Uplight Technical Assessment

Introduction

We're excited that you're working through the interview process with us, and we'd love you to complete a small assignment to show off your coding skills.

This assignment is an opportunity for us to see how you approach your work. It will serve as a jumping off point for an interview session with the team. Please only spend 1-2 hours on this assignment; we are not looking for perfect code, but rather good ideas.

Submitting Your Assignment for Evaluation

Please include the following with your submission:

- All source code in an archive format, like zip or tar.gz.
- Documentation on how to run your service. Optionally you may include the following documentation:
    - Any parts of your code that you would like us to pay closer attention to.
    - Any other notes or insights you would like to share.

Reviewer Criteria

We will be reviewing your submission based on the following criteria:

- Does it work? If it throws errors, we'll probably contact you to make sure that it isn't our fault.
- Feature complete: does the service implement all of the specified functionality?
- Code Quality: is the code consistently formatted, are there unit tests, is it readable?

It is not necessary to implement any form of CICD or integration tests. We don't expect any above-and-beyond level of effort.

You are welcome to use any third-party libraries; you are not expected to implement a hashing algorithm from scratch. Please provide the relevant documentation of resources used when submitting the project.

The Assignment: Generate an HMAC Token

HMAC is a type of authentication code that verifies the data integrity and authenticity of a message (https://en.wikipedia.org/wiki/HMAC).

Requirements

1. Implement a POST endpoint that receives a request with a JSON payload
2. Use the request payload to generate the HMAC token.
3. Return a response with a JSON payload that includes a signature field with the generated HMAC token.

We prefer submission done in Python with frameworks like Fast API or Flask, but you are welcome to use the language/stack of your preference. Please include clear instructions for running and testing your code.

Example 1:

Request

```
curl --request POST \
  --url http://localhost:8081/generate-token \
  --header 'Content-Type: application/json' \
  --data '{
    "id": "MDAwMDAwMDAtMDAwMC0wMDBiLTAxMmMtMDllZGU5NDE2MDAz"
  }'
```

Response

```
{
  "id": "MDAwMDAwMDAtMDAwMC0wMDBiLTAxMmMtMDllZGU5NDE2MDAz",
  "signature":
"bf5272df638ab6f15c8af5a5d8d98fa48d573a94f9a3a73bb294853f98174d38a92ae2a68f397e9cd
4a44ade5a9f6f095ae2195a5081e88f45439b3a6bfe05fd"
}
```

Example 2:

Request

```
curl --request POST \
  --url http://localhost:8081/generate-token \
  --header 'Content-Type: application/json' \
  --data '{
    "message":"Apiary: a place where bees and beehives are kept, especially a place
where bees are raised for their honey."
  }'
```

Response

```
{
  "message": "Apiary: a place where bees and beehives are kept, especially a place
where bees are raised for their honey.",
  "signature":
"b99c5564b96173cb14a6571e35fe397895c7d0756c5f8ec8dcd9afb0eeb0ab3b428aebfa6018e3a2495
dd831a37de2fc8806f917bf576706027c2d7d37f39895"
}
```

uplight