

News Manager System

40225526 – Jimi Mehta
Report

Version 1.0

Github Link: <https://github.com/jimimehta/NewsManager-App-Project>

Table of Contents

1.	Introduction	3
1.1	Purpose	3
1.2	Scope	3
1.3	Definitions, Acronyms, and Abbreviations	3
1.4	References	3
1.5	Overview	3
2.	Tools and Technologies Used.....	4
2.1	Server-side tools and technologies used.....	4
2.2	Client-side tools and technologies used.....	4
3.	Use-Case View	5
3.1	Architecturally significant use cases	5
4.	Data View	6
5.	Object Relational Structural Patterns	6
6.	Design Pattern	7
7.	Object Relational Structural Patterns	8
8.	Spring Boot React Full-Stack Architecture	9
9.	Testing Tools	10

1. Introduction

1.1 Purpose

This document provides an overview of the architecture, testing and functionalities of the News Manager system. It embodies the significant decisions that were made concerning the architecture of the application.

1.2 Scope

This Software Architecture Document gives a global view of the architecture of the News Manager System. This system is developed by one student registered in the SOEN6441 Advance Programming Practices.

The News Manager provides users with the ability to manage and categorize their News, track their News, and create News articles. All those features are available to the users wherever they can find a computer with an Internet connection.

1.3 Definitions, Acronyms, and Abbreviations

HTML = Hypertext Markup Language.

CSS = Cascading Style Sheet

JS = Javascript

JPA= Java Persistence API

UI = User Interface.

1.4 References

"SOEN 6481, Software Architecture" Course Page

[Website] Accessible:

<https://users.encs.concordia.ca/~cc/soen6441//>.

1.5 Overview

In compliance with the specified purpose and scope, the rest of this document will deal with the following topics:

2. Tools and Technologies Used

2.1 Server-side tools and technologies used

- Spring Boot 2 +
- SpringData JPA (Hibernate)
- Maven 3.2 +
- JDK 1.8
- Embedded Tomcat 8.5+
- Eclipse 2021-12
- Microsoft SQL Database

2.2 Client-side tools and technologies used

- React
- Modern JavaScript (ES6)
- NodeJS and NPM
- VS Code IDE
- Create React App CLI
- Bootstrap 4.5 and Axios HTTP Library

3. Use-Case View

This Use Case view will be used to present the architecturally significant use cases and identify the set of scenarios and use cases that represent important and fundamental functionalities of the system.

