Canary Homework Assignment

Introduction

Imagine a system where hundreds of thousands of hardware devices are concurrently uploading sensor data bundles to a service.

Each bundle uploaded to the service is in the following format:

```
{
  "device_uuid": "b21ad0676f26439482cc9b1c7e827de4",
  "sensor_type": "temperature",
  "sensor_value": 50.0,
  "sensor_reading_time": 1510093202
}
```

External clients make requests to this service to retrieve sensor data for a given device in a time range using the following values:

```
{
  "device_uuid": "b21ad0676f26439482cc9b1c7e827de4",
  "sensor_type": "temperature",
  "start_time": 1510093202,
  "end_time" 1510099302
}
```

The service will return a list of bundles for a given device in the time range matching the above format.

Data Constraints

There are 2 valid **sensor_types**:

- humidity
- temperature

The field **sensor_value** is a number between 0.0 and 100.0.
The values in **start_time** and **end_time** are both inclusive.

Service Functionality

1. The service can handle many (but not too many) requests concurrently.
2. When the service receives a PUT operation (can be an HTTP POST, a raw socket message, or any other protocol), it must:
    a. Validate that the input data consists of the fields listed above
    b. Store the data as needed
    c. Return a confirmation message.
3. When the service receives a GET operation (can be an HTTP GET, a raw socket message, or any other protocol), it must:
    a. Validate the query
    b. Return a list of bundles matching the requested criteria
4. The service must be able to handle incorrect data and return errors. The error formats can be in any form, as long as they're consistent and the error codes adhere to the relevant protocol.
5. Don't worry about auth. Assume we live in a world where we can all trust each other.

Deliverables

Without using applications like uwsgi (gevent and other concurrency libraries are fine, however), design and implement a service that adheres to the constraints listed above.

Write the service in one of the following languages: Javascript, Python, Java or Go.

Include a brief README containing instructions on how to run your project and a brief overview of any design decisions you made and their tradeoffs. Diagrams are not necessary, but can be included if they help explain your work. Your code should be tested, and simulated clients are a big plus. This assignment is intended take no longer than 4 hours to complete.

You should use a version control system for your work and send us a link to your repo in an email. If you have any questions, feel free to contact Tiernan Kennedy at tiernan@canary.is.