**Model**

package com.feedback.model;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

import javax.validation.constraints.Max;

import javax.validation.constraints.Min;

*@Entity*

*@Table*

public class Feedback {

*@Id*

*@GeneratedValue*(strategy = *GenerationType*.***IDENTITY***)

private int id;

private String feedbackForAuthor;

*@Min*(0)

*@Max*(10)

private int starsOverall;

*@Min*(0)

*@Max*(10)

private int starsAuthor;

private String feedbackOverall;

private String bookName;

private String extraComments;

public Feedback() {

super();

}

public Feedback(int id, String feedbackForAuthor, *@Min*(0) *@Max*(10) int starsOverall,

*@Min*(0) *@Max*(10) int starsAuthor, String feedbackOverall, String bookName, String extraComments) {

super();

this.id = id;

this.feedbackForAuthor = feedbackForAuthor;

this.starsOverall = starsOverall;

this.starsAuthor = starsAuthor;

this.feedbackOverall = feedbackOverall;

this.bookName = bookName;

this.extraComments = extraComments;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getFeedbackForAuthor() {

return feedbackForAuthor;

}

public void setFeedbackForAuthor(String feedbackForAuthor) {

this.feedbackForAuthor = feedbackForAuthor;

}

public int getStarsOverall() {

return starsOverall;

}

public void setStarsOverall(int starsOverall) {

this.starsOverall = starsOverall;

}

public int getStarsAuthor() {

return starsAuthor;

}

public void setStarsAuthor(int starsAuthor) {

this.starsAuthor = starsAuthor;

}

public String getFeedbackOverall() {

return feedbackOverall;

}

public void setFeedbackOverall(String feedbackOverall) {

this.feedbackOverall = feedbackOverall;

}

public String getBookName() {

return bookName;

}

public void setBookName(String bookName) {

this.bookName = bookName;

}

public String getExtraComments() {

return extraComments;

}

public void setExtraComments(String extraComments) {

this.extraComments = extraComments;

}

}

**Controller**

package com.feedback.controller;

import java.util.List;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.util.LinkedMultiValueMap;

import org.springframework.util.MultiValueMap;

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

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

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

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

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

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

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

import com.feedback.exception.BusinessException;

import com.feedback.model.Feedback;

import com.feedback.service.FeedbackService;

@RestController

public class FeedbackController {

@Autowired

private FeedbackService service;

private MultiValueMap<String, String> map;

@PostMapping("/feedback")

public Feedback createFeedback(@RequestBody Feedback Feedback) {

return service.createFeedback(Feedback);

}

@GetMapping("/feedback/{id}")

public ResponseEntity<Feedback> getFeedbackById(@PathVariable("id") int id) {

try {

return new ResponseEntity<Feedback>(service.getFeedbackById(id), HttpStatus.OK);

} catch (BusinessException e) {

map = new LinkedMultiValueMap<>();

map.add("message", e.getMessage());

return new ResponseEntity<Feedback>(null, map, HttpStatus.NOT\_FOUND);

}

}

@PutMapping("/feedback")

public Feedback updateFeedback(@RequestBody Feedback Feedback) {

return service.updateFeedback(Feedback);

}

@DeleteMapping("/feedback/{id}")

public ResponseEntity<Feedback> deleteFeedback(@PathVariable("id") int id) {

try {

service.deleteFeedback(id);

return new ResponseEntity<Feedback>(HttpStatus.OK);

} catch (BusinessException e) {

map = new LinkedMultiValueMap<>();

map.add("message", e.getMessage());

return new ResponseEntity<Feedback>(null, map, HttpStatus.NOT\_FOUND);

}

}

@GetMapping("/feedbacks")

public List<Feedback> getAllFeedbacks() {

return service.getAllFeedbacks();

}

}

**Exception**

package com.feedback.exception;

public class BusinessException extends Exception {

public BusinessException() {

super();

}

public BusinessException(final String message) {

super(message);

}

**Main**

}

package com.feedback;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.Bean;

import springfox.documentation.builders.RequestHandlerSelectors;

import springfox.documentation.spi.DocumentationType;

import springfox.documentation.spring.web.plugins.Docket;

import springfox.documentation.swagger2.annotations.EnableSwagger2;

@SpringBootApplication

@EnableSwagger2

public class FeedbackApplication {

public static void main(String[] args) {

SpringApplication.run(FeedbackApplication.class, args);

}

@Bean

public Docket productApi() {

return new Docket(DocumentationType.SWAGGER\_2).select().apis(RequestHandlerSelectors.basePackage("com.feedback"))

.build();

}

**DAO**

package com.feedback.dao;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

import com.feedback.model.Feedback;

*@Repository*

public interface FeedbackDAO extends JpaRepository<Feedback, Integer> {

public List<Feedback> findBystarsOverall(int age);

}

**Service**

package com.feedback.service;

import java.util.List;

import com.feedback.exception.BusinessException;

import com.feedback.model.Feedback;

public interface FeedbackService {

public Feedback createFeedback(Feedback Feedback);

public Feedback getFeedbackById(int id) throws BusinessException;

public Feedback updateFeedback(Feedback Feedback);

public void deleteFeedback(int id) throws BusinessException;

public List<Feedback> getAllFeedbacks();

}

**Service Implementation**

package com.feedback.service.impl;

import java.util.List;

import java.util.NoSuchElementException;

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

import org.springframework.dao.EmptyResultDataAccessException;

import org.springframework.stereotype.Service;

import com.feedback.exception.BusinessException;

import com.feedback.dao.FeedbackDAO;

import com.feedback.model.Feedback;

import com.feedback.service.FeedbackService;

@Service

public class FeedbackServiceImpl implements FeedbackService {

@Autowired

private FeedbackDAO dao;

@Override

public Feedback createFeedback(Feedback Feedback) {

return dao.save(Feedback);

}

@Override

public Feedback getFeedbackById(int id) throws BusinessException {

Feedback feedback = null;

try {

feedback = dao.findById(id).get();

}catch (NoSuchElementException e) {

throw new BusinessException("No feedback found for id " + id);

}

return feedback;

}

@Override

public Feedback updateFeedback(Feedback Feedback) {

return dao.save(Feedback);

}

@Override

public void deleteFeedback(int id) throws BusinessException {

try {

dao.deleteById(id);

}catch (EmptyResultDataAccessException e) {

throw new BusinessException("No feedback found for id " + id);

}

}

@Override

public List<Feedback> getAllFeedbacks() {

return dao.findAll();

}

}