

## Application JAVA SE + JPA

### Entity Client

@Entity

@Table(name = "CLIENT")

@XmlRootElement

@NamedQueries({

    @NamedQuery(name = "Client.findAll", query = "SELECT c FROM Client c")

    , @NamedQuery(name = "Client.findByNcli", query = "SELECT c FROM Client c WHERE c.ncli = :ncli")

    , @NamedQuery(name = "Client.findByNom", query = "SELECT c FROM Client c WHERE c.nom = :nom")

    , @NamedQuery(name = "Client.findByPrenom", query = "SELECT c FROM Client c WHERE c.prenom = :prenom")

    , @NamedQuery(name = "Client.findByLoc", query = "SELECT c FROM Client c WHERE c.loc = :loc")

    , @NamedQuery(name = "Client.findBySexe", query = "SELECT c FROM Client c WHERE c.sexe = :sexe")

    , @NamedQuery(name = "Client.findByBareme", query = "SELECT c FROM Client c WHERE c.bareme = :bareme")

    , @NamedQuery(name = "Client.findByPasswd", query = "SELECT c FROM Client c WHERE c.passwd = :passwd"))}}

public class Client implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id

    @Basic(optional = false)

    @Column(name = "NCLI")

    private String ncli;

    @Basic(optional = false)

    @Column(name = "NOM")

    private String nom;

    @Basic(optional = false)

    @Column(name = "PRENOM")

```

private String prenom;
@Basic(optional = false)
@Column(name = "LOC")
private String loc;
@Basic(optional = false)
@Column(name = "SEXE")
private Character sexe;
@Basic(optional = false)
@Column(name = "BAREME")
private double bareme;
@Basic(optional = false)
@Column(name = "PASSWD")
private String passwd;

public Client() { }
public Client(String ncli) { this.ncli = ncli; }
public Client(String ncli, String nom, String prenom, String loc, Character sexe, double bareme,
                String passwd)
{ this.ncli = ncli;    this.nom = nom;    this.prenom = prenom;    this.loc = loc;
  this.sexe = sexe;    this.bareme = bareme;    this.passwd = passwd; }

public String toString() {
return "Nom:"+this.nom+" Prénom:"+this.prenom+" Localité:"+this.loc; }

// + getter et setter !!!!
}

```

## Class Tools

```
public class Tools {

    public static EntityManager getEntityManager()
    {
        EntityManagerFactory emf = Persistence.createEntityManagerFactory("JavaApplication17PU");
        EntityManager em = emf.createEntityManager();
        return em; }

    public static Client findClient(EntityManager em, String ncli)
    { return em.find(Client.class, ncli); }

    public static List<Client> findAll(EntityManager em)
    { Query q=em.createNamedQuery("Client.findAll");
        List<Client> list=q.getResultList();
        return list; }

    public static boolean removeClient(EntityManager em, String ncli)
    { boolean ok=false;
        Client C=findClient(em,ncli);
        if(C!=null)
        { EntityTransaction T=em.getTransaction();
            T.begin();
            em.remove(C);
            T.commit();
            C=findClient(em,ncli);
            if(C==null) ok=true;
        }
        return ok;
    }
}
```

```

public static Client updateClient(EntityManager em, String ncli, float bareme)
{
    Client C=findClient(em,ncli);
    if(C!=null)
        {
            EntityTransaction T=em.getTransaction();
            T.begin();
            C.setBareme(bareme);
            em.merge(C);
            T.commit();

        }
    return C;}

public static void persist(EntityManager em,Object object) {
    em.getTransaction().begin();
    try {
        em.persist(object);
        em.getTransaction().commit();
    } catch (Exception e) {
        e.printStackTrace();
        em.getTransaction().rollback();
    } finally {
        em.close();
    }
}
}

```

## Classe Application

```
public class Main {  
    public static void main(String[] args) {  
  
        EntityManager em=getEntityManager();  
        Client C=new Client();  
  
        /* C.setNcli("6");C.setNom("Albert2");C.setPrenom("Luc2");  
        C.setLoc("Mons");C.setSexe('M');C.setPasswd("azerty");  
        C.setBareme(2500.0f);  
        Tools.persist(em, C);*/  
  
        C=Tools.findClient(em, "1");  
        System.out.println(C.toString());  
  
        /*  
        String ncli="6";  
        if(Tools.removeClient(em, ncli))System.out.println("Client "+ncli+" effacé");  
        else System.out.println("Client "+ncli+" pas effacé"); */  
  
        Tools.updateClient(em, "1", 3666.0f);  
  
        List<Client> list=Tools.findAll(em);  
        for(Client cli:list)System.out.println(cli.toString());  
    }  
}
```