8. Laborator: Client REST, Facelets – part2

8.1 Client Rest – create,delete

Vom folosi serviciile web oferite de laboratorul 6.

In continuare ne vom folosi de arhiva starter: laboratoree8_starter.zip. Aceasta arhiva contine laboratorul 6 cu serviciile CRUD pe Departments si Employee, dar si serviciile Rest aferente.

Continuam consumarea serviciilor REST cu privire la crearea si stergerea elementelor din DB.

Ne folosim de aceeasi implementare a serviciilor REST, anume de libraria Jersey folosita si in laboratoarele anterioare.

In continuare pachetul `ro.ulbs.ip.an3.frontend` si apoi clasa `DepartmentRest`:

```
package ro.ulbs.ip.an3.frontend;

import java.util.List;
import javax.ejb.LocalBean;
import javax.ejb.Stateless;
import javax.ws.rs.ClientErrorException;
import javax.ws.rs.client.Client;
import javax.ws.rs.client.WebTarget;
import javax.ws.rs.core.GenericType;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.Response;

@Stateless
@LocalBean
public class DepartmentRest {

    private static final String BASE_URI = "http://localhost:8080/restservices/departments";

    private final WebTarget webTarget;
    private final Client client;

    public DepartmentRest() {
        //instantierea unui client REST
        client = javax.ws.rs.client.ClientBuilder.newClient();

        //fiecare url va necesita o noua instanta WebTarget
        webTarget = client.target(BASE_URI);
    }

    public List<DepartmentDto> listAll() throws javax.ws.rs.ClientErrorException {
        //apel GET cu precizarea formatului dorit al datelor
        Response resp = webTarget.
                request(MediaType.APPLICATION_JSON).
                get(Response.class);

        //transformarea datelor in obiecte java
        List<DepartmentDto> ret = resp.readEntity(new GenericType<List<DepartmentDto>>() {
        });
        return ret;
    }

    public List<DepartmentDto> filterBy(String name) throws ClientErrorException {
        Response resp = webTarget.path("/search")
                .queryParam("filter", name)
                .request(MediaType.APPLICATION_JSON)
                .get(Response.class);
        return resp.readEntity(new GenericType<List<DepartmentDto>>() {
        });
    }

    public Integer create(DepartmentDto entry) throws ClientErrorException {
        Response resp = webTarget.request(MediaType.APPLICATION_JSON).put(javax.ws.rs.client.Entity.entity(entry,
MediaType.APPLICATION_JSON), Response.class);
```

```
        String s = resp.readEntity(String.class);
        try {
            Integer id = Integer.parseInt(s);
            return id;
        } catch (NumberFormatException nfe) {
            return 0;
        }
    }

    public void delete(DepartmentDto entry) throws javax.ws.rs.ClientErrorException{
        webTarget.path("/"+entry.getId())
                .request(MediaType.APPLICATION_JSON)
                .delete(Response.class);
    }
}
```

De asemenea managed bean-ul pentru gestiunea departamentelor:

```
package ro.ulbs.ip.an3.frontend;

import java.io.Serializable;
import java.util.List;
import javax.ejb.EJB;
import javax.enterprise.context.SessionScoped;
import javax.inject.Named;
import javax.faces.application.FacesMessage;
import javax.faces.context.FacesContext;

@SessionScoped
@Named("deptMBean")
public class DepartmentMBean implements Serializable {

    private static final long serialVersionUID = 10112;

    @EJB
    private DepartmentRest restClient;

    private String filterText;
    private DepartmentDto selectedDepartment;
    private boolean isCreate;
    private boolean isDelete;
    private Integer createdId;

    public DepartmentMBean() {
    }

    public List<DepartmentDto> getDepartments() {
        if (filterText != null && filterText.length() > 0) {
            return restClient.filterBy(filterText);
        } else {
            return restClient.listAll();
        }
    }

    public String getFilterText() {
        return filterText;
    }

    public void setFilterText(String filterText) {
        this.filterText = filterText;
    }

    public void filter() {

    }

    public void setSelected(DepartmentDto dpt) {
        this.selectedDepartment = dpt;
    }

    public DepartmentDto getSelected() {
        return selectedDepartment;
    }

    public String startCreate() {
        isCreate = true;
        createdId = null;
        selectedDepartment = new DepartmentDto();
        return "department_create";
```

```
    }

    public String endCreate() {
        createdId = restClient.create(selectedDepartment);
        isCreate = false;
        return "index";
    }

    public Integer getCreatedId() {
        return createdId;
    }

    public void setCreatedId(Integer createdId) {
        this.createdId = createdId;
    }

    public String startDelete() {
        isDelete = true;
        return "department_delete";
    }

    public String endDelete() {
        isDelete = false;
        try {
            restClient.delete(selectedDepartment);
        } catch (Exception e) {
            FacesContext.getCurrentInstance().addMessage("deptFilterForm", new FacesMessage("Error", e.getMessage()));
        }
        return "index";
    }

    public boolean isIsCreate() {
        return isCreate;
    }

    public void setIsCreate(boolean isCreate) {
        this.isCreate = isCreate;
    }

    public boolean isIsDelete() {
        return isDelete;
    }

    public void setIsDelete(boolean isDelete) {
        this.isDelete = isDelete;
    }
}
```

