

```
spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:mysql://${MYSQL_HOST:localhost}:3306/users
spring.datasource.username=enter username here
spring.datasource.password=enter password here
```

```
logging.level.org.springframework.web: DEBUG
spring.mvc.view.prefix=/WEB-INF/jsp/
spring.mvc.view.suffix=.jsp
server.port=8090
```

```
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);
```

```
    }
```

```
//localhost:8090/
```

```
}
```

```
package com.example.UserManager.controllers;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.SessionAttributes;

@Controller
public class MainController {

    @GetMapping(value="/")
    public String showIndexPage(ModelMap model,
        @RequestParam(value="name", required=false, defaultValue="World")
String name){
        model.addAttribute("name", name);
        return "index";
    }
}
```

```

package com.example.UserManager.controllers;

import java.util.Arrays;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;

import com.example.UserManager.entities.User;
import com.example.UserManager.services.UserService;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

@Controller
//@RequestMapping("/api")
public class UserController {

    @Autowired
    private UserService userService;

    Logger logger = LoggerFactory.getLogger(UserController.class);

    @GetMapping("/users")
    public String showUsers(ModelMap model) {

        logger.info("Getting all Users");
        Iterable<User> users = userService.GetAllUsers();

        logger.info("Passing users to view");
        model.addAttribute("users", users);

        return "users";
    }

    @RequestMapping(value="/find", method=RequestMethod.POST)//url
    public String findUser(@RequestParam("user_id") int u_id, ModelMap mod) {
        logger.info("Searching for user");
        User user = userService.GetUserById(u_id);
        String whatToDo;
        if(user != null) {
            //Iterable<User> uList = Arrays.asList(user);
            mod.addAttribute("user", user);//attribute variable different from .jsp
file
            whatToDo = "find";

```

```

        }else {
            whatToDo = "Error";
        }
        return whatToDo;//name of .jsp file
    }

    @RequestMapping(value="/save", method=RequestMethod.POST)//url
    public String saveUser(@RequestParam("su_id") int su_id,
        @RequestParam("fName") String fName, @RequestParam("eMail") String eMail,
        @RequestParam("passWord") String passWord, ModelMap mod) {
        logger.info("Updating user info");
        User user = userService.GetUserById(su_id);
        user.setName(fName);
        user.setEmail(eMail);
        user.setPassword(passWord);
        userService.UpdateUser(user);
        Iterable<User> uList = userService.GetAllUsers();
        logger.info("Passing users to view");
        mod.addAttribute("users", uList);//attribute variable different from
.jsp file
        return "users";//name of .jsp file
    }
}

```

```
package com.example.UserManager.controllers;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;

import com.example.UserManager.exceptions.UserNotFoundException;

@ControllerAdvice
public class UserExceptionHandler {
    @ExceptionHandler(value = UserNotFoundException.class)
    public ResponseEntity<Object> exception(UserNotFoundException exception) {
        return new ResponseEntity<>("User not found", HttpStatus.NOT_FOUND);
    }
}
```

```

package com.example.UserManager.entities;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity // This tells Hibernate to make a table out of this class
public class User {
    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private Integer id;

    private String name;

    private String password;

    private String email;

    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);
    }
}

```

}


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

public class UserNotFoundException extends RuntimeException {
    private static final long serialVersionUID = 1L;
}
```

```
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);
}
```

```
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()) {
            throw new UserNotFoundException();
        }

        return foundUser.get();
    }

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

```
<html>
<body>
<h2>Spring Application</h2>

<h2 class="hello-title">Hello ${name}!</h2>

<a href="users">List Users</a>

</body>
</html>
```

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

```

<html>
<head>
<style>
table, th, td {
    border: 1px solid black;
}
</style>
</head>
<body>
<h2>Users</h2>

<table style="float:left">
    <tr><th>ID</th><th>Name</th><th>Email</th><th>Password</th></tr>
    <c:forEach items="${users}" var="user" varStatus="count">
        <tr id="${count.index}">
            <td>${user.id}</td>
            <td>${user.name}</td>
            <td>${user.email}</td>
            <td>${user.password}</td>
        </tr>
    </c:forEach>
</table><br>

    <form action="find" method="post">
        Enter Users ID: <input type="text" name="user_id" /><br>
        <input type="submit" name="edit" />
    </form>

</body>
</html>

```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
    <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Edit Page</title>
</head>
<body>
<h1>Editing ${user.name }</h1>
<form action="save" method="post">
    <label for="su_id">ID:</label><br>
    <input type="text" id="su_id" name="su_id" value="${user.id}" readonly><br>
    <label for="fName">First name:</label><br>
    <input type="text" id="fName" name="fName" value="${user.name}"><br>
    <label for="eMail">email:</label><br>
    <input type="text" id="eMail" name="eMail" value="${user.email}"><br>
    <label for="passWord">password:</label><br>
    <input type="text" id="passWord" name="passWord" value="${user.password}"><br>
    <input type="submit" value="save">
</form>
</body>
</html>
```