

Displaying User Feedback

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";
        }
    }
}
```

```
}
```

```
}
```

```
}
```

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;
```

```
}
```

```
}
```

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);
    }

```

```

}

```

FeedbackApplicationTests.java

```

package com.project.Feedback;

```

```

import static org.junit.jupiter.api.Assertions.assertEquals;

```

```

import javax.persistence.EntityManager;

```

```

import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;

```

```

import com.project.Feedback.entities.Feedback;
import com.project.Feedback.repositories.FeedbackRepository;

```

```

@SpringBootTest
class FeedbackApplicationTests {

```

```

    @Autowired
    EntityManagerentityManager;

```

```

        @Autowired
        FeedbackRepository feedbackRepo;

//        @Autowired

        @Test
        void shouldFindByUser() {
            Feedback testFeedback = new Feedback("Dummy Test", 5, "dummy");
            entityManager.persist(testFeedback);
            entityManager.flush();
            Feedback cmp = feedbackRepo.findByUser(testFeedback.getUser());
            assertEquals(cmp.getUser(), testFeedback.getUser());
        }
}

```

FeedbackController.java

```

package com.project.Feedback.controllers;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.MediaType;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.bind.annotation.RestController;

import com.project.Feedback.entities.Feedback;
import com.project.Feedback.services.FeedbackService;

@RestController
public class FeedbackController {

    @Autowired
    FeedbackService feedbackService;

    @GetMapping("/feedback")
    public Iterable<Feedback> getAllFeedbacks(){
        return feedbackService.GetAllFeedback();
    }

    @PostMapping(path="/feedback", consumes= {MediaType.APPLICATION_JSON_VALUE})
    public Feedback addNewFeedback(@RequestBody Feedback fb) {
        Feedback newFb = new Feedback(fb.getComments(), fb.getRating(), fb.getUser());
        feedbackService.addNewFeedback(newFb);
        return newFb;
    }
}

```

TestFormController.java

```
package com.project.Feedback.controllers;

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

import com.project.Feedback.entities.Feedback;
import com.project.Feedback.services.FeedbackService;

@Controller
public class TestFormController {

    @Autowired
    FeedbackService feedbackService;

    @GetMapping("/test_form")
    public String showTestForm(ModelMap model) {
        model.addAttribute("test", new Feedback());
        return "testformjsp";
    }

    @PostMapping("/test_form")
    public String submitTestForm(@ModelAttribute("testUser") Feedback fb, ModelMap m) {
        feedbackService.addNewFeedback(fb);
        m.addAttribute("test", fb);
        return "post";
    }

    // TODO: Implement form submission
    // TODO: call RestTemplate and make json request to localhost.../feedback

}

//RestTemplaterestTemplate = new RestTemplate();
//URL testForm = new URL("http://localhost:8090/feedback/{feedback}");
//ResponseEntity<String> response = restTemplate.getForEntity(testForm + "/7", String.class);
//ObjectMapper mapper = new ObjectMapper();
//JsonNode root = mapper.readTree(response.getBody());
//JsonNode name = root.path("name");
//model.addAttribute(name);
//String result = restTemplate.getForObject("http://localhost:8090/feedback/{feedback}",
String.class, 7);
```

Feedback.java

```
package com.project.Feedback.entities;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.validation.constraints.NotNull;

import lombok.Data;

@Entity
@Data
public class Feedback {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="id")
    @NotNull
    private Integer id;

    @Column(name="comments")
    private String comments;

    @Column(name="rating")
    @NotNull
    private int rating;

    @Column(name="user")
    private String user;

    public Feedback() {
        super();
    }

    public Feedback(String comments, Integer rating, String user) {
        this.comments = comments;
        this.rating = rating;
        this.user = user;
    }

    /*
     * Needed the setters and getters to be able to add name and comments otherwise
     * they are nulls when entering the SQL DB
     */
    public String getComments() {
        return comments;
    }

    public void setComments(String comments) {
        this.comments = comments;
    }
}
```

```

    public Integer getRating() {
        return rating;
    }

    public void setRating(Integer rating) {
        this.rating = rating;
    }

    public String getUser() {
        return user;
    }

    public void setUser(String user) {
        this.user = user;
    }

    @Override
    public String toString() {
        return "Feedback [id=" + id + ", comments=" + comments + ", rating=" + rating + ",
user=" + user + "]\n";
    }
}

```

FeedbackRepository.java

```

package com.project.Feedback.repositories;

import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;

import com.project.Feedback.entities.Feedback;

@Repository
public interface FeedbackRepository extends CrudRepository<Feedback, Integer> {

    public Feedback findByUser(String feedback);
}

```

FeedbackService.java

```

package com.project.Feedback.services;

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

import com.project.Feedback.entities.Feedback;
import com.project.Feedback.repositories.FeedbackRepository;

@Service
public class FeedbackService {

    @Autowired
    FeedbackRepository feedbackRepo;
}

```

```

        public Iterable<Feedback>GetAllFeedback() {
            return feedbackRepo.findAll();
        }

        public Feedback addNewFeedback(Feedback fb) {
            return feedbackRepo.save(fb);
        }
    }

```

Index.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>Welcome Page</title>
</head>
<h2>Landing Page</h2>
<body>
<a href="test_form">Test Form</a><br/><br/>
<a href="feedback">See all Feedbacks</a><br/><br/>
</body>
</html>

```

Post.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>Post test</title>
</head>
<body>
Successfully added: ${testUser.toString()}
</body>
</html>

```

Testform.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>Spring test App</title>
</head>
<body>
<form:form action="/test_form" method="post" commandName="testUser">
<label for="user">User:</label><br>

```



```
<input type="text" id="user" name="user" placeholder="Teju"><br>
<label for="comments">Comments:</label><br>
<input type="text" id="comments" name="comments" placeholder="Tejasri"><br><br>
<input type="submit" value="Submit">
<label for="rating">Rating:</label><br>

<input type="range" name="rating" id="rating" min="0" max="10" value="5" class="slider">
</form:form>
</body>
</html>
```