DAO

```java
package fr.insalyon.dasi.gustatif.dao;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;
import javax.persistence.RollbackException;

/**
 * Cette classe fournit des méthodes statiques utiles pour accéder aux
 * fonctionnalités de JPA (Entity Manager, Entity Transaction). Le nom de
 * l'unité de persistance (PERSISTENCE_UNIT_NAME) doit être conforme à la
 * configuration indiquée dans le fichier persistence.xml du projet.
 *
 * Version du 22 Mars 2016
 *
 * @author DASI Team
 */
public class JpaUtil {

    //**********************************************************************
    // * TODO: IMPORTANT -- Adapter le nom de l'Unité de Persistance (cf. persistence.xml) *
    //**********************************************************************
    /**
     * Nom de l'unité de persistance utilisée par la Factory de Entity Manager.
     * <br/><strong>Vérifier le nom de l'unité de persistance
     * (cf. persistence.xml)</strong>
     */
    public static final String PERSISTENCE_UNIT_NAME = "fr.insalyon.dasi_Gustatif_database";
    /**
     * Factory de Entity Manager liée à l'unité de persistance.
     * <br/><strong>Vérifier le nom de l'unité de persistance indiquée dans
     * l'attribut statique PERSISTENCE_UNIT_NAME
     * (cf. persistence.xml)</strong>
     */
    private static EntityManagerFactory entityManagerFactory = Persistence.createEntityManagerFactory(PERSISTENCE_UNIT_NAME);
    /**
     * Gère les instances courantes de Entity Manager liées aux Threads.
     * L'utilisation de ThreadLocal garantie une unique instance courante par
     * Thread.
     */
    private static final ThreadLocal<EntityManager> threadLocalEntityManager = new ThreadLocal<EntityManager>() {

        @Override
```

```
44          protected EntityManager initialValue() {
45              return null;
46          }
47      };
48
49      // Pause (sans exception)
50      private static void pause(long milliseconds) {
51          try {
52              Thread.sleep(milliseconds);
53          } catch (InterruptedException ex) {
54          }
55      }
56
57      // Log sur la console
58      // Les flush & pause sont là pour (tenter de) synchroniser les sorties sur la console
59      private static void log(String message) {
60          System.out.flush();
61          pause(5);
62          System.err.println("[JpaUtil:Log] " + message);
63          System.err.flush();
64          pause(5);
65      }
66
67      /**
68       * Initialise la Factory de Entity Manager (nécessaire au fonctionnement de
69       * JpaUtil dans une Application Web sous Glassfish).
70       * <br/><strong>À utiliser uniquement dans la méthode init() de la Servlet
71       * Contrôleur (ActionServlet).</strong>
72       */
73      public static synchronized void init() {
74          log("Initialisation de la factory de contexte de persistance");
75          if (entityManagerFactory != null) {
76              entityManagerFactory.close();
77          }
78          entityManagerFactory = Persistence.createEntityManagerFactory(PERSISTENCE_UNIT_NAME);
79      }
80
81      /**
82       * Libère la Factory de Entity Manager (pour permettre un futur rechargement
83       * propre d'une Application Web sous Glassfish).
84       * <br/><strong>À utiliser uniquement dans la méthode destroy() de la
85       * Servlet Contrôleur (ActionServlet).</strong>
86       */
87      public static synchronized void destroy() {
88          log("Libération de la factory de contexte de persistance");
89          if (entityManagerFactory != null) {
90              entityManagerFactory.close();
91              entityManagerFactory = null;
92          }
```

```
93      }
94
95      /**
96       * Créée l'instance courante de Entity Manager (liée à ce Thread).
97       * <br/><strong>À utiliser uniquement au niveau Service.</strong>
98       */
99      public static void creerEntityManager() {
100         log("Création du contexte de persistance");
101         threadLocalEntityManager.set(entityManagerFactory.createEntityMa
            nager());
102     }
103
104     /**
105      * Ferme l'instance courante de Entity Manager (liée à ce Thread).
