Server Side Foundation Assessment

Date: Friday August 12 2022
Assessment Time: 0900 - 1715 (including meal breaks)

Overview

There are **5 tasks** in this assessment. Complete all tasks.

Passing mark is **65%** (**49 marks**). Total marks is **75**.

Read this entire document before attempting the assessment. There are 8 pages in this document.

Assessment Setup

Create a Git repository in Github. This repository must initially be a **PRIVATE** repository. Click on the 'Private' radio button when you create the repository.

Make your repository **PUBLIC** after **1715 Friday Aug 12 2022** so that the instructors can access your work.

IMPORTANT: your assessment repository is PRIVATE and is only accessible to yourself and nobody during the duration of the assessment until **AFTER 1715 Friday Aug 12 2022**. If your work is plagiarised by others, you will be considered as a willing party in the aiding and abetting of the dishonest act.

Clone the repository to your notebook. The following 2 assessment tasks must be created inside this repository. Add a `.gitignore` file to ignore the `target` directory.

In order to safeguard your work, you should commit and push your work often and do not wait until the end of the assessment.

Assessment

In this assessment, you will be writing a SpringBoot application that retrieves the latest crypto currency news from the min-api.cryptocompare.com REST endpoint.

Generate a SpringBoot application Maven project using Java 18; include all the necessary dependencies that you will need in the Maven project in order for you to successfully complete this assessment.

Task 1 (15 Marks)

Use the 'Latest News Articles' (Figure 1) API from the cryptocompare.com site to return a list of news articles.

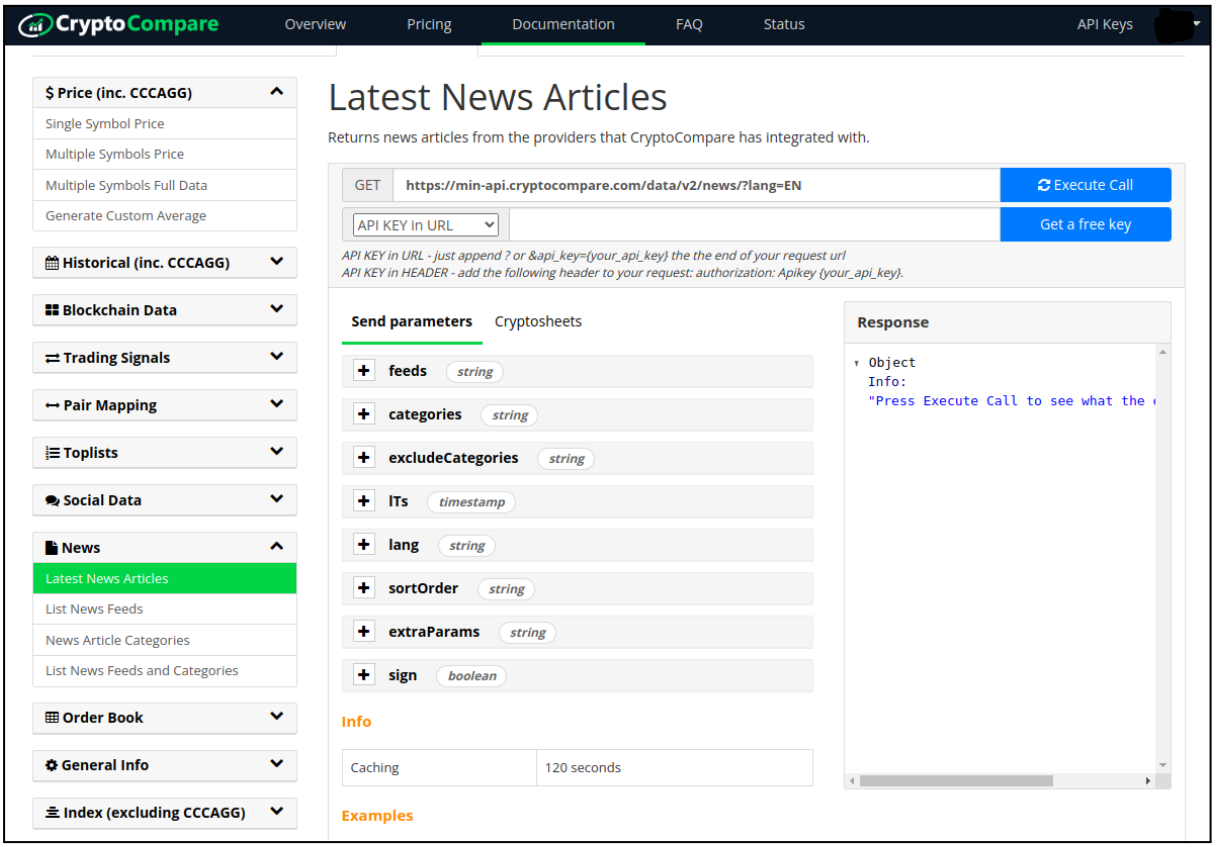

Figure 1

Write a service called `NewsService` that returns a list of news articles from the 'Latest News Articles' endpoint; call the method `getArticles()`.

Extract the following fields from each of the news articles

- `id`
- `published_on`
- `title`
- `url`
- `imageurl`
- `body`
- `tags`
- `categories`

`getArticles()` method should return a `List` of articles.

