**MavenWrapperDownloader.java**

import java.net.\*;

import java.io.\*;

import java.nio.channels.\*;

import java.util.Properties;

public class MavenWrapperDownloader {

private static final String WRAPPER\_VERSION = "0.5.6";

private static final String DEFAULT\_DOWNLOAD\_URL = "https://repo.maven.apache.org/maven2/io/takari/maven-wrapper/"

+ WRAPPER\_VERSION + "/maven-wrapper-" + WRAPPER\_VERSION + ".jar";

private static final String MAVEN\_WRAPPER\_PROPERTIES\_PATH =

".mvn/wrapper/maven-wrapper.properties";

private static final String MAVEN\_WRAPPER\_JAR\_PATH =

".mvn/wrapper/maven-wrapper.jar";

private static final String PROPERTY\_NAME\_WRAPPER\_URL = "wrapperUrl";

public static void main(String args[]) {

System.out.println("- Downloader started");

File baseDirectory = new File(args[0]);

System.out.println("- Using base directory: " + baseDirectory.getAbsolutePath());

File mavenWrapperPropertyFile = new File(baseDirectory, MAVEN\_WRAPPER\_PROPERTIES\_PATH);

String url = DEFAULT\_DOWNLOAD\_URL;

if(mavenWrapperPropertyFile.exists()) {

FileInputStream mavenWrapperPropertyFileInputStream = null;

try {

mavenWrapperPropertyFileInputStream = new FileInputStream(mavenWrapperPropertyFile);

Properties mavenWrapperProperties = new Properties();

mavenWrapperProperties.load(mavenWrapperPropertyFileInputStream);

url = mavenWrapperProperties.getProperty(PROPERTY\_NAME\_WRAPPER\_URL, url);

} catch (IOException e) {

System.out.println("- ERROR loading '" + MAVEN\_WRAPPER\_PROPERTIES\_PATH + "'");

} finally {

try {

if(mavenWrapperPropertyFileInputStream != null) {

mavenWrapperPropertyFileInputStream.close();

}

} catch (IOException e) {

// Ignore ...

}

}

}

System.out.println("- Downloading from: " + url);

File outputFile = new File(baseDirectory.getAbsolutePath(), MAVEN\_WRAPPER\_JAR\_PATH);

if(!outputFile.getParentFile().exists()) {

if(!outputFile.getParentFile().mkdirs()) {

System.out.println(

"- ERROR creating output directory '" + outputFile.getParentFile().getAbsolutePath() + "'");

}}

System.out.println("- Downloading to: " + outputFile.getAbsolutePath());

try {

downloadFileFromURL(url, outputFile);

System.out.println("Done");

System.exit(0);

} catch (Throwable e) {

System.out.println("- Error downloading");

e.printStackTrace();

System.exit(1);

} }

private static void downloadFileFromURL(String urlString, File destination) throws Exception {

if (System.getenv("MVNW\_USERNAME") != null && System.getenv("MVNW\_PASSWORD") != null) {

String username = System.getenv("MVNW\_USERNAME");

char[] password = System.getenv("MVNW\_PASSWORD").toCharArray();

Authenticator.setDefault(new Authenticator() {

@Override

protected PasswordAuthentication getPasswordAuthentication() {

return new PasswordAuthentication(username, password);

}

});

}

URL website = new URL(urlString);

ReadableByteChannel rbc;

rbc = Channels.newChannel(website.openStream());

FileOutputStream fos = new FileOutputStream(destination);

fos.getChannel().transferFrom(rbc, 0, Long.MAX\_VALUE);

fos.close();

rbc.close();

}}

maven-wrapper.properties:

**maven-wrapper.properties:**

distributionUrl=https://repo.maven.apache.org/maven2/org/apache/maven/apache-maven/3.6.3/apache-maven-3.6.3-bin.zip

wrapperUrl=https://repo.maven.apache.org/maven2/io/takari/maven-wrapper/0.5.6/maven-wrapper-0.5.6.jar

**feedbackControlleer.java**

package com.controller;

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

import org.springframework.stereotype.Controller;

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

import com.service.AppService;

@Controller

public class FeedbackController {

@Autowired

private AppService appService;

@GetMapping("/feedback")

public String feedback() {

return "feedback";

}

}

**MyRestApp.java**

package com.controller;

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

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

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

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

import com.entity.Feedback;

import com.service.AppService;

