horizontal line

Assignment - 1

**11th May 2020**

# OVERVIEW

The first assignment was to create a simple Hello-World Spring boot project. This task was assigned to every team member separately.

# GOALS

1. Create a Hello-World Spring boot project.

# TECHNOLOGIES USED

1. Java
2. Spring Boot v 2.2.7
3. Spring Web Dependency

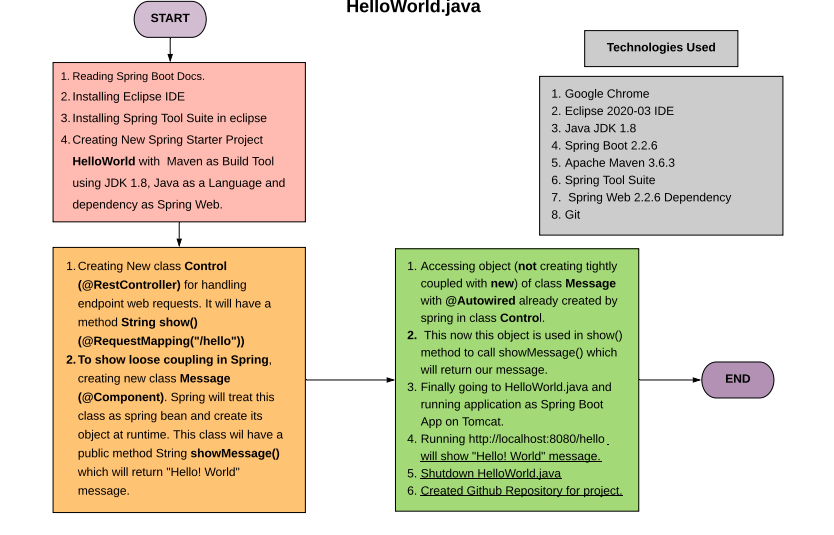
# DESCRIPTION

A basic Spring boot hello - world project was created using Spring Initialiser / Spring tools suite(STS) in Eclipse IDE. The project included a spring web dependency that provided an embedded tomcat server.



It had the following components

* Main Application Class
* Rest Controller ( with @GetMapping and @RequestParams)



Assignment - 2

**13th May 2020**

# OVERVIEW

The task was to create a Micro-service that supports CRUD operations for an entity using GET and POST requests with H2 in-memory database.

This task was assigned to every team member separately.

# GOALS

1. Implement CRUD operations on an entity using GET(all and by Id) and POST
2. Add H2 in-memory database

# TECHNOLOGIES USED

1. Java
2. Spring boot v 2.2.7
3. Spring Web
4. JPA
5. H2

# DESCRIPTION

A basic spring boot project was created using STS in eclipse IDE. Following dependencies were added : JPA, H2, Spring Web.

The microservice consists of these components :

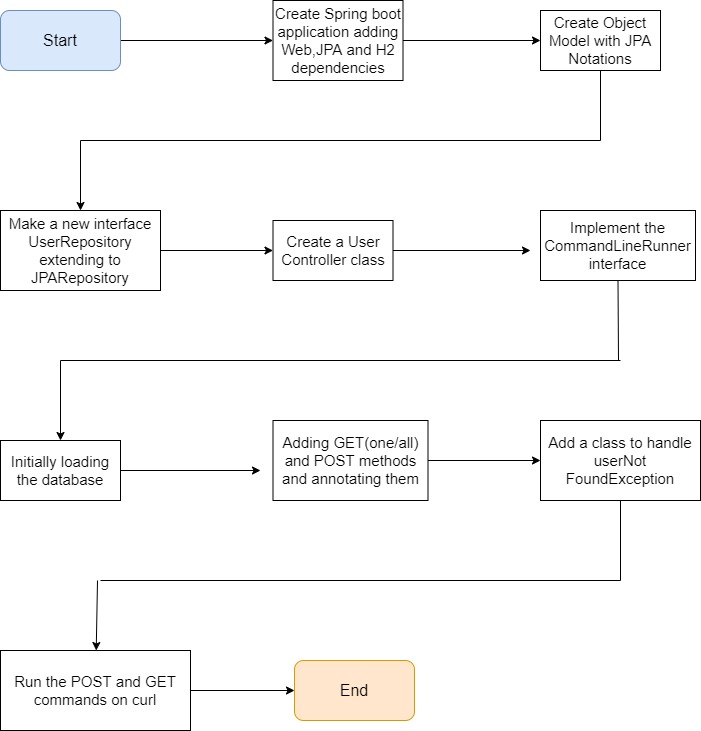
* Model
* Controller
* Repository
* Exception handler

The domain class is annotated with @Entity annotation to convert it into domain model.