3.1 Architecturally significant use cases

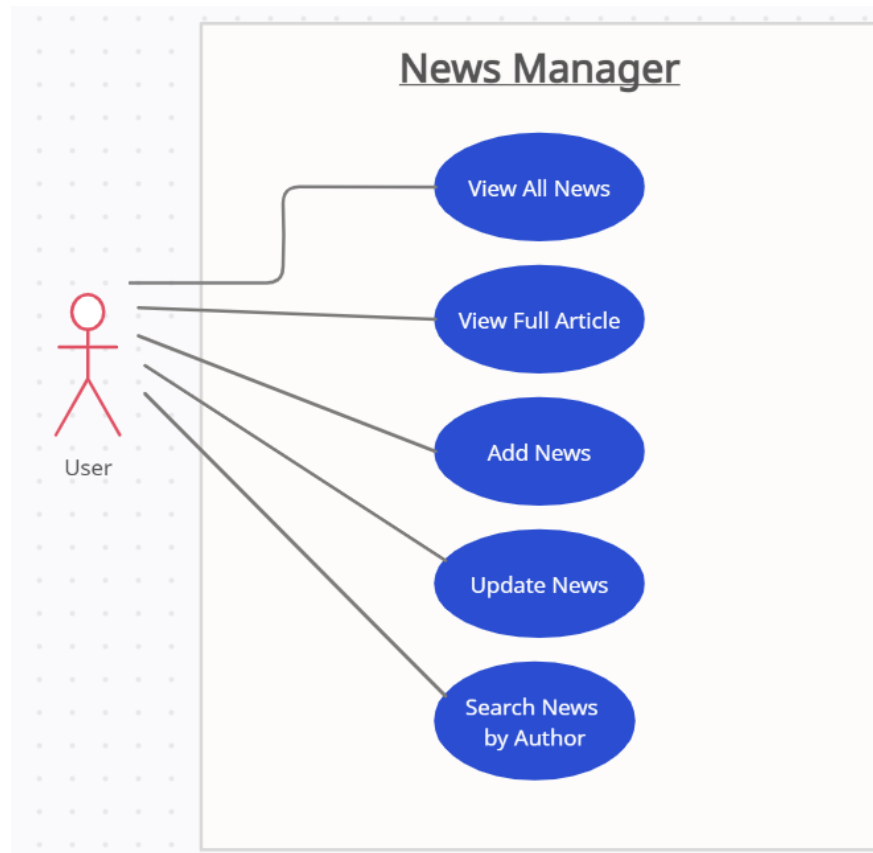


Figure 1. Architecturally Significant Use Case Diagram

4. Data View

The database provides persistence for all data input into the News Manager. The schema in which the relevant data is represented is as illustrated in the Database diagram below.

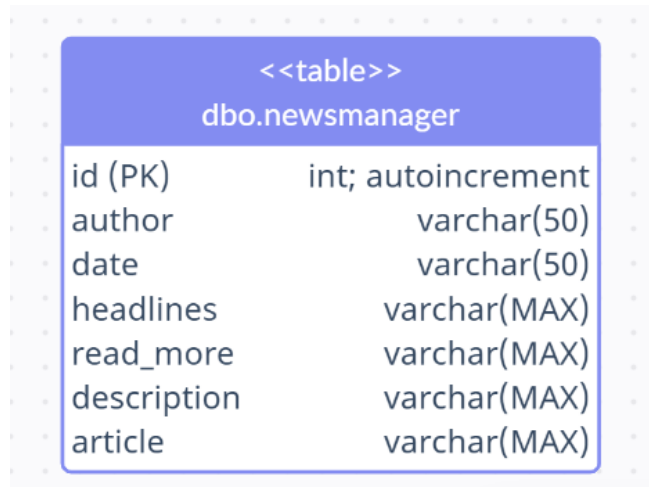


Figure 2. Diagram for the News Manager's database

5. Object Relational Structural Patterns

Single-table inheritance strategy: Map the entire class hierarchy to a single table ("filtered mapping").

The database provides persistence for all data input into the News Manager which is mapped using Single-table inheritance strategy. ORM of News Manager is as illustrated in the diagram below.

Table-per-class inheritance mapping (Vertical mapping)

A table-per-class inheritance strategy maps each class to its own table and maps each attribute to a column in the mapped table.

