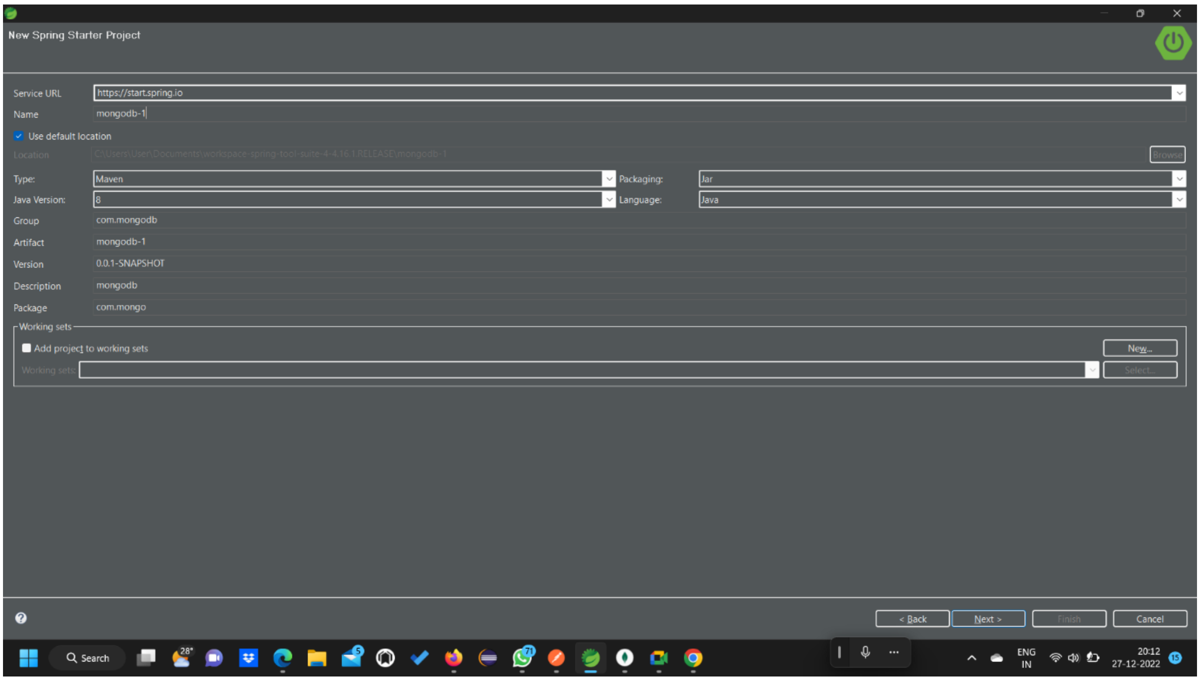
Spring Boot MongoDB CRUD example with Maven

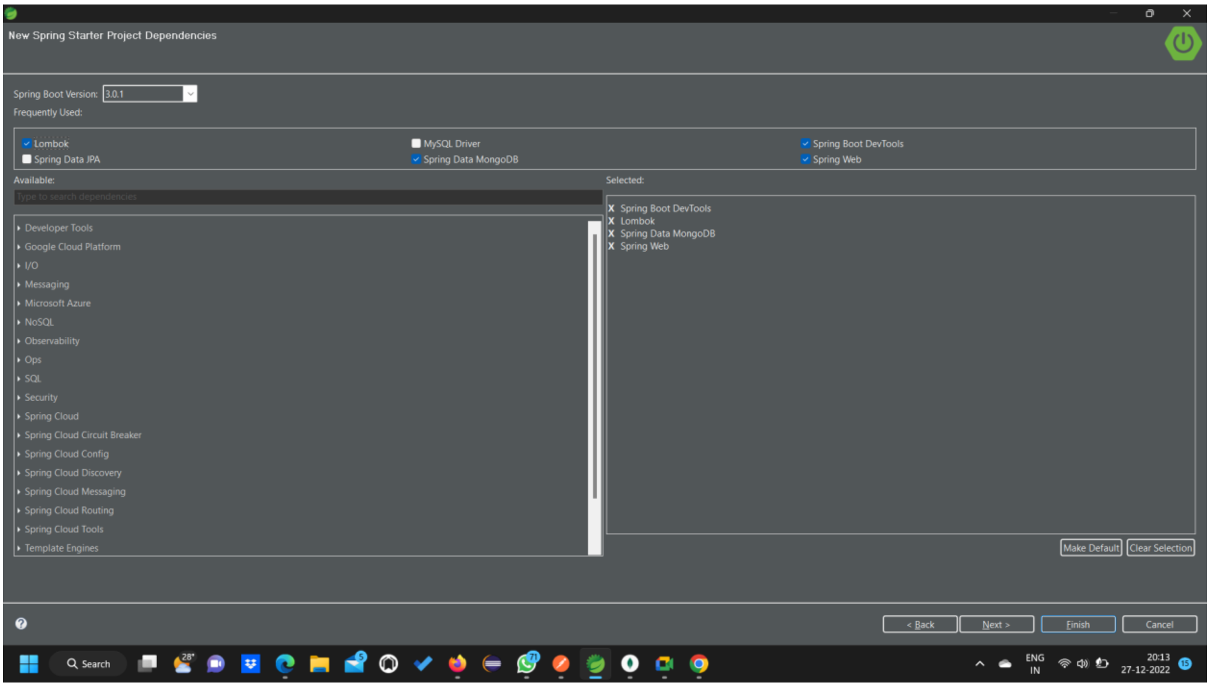
- ➢ Create a new project
- ➢ Select Spring Starter Project



