VirginiaTech
Invent the Future

Assignment 3

My APIs

Due Date: Mon, November 12, 2018 @ 23:55

ECE 4564 - Network Application Design

ece
BRADLEY DEPARTMENT
of ELECTRICAL & COMPUTER ENGINEERING

Learning Objectives

RESTful API
- Python Requests Library
- Flask Microframework

HTTP Authentication

cURL

Raspberry Pi GPIO (PWM)

noSQL
- MongoDB

Service Advertisement
- Zeroconf

Overview

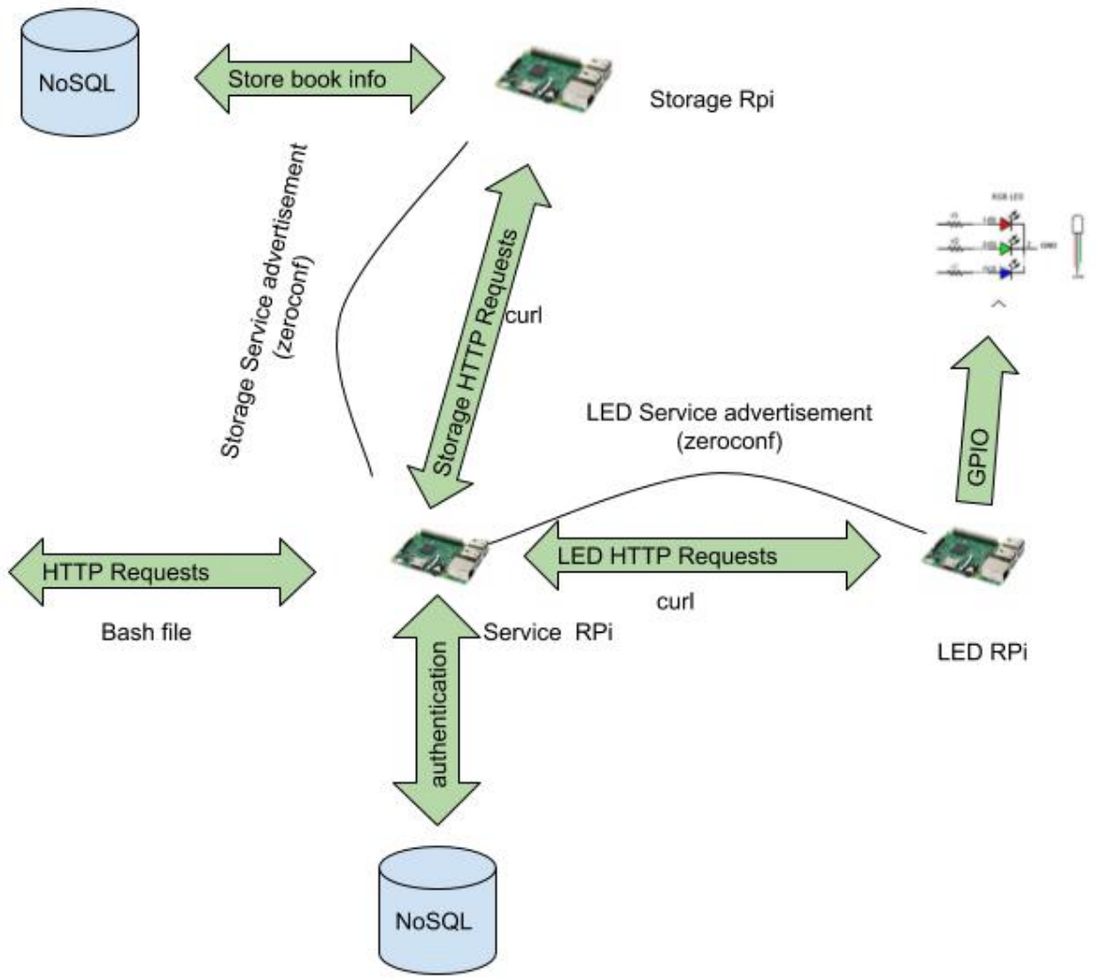
Demonstrate web service interactions using REST

- Build Requirements
  - Flask Microframework
  - Python Requests Library
  - HTTP Basic Authentication
    - Auth data maintained in MongoDB datastore
  - Service Advertisement
    - Zeroconf

- Supported Services
  - RGB LED Controller
  - Storage Server Controller

- All Services should use default flask port number (5000)

System Overview



Overview

- Client
  - Will be a generic desktop, laptop, mobile device, etc.
  - Will make standard HTTP Requests.

- Service Server

  Sole point of contact to entire system for clients:
  - MongoDB & HTTP Authentication
  - Storage APIs
  - LED APIs

Client - Service Server

- Client can be any kind of device that generates properly formatted HTTP Requests (curl, python Requests, etc).
- Client will support file uploading
- All upload request made by the client must go through basic HTTPAuthentication.

