Recipe Web Service

Recipe Web Service contains REST APIs in order to Create, Get, Update and Delete recipes from the database and render the requested details as JSON response to the end user. The response from REST APIs can be further integrated with the front end view for better presentation using Java and Javascript.

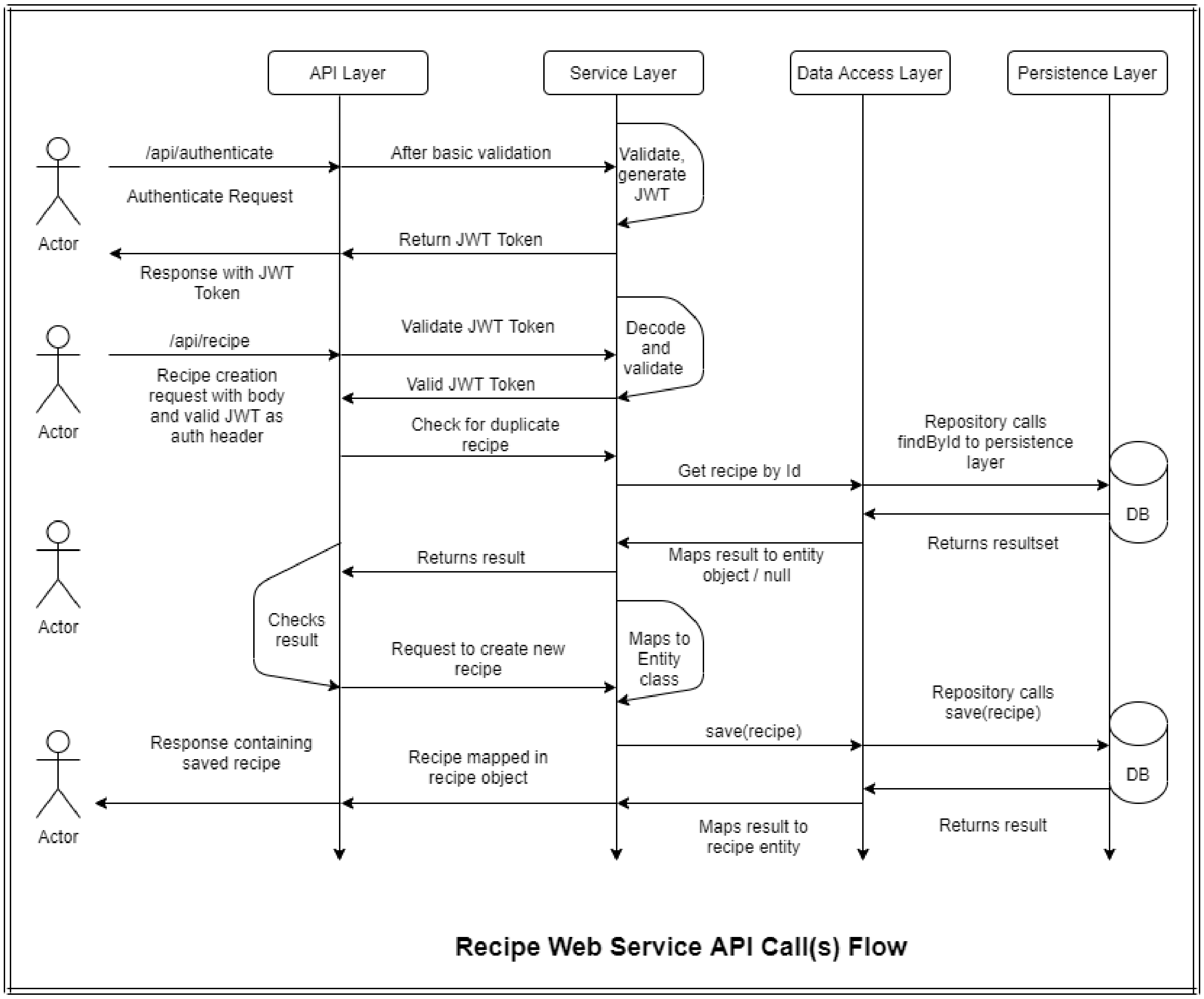
System Design

Recipe Web Service is microservice based layered architectured RESTful Web Service. This service can be deployed independently on premise / cloud and can also be containerized to execute as docker containers. There are 4 layers from top to bottom:

- API Layer
    - Top layer, which is main interface available for integration and interaction with front-end or end user to consume APIs
    - Contains secure API endpoints implementation
    - Springboot-starter-security Module along with JWT is used to implement authentication for APIs
    - Springboot-starter-web module used as a framework to implement ReSTful api endpoints
- Service Layer
    - This layer sits in between API layer and Data access layer with some utility functionality
    - Mainly responsible for interacting with Data Access Layer and transferring the recipes data as required by top and below layers
    - It's just another module added to decouple business logic of recipes data transfer and mapping from/to API layer
    - Further, service layer can be enhanced to support advanced features like Caching, Interacting with external Authorization Service etc
- Data Access Layer
    - Responsible to provide Object Relationship Mapping (ORM) between higher level recipe Java objects and persistence layer tables
    - Springboot-starter-data-JPA module is used to implement mappings between objects and tables

    - This layer contains recipe entity classes and JPA repositories which implement lower level functionality of storing/retrieving recipes data
- Persistence Layer
    - Bottom most layer, responsible for physically storing the recipes data into database table
    - Just one physical table - `recipes` is used to store the recipes data for the service
    - [MySQL]((https://www.mysql.com/) is configured to be used as database service
    - For development and testing purposes, the Embedded H2 Database provided by Spring Boot framework is also utilized.

Architectural Diagram (Service Calls)



Supported Features

- ReSTful API : Springboot
- API Authentication: Spring Security with JWT Token
- Object Relationship Mapping: Spring Data JPA
- Exception Handling: Controller Advice and ExceptionHandler
- Logging: SLF4J Logger
- Unit Tests: Junit 5 with AssertJ