Solution Overview

This bookstore solution is built with Spring Boot and secure it with Spring Security and JWT.

We'll rely on the following technology stack to write the necessary code required to build and secure our Employee REST API:

- Spring Boot
- Spring Security
- H2 database
- Jjwt
- Maven

Security

The solution is security with Spring Security and JWT. There are 3 roles provided.

Role	Operation
MANAGER	Retrieve, Create, Update, Delete
ADMIN	Retrieve, Create, Update
GUEST	Retrieve information only

There are 3 preconfigure user is provided for testing purposes.

1) Manager
2) Admin
3) Guest

Rest API

In general, the API will expose the following endpoints:

`GET /api/find-all-books`	Get all Books available.
`GET /api/find-books`	Retrieve books by Book Title or Author Name.
`POST /api/create-book-record`	Create a new book record

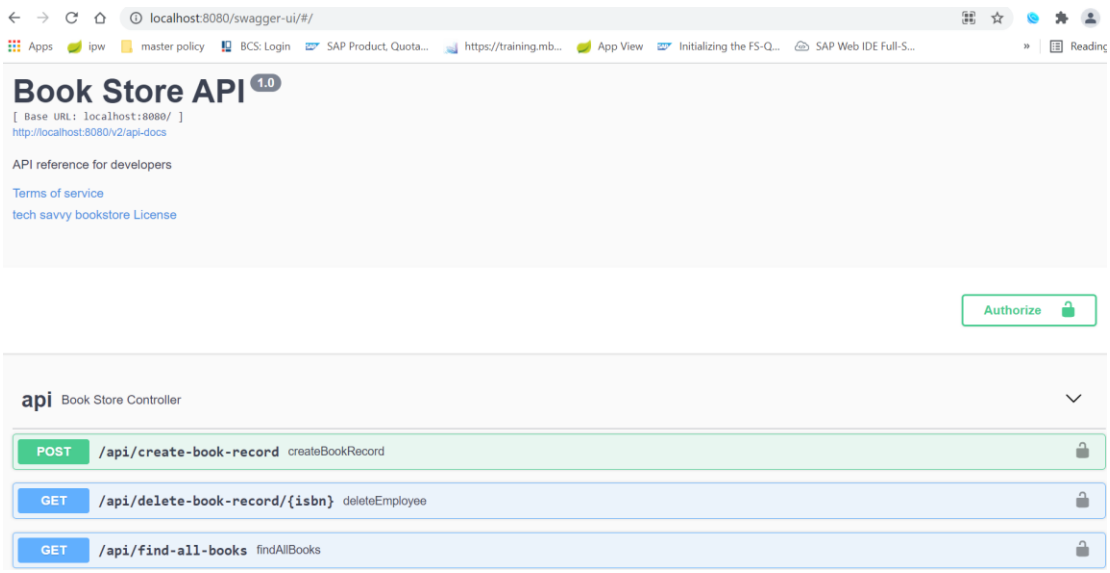
`POST /api/update-book-record`	Update book record base on isbn.
`GET /api/delete-book-record/{id}`	delete a particular book record based on isbn.
`POST /auth/signin`	Sign in using username and password which will return a JWT token to access API resources.

Starting the application.

1) Download the jar file **bookstore-0.0.1-SNAPSHOT.jar**
2) Open a command prompt and navigate to the folder which contains the jar file.
3) Issue the cmd **java -jar bookstore-0.0.1-SNAPSHOT.jar**
4) When you see this out put at console. The application has been started.

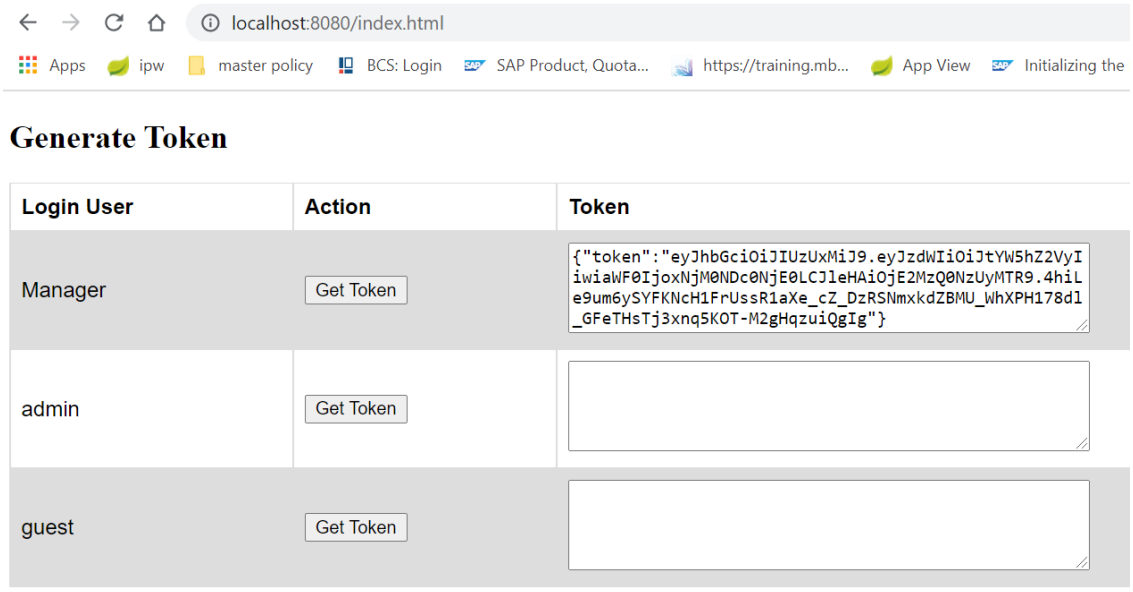
```
main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) with context path ''
main] tech.savvy.bookstore.Application         : Started Application in 11.192 seconds (JVM running for 11.732)
```

5) Open your browser and enter the url **http://localhost:8080/swagger-ui/**
6) You should be able to view the details of the api in the swagger ui.