New Spring Starter Project

Service URL: https://start.spring.io
Name: mongodb-1
☑ Use default location
Location: C:\Users\User\Documents\workspace-spring-tool-suite-4-4.16.1.RELEASE\mongodb-1
Type: Maven
Packaging: Jar
Java Version: 8
Language: Java
Group: com.mongodb
Artifact: mongodb-1
Version: 0.0.1-SNAPSHOT
Description: mongodb
Package: com.mongo
Working sets
□ Add project to working sets

- ➢ Add the Dependency



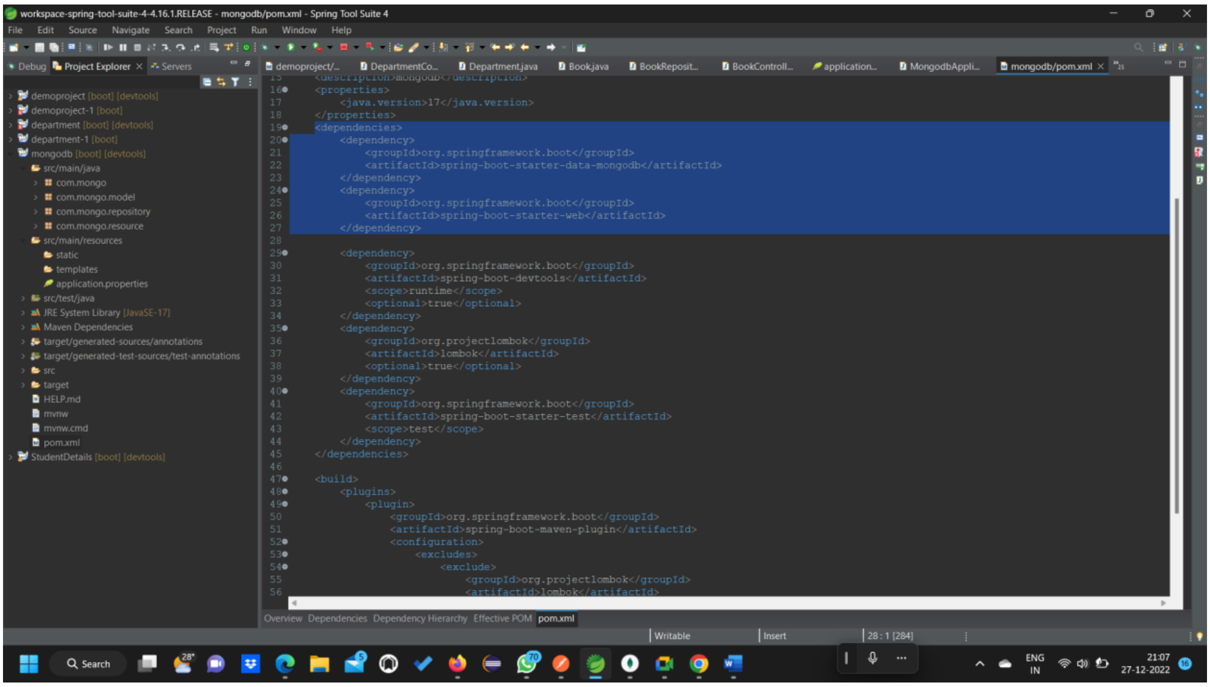
New Spring Starter Project Dependencies

Spring Boot Version: 3.0.1
Frequently Used:
☑ Lombok
□ Spring Data JPA
□ MySQL Driver
☑ Spring Data MongoDB
☑ Spring Boot DevTools
☑ Spring Web