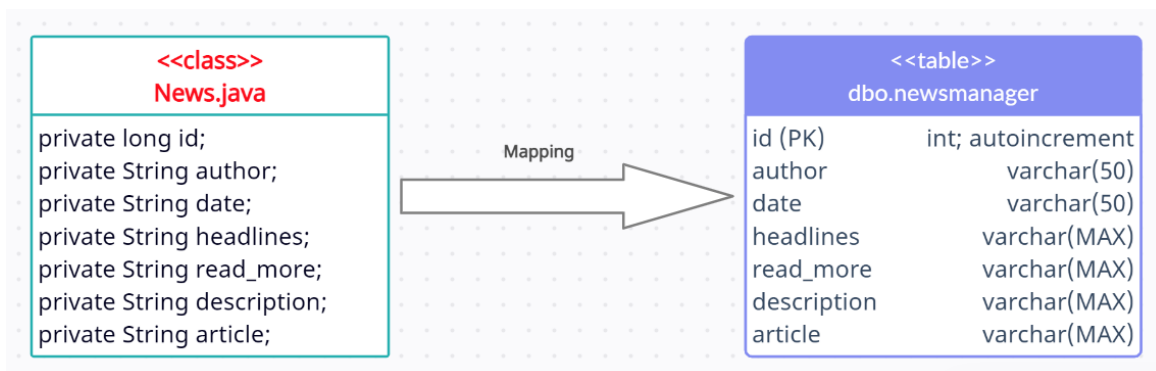


Figure 3. Diagram for the News Manager's ORM

6. Design Pattern

In News Manager, React is used to handle the UI — frontend, and Spring Boot Java is used to handle the data — backend. And to connect the two of them, we need REST API.

Now it becomes **Model-View-Controller(MVC)**

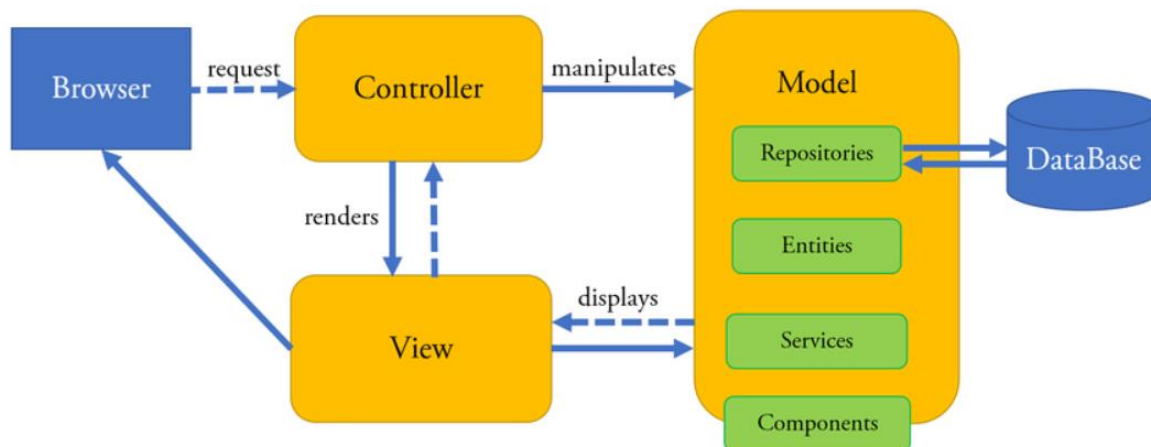


Figure 4. Diagram for the News Manager's DESIGN PATTERN

In News Manager, repository pattern have two purposes; first it is an abstraction of the data layer and second it is a way of centralising the handling of the domain objects.

Now it becomes **Repository Design Pattern.**

The idea with this pattern is to have a generic abstract way for the app to work with the data layer without being bother with if the implementation is towards a local database or towards an online API.

The methods are based on the CRUD methods; Create, Read, Update and Delete.

It can be easily looked at from the diagram 5.

7. Object Relational Structural Patterns

In News Manager, Presentation, Domain and Data Layer are divided as per the diagram. It has data mapping as well as TDG for their operation.

