Displaying User Feedback

DESCRIPTION

Project objective:

Create a Spring Boot project that will capture user feedback using a REST endpoint. The REST resource will take in parameters using HTTP POST. The feedback data will be then added to a database table.

Background of the problem statement:

As a part of developing an ecommerce web application, a REST resource is needed to capture user feedback. Feedback data will be received from third-party apps and websites. The data will be sent to the REST API which will collect feedback from various sources.

Source Code:

com.controller
FeedbackController.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";
	}

	}
}
```

Com.dao
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>{

}
```

Com.entity
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;
    }


}
```

Com.example.springcore

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);

    }

}
```

Com.service
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;
    }
}
```

applicationProperties.txt

```
spring.mvc.view.prefix=/WEB-INF/pages/
spring.mvc.view.suffix=.jsp


spring.datasource.url=jdbc:mysql://localhost:3306/feedback
spring.datasource.username=root
spring.datasource.password=ramani44
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.jsp

```
<%@ 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 Feedback</title>
<style>
/* Style inputs with type="text", select elements and textareas */
input[type=text], input[type=email],textarea {
  width: 100%; /* Full width */
  padding: 12px; /* Some padding */
  border: 1px solid #ccc; /* Gray border */
  border-radius: 4px; /* Rounded borders */
  box-sizing: border-box; /* Make sure that padding and width stays in place */
  margin-top: 6px; /* Add a top margin */
  margin-bottom: 16px; /* Bottom margin */
  resize: vertical /* Allow the user to vertically resize the textarea (not horizontally) */
}

/* Style the submit button with a specific background color etc */
input[type=submit] {
  background-color: #4CAF50;
  color: white;
  padding: 12px 20px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

/* When moving the mouse over the submit button, add a darker green color */
input[type=submit]:hover {
  background-color: #45a049;
}

/* Add a background color and some padding around the form */
.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>
```

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>
```

Pushing the code to your GitHub repositories:

- Open your command prompt and navigate to the folder where you have created your files.

  ```
  Cd<folder path>
  ```

- Initialize your repository using the following command:

  ```
  git init
  ```

- Add all the files to your git repository using the following command:

  ```
  git add .
  ```

- Commit the changes using the following command:

  ```
  git commit . -m “Changes have been committed.”
  ```

- Push the files to the folder you initially created using the following command:

  ```
  git push -u origin master
  ```