast": {
    "lat": 37.4254371197085
    "lng": -122.0813521802915
  }
},
"place_id": "chIJYnuov25j4ARWU504wWR3GE",
"plus_code": {
  "compound_code": "CWCB+R3",
  "global_code": "849VCWCB+R3",
  "unit": "United States"
},
"types": ["street_address"]
},
"status": "OK"
}

```

Reverse geocoding request & response (address lookup) The following example requests that the address responding to given latitude/longitude in Brooklyn, NY, USA. It specifies that the output must be in JSON format. You can test this by entering the url into your web browser (be sure to replace API key with your actual API key). The response includes the latitude & longitude of the address.

View the developer's guide for more information about building geocoding request url & available parameters & understanding the response.

Below is a sample geocoding response, in JSON:

global_code: 8798270000

"long name" "Gordon Davidson"

"a little name" "Bodgerdrome"

"long name": "Williamburg"

*Child name: William Luigi

"types": ["neighborhood", "political"]

long name: "brooklyn"

chest name Brooklyn

"types": ["political", "sublocality", "sublocality_level1"]

'long name': 'King County'

"lastName": "Longcountly",

type: ("administrative_level": "political")

"long name": "United States"

'shortname': 'united states'

"type": ("country", "political")

```

{
  "long-name": "Newyork",
  "short-name": "Newyork",
  "type": ["administrative_area_level_1", "political"]
},
{
  "long-name": "11211",
  "short-name": "11211",
  "type": ["postal_code"]
},
],
"formatted_address": "279 Bedford Ave, Brooklyn, NY 11211, USA",
"geometry": {
  "location": {
    "lat": 40.7142484,
    "lng": 73.9614103
  },
  "location_type": "RoofTop",
  "viewport": {
    "northeast": {
      "lat": 40.71559738029149,
      "lng": 73.9600613197085
    },
    "southwest": {
      "lat": 40.71287941970849,
      "lng": 73.962755928029151
    }
  }
},
"place_id": "chIJT2x8Q2BZwokRrPBu2jvZX3dE",
"plus_code": {
  "compound_code": "P27Q+Mc Brooklyn, Newyork, United States",
  "global_code": "8798927Q+Mc"
},

```


"type" 1
"battery"
"eyes"
"establishment"
"good"
"point of interest"
"etc."

1.
Additional words furnished in this call...

2.
"stated" "or"