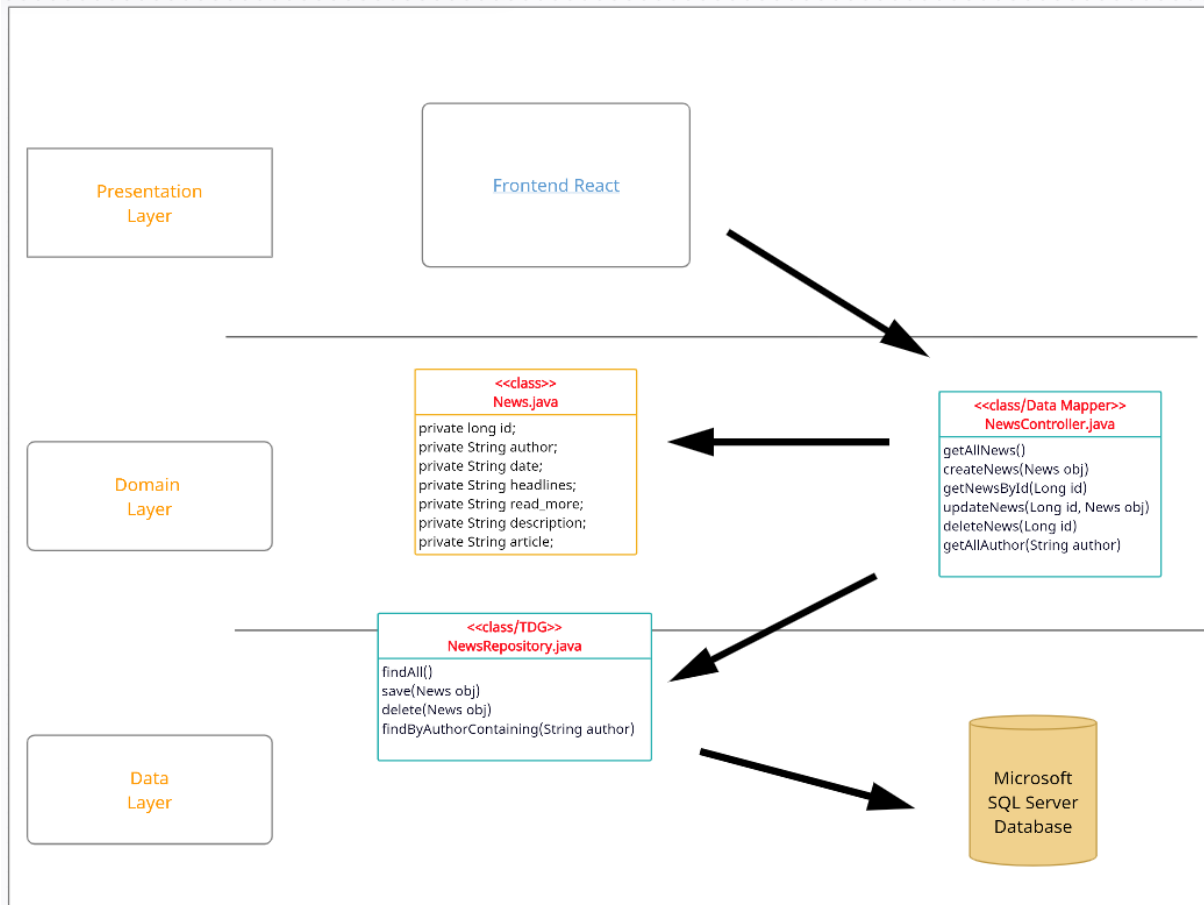


Figure 5. Diagram for the News Manager's Object Relational Structure Pattern

