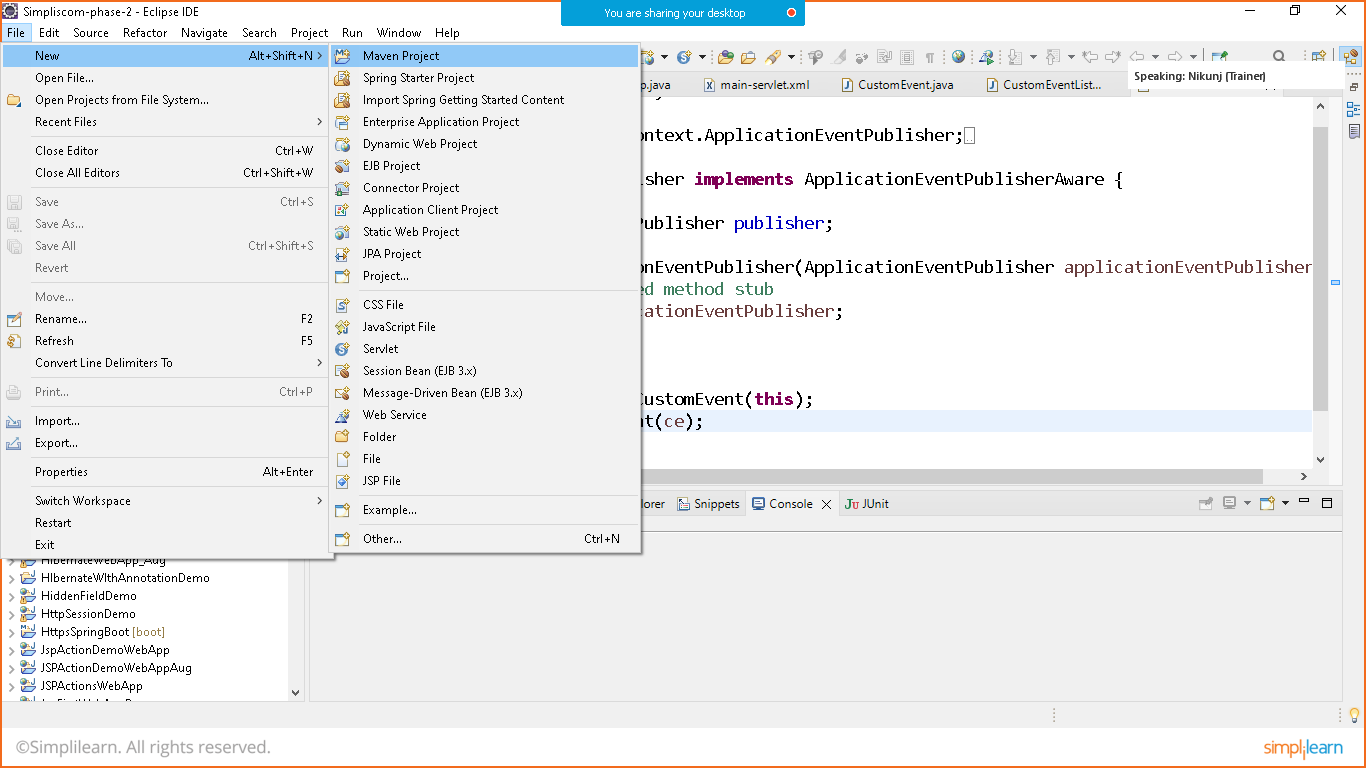
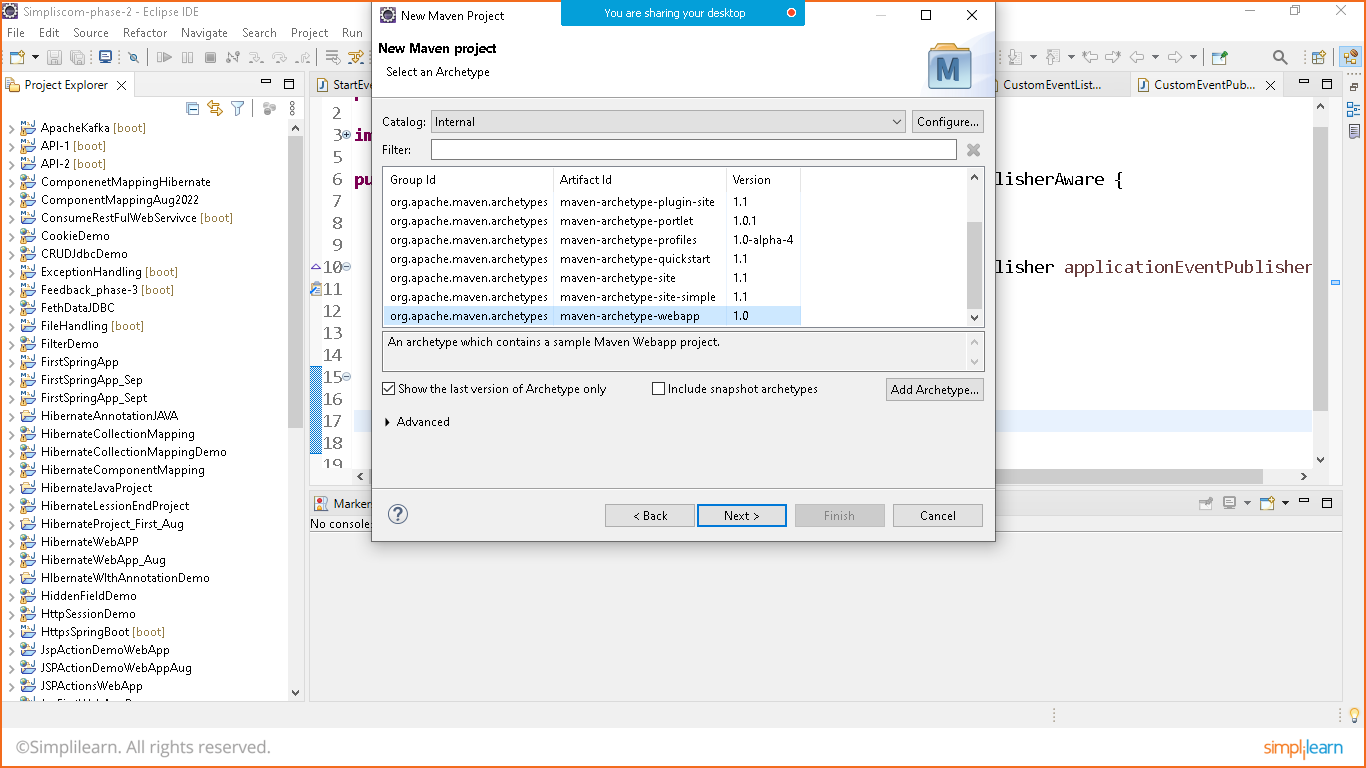