Task 2 (25 marks)

Write a controller called `NewsController` that will use the `NewsService` (from **Task 1**) to retrieve a list of the latest news articles to generate a view for the following HTTP request

```
GET /
Content-Type: text/html
```

An example of the view return by `NewsController` is shown in Figure 2

Figure 2 List of articles

For each news article, display the following mandatory fields

- id - news article id
- published_on - timestamp from the article
- title
- url - a clickable link that will open the full news article
- imageurl - display the news article image
- body - summary of the article
- tags
- categories
- save - a checkbox to indicate if the article should be saved

An example of this is shown in Figure 3

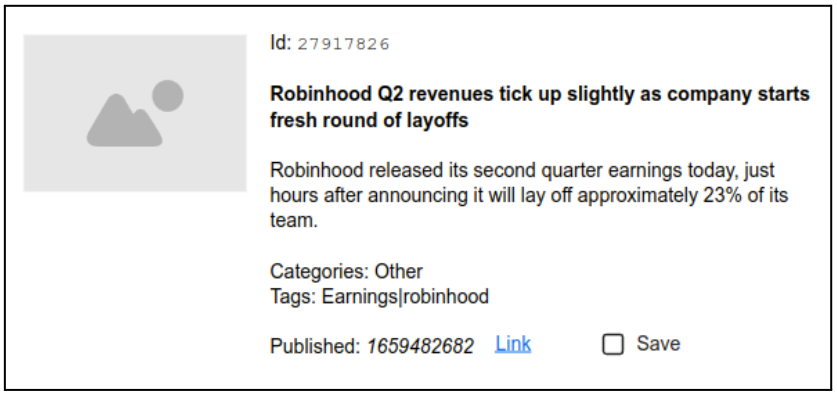

Figure 3 An example of a news article

Include a 'Save' button on the generated view. Users can save one or more articles by selecting the 'Save' checkbox and pressing the 'Save' button (see Figure 2)

When the 'Save' button is pressed, it will make the following request

```
POST /articles
```

Add an additional request handler to `NewsController` to process and save the articles. More details on the save in **Task 3**. Once the articles have been saved, redisplay list of news (Figure 2).

In summary, write a controller called `NewsController` that performs the following

1. Dynamically generate a view in response to a HTTP `GET /` request with the all the latest news
2. Save one or more articles in response to a HTTP `POST /articles` request

Note: You may layout the views (Figure 2, Figure 3) according to your preference

Task 3 (15 marks)

Provision a Redis database in the cloud (eg. Redis Labs) if you haven't already done so.

Write a method called `saveArticles` in `NewsService` that saves selected news articles (from **Task 2**) to your Redis database. The parameter of `saveArticles` method should be a `List` of articles.

Use `saveArticles` in the `POST /articles` request handlers in `NewsController` to persist news articles selected by a user to your Redis database.

Task 4 (15 marks)
Write a REST controller called `NewsRESTController`. This REST controller should handle the following request

```
GET /news/<id>
Accept: application/json
```

where `<id>` is the id of one of the news articles that you have saved to the Redis database (see **Task 3**).

For example, if the request to retrieve the new id `abc123`, the `NewsRESTController` request handler should respond with the following response if the news article exists

```
200 OK
Content-Type: application/json
\r\n
{
     “id”: “abc123”,
     “title”: “...”,
     “body”: “...”,
     “published_on”: “...”,
     “url”: “...”,
     “imageurl”: “...”
     “tags”: “...”
     “categories”: “...”
}
```

If the new article does not exists, the `NewsRESTController` request handler should return the following response

```
404 Not Found
Content-Type: application/json
\r\n
{
     “error”: “Cannot find news article abc123”
}
```

Task 5 (5 marks)
Deploy your assessment to Heroku.

Your Redis password and your min-api.cryptocompare.com API key should not be exposed either in `application.properties` or hard coded in the source code. Marks will be deducted if they are exposed.

Do not undeploy your application from Heroku or tear down your Redis database instance until after **0000 (12AM) Saturday Aug 27 2022**.

Submission

You must submit your assessment by pushing it to your repository at GitHub.

Only commits on or before **1715 Aug 12 2022** will be accepted. Any commits after **1715 Aug 12 2022** will not be accepted. No other form of submission will be accepted (eg. ZIP file).

Remember to **make your repository public after 1715 Aug 12 2022** so the instructors can review your submission.

After committing your work, post the following information to Slack channel `#02-ssf-submission`

1. Your name (as it is shown in your NRIC)
2. Your email
3. Batch - 2a or 2b
4. Git repository URL
5. Heroku deployment URL. Please do not undeploy your application from Heroku or tear down your Redis database until after **0000 (12AM) Saturday Aug 27 2022**

It is your responsibility to ensure that all the above submission requirements are met. Your assessment submission will not be accepted if

1. Any of the 5 items mentioned above is missing from `#02-ssf-submission` channel, and/or
2. Your information did not comply with the submission requirements eg. not providing your full name as per your NRIC, and/or
3. Your repository is not publicly accessible **after 1715 Aug 12 2022**

You should post the submission information to the Slack channel `#02-ssf-submission` **no later than 1730 Aug 12 2022.**

Academic Integrity

This is an open book assessment. You may search the Internet for resources or use reference books during the assessment. The assessment must be your own work. You cannot ask a third party to write any part of this assessment. This will result in an automatic failure.

You are to ensure the integrity and working condition of your PC/notebooks (eg. wireless/internet connection, battery, screen, accidents like water spillage) during the assessment. NUS ISS will not accept any of these as a reason for deferring or retaking your assessment.