**WAES Scalable Web**

# **FUNCTIONAL SPECIFICATION DESIGN**

Carlos Camarena

6/3/18

Contents

[**FUNCTIONAL SPECIFICATION DESIGN** 1](#_Toc515801092)

[OVERVIEW 3](#_Toc515801093)

[ASSUMPTIONS, RISKS, ISSUES AND DEPENDENCIES 3](#_Toc515801094)

[TESTING 3](#_Toc515801095)

[USER INTERFACE 3](#_Toc515801096)

[ARCHITECTURE 4](#_Toc515801097)

[USE CASE / stories DIAGRAM 5](#_Toc515801098)

[USER STORIES 5](#_Toc515801099)

[CLASS DIAGRAM 5](#_Toc515801100)

[DATABASE 6](#_Toc515801101)

[APPLICATION SERVER 6](#_Toc515801102)

**OVERVIEW**

The goal of this assignment is to show your coding skills and what you value in software engineering. We value new ideas so next to the original requirement feel free to improve/add/extend.

Here we will provide the Functional Specification Design solution to meet the requirements specified in the Requirements document.

## **ASSUMPTIONS, RISKS, ISSUES AND DEPENDENCIES**

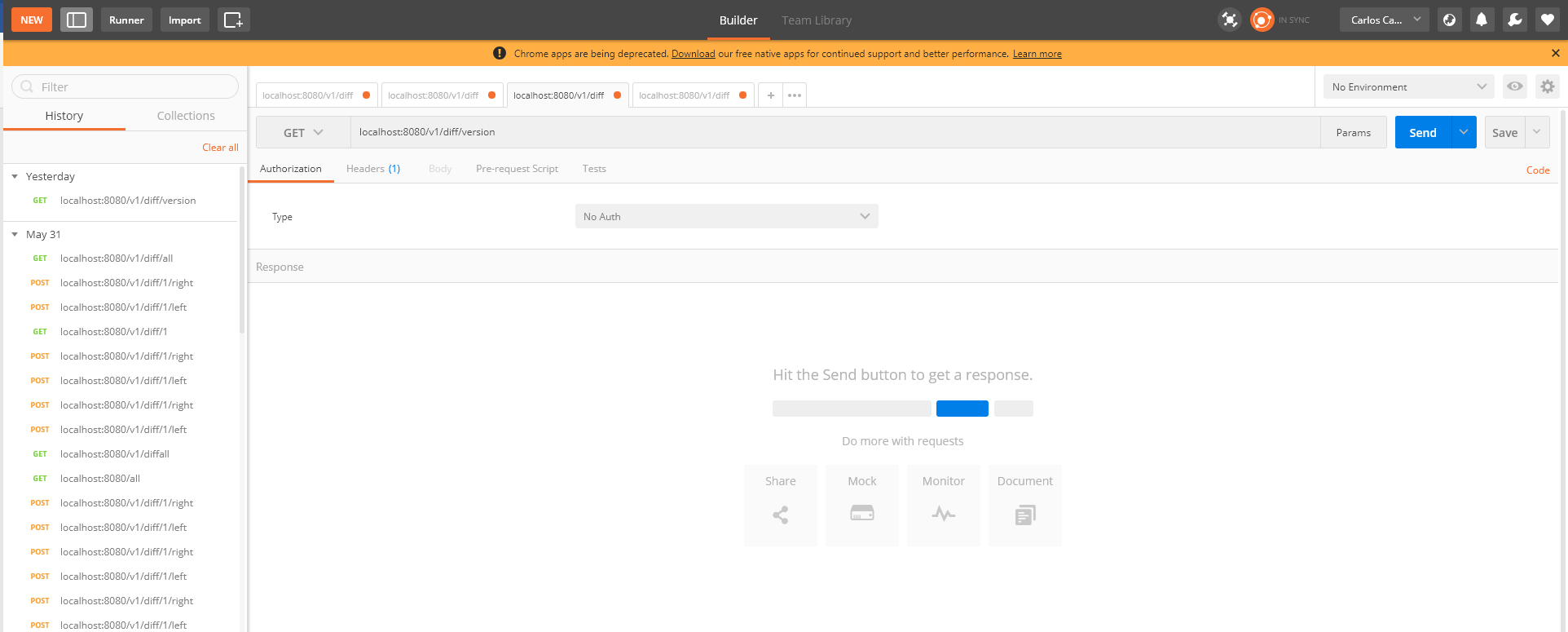
1. Encoding is done with Base64 UTF-8 standard.
2. Default port Tomcat 8080 is available.
3. Browser works over http standard.
4. Java 8 is being used.