@RestController

public class MyRestApp {

@Autowired

private AppService service;

@PostMapping("/feedback")

public String userRegister(@RequestParam("firstname") String firstname, @RequestParam("lastname") String lastname, @RequestParam("email") String email , @RequestParam("feedback1") String feedback1) {

Feedback f = new Feedback(firstname, lastname, email,feedback1);

boolean data= service.addFeedback(f);

if(data) {

return "Feedback added succesfully!";

}

else {

return "Unable to add the feedback";

}

}

}

**MyRepo.java**

package com.dao;

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

import com.entity.Feedback;

@Repository

public interface MyRepo extends CrudRepository<Feedback, Integer>{

}

**Feedback.java**

package com.entity;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity @Table(name="feedback")

public class Feedback {

@Override

public String toString() {

return "Feedback [id=" + id + ", firstname=" + firstname + ", lastname=" + lastname + ", email=" + email

+ ", feedback1=" + feedback1 + "]";

}

@Id @GeneratedValue

private int id;

private String firstname;

private String lastname;

private String email;

private String feedback1;

public Feedback() {

super();

// TODO Auto-generated constructor stub

}

public Feedback(String firstname, String lastname, String email ,String feedback1) {

super();

this.firstname = firstname;

this.lastname = lastname;

this.email= email;

this.feedback1 = feedback1;

}

}

**SpringCoreApplication.java**

package com.example.springcore;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.boot.autoconfigure.domain.EntityScan;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

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

@SpringBootApplication

@ComponentScan(basePackages="com")

@EntityScan(basePackages="com")

@EnableJpaRepositories(basePackages="com")

public class SpringcoreApplication {

public static void main(String[] args) {

// TODO Auto-generated method stub

SpringApplication.run(SpringcoreApplication.class, args);

}

}

**AppService.java**

package com.service;

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

import org.springframework.stereotype.Service;

import com.dao.MyRepo;

import com.entity.Feedback;

@Service

public class AppService {

@Autowired

private MyRepo myRepo;

public boolean addFeedback( Feedback f) {

myRepo.save(f);

return true;

}

}

**application.properties**

spring.mvc.view.prefix=/WEB-INF/pages/

spring.mvc.view.suffix=.jsp

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

spring.datasource.username=root

spring.datasource.password=Lovely1811@

spring.jpa.hibernate.ddl-auto=create

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect

spring.jpa.properties.hibernate.id.new\_generator\_mappings = false

spring.jpa.properties.hibernate.format\_sql = true

**feedback.java**

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

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

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

<title>SpringBoot Lesson End Project</title>

<style>

input[type=text], input[type=email],textarea {

width: 100%;

padding: 12px;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

margin-top: 6px;

margin-bottom: 16px;

resize: vertical

}

input[type=submit] {

background-color: #4CAF50;

color: white;

padding: 12px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

}

input[type=submit]:hover {

background-color: #45a049;

}

.container {

border-radius: 5px;

background-color: #f2f2f2;

padding: 20px;

}

</style>

</head>

<body>

<div class="container">

<form method="post" action="/feedback">

<label for="firstname">First Name</label>

<input type="text" id="firstname" name="firstname" placeholder="Your first name" required>

<br>

<label for="lastname">Last Name</label>

<input type="text" id="lastname" name="lastname" placeholder="Your last name" required>

<br>

<label for="email">Email Id</label>

<input type="email" id="email" name="email" placeholder="Your Email Address" required>

<br>

<label for="feedback1">Feedback</label>

<textarea id="feedback1" name="feedback1" placeholder="Write your feedback here..." style="height:200px" required></textarea>

<br>

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

</form>

</div>

</body>

</html>

**SpringCoreApplicationTest.java**

package com.example.springcore;

import org.junit.jupiter.api.Test;

import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest

class SpringcoreApplicationTests {

@Test

void contextLoads() {

}

}

**Pom.xml**

<?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>2.3.1.RELEASE</version>

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

</parent>

<groupId>com.example</groupId>

<artifactId>springcore</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>springcore</name>

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

<properties>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

<groupId>mysql</groupId>

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

</dependency>

<dependency>

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

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

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

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

<scope>test</scope>

<exclusions>

<exclusion>

<groupId>org.junit.vintage</groupId>

<artifactId>junit-vintage-engine</artifactId>

</exclusion>

</exclusions>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

</plugins>

</build>

</project>