CSCD 330-040 Computer Networks

Lab 3, Web API Part 2 - Due, October 11, 2023

October 4, 2023

Overview

Let's make our own web API server! Last week we used 2 API servers, this week we'll write our own using the flask library. Your web API server must be able to accept two different API calls.

You'll have two options. If you do both you'll only get points for one!

1. Option 1 is worth 100 of the 100 points and will reuse your code from last week's lab.
2. Option 2 worth 110 of 100 points and will require more data manipulation.

To create a simple web server you'll use the flask library in python. Provided is a simple flask server that you may use as a base for your code regardless of option.

Option 1

Worth 100 points

Step 1 (90):

Write a flask server that can accept 2 different API calls to the following specifications:

1. Accepts a URL or IP address and returns the physical address from the `whois` entry.
2. Accepts a URL or IP address and returns the weather. This is just your lab 2 results as an API call.

Step 2 (10):

Cache the data so that only new requests are run. To save on network traffic make your program remember old requests and reuse the data on subsequent duplicate calls.

Option 2

Worth 110 points.

Step 1:

Download the “global summary of year” data from the NOAA website. You’ll need to download and decompress the archive file (in the `archive` directory). If you are curious as to what the data means, the documentation is in the `doc` directory.

Step 2:

Write python code to parse each CSV file’s data into a dictionary.

Step 3 (100):

Write an API server that returns the data as a json string. You must have 2 different API calls. One to return each of the following:

1. The list of possible files for which your program can retrieve data.
2. The json data of a specific file name.

Turn in:

Submit a tarball of everything (source code, test files, etc.). `tar` is an archiving and compression tool that is very helpful. To “tar up,” say all the pdf files, all the python files, all the bash files and the README in the directory you are in, you would use the command:

```
tar -czvf lab2b_turnin_MYNAME.tar.gz *.sh *.py README
```

I expect a test file/program that runs the program on sample input (3-5). You must submit a script (preferably bash) that call uses `curl` to test you flask server. I should be able to start your server in on terminal and run the test script in another.