An interface is created that extends JpaRepository interface providing all the CRUD functionality.

The controller is annotated with @RestController and has the @GetMapping and @PostMapping annotations to the respective functions for im[plementing GET and POST requests.

The controller also implements command line interface for pre-populating the H2 in-memory database during startup of the application.

UserNotFound exception is handled by exception class. 

The requests have been called through curl/Postman.

Assignment - 3

**15th May 2020**

# OVERVIEW

This task included adding external database connectivity to the microservice created in the previous assignment.

This task was also assigned to every team member separately.

# GOALS

1. Add Postgres database connectivity to the Microservice created before.

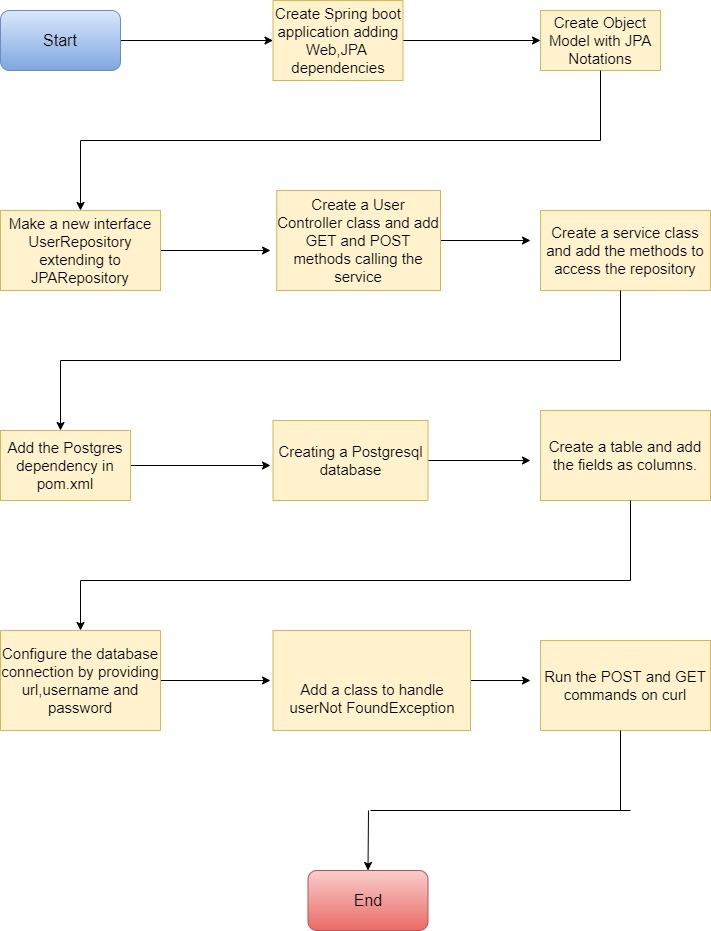
# TECHNOLOGIES USED

1. Java
2. Spring boot v 2.2.7
3. Spring Web
4. JPA
5. Postgresql

# DESCRIPTION

An external Postgres database is constructed with the table and its fields corresponding to the entity model created before.

The postgres database is connected to the spring boot by configuring the username and password and connection url in the application.properties of the project.



Assignment - 4

**18th May 2020**

# OVERVIEW

A similar microservice was created in this task but this time also involving PUT and DELETE mapping.Also all the operations were performed on a Customer entity.

From this assignment onwards all tasks were allotted as team tasks.

# GOALS

1. Create the microservice supporting CRUD operations on Customer entity using GET,POST,PUT and DELETE
2. Handle all the edge cases by implementing Error and Exception handling.

