

Searching for a Specific User and Updating the User Information.

1.) UsermanagerApplication.java

```
package com.example.UserManager;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class UserManagerApplication {

    public static void main(String[] args) {
        SpringApplication.run(UserManagerApplication.class, args);
    }

}
```

User.java

```
package com.example.UserManager.entities;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity // This tells Hibernate to make a table out of this class
@Table(name = "users")
public class User {

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private Integer id;

    private String name;

    private String email;

    private String password;

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public Integer getId() {
        return id;
    }
}
```

```
}
```

```
public void setId(Integer id) {  
    this.id = id;  
}
```

```
public String getName() {  
    return name;  
}
```

```
public void setName(String name) {  
    this.name = name;  
}
```

```
public String getEmail() {  
    return email;  
}
```

```
public void setEmail(String email) {  
    this.email = email;  
}
```

```
@Override
```

```
public String toString() {  
    return (id.toString() + " " + name + " " + email + " " + password);  
}
```

```
}
```

UserNotfound.java

```
package com.example.UserManager.exceptions;
```

```
public class UserNotFoundException extends RuntimeException {
```

```
    private static final long serialVersionUID = 1L;
```

```
    /*
```

```
    private final String userId;
```

```
    public UserNotFoundException() {
```

```
        super();
```

```
        this.userId = id;
```

```
    }
```

```
    public String getUserId() {
```

```
        return userId;
```

```
    }
```

```
    */
```

```
}
```

UserRepository.java

```
package com.example.UserManager.repositories;

import org.springframework.data.repository.CrudRepository;

import com.example.UserManager.entities.User;

public interface UserRepository extends CrudRepository<User, Integer> {

    public User findByName(String name);
}
```

UserService.java

```
package com.example.UserManager.services;

import java.util.Optional;

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

import com.example.UserManager.entities.User;
import com.example.UserManager.exceptions.UserNotFoundException;
import com.example.UserManager.repositories.UserRepository;
```

@Service

public class UserService {

@Autowired

private UserRepository userRepository;

public Iterable<User> GetAllUsers()

{

return userRepository.findAll();

}

public User GetUserByName(String name) {

User foundUser = userRepository.findByName(name);

return foundUser;

}

public User GetUserById(int id) {

Optional<User> foundUser = userRepository.findById(id);

//TODO: we need to decide how to handle a "Not Found" condition

if (!foundUser.isPresent()) {

System.out.println("user does not exist");

```
        throw new UserNotFoundException();
    }

    return(foundUser.get());
}

public void UpdateUser(User usertoUpdate) {
    userRepository.save(usertoUpdate);
}

}
```

Output Screenshot

Spring Application

Hello World!

[List Users](#)



Users

ID	Name	Email	Password
123	himanshu	himanshukumar@gmail.com	admin

ID:



MySQL Workbench

himanshu (adding) x

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

Query 1

```
1 show databases;
2 create database adding;
3 use adding;
4 show tables;
5 select * from users;
6 insert into users values("123","himanshukumar@gmail.com","himanshu","admin")
7
8
```

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Output

#	Time	Action	Message	Duration / Fetch
1	13:44:25	show databases	6 row(s) returned	0.000 sec / 0.000 sec
2	13:45:11	create database adding	1 row(s) affected	0.015 sec
3	13:45:16	create database adding	Error Code: 1007. Can't create database 'adding'; database exists	0.000 sec
4	13:45:45	show databases	7 row(s) returned	0.000 sec / 0.000 sec
5	13:46:06	use adding	0 row(s) affected	0.000 sec
6	14:02:34	show tables	0 row(s) returned	0.031 sec / 0.000 sec
7	15:17:00	select * from products LIMIT 0, 1000	Error Code: 1146. Table 'adding.products' doesn't exist	0.000 sec
8	15:23:29	select * from products LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec
9	16:16:45	select * from products LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Edit User

ID	Name	Email	Password
123	himanshu	himanshukumar@gmail.com	admin

Name:

Email:

Password:

[Back](#)



Edit User

ID	Name	Email	Password
123	himanshu	himanshukumar@gmail.com	admin

Name:

Email:

Password:

[Back](#)

Update successful

[Back to Users](#)

