Lab 2: Building your own web application with a RESTful API

50.012 Networks (2018)

Hand out: 19^{th} Sept (C1) / 21^{st} Sept (C2)
Hand in: 25^{th} Sept (C1) / 27^{th} Sept (C2)

1 Objectives

- Your goal is to implement a RESTful HTTP api for a service of your choice
- Decide on a simple data scenario to represent (*which nouns*?)
- Define a couple of simple RESTful API calls (which verb for which noun?)
- Implement and test the API on your own server running in the lab

2 Set up of your machine

- Connect to `SUTD_student` over the wireless network, test that you have Internet access with `ping` or similar
- For this lab we are going to use `python2`. Install flask with `sudo pip install Flask`

3 Notes

- You may work individually or collaborate with another student. If you collaborate with another student, please hand in code individually with both authors noted in the header.
- Client side: To interact with the API, we suggest either `curl` or Python with `requests` library
  - For a nice `requests` tutorial, see here: `http://www.python-requests.org/en/latest/user/quickstart/`
- Server side: REST `flask` (and `curl`) tutorial: `http://blog.luisrei.com/articles/flaskrest.html`. You can also see https://curl.haxx.se for help with curl.
  - For debugging it helps to turn on flask's debugging functionality by setting: `app.run(debug=True)`
  - In your API, don't just copy the nouns/verbs from the tutorial, this will lose you points!
  - No need to implement everything in the tutorial. We will point you towards specific content in the following.
- For more information about flask visit: `http://flask.pocoo.org/docs/0.12/quickstart/`
  - If you have problem starting the server please look at `http://flask.pocoo.org/docs/0.12/quickstart/#what-to-do-if-the-server-does-not-start` or ask for our help

4 Basic API

- Your goal is to implement a RESTful HTTP api for a service of your choice
- Decide on a simple data scenario to represent (*which nouns*?)
  - Example: SUTD room information service
    - Which floor is the room on?
    - Is this an office, and who is using it?
    - What is the seating capacity?
    - Any other resources in this room (projector)?
    - Maybe even a reservation system for rooms (ours is kindof bad)
  - Simple message board with users and messages (or something like twitter?)
  - Maybe an API that aggregates information from other student's APIs? Get room number for user on message board!
- Define a couple of simple RESTful API calls (which verb for which noun?)
  - You should have at least 5 different API calls/ways to interact with your API
    - At least two nouns, and different verbs (GET/PUT/DELETE)
- Send/receive data in JSON datastructures
  - No need to have something super complicated, a dictionary with a couple of keys should be enough for most things
  - JSON tutorial at `http://www.tutorialspoint.com/json/json_python_example.htm`
  - Implement one of the API calls to support two different mimetypes (as discussed in tutorial and lecture)
    - For example, provide a natural text or JSON version of the same data

5 User Authentication

- You should also implement some simple HTTP authentication with a username and a password
  - No need to implement any HTTPS/TLS yet
- Choose one or more username and passwords, then implement the authentication for at least one API resource as suggested in the tutorial.

6 What to Hand in

6.1 eDimension submission:

- You will submit in the complete API code via eDimension. You may collaborate with a friend on the code, in that case please state both your names in the comments at the start of the file. Both students will then submit the file individually.

- Please add a header at the beginning that comments on the implementation details.
- Please also provide a set of example calls for `curl`, or a Python script that is using `requests` as an HTTP client to test your API

NOTE: Please note that the due dates for submission are 25^{th} Sept for Cohort 1 and 27^{th} Sept for Cohort 2. We will look at the timestamps of your submission to evaluate timely / late submission.

6.2 Checkoff:

- Demo your Python code to the Prof/TA and explain it. In particular:
  - What is the general API setting?
  - Which resources did you implement?
  - Showcase some of the calls you made