Selected:
X Spring Boot DevTools
X Lombok
X Spring Data MongoDB
X Spring Web

Create & Setup Spring Boot project

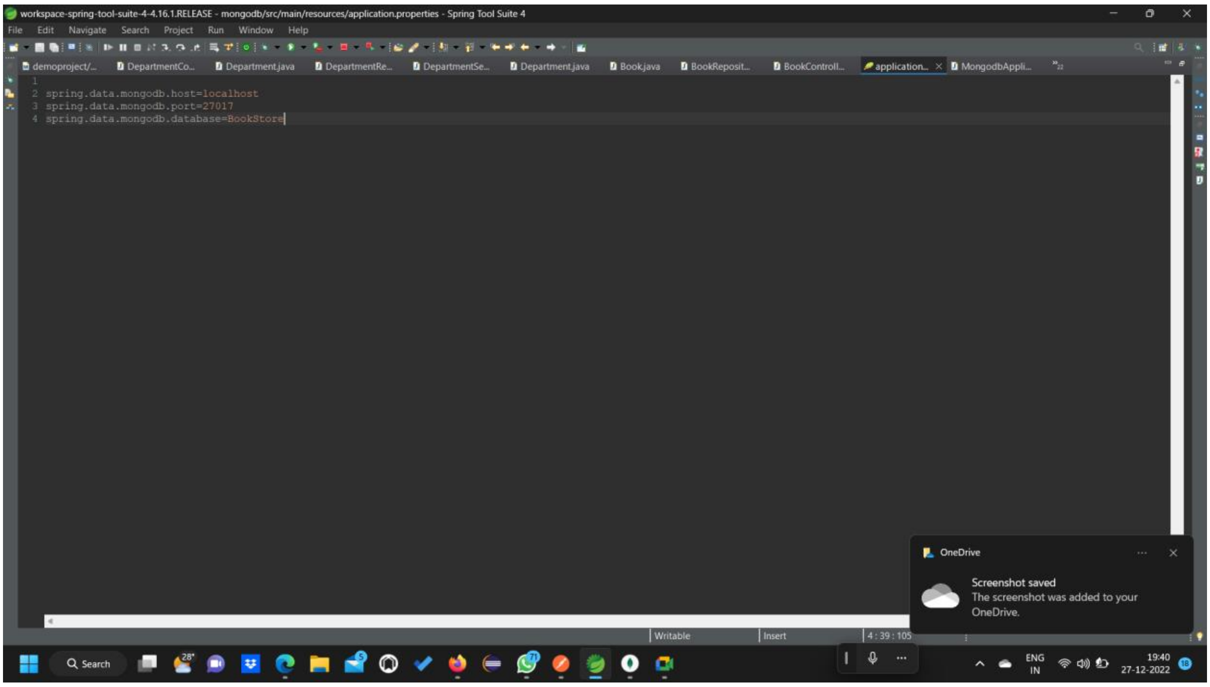
- Then open pom.xml and add these dependencies:



```xml
	<description>mongodb</description>
	<properties>
		<java.version>17</java.version>
	</properties>
	<dependencies>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-data-mongodb</artifactId>
		</dependency>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-web</artifactId>
		</dependency>

		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-devtools</artifactId>
			<scope>runtime</scope>
			<optional>true</optional>
		</dependency>
		<dependency>
			<groupId>org.projectlombok</groupId>
			<artifactId>lombok</artifactId>
			<optional>true</optional>
		</dependency>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-test</artifactId>
			<scope>test</scope>
		</dependency>
	</dependencies>

	<build>
		<plugins>
			<plugin>
				<groupId>org.springframework.boot</groupId>
				<artifactId>spring-boot-maven-plugin</artifactId>
				<configuration>
					<excludes>
						<exclude>
							<groupId>org.projectlombok</groupId>
							<artifactId>lombok</artifactId>
```

Configure Spring Data MongoDB

- Under src/main/resources folder, open application.properties and add following lines.

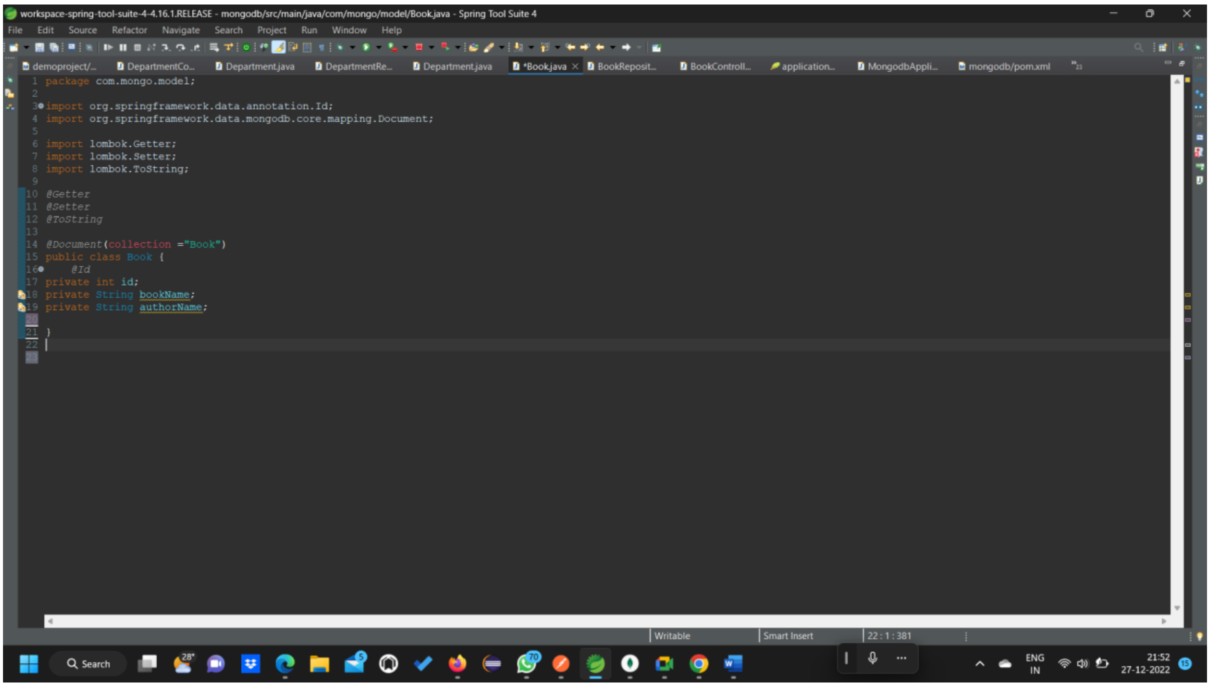


```properties

spring.data.mongodb.host=localhost
spring.data.mongodb.port=27017
spring.data.mongodb.database=BookStore
```