Service Server

- Service Server provides add_user api to store new user into MongoDB
- Uses a local NoSQL DB to store usernames and passwords used for Basic HTTPAuthentication.
- MongoDB Entity Format

```
{
   ”username”: “admin”,
   “password”: “password”
}
```

- Please take look api_documentation.txt to find out service server API details

Service Server – Storage Server

- Store the bash file received from /upload/storage in local, and execute it with correct Storage Server advertised server name.
- The bash file template will provide to you.

Service Server – LED Server

- Store the bash file received from /upload/led in local, and execute it with correct LED Server advertised server name.
- The bash file template will provide to you.

Storage Server

- This part is similar as HW2. It is a book inventory server, and should support ADD/LIST/BUY/SELL/DELETE/COUNT 6 actions. However, you will use RESTful API as interface to interact with your local NoSQL DB.

- Announces itself with Zeroconf

- For MongoDB part, you can directly reuse your HW2 MongoDB code.

- `Please take look api_documentation.txt to find out storage server API details`

LED Server

- Supports pulse width modulation (for changing intensity) to manipulate LED on/off, color and brightness.
- Announces itself with Zeroconf
- Please take look api_documentation.txt to find out LED server API details

Sample:

color red with intensity 50

Burning half as bright …

Encapsulation

- Encapsulation is required for MongoDB & LED manipulation part.
- Please put AuthenticationDB & StorageDB classes in two different *.py file.
  - AuthDB.py
  - StorageDB.py
  - LED_PWM.py

Zeroconf

- "The goal of the Zero Configuration Networking (Zeroconf) is to enable networking in the absence of configuration and administration.

- Zero configuration networking is required for environments where administration is impractical or impossible, such as in the home or small office, embedded systems 'plugged together' as in an automobile, or to allow impromptu networks as between the devices of strangers on a train.

Essentially, to reduce network configuration to zero (or near zero) in Internet Protocol (IP) networks, it is necessary, inter alia, to:

- Distribute IP addresses (without a Dynamic Host Configuration Protocol [DHCP] server)
- Provide name resolution (without a Domain Name System [DNS] server)
- Find and list services (without a directory service), and
- Distribute multicast IP addresses, if necessary (without a multicast server)."

Zeroconf Working Group of the Internet Engineering Task Force (IETF)

Zeroconf

- Service Registration (Service Announcement)
  - You can use any name for your service.

- Here is an good example to help you to understand the service announcement & python-zeroconf service browser. However, in our project, you are prohibited to announce service with tool like Avahi.

- You have to register your service with python-zeroconf

Run template bash file.

- LED Manipulation

  bash control_led.sh <LED server IP>

- Storage Server Testing
  - bash storage_test.sh <Storage server IP>

- Service Server
  - Add user
    - bash add_user.sh <Service Server IP>
  - Upload file
    - bash upload-led.sh <Uploaded filename> <Service Server IP>
    - bash upload-storage.sh <Uploaded filename> <Service Server IP>

Execute Uploaded Bash File on Service Server

- File from /upload/led

  bash <uploaded bash file> <LED server announced name>

- File from /upload/storage

  bash <uploaded bash file> <Storage server announced name>

Assignment References

REST and Flask

Serving Raspberry Pi with Flask

Designing a RESTful API with Python and Flask

Flask and MongoDB

Flask Rest API with MongoDB

Python Requests

Requests:HTTP for Humans

cURL

Conquering the Command Line

Assignment References

Authorization

HTTP Basic Auth

GPIO PWM

https://sourceforge.net/p/raspberry-gpio-python/wiki/PWM/

Zeroconf (… start here …)

Register Services (Advertisement)

Brower Services

Grading

GTA will provide grading rubric

Python Style

Follow style guide PEP0008 when writing and commenting your code

https://www.python.org/dev/peps/pep-0008/

What You Turn In

All assignments must be submitted through Canvas, no later than the due date of Nov 12 2018 @ 23:55

Your assignment should be a single tar file which contains the following:

- All source code you wrote for this assignment
  - storage.py
  - service.py
  - led.py
  - LED_PWM.py
  - AuthDB.py
  - StorageDB.py
- Readme
  - System syntax
  - Libraries
  - Description of team member contributions

Academic Integrity

- For this assignment, it is expected that a team's work is their own.
- The code you turn in must be your own (i.e. you need to have written your assignment).
- You are allowed to copy and paste example code from other websites, but you must include a comment in your code that attributes the website you copied the code from (i.e. original author's name and URL to the original code).
- You can discuss the assignment with other teams.
- However, you cannot just tell another team the answer to a particular problem.

Final Thoughts

In many cases, engineers are expected to just make things work given a particular design constraint (e.g. software package to use or are limited to a particular hardware platform).

You will likely run into similar situations in this class while designing and implementing your assignments and project.

.

When you're stuck, try searching online for a solution. Many times others have tried something similar and documented their experiences for others to learn and benefit from

If you find a neat way of doing something on your Raspberry Pi, please share your findings in a discussion post on Canvas.

Do not publically post answers to assignments, or example code until after the assignment due date.

Contact your instructor or GTA as soon as you encounter a problem you're unable to solve.

Don't wait until right before the assignment is due.