

## OrderRepository.java

```

1 package model.repository;
2
3 import java.io.Serializable;
4
16
17 public class OrderRepository implements ServletContextListener {
18
19     @PersistenceUnit(unitName = "store-pu")
20     private EntityManagerFactory emf;
21
22     private EntityManager em;
23
24     public OrderRepository() {
25         emf = (EntityManagerFactory) Persistence.createEntityManagerFactory
26         ("store-pu");
27         System.out.println("[SERVLET-TEST-INFO]: OrderRepository
28         Constructor");
29     }
30
31     public OrderRepository(EntityManagerFactory emf, EntityManager em) {
32         this.emf = emf;
33         this.em = em;
34         System.out.println("[SERVLET-TEST-INFO]: OrderRepository
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("orderRepository", this);
49         System.out.println("[SERVLET-TEST-INFO]: Contexto inicializado!
50         OrderRepository");
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         if (entity == null) {
63             throw new IllegalArgumentException("entity");
64         }
65     }

```

## OrderRepository.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 Order findOrderById(Long id) {
76     em = entityManager();
77     return em.find(Order.class, id);
78 }
79
80 public List<Order> findAllOrders() {
81     em = entityManager();
82     Query q = em.createQuery("select o from Order o");
83     return q.getResultList();
84 }
85
86 public Order newOrder() {
87     Order o = new Order();
88     persistOrMerge(o, o.getId());
89
90     return o;
91 }
92 }
```