OneDrive

Screenshot saved
The screenshot was added to your OneDrive.

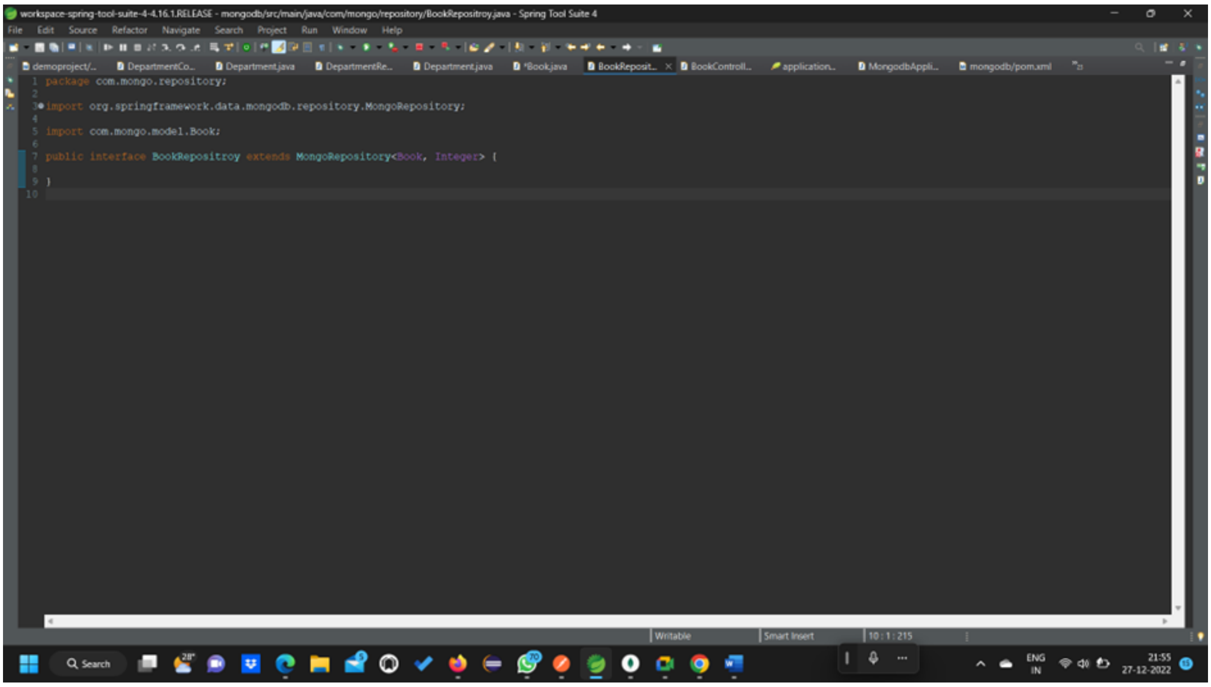
- ➢ Create 4 packages as listed and create some classes and interfaces inside these packages as seen in the below image as shown:
  - model
  - Repository
  - Resource
- ➢ Inside the model Package Create Book.java Class



workspace-spring-tool-suite-4-4.16.1.RELEASE - mongodb/src/main/java/com/mongo/model/Book.java - Spring Tool Suite 4

```java
package com.mongo.model;

import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;

import lombok.Getter;
import lombok.Setter;
import lombok.ToString;

@Getter
@Setter
@ToString

@Document(collection ="Book")
public class Book {
    @Id
private int id;
private String bookName;
private String authorName;

}

