Hiding API Keys

Learning Objectives:

- Students will explain how to hide API Keys.
- Students will apply environmental variables to their applications.

Protecting Your Keys

As we know from previous modules, most API require you to have a key or access token. This is so they can see who is accessing their data, and how often. They can also regulate how many times a user calls the api to minimize the load on the server, and to enforce payments.

As developers we must keep our API keys safe from browsing eyes. If we made an api call from our javascript file to a server that needed a key, we could open up inspect and view the api key. This will give others access to your account, and you could end up with a charged account or you might be banned from the api.

The solution is to have our Flask server make the call to consume the api data and hide our key!!!

.env

Many different full stack applications use dot-env to hide their sensitive environmental variables, and Python is no different.

1. Install Python Dot-env:

   `pipenv install python-dotenv`

2. Create a .env file inline with server.py.
3. Declare your variables in .env:

   ```
   FLASK_APP_API_KEY=RANDOMAPIKEY
   ```

4. Access the variables with the os module:

   Anywhere in a python file in your project . . .

   ```
   import os
   print( os.environ.get("FLASK_APP_API_KEY") )
   ```

You can use dot-env for any sensitive variables that you use in your project now!!!

Python v22.1 (Online Part-time)

AJAX

AJAX + API's

AJAX + Flask

Flask w/AJAX ✓

POST FETCH ✓

Username Form ✓

Recipes AJAX ✓

Hiding API Keys

Consuming APIs w/Flask

Super Heros

Project Week

Previous

Next