

Session: 21 March 2025

Topics:

- Introduction to Spring Boot
- Understanding Project Structure
- Understanding Controller
- Understanding Routing
- Additional Dependencies
- Build Simple Pages and Forms
- Using Lombok
- Writing Java in JSP

Codes:

```
package com.seu.hello_spring;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.Getter;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class Course {

    private String courseCode;

    private String courseName;

    private int section;

    public Course(String courseCode) {

        this.courseCode = courseCode;

    }

}
```

```
}
```

```
package com.seu.hello_spring;

import jakarta.servlet.http.HttpServletRequest;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
@RequestMapping("/demo")
public class DemoController {

    @GetMapping("/hello")
    public String hello() {
        return "hello";
    }

    @PostMapping("/login")
    public String login(HttpServletRequest request) {
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        User user = new User(username, password, null);
//        user.setUsername(username);
//        user.setPassword(password);
    }
}
```

```
        Course c1 = new Course("CSE351", "AJ", 3);

        request.setAttribute("user", user);
        request.setAttribute("login", true);

        return "home";
    }
}
```

```
package com.seu.hello_spring;

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

@SpringBootApplication
public class HelloSpringApplication {

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

```
package com.seu.hello_spring;
```

```
import org.springframework.boot.builder.SpringApplicationBuilder;

import
org.springframework.boot.web.servlet.support.SpringBootServletInitializer;

public class ServletInitializer extends SpringBootServletInitializer {

    @Override

    protected SpringApplicationBuilder configure(SpringApplicationBuilder
application) {

        return application.sources(HelloSpringApplication.class);

    }

}
```

```
package com.seu.hello_spring;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

import java.util.List;

@Data
@NoArgsConstructor
@AllArgsConstructor

public class User {

    private String username;
```

```
    private String password;

    private List<Course> courseList;
}
```

```
spring.application.name=hello-spring
```

```
server.port=8081
```

```
spring.mvc.view.suffix=.jsp
```

```
spring.mvc.view.prefix=/pages/
```

```
<%--
```

```
    Created by IntelliJ IDEA.
```

```
    User: iftekher
```

```
    Date: 3/21/25
```

```
    Time: 9:42 AM
```

```
    To change this template use File | Settings | File Templates.
```

```
--%>
```

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
```

```
<html>
```

```
<head>
```

```
    <title>Title</title>
```

```
</head>
```

```
<body>
```

```
    <h2>Hello</h2>
```

```

    <form action="/demo/login" method="POST">

        <input type="text" name="username"/><br/>

        <input type="password" name="password"/><br>

        <input type="submit" value="Submit"/>

    </form>
</body>
</html>

```

```

<%@ page import="java.io.PrintWriter" %><!--

Created by IntelliJ IDEA.

User: iftekher

Date: 3/21/25

Time: 10:06 AM

To change this template use File | Settings | File Templates.

--%>

<%@ page contentType="text/html; charset=UTF-8" language="java" %>

<html>

<head>

    <title>Title</title>

</head>

<body>

    <h1>Welcome ${user.username}</h1>

    <%

        boolean isLogin = (boolean) request.getAttribute("login");

        if(isLogin) {

            %>

                <h1>Login Success</h1>

```

```

<%
    } else {
%>

        <h1>Login Failed</h1>

<%
    }
%>

</body>

</html>

```

```

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <parent>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-parent</artifactId>

        <version>3.4.3</version>

        <relativePath/> <!-- lookup parent from repository -->

    </parent>

    <groupId>com.seu</groupId>

    <artifactId>hello-spring</artifactId>

    <version>0.0.1-SNAPSHOT</version>

    <packaging>war</packaging>

    <name>hello-spring</name>

    <description>Demo project for Spring Boot</description>

```

```
<url/>

<licenses>

    <license/>

</licenses>

<developers>

    <developer/>

</developers>

<scm>

    <connection/>

    <developerConnection/>

    <tag/>

    <url/>

</scm>

<properties>

    <java.version>21</java.version>

</properties>

<dependencies>

    <dependency>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-web</artifactId>

    </dependency>

    <dependency>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-tomcat</artifactId>

        <scope>provided</scope>

    </dependency>

    <dependency>
```



```
        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-test</artifactId>

        <scope>test</scope>
    </dependency>

    <!--
https://mvnrepository.com/artifact/org.apache.tomcat.embed/tomcat-embed-jasper
-->

    <dependency>

        <groupId>org.apache.tomcat.embed</groupId>

        <artifactId>tomcat-embed-jasper</artifactId>

<!--        <version>11.0.5</version>-->

    </dependency>

    <!--
https://mvnrepository.com/artifact/jakarta.servlet/jakarta.servlet-api -->

    <dependency>

        <groupId>jakarta.servlet</groupId>

        <artifactId>jakarta.servlet-api</artifactId>

<!--        <version>6.1.0</version>-->

        <scope>provided</scope>

    </dependency>

    <dependency>

        <groupId>org.projectlombok</groupId>

        <artifactId>lombok</artifactId>

    </dependency>

</dependencies>
```

```
<build>

  <plugins>

    <plugin>

      <groupId>org.springframework.boot</groupId>

      <artifactId>spring-boot-maven-plugin</artifactId>

    </plugin>

  </plugins>

</build>

</project>
```

