



THE STATE UNIVERSITY OF ZANZIBAR

SCHOOL OF COMPUTING, COMMUNICATION AND MEDIA STUDIES

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

COURSE TITLE : ADVANCE WEBSITE PROGRAMMING

REG NO : BITA/5/21/021/TZ

NAME : SULEIMAN SAID MUSSA

LECTURE NAME : MR MASOUD HAMAD

ACADEMIC YEAR : 2022-2023

TASK : INDIVIDUAL ASSIGNMENT

a) The step of creating a project using spring initializer

Project

☐ Gradle - Groovy ☐ Gradle - Kotlin ☒ Java ☐ Kotlin ☐ Groovy
☒ Maven

Spring Boot

☐ 3.2.0 (SNAPSHOT) ☐ 3.2.0 (M1) ☐ 3.1.3 (SNAPSHOT) ☐ 3.1.2
☐ 3.0.10 (SNAPSHOT) ☐ 3.0.9 ☐ 2.7.15 (SNAPSHOT) ☒ 2.7.14

Project Metadata

Group	EmployeeManagement
Artifact	Employee
Name	Employee
Description	Demo project for Spring Boot
Package name	EmployeeManagement.Employee
Packaging	<input checked="" type="radio"/> Jar <input type="radio"/> War
Java	<input type="radio"/> 20 <input checked="" type="radio"/> 17 <input type="radio"/> 11 <input type="radio"/> 8

b) Maven Dependence

Dependencies

ADD DEPENDENCIES... CTRL + B

Spring Web WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Spring Data JPA SQL

Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

MS SQL Server Driver SQL

A JDBC and R2DBC driver that provides access to Microsoft SQL Server and Azure SQL Database from any Java application.

MyBatis Framework SQL

Persistence framework with support for custom SQL, stored procedures and advanced mappings. MyBatis couples objects with stored procedures or SQL statements using a XML descriptor or annotations.

c) Application. Properties

- d)
- e) `spring.datasource.url=jdbc:mysql://localhost:3306/Employee`
- f) `spring.datasource.username=root`
- g) `spring.datasource.password=`
- h)
- i) `spring.jpa.show-sql=true`
- j) `spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver`
- k) `spring.jpa.hibernate.ddl-auto=update`
- l) `server.port=8090`
- m)

d) Models

Employee.java

```
package com.example.Employee.model;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;
```

```
@Entity
@Table
public class employee {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int Id;
    private String FirstName;
    private String LastName;
    private String Email;

    public employee(){

    }

    public employee(int id, String firstName, String lastName, String email) {
        Id = id;
        FirstName = firstName;
        LastName = lastName;
        Email = email;
    }

    public String getFirstName() {
        return FirstName;
    }

    public void setFirstName(String firstName) {
        FirstName = firstName;
    }

    public String getLastName() {
        return LastName;
    }

    public void setLastName(String lastName) {
        LastName = lastName;
    }

    public String getEmail() {
```

```

        return Email;
    }

    public void setEmail(String email) {
        Email = email;
    }

    public static boolean isPresent() {
        return false;
    }

    public int getId() {
        return 0;
    }
}

```

f) Repository

Employee_repo.java

```

package com.example.Employee.repository;

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

import com.example.Employee.model.employee;

public interface Employee_repo extends JpaRepository<employee,Integer> {

}

```

g) Controller

Employee_Controller.java

```
package com.example.Employee.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.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.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.example.Employee.model.employee;
import com.example.Employee.repository.Employee_repo;

@RestController
@RequestMapping("/Employee")
@CrossOrigin
public class Employee_controller {
    @Autowired
    Employee_repo E_repo;

    @PostMapping("")
    public ResponseEntity<employee> newLicense(@RequestBody employee lc){
        employee newLicense_Application = E_repo.save(new employee(lc.getId(),lc.getFirstName(),lc.getLastName(),lc.getEmail()));

        return new ResponseEntity<>(newLicense_Application, HttpStatus.CREATED);
    }

    @GetMapping("")
    public ResponseEntity<List<Employee>> getAll(){
        List<employee> employee = E_repo.findAll();

        return new ResponseEntity<List<Employee>>(HttpStatus.OK);
    }

    @GetMapping("/{id}")
    public ResponseEntity<employee> getById(@PathVariable("id")int id){
        java.util.Optional<employee> employee = E_repo.findById(id);
```

```
        if(employee.isPresent()){
            return new ResponseEntity<>(employee.get(), HttpStatus.OK);
        }

        return null;
    }

    @DeleteMapping("/{id}")
    public void deleteLicense(@PathVariable Integer id){
        E_repo.deleteById(id);
    }

    @PutMapping("")
    public employee updateLicense(@RequestBody employee newemployee){
        return E_repo.save(newemployee);
    }
}
```