

```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package fr.insalyon.dasi.gustatif.dao;
7
8  import fr.insalyon.dasi.gustatif.metier.modele.Commande;
9  import fr.insalyon.dasi.gustatif.metier.modele.Livreur;
10 import java.util.List;
11 import javax.persistence.EntityManager;
12 import javax.persistence.Query;
13
14 /**
15  *
16  * @author Nico
17  */
18 public class CommandeDao {
19     public void create(Commande commande) throws Throwable {
20         EntityManager em = JpaUtil.obtenirEntityManager();
21         try {
22             em.persist(commande);
23         }
24         catch (Exception e) {
25             throw e;
26         }
27     }
28
29     public Commande update(Commande commande) throws Throwable {
30         EntityManager em = JpaUtil.obtenirEntityManager();
31         try {
32             commande = em.merge(commande);
33         }
34         catch (Exception e) {
35             throw e;
36         }
37         return commande;
38     }
39
40     public Commande findById(Long id) throws Throwable {
41         EntityManager em = JpaUtil.obtenirEntityManager();
42         Commande commande = null;
43         try {
44             commande = em.find(Commande.class, id);
45         }
46         catch (Exception e) {
47             throw e;
48         }
49         return commande;
50     }
51
52     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 "
                                     + "IS NULL AND c.livreur = :livreur");
59
60             q.setParameter("livreur", livreur);
61             commandes = (List<Commande>) q.getResultList();
62             if(!commandes.isEmpty()) {
63                 commande = commandes.get(0);
64             }
65         }
66         catch(Exception e) {
67             throw e;
68         }
69         return commande;
70     }
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
```