Session: 11 April 2025

Topics:

- Introduction to CSS
- Ways to implement CSS
- Ways to declare CSS
- Using Bootstrap

Code:

```
package com.seu.hello_spring;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.Getter;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor

public class Course {
```

```
private String courseCode;

private String courseName;

private int section;


public Course(String courseCode) {

    this.courseCode = courseCode;

}

}
```

```
package com.seu.hello_spring;

import jakarta.servlet.http.HttpServletRequest;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
@RequestMapping("/demo")
public class DemoController {

    @GetMapping("/hello")
    public String hello() {

        return "hello";

    }

}
```

```

@PostMapping("/login")

public String login(HttpServletRequest request) {

    String username = request.getParameter("username");

    String password = request.getParameter("password");

    User user = new User(username, password, null);
//    user.setUsername(username);
//    user.setPassword(password);

    Course c1 = new Course("CSE351", "AJ", 3);

    request.setAttribute("user", user);
    request.setAttribute("login", true);

    return "home";

}

}

```

```

package com.seu.hello_spring;

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

@SpringBootApplication

public class HelloSpringApplication {

```

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

```
package com.seu.hello_spring;  
  
import org.springframework.boot.builder.SpringApplicationBuilder;  
  
import  
org.springframework.boot.web.servlet.support.SpringBootServletInitializer;  
  
public class ServletInitializer extends SpringBootServletInitializer {  
  
    @Override  
  
    protected SpringApplicationBuilder configure(SpringApplicationBuilder  
application) {  
  
        return application.sources(HelloSpringApplication.class);  
  
    }  
  
}
```

```
package com.seu.hello_spring;  
  
import lombok.AllArgsConstructor;  
  
import lombok.Data;  
  
import lombok.NoArgsConstructor;
```

```
import java.util.List;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class User {

    private String username;

    private String password;

    private List<Course> courseList;
}
```

```
spring.application.name=hello-spring

server.port=8081

spring.mvc.view.suffix=.jsp
spring.mvc.view.prefix=/pages/
```

```
<%--
Created by IntelliJ IDEA.
User: iftekher
Date: 3/21/25
Time: 9:42 AM
To change this template use File | Settings | File Templates.
--%>
```

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>

<html>

<head>

    <title>Title</title>

    <link rel="stylesheet" href="/style.css">

    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-SgOJa3DmI69IUzQ2PVdRZhwQ+dy64/BUtbMJw1MZ8t5HZApCHrRKUc4W0kG87
9m7" crossorigin="anonymous">

</head>

<body>

    <h1>Hello</h1>

    

<%--    <form action="/demo/login" method="POST">--%>

<%--        <input type="text" name="username"/><br/>--%>

<%--        <input type="password" name="password"/><br/>--%>

<%--        <input type="submit" value="Submit"/>--%>

<%--    </form>--%>

    <div class="container">

        <form action="/demo/login" method="POST">

            <div class="mb-3">

                <label for="exampleInputEmail1"
class="form-label">Username</label>

                <input type="text" class="form-control" id="exampleInputEmail1"
aria-describedby="emailHelp">

            </div>

            <div class="mb-3">
```

```
        <label for="exampleInputPassword1"
class="form-label">Password</label>

        <input type="password" class="form-control"
id="exampleInputPassword1">

    </div>

    <button type="submit" class="btn btn-primary">Submit</button>

</form>

</div>

</body>

</html>
```

```
<%@ page import="java.io.PrintWriter" %><!--

Created by IntelliJ IDEA.

User: iftekher

Date: 3/21/25

Time: 10:06 AM

To change this template use File | Settings | File Templates.

--%>

<%@ page contentType="text/html; charset=UTF-8" language="java" %>

<html>

<head>

    <title>Title</title>

    <link rel="stylesheet" href="/style.css">

</head>

<body>

    <h1 id="main-heading-id">Welcome ${user.username}</h1>
```



```

<%
    boolean isLogin = (boolean) request.getAttribute("login");
    if(isLogin) {
        %>
        <h1>Login Success</h1>
        <%
    } else {
        %>
        <h1>Login Failed</h1>
        <%
    }
    %>
</body>
</html>

```

```

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <parent>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-parent</artifactId>

        <version>3.4.3</version>

        <relativePath/> <!-- lookup parent from repository -->

    </parent>

    <groupId>com.seu</groupId>

```

```
<artifactId>hello-spring</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>war</packaging>

<name>hello-spring</name>

<description>Demo project for Spring Boot</description>

<url/>

<licenses>

    <license/>

</licenses>

<developers>

    <developer/>

</developers>

<scm>

    <connection/>

    <developerConnection/>

    <tag/>

    <url/>

</scm>

<properties>

    <java.version>21</java.version>

</properties>

<dependencies>

    <dependency>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-web</artifactId>

