/\*

\* 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.Credential;

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.credential")

public class CredentialFacadeREST extends AbstractFacade<Credential> {

@PersistenceContext(unitName = "CalorieTrackerApplicationPU")

private EntityManager em;

public CredentialFacadeREST() {

super(Credential.class);

}

@POST

@Override

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

public void create(Credential entity) {

super.create(entity);

}

@PUT

@Path("{id}")

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

public void edit(@PathParam("id") Short id, Credential 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 Credential find(@PathParam("id") Short id) {

return super.find(id);

}

@GET

@Override

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

public List<Credential> findAll() {

return super.findAll();

}

@GET

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

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

public List<Credential> 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("findByCreUserid/{creUserid}")

@Produces({"application/json"})

public List<Credential> findByCreUserid(@PathParam("creUserid") int creUserid) {

Query query = em.createNamedQuery("Credential.findByCreUserid");

query.setParameter("creUserid", creUserid);

return query.getResultList();

}

@GET

@Path("findByCreSignupdate/{creSignupdate}")

@Produces({"application/json"})

public List<Credential> findByCreSignupdate(@PathParam("creSignupdate") String creSignupdate) {

Query query = em.createNamedQuery("Credential.findByCreSignupdate");

query.setParameter("creSignupdate", creSignupdate);

return query.getResultList();

}

@GET

@Path("findByCreUsername/{creUsername}")

@Produces({"application/json"})

public List<Credential> findByCreUsername(@PathParam("creUsername") String creUsername) {

Query query = em.createNamedQuery("Credential.findByCreUsername");

query.setParameter("creUsername", creUsername);

return query.getResultList();

}

@GET

@Path("findByCrePasshash/{crePasshash}")

@Produces({"application/json"})

public List<Credential> findByCrePasshash(@PathParam("crePasshash") String crePasshash) {

Query query = em.createNamedQuery("Credential.findByCrePasshash");

query.setParameter("crePasshash", crePasshash);

return query.getResultList();

}

@GET

@Path("findByCreIdAndUserName/{creId}/{userName}")

@Produces({"application/json"})

public List<Credential> findByCreIdAndUserName(@PathParam("creId") Short creId,@PathParam("userName") String userName) {

TypedQuery<Credential> query = em.createQuery("SELECT c FROM Credential c WHERE c.creId = :creId AND c.creUserid.userName = :userName", Credential.class);

query.setParameter("creId", creId);

query.setParameter("userName", userName);

return query.getResultList();

}

@GET

@Path("findByCreIdAndUserSurname/{creId}/{userSurname}")

@Produces({"application/json"})

public List<Credential> findByCreIdAndUserSurname(@PathParam("creId") Short creId,@PathParam("userSurname") String userSurname) {

Query query = em.createNamedQuery("Credential.findByCreIdAndUserSurname");

query.setParameter("creId", creId);

query.setParameter("userSurname", userSurname);

return query.getResultList();

}

@GET

@Path("findByCreUsernameAndCrePasshash/{creUsername}/{crePasshash}")

@Produces({"application/json"})

public List<Credential> findByCreUsernameAndCrePasshash(@PathParam("creUsername") String creUsername,@PathParam("crePasshash") String crePasshash) {

Query query = em.createNamedQuery("Credential.findByCreUsernameAndCrePasshash");

query.setParameter("creUsername", creUsername);

query.setParameter("crePasshash", crePasshash);

return query.getResultList();

}

}