106      * <br/><strong>À utiliser uniquement au niveau Service.</strong>
107      */
108     public static void fermerEntityManager() {
109         log("Fermeture du contexte de persistance");
110         EntityManager em = threadLocalEntityManager.get();
111         em.close();
112         threadLocalEntityManager.set(null);
113     }
114
115     /**
116      * Démarre une transaction sur l'instance courante de Entity Manager.
117      * <br/><strong>À utiliser uniquement au niveau Service.</strong>
118      */
119     public static void ouvrirTransaction() {
120         log("Début de la transaction");
121         EntityManager em = threadLocalEntityManager.get();
122         if (em.getTransaction().isActive()) {
123             log("ATTENTION: la transaction est déjà ouverte");
124         }
125         em.getTransaction().begin();
126     }
127
128     /**
129      * Valide la transaction courante sur l'instance courante de
        Entity Manager.
130      * <br/><strong>À utiliser uniquement au niveau Service.</strong>
131      *
132      * @exception RollbackException lorsque le <em>commit</em> n'a pas
        réussi.
133      */
134     public static void validerTransaction() throws RollbackException {
135         log("Validation (commit) de la transaction");
136         EntityManager em = threadLocalEntityManager.get();
137         if (!em.getTransaction().isActive()) {
138             log("ATTENTION: la transaction N'est PAS ouverte");
139         }
140         em.getTransaction().commit();
141     }
142
```

```
143         /**
144          * Annule la transaction courante sur l'instance courante de
             Entity Manager.
145          * Si la transaction courante n'est pas démarrée, cette méthode
             n'effectue
146          * aucune opération.
147          * <br/><strong>À utiliser uniquement au niveau Service.</strong>
148          */
149         public static void annulerTransaction() {
150             log("Annulation (rollback) de la transaction");
151             EntityManager em = threadLocalEntityManager.get();
152             if (!em.getTransaction().isActive()) {
153                 log("ATTENTION: la transaction N'est PAS ouverte =>
                    annulation ignorée par JpaUtil");
154             } else {
155                 em.getTransaction().rollback();
156             }
157         }
158 
159         /**
160          * Retourne l'instance courante de Entity Manager.
161          * <br/><strong>À utiliser uniquement au niveau DAO.</strong>
162          *
163          * @return instance de Entity Manager
164          */
165         public static EntityManager obtenirEntityManager() {
166             log("Obtention du contexte de persistance");
167             return threadLocalEntityManager.get();
168         }
169 
170     }
171 
```

```java
package fr.insalyon.dasi.gustatif.dao;

import fr.insalyon.dasi.gustatif.metier.modele.Client;
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Query;


public class ClientDao {

    public void create(Client client) throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        try {
            em.persist(client);
        }
        catch(Exception e) {
            throw e;
        }
    }

    public Client update(Client client) throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        try {
            client = em.merge(client);
        }
        catch(Exception e){
            throw e;
        }
        return client;
    }

    public Client findById(Long id) throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        Client client = null;
        try {
            client = em.find(Client.class, id);
        }
        catch(Exception e) {
            throw e;
        }
        return client;
    }

    public Client findByMailAndPassword(String mail, String password)
            throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        List<Client> clients = null;
        Client c = null;
        try {
            Query q = em.createQuery("SELECT c FROM Client c WHERE
            c.mail = "
                    + ":mail AND c.motDePasse = :password");
            q.setParameter("mail", mail);
            q.setParameter("password", password);
```

```
54                  clients = (List<Client>) q.getResultList();
55              }
56              catch(Exception e) {
57                  throw e;
58              }
59              try {
60                  c = clients.get(0);
61              } catch (Exception e){
62              }
63              return c;
64          }
65
66          public Client findByMail(String mail) throws Throwable {
67              EntityManager em = JpaUtil.obtenirEntityManager();
