

REST API

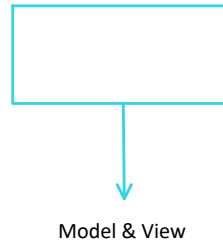
03 August 2025 10:43



REST => Representational state transfer.

Note: REST APIs expose resource via http endpoints like

1. Get /products
2. Post /products
3. Get /product/{id}
4. PUT /product/{id}
5. DELETE /product/{id}
6. PATCH /product/{id}



Understanding HTTP status code:

200 OK -> Success(Get, Put, Delete)

201 Created -> Success(POST)

204 No content -> success without response body

400 Bad Request -> client error (invalid input) -> missing field, validation error

401 Unauthorized -> Not Authenticated

403 Forbidden -> Authenticated but no access

404 Not found -> resource not found

500 Internal server error -> server-side bug

Project : Design Rest API for a book resource.

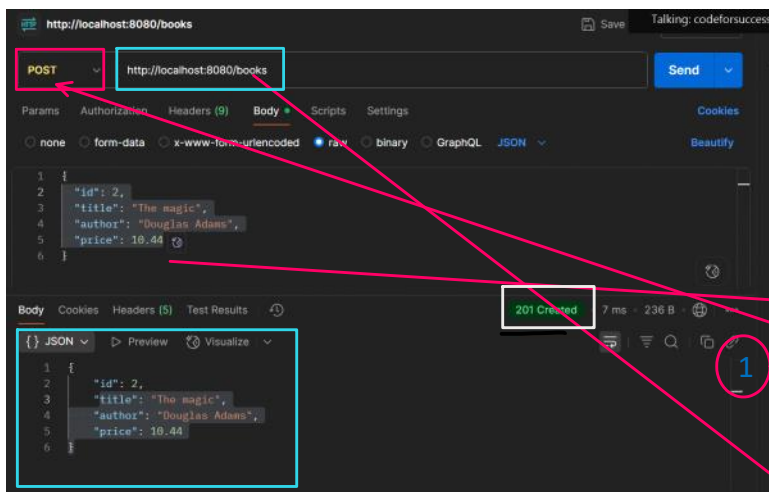
Entity : Book

Book :

- id
- title
- author
- price

Define restful points

Operation	Http Methods	Url	Des
Get all books	GET	/books	Fetch list of all books: 200 ok
Get a book by ID	GET	/book{id}	Fetch a single book : 200 OK, 404
Add new book	POST	/books	Create new Book : 201 created, 404



response

```
13 usages new *
public class Book {
    3 usages
    private Long id;
    3 usages
    private String title;
    3 usages
    private String author;
    3 usages
    private Double price;
}
```

Json to java object

```
@PostMapping
public ResponseEntity<Book> createBook(@RequestBody Book book)
{
    3 bookDB.put(book.getId(), book);
    return ResponseEntity.status(HttpStatus.CREATED).body(book);
}

@RestController
@RequestMapping("/books")
public class BookController {
```

2

4