In plus fata de laboratorul anterior, adaugam in index.xhtml o noua coloana tabelului existent:

```
<h:column>
    <f:facet name="header">Action</f:facet>
        <h:commandButton value="Delete" action="#{deptMBean.startDelete}">
        <f:setPropertyActionListener value="#{dept}" target="#{deptMBean.selected}"/>
    </h:commandButton>
</h:column>
```

Si in afara tabelului, dar tot in interiorul tagului <form> avem nevoie de butonul de creare :

```
<h:commandButton value="New department" action="#{deptMBean.startCreate}"/>
<h:outputText value="Departament creat cu succes:" rendered="#{deptMBean.createdId != null}"/>
<h:outputText value="#{deptMBean.createdId}" rendered="#{deptMBean.createdId != null}"/>
```

Apoi cream fisierele *department_create.xhtml* si *department_delete.xhtml*:

Selectati nodul „Web Pages” si creati un noi fisiere facelet. (new > other > JavaServer Faces > JSF Page)

department_create.xhtml:

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets">
    <h:head>
        <title>Department</title>
    </h:head>
    <h:body>
        <h3>
        Department <h:outputText value="create" rendered="#{deptMBean.isCreate}"/>
        </h3>
        <br/>
        <f:view>
            <h:form id="DepartmemtCreateOrEditForm">
                Name:<h:inputText  value="#{deptMBean.selected.name}"/>
                <br/>
                <br/>
                <h:commandButton value="Create" action="#{deptMBean.endCreate}" rendered="#{deptMBean.isCreate}"/>

            </h:form>
        </f:view>
        <br/>
        <br/>
        <ui:include src="/goBack.xhtml"/>
    </h:body>
</html>
```

department_delete.xhtml:

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets">
    <h:head>
        <title>Department</title>
    </h:head>
    <h:body>
        <h3>
        Department Deletion
        </h3>
        <br/>
        <f:view>
            <h:form id="DepartmemtDeleteForm">
                Are you sure you want to delete Department (
                Id:<h:inputText  value="#{deptMBean.selected.id}"/>
                ,
                Name:<h:inputText  value="#{deptMBean.selected.name}"/>
                ) ?
                <br/>
                <h:commandButton value="Yes" action="#{deptMBean.endDelete}"/>
                <h:commandButton value="No"  action="index"/>
            </h:form>
        </f:view>
        <ui:include src="/goBack.xhtml"/>
    </h:body>
</html>
```

Compilati proiectul si faceti un nou deployment (http://localhost:4848/ > application > deploy).

Accesati aceste pagini web la: **http://localhost:8080/frontend/**

Sumar

Aplicatia de fata foloseste urmatoarele : Java EE, Maven, Facelets, client REST citire date (GET)
Serviciile WEB tip REST din laboratorul 6 sunt expuse la http://localhost:8080/restservices/departments
Paginile JSF sunt disponibile la http://localhost:8080/frontend/

Mai multe detalii pentru CDI:
https://www.javacodegeeks.com/cdi-tutorials
https://dzone.com/articles/cdi-di-p1
https://docs.oracle.com/javaee/6/tutorial/doc/giwhl.html

8.2 Probleme

8.2.1 Adaugati cod astfel incat sa avem in pagina index.xhtml un buton pentru actualizare (update) (similar ca pozitionare butonului 'delete')

8.2.2 Adaugati cod similar pentru a manipula (CRUD) datele din tabela Employee

Nota.
Fisierul department_create.xhtml are acelasi nume cu valoare returnata de functia startCreate.
Fisierul department_delete.xhtml are acelasi nume cu voarea returnata de functia startDelete. Daca doriti sa schimbati aceste denumiri trebuie sa actualizati valoarea returnata de functiile de mai sus, care valoare ajunge in atributul 'action' in jsf.

8.3 Configurare Glassfish
In cazul in care aveti eroare ce se refera la moxy atunci trebuie sa actualizati libraria moxy in Glassfish.
Bug: https://bugs.eclipse.org/bugs/show_bug.cgi?id=463169.
Gasiti moxy-2.6.1.jar in lista de resurse.
Redenumiti libraria existenta, copiati in …\glassfish\modules\ noul jar si reporniti Glassfish-ul

```
if exist %GF_DIR%\glassfish\modules\org.eclipse.persistence.moxy.jar (
        ren %GF_DIR%\glassfish\modules\org.eclipse.persistence.moxy.jar
%GF_DIR%\glassfish\modules\org.eclipse.persistence.moxy.jar_disabled
)

copy .\org.eclipse.persistence.moxy-2.6.1.jar %GF_DIR%\glassfish\modules\
```