ПРИЛОЖЕНИЕ А  
Листинг компонентов системы

AbstractFacade.java

package DAO;

import java.util.List;

import javax.persistence.EntityManager;

public abstract class AbstractFacade<T> {

private Class<T> entityClass;

public AbstractFacade(Class<T> entityClass) {

this.entityClass = entityClass;

}

protected abstract EntityManager getEntityManager();

public void create(T entity) {

getEntityManager().persist(entity);

}

public void edit(T entity) {

getEntityManager().merge(entity);

}

public void remove(T entity) {

getEntityManager().remove(getEntityManager().merge(entity));

}

public T find(Object id) {

return getEntityManager().find(entityClass, id);

}

public List<T> findAll() {

javax.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();

cq.select(cq.from(entityClass));

return getEntityManager().createQuery(cq).getResultList();

}

public List<T> findRange(int[] range) {

javax.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();

cq.select(cq.from(entityClass));

javax.persistence.Query q = getEntityManager().createQuery(cq);

q.setMaxResults(range[1] - range[0] + 1);

q.setFirstResult(range[0]);

return q.getResultList();

}

public int count() {

javax.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();

javax.persistence.criteria.Root<T> rt = cq.from(entityClass);

cq.select(getEntityManager().getCriteriaBuilder().count(rt));

javax.persistence.Query q = getEntityManager().createQuery(cq);

return ((Long) q.getSingleResult()).intValue();

}

}

**ArchiveFacade.java**

package DAO;

import Model.Archive;

import Model.Company;

import Model.Customer;

import Model.Document;

import java.util.Calendar;

import java.util.List;

import javax.annotation.Resource;

import javax.ejb.EJB;

import javax.ejb.SessionContext;

import javax.ejb.Stateless;

import javax.persistence.EntityManager;

import javax.persistence.PersistenceContext;

@Stateless

public class ArchiveFacade extends AbstractFacade<Archive> implements ArchiveFacadeLocal {

@PersistenceContext(unitName = "EnterpriseApplication3-ejbPU")

private EntityManager em;

@Resource

private SessionContext sc;

@EJB

DocumentFacadeLocal documentFacade;

@EJB

CompanyFacadeLocal companyFacade;

@EJB

CustomerFacadeLocal customerFacade;

@Override

protected EntityManager getEntityManager() {

return em;

}

@Override

public void addDoc(Document document, String code, Customer customer){

List<Document> list = documentFacade.findAll();

int lastId = list.size();

Company company = companyFacade.find((document.getIdDocument()));

if (code.equals(company.getCodename())){

Archive archive = new Archive();

archive.setDocument(document);

archive.setCustomer(customer);

archive.setCompany(company);

java.sql.Date date = new java.sql.Date(Calendar.getInstance().getTime().getTime());

archive.setDate(date);

customer.setName(customer.getName()+"|"+document.getName());

em.flush();

em.persist(customer);

em.persist(archive);

}

else {

sc.setRollbackOnly();

}

}

public ArchiveFacade() {

super(Archive.class);

}

}

**CompanyFacade**

package DAO;

import Model.Company;

import javax.ejb.Stateless;

import javax.persistence.EntityManager;

import javax.persistence.PersistenceContext;

@Stateless

public class CompanyFacade extends AbstractFacade<Company> implements CompanyFacadeLocal {

@PersistenceContext(unitName = "EnterpriseApplication3-ejbPU2")

private EntityManager em;

@Override

protected EntityManager getEntityManager() {

return em;

}

public CompanyFacade() {

super(Company.class);

}

}

**ArchiveView.java**

package presentation;

import DAO.ArchiveFacadeLocal;

import DAO.DocumentFacadeLocal;

import java.io.Serializable;

import javax.ejb.EJB;

import javax.enterprise.context.SessionScoped;

import javax.inject.Named;

import Model.Archive;

import static Model.Archive\_.document;

import Model.Document;

import java.util.List;

@Named(value = "archiveView")

@SessionScoped

public class ArchiveView implements Serializable{

@EJB

ArchiveFacadeLocal archiveFacade;

@EJB

DocumentFacadeLocal documentFacade;

public ArchiveView(){

this.archive = new Archive();

this.codeName = new String();

}

private Archive archive;

private String codeName;

public String getCodeName() {

return codeName;

}

public void setCodeName(String codeName) {

this.codeName = codeName;

}

public Archive getArchive() {

return archive;

}

public void setArchive(Archive archive) {

this.archive = archive;

}

public List<Archive> getAllArhive(){

List<Archive> list = archiveFacade.findAll();

return list;

}

public void addToArchive(int idDoc){

Document document = documentFacade.find(idDoc);

archiveFacade.addDoc(document, codeName, archive.getCustomer());

}

}

**PersonalView.java**

package presentation;

import DAO.PersonalFacadeLocal;

import Model.Personal;

import java.io.Serializable;

import java.util.List;

import javax.ejb.EJB;

import javax.enterprise.context.SessionScoped;

import javax.inject.Named;

@Named(value = "personalView")

@SessionScoped

public class PersonalView implements Serializable{

@EJB

PersonalFacadeLocal personalFacade;

public PersonalView(){

this.personal = new Personal();

}

private Personal personal;

public Personal getPersonal() {

return personal;

}

public void setPersonal(Personal personal) {

this.personal = personal;

}

public List<Personal> getAllPersonal(){

List<Personal> list = personalFacade.findAll();

return list;

}

public String addPersonal(){

personalFacade.create(personal);

return "/faces/personal.xhtml";

}

public void deletePersonal(int id){

personal = personalFacade.find(id);

personalFacade.remove(this.personal);

}

public String editPersonal(int id){

this.personal = personalFacade.find(id);

return "/faces/editPersonal.xhtml";