## **TESTING**

1. Unit Test
   1. Junit
   2. White box testing
2. Functional Test
   1. Manual testing
   2. Black box testing
3. Integration Test
   1. Junit
   2. Black box testing
4. Automation Test (nice to have)
   1. Cucumber
   2. Behavioral Testing

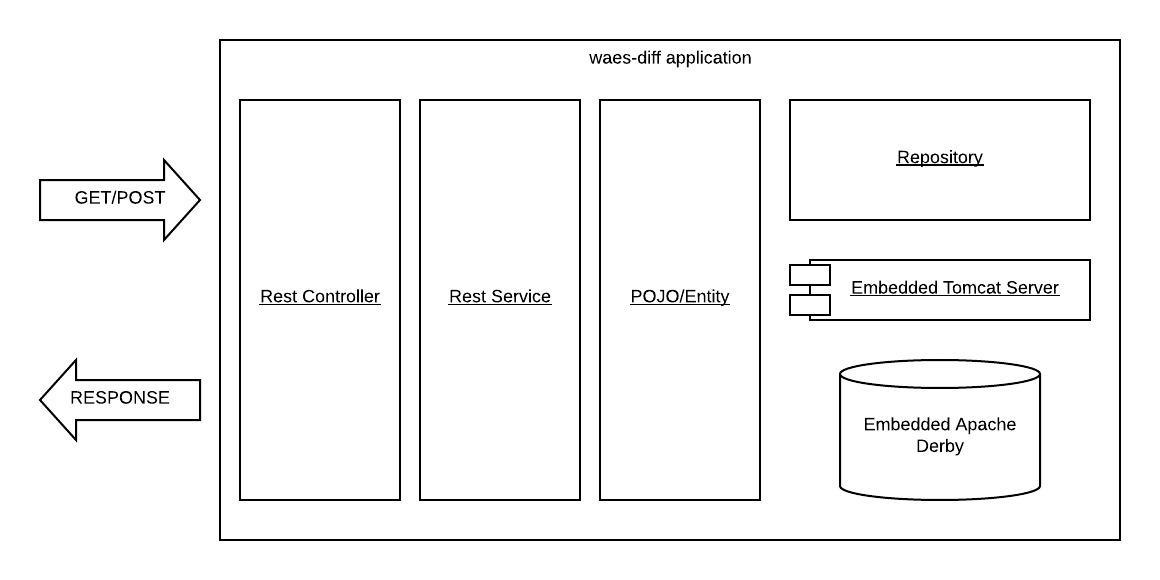
## **USER INTERFACE**

The application doesn’t provide a user interface. It is a web-based application who exposes endpoints to be consumed by another application. To be tested an external tool (or consumer application) is needed. Postman is being used during the development of the application and is the one recommended in order to test.

