UP42 backend coding challenge

We are excited that you made it to the coding challenge stage, and we are looking forward to reviewing your solution.

For this challenge, we want you to build a simple backend application that is capable of working with image metadata, that is typically generated on the UP42 platform.
The goal is to expose three endpoints for *listing features*, *retrieving a specific feature by its ID* and to *return an actual image from a base64 encoded string*.

The data

The given data set has been generated on our platform. It contains a list of features, each representing an actual image, with its associated metadata.

Please treat this data as a *static data source* (download the file and include it in your project). source json data

Expected endpoints

For this challenge, we do expect to see the following endpoints:

```
GET /features
```

Should return a list of features. Please see the desired feature structure below in this document.

```
GET /features/{id}
```

Should return a single representation of a feature. Please see the desired feature structure below in this document.

```
GET /features/{id}/quicklook
```

Should return the image for the given id. You can find the image as base64 encoded string in the static data with the key `quicklook` per feature.

API specification

In general, the API should return results with the `application/json` content type, for the image endpoint `image/png` should be used.

Feature structure

For any feature response, please use the following format. (this example data matches the first item in the static feature collection)

```
{
  "id": "39c2f29e-c0f8-4a39-a98b-deed547d6aea",
  "timestamp": 1554831167697,
  "beginViewingDate": 1554831167697,
  "endViewingDate": 1554831202043,
  "missionName": "Sentinel-1B"
}
```

Requirements and submission guidelines

Please make sure to meet the following requirements for your submission:

- Build your solution in Spring Boot with Java or Kotlin. There are no requirements on specific versions
- Code and commit messages should be treated as you would on a real-world task
- Please take some time to think about code quality and testing, and demonstrate your approach to these aspects
- Provide a README with instructions on how to set up, run and test the application
- When submitting, please provide a link to an online Git repository or a ZIP file which includes the local `.git` folder

We estimate that this challenge should take a maximum of 3 hours to complete, but this is not a hard limit. Feel free to take the time you need to ensure that it is reflective of your skills.

Challenge assessment

We assess the challenge according to the following points:

- Clean code and general code structure (separation of concerns)
- Code quality (testing and maintainability)
- API design and error handling
- Data management

We hope that you will find this challenge enjoyable & we are looking forward to your solution!

- The Engineering Team at *UP42*