

**The course “Application development with Java”**

Practical assignment 4

Student name: Roberts

Student surname: Dubovskis

St. code: 62085

**RIGA**

**2022**

Contents

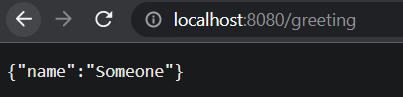
[Exercise 1 3](#_Toc101104728)

[Exercise 2 5](#_Toc101104729)

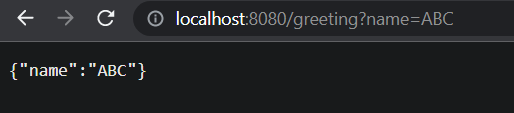
[Exercise 3 7](#_Toc101104730)

[Conclusion 12](#_Toc101104731)

# Exercise 1



Img.1 (result)



Img.2 (result)

**HelloApplication.java**

package lv.tsi.hello;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class HelloApplication {

public static void main(String[] args) {

SpringApplication.run(HelloApplication.class, args);

}

}

**GreetingController.java**

package lv.tsi.hello;

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

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

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

@RestController

public class GreetingController {

@GetMapping("/greeting")

public Greeting greeting(@RequestParam(value = "name", defaultValue = "Someone") String name) {

return new Greeting(name);

}

}

**Greeting.java**

package lv.tsi.hello;

public class Greeting {

String name;

public Greeting(String name) {

this.name = name;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

# Exercise 2

**BookApplication.java**

package lv.tsi.book;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class BookApplication {

public static void main(String[] args) {

SpringApplication.run(BookApplication.class, args);

}

}

**BookRepository.java**

package lv.tsi.book.Repository;

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

@Repository

public interface BookRepository extends CrudRepository<Book,Long> {

}

**Book.java**

package lv.tsi.book.model;

import com.fasterxml.jackson.annotation.JsonProperty;

import javax.persistence.Entity;

import javax.persistence.Id;

@Entity

public class Book {

@JsonProperty

@Id

Long id;

@JsonProperty

String author;

@JsonProperty

String name;

@JsonProperty

int copies;

}

**BookController.java**

package lv.tsi.book.Controler;

import lv.tsi.book.Repository.BookRepository;

import lv.tsi.book.model.Book;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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

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

@RestController

public class BookController {

private final BookRepository bookRepository;

@Autowired

public BookController(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

/\* @GetMapping("/getBook")

public ResponseEntity<Book> findBook(int id) {

return ResponseEntity.of(bookRepository.findById(id), HttpStatus.OK);

}

@GetMapping("/getBooks")

public ResponseEntity<List<Books>> findBooks() {

return new ResponseEntity<List<Books>>(bookRepository.findAll(), HttpStatus.OK);

}

\*/

}

# Exercise 3

**UniversityApplication.java**

package lv.tsi.university;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class UniversityApplication {

public static void main(String[] args) {

SpringApplication.run(UniversityApplication.class, args);

}

}

**UniversityRepository.java**

package lv.tsi.university.Repository;

import lv.tsi.university.Model.University;

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

@Repository

public interface UniversityRepository extends CrudRepository<University,Long> {

}

**UniversityService.java**

package lv.tsi.university.Service;

import lombok.RequiredArgsConstructor;

import lv.tsi.university.Model.University;

import lv.tsi.university.Repository.UniversityRepository;

import org.springframework.http.ResponseEntity;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

@RequiredArgsConstructor

public class UniversityService {

private final UniversityRepository universityRepository;

public ResponseEntity<List<University>> getUniversities() {

return (ResponseEntity<List<University>>) universityRepository.findAll();

}

}

package lv.tsi.university.Repository;

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

@Repository

public interface UniversityRepository extends CrudRepository {

}

**University.java**

package lv.tsi.university.Model;

import com.fasterxml.jackson.annotation.JsonProperty;

import lombok.Data;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.Table;

@Data

@Entity

@Table(name = "University")

public class University {

@JsonProperty

@Id

Long id;

@JsonProperty

String name;

@JsonProperty

String address;

@JsonProperty

int yearOfAccreditation;

}

**UniversityController.java**

package lv.tsi.university.Controller;

import lv.tsi.university.Model.University;

import lv.tsi.university.Service.UniversityService;

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

import org.springframework.http.ResponseEntity;

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

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

import java.util.List;

@RestController

public class UniversityController {

/\* private final UniversityRepository universityRepository;

@Autowired

public UniversityController(UniversityRepository universityRepository) {

this.universityRepository = universityRepository;

}\*/

private final UniversityService universityService;

@Autowired

public UniversityController(UniversityService universityService) {

this.universityService = universityService;

}

@GetMapping("/university")

public ResponseEntity<List<University>> getAllUniversity() {

return universityService.getUniversities();

}

/\* @GetMapping("/getUniversity/{id})

public ResponseEntity getUniversityById(@PathVariable Long id) {

return employeeService.getEmployeeById(id);

}\*/

}

# Conclusion

Mostly all tasks have been accomplished.