[API] getLocation

1. Arguments

- **Google map key:** google map key

2. Chalice

/Users/dajeongjeon/Desktop/CarVi/environment_loc/test/microservice/chalice2/test70-chalice

API:

General info:

https://jv17oiv64f.execute-api.us-west-2.amazonaws.com/api

1-1. Main usage:

```
https://jv17oiv64f.execute-api.us-west-2.amazonaws.com/api/location
params='{"latlng":"39.7166,116.5420"}'
```


```
{
    "address": "Tongzhou Qu, Beijing Shi",
    "country": "CN",
    "location": "39.7166,116.5420"
}
```


1-2. APP usage:

```
https://jv17oiv64f.execute-api.us-west-2.amazonaws.com/api/getLocation
params='{"source": {"lat": 35.5786,"lng": 139.7447}, "destination":
{"lat": 35.7378,"lng": 139.7604}}'
```


```
{
    "city": {
        "destination": "Kita-ku, Tōkyō-to",
        "source": "Ōta-ku, Tōkyō-to"
    },
    "location": {
        "destination": {
            "lat": 35.7378,
            "lng": 139.7604
        },
        "source": {
            "lat": 35.5786,
            "lng": 139.7447
        }
    }
}
```

3. Code

```
import os, sys, boto3, ast, googlemaps, re
from chalice import Chalice, BadRequestError, NotFoundError
import numpy as np
import pandas as pd
gmaps = googlemaps.Client(key='AIzaSyCL8LykrYfie-rTNsi1KJOkF-n-V0yoct0')

app = Chalice(app_name='test70-chalice')
app.debug = True

# # Index information
@app.route('/')
def index():
    return {
        'status': 'API is available',
        'discription':'This api is for getting an address from given lat and long info',
        'user':'CarVi@ejeon'
        }

@app.route('/location', methods=['POST'], content_types=['application/json'])
def getAddress():
    # latlng = '0.0000,0.0000'
    # http https://jv17oiv64f.execute-api.us-west-2.amazonaws.com/api/location params='{"latlng":"39.716
    param_json = ast.literal_eval(app.current_request.json_body['params'])

    latlng = param_json['latlng']

    try:
        lan, lng= latlng.split(',')
        location = gmaps.reverse_geocode((float(lan),float(lng)))
        lst=[]
        for add in location[0]['address_components']:
            lst.append(add['short_name'])
        address = ','.join(lst)

        country = re.findall('[A-Z][A-Z]', address)[-1]
        address = address.split(',')
        indices = [i for i, s in enumerate(address) if country in s]
        countyIdx = indices[-1] # idx for country code in the address list
        display_add = address[countyIdx-2]+', '+address[countyIdx-1]
    except:
        display_add = ''

    return {'location':latlng, 'address' : display_add, 'country': country}
@app.route('/getLocation', methods=['POST'], content_types=['application/json'])
def getLocation():
    # latlng = '0.0000,0.0000'
    # http https://jv17oiv64f.execute-api.us-west-2.amazonaws.com/api/getLocation params='{"source

    city_json = {}

    try:
        city_json['source'] = getAppAddress(param_json['source'])
        city_json['destination'] = getAppAddress(param_json['destination'])

    except: pass
    response = {'location':param_json, 'city':city_json}

    return response
```

```
(venv) Dajeongs-MacBook-Pro:test70-chalice dajeongjeon$ http https://jv17oiv64f.execute-api.us-west-2.amazonaws.com/api/