8. Spring Boot React Full-Stack Architecture

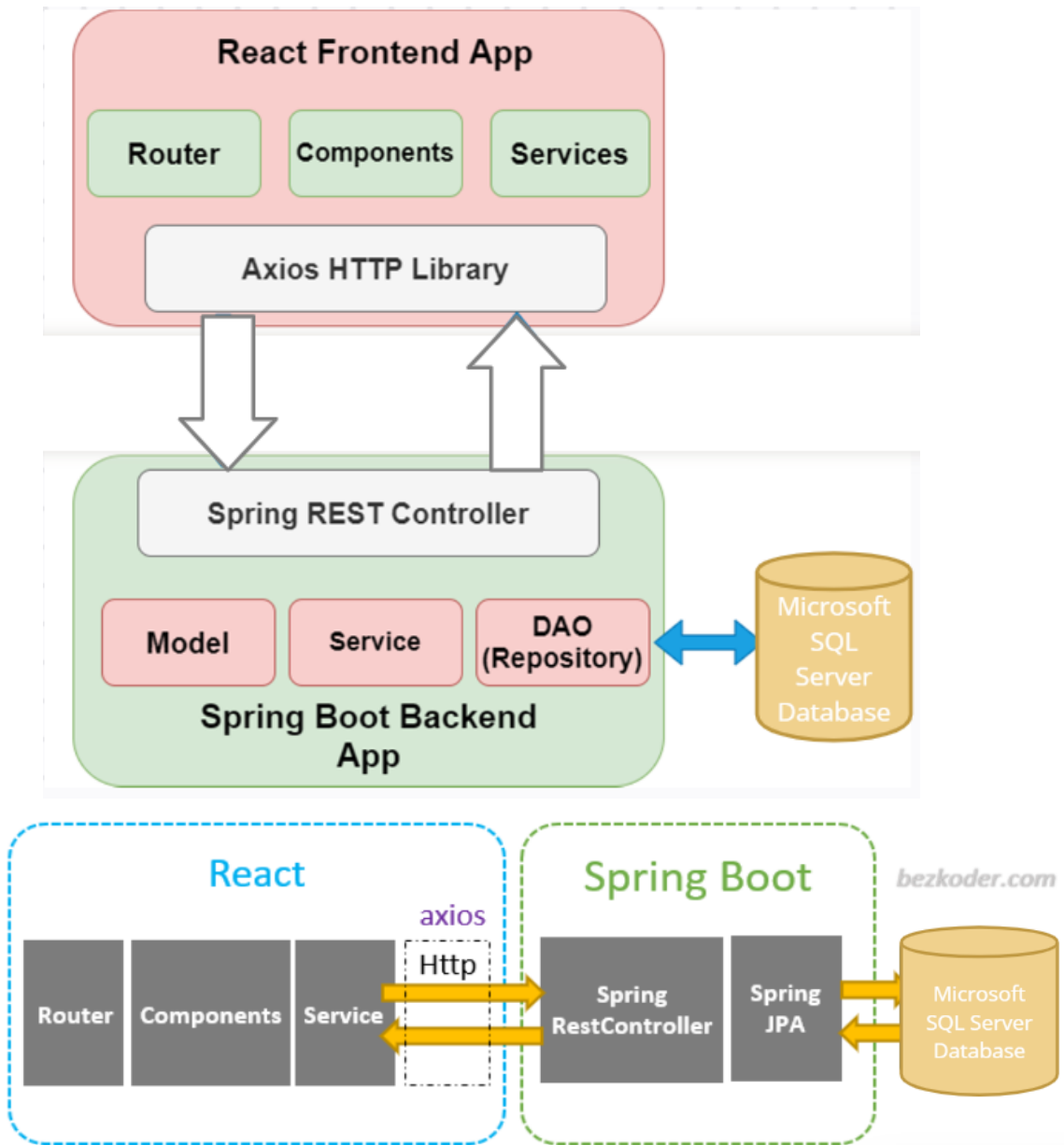
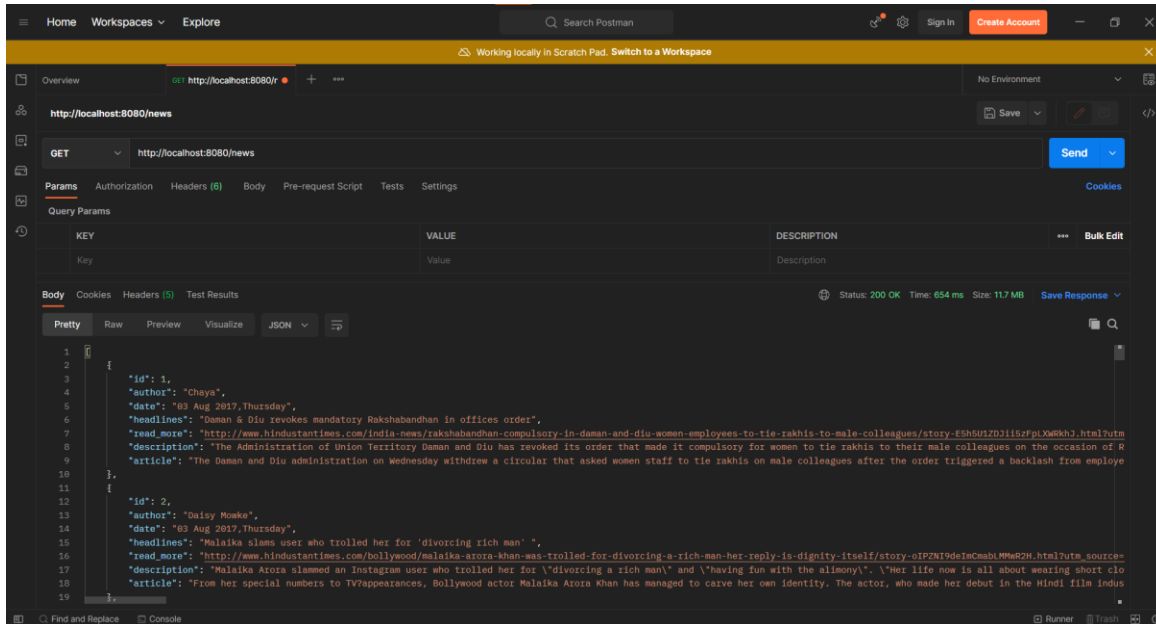