    </dependency>

    <dependency>
```

```

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-tomcat</artifactId>

        <scope>provided</scope>
    </dependency>

    <dependency>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-test</artifactId>

        <scope>test</scope>
    </dependency>

    <!--
https://mvnrepository.com/artifact/org.apache.tomcat.embed/tomcat-embed-jasper
-->

    <dependency>

        <groupId>org.apache.tomcat.embed</groupId>

        <artifactId>tomcat-embed-jasper</artifactId>

<!--          <version>11.0.5</version>-->

    </dependency>

    <!--
https://mvnrepository.com/artifact/jakarta.servlet/jakarta.servlet-api -->

    <dependency>

        <groupId>jakarta.servlet</groupId>

        <artifactId>jakarta.servlet-api</artifactId>

<!--          <version>6.1.0</version>-->

        <scope>provided</scope>
    </dependency>

    <dependency>

        <groupId>org.projectlombok</groupId>

```

```
        <artifactId>lombok</artifactId>

    </dependency>

</dependencies>

<build>

    <plugins>

        <plugin>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-maven-plugin</artifactId>

        </plugin>

    </plugins>

</build>

</project>
```

Session: 02 May 2025

Topics:

- Introduction to Client Server Architecture
- Introduction to Microservice Architecture
- Introduction to API
- Rest Vs SOAP
- API Calling Methods
- Developing Rest APIs
- Explain handling error cases of APIs

Codes:

```
package com.seu.hello_spring;

import lombok.AllArgsConstructor;

import lombok.Data;
```

```
import lombok.Getter;

import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor

public class Course {

    private String courseCode;

    private String courseName;

    private int section;

    public Course(String courseCode) {

        this.courseCode = courseCode;

    }

}
```

```
package com.seu.hello_spring;

import jakarta.servlet.http.HttpServletRequest;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestMapping;

@Controller

@RequestMapping("/demo")
```

```
public class DemoController {

    @GetMapping("/hello")
    public String hello() {
        return "hello";
    }

    @PostMapping("/login")
    public String login(HttpServletRequest request) {
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        User user = new User(username, password, null);
//        user.setUsername(username);
//        user.setPassword(password);

        Course c1 = new Course("CSE351", "AJ", 3);

        request.setAttribute("user", user);
        request.setAttribute("login", true);

        return "home";
    }
}
```

```
package com.seu.hello_spring;
```

```
import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController

@RequestMapping("/demo-rest")

public class DemoRestController {

    @PostMapping("/student")

    public ResponseEntity<Student> student(@RequestBody Student student) {

        return ResponseEntity.internalServerError().body(student);

    }

}
```

```
package com.seu.hello_spring;

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

@SpringBootApplication

public class HelloSpringApplication {
```

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

```
package com.seu.hello_spring;  
  
import lombok.AllArgsConstructor;  
import lombok.Data;  
import lombok.NoArgsConstructor;  
  
@Data  
@NoArgsConstructor  
@AllArgsConstructor  
public class Response {  
    private String code;  
    private String msg;  
  
}
```

```
package com.seu.hello_spring;  
  
import org.springframework.boot.builder.SpringApplicationBuilder;  
import  
org.springframework.boot.web.servlet.support.SpringBootServletInitializer;
```



```
public class ServletInitializer extends SpringBootServletInitializer {

    @Override

    protected SpringApplicationBuilder configure(SpringApplicationBuilder
application) {

        return application.sources(HelloSpringApplication.class);

    }

}
```

```
package com.seu.hello_spring;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor

public class Student {

    private String id;

    private String name;

    private int batch;

}
```

```
package com.seu.hello_spring;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

import java.util.List;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class User {

    private String username;

    private String password;

    private List<Course> courseList;
}
```



```
h1{  
  
  background-color: black;  
  
  color: white;  
  
  text-align: center;  
}  
  
.main-heading{  
  
  background-color: cornflowerblue;  
}
```

```
#main-heading-id{  
  
    background-color: red;  
  
}  
  
#logo{  
  
    max-width: 100px;  
  
    max-height: 100px;  
  
}
```

```
spring.application.name=hello-spring  
  
server.port=8081  
  
spring.mvc.view.suffix=.jsp  
spring.mvc.view.prefix=/pages/
```

```
<%--  
  
    Created by IntelliJ IDEA.  
  
    User: iftekher  
  
    Date: 3/21/25  
  
    Time: 9:42 AM  
  
    To change this template use File | Settings | File Templates.  
  