```

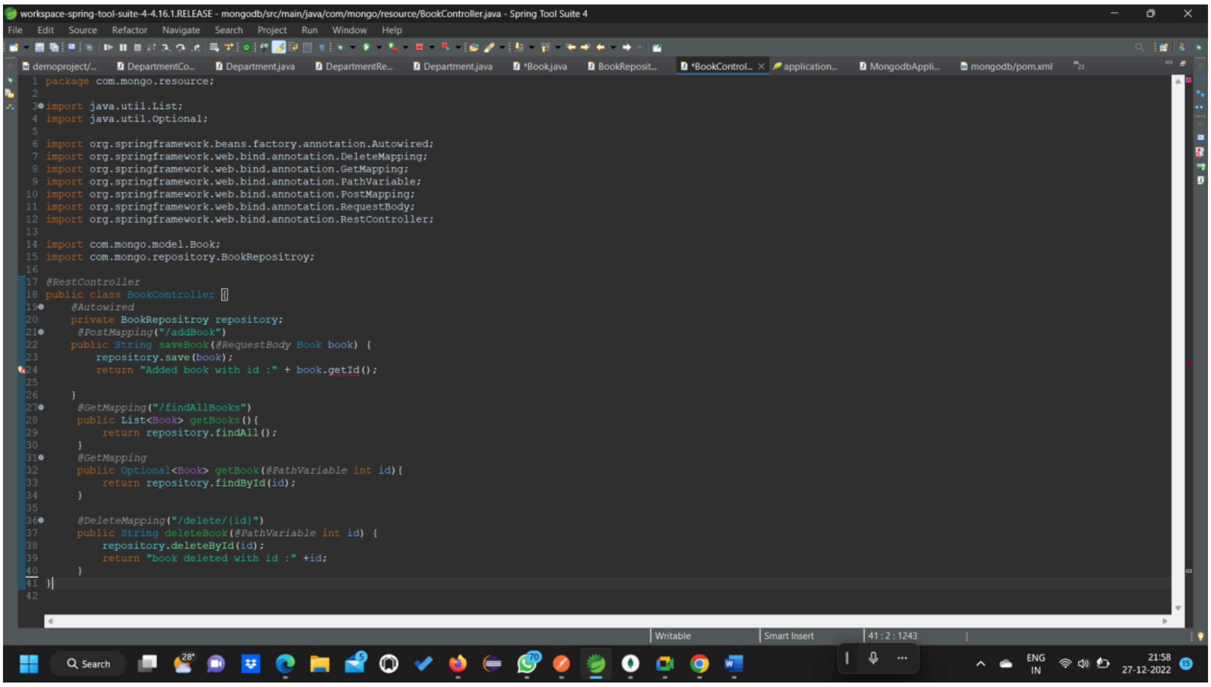
- ➢ Inside the repository Package Create Interface as BookRepository



workspace-spring-tool-suite-4-4.16.1.RELEASE - mongodb/src/main/java/com/mongo/repository/BookRepositroy.java - Spring Tool Suite 4

```java
package com.mongo.repository;

import org.springframework.data.mongodb.repository.MongoRepository;

import com.mongo.model.Book;

public interface BookRepositroy extends MongoRepository<Book, Integer> {

}