Figure 5. Diagram for the News Manager's Full Stack Architecture

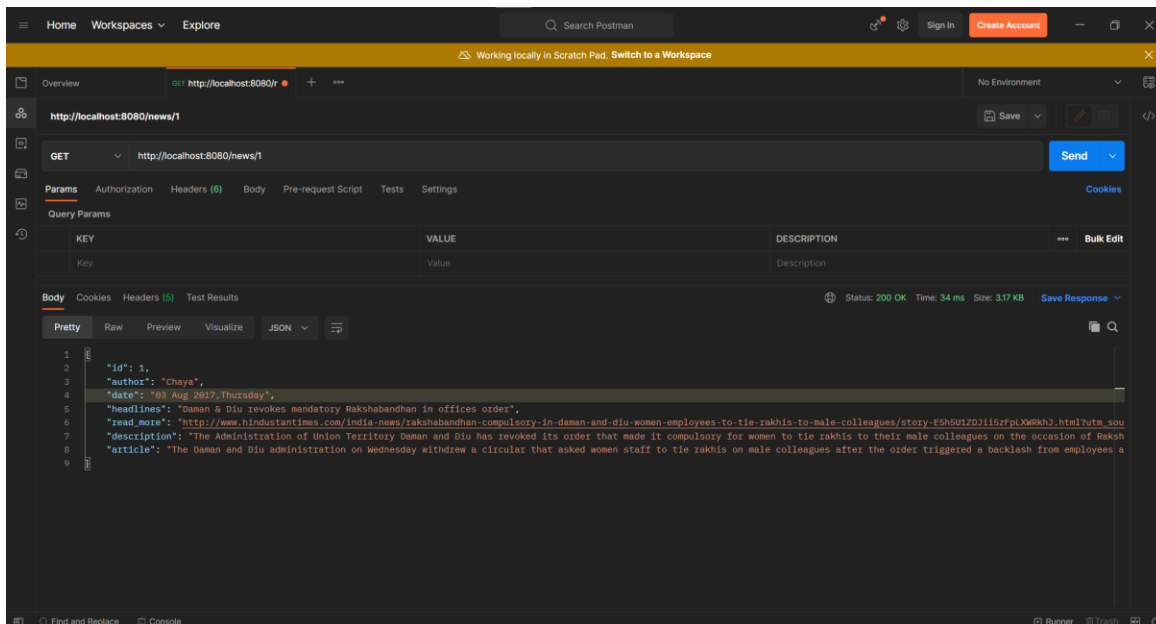
9. Testing Tools

Postman is an API platform for developers to design, build, test and iterate their APIs. Below Screenshots of API TESTING in POSTMAN are illustrated.

