Assignment 5\_1 Java

991667681

Manmeet Singh Kohli

Balwinder Kaur

Step 1

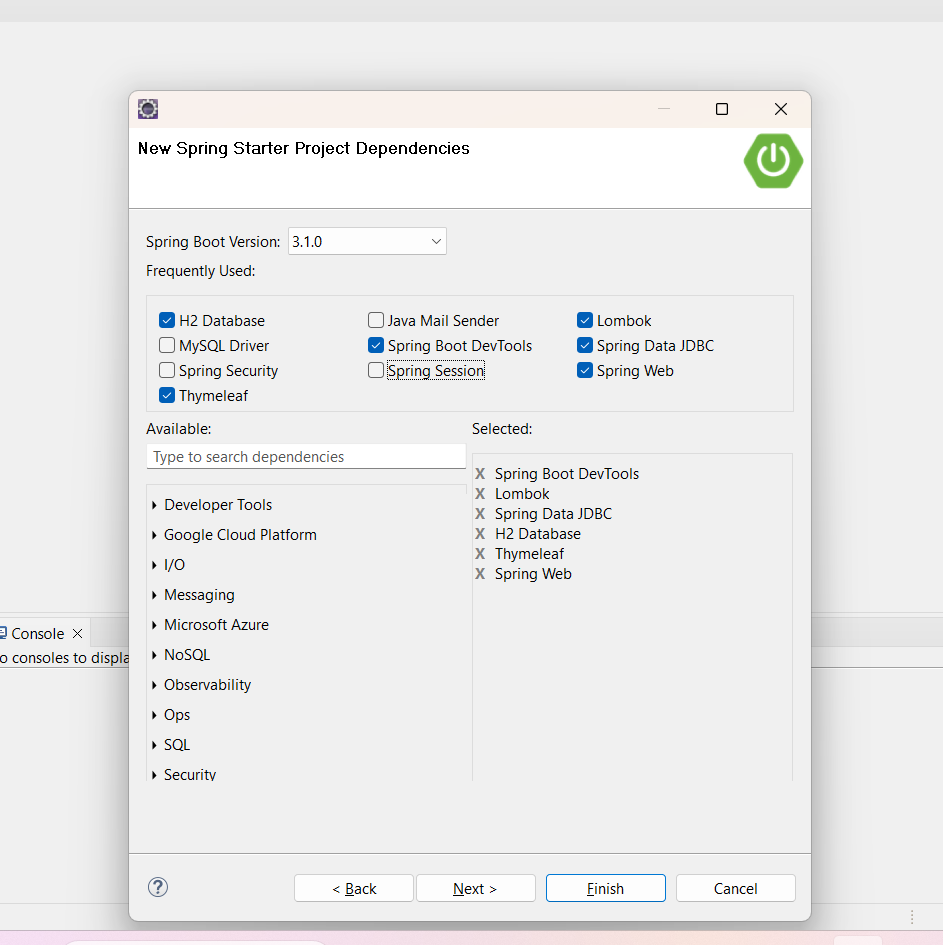
Click on new and than other

A screenshot of a computer

Description automatically generated

Make a project with name lec51\_H2andJDBC as shown

A screenshot of a computer

Description automatically generated  
  
  
Than we need to select those dependencies that we need in this project  
  
  
  
Than click on finish  
Than after that we are making pojo class or we can say a beans package class that includes all the getters and setters

A screenshot of a computer

Description automatically generated  
  
  
Than declare the private Long Id and private String name

A screenshot of a computer

Description automatically generated with medium confidence

Than add the other Lombok annotations and import the libraries as shown below  
  
A screenshot of a computer program

Description automatically generated with medium confidence  
  
  
  
Than make another Class StudentController within the same package src and main

A screenshot of a computer

Description automatically generated  
  
Than we added one method in this class mapping to / and returning the index page   
A screenshot of a computer program

Description automatically generated with low confidence

Than right click on the src/main/resources and than click on the other and select Html file

2A screenshot of a computer

Description automatically generated  
  
  
  
A screenshot of a computer

Description automatically generated with medium confidence  
  
  
  
  
Add this line to the html tag

A screenshot of a computer

Description automatically generated with low confidence  
  
Than add table and form as given in the lecture  
appA screenshot of a computer

Description automatically generated with medium confidence

Than we need to these two spring lines in the application properties

A screenshot of a computer

Description automatically generated with medium confidence  
  
  
  