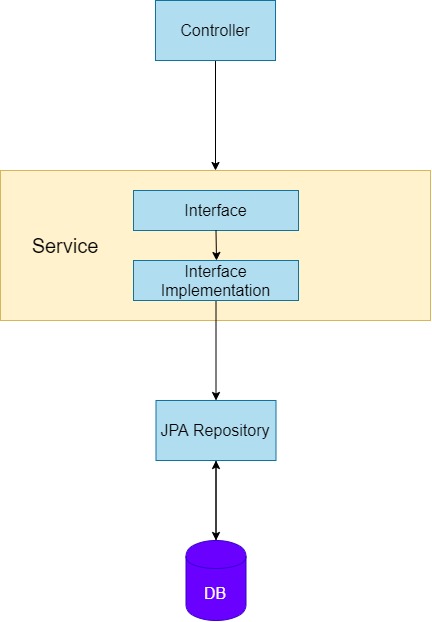
# TECHNOLOGIES USED

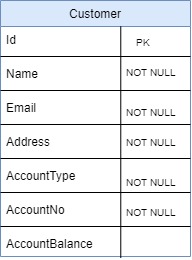
1. Java
2. Spring boot v 2.2.7
3. Spring Web
4. JPA
5. H2

# DESCRIPTION

The basic components of the backend are:

1. Model
2. Controller
3. Service
4. Repository
5. CryptoConverter
6. Exception Handler
7. Error Handler



A customer entity model was created whose attributes were as follows :

* Id
* Name
* Email
* Address
* AccountType
* AccountNo
* AccountBalance

A CustomersController was created that contained all the mappings including @GetMapping, @PostMapping, @PutMapping, @DeleteMapping.

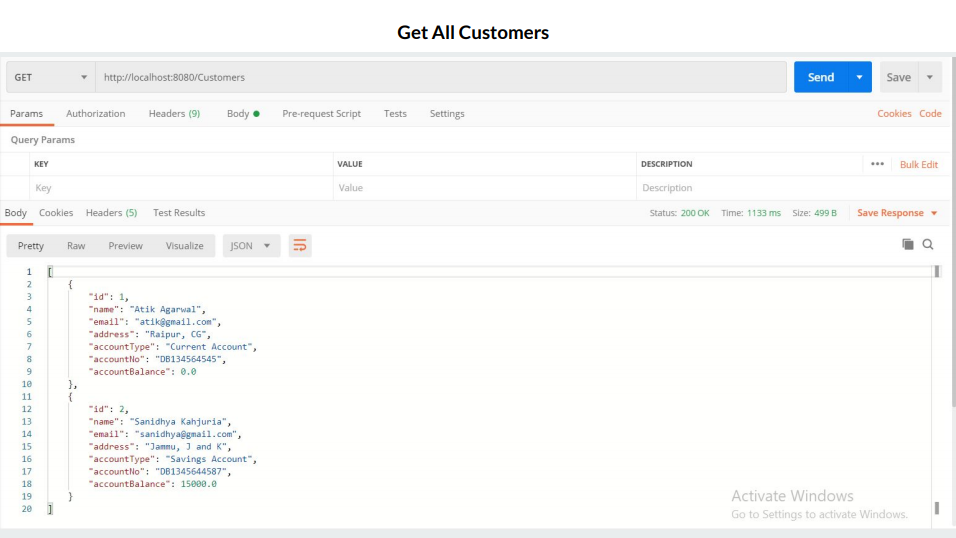
A customerRepository interface was created by extending the JPARepository interface.