Test 1:



Test 2:



Test 3:

The image displays two screenshots of the Postman application interface, showing the results of two different HTTP requests to the endpoint `http://localhost:8080/news/1001`.

Top Screenshot (GET Request):

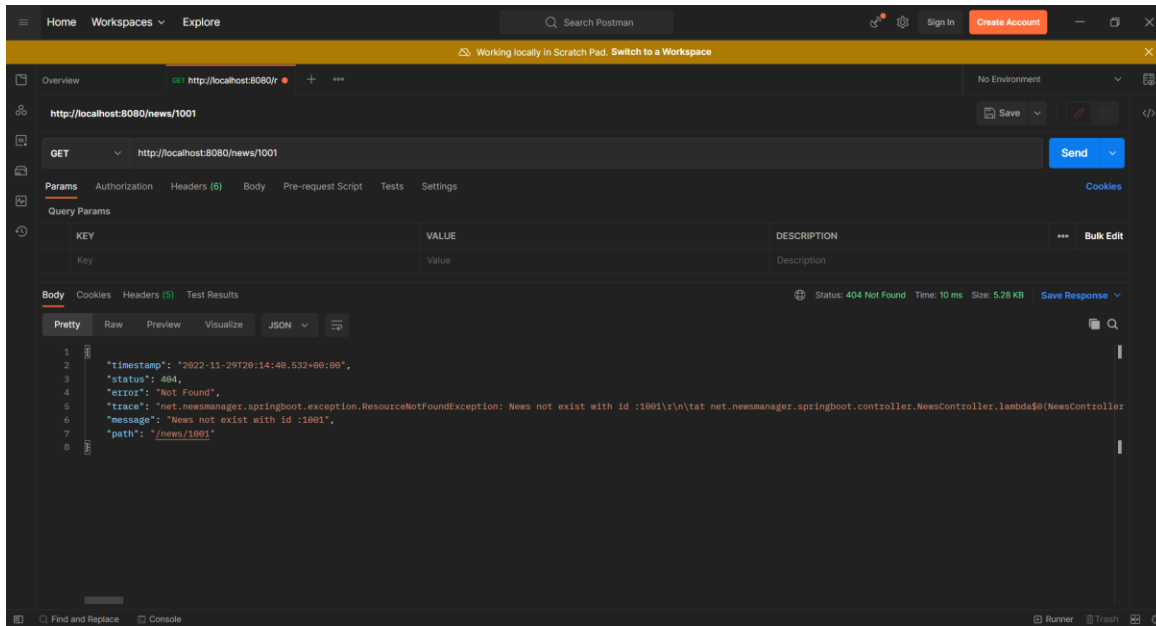
- Method:** GET
- URL:** `http://localhost:8080/news/1001`
- Status:** 200 OK
- Time:** 11 ms
- Size:** 1.55 KB
- Body (Pretty):**

```
{
  "id": 1001,
  "author": "Arshiya Chopra",
  "date": "21 Jun 2017,Wednesday",
  "headlines": "Supreme Court denies bail to former HC judge CS Karnan",
  "read_more": "http://indiatoday.intoday.in/story/supreme-court-denies-bail-to-hc-karnan/1/903697.html",
  "description": "The Supreme Court on wednesday denied interim bail to former High Court judge CS Karnan, who was sentenced to six-month jail term for contempt of court. The vacation bench",
  "article": "The Supreme Court today denied interim bail for controversial former judge CS Karnan. Karnan, who retired in disgrace as a Calcutta High Court judge, was arrested in Coimbatore"
}
```