68              List<Client> clients = null;
69              Client c = null;
70              try {
71                  Query q = em.createQuery("SELECT c FROM Client c "
72                          + "WHERE c.mail = :mail");
73                  q.setParameter("mail", mail);
74                  clients = (List<Client>) q.getResultList();
75              }
76              catch(Exception e) {
77                  throw e;
78              }
79              try {
80                  c = clients.get(0);
81              } catch (Exception e){
82              }
83              return c;
84          }
85
86          public List<Client> findAll() throws Throwable {
87              EntityManager em = JpaUtil.obtenirEntityManager();
88              List<Client> clients = null;
89              try {
90                  Query q = em.createQuery("SELECT c FROM Client c ORDER BY
                    c.nom");
91                  clients = (List<Client>) q.getResultList();
92              }
93              catch(Exception e) {
94                  throw e;
95              }
96              return clients;
97          }
98      }
99
```

```java
package fr.insalyon.dasi.gustatif.dao;


import fr.insalyon.dasi.gustatif.metier.modele.Restaurant;
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Query;


public class RestaurantDao {

    public void create(Restaurant restaurant) throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        try {
            em.persist(restaurant);
        }
        catch(Exception e) {
            throw e;
        }
    }

    public Restaurant update(Restaurant restaurant) throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        try {
            restaurant = em.merge(restaurant);
        }
        catch(Exception e){
            throw e;
        }
        return restaurant;
    }

    public Restaurant findById(Long id) throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        Restaurant restaurant = null;
        try {
            restaurant = em.find(Restaurant.class, id);
        }
        catch(Exception e) {
            throw e;
        }
        return restaurant;
    }

    public List<Restaurant> findAll() throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        List<Restaurant> restaurants = null;
        try {
            Query q = em.createQuery("SELECT r FROM Restaurant r ORDER
            BY r.denomination");
            restaurants = (List<Restaurant>) q.getResultList();
        }
        catch(Exception e) {
            throw e;
```

```
54              }
55
56              return restaurants;
57          }
58
59
60      }
61
```

```java
package fr.insalyon.dasi.gustatif.dao;

import fr.insalyon.dasi.gustatif.metier.modele.Produit;
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Query;

public class ProduitDao {
    public void create(Produit produit) throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        try {
            em.persist(produit);
        }
        catch(Exception e) {
            throw e;
        }
    }

    public Produit update(Produit produit) throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        try {
            produit = em.merge(produit);
        }
        catch(Exception e){
            throw e;
        }
        return produit;
    }

    public Produit findById(Long id) throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        Produit produit = null;
        try {
            produit = em.find(Produit.class, id);
        }
        catch(Exception e) {
            throw e;
        }
        return produit;
    }

    public List<Produit> findAll() throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        List<Produit> produits = null;
        try {
            Query q = em.createQuery("SELECT p FROM Produit p");
            produits = (List<Produit>) q.getResultList();
        }
        catch(Exception e) {
            throw e;
        }
        return produits;
    }
}
```

```java
/*
 * 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 fr.insalyon.dasi.gustatif.dao;

import fr.insalyon.dasi.gustatif.metier.modele.Livreur;
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Query;

/**
 *
 * @author Nico
 */
public class LivreurDao {
    public void create(Livreur livreur) throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        try {
            em.persist(livreur);
        }
        catch(Exception e) {
            throw e;
        }
    }

    public Livreur update(Livreur livreur) throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        try {
            livreur = em.merge(livreur);
        }
        catch(Exception e){
            throw e;
        }
        return livreur;
    }