```

- Inside the resource Package create Class as BookController

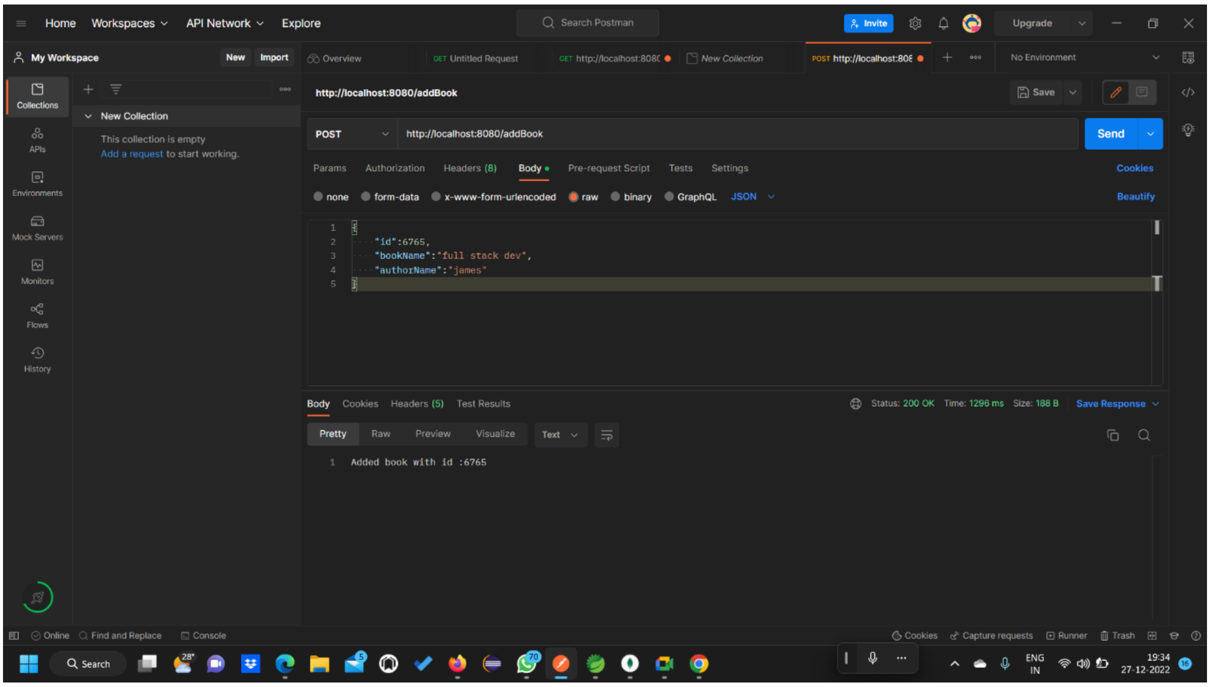


workspace-spring-tool-suite-4-4.16.1.RELEASE - mongodb/src/main/java/com/mongo/resource/BookController.java - Spring Tool Suite 4

```java
package com.mongo.resource;

import java.util.List;
import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;

import com.mongo.model.Book;
import com.mongo.repository.BookRepositroy;

@RestController
public class BookController {
    @Autowired
    private BookRepositroy repository;
     @PostMapping("/addBook")
    public String saveBook(@RequestBody Book book) {
        repository.save(book);
        return "Added book with id :" + book.getId();

    }
     @GetMapping("/findAllBooks")
     public List<Book> getBooks(){
         return repository.findAll();
     }
     @GetMapping
     public Optional<Book> getBook(@PathVariable int id){
         return repository.findById(id);
     }

     @DeleteMapping("/delete/{id}")
     public String deleteBook(@PathVariable int id) {
         repository.deleteById(id);
         return "book deleted with id :" +id;
     }
}
```

Testing the Endpoint in Postman:-

Endpoint : POST – http://localhost:8080/addBook



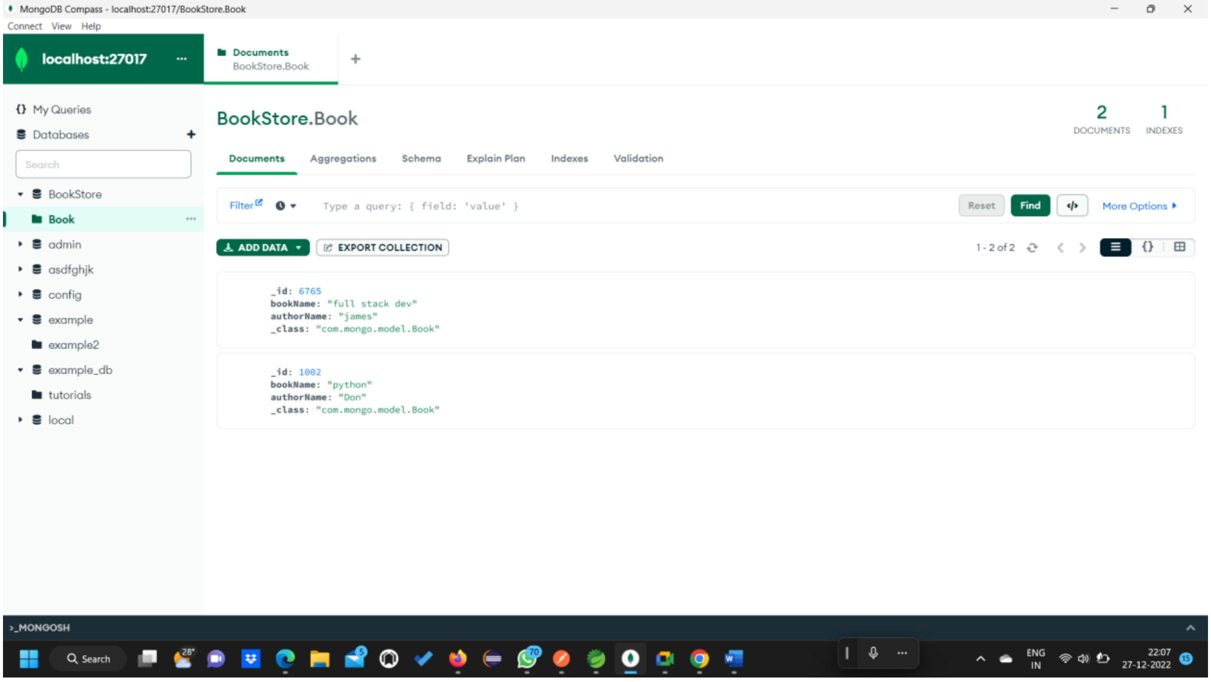
POST http://localhost:8080/addBook

```json
{
    "id":6765,
    "bookName":"full stack dev",
    "authorName":"james"
}
```