--%>  
  
<%@ page contentType="text/html; charset=UTF-8" language="java" %>  
  
<html>  
  
<head>
```

```

<title>Title</title>

<link rel="stylesheet" href="/style.css">

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-SgOJa3DmI69IUzQ2PVdRZhwQ+dy64/BUTbMJw1MZ8t5HZApCHrRKUc4W0kG87
9m7" crossorigin="anonymous">
</head>

<body>

  <h1>Hello</h1>

  

<!-- <form action="/demo/login" method="POST">--%>
<!-- <input type="text" name="username"/><br/>--%>
<!-- <input type="password" name="password"/><br/>--%>
<!-- <input type="submit" value="Submit"/>--%>
<!-- </form>--%>

  <div class="container">

    <form action="/demo/login" method="POST">

      <div class="mb-3">

        <label for="exampleInputEmail1"
class="form-label">Username</label>

        <input type="text" class="form-control" id="exampleInputEmail1"
aria-describedby="emailHelp">

      </div>

      <div class="mb-3">

        <label for="exampleInputPassword1"
class="form-label">Password</label>

        <input type="password" class="form-control"
id="exampleInputPassword1">

      </div>

```

```

        <button type="submit" class="btn btn-primary">Submit</button>

    </form>

</div>

</body>

</html>

```

```

<%@ page import="java.io.PrintWriter" %><!--
    Created by IntelliJ IDEA.
    User: iftekher
    Date: 3/21/25
    Time: 10:06 AM
    To change this template use File | Settings | File Templates.
--%>

<%@ page contentType="text/html; charset=UTF-8" language="java" %>

<html>

<head>

    <title>Title</title>

    <link rel="stylesheet" href="/style.css">

</head>

<body>

    <h1 id="main-heading-id">Welcome ${user.username}</h1>

    <%

        boolean isLogin = (boolean) request.getAttribute("login");

        if(isLogin) {

    %>

```

```

        <h1>Login Success</h1>
    <%>
    } else {
    <%>

        <h1>Login Failed</h1>

    <%>
    }
    <%>
</body>
</html>

```

Session: 09 May 2025

Topics:

- Recap on API exposure
- Introduction to Spring data jpa
- Connecting MySQL from Spring boot
- Introduction to Repository and Service
- Using Spring data jpa

Codes:

```

package com.example.hello_rest.repos;

import com.example.hello_rest.User;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

@Repository
public interface UserRepo extends JpaRepository<User, Integer> {

    User findByUsername(String username);
}

```

```
}
```

```
package com.example.hello_rest.services;

import com.example.hello_rest.LoginRequest;
import com.example.hello_rest.Response;
import com.example.hello_rest.User;
import com.example.hello_rest.repos.UserRepo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class UserService {

    @Autowired
    private UserRepo userRepo;

    public List<User> readAll() {
        return userRepo.findAll();
    }

    public User readUser(String username) {
        return userRepo.findByUsername(username);
    }
}
```



```
public Response handleLogin(LoginRequest request) {

    User user = userRepo.findByUsername(request.getUsername());

    Response response = new Response();

    if(user != null && user.getPassword().equals(request.getPassword())) {

        response.setCode("0");

        response.setMsg("SUCCESS");

    } else {

        response.setCode("1");

        response.setMsg("Invalid Credentials");

    }

    return response;

}
```

```
package com.example.hello_rest;

import com.example.hello_rest.services.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

@RestController
@RequestMapping("/demo-rest")
public class DemoRestController {
```

```

    @Autowired

    private UserService userService;

    @GetMapping("/hello")

    public Response hello(@RequestParam("name") String name) {

        Response response = new Response();

        response.setCode("0");

        response.setMsg("Hello " + name);

        return response;

    }

    @PostMapping("/login")

    public Response login(@RequestBody LoginRequest request) {

        System.out.println("request --> " + request);

        Response response = userService.handleLogin(request);

        return response;

    }

}

```

```

package com.example.hello_rest;

```

```

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

```

```
@SpringBootApplication

public class HelloRestApplication {

    public static void main(String[] args) {

        SpringApplication.run(HelloRestApplication.class, args);

    }

}
```

```
package com.example.hello_rest;

import com.fasterxml.jackson.annotation.JsonProperty;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor

public class LoginRequest {

    private String username;

    private String password;

}
```

```
package com.example.hello_rest;
```

```
import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;

@Data

@NoArgsConstructor

@AllArgsConstructor

public class Response {

    private String code;

    private String msg;

}
```

```
package com.example.hello_rest;

import jakarta.persistence.*;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;

@Data

@NoArgsConstructor

@AllArgsConstructor

@Entity

@Table(name = "tbl_user")

public class User {
```

```
@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;


@Column(unique = true, nullable = false, length = 50)

private String username;

private String password;

}
```

```
spring.application.name=hello-rest

spring.datasource.url=jdbc:mysql://localhost:3306/cse351

spring.datasource.username=root

spring.datasource.password=root


spring.jpa.hibernate.ddl-auto=none

spring.jpa.show-sql=true
```

Session: 16 May 2025

Topics:

- Brief discussion on API Consuming
- API consuming using RestTemplate
- Limitations and Solutions of Singleton Instance

Codes:

hello-rest:

```
package com.example.hello_rest.repos;


import com.example.hello_rest.User;
```

```
import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import org.springframework.stereotype.Repository;

@Repository

public interface UserRepo extends JpaRepository<User, Integer> {

    User findByUsername(String username);

