

PaymentRepository.java

```

1 package model.repository;
2
3 import java.io.Serializable;
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-INFO]: PaymentRepository
28 Constructor");
29     }
30
31     public PaymentRepository(EntityManagerFactory emf, EntityManager em) {
32         this.emf = emf;
33         this.em = em;
34         System.out.println("[SERVLET-TEST-INFO]: PaymentRepository
35 Constructor");
36     }
37
38     @Override
39     public void contextDestroyed(ServletContextEvent sce) {
40         if (emf.isOpen()) {
41             emf.close();
42         }
43     }
44
45     @Override
46     public void contextInitialized(ServletContextEvent sce) {
47         ServletContext context = sce.getServletContext();
48         context.setAttribute("paymentRepository", this);
49         System.out.println("[SERVLET-TEST-INFO]: Contexto inicializado!
50 PaymentRepository");
51     }
52
53     public final EntityManager entityManager() {
54         if (em == null || !em.isOpen()) {
55             em = emf.createEntityManager();
56         }
57         return em;
58     }
59
60     public void persistOrMerge(Serializable entity, Serializable id) {
61         em = entityManager();
62
63         if (entity == null) {
64             throw new IllegalArgumentException("entity");
65         }

```

PaymentRepository.java

```
62     try {
63         em.getTransaction().begin();
64         if (id == null) {
65             em.persist(entity);
66         } else {
67             em.merge(entity);
68         }
69         em.getTransaction().commit();
70     } finally {
71         em.close();
72     }
73 }
74
75 public Payment findPaymentById(Long id) {
76     em = entityManager();
77     return em.find(Payment.class, id);
78 }
79
80 public List findAllPayments() {
81     em = entityManager();
82     Query q = em.createQuery("select obj from Payment obj");
83     return q.getResultList();
84 }
85
86 public Payment newPayment() {
87     Payment object = new Payment();
88
89     persistOrMerge(object, object.getId());
90     return object;
91 }
92 }
```