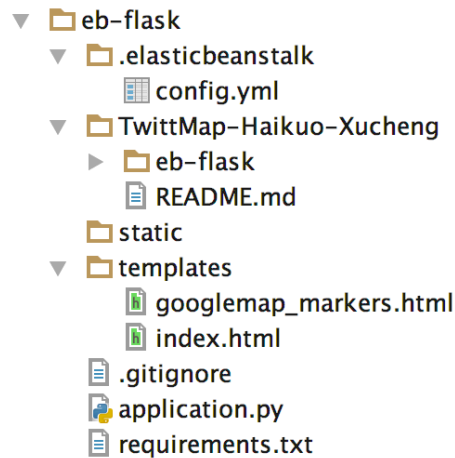


TwittMap Project Report – Haikuo Liu

Haikuo Liu(hl3023)

We've chosen flask as the server and used the jinja2 as the template.

Our file path looks like this:



The application.py (shown below) contains the main logic of our application, and we use index.html as the index page, where there is a drop down of several keywords and you can click on the show button to the benchmarks of filtered twitters.

```
@application.route('/')
def index():
    return render_template('index.html');

@application.route('/data/<keyword>')
def data(keyword):
    r=requests.post('http://search-twitter-1-kf5qeriqw5iu6uasbyv6dmwfbq.us-west-2.es.amazonaws.com/_search/', "", {"que
    results = json.loads(r.content);
    results = results['hits'];
    js_results = "eqfeed_callback(" + json.dumps(results) + "));";
    return js_results;

@application.route('/updatedata/<keyword>')
def updatedata(keyword):
    r=requests.post('http://search-twitter-1-kf5qeriqw5iu6uasbyv6dmwfbq.us-west-2.es.amazonaws.com/_search/', "", {"que
    results = json.loads(r.content);
    results = results['hits'];
    js_results = json.dumps(results);
    return js_results;

@application.route('/googlemap/<keyword>')
def google_map(keyword):
    return render_template('googlemap_markers.html', keyword=keyword)

if __name__ == '__main__':
    application.run()
```

After we press the button, the page will go to hostname/googlemap/<keyword> which will render the Google Map on the page. And here we store the keyword in a hidden div, set the script.src as the path (hostname/data/<keyword>) in the server using the keyword. When the

browser loading the page, it will interpret the script tag and get the contents from this specific path on server.

```
// This example uses a local copy of the GeoJSON stored at
// http://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/2.5_week.geojsonp
script.src = location.origin + "/data/" + document.getElementById('keyword').innerText;
document.getElementsByTagName('head')[0].appendChild(script);
```

In hostname/data/<keyword>, we get the tweets from the Elastic Search server by sending a post request. Then we combine the results and make it a javascript string, return it.

When browser loads the javascript, it'll call the function eqfeed_callback(results), with the tweets fetched from server as parameter.

Then we set the markers on the Google Map using its API.

Here, to update the map in real time, we use Ajax to fetch latest data from server (using basically same method as before). And land the new markers on the map every five seconds.

```
for (var j = 1; j < 10000; j++) {
    !function (i) {
        setTimeout(function () {
            var src = location.origin + "/updatedata/" + document.getElementById('keyword').innerText;
            var results = "";
            $.get(src, function (data) {
                results = data;
            });
            results = JSON.parse(results);
            console.log(results.hits.length);
            for (var i = 0; i < results.hits.length; i++) {
                var latLng = new google.maps.LatLng(results.hits[i]._source.lat, results.hits[i]._source.lon);
                var marker = new google.maps.Marker({
                    position: latLng,
                    map: map
                });
            }
        }, i * 5000);
    }(j)
}
}
```

Finally, we upload the project files to the Elastic Beanstalk using EBCLI:

```
eb-flask — -bash — Solarized Dark ansi — 80x24
application.py          static
[HaikuoLiu@dyn-129-236-214-225:~/PycharmProjects/homework1_haikuo/eb-flask$ eb in]
it -p python2.7 homework4
Application homework4 has been created.
[HaikuoLiu@dyn-129-236-214-225:~/PycharmProjects/homework1_haikuo/eb-flask$ eb cr]
eate homework5
Creating application version archive "app-161024_200001".
Uploading homework4/app-161024_200001.zip to S3. This may take a while.
Upload Complete.
Environment details for: homework5
  Application name: homework4
  Region: us-west-2
  Deployed Version: app-161024_200001
  Environment ID: e-e3hksm3brt
  Platform: 64bit Amazon Linux 2016.03 v2.1.6 running Python 2.7
  Tier: WebServer-Standard
  CNAME: UNKNOWN
  Updated: 2016-10-25 00:00:07.026000+00:00
Printing Status:
INFO: createEnvironment is starting.

[HaikuoLiu@dyn-129-236-214-225:~/PycharmProjects/homework1_haikuo/eb-flask$ ]
[HaikuoLiu@dyn-129-236-214-225:~/PycharmProjects/homework1_haikuo/eb-flask$ ]
```

Overview

[Refresh](#)

Health

Ok

[Causes](#)

Running Version

app-161024_200001

[Upload and Deploy](#)

Configuration

64bit Amazon Linux 2016.03
v2.1.6 running Python 2.7

[Change](#)

Recent Events

[Show All](#)

Time	Type	Details
2016-10-24 20:04:19 UTC-0400	INFO	Successfully launched environment: homework5