    public Livreur findById(Long id) throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        Livreur livreur = null;
        try {
            livreur = em.find(Livreur.class, id);
        }
        catch(Exception e) {
            throw e;
        }
        return livreur;
    }


    public List<Livreur> findAll() throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
```

```
54            List<Livreur> livreurs = null;
55            try {
56                Query q = em.createQuery("SELECT l FROM Livreur l");
57                livreurs = (List<Livreur>) q.getResultList();
58            }
59            catch(Exception e) {
60                throw e;
61            }
62            return livreurs;
63        }
64
65        public List<Livreur> findAllDisponible() throws Throwable
66        {
67            EntityManager em = JpaUtil.obtenirEntityManager();
68            List<Livreur> livreurs = null;
69            try {
70                Query q = em.createQuery("SELECT l FROM Livreur l WHERE "
71                        + "l.disponible = TRUE");
72                livreurs = (List<Livreur>) q.getResultList();
73            }
74            catch(Exception e) {
75                throw e;
76            }
77            return livreurs;
78        }
79
80        public List<Livreur> findAllDisponible(Double commandeWeight)
          throws Throwable
81        {
82            EntityManager em = JpaUtil.obtenirEntityManager();
83            List<Livreur> livreurs = null;
84            try {
85                Query q = em.createQuery("SELECT l FROM Livreur l WHERE "
86                        + "l.disponible = TRUE AND l.chargeMaxEnGrammes >=
                          :commandeWeight");
87                q.setParameter("commandeWeight", commandeWeight);
88                livreurs = (List<Livreur>) q.getResultList();
89            }
90            catch(Exception e) {
91                throw e;
92            }
93            return livreurs;
94        }
95    }
96
```

```java
package fr.insalyon.dasi.gustatif.dao;

import fr.insalyon.dasi.gustatif.metier.modele.Commande;
import fr.insalyon.dasi.gustatif.metier.modele.LigneDeCommande;
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Query;

/*
 * 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.
 */

/**
 *
 * @author Nico
 */
public class LigneDeCommandeDao {

    public void create(LigneDeCommande ligneDeCommande) throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        try {
            em.persist(ligneDeCommande);
        }
        catch(Exception e) {
            throw e;
        }
    }

    public LigneDeCommande update(LigneDeCommande ligneDeCommande)
            throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        try {
            ligneDeCommande = em.merge(ligneDeCommande);
        }
        catch(Exception e){
            throw e;
        }
        return ligneDeCommande;
    }

    public LigneDeCommande findById(Long id) throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        LigneDeCommande ligneDeCommande = null;
        try {
            ligneDeCommande = em.find(LigneDeCommande.class, id);
        }
        catch(Exception e) {
            throw e;
        }
        return ligneDeCommande;
```

```
53      }
54
55      public List<LigneDeCommande> findByCommandeID(Long commandeId)
56              throws Throwable {
57          EntityManager em = JpaUtil.obtenirEntityManager();
58          List<LigneDeCommande> lignesDeCommande = null;
59
60          try {
61              Commande c = em.find(Commande.class, commandeId);
62              if(c != null)
63              {
64                  lignesDeCommande = c.getLignesDeCommande();
65                  if(lignesDeCommande.isEmpty())
66                  {
67                      lignesDeCommande = null;
68                  }
69              }
70          }
71          catch(Exception e) {
72              throw e;
73          }
74          return lignesDeCommande;
75      }
76
77      public List<LigneDeCommande> findAll() throws Throwable {
78          EntityManager em = JpaUtil.obtenirEntityManager();
79          List<LigneDeCommande> lignesDeCommande = null;
80          try {
81              Query q = em.createQuery("SELECT l FROM LigneDeCommande l");
82              lignesDeCommande = (List<LigneDeCommande>) q.getResultList();
83          }
84          catch(Exception e) {
85              throw e;
86          }
87          return lignesDeCommande;
88      }
89  }
90
```

```java
package fr.insalyon.dasi.gustatif.dao;

import fr.insalyon.dasi.gustatif.metier.modele.LivreurDrone;
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Query;

/*
 * 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.
