
This week, we have to complete Attraction and MRT Station APIs and deploy your website to AWS EC2 instance.

For separation of the front-end and back-end architecture, never modify any code marked as # Static Pages.

Part 1-1: Process attraction raw data and save into MySQL database

In the initial project, you get a taipei-attractions.json file in the data folder which includes all the attraction data in JSON format, we will use them later.

For supporting API development, you have to design your MySQL table structure properly. **Write a standalone Python program to load raw data from taipei-attractions.json and save to your database.**

Note: For the image URLs of each attraction, we have to filter out URLs which do not end with JPG/jpg or PNG/png. **Every image URLs should be saved into your database for later use.**

Part 1-2: Complete Attraction APIs

Complete 3 APIs, refer to Attraction and MRT Station sections in the API Specifications.

Note: Image URLs of each attraction and MRT station names are in array format, including 1 or more data.

Part 1-3: Deploy Website to AWS EC2

Launch an AWS EC2 instance with Linux operating system. You can manage your instance and deploy your project on it by remote access tools. Refer to the following steps:

1.
Launch a new instance with Ubuntu (recommended), Red Hat Enterprise Linux (RHEL) or any other Linux operating system. You should choose a t2.micro instance to get the 1 year free plan. After launching, you will get a key file for accessing your remote instance, **keep it safe, keep it secret.**

2.
Refer to the official document, learn how to remote access to your EC2 Instance by command line for later management.

3.
Install git tool in your EC2 Instance if not already installed. Command `sudo` is what you need if you meet any accessing denied issues.
(e.g. `sudo yum install git` in RHEL8)

4.
Use "git clone" and "git pull" commands to clone or pull down the latest version of your project from your GitHub repository to your EC2 instance.

5.
Install/Initiate/Start MySQL 8.x in your EC2 Instance. For practice, try to use the mysql command line interface in your EC2 instance to do some simple operations.

You may use the mysqldump tool to export your database from your local machine, upload to your EC2 instance and import to the MySQL server in the remote instance. Or you can build the same table structure from scratch in the remote MySQL server directly.

6.
Confirm your EC2 Instance is Python ready. You may need python3 and pip3 commands to execute code in Python 3 version. Find ways to install any Python package you want to use in your website project, e.g. FastAPI, Uvicorn and MySQL Driver. **Here, recommend you install packages globally for simplicity, without virtual environment management.**

7.
Try to execute your Python code to start a website server. If you use Uvicorn, you may need to set --host to "0.0.0.0" in your launch command.

If anything is wrong, keep patient, keep track of what you have done/not done. It's not stupid to try these steps again and again until everything is OK.

8.
In the web management interface of AWS EC2 service, get an Elastic IP, associate this IP with your EC2 Instance. Setup inbound rules in the Security setting of your instance to allow port 8000 access. (If you use another port number, replace it with what you actually used.)

If you do not associate Elastic IP with any EC2 instance, you will be charged.

9.
Confirm your website is online: you or anyone can see/use your website by visiting http://YOUR_IP:PORT/. For example, http://140.112.3.4:8000/

10.
We have to run our server in the background, so that even if we close the terminal, our server is still online. You may use the `nohup` command to run any command in the background.
e.g. `nohup uvicorn app:app &`

You may use `top` or any other process management tools to see/manage/kill your background processes.

How to Submit Task:

Following the Git, GitHub workflow we have done in Part 0. Use the develop branch to do any development, deployment, and testing. When you want to submit your task, create a Pull Request in the GitHub website to request merging from develop to main branch. Invite Chao-Wei Peng (cwpeng) as code reviewer, wait for the feedback.

Write down your name and online API URLs in the Pull Request comment.
For example:
彭兆蔚
http://140.112.3.5:8000/api/attractions?page=1
http://140.112.3.5:8000/api/attractions?page=0&keyword=劍潭
http://140.112.3.5:8000/api/attractions?page=0&keyword=北
http://140.112.3.5:8000/api/attraction/10
http://140.112.3.5:8000/api/mrts