Java MVC Application

Part II Custom MVC Framework - Integrating with Database

1. Modificati metoda *getOneEmployee()* din clasa *EmployeeController.java,*  astfel incat sa returneze o instanta a clasei *Employee.java*.
2. Adaugati in proiect urmatoarea dependinta:

<dependency>

<groupId>org.codehaus.jackson</groupId>

<artifactId>jackson-jaxrs</artifactId>

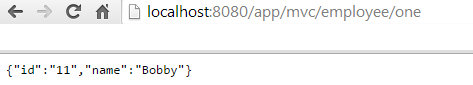
<version>${jackson-jaxrs.version}</version>

</dependency>

Unde jackson-jaxrs.version = 1.9.13

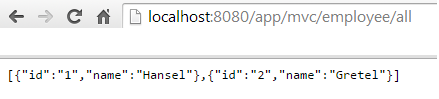
1. Modificati DispatcherServlet-ul astfel incat sa returneze catre client raspunsul in format JSON.

TEST: <http://localhost:8080/app/mvc/employees/one>



1. Modificati metoda *getAllEmployees()* din clasa *EmployeeController.java* astfel incat sa returneze o lista de angajati.

TEST: <http://localhost:8080/app/mvc/employees/all>



1. Modificati metoda *getAllDepartments()* din clasa *DepartmentController.java* astfel incat sa returneze o lista de departamente.

1. In pachetul ro.teamnet.zth.app.service creati interfata *EmployeeService.java.* Interfata va contine metoda *findAllEmployees()* si va returna o lista de angajati*.*
2. In pachetul ro.teamnet.zth.app.service creati clasa *EmployeeServiceImpl.java,* care implementeaza interfata *EmployeeService.java*. Metoda *findAllEmployees()* va returna toti utilizatorii din baza de date. Utilizati layaerul de DAO pentru a accesa baza de date.
3. Modificati metoda *getAllEmployees()* din clasa *EmployeeController.java* astfel incat sa returneze toti angajatii din baza de date, trecand prin layerul de service al aplicatiei.
4. In interfata *EmployeeService.java* adaugati metoda *findOneEmployee.* Aceasta va primi ca parametru id-ul unui angajat si va returna inregistrarea aferenta din baza de date. Implementati metoda.
5. Modificati metoda *getOneEmployee()* din clasa *EmployeeController.java* astfel incat sa utilizeze layerul de service implementat la punctul 10.
6. Modificati metoda *getOneEmployee()* din clasa *EmployeeController.java* astfel incat sa primeasca un parametru care sa specifice id-ul angajatului cautat.
7. Modificati DispatcherServlet-ul astfel incat, daca primeste parametri pe request, sa transmita metodei careia ii deleaga sarcina de a face prelucrarile de business acesti parametri.

TEST: <http://localhost:8080/app/mvc/employee/one?idEmployee=1>

1. Creati si utilizati layerul de service aferent procesarilor de forma “/departments”, “/jobs”, “/locations”.
2. Adaugati metoda *deleteOneEmployee()* in clasa *EmployeeController.java*. Implementati metoda astfel incat sa stearga un angajat din baza de date. Id-ul angajatului este primit ca parametru. Respectati layerele aplicatiei (controller – service – dao) la implementarea functionalitatii.
3. Pentru salvarea unui nou angajat in baza de date, este necesara transmiterea informatiilor despre noul angajat din browser catre serverul de aplicatie. Ce metoda HTTP va fi folosita in acest caz? Cum putem transmite informatiile (tip metoda HTTP si locul incapsularii datelor in pachetul de date).
4. In clasa *EmployeeController.java*, adaugati metoda *saveEmployee* care sa insereze un angajat.
5. Gasiti o modalitate prin care DispatcherServletul, in etapa de dispatch, poate transfera informatiile primite la punctul **17** catre metoda creata la punctul **18**.
6. Apelati metoda de salvare a unui angajat utilizand, de exemplu, pluginul Postman al browserului Chrome.
7. Introduceti metoda HTTP din request in Map-ul de Application Controllere. Utilizati aceasta informatie in mecanismul de regasire a unui Controller caruia sa i se delege executarea sarcinii de business.