}
```

```
package com.example.hello_rest;

import com.example.hello_rest.services.UserService;

import lombok.Getter;

import lombok.Setter;

public class AppCache {

    @Getter

    @Setter

    private static UserService userService;

}
```

```
package com.example.hello_rest;
```

```
import com.example.hello_rest.services.UserService;

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

import org.springframework.web.bind.annotation.*;

@RestController

@RequestMapping("/demo-rest")

public class DemoRestController {

    @Autowired

    private UserService userService;

    @GetMapping("/hello")

    public Response hello(@RequestParam("name") String name) {

        Response response = new Response();

        response.setCode("0");

        response.setMsg("Hello " + name);

        return response;

    }

    @PostMapping("/login")

    public Response login(@RequestBody LoginRequest request) {

        System.out.println("request --> " + request);

        LoginHandler handler = new LoginHandler();

        handler.setRequest(request);

        Response response = handler.handleLogin();
```

```
        return response;
    }
}
```

```
package com.example.hello_rest;

import com.example.hello_rest.services.UserService;
import jakarta.annotation.PostConstruct;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class HelloRestApplication {

    @Autowired
    private UserService userService;

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

    @PostConstruct
    public void loadCache() {
        AppCache.setUserService(userService);
    }
}
```



```
}
```

```
package com.example.hello_rest;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class LoginHandler {

    private LoginRequest request;

    private Response response;

    public Response handleLogin() {

        User user = AppCache.getUserService().handleLogin(request);

        Response response = new Response();

        if(user != null && user.getPassword().equals(request.getPassword())) {

            response.setCode("0");

            response.setMsg("SUCCESS");

        } else {

            response.setCode("1");

            response.setMsg("Invalid Credentials");

        }

    }

}
```

```
    }

    return response;
}
}
```

```
package com.example.hello_rest;

import com.fasterxml.jackson.annotation.JsonProperty;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class LoginRequest {
    private String username;
    private String password;
}
```

```
package com.example.hello_rest;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
```

```
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Response {

    private String code;

    private String msg;

}
```

```
package com.example.hello_rest;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
@Entity
@Table(name = "tbl_user")
public class User {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private int id;
```

```
@Column(unique = true, nullable = false, length = 50)

private String username;

private String password;
}
```

```
spring.application.name=hello-rest

spring.datasource.url=jdbc:mysql://localhost:3306/cse351
spring.datasource.username=root
spring.datasource.password=root

spring.jpa.hibernate.ddl-auto=none
spring.jpa.show-sql=true
```

```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <parent>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-parent</artifactId>

        <version>3.4.5</version>

        <relativePath/> <!-- lookup parent from repository -->

    </parent>
```

```
<groupId>com.example</groupId>

<artifactId>hello-rest</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>hello-rest</name>

<description>Demo project for Spring Boot</description>

<url/>

<licenses>

    <license/>

</licenses>

<developers>

    <developer/>

</developers>

<scm>

    <connection/>

    <developerConnection/>

    <tag/>

    <url/>

</scm>

<properties>

    <java.version>21</java.version>

</properties>

<dependencies>

    <dependency>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-web</artifactId>

    </dependency>

    <dependency>
```

```

        <groupId>org.projectlombok</groupId>

        <artifactId>lombok</artifactId>

        <optional>true</optional>
    </dependency>

    <dependency>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-test</artifactId>

        <scope>test</scope>
    </dependency>

    <!--
https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter
-data-jpa -->

    <dependency>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>

    <!-- https://mvnrepository.com/artifact/com.mysql/mysql-connector-j -->

    <dependency>

        <groupId>com.mysql</groupId>

        <artifactId>mysql-connector-j</artifactId>

<!--         <version>9.3.0</version>-->

    </dependency>

</dependencies>

<build>

    <plugins>

        <!--<plugin>

```

```
<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<configuration>

    <annotationProcessorPaths>

        <path>

            <groupId>org.projectlombok</groupId>

            <artifactId>lombok</artifactId>

        </path>

    </annotationProcessorPaths>

</configuration>

</plugin>-->

<plugin>

    <groupId>org.springframework.boot</groupId>

    <artifactId>spring-boot-maven-plugin</artifactId>

    <configuration>

        <excludes>

            <exclude>

                <groupId>org.projectlombok</groupId>

                <artifactId>lombok</artifactId>

            </exclude>

        </excludes>

    </configuration>

</plugin>

</plugins>

</build>

</project>
```

hello-spring:

```
package com.seu.hello_spring;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.Getter;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class Course {

    private String courseCode;

    private String courseName;

    private int section;

    public Course(String courseCode) {

        this.courseCode = courseCode;

    }

}
```

```
package com.seu.hello_spring;

import jakarta.servlet.http.HttpServletRequest;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
```



```
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
@RequestMapping("/demo")
public class DemoController {

    @Autowired
    private LoginHandler loginHandler;

    @GetMapping("/hello")
    public String hello() {

        HealthCheckHandler healthCheckHandler = new HealthCheckHandler();
        healthCheckHandler.setName("ABC");
        Response response = healthCheckHandler.handleHealthCheck();