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 135
Content-Type: application/json
Date: Fri, 09 Nov 2018 18:14:14 GMT
Via: 1.1 988c38d4cb105c16dc04d6da3789fea9.cloudfront.net (CloudFront)
X-Amz-Cf-Id: q1BYhOHJoULlAhusZIqeU949btoUIWGAXD-rEBIjTkvgOcW97YvgQQ==
X-Amzn-Trace-Id: Root=1-5be5ce76-c8080f35f5261ebcf38b7fcc;Sampled=0
X-Cache: Miss from cloudfront
x-amz-apigw-id: QG0yeHnGPHcFrSg=
x-amzn-RequestId: 43988836-e44b-11e8-bfc2-9155aa462db8

{
    "discription": "This api is for getting an address from given lat and long info",
    "status": "API is available",
    "user": "CarVi@ejeon"
}

(venv) Dajeongs-MacBook-Pro:test70-chalice dajeongjeon$ http https://jv17oiv64f.execute-api.us-west-2.amazonaws.com/api/location params='{"latlng":"39.7166,116.5420"}'
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 88
Content-Type: application/json
Date: Fri, 09 Nov 2018 18:14:20 GMT
Via: 1.1 b9511ab539dfad718af86d0950c2e325.cloudfront.net (CloudFront)
X-Amz-Cf-Id: glxSqDJnw0DPMuMeC99bH5o2axawFTPVNGJQUy_kS5LHruJ_8wYAOQ==
X-Amzn-Trace-Id: Root=1-5be5ce7c-38ed73894e15c0249a95e6e5;Sampled=0
X-Cache: Miss from cloudfront
x-amz-apigw-id: QG0zZHHkvHcFQKQ=
x-amzn-RequestId: 47183942-e44b-11e8-979a-b7c784eca417

{
    "address": "Tongzhou Qu, Beijing Shi",
    "country": "CN",
    "location": "39.7166,116.5420"
}

(venv) Dajeongs-MacBook-Pro:test70-chalice dajeongjeon$ http https://jv17oiv64f.execute-api.us-west-2.amazonaws.com/api/secretinfo
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 77
Content-Type: application/json
Date: Fri, 09 Nov 2018 18:14:26 GMT
Via: 1.1 ae3a2b106871a2f76960e84345c0398f.cloudfront.net (CloudFront)
X-Amz-Cf-Id: tozOO4Y822V9b65v9Eo3YApONEZau5UV2zHrzT3ewuEYFnFbv1w1yw==
X-Amzn-Trace-Id: Root=1-5be5ce82-bff28a578d18287d7dd0615a;Sampled=0
X-Cache: Miss from cloudfront
x-amz-apigw-id: QG00ZHQOPHcFmgQ=
x-amzn-RequestId: 4aed83c1-e44b-11e8-a968-ff13c219bd6f

{
    "secretinfo": {
        "googlemaps key": "AIzaSyCL8LykrYfie-rTNsi1KJOkF-n-V0yoct0"
    }
}

(venv) Dajeongs-MacBook-Pro:test70-chalice dajeongjeon$ http https://jv17oiv64f.execute-api.us-west-2.amazonaws.com/api/getLocation params='{"source": {"lat": 35.5786,"lng": 139.7447}, "destination": {"lat": 35.7378,"lng": 139.7604}}'
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 210
Content-Type: application/json
Date: Sat, 10 Nov 2018 05:48:11 GMT
Via: 1.1 b0e24dd8aec73f015dc37cf4b7de9dea.cloudfront.net (CloudFront)
X-Amz-Cf-Id: 9rhYy4kPKyR-xFMCL6ZdW_lzZmR2RVMdFG86Ap-GqeatbBUCaG9DYg==
X-Amzn-Trace-Id: Root=1-5be67118-c0a449fe56150d5c1f3fc1f8;Sampled=0
X-Cache: Miss from cloudfront
x-amz-apigw-id: QIabxF0WPHcFRrg=
x-amzn-RequestId: 33502bae-e4ac-11e8-b102-6fe4c4e30e9c

{
    "city": {
        "destination": "Kita-ku, Tōkyō-to",
        "source": "Ōta-ku, Tōkyō-to"
    },
    "location": {
        "destination": {
            "lat": 35.7378,
            "lng": 139.7604
        },
        "source": {
            "lat": 35.5786,
            "lng": 139.7447
        }
    }
}
```

[reference]

requirements.txt

```
-i https://pypi.org/simple
numpy==1.15.0
pandas==0.23.4
psycopg2==2.7.5
python-dateutil==2.7.3
pytz==2018.5
sqlalchemy==1.2.10
tornado==5.1.1
gmaps==0.8.2
```