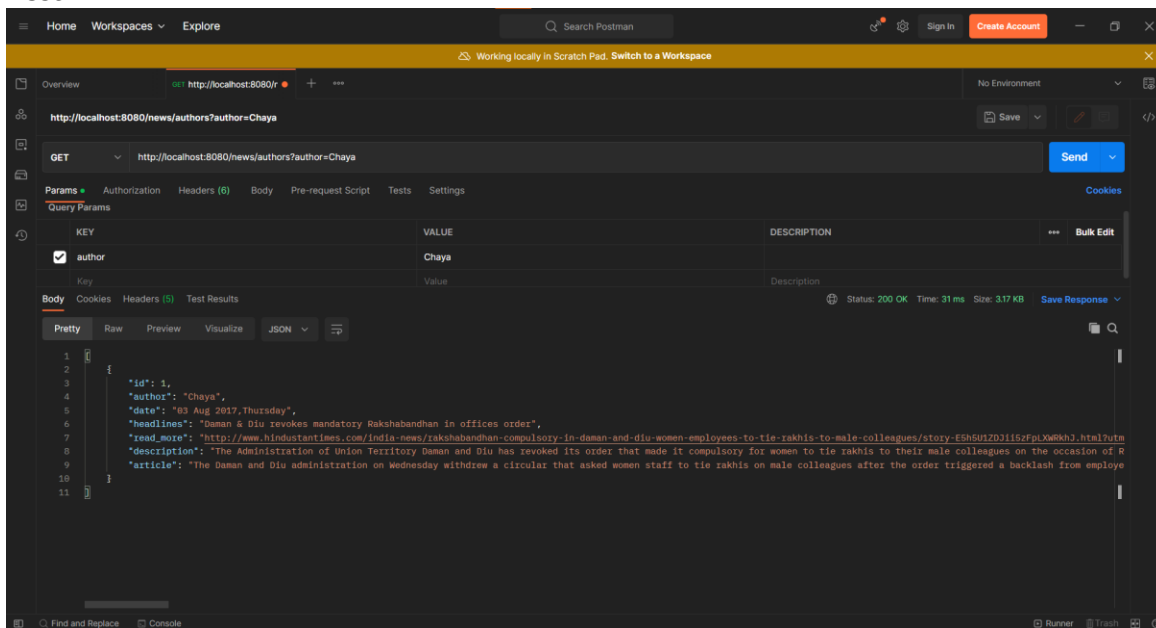
Bottom Screenshot (DELETE Request):

- Method:** DELETE
- URL:** `http://localhost:8080/news/1001`
- Status:** 200 OK
- Time:** 10 ms
- Size:** 180 B
- Body (Pretty):**

```
{
  "deleted": true
}
```



Test 4:



Test 6: Junit Testing

```
@Test
public void saveNews_success() {
    System.out.print(mockNews.toString());
    newsRepository.save(mockNews);
    List<News> newss = newsRepository.findAll();
    assertThat(newss).isNotNull();
}

@Test
public void findAll_success() {
    Optional<News> newss = newsRepository.findById((long)1);
    assertThat(newss).isEmpty();
}

@Test
public void should_find_no_news_if_repository_is_empty() {
    List<News> newss = newsRepository.findAll();
    assertThat(newss).isEmpty();
}

@Test
public void should_find_news_by_author() {
    List<News> newss = new ArrayList<News>();
    newsRepository.findByAuthorContaining("Chaya").forEach(newss::add);

    assertThat(newss).isEmpty();
}
```

Finished after 1.646 seconds

Runs: 4/4 Errors: 0 Failures: 0

SpringbootBackendApplicationTests [Runner: JUnit 5] (0.060 s)

- should_find_no_news_if_repository_is_empty() (0.046 s)
- saveNews_success() (0.004 s)
- should_find_news_by_author() (0.003 s)
- findAll_success() (0.004 s)

Failure Trace

Figure12. Diagram for the News Manager's Testing