7) Open your browser and enter the url http://localhost:8080/index.html

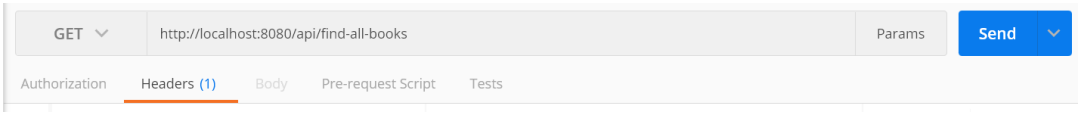
8) You will see a page which you can generate access token to use when testing the end point.



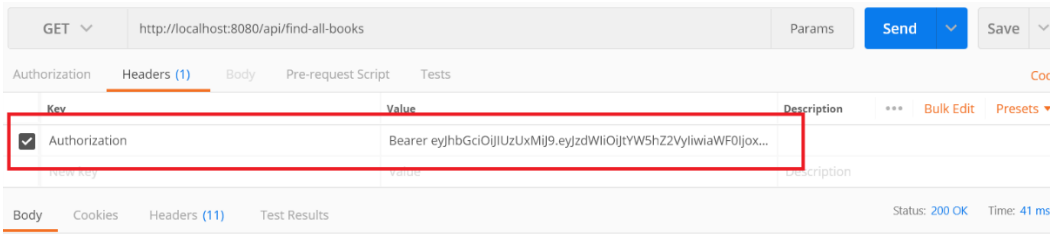
Testing the endpoint

You can test using curl or post man. It would be easier to use a post man for this purposes.

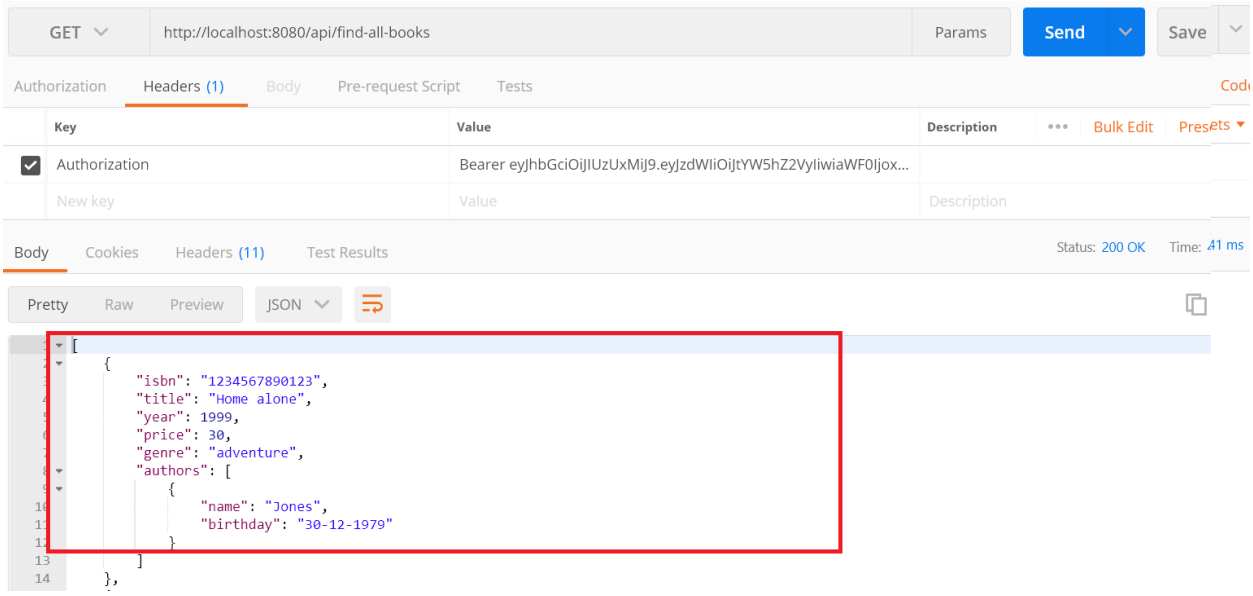
1) Open postman application.
2) Choose the action type and enter the endpoint. (for example)



3) Get an access token from a user in the index.html page (such as manager) and create a header with the token. (take note it has an expiry time of 10 min, a new token is required after 10 min)



4) Click on the send button and you should be able to see the result.



5) Similar step can be use for others endpoint. Take note of the parameter and body needed. The information can be found the swagger ui page.