}

public String edit(){

personalFacade.edit(this.personal);

return "faces/personal.xhtml";

}

}

**Company.java**

package Model;

import java.io.Serializable;

import java.util.List;

import java.util.Objects;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToMany;

import javax.persistence.Table;

@Entity

@Table(name="companyEntity")

public class Company implements Serializable {

private static final long serialVersionUID = 1L;

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Integer idCompany;

private String Name;

private String INN;

private String Codename;

@OneToMany(fetch=FetchType.LAZY, cascade = CascadeType.REMOVE)

private List<Customer> customers;

@OneToMany(fetch=FetchType.LAZY, cascade = CascadeType.REMOVE)

private List<Archive> documents;

public Integer getIdCompany() {

return idCompany;

}

public void setIdCompany(Integer idCompany) {

this.idCompany = idCompany;

}

public String getName() {

return Name;

}

public void setName(String Name) {

this.Name = Name;

}

public String getINN() {

return INN;

}

public void setINN(String INN) {

this.INN = INN;

}

public String getCodename() {

return Codename;

}

public void setCodename(String Codename) {

this.Codename = Codename;

}

public List<Customer> getCustomers() {

return customers;

}

public void setCustomers(List<Customer> customers) {

this.customers = customers;

}

public List<Archive> getDocuments() {

return documents;

}

public void setDocuments(List<Archive> documents) {

this.documents = documents;

}

@Override

public boolean equals(Object obj) {

if (this == obj) {

return true;

}

if (obj == null) {

return false;

}

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

return false;

}

final Company other = (Company) obj;

if (!Objects.equals(this.idCompany, other.idCompany)) {

return false;

}

return true;

}

@Override

public String toString() {

return "Company{" + "idCompany=" + idCompany + '}';

}

}

**GOST.java**

package Model;

import java.io.Serializable;

import java.util.Objects;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name="GOST")

public class GOST implements Serializable {

private static final long serialVersionUID = 1L;

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Integer idGost;

private String number;

private String name;

private String description;

public Integer getIdGost() {

return idGost;

}

public void setIdGost(Integer idGost) {

this.idGost = idGost;

}

public String getNumber() {

return number;

}

public void setNumber(String number) {

this.number = number;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

@Override

public boolean equals(Object obj) {

if (this == obj) {

return true;

}

if (obj == null) {

return false;

}

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

return false;

}

final GOST other = (GOST) obj;

if (!Objects.equals(this.idGost, other.idGost)) {

return false;

}

return true;

}

@Override

public String toString() {

return "GOST{" + "idGost=" + idGost + '}';

}

}

**CompanyConverter.java**

package Converter;

import DAO.CompanyFacadeLocal;

import Model.Company;

import Model.GOST;

import javax.annotation.ManagedBean;

import javax.ejb.EJB;

import javax.enterprise.context.RequestScoped;

import javax.faces.application.FacesMessage;

import javax.faces.component.UIComponent;

import javax.faces.context.FacesContext;

import javax.faces.convert.Converter;

import javax.faces.convert.ConverterException;

import javax.inject.Named;

@ManagedBean

@RequestScoped

@Named("companyConverter")

public class CompanyConverter implements Converter{

@EJB

CompanyFacadeLocal companyFacade;

@Override

public Object getAsObject(FacesContext context, UIComponent component, String submittedValue) {

if (submittedValue == null || submittedValue.isEmpty()) {

return null;

}

try {

return companyFacade.find(Integer.valueOf(submittedValue));

} catch (NumberFormatException e) {

throw new ConverterException(new FacesMessage(String.format("%s is not a valid bonus ID", submittedValue)), e);

}

}

@Override

public String getAsString(FacesContext context, UIComponent component, Object modelValue) {

if (modelValue == null) {

return "";

}

if (modelValue instanceof Company) {

return String.valueOf(((Company) modelValue).getIdCompany());

} else {

throw new ConverterException(new FacesMessage(String.format("%s is not a valid bonus", modelValue)));

}

}

}

**CustomerConverter.java**

package Converter;

import DAO.CustomerFacadeLocal;

import Model.Customer;

import javax.annotation.ManagedBean;

import javax.ejb.EJB;

import javax.enterprise.context.RequestScoped;

import javax.faces.application.FacesMessage;

import javax.faces.component.UIComponent;

import javax.faces.context.FacesContext;

import javax.faces.convert.Converter;

import javax.faces.convert.ConverterException;

import javax.inject.Named;

@ManagedBean

@RequestScoped

@Named("customerConverter")

public class CustomerConverter implements Converter{

@EJB

CustomerFacadeLocal customerFacade;

@Override

public Object getAsObject(FacesContext context, UIComponent component, String submittedValue) {

if (submittedValue == null || submittedValue.isEmpty()) {

return null;

}

try {

return customerFacade.find(Integer.valueOf(submittedValue));

} catch (NumberFormatException e) {

throw new ConverterException(new FacesMessage(String.format("%s is not a valid bonus ID", submittedValue)), e);

}

}

@Override

public String getAsString(FacesContext context, UIComponent component, Object modelValue) {

if (modelValue == null) {

return "";

}

if (modelValue instanceof Customer) {

return String.valueOf(((Customer) modelValue).getIdCustomer());

} else {

throw new ConverterException(new FacesMessage(String.format("%s is not a valid bonus", modelValue)));

}

}

}

**DocumentFacadeLocal.java**

package DAO;

import Model.Document;

import java.util.List;

import javax.ejb.Local;

@Local

public interface DocumentFacadeLocal {

void create(Document document);

void edit(Document document);

void remove(Document document);

Document find(Object id);

List<Document> findAll();

List<Document> findRange(int[] range);

int count();

}

Исходный код программы целиком Вы можете увидеть, перейдя по ссылке

https://github.com/Haragan/RPS