

## JDBC Client - Spring controller 3.2

```
BookController.java x Book.java x JdbcClientExampleApplication.java x application.properties x
4 import lombok.Data;
5 import lombok.NoArgsConstructor;
6
7 @Data
8 @AllArgsConstructor
9 @NoArgsConstructor
10 public class Book {
11
12     private int id;
13     private String name;
14     private String title;
15 }
16
```

```
pom.xml (jdbc-client-example) x Book.java x schema.sql x
DB Connections
1 DROP TABLE IF EXISTS Book;
2
3 CREATE TABLE `Book` (
4     `id` int NOT NULL,
5     `name` varchar(255) DEFAULT NULL,
6     `title` varchar(255) DEFAULT NULL,
7     PRIMARY KEY (`id`)
8 );
```

The above query executes always if we add below property.

```
pom.xml (jdbc-client-example) x Book.java x schema.sql x application.properties x
1 spring.sql.init.mode=always
```

```
BookController.java
16 @Autowired
17 private JdbcClient jdbcClient;
18
19 @PostMapping
20 @RequestMapping("/addNewBook")
21 public String addNewBook(@RequestBody Book book) {
22     jdbcClient.sql("INSERT INTO book(id,name,title) values(?,?,?)")
23         .param(List.of(book.getId(), book.getName(), book.getTitle()))
24         .update();
25     return "book added to the system";
26 }
```

```

26
27 @GetMapping
28 public List<Book> getAllBooks() {
29     return jdbcClient.sql("SELECT id, name, title FROM book")
30         .query(Book.class)
31         .list();
32 }
33

```

```

3
4 @GetMapping("/{id}")
5 public Optional<Book> getBookById(@PathVariable int id) {
6     return jdbcClient.sql("SELECT id, name, title FROM book where id=:id")
7         .param(name: "id", id)
8         .query(Book.class).optional();
9 }
10

```

```

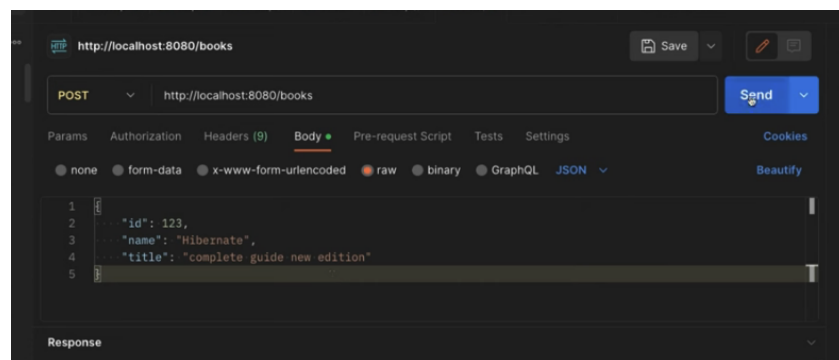
jdbc-client-example src main java com.javatechie controller BookController getBookById Add Configuration...
BookController.java
41 @PutMapping("/{id}")
42 @
43 public String updateBook(@PathVariable int id, @RequestBody Book book) {
44     jdbcClient.sql("update book set title = ?, name = ? where id = ?")
45         .param(List.of(book.getTitle(), book.getName(), id))
46         .update();
47     return "book modified in system";
48 }
49

```

```

@DeleteMapping("/{id}")
public String deleteBook(@PathVariable int id) {
    jdbcClient.sql("delete from book where id=:id")
        .param(name: "id", id)
        .update();
    return "book record removed !";
}

```



book 1 23

SQL Editor: `SELECT * FROM Book b` Enter a SQL expression to filter results (use Ctrl+Space)

	id	name	title
1	123	Hibernate	complete guide new edition
2	143	core java	complete java guide latest edition