 */

/**
 *
 * @author Nico
 */
public class LivreurDroneDao extends LivreurDao {

    public List<LivreurDrone> findAllLivreursDrones() throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        List<LivreurDrone> livreurs = null;
        try {
            Query q = em.createQuery("SELECT l FROM LivreurDrone l");
            livreurs = (List<LivreurDrone>) q.getResultList();
        }
        catch(Exception e) {
            throw e;
        }
        return livreurs;
    }

    public List<LivreurDrone> findAllLivreursDronesDisponibles() throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        List<LivreurDrone> livreurs = null;
        try {
            Query q = em.createQuery("SELECT l FROM LivreurDrone l WHERE "
                    + "l.disponible = TRUE");
            livreurs = (List<LivreurDrone>) q.getResultList();
        }
        catch(Exception e) {
            throw e;
        }
        return livreurs;
    }

    public List<LivreurDrone> findAllLivreursDronesDisponibles(Double commandeWeight)
            throws Throwable
    {
        EntityManager em = JpaUtil.obtenirEntityManager();
```

```
51          List<LivreurDrone> livreurs = null;
52          try {
53              Query q = em.createQuery("SELECT l FROM LivreurDrone l
                WHERE "
54                      + "l.disponible = TRUE AND l.chargeMax >=
                        :commandeWeight");
55              q.setParameter("commandeWeight", commandeWeight);
56              livreurs = (List<LivreurDrone>) q.getResultList();
57          }
58          catch(Exception e) {
59              throw e;
60          }
61          return livreurs;
62      }
63
64      public List<LivreurDrone>
        findLivreursDronesByMailAndPassword(String mail, String password)
65              throws Throwable {
66          EntityManager em = JpaUtil.obtenirEntityManager();
67          List<LivreurDrone> livreurs = null;
68          try {
69              Query q = em.createQuery("SELECT l FROM LivreurDrone l
                WHERE l.mail = :mail"
70                      + " AND l.motDePasse = :password");
71              q.setParameter("mail", mail);
72              q.setParameter("password", password);
73              livreurs = (List<LivreurDrone>) q.getResultList();
74              if(livreurs.isEmpty()) {
75                  livreurs = null;
76              }
77          }
78          catch(Exception e) {
79              throw e;
80          }
81
82          return livreurs;
83      }
84  }
85
```

```java
/*
 * 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 fr.insalyon.dasi.gustatif.dao;

import fr.insalyon.dasi.gustatif.metier.modele.LivreurVelo;
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Query;

/**
 *
 * @author Nico
 */
public class LivreurVeloDao extends LivreurDao {

    public List<LivreurVelo> findAllLivreursVelos() throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        List<LivreurVelo> livreurs = null;
        try {
            Query q = em.createQuery("SELECT l FROM LivreurVelo l");
            livreurs = (List<LivreurVelo>) q.getResultList();
        }
        catch(Exception e) {
            throw e;
        }
        return livreurs;
    }

    public List<LivreurVelo> findAllLivreursVelosDisponibles() throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        List<LivreurVelo> livreurs = null;
        try {
            Query q = em.createQuery("SELECT l FROM LivreurVelo l WHERE "
                    + "l.disponible = TRUE");
            livreurs = (List<LivreurVelo>) q.getResultList();
        }
        catch(Exception e) {
            throw e;
        }
        return livreurs;
    }

    public List<LivreurVelo> findAllLivreursVelosDisponibles(Double commandeWeight)
            throws Throwable
    {
        EntityManager em = JpaUtil.obtenirEntityManager();
        List<LivreurVelo> livreurs = null;
        try {
```

```
52              Query q = em.createQuery("SELECT l FROM LivreurDrone l
                WHERE "
53                      + "l.disponible = TRUE AND l.chargeMax >=
                        :commandeWeight");
54              q.setParameter("commandeWeight", commandeWeight);
55              livreurs = (List<LivreurVelo>) q.getResultList();
56          }
57          catch(Exception e) {
58              throw e;
59          }
60          return livreurs;
