package com.example.bookstore.model;

import org.springframework.data.annotation.Id;

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

@Document(collection = "books")

public class Book {

@Id

private String id;

private String title;

private String author;

private String description;

private Integer year;

// Getters and Setters

}

package com.example.bookstore.repository;

import com.example.bookstore.model.Book;

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

public interface BookRepository extends MongoRepository<Book, String> {

}

package com.example.bookstore.service;

import com.example.bookstore.model.Book;

import com.example.bookstore.repository.BookRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.util.List;

import java.util.Optional;

@Service

public class BookService {

@Autowired

private BookRepository repository;

public Book createBook(Book book) {

return repository.save(book);

}

public List<Book> getAllBooks() {

return repository.findAll();

}

public Optional<Book> getBookById(String id) {

return repository.findById(id);

}

public Book updateBook(String id, Book bookDetails) {

Book book = repository.findById(id).orElseThrow();

book.setTitle(bookDetails.getTitle());

book.setAuthor(bookDetails.getAuthor());

book.setDescription(bookDetails.getDescription());

book.setYear(bookDetails.getYear());

return repository.save(book);

}

public void deleteBook(String id) {

repository.deleteById(id);

}

}

package com.example.bookstore.controller;

import com.example.bookstore.model.Book;

import com.example.bookstore.service.BookService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

@RestController

@RequestMapping("/books")

public class BookController {

@Autowired

private BookService service;

@PostMapping

public ResponseEntity<Book> createBook(@RequestBody Book book) {

return ResponseEntity.status(201).body(service.createBook(book));

}

@GetMapping

public List<Book> getAllBooks() {

return service.getAllBooks();

}

@GetMapping("/{id}")

public ResponseEntity<Book> getBookById(@PathVariable String id) {

return service.getBookById(id)

.map(ResponseEntity::ok)

.orElse(ResponseEntity.notFound().build());

}

@PutMapping("/{id}")

public ResponseEntity<Book> updateBook(@PathVariable String id, @RequestBody Book bookDetails) {

return ResponseEntity.ok(service.updateBook(id, bookDetails));

}

@DeleteMapping("/{id}")

public ResponseEntity<Void> deleteBook(@PathVariable String id) {

service.deleteBook(id);

return ResponseEntity.noContent().build();

}

}

package com.example.bookstore;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class BookstoreApplication {

public static void main(String[] args) {

SpringApplication.run(BookstoreApplication.class, args);

}

}

MongoDB Document books collection

{

"\_id": "662ffb2c1234567890abcd",

"title": "Spring in Action",

"author": "Craig Walls",

"description": "A comprehensive guide to Spring",

"year": 2021

}

To Test with Postman

{

"title": "Effective Java",

"author": "Joshua Bloch",

"description": "Best practices for Java programming",

"year": 2018

}

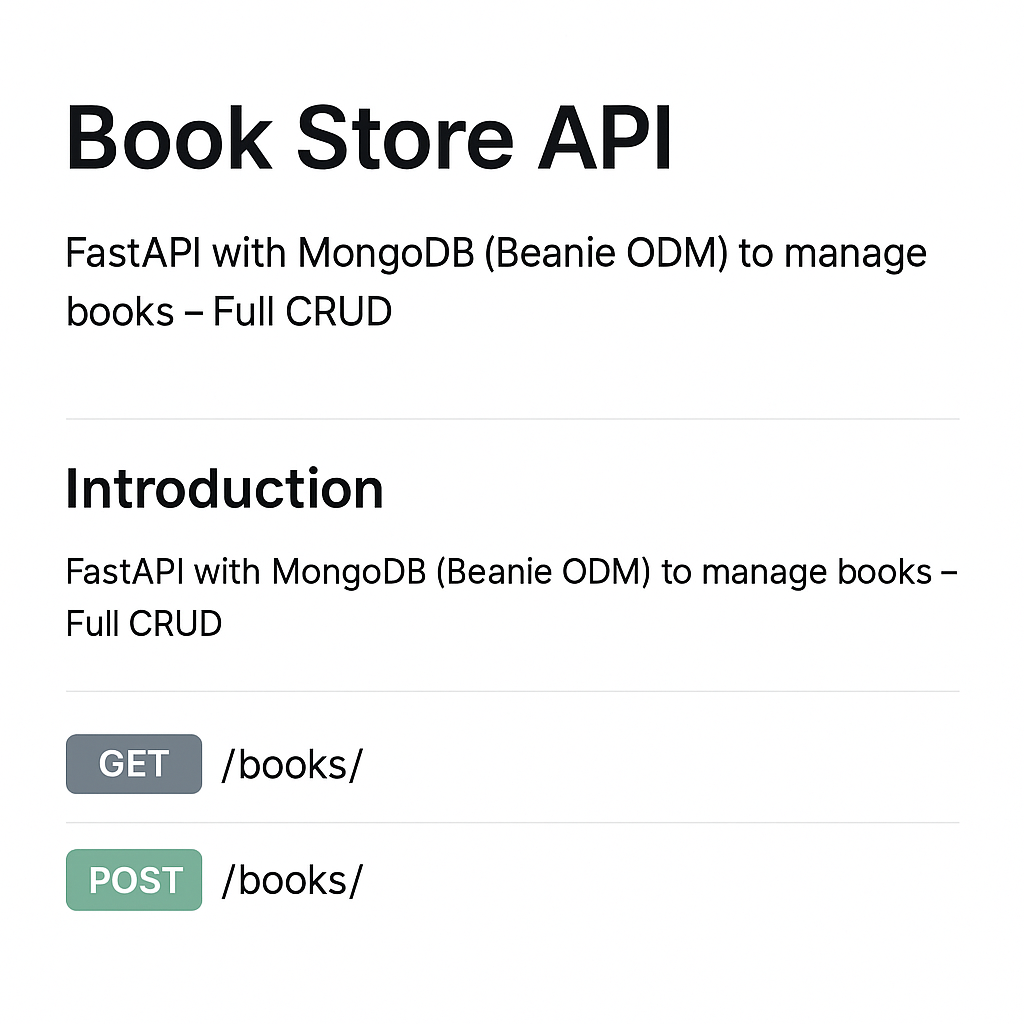


IMAGE FOR CRUD OPERATIONS