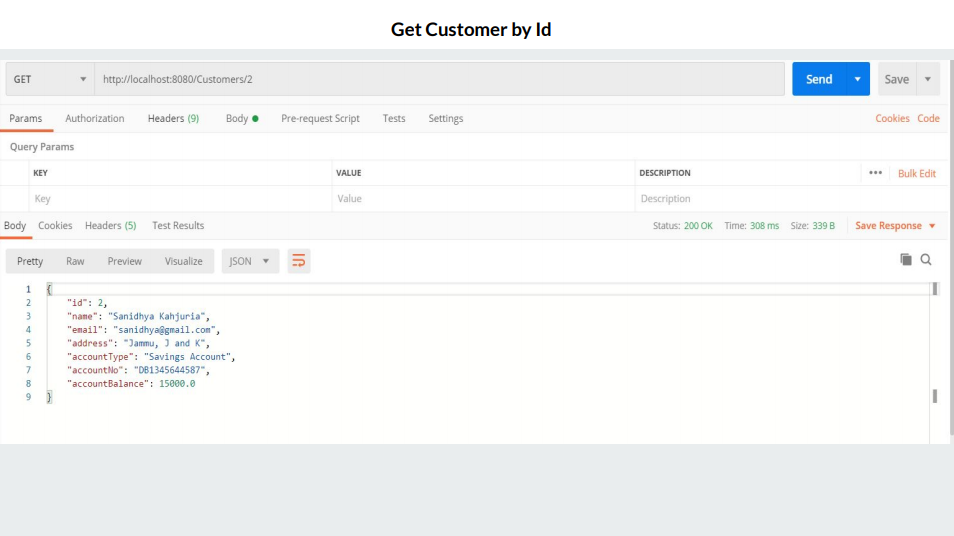
* getAllCustomers()
* getCustomerById()
* createNewCustomer()
* updateCustomerById()
* deleteCustomerById()

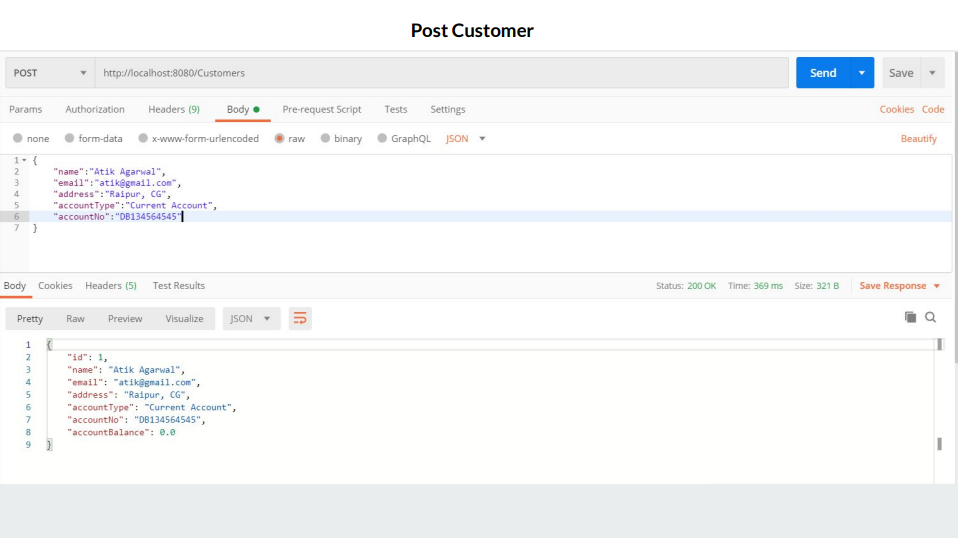
The accountNo attribute has been encrypted in the database by an Encryptor class that implements JPA Attribute Converter and Uses Cipher class for encryption.

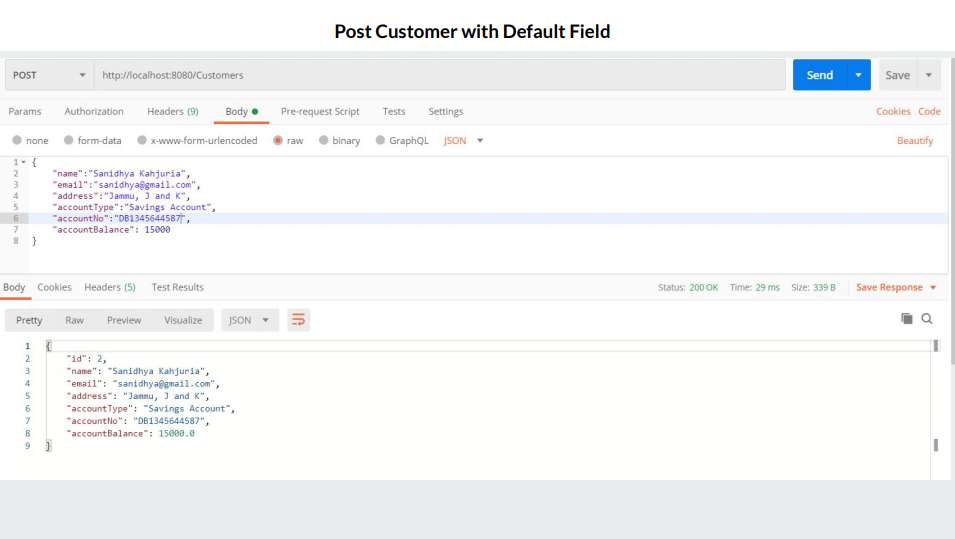
The edge cases have been handled by The ResourceNotFoundException class that handles the exception if the required resource is not found. Error Handler has also been implemented.

