

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package predictif.service;

import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.GregorianCalendar;
import java.util.List;
import javax.persistence.EntityTransaction;
import predictif.dao.ClientDao;
import predictif.dao.EmployeDao;
import predictif.dao.MediumDao;
import predictif.dao.PredictionDao;
import predictif.dao.SigneAstrologiqueDao;
import predictif.modele.Employe;
import predictif.modele.Medium;
import predictif.modele.Predictions.AmourPrediction;
import predictif.modele.Predictions.SantePrediction;
import predictif.modele.Predictions.TravailPrediction;
import predictif.modele.SigneAstrologique;
import predictif.Util.JpaUtil;

/**
 *
 * @author Pitou
 */
public class PrepareBD
{
    protected Service service;

    protected EmployeDao employeDao;
    protected MediumDao mediumDao;
    protected SigneAstrologiqueDao signeDao;
    protected PredictionDao predictionDao;
    protected ClientDao clientDao;
    public static MessageDigest messageDigest;

    public PrepareBD()
    {
        AmourPrediction.setSigle();
        SantePrediction.setSigle();
        TravailPrediction.setSigle();
        employeDao = EmployeDao.getInstance();
        mediumDao = MediumDao.getInstance();
        signeDao = SigneAstrologiqueDao.getInstance();
        predictionDao = PredictionDao.getInstance();
        clientDao = ClientDao.getInstance();
        service = new Service();
        try
        {
            messageDigest = MessageDigest.getInstance("MD5");
        }
        catch (NoSuchAlgorithmException ex)
        {
            System.err.println("Erreur d'algorithme");
        }
    }

    /**
     * Remplissage de la BD en dur :
     * <ul>
     * <li>Prédictions</li>
     * <li>Signes astro</li>
     * <li>Employés</li>
     * <li>Médiums</li>
     * </ul>

```

```

    */
    public void populate()
    {
        ArrayList<Employe> employes = new ArrayList<Employe>();
        employes.add(createEmploye("Thibaut", "Bernard", "hashedpasswd"));
        employes.add(createEmploye("Boris", "Aloin", "hashedpasswd"));
        employes.add(createEmploye("Jean", "Durand", "hashedpasswd"));
        employes.add(createEmploye("Jean", "Dupond", "hashedpasswd"));
        employes.add(createEmploye("Gosouep", "Dasilva", "hashedpasswd"));
        employes.add(createEmploye("Icare", "Rodriguez", "hashedpasswd"));

        ArrayList<SigneAstrologique> signes = new ArrayList<SigneAstrologique>();
        signes.add(new SigneAstrologique("Bélier", 21, 19, 3, 4));
        signes.add(new SigneAstrologique("Taureau", 20, 20, 4, 5));
        signes.add(new SigneAstrologique("Gémeaux", 21, 21, 5, 6));
        signes.add(new SigneAstrologique("Cancer", 22, 22, 6, 7));
        signes.add(new SigneAstrologique("Lion", 23, 22, 7, 8));
        signes.add(new SigneAstrologique("Vierge", 23, 22, 8, 9));
        signes.add(new SigneAstrologique("Balance", 23, 22, 9, 10));
        signes.add(new SigneAstrologique("Scorpion", 23, 22, 10, 11));
        signes.add(new SigneAstrologique("Sagittaire", 23, 21, 11, 12));
        signes.add(new SigneAstrologique("Verseau", 21, 18, 1, 2));
        signes.add(new SigneAstrologique("Poisson", 19, 20, 2, 3));

        ArrayList<TravailPrediction> travailPredictions = new ArrayList<TravailPrediction>();
        travailPredictions.add(new TravailPrediction(1, "Que vous arrive-t-il aujourd'hui? Vous voyez tout en noir, "
            + "Le pessimisme vient éteindre ce bon moral des jours derniers"));
        travailPredictions.add(new TravailPrediction(5, "Les astres dans votre thème aujourd'hui vous apporteront tout leur concours chanceux et "
            + "vous garantiront des appuis en haut lieu."));
        travailPredictions.add(new TravailPrediction(3, "L'évolution professionnelle amorcée depuis quelque temps déjà se poursuivra activement et devrait commencer à porter ses fruits."));
        travailPredictions.add(new TravailPrediction(2, "Si vous travaillez sur de grands projets, soyez patient et persévérant. En effet, difficultés et retards pourraient se multiplier."));
        travailPredictions.add(new TravailPrediction(3, "ur le plan professionnel, les gros soucis devraient vous être épargnés."));
        travailPredictions.add(new TravailPrediction(4, "L'ambition vous guidera : ce sera un ressort significatif de votre personnalité. "
            + "Vous irez au fond des choses"));

