

## TÍCH HỢP HIBERNATE

ThS. Nguyễn Nghiệm 0913.745.789 - NghiemN@fpt.edu.vn

### Nội dung

- Giới thiệu
- Tích hợp Hibernate
- Cơ chế điều khiển transaction
- Hoàn thiện AccountController

### GIỚI THIỆU

#### Spring AOP

AOP infrastructure Metadata support Declarative transaction management

#### Spring ORM

Hibernate support iBatis support JDO support

#### Spring DAO

Transaction Infrastructure JDBC support DAO support

#### **Spring Web**

WebApplicationContext Struts integration Tiles integration Web utilities

#### Spring Context

ApplicationContext JNDI, EJB support Remoting

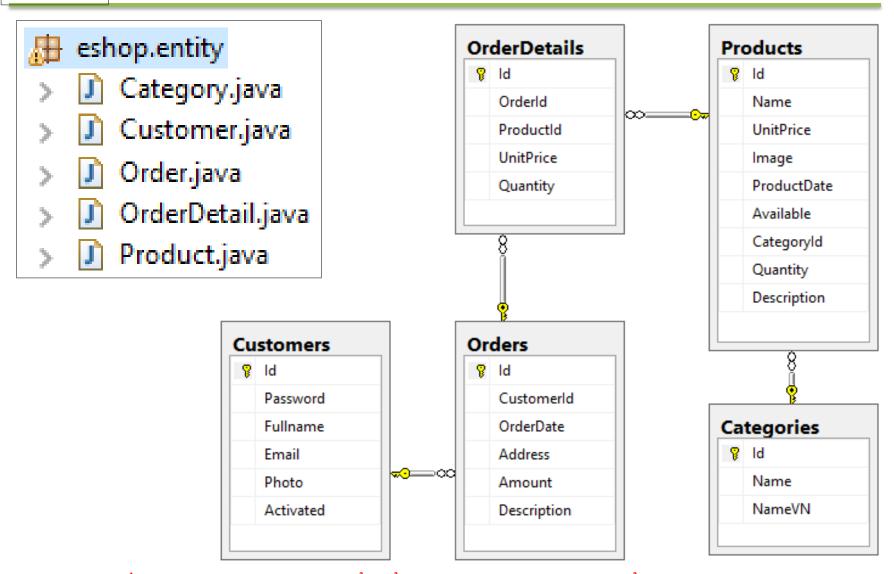
#### Spring MVC

Web MVC
Framework
JSP support
Velocity/FreeMarker
support
PFD/Excel support

#### **Spring Core**

IoC Container

#### REVIEW ESHOPV10



Xem lại phần bài học Hibernate để hiