

# CRUD Application

Using MySQL

# Add Dependencies

## Dependencies

ADD DEPENDENCIES... CTRL + B

### Spring Web

WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.



### Spring Boot DevTools

DEVELOPER TOOLS

Provides fast application restarts, LiveReload, and configurations for enhanced development experience.



### MySQL Driver

SQL

MySQL JDBC driver.



### Spring Data JPA

SQL

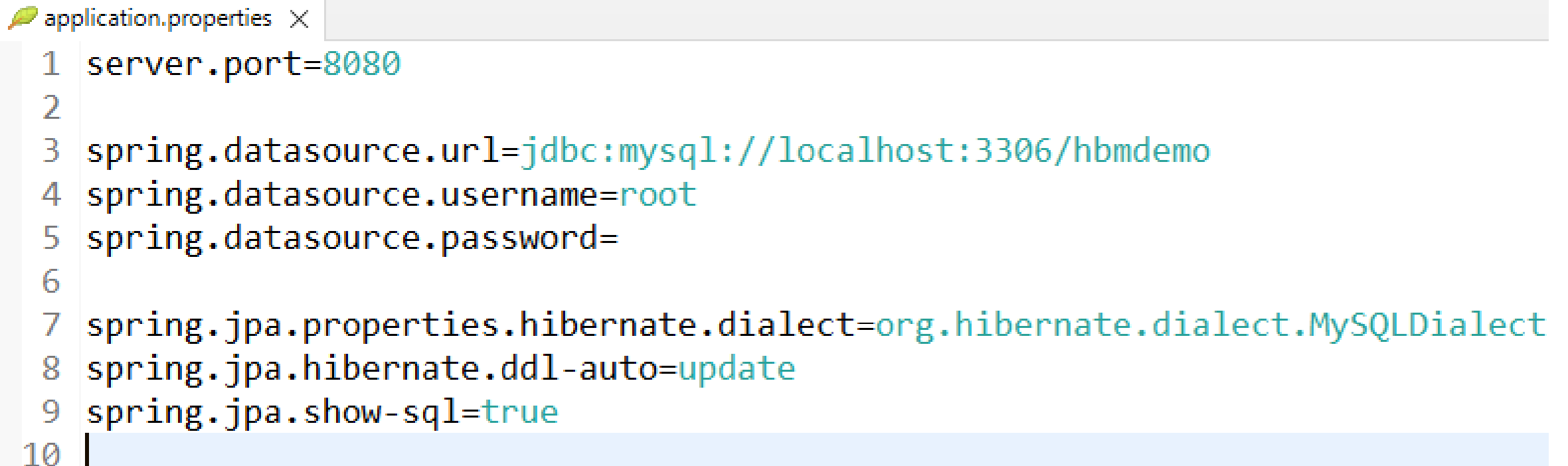
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.



# Add Dependencies

```
<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>com.mysql</groupId>
  <artifactId>mysql-connector-j</artifactId>
  <scope>runtime</scope>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```

# Add Configuration Details

A screenshot of an IDE's snippet editor for the file 'application.properties'. The editor shows a list of configuration properties for a Spring application, including server port, database URL, username, password, and JPA settings. The text is color-coded: keywords like 'spring' are in blue, and values like '8080' and 'root' are in green. Line numbers 1 through 10 are visible on the left side of the editor.







```
application.properties ×  
1 server.port=8080  
2  
3 spring.datasource.url=jdbc:mysql://localhost:3306/hbmdemo  
4 spring.datasource.username=root  
5 spring.datasource.password=  
6  
7 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect  
8 spring.jpa.hibernate.ddl-auto=update  
9 spring.jpa.show-sql=true  
10 |
```

# Entity

```
Student.java x
1 package com.rit.p4crud;
2
3+ import jakarta.persistence.Entity;
7
8 @Entity
9 public class Student {
10
11-     @Id
12     @GeneratedValue(strategy = GenerationType.IDENTITY)
13     private Long roll;
14     private String name;
15     private double mark;
16
17     //Constructor, Getter & Setter
18 }
```

- > com.rit.p3aop
- ▼ com.rit.p4crud
  - > Student.java
  - > StudentController.java
  - > StudentRepository.java
- > src/main/resources

# Repository

- >  com.rit.p3aop
- ▼  com.rit.p4crud
  - >  Student.java
  - >  StudentController.java
  - >  StudentRepository.java
- >  src/main/resources

```
StudentRepository.java ×  
1 package com.rit.SBDemo1.controller;  
2  
3 import org.springframework.data.jpa.repository.JpaRepository;  
4  
5  
6 @Repository  
7 public interface StudentRepository extends JpaRepository<Student, Long> {  
8  
9 }  
10
```

# Controller

StudentController.java ×

```
15 @RestController
16 @RequestMapping("stud")
17 public class StudentController {
18
19     @Autowired
20     private StudentRepository repo;
21
22     @PostMapping
23     public Student addStud(@RequestBody Student stud) {
24         return repo.save(stud);
25     }
26
27     @GetMapping
28     public List<Student> viewAll() {
29         return repo.findAll();
30     }
31 }
```

> com.rit.p3aop

▼ com.rit.p4crud

> Student.java

> StudentController.java

> StudentRepository.java

> src/main/resources

# Controller (cont...)

---

```
31
32 @GetMapping("/{id}")
33 public Student viewStud(@PathVariable Long id) {
34     return repo.findById(id).get();
35 }
36
37 @PutMapping
38 public Student updateStud(@RequestBody Student stud) {
39     return repo.save(stud);
40 }
41
42 @DeleteMapping("/{id}")
43 public String deleteStud(@PathVariable Long id) {
44     if(repo.findById(id).isPresent()) {
45         repo.deleteById(id);
46         return "Student Record Deleted Successfully";
47     }
48     else
49         return "Record Not Found !!!";
50 }
51 }
```



Test the api's with Postman

POST



http://localhost:8080/stud

Send



Body



...

raw



JSON



Beautify

```
1 {  
2   ... "name": "Vidhya",  
3   ... "mark": 83.55  
4 }
```

Body



200 OK

524 ms

203 B



Pretty



JSON



```
1 {  
2   "roll": 1,  
3   "name": "Vidhya",  
4   "mark": 83.55  
5 }
```

GET



http://localhost:8080/stud

Send



Body



raw



JSON



Beautify

1

Body



200 OK

28 ms

282 B



Pretty



JSON



1

[

2

{

3

"roll": 1,

4

"name": "Vidhya",

5

"mark": 83.55

6

},

7

{

8

"roll": 2,

9

"name": "Ram",

10

"mark": 87.55

11

].

GET



http://localhost:8080/stud/1



Send



Body



raw



JSON



Beautify

1

Body



200 OK

128 ms

203 B



Pretty



JSON



1

{

2

"roll": 1,

3

"name": "Vidhya",

4

"mark": 83.55

5

}

PUT



http://localhost:8080/stud

Send



Body



raw



JSON



Beautify

```
1 {  
2   .... "roll": 1,  
3   .... "name": "RAMYA",  
4   .... "mark": 77.55  
5 }
```

Body



200 OK

199 ms

202 B



Pretty



JSON



```
1 {  
2   "roll": 1,  
3   "name": "RAMYA",  
4   "mark": 77.55  
5 }
```

DELETE

http://localhost:8080/stud/1

Send

Body

...

raw

JSON

Beautify

1

Body



200 OK

139 ms

199 B

...

Pretty

Text



1

Student Record Deleted Successfully

DELETE

http://localhost:8080/stud/101

Send

Body

...

raw

JSON

Beautify

1

Body



200 OK

7 ms

184 B

...

Pretty

Text



1

Record Not Found !!!

GET

http://localhost:8080/stud

Send

Body

...

raw

JSON

Beautify

1

Body



200 OK

7 ms

242 B

...

Pretty

JSON



1

2

3

4

5

6

7

8

9

10

11

{

"roll": 2,

"name": "Ram",

"mark": 87.55

},

{

"roll": 3,













"name": "Balaji",

"mark": 45.55

}

# CRUD Application

Hibernate Mapping

- ▼  SBDemo1 [boot] [devtools]
  - ▼  src/main/java
    - >  com.rit
    - >  com.rit.p1intro
    - >  com.rit.p2mapping
    - >  com.rit.p3aop
    - >  com.rit.p4crud
    - ▼  com.rit.p5crud
      - >  Address.java
      - >  EmpController.java
      - >  Employee.java
      - >  EmpRepository.java



Address.java X

```
/
8 @Entity
9 public class Address {
10
11     @Id
12     @GeneratedValue(strategy = GenerationType.IDENTITY)
13     private Long addrId;
14     private String city;
15     private Integer pincode;
16
17     //Constructor, Getter & Setter
18
```

```
10
11 @Entity
12 public class Employee {
13
14     @Id
15     @GeneratedValue(strategy = GenerationType.IDENTITY)
16     private Long empid;
17     private String name;
18     private String email;
19     private String phone;
20
21     @OneToOne(cascade=CascadeType.ALL)
22     @JoinColumn(name="addr_id")
23     private Address address;
24
25     //Constructor, Getter & Setter
```

EmpRepository.java ×

```
1 package com.rit.p5crud;  
2  
3 import org.springframework.data.jpa.repository.JpaRepository;  
4  
5  
6 @Repository  
7 public interface EmpRepository extends JpaRepository<Employee, Long> {  
8  
9 }  
10
```

```
14
15 @RestController
16 @RequestMapping("emp")
17 public class EmpController {
18
19     @Autowired
20     private EmpRepository repo;
21
22     @PostMapping
23     public Employee addStud(@RequestBody Employee emp) {
24         return repo.save(emp);
25     }
26
27     @GetMapping
28     public List<Employee> viewAll() {
29         return repo.findAll();
30     }
```

```
⊖ @GetMapping("/{id}")  
    public Employee viewEmp(@PathVariable Long id) {  
        return repo.findById(id).get();  
    }  
  
⊖ @PutMapping  
    public Employee updateEmp(@RequestBody Employee emp) {  
        return repo.save(emp);  
    }  
  
⊖ @DeleteMapping("/{id}")  
    public String deleteEmp(@PathVariable Long id) {  
        if(repo.findById(id).isPresent()) {  
            repo.deleteById(id);  
            return "Employee Record Deleted Successfully";  
        }  
        else  
            return "Record Not Found !!!";  
    }  
}
```

POST



http://localhost:8080/emp

Send



Body



raw



JSON



Beautify

```
1  {
2    .... "name": "Raj",
3    .... "email": "raj@gmail.com",
4    .... "phone": "9877657743",
5    .... "address": {
6      .... "city": "chennai",
7      .... "pincode": 600100
8    .... }
9  }
```

Body



200 OK

441 ms

290 B



Pretty



JSON



```
1  {
2    "empid": 1,
3    "name": "Raj",
4    "email": "raj@gmail.com",
5    "phone": "9877657743",
6    "address": {
7      "addrId": 1,
8      "city": "chennai",
9      "pincode": 600100
10   }
11 }
```

GET

http://localhost:8080/emp

>

Body



200 OK

142 n

Pretty

Raw

Preview

Visualize

JSON



Editor

```
1  [
2      {
3          "empid": 1,
4          "name": "Raj",
5          "email": "raj@gmail.com",
6          "phone": "9877657743",
7          "address": {
8              "addrId": 1,
9              "city": "chennai",
10             "pincode": 600100
11         }
12     },
13     {
14         "empid": 2,
15         "name": "Vijay",
16         "email": "vijay@gmail.com",
```

GET



http://localhost:8080/emp/3



Body



200 OK

71 ms

20

Pretty



JSON



```
1  {}
2    "empid": 3,
3    "name": "Devi",
4    "email": "devi@gmail.com",
5    "phone": "9338261743",
6    "address": {
7      "addrId": 3,
8      "city": "mumbai",
9      "pincode": 532123
10   }
11  {}
```

Editor



PUT

http://localhost:8080/emp

Send

Body

ooo

raw

JSON

Beautify

```
1 {
2   .... "empid": 3,
3   .... "name": "Devi",
4   .... "email": "devi@gmail.com",
5   .... "phone": "1111111111",
6   .... "address": {
7     .... "addrId": 3,
8     .... "city": "mumbai",
9     .... "pincode": 111111
10  .... }
11 }
```

Body



200 OK

216 ms

292 B

...

Pretty

JSON



```
1 {
2   "empid": 3,
3   "name": "Devi",
4   "email": "devi@gmail.com",
5   "phone": "1111111111",
6   "address": {
7     "addrId": 3,
8     "city": "mumbai",
9     "pincode": 111111
10  }
11 }
```

DELETE



http://localhost:8080/emp/3



Body



200 C

Pretty

Raw

Preview

Visualize

Text



1 Employee Record Deleted Successfully

Result Grid



Filter Rows:

Edit:



Export/Import:



	empid	email	name	phone	addr_id
▶	1	raj@gmail.com	Raj	9877657743	1
	2	vijay@gmail.com	Vijay	9332257743	2
*	NULL	NULL	NULL	NULL	NULL

Result Grid



Filter Rows:

Edit:

	addr_id	city	pincode
▶	1	chennai	600100
	2	chennai	600123
*	NULL	NULL	NULL

Thank you