

Python backend assignment

In this assignment we would like you to create a small application that exposes an API that is able to receive sensor data, save it and return this data in another endpoint.

- We expect you to write the application in Python 3
- The application should host an HTTP API accepting and returning JSON
- Framework choice (and other libraries / tools) is completely up to you.
- Make sure any developer can run your application (provide a requirements.txt)
- Most importantly, we expect you to have a solid justification about your decisions

Scope

We expect the application to have two endpoints, one to receive data, and another to provide the processed data.

- The receiving endpoint should handle a Google Cloud Pub/Sub push handler subscription that sends sensor data (see below)
- You are free to choose the format of the data endpoint
- Take into account that a front-end needs to generate a line graph of the data
- Make sure your API adheres to standards (of your choosing) and returns proper HTTP status codes
- You are free to choose a database of your liking
- The front-end itself is out-of-scope

Bonus points

- Provide an endpoint to return an aggregated "KPI" value
- Make sure the sensor data endpoint is secured in some way so not everyone on the web is able to push new sensor data

Data specifications

Data is coming in from Google Cloud Pub/Sub, the message format is described in the [documentation](#):

```
{
  "message": {
    "attributes": {
      "key": "value"
    },
    "data": "eyJzZXJpYWwiOiIwMDAxMDAwMDAxMDAiLCJhcHBsaWNoZGlvdjI6MTEsIlRpbWUiOiIyMDIyLTExLTQ4VDA0OjAwOjA0LjMxNzgWMSIsIlR5cGUiOiJ4a2d:",
    "messageId": "2070443601311540",
    "message_id": "2070443601311540",
    "publishTime": "2021-02-26T19:13:55.749Z",
    "publish_time": "2021-02-26T19:13:55.749Z"
  },
  "subscription": "projects/myproject/subscriptions/mysubscription"
}
```

The data field is Base64 encoded (see the Pub/Sub documentation) and contains the following JSON data:

```
{
  "serial": "000100000100",
  "application": 11,
  "Time": "2022-11-08T04:00:04.317801",
  "Type": "xkgw",
  "device": "TestDevice",
  "v0": 100013,
  "v1": 0.69,
  "v2": 1.31,
  "v3": 0.18,
  "v4": 0,
  "v5": 0.8,
  "v6": 0,
  "v7": 26965,
  "v8": 0.1,
  "v9": 97757496,
  "v10": 0,
  "v11": 0,
  "v12": 1.84,
  "v13": 0,
  "v14": 0.7,
  "v15": 10010,
  "v16": 100013,
  "v17": 26965,
  "v18": 2.72
}
```

- `v0` contains the origin sensor id.
- `v18` contains the data that we are interested in (we call it "dwell time")
- `Time` contains the timestamp of the data in ISO8601 format
- You can ignore the other fields

For sample data, you can use the included CSV file.

Conditions

Consider not spending more than 4 hours to complete the assignment. In case you are not able to complete the assignment within the given timeframe, consider submitting the assignment with what you have completed.

Submission instructions

All applicants must share their code by creating a zip file of their project folder (excluding virtual environment / dependencies) and submit by either uploading it to a cloud service (sharing the link with us like wetransfer or Google Drive), or sending the link of a public git repository (GitHub).

We would like to get the assignment back from you **48 hours before the interview** at the latest.

Questions

- **Question:** How much time do I have?
- **Answer:** Try to limit your time to 4 hours in total.
- **Question:** Can I use X or Y framework?
- **Answer:** Yes, you can. Just make sure that you can support your architecture decisions.