        System.out.println("response --> " + response);

        return "hello";
    }

    @PostMapping("/login")
    public String login(HttpServletRequest request) {

        String username = request.getParameter("username");
        String password = request.getParameter("password");
```

```
User user = new User(username, password, null);

System.out.println("user --> " + user);

//      loginHandler = new LoginHandler();
//      loginHandler.setUser(user);

Response response = loginHandler.handleLogin(user);

System.out.println("response --> " + response);

String page = "hello";

if(response.getCode().equals("0")) {
    page = "home";
}

/*Course c1 = new Course("CSE351", "AJ", 3);

request.setAttribute("user", user);
request.setAttribute("login", true);*/

return page;
}
}
```

```
package com.seu.hello_spring;

import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/demo-rest")
public class DemoRestController {

    @PostMapping("/student")
    public ResponseEntity<Student> student(@RequestBody Student student) {

        return ResponseEntity.internalServerError().body(student);
    }

}
```

```
package com.seu.hello_spring;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import org.springframework.web.client.RestTemplate;
```

```
@Data

@NoArgsConstructor

@AllArgsConstructor

public class HealthCheckHandler {

    private String name;

    private Response response;

    public Response handleHealthCheck() {

        String url = "http://localhost:8080/demo-rest/hello?name="+name;

        RestTemplate restTemplate = new RestTemplate();

        response = restTemplate.getForObject(url, Response.class);

        return response;

    }

}
```

```
package com.seu.hello_spring;

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

@SpringBootApplication

public class HelloSpringApplication {

    public static void main(String[] args) {

        SpringApplication.run(HelloSpringApplication.class, args);

    }

}
```

```
}  
  
}
```

```
package com.seu.hello_spring;  
  
import lombok.AllArgsConstructor;  
import lombok.Data;  
import lombok.NoArgsConstructor;  
import org.springframework.http.HttpEntity;  
import org.springframework.http.HttpMethod;  
import org.springframework.http.ResponseEntity;  
import org.springframework.stereotype.Component;  
import org.springframework.web.client.RestTemplate;  
  
@Component  
@Data  
@NoArgsConstructor  
//@AllArgsConstructor  
public class LoginHandler {  
  
    //    private User user;  
    //    private Response response;  
  
    public Response handleLogin(User user) {  
        RestTemplate restTemplate = new RestTemplate();  
        String url = "http://localhost:8080/demo-rest/login";  
    }  
}
```

```

        HttpEntity<User> httpEntity = new HttpEntity<>(user);

//        response = restTemplate.postForObject(url, user, Response.class);

        ResponseEntity<Response> responseEntity = restTemplate.exchange(url,
HttpMethod.POST, httpEntity, Response.class);

        Response response = responseEntity.getBody();

        return response;
    }
}

```

```

package com.seu.hello_spring;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor

public class Response {

    private String code;

    private String msg;

}

```

```
package com.seu.hello_spring;

import org.springframework.boot.builder.SpringApplicationBuilder;
import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;

public class ServletInitializer extends SpringBootServletInitializer {

    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder
application) {

        return application.sources(HelloSpringApplication.class);
    }

}
```

```
package com.seu.hello_spring;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor

public class Student {
```

```
    private String id;

    private String name;

    private int batch;
}
```

```
package com.seu.hello_spring;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import org.springframework.context.annotation.Scope;

import java.util.List;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class User {

    private String username;

    private String password;

    private List<Course> courseList;
}
```

```
spring.application.name=hello-spring

server.port=8081
```



```
spring.mvc.view.suffix=.jsp

spring.mvc.view.prefix=/pages/
```

```
<%--

Created by IntelliJ IDEA.

User: iftekher

Date: 3/21/25

Time: 9:42 AM

To change this template use File | Settings | File Templates.

--%>

<%@ page contentType="text/html; charset=UTF-8" language="java" %>

<html>

<head>

    <title>Title</title>

    <link rel="stylesheet" href="/style.css">

    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-SgOJa3DmI69IUzQ2PVdRZhwQ+dy64/BUtbMJw1MZ8t5HZApC4W0kG87
9m7" crossorigin="anonymous">

</head>

<body>

    <h1>Hello</h1>

    

    <form action="/demo/login" method="POST">--%>

        <input type="text" name="username"/><br/>--%>

        <input type="password" name="password"/><br/>--%>
```

```

<%--      <input type="submit" value="Submit"/>--%>

<%--      </form>--%>

    <div class="container">

        <form action="/demo/login" method="POST">

            <div class="mb-3">

                <label for="exampleInputEmail1"
class="form-label">Username</label>

                <input type="text" class="form-control" id="exampleInputEmail1"
aria-describedby="emailHelp" name="username">

            </div>

            <div class="mb-3">

                <label for="exampleInputPassword1"
class="form-label">Password</label>

                <input type="password" class="form-control"
id="exampleInputPassword1" name="password">

            </div>

            <button type="submit" class="btn btn-primary">Submit</button>

        </form>

    </div>

</body>

</html>

```

```

<%@ page import="java.io.PrintWriter" %><%--

Created by IntelliJ IDEA.

