Dear all,

Thanks for those who coming today. If you cannot attend the lectures, I hope that at least you catch up with our weekly practices. If you cannot make it, please let me know ASAP!

**Video:**

Recorded full lecture: <https://youtu.be/YO0nrJ7OQ00>

The Power of API: <https://www.youtube.com/watch?v=G1Q9LtnnE4w>

Reading materials about JSON: Working With JSON Data in Python <https://www.realpython.com/python-json/>

**Practice:**

This week’s practice is to develop a sightseeing recommendation application based on AWS Lambda, API Gateway and Amazon Simple Storage Service (S3). The crucial knowledge points covered in the practice are mainly RESTful API and JSON.

1. Given the rough region division of Chile, please find geographical coordinates (Latitude, Longitude) of 6-10 sightseeing points in each of the seven portions. Let’s follow the student list in INFODA to arrange the tasks:

**Norte Grande:** Albornoz Araya Juan Guillermo

**Norte Chico:** Gacitúa Palomino Yonis Esteban Andrés

**Central Chile:** Henríquez Garrido Rodrigo Nicolás

**Southern Chile:** Monje Ourcilleon Martín Enrique

**Los Lagos:** Mora Prado Matías Ignacio

**Aysén:** Muñoz Rebolledo Cristóbal Andrés

**Magallanes:** Soto Videla Joaquín Enrique

.

1. Show up multiple markers on Google Maps by using Javascript. Please refer to the example and directly reuse the source code at

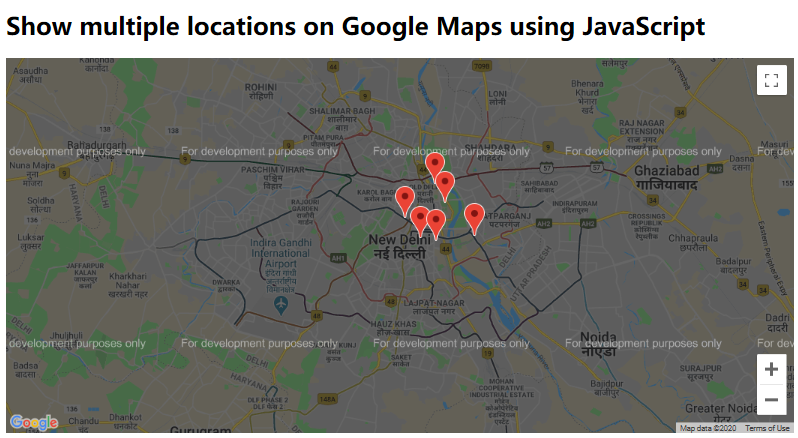
<https://dotnettec.com/how-to-show-multiple-locations-on-google-maps-using-javascript/>

Note that you need to remove the APIKey to make it work in the development mode. i.e. changing

<script type="text/javascript" src="https://maps.googleapis.com/maps/api/js?key=YourAPIKey"></script>

Into:

<script type="text/javascript" src="https://maps.googleapis.com/maps/api/js?key="></script>



1. Try to handle the zooming event and get the zoom level. Note that the default zoom level should be 4. Please refer to the example at

<https://developers.google.com/maps/documentation/javascript/examples/event-properties>

        map.addListener('zoom\_changed', function() {  
          infowindow.setContent('Zoom: ' + map.getZoom());  
        }

1. Given the received zoom level, correspondingly change the markers by using the following number icons. Please refer to the example at

<https://developers.google.com/maps/documentation/javascript/custom-markers>

Eventually, we will use the number icons to show not only the locations, but also the size of the location clusters.

1. Obtain the locations and clusters information from an AWS Lambda function. The logic in the Lambda function has been explained in the lecture. Please pre-calculate all the necessary cluster locations and directly hard code them into your lambda function.

Please refer to the tutorial of making the lambda function accessible via API.

<https://medium.com/@dwdraju/python-function-on-aws-lambda-with-api-gateway-endpoint-288eae7617cb>

<http://www.awslessons.com/2017/setting-up-lambda-with-api-gateway/>

You may also refer to the formal tutorial offered by Amazon, but we do NOT use any database yet!

<https://docs.aws.amazon.com/lambda/latest/dg/services-apigateway-tutorial.html>

Please pay attention to the way of preparing JSON-format output (json.dumps). You may want to refer to:

<https://stackoverflow.com/questions/52297481/how-to-retrieve-well-formatted-json-from-aws-lambda-using-python>

Then what about a list of locations? Firstly try a single location, then consider each location as an item to prepare the list of locations, json.dumps will use the same way to convert the list into JSON format. It is normal if you cannot make it work for the first time, but please keep trying (debugging) and you will get more familiar with the JSON format and the usage of both json.dumps and lambda.

<https://www.quora.com/How-do-I-convert-a-list-to-JSON-in-Python>

1. Deploy the developed webpage including the icon files onto Amazon Simple Storage Service (S3). Note that, if the previous tutorial has been gone through for your local version, what you need to do will be the deployment only. Please refer to the instructions of Module 1. Static Web Hosting at

<https://aws.amazon.com/getting-started/hands-on/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/module-1/>

**Requirements and Suggestions:**

There are quite a few trivial tasks to address in this practice. Each of them is not difficult, but combining them together into a single application could be challenging. So please try to work on this practice as early as possible, and you may divide the whole work into three milestones to finish. For example, the first milestone: Task 1-4; the second milestone: Task 5, and the third milestone: Task 6. In addition, please do the six steps one by one along with the documentation. Thus, the previous documented details will significantly help you replicate your previous work if needed especially when debugging.

Note that you need to make a cheat-sheet type of document for me to check your working details, and submit it to me through **email by 6 pm on May 13 (Wednesday)**.

I have made an estimate, and the workload should be suitable as a one-week practice. But please feel free to let me know if you have time conflicts or other difficulties to work it out on time.

The weekly notes will be posted to Canvas, uploaded to INFODA, and emailed out to all of you. You can also leave me feedback or ask questions through emails or the Canvas forum. Let’s try all the communication channels that could facilitate your study.