You will get this error when you will enter the local host link of port 8080

A screenshot of a computer

Description automatically generated with medium confidence

Now change the path of the link and enter /h2-console at the end like this

ClickA screenshot of a computer

Description automatically generated  
  
Click on the connect button as everything default is correct

A screenshot of a computer

Description automatically generated

We don’t have any stored info yet so if we will run any query obvisously that will not run

So we need to create a table first in the schema.sql in the src/main/resources as shown

A screenshot of a computer

Description automatically generated with medium confidence”  
Now we will make a data.sql file as shown

A screenshot of a computer screen

Description automatically generated with medium confidence  
  
Let’s add some data in the data.sql file

A screenshot of a computer

Description automatically generated with medium confidence  
  
  
  
Now we need to make a another class with .database and name DatabaseAccess

A screenshot of a computer

Description automatically generated  
  
so we need to do some few steps as shown in the figure below

A screenshot of a computer program

Description automatically generated with low confidence  
  
  
  
Now add or we can call the insertStudent method from the database class

A screenshot of a computer

Description automatically generated with medium confidence

Now debug the projrct and run again and than run the local host again

Run the query select \* from students

A screenshot of a computer

Description automatically generated with medium confidence

Code for different classes

Beans Package

package ca.sheridancollege.kohliman.beans;

import lombok.Data;

import lombok.NoArgsConstructor;

@Data

@NoArgsConstructor

public class Student {

private long id;

private String name;

}

Controller Class

package ca.sheridancollege.kohliman.controllers;

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

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

import ca.sheridancollege.kohliman.database.DatabaseAccess;

public class StudentController {

@Autowired

private DatabaseAccess da;

@GetMapping("/")

public String index() {

da.insertStudentHardCoded();

return "index";

}

}  
  
  
  
package ca.sheridancollege.kohliman.database;

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

import org.springframework.jdbc.core.namedparam.MapSqlParameterSource;

import org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;

Database class

import org.springframework.stereotype.Repository;

@Repository

public class DatabaseAccess {

@Autowired

protected NamedParameterJdbcTemplate jdbc;

public void insertStudentHardCoded() {

MapSqlParameterSource namedParameters = new MapSqlParameterSource();

String query="INSERT INTO student(name) VALUES ('Frank')";

int rowsAffected = jdbc.update(query, namedParameters);

if (rowsAffected > 0)

System.out.println("Inserted student into database.");

}

}

Data.sql file

**INSERT** **INTO** student(name) **VALUES**

('Sally'),

('Harminder'),

('Hao'),

('Jaspreet');

Schema.sql File

**CREATE** **TABLE** student(

id LONG **PRIMARY** **KEY** AUTO\_INCREMENT,

name **VARCHAR**(255)

);

index.html page

<!DOCTYPE html>

<html xmlns:th=*"https://www.thymeleaf.org"*>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

</head>

<body>

<h1>This is index.html!</h1>

<table th:if=*"${studentList != null AND studentList != ''}"*>

<tr th:each=*"s : ${studentList}"*>

<td th:text=*"${s.name}"*>Frank</td>

</tr>

</table>

<hr>

<h1>This is index.html!</h1>

<table th:if=*"${studentList != null AND studentList != ''}"*>

<tr th:each=*"s : ${studentList}"*>

<td th:text=*"${s.name}"*>Frank</td>

</tr>

</table>

<hr>

<form method=*"post"* action=*""* th:action=*"@{/insertStudent}"*

th:object=*"${student}"*>

<input type=*"hidden"* name=*"id"* th:field=*"\*{id}"*/>

Name: <input type=*"text"* name=*"name"*

th:field=*"\*{name}"* /><br>

<input type=*"submit"* value=*"Go!"* />

</form>

</body>

</html>

Summary

So in this Project what we did in this is project, first we made a beans package and declared Long id and String name and added some Lombok annotations @data and @NoArgsConstructor with which all the getters and setters , toString() and hashcode() method are generated automatically .Than after that we made a Controller class in which we declared a method index mapping to (“/”) and returning a index page .Than we made a index page where we added this to html page so that it can connect to Thymleaf

<html xmlns:th=*"https://www.thymeleaf.org"*>

.Than we declared a table where we are putting a general entry in the table.

Than we made a data.sql and schema.sql file where we are inserting queries in the data.sql and in schema.sql where we creating table and inserting data into it.Than @AutoWired mean automatically entering the dependencies than we declared private DatabaseAccess da and in the index method we are calling the inserhardcodeCoded that is in are DatabaseAccess class. So when we run and write localhost:8080 this will not run but when we extend this to /h2-console this will give the h2-console login page and click on the connect button that will connect to the database by putting everything default and than when will run the query Select \* from Students , this will show the resulted table.