Status: 200 OK Time: 1296 ms Size: 188 B

```
Added book with id :6765
```

- Check MongoDB database:



BookStore.Book

2 DOCUMENTS 1 INDEXES

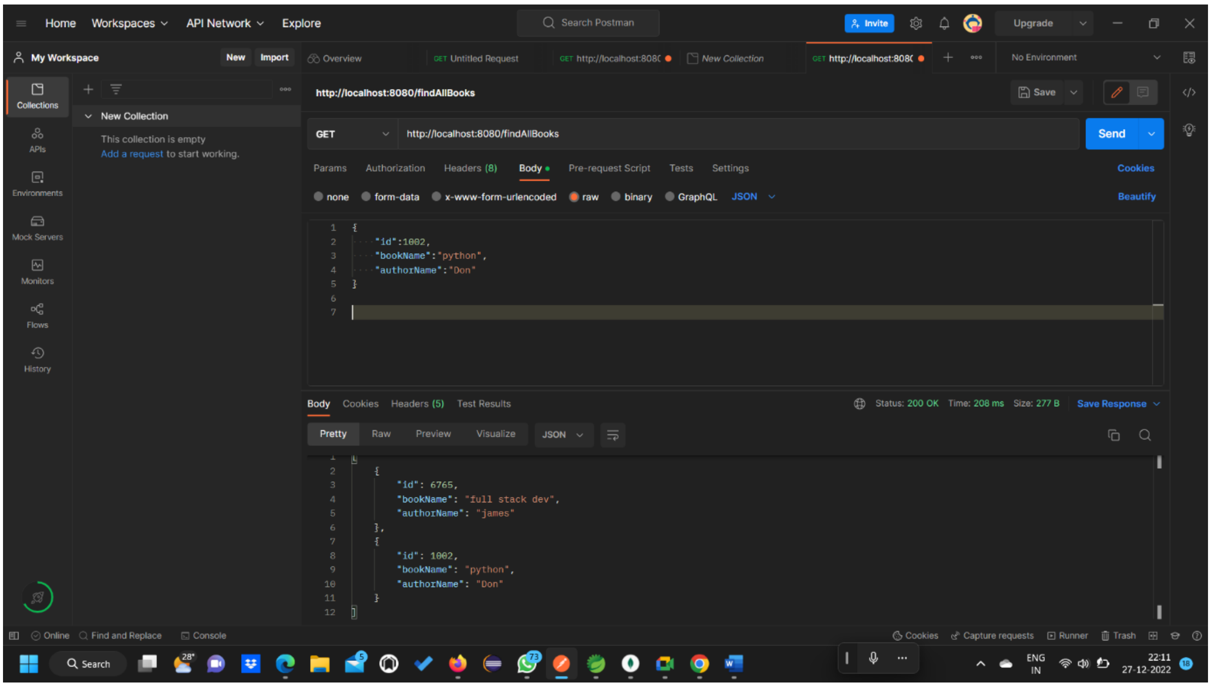
Type a query: { field: 'value' }

1 - 2 of 2

```
_id: 6765
bookName: "full stack dev"
authorName: "james"
_class: "com.mongo.model.Book"
```

```
_id: 1002
bookName: "python"
authorName: "Don"
_class: "com.mongo.model.Book"
```

- Get all Books :



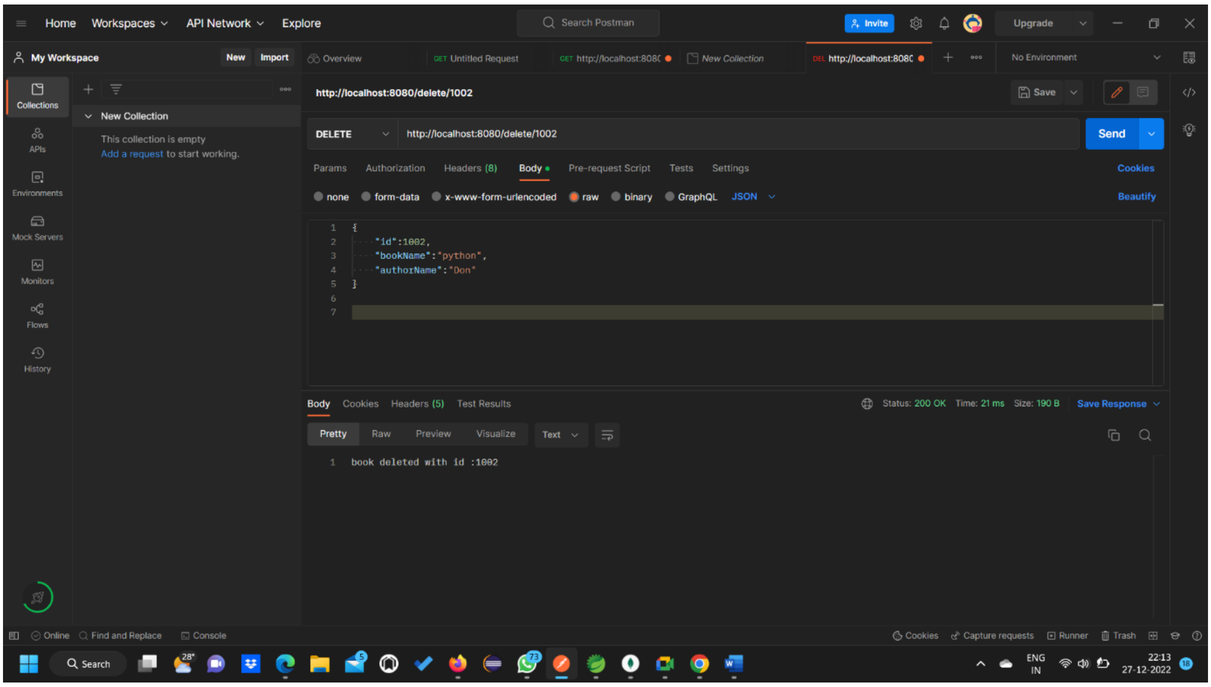
GET http://localhost:8080/findAllBooks

```
{
    "id":1002,
    "bookName":"python",
    "authorName":"Don"
}
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 208 ms Size: 277 B