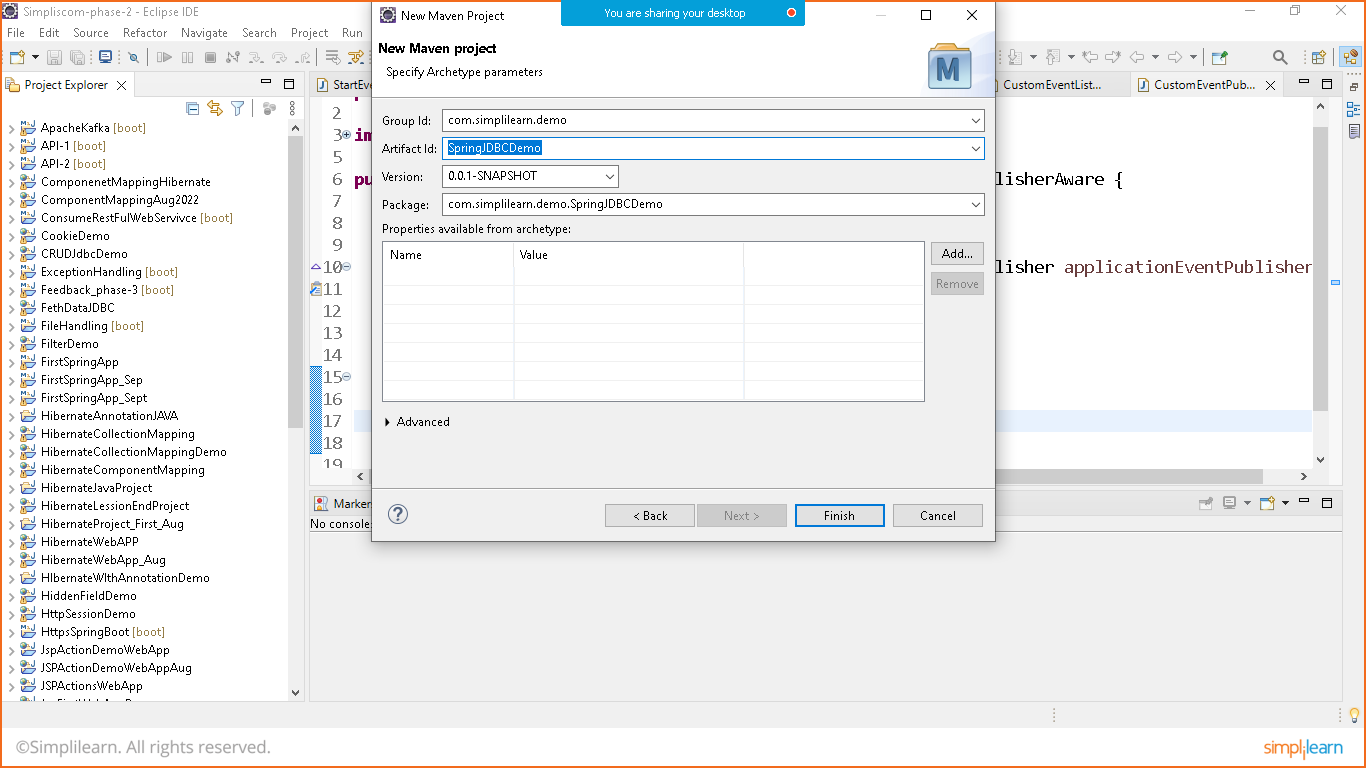
STEP-1 Prepare Maven Project>Choose webapp



