

PaymentRepository.java

```
1 package model.repository;
2
3 import java.io.Serializable;
4
5
6
7
8
9
10
11
12
13
14
15
16
17 public class PaymentRepository implements ServletContextListener {
18
19     @PersistenceUnit(unitName = "store-pu")
20     private EntityManagerFactory emf;
21
22     private EntityManager em;
23
24     public PaymentRepository() {
25         emf = (EntityManagerFactory) Persistence.createEntityManagerFactory
26         ("store-pu");
27         System.out.println("[SERVLET-TEST-INF0]: PaymentRepository
28         Constructor");
29     }
30
31     @Override
32     public void contextDestroyed(ServletContextEvent sce) {
33         if (emf.isOpen()) {
34             emf.close();
35         }
36     }
37
38     @Override
39     public void contextInitialized(ServletContextEvent sce) {
40         ServletContext context = sce.getServletContext();
41         context.setAttribute("paymentRepository", this);
42         System.out.println("[SERVLET-TEST-INF0]: Contexto inicializado!
43         PaymentRepository");
44     }
45
46     public final EntityManager entityManager() {
47         if (em == null || !em.isOpen()) {
48             em = emf.createEntityManager();
49         }
50         return em;
51     }
52
53     public void persistOrMerge(Serializable entity, Serializable id) {
54         em = entityManager();
55
56         if (entity == null) {
57             throw new IllegalArgumentException("entity");
58         }
59         try {
60             em.getTransaction().begin();
61             if (id == null) {
62                 em.persist(entity);
63             } else {
64                 em.merge(entity);
65             }
66         }
67     }
68 }
```

PaymentRepository.java

```
63         em.getTransaction().commit();
64     } finally {
65         em.close();
66     }
67 }
68
69     public Payment findPaymentById(Long id) {
70         em = entityManager();
71         return em.find(Payment.class, id);
72     }
73
74     public List findAllPayments() {
75         em = entityManager();
76         Query q = em.createQuery("select obj from Payment obj");
77         return q.getResultList();
78     }
79
80     public Payment newPayment() {
81         Payment object = new Payment();
82
83         persistOrMerge(object, object.getId());
84         return object;
85     }
86 }
```