The requests and responses of the Rest API are monitored through Postman:

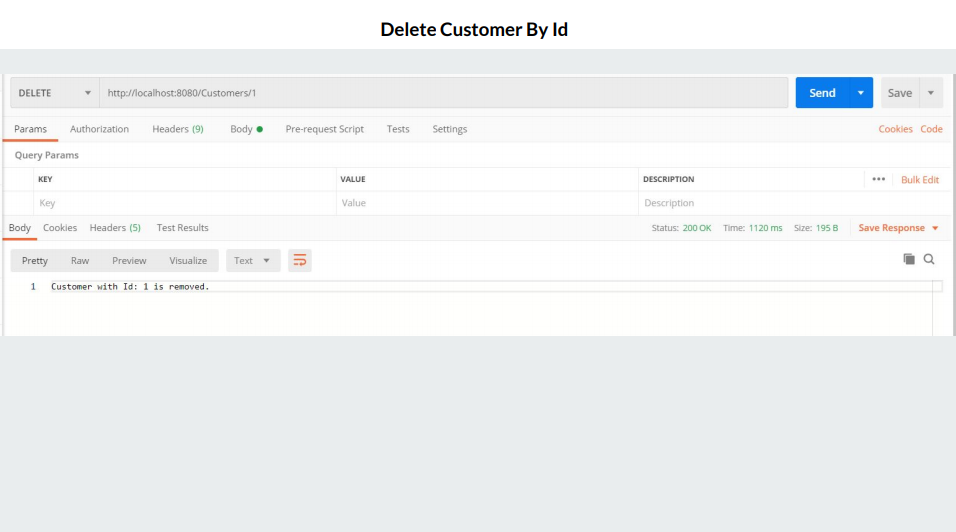












Assignment - 5

**20th May 2020**

# OVERVIEW

This task involved creating a basic react app using create-react-app

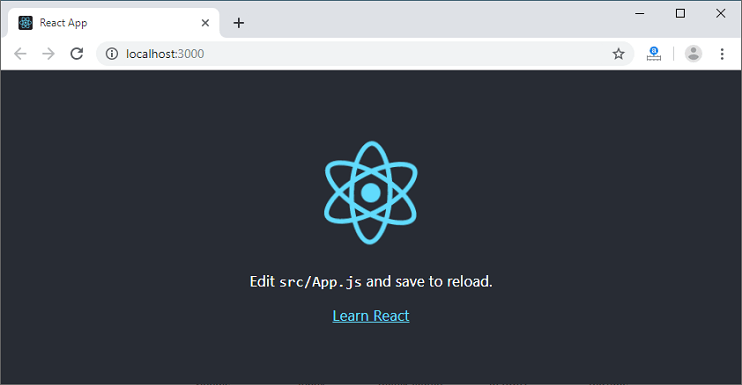
# GOALS

1. Create a basic react app and learn react basics.

# TECHNOLOGIES USED

1. React

# DESCRIPTION



Assignment - 6

**21th May 2020**

# OVERVIEW

This task involved creating a UI interface for the backend that was created before.

# GOALS

1. Create a login page
2. Create a dashboard page
3. Create a add/update user page
4. Add form validation in add user page
5. Establish Integration of all the pages
6. Enable routing of the HTTP requests to integrate with backend

# TECHNOLOGIES USED

1. React
2. React-router
3. Bootstrap

# DESCRIPTION

UI is mainly divided into these these screens :

* Landing page
* Login page
* Home page (Customers list)
* Delete popup
* Add customer page
* Customer details page
* Edit Customer page