        ArrayList<SantePrediction> santePredictions = new ArrayList<SantePrediction>();
        santePredictions.add(new SantePrediction(1, "Vous aurez besoin aujourd'hui de beaucoup de vitamine C pour renforcer vos défenses naturelles. "
            + "Prenez au petit-déjeuner un verre de jus de cassis"));
        santePredictions.add(new SantePrediction(2, "Pas de problème de santé particulier en ce moment, les planètes régissant la vitalité étant toutes bien aspectées"));
        santePredictions.add(new SantePrediction(4, "Quelle santé ! Ce n'est sans doute pas en ce moment que vous rendrez visite à un médecin !"));
        santePredictions.add(new SantePrediction(3, "Votre bonne santé sera conditionnée par un bon équilibre de vos fonctions digestives mais également par une hygiène de vie très stricte"));
        santePredictions.add(new SantePrediction(2, "Vous aurez raison d'observer la plus stricte prudence en ce qui concerne les maladies sexuellement transmissibles. "
            + "Sachez que les accidents en ce domaine ne sont nullement une punition infligée par un hypothétique Dieu de la vengeance(trouvé sur un site!));

        List<AmourPrediction> amourPredictions = new ArrayList<AmourPrediction>();
        amourPredictions.add(new AmourPrediction(1, "Pas de chance en amour"));
        amourPredictions.add(new AmourPrediction(4, "En ce qui concerne votre vie sentimentale actuelle ou à venir, votre charme sera sûr d'opérer."));
        amourPredictions.add(new AmourPrediction(3, "L'amour arrive souvent là ou on s'y attend le moins, et vous en aurez un bel exemple aujourd'hui."));
        amourPredictions.add(new AmourPrediction(3, "Du nouveau au niveau de votre vie amoureuse qui se portera bien mieux que ces derniers jours."));
    }

```

```

ArrayList<Medium> mediums = new ArrayList<Medium>();
mediums.add(new Medium("Madame Irma"));
mediums.add(new Medium("Madame Soleil"));
mediums.add(new Medium("Big Brother"));
mediums.add(new Medium("Luna"));
mediums.add(new Medium("Omni"));

```

```

JpaUtil.openEntityManager();
EntityTransaction tx = null;

```

```

try
{
    tx = JpaUtil.getEntityManagerTransaction();
    tx.begin();
    //Remplissage
    employeDao.create(employes);
    signeDao.create(signes);
    mediumDao.create(mediums);
    predictionDao.create(amourPredictions);
    predictionDao.create(travailPredictions);
    predictionDao.create(santePredictions);
    tx.commit();
}
catch (Exception e)
{
    System.err.println("Erreur rencontrée à preparerBD : " + e.toString());
    if (tx != null && tx.isActive())
    {
        tx.rollback();
    }
}
finally
{
    JpaUtil.closeEntityManager();
    addClients();
}
}

```

```
//Permet de tester la fonction permettant de calculer le min
```

```
public void addClients()
```

```

{
    GregorianCalendar calendar = new GregorianCalendar(1991, 10, 21);
    List<Medium> mediums = service.getAllMediums();
    List<Employe> employes = getAllEmployes();
    service.createClient("fouchet", "pierre", "rue titi", "jean@email", "0665642480",
calendar, mediums, employes.get(0));
    service.createClient("fouchet2", "pierre", "rue titi", "jean@email", "0665642480",
calendar, mediums, employes.get(0));
    service.createClient("fouchet3", "pierre", "rue titi", "jean@email", "0665642480",
calendar, mediums, employes.get(1));
    service.createClient("fouchet", "pierre", "rue titi", "jean@email", "0665642480",
calendar, mediums, employes.get(1));
    service.createClient("fouchet2", "pierre", "rue titi", "jean@email", "0665642480",
calendar, mediums, employes.get(2));
    service.createClient("fouchet3", "pierre", "rue titi", "jean@email", "0665642480",
calendar, mediums, employes.get(2));
    service.createClient("fouchet", "pierre", "rue titi", "jean@email", "0665642480",
calendar, mediums, employes.get(3));
    service.createClient("fouchet2", "pierre", "rue titi", "jean@email", "0665642480",
calendar, mediums, employes.get(3));
    service.createClient("fouchet2", "pierre", "rue titi", "jean@email", "0665642480",
calendar, mediums, employes.get(5));
}

```

```

public List<Employe> getAllEmployes()
{
    List<Employe> employes = null;

```

```
JpaUtil.openEntityManager();

    try
    {
        employes = employeeDao.findAllEmploye();
    }
    catch (Exception e)
    {
        System.err.println("erreur getAllEmployes service : "+e.getMessage());
    }
    finally
    {
        JpaUtil.closeEntityManager();
        return employes;
    }
}

public Employee createEmployee(String nom, String prenom, String mdp)
{
    byte[] passwd = messageDigest.digest(mdp.getBytes());
    Employee employe = new Employee(nom, prenom, passwd);
    return employe;
}
}
```