give the name to the project



STEP:2 pom.xml

add below dependencies from maven repository

<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/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.simplilearn.demo</groupId>

<artifactId>SpringJDBCDemo</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>SpringJDBCDemo Maven Webapp</name>

<url>http://maven.apache.org</url>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<!--spring context-->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.22</version>

</dependency>

<!--Spring core-->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>5.3.22</version>

</dependency>

<!--Spring beans-->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-beans</artifactId>

<version>5.3.22</version>

</dependency>

<!--Spring web-->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>5.3.22</version>

</dependency>

<!--spring web mvc-->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.3.22</version>

</dependency>

<!--spring jdbc-->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>5.3.22</version>

</dependency>

<!--mysql connector-->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.30</version>

</dependency>

<!--java servlet api-->

<dependency>

<groupId>javax.servlet</groupId>

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

<version>4.0.1</version>

<scope>provided</scope>

</dependency>

<!--java sevlet JSTL-->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

</dependencies>

<build>

<finalName>SpringJDBCDemo</finalName>

<!--To remove error add below code-->

<plugins>

<plugin>

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

<artifactId>maven-war-plugin</artifactId>

<version>3.3.1</version>

</plugin>

<plugin>

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

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

<version>3.10.1</version>

</plugin>

</plugins>

<!--Maven Code ends here-->

</build>

</project>

STEP:3 Index.jsp

<html>

<body>

<h2>Hello World!</h2>

<a href=*"listproducts"*>List All Products</a>

</body>

</html>

STEP-4 Web.xml

<web-app>

<display-name>Archetype Created Web Application</display-name>

<servlet>

<servlet-name>test</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>test</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/test-servlet.xml</param-value>