User: iftekher

Date: 3/21/25

```

Time: 10:06 AM

To change this template use File | Settings | File Templates.

--%>

<%@ **page** contentType="text/html; charset=UTF-8" language="java" %>

<html>

<head>

<title>Title</title>

<link rel="stylesheet" href="/style.css">

</head>

<body>

<h1 id="main-heading-id">Welcome **\${user.username}**</h1>

<h1>Login Success</h1>

<%--<%

boolean isLogin = (boolean) request.getAttribute("login");

if(isLogin) {

%>

<%

} else {

%>

<h1>Login Failed</h1>

<%

}

%>--%>

</body>

</html>

```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <parent>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-parent</artifactId>

        <version>3.4.3</version>

        <relativePath/> <!-- lookup parent from repository -->
    </parent>

    <groupId>com.seu</groupId>

    <artifactId>hello-spring</artifactId>

    <version>0.0.1-SNAPSHOT</version>

    <packaging>war</packaging>

    <name>hello-spring</name>

    <description>Demo project for Spring Boot</description>

    <url/>

    <licenses>

        <license/>

    </licenses>

    <developers>

        <developer/>

    </developers>

    <scm>

        <connection/>

        <developerConnection/>
```

```
</tag/>

<url/>

</scm>

<properties>
    <java.version>21</java.version>
</properties>

<dependencies>

    <dependency>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-web</artifactId>

    </dependency>

    <dependency>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-tomcat</artifactId>

        <scope>provided</scope>

    </dependency>

    <dependency>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-test</artifactId>

        <scope>test</scope>

    </dependency>

    <!--
https://mvnrepository.com/artifact/org.apache.tomcat.embed/tomcat-embed-jasper
-->

    <dependency>

        <groupId>org.apache.tomcat.embed</groupId>

        <artifactId>tomcat-embed-jasper</artifactId>
```

```
<!--           <version>11.0.5</version>-->

    </dependency>

    <!--
https://mvnrepository.com/artifact/jakarta.servlet/jakarta.servlet-api -->
    <dependency>

        <groupId>jakarta.servlet</groupId>

        <artifactId>jakarta.servlet-api</artifactId>

<!--           <version>6.1.0</version>-->

        <scope>provided</scope>
    </dependency>

    <dependency>

        <groupId>org.projectlombok</groupId>

        <artifactId>lombok</artifactId>

    </dependency>

</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>

</project>
```

Session: 23 May 2025

Topics:

- Understanding relationship categories between tables
- Implementing One to One relation
- Implementing One to Many relation
- Implementing Many to Many relation

Codes:

```
package com.example.hello_rest.entitites;

import com.fasterxml.jackson.annotation.JsonIgnore;
import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
@Entity
@Table(name = "tbl_course")
public class Course {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    private String code;
    private String name;

    @ManyToOne
    @JoinColumn(name = "student_id", referencedColumnName = "id")
    @JsonIgnore
    private Student student;
}
```

```
package com.example.hello_rest.entitites;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
```

```
import java.util.List;

@Data
@NoArgsConstructor
@AllArgsConstructor
@Entity
@Table(name = "tbl_student")
public class Student {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @OneToOne
    @JoinColumn(name = "user_id", referencedColumnName = "id")
    private User user;

    private String name;
    private int batch;
    private String gender;

    @OneToMany(mappedBy = "student")
    private List<Course> courses;
}
```

```
package com.example.hello_rest.entitites;

public class StudentCourseMap {
    private int id;
    private Student student;
    private Course course;
}
```



```

package com.example.hello_rest.entitites;

import jakarta.persistence.*;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
@Entity
@Table(name = "tbl_user")
public class User {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @Column(unique = true, nullable = false, length = 50)
    private String username;
    private String password;
}

```

```

package com.example.hello_rest.repos;

import com.example.hello_rest.entitites.Student;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Repository;

import java.util.Optional;

@Repository
public interface StudentRepo extends JpaRepository<Student, Integer> {

    Optional<Student> findByUserId(int userId);

    @Query(value = "select * from tbl_student where user_id=:userId",
nativeQuery = true)
    Optional<Student> getStudent(int userId);

    @Query(value = "select m.* from Student m where m.user.id=:userId")
    Optional<Student> getStudentAlt(int userId);
}

```

```

package com.example.hello_rest.repos;

import com.example.hello_rest.entitites.User;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface UserRepo extends JpaRepository<User, Integer> {

    User findByUsername(String username);
}

```

```

package com.example.hello_rest.services;

import com.example.hello_rest.entitites.Student;
import com.example.hello_rest.repos.StudentRepo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.Optional;

@Service
public class StudentService {

    @Autowired
    private StudentRepo studentRepo;

    public Student readStudent(int id) {

        Optional<Student> student = studentRepo.findById(id);

        if(student.isPresent()) {
            return student.get();
        }

        return null;
    }
}

```

```

package com.example.hello_rest.services;

import com.example.hello_rest.LoginRequest;
import com.example.hello_rest.entitites.User;
import com.example.hello_rest.repos.UserRepo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

```

