Todo.java

import java.util.Date;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

@Entity

public class Todo {

@Id

@GeneratedValue

private Long id;

private String username;

private String description;

private Date targetDate;

private boolean isDone;

protected Todo() {

}

public Todo(long id, String username, String description, Date targetDate, boolean isDone) {

super();

this.id = id;

this.username = username;

this.description = description;

this.targetDate = targetDate;

this.isDone = isDone;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public Date getTargetDate() {

return targetDate;

}

public void setTargetDate(Date targetDate) {

this.targetDate = targetDate;

}

public boolean isDone() {

return isDone;

}

public void setDone(boolean isDone) {

this.isDone = isDone;

}

@Override

public int hashCode() {

final int prime = 31;

int result = 1;

result = prime \* result + (int) (id ^ (id >>> 32));

return result;

}

@Override

public boolean equals(Object obj) {

if (this == obj)

return true;

if (obj == null)

return false;

if (getClass() != obj.getClass())

return false;

Todo other = (Todo) obj;

if (id != other.id)

return false;

return true;

}

}

TodoJpaRepository

import java.util.List;

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

import org.springframework.stereotype.Repository;

@Repository

public interface TodoJpaRepository extends JpaRepository<Todo, Long>{

List<Todo> findByUsername(String username);

}

TodoJpaResource

**import** java.net.URI;

**import** java.util.List;

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

**import** org.springframework.http.HttpStatus;

**import** org.springframework.http.ResponseEntity;

**import** org.springframework.web.bind.annotation.CrossOrigin;

**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** org.springframework.web.servlet.support.ServletUriComponentsBuilder;

**import** com.in28minutes.rest.webservices.restfulwebservices.todo.Todo;

@CrossOrigin(origins="http://localhost:4200")

@RestController

**public** **class** TodoJpaResource {

/\*@Autowired

private TodoHardcodedService todoService;\*/

@Autowired

**private** TodoJpaRepository todoJpaRepository;

/\*@Autowired

private FeedbackJpaRepo feedbackRepo;\*/

@GetMapping("/jpa/users/{username}/todos")

**public** List<Todo> getAllTodos(@PathVariable String username){

**return** todoJpaRepository.findByUsername(username);

//return todoService.findAll();

}

@GetMapping("/jpa/users/{username}/todos/{id}")

**public** Todo getTodo(@PathVariable String username, @PathVariable **long** id){

**return** todoJpaRepository.findById(id).get();

//return todoService.findById(id);

}

//DELETE /users/{username}/todos/{id}

@DeleteMapping("/jpa/users/{username}/todos/{id}")

**public** ResponseEntity<Void> deleteTodo(

@PathVariable String username, @PathVariable **long** id){

//Todo todo = todoService.deleteById(id);

todoJpaRepository.deleteById(id);

**return** ResponseEntity.*noContent*().build();

//return ResponseEntity.notFound().build();

}

//Edit/Update a Todo

//PUT /users/{user\_name}/todos/{todo\_id}

@PutMapping("/jpa/users/{username}/todos/{id}")

**public** ResponseEntity<Todo> updateTodo(

@PathVariable String username,

@PathVariable **long** id, @RequestBody Todo todo){

//Todo todoUpdated = todoService.save(todo);

Todo todoUpdated = todoJpaRepository.save(todo);

**return** **new** ResponseEntity<Todo>(todo, HttpStatus.***OK***);

}

@PostMapping("/jpa/users/{username}/todos")

**public** ResponseEntity<Void> createTodo(

@PathVariable String username, @RequestBody Todo todo){

//Todo createdTodo = todoService.save(todo);

todo.setUsername(username);

Todo createdTodo = todoJpaRepository.save(todo);

//Location

//Get current resource url

///{id}

URI uri = ServletUriComponentsBuilder.*fromCurrentRequest*()

.path("/{id}").buildAndExpand(createdTodo.getId()).toUri();

**return** ResponseEntity.*created*(uri).build();

}

/\*@PostMapping("jpa/users/feedback")

public ResponseEntity<Void> createFeedBack(@RequestBody Feedback feedback){

//Todo createdTodo = todoService.save(todo);

// todo.setUsername(username);

Feedback createdTodo =feedbackRepo.save(feedback);

//Location

//Get current resource url

///{id}

URI uri = ServletUriComponentsBuilder.fromCurrentRequest()

.path("/{id}").buildAndExpand(createdTodo.getId()).toUri();

return ResponseEntity.created(uri).build();

}\*/

}