</context-param>

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

</web-app>

STEP: 5 Test-Servlet.xml

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

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xsi:schemaLocation=*"*

*http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context.xsd"*>

<!-- to work with annotation -->

<context:annotation-config></context:annotation-config>

<!-- to scan the controller -->

<context:component-scan base-package=*"com.simplilearn.controller"*></context:component-scan>

<bean id=*"jspViewResolver"* class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<property name=*"suffix"* value=*".jsp"*></property>

<property name=*"prefix"* value=*"/WEB-INF/view/"*></property>

</bean>

<!-- Database -->

<bean id=*"dataSource"* class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>

<property name=*"driverClassName"* value=*"com.mysql.cj.jdbc.Driver"*></property>

<property name=*"url"* value=*"jdbc:mysql://localhost:3306/ecommerce"*></property>

<property name=*"username"* value=*"root"*></property>

<property name=*"password"* value=*"Nikunj@123"*></property>

</bean>

<bean id=*"jdbcTemplate"* class=*"org.springframework.jdbc.core.JdbcTemplate"*>

<property name=*"dataSource"* ref=*"dataSource"*></property>

</bean>

<bean id=*"dao"* class=*"com.simplilearn.dao.EProductDao"*></bean>

</beans>

Step:6 EProductEntity

package com.simplilearn.entity;

import java.math.BigDecimal;

import java.util.Date;

public class EProductEntity {

private long id;

private String name;

private BigDecimal price;

private Date dateAdded;

public EProductEntity() {

// TODO Auto-generated constructor stub

}

public EProductEntity(String name, BigDecimal price, Date dateAdded) {

this.name = name;

this.price = price;

this.dateAdded = dateAdded;

}

public long getId() {

return id;

}

public void setId(long id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public BigDecimal getPrice() {

return price;

}

public void setPrice(BigDecimal price) {

this.price = price;

}

public Date getDateAdded() {

return dateAdded;

}

public void setDateAdded(Date dateAdded) {

this.dateAdded = dateAdded;

}

@Override

public String toString() {

return "EProductEntity [id=" + id + ", name=" + name + ", price=" + price + ", dateAdded=" + dateAdded + "]";

}

}

Step: 7 EproductDao

package com.simplilearn.dao;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.List;

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

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.jdbc.core.RowMapper;

import org.springframework.stereotype.Repository;

import com.simplilearn.entity.EProductEntity;

@Repository

public class EProductDao {

@Autowired

JdbcTemplate jdbcTemplate;

public List<EProductEntity> getAllProducts(){

return jdbcTemplate.query("select \* from eproduct", new RowMapper<EProductEntity>() {

public EProductEntity mapRow(ResultSet rs,int rowNum)throws SQLException{

EProductEntity obj= new EProductEntity();

obj.setId(rs.getLong(1));

obj.setName(rs.getString(2));

obj.setPrice(rs.getBigDecimal(3));

obj.setDateAdded(rs.getTimestamp(4));

return obj;

}

});

}

}

Step:8 Main Controller

package com.simplilearn.controller;

import java.util.List;

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.context.ContextLoaderListener;

import org.springframework.web.servlet.DispatcherServlet;

import com.simplilearn.dao.EProductDao;

import com.simplilearn.entity.EProductEntity;

@Controller

public class MainController {

@Autowired

EProductDao dao;

@GetMapping("/listproducts")

public String getAllProducts(ModelMap map) {

List<EProductEntity> list= dao.getAllProducts();

map.addAttribute("list",list);

return "products";

}

}

Step: 9 Product.jsp

Goto> WEB-INF/view/products.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

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

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<h1>ECommerce Products</h1>

<table>

<tr><th>Id</th><th>Name</th><th>Price</th><th>Date</th></tr>

<c:forEach var=*"p"* items=*"*${list }*"*>

<tr>

<td>${p.id }</td>

<td>${p.name }</td>

<td>${p.price }</td>

<td>${p.dateAdded }</td>

</tr>

</c:forEach>

</table>

</body>

</html>

Save and run on server

