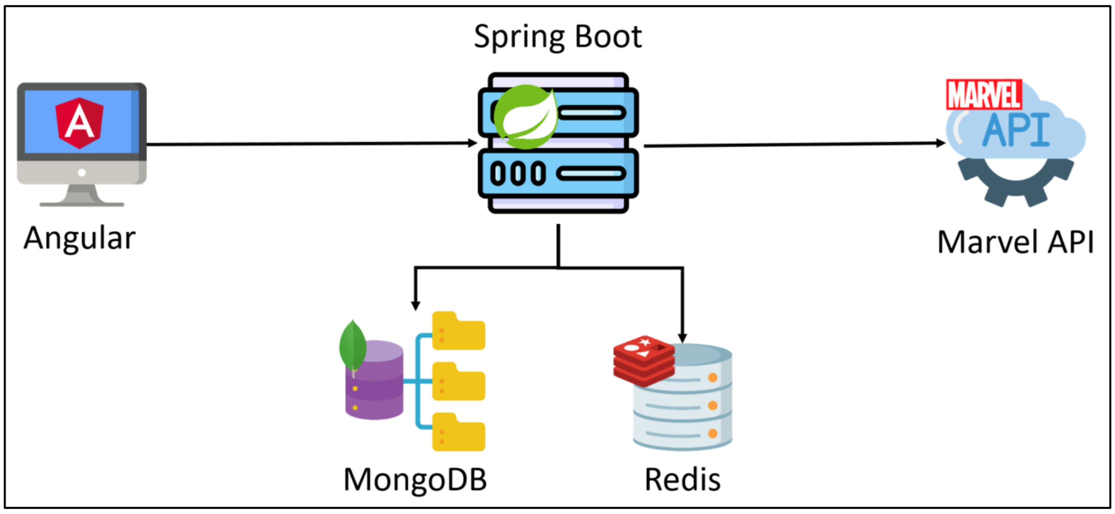
Objective

The objective of this workshop is to build a (semi) PWA application with Angular.

Setup

a. Generate a SpringBoot application. Add the following dependencies
   i. Spring Boot Dev Tools
   ii. Spring Web
   iii. Spring Data Redis
   iv. Jedis (3.9.x)
   v. JSON-P
b. Generate an Angular application
c. Create an account in Marvel Developer Portal (https://developer.marvel.com) and get your keys from https://developer.marvel.com/account

Workshop



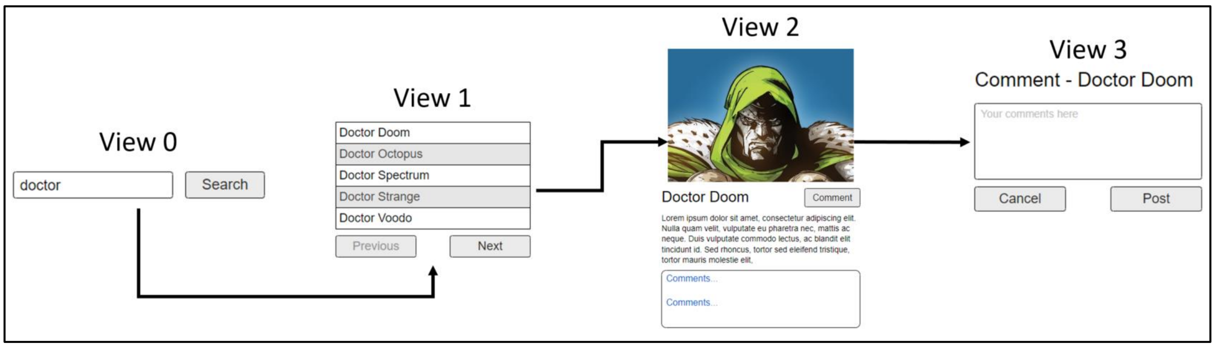
This application allows users to search for Marvel character and lookup information about them.

Angular Frontend

The Angular is a SPA with the following 3 views

1. View 0 - Search. This view allows a user to enter the starting name of their character that they wish to lookup
2. View 1 - Character list. This view displays the list of names that matches the search criteria. Uses can navigate the list by with the Next and Previous buttons
3. View 2 - Character details. This view displays the detailed information about a character.
4. View 3 - Comments. Post a comment regarding the character.

The following is a summary flow of the SPA


View 0
doctor
Search
View 1
Doctor Doom
Doctor Octopus
Doctor Spectrum
Doctor Strange
Doctor Voodo
Previous
Next
View 2
Doctor Doom
Comment
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit,
Comments...
Comments...
View 3
Comment - Doctor Doom
Your comments here
Cancel
Post

Angular makes HTTP request to the Spring Boot back end to populate View 1 and View 2.

Angular makes the following HTTP requests to populate the respective views

- View 1

```
GET /api/characters
Accept: application/json
```

- View 2

```
GET /api/character/<characterId>
Accept: application/json
```

- View 3

```
POST /api/character/<characterId>
Content-Type: application/json
Accept: application/json
```

You are free to determine what information you wish to display.

Spring Boot Backend

`GET /api/characters`

The backend uses `/v1/public/characters` from Marvel API to return a list of characters that matches the search criterial send by the Angular application from View 1.

`/v1/public/characters` (https://developer.marvel.com/docs#!/public/getCreatorCollection_get_0)

Use this endpoint to search for a list of Marvel characters. Use the `nameStartsWith` parameter. The `limit` and `offset` parameters defaults to 20 and 0 respectively.

When the search results return from Marvel endpoint, save the individual character details into the Redis database. Cache the result for 1 hour.

`GET /api/character/<characterId>`

To populate View 2, the backend first checks Redis to see if the result is available. If this is, then serve the result from the Redis look-aside cache.

If the requested character is not available, use the following endpoint `/v1/public/characters/<characterId>` (https://developer.marvel.com/docs#!/public/getCharacterIndividual_get_1) to retrieve the details from Marvel endpoint. When the result returns, cache the result in Redis for 1 hour.

All comments for the selected character should be retrieved from the Mongo database. You should only return the 10 most recent comments.

The character details and its corresponding comments are returned to the frontend.

`POST /api/character/<characterId>`

The comments for the Marvel characters are saved to the Mongo database. Add a timestamp to the comments.

Important: Marvel API requires request to be signed (authenticated). See https://developer.marvel.com/documentation/authorization

Deployment

You may deploy the application as a single or 2 separate applications.

Generate an PWA application manifest for the Angular application before deployment.

Optional Workshop

Only attempt this if you have completed the workshop.

Add a 'Share' button to View 2 of the frontend; this will allow your user to share his/her favourite character on social media, messaging platforms or email. The link should be View 2 of that character.

Use the Web Share API

See https://developer.mozilla.org/en-US/docs/Web/API/Navigator/share

The Share button should be disable if the browser does not support the Web Share API.