61      }
62
63      public LivreurVelo findLivreurVeloByMailAndPassword(String mail,
        String password)
64              throws Throwable {
65          EntityManager em = JpaUtil.obtenirEntityManager();
66          List<LivreurVelo> livreurs = null;
67          LivreurVelo lv = null;
68          try {
69              Query q = em.createQuery("SELECT l FROM LivreurVelo l WHERE "
70                      + "l.mail = :mail AND l.motDePasse = :password");
71              q.setParameter("mail", mail);
72              q.setParameter("password", password);
73              livreurs = (List<LivreurVelo>) q.getResultList();
74          }
75          catch(Exception e) {
76              throw e;
77          }
78          try {
79              lv = livreurs.get(0);
80          } catch (Exception e){
81          }
82          return lv;
83      }
84  }
85
```

```java
/*
 * 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 fr.insalyon.dasi.gustatif.dao;

import fr.insalyon.dasi.gustatif.metier.modele.Commande;
import fr.insalyon.dasi.gustatif.metier.modele.Livreur;
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Query;

/**
 *
 * @author Nico
 */
public class CommandeDao {
        public void create(Commande commande) throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        try {
            em.persist(commande);
        }
        catch(Exception e) {
            throw e;
        }
    }

    public Commande update(Commande commande) throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        try {
            commande = em.merge(commande);
        }
        catch(Exception e){
            throw e;
        }
        return commande;
    }

    public Commande findById(Long id) throws Throwable {
        EntityManager em = JpaUtil.obtenirEntityManager();
        Commande commande = null;
        try {
            commande = em.find(Commande.class, id);
        }
        catch(Exception e) {
            throw e;
        }
        return commande;
    }


    public Commande findNotEndedByLivreur(Livreur livreur) throws
```

```
      Throwable {
54        EntityManager em = JpaUtil.obtenirEntityManager();
55        Commande commande = null;
56        List<Commande> commandes = null;
57        try {
58            Query q = em.createQuery("SELECT c FROM Commande c WHERE
              c.dateFin "
59                    + "IS NULL AND c.livreur = :livreur");
60
61            q.setParameter("livreur", livreur);
62            commandes = (List<Commande>) q.getResultList();
63            if(!commandes.isEmpty()) {
64                commande = commandes.get(0);
65            }
66        }
67        catch(Exception e) {
68            throw e;
69        }
70        return commande;
71    }
72
73    public List<Commande> findAll() throws Throwable {
74        EntityManager em = JpaUtil.obtenirEntityManager();
75        List<Commande> commandes = null;
76        try {
77            Query q = em.createQuery("SELECT c FROM Commande c");
78            commandes = (List<Commande>) q.getResultList();
79        }
80        catch(Exception e) {
81            throw e;
82        }
83        return commandes;
84    }
85
86    public List<Commande> findByLivreurID(Long livreurId) throws
      Throwable {
87        EntityManager em = JpaUtil.obtenirEntityManager();
88        List<Commande> commandes = null;
89
90        try {
91            Livreur l = em.find(Livreur.class, livreurId);
92            if(l != null)
93            {
94                commandes = l.getCommandes();
95                if(commandes.isEmpty())
96                {
97                    commandes = null;
98                }
99            }
100       }
101       catch(Exception e) {
102           throw e;
103       }
104       return commandes;
```

```
105         }
106 
107         public List<Commande> findAllNotEnded() throws Throwable {
108             EntityManager em = JpaUtil.obtenirEntityManager();
109             List<Commande> commandes = null;
110             try {
111                 Query q = em.createQuery("SELECT c FROM Commande c WHERE "
112                         + "c.dateFin IS NULL");
113                 commandes = (List<Commande>) q.getResultList();
114             }
115             catch(Exception e) {
116                 throw e;
117             }
118             return commandes;
119         }
120     }
121 
```