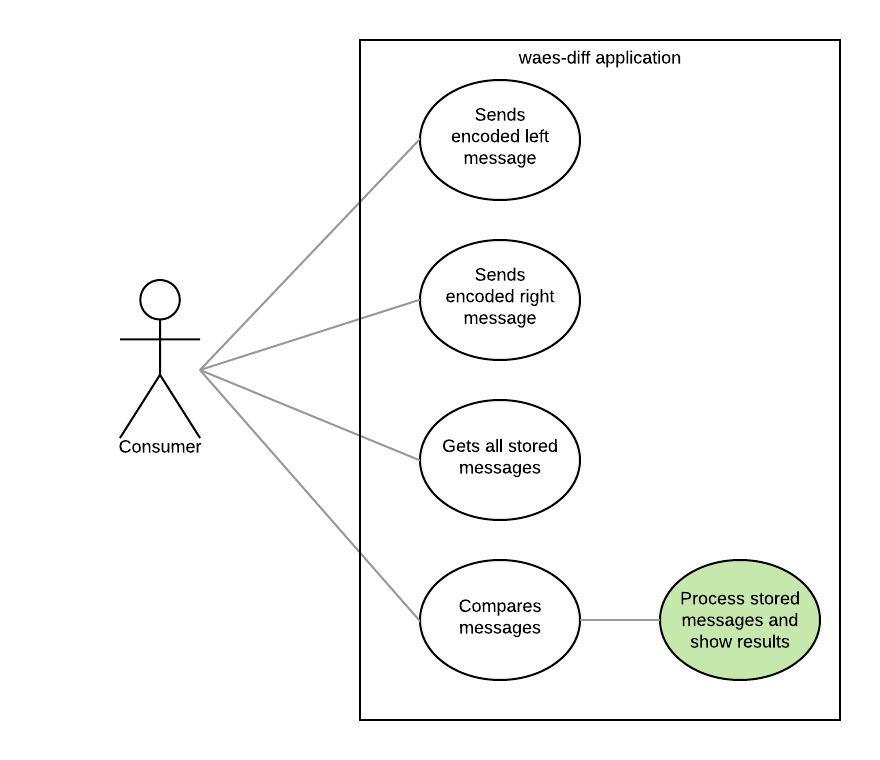


## **ARCHITECTURE**

Spring Boot Architecture



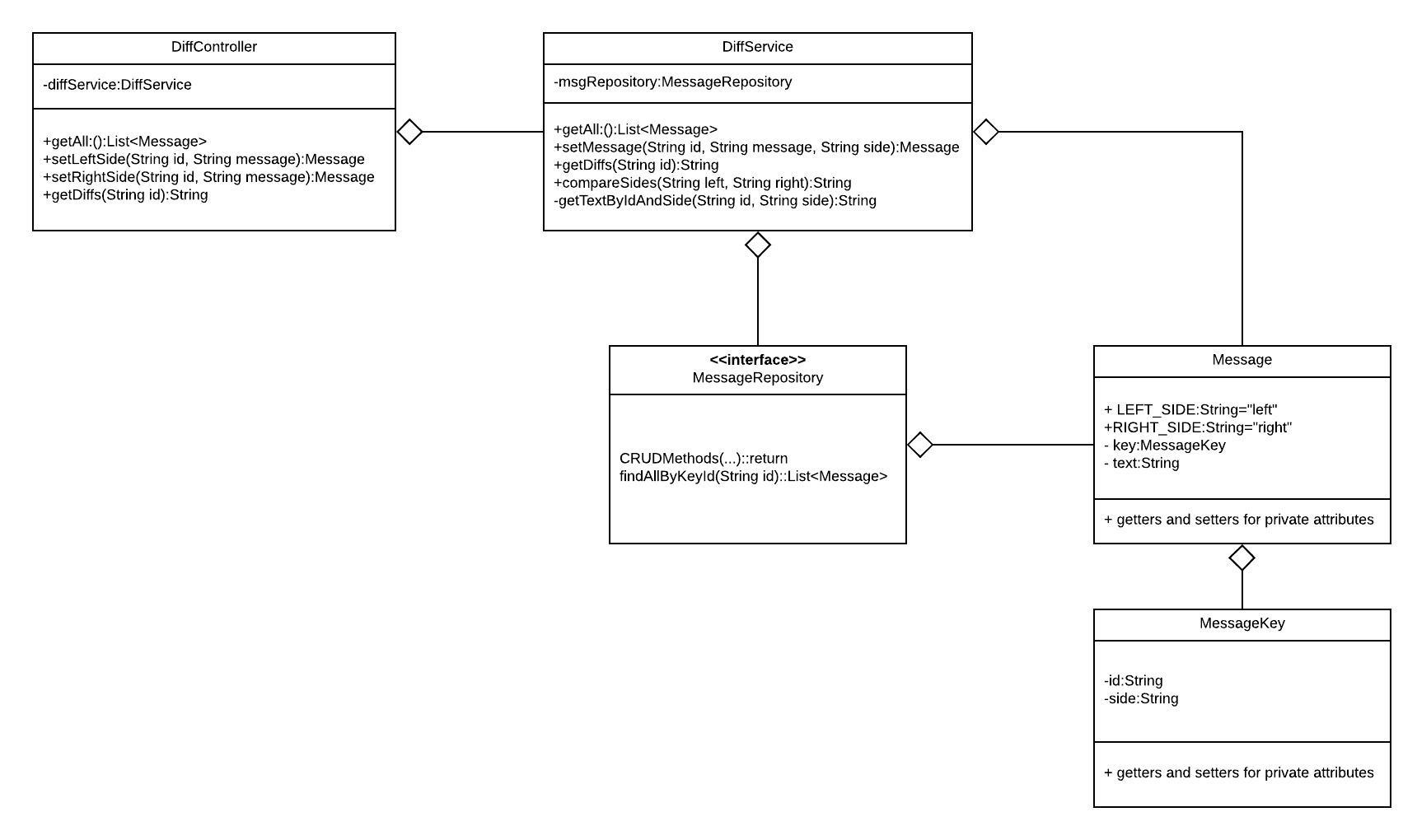
## **USE CASE / stories DIAGRAM**



**USER STORIES**

1. As a consumer I want to be able to send an encoded JSON message to the application so that it will be stored and compared to a second message with same characteristics.
2. As a consumer I want to be able to send a second encoded JSON message to the application so that it will be stored and compared to the first message
3. As a consumer I want to be able to request to the application to compare my 2 encoded JSON message and get a response with differences.
4. As a consumer I want to be able to request to the application to fetch all the messages of ta given ID in at any given moment.

**CLASS DIAGRAM**



**DATABASE**

The DB to use is Embedded Apache Derby, implemented through JPA/Hibernate ORM and CRUD Interface. The only extra method custom made is “findAllByKeyId(String id)”.

Spring Boot framework and ORM will take care of the tables specification.

**APPLICATION SERVER**

The application runs over Embedded Tomcat Server. Deployment and start/stop is managed by Spring Boot framework.