Computer Networks Spring 2019 – Lab 2 – RESTful Service

Assigned: 4/10/19

Due: 4/17/19, 11:59pm

This lab is open-ended, like Lab 1, but “higher level” than the first lab. Instead of working with TCP sockets, you will use Python (or whatever language) modules to create a RESTful service. See my example on Canvas (witwidad.py).

1. A RESTful service (45%): design a service. Set it up somewhere with some data.
    a. Suggested approaches:
        i. Create a Firebase database, access endpoints as I did
        ii. Use Flask – Python-based micro-framework, can handle HTTP
    b. Requirements:
        i. At least three levels of links, similar to what we discussed on the “Adding Admin Functionality” slide.
        ii. Do your best to follow a RESTful design – start with a set of links to other objects, which in turn should have links (hence the “three level” requirement)

2. A client (45%): design a client to access your RESTfulservice. The client should be able to access the “top level” set of links and explore all child links from there, operating on each as appropriate. Information about what the client can do should be stored in the objects themselves.

3. README (10%)
    a. Thoroughly document your service so I understand what you did. Explain what each object is, how it is accessed (i.e., the URL), and what is returned. This documentation is not technically necessary if you have a “truly RESTful” service, but it will help me to understand what you did.

4. Note the following:
    a. Programming language choice: up to you. Python3 is encouraged.
    b. Platform: I need to be able to test in Linux (e.g., `luke`) or in Windows 10. If using Python, it should work fine across platforms. You may test on your personal computers, on `luke.cs.spu.edu`, or on computers in OMH (ports 43500-43505).
    c. You may work individually or in pairs for this assignment.
    d. Your code must be commented following good programming practices. In addition to appropriate use of comments throughout your code (e.g., within functions), you must:
        i. Include comments at the top of each file explaining the contents of the file,
        ii. Comment all functions, classes, etc. with explanatory headers (e.g., function parameters, returns, effects),
        iii. Failure to comment appropriately will be -15%.

Turn in: your client program and README.