/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package cta.service;

import cta.Consumption;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.List;

import javax.ejb.Stateless;

import javax.persistence.EntityManager;

import javax.persistence.PersistenceContext;

import javax.persistence.Query;

import javax.persistence.TypedQuery;

import javax.ws.rs.Consumes;

import javax.ws.rs.DELETE;

import javax.ws.rs.GET;

import javax.ws.rs.POST;

import javax.ws.rs.PUT;

import javax.ws.rs.Path;

import javax.ws.rs.PathParam;

import javax.ws.rs.Produces;

import javax.ws.rs.core.MediaType;

/\*\*

\*

\* @author nafiy

\*/

@Stateless

@Path("cta.consumption")

public class ConsumptionFacadeREST extends AbstractFacade<Consumption> {

@PersistenceContext(unitName = "CalorieTrackerApplicationPU")

private EntityManager em;

public ConsumptionFacadeREST() {

super(Consumption.class);

}

@POST

@Override

@Consumes({MediaType.APPLICATION\_XML, MediaType.APPLICATION\_JSON})

public void create(Consumption entity) {

super.create(entity);

}

@PUT

@Path("{id}")

@Consumes({MediaType.APPLICATION\_XML, MediaType.APPLICATION\_JSON})

public void edit(@PathParam("id") Short id, Consumption entity) {

super.edit(entity);

}

@DELETE

@Path("{id}")

public void remove(@PathParam("id") Short id) {

super.remove(super.find(id));

}

@GET

@Path("{id}")

@Produces({MediaType.APPLICATION\_XML, MediaType.APPLICATION\_JSON})

public Consumption find(@PathParam("id") Short id) {

return super.find(id);

}

@GET

@Override

@Produces({MediaType.APPLICATION\_XML, MediaType.APPLICATION\_JSON})

public List<Consumption> findAll() {

return super.findAll();

}

@GET

@Path("{from}/{to}")

@Produces({MediaType.APPLICATION\_XML, MediaType.APPLICATION\_JSON})

public List<Consumption> findRange(@PathParam("from") Integer from, @PathParam("to") Integer to) {

return super.findRange(new int[]{from, to});

}

@GET

@Path("count")

@Produces(MediaType.TEXT\_PLAIN)

public String countREST() {

return String.valueOf(super.count());

}

@Override

protected EntityManager getEntityManager() {

return em;

}

@GET

@Path("findByConUserid/{conUserid}")

@Produces({"application/json"})

public List<Consumption> findByConUserid(@PathParam("conUserid") int conUserid) {

Query query = em.createNamedQuery("Consumption.findByConUserid");

query.setParameter("conUserid", conUserid);

return query.getResultList();

}

@GET

@Path("findByConDate/{conDate}")

@Produces({"application/json"})

public List<Consumption> findByConDate(@PathParam("conDate") String conDate)throws Exception {

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

Date dt = sdf.parse(conDate);

Query query = em.createNamedQuery("Consumption.findByConDate");

query.setParameter("conDate", dt);

return query.getResultList();

}

@GET

@Path("findByConFoodid/{conFoodid}")

@Produces({"application/json"})

public List<Consumption> findByConFoodid(@PathParam("conFoodid") int conFoodid) {

Query query = em.createNamedQuery("Consumption.findByConFoodid");

query.setParameter("conFoodid", conFoodid);

return query.getResultList();

}

@GET

@Path("findByConFoodamount/{conFoodamount}")

@Produces({"application/json"})

public List<Consumption> findByConFoodamount(@PathParam("conFoodamount") Double conFoodamount) {

Query query = em.createNamedQuery("Consumption.findByConFoodamount");

query.setParameter("conFoodamount", conFoodamount);

return query.getResultList();

}

@GET

@Path("findTotalCalorieConsumed/{conUserid}/{conDate}")

@Produces("text/plain")

public double findTotalCalorieConsumed(@PathParam("conUserid") int conUserid,@PathParam("conDate") String conDate)throws Exception {

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

Date dt = sdf.parse(conDate);

TypedQuery<Object[]> query = em.createQuery("SELECT c.conFoodid.foodCalorie,c.conFoodamount FROM Consumption AS c WHERE c.conUserid.userId = :conUserid AND c.conDate = :conDate",Object[].class);

query.setParameter("conUserid", conUserid);

query.setParameter("conDate", dt);

List<Object[]> queryList = query.getResultList();

double con = 0.0;

for (Object[] row : queryList)

{

double fc = (Double)row[0];

double ca = (Double)row[1];

con = con + fc \* ca;

}

return con;

}

}