```
[
    {
        "id": 6765,
        "bookName": "full stack dev",
        "authorName": "james"
    },
    {
        "id": 1002,
        "bookName": "python",
        "authorName": "Don"
    }
]
```

- Delete a Book:



http://localhost:8080/delete/1002

DELETE http://localhost:8080/delete/1002

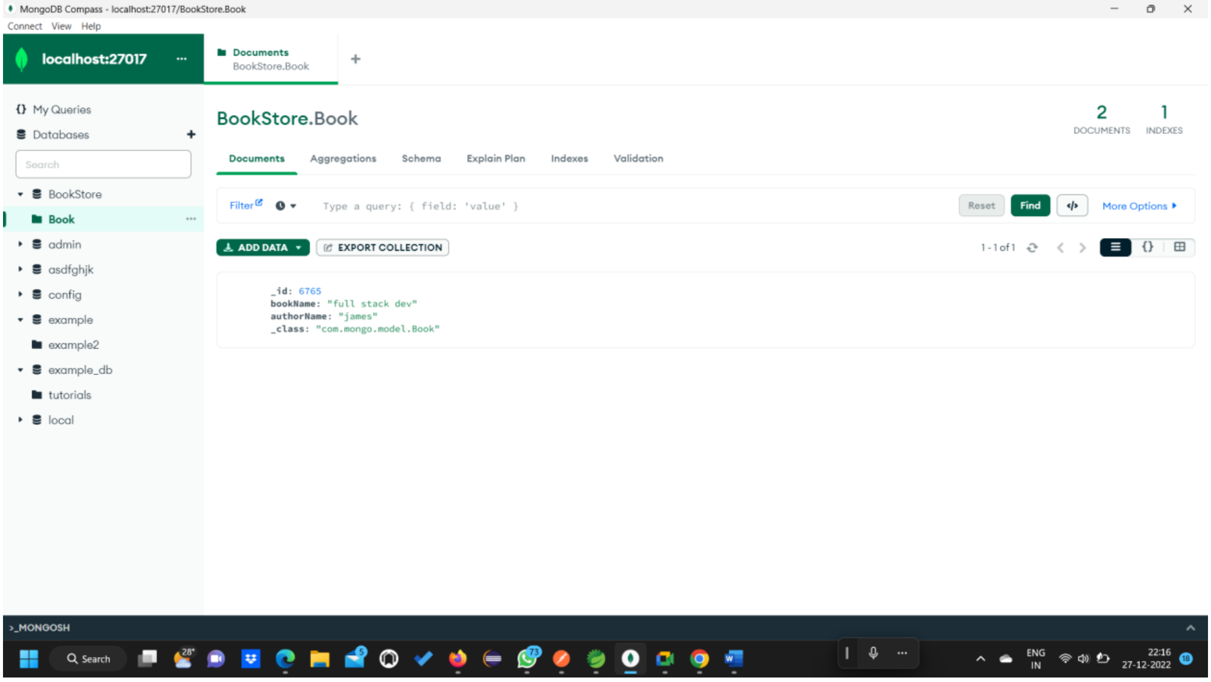
Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "id":1002,
    "bookName":"python",
    "authorName":"Don"
}
```

Status: 200 OK Time: 21 ms Size: 190 B

```
book deleted with id :1002
```

- Check MongoDB database after deleting the Tutorial:



BookStore.Book

2 DOCUMENTS 1 INDEXES

```
_id: 6765
bookName: "full stack dev"
authorName: "james"
_class: "com.mongo.model.Book"
```