```

import java.util.List;

@Service
public class UserService {

    @Autowired
    private UserRepo userRepo;

    public List<User> readAll() {
        return userRepo.findAll();
    }

    public User readUser(String username) {
        return userRepo.findByUsername(username);
    }

    public User handleLogin(LoginRequest request) {
        User user = userRepo.findByUsername(request.getUsername());

        return user;
    }
}

```

```

package com.example.hello_rest;

import com.example.hello_rest.services.UserService;
import lombok.Getter;
import lombok.Setter;

public class AppCache {

    @Getter
    @Setter
    private static UserService userService;

}

```

```

package com.example.hello_rest;

import com.example.hello_rest.entitites.Student;
import com.example.hello_rest.services.StudentService;
import com.example.hello_rest.services.UserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

```

```

@RestController
@RequestMapping("/demo-rest")
public class DemoRestController {

    @Autowired
    private UserService userService;

    @Autowired
    private StudentService studentService;

    @GetMapping("/hello")
    public Response hello(@RequestParam("name") String name) {
        Response response = new Response();
        response.setCode("0");
        response.setMsg("Hello " + name);
        return response;
    }

    @PostMapping("/login")
    public StudentResponse login(@RequestBody LoginRequest request) {
        System.out.println("request --> " + request);
        StudentResponse studentResponse = new StudentResponse();

        LoginHandler handler = new LoginHandler();
        handler.setRequest(request);

        Response response = handler.handleLogin();
        studentResponse.setCode(response.getCode());
        studentResponse.setMsg(response.getMsg());

        if(studentResponse.getCode().equals("0")) {
            Student student =
studentService.readStudent(response.getUser().getId());

            if(student == null) {
                studentResponse.setCode("1");
                studentResponse.setMsg("Student Not Available");
            } else {
                studentResponse.setStudent(student);
            }
        }

        return studentResponse;
    }
}

```

```

package com.example.hello_rest;

import com.example.hello_rest.services.UserService;
import jakarta.annotation.PostConstruct;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class HelloRestApplication {

    @Autowired
    private UserService userService;

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

    @PostConstruct
    public void loadCache() {
        AppCache.setUserService(userService);
    }
}

```

```

package com.example.hello_rest;

import com.example.hello_rest.entitites.User;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class LoginHandler {
    private LoginRequest request;
    private Response response;

    public Response handleLogin() {
        User user = AppCache.getUserService().handleLogin(request);

        Response response = new Response();

        if(user != null && user.getPassword().equals(request.getPassword())) {
            response.setCode("0");
            response.setMsg("SUCCESS");
            response.setUser(user);
        }
    }
}

```

```

        } else {
            response.setCode("1");
            response.setMsg("Invalid Credentials");
        }

        return response;
    }
}

```

```

package com.example.hello_rest;

import com.fasterxml.jackson.annotation.JsonProperty;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class LoginRequest {
    private String username;
    private String password;
}

```

```

package com.example.hello_rest;

import com.example.hello_rest.entitites.User;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class Response {

    private String code;
    private String msg;
    private User user;

}

```

```

package com.example.hello_rest;

import com.example.hello_rest.entitites.Student;
import lombok.AllArgsConstructor;

```

```

import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@AllArgsConstructor
public class StudentResponse extends Response {
    private Student student;
}

```

```

spring.application.name=hello-rest

spring.datasource.url=jdbc:mysql://localhost:3306/cse351
spring.datasource.username=root
spring.datasource.password=root

spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true

```

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.4.5</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.example</groupId>
  <artifactId>hello-rest</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>hello-rest</name>
  <description>Demo project for Spring Boot</description>
  <url/>
  <licenses>
    <license/>
  </licenses>
  <developers>
    <developer/>
  </developers>
  <scm>
    <connection/>
    <developerConnection/>
  </scm>
</project>

```

```

        <url/>
    </scm>
    <properties>
        <java.version>21</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
            <optional>true</optional>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>

        <!--
https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter
-data-jpa -->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>

        <!-- https://mvnrepository.com/artifact/com.mysql/mysql-connector-j -->
        <dependency>
            <groupId>com.mysql</groupId>
            <artifactId>mysql-connector-j</artifactId>
            <!--
                <version>9.3.0</version>-->
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <!--<plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <configuration>
                    <annotationProcessorPaths>
                        <path>
                            <groupId>org.projectlombok</groupId>
                            <artifactId>lombok</artifactId>
                        </path>
            </plugin>

```



```
        </annotationProcessorPaths>
    </configuration>
</plugin>-->
<plugin>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-maven-plugin</artifactId>
    <configuration>
        <excludes>
            <exclude>
                <groupId>org.projectlombok</groupId>
                <artifactId>lombok</artifactId>
            </exclude>
        </excludes>
    </configuration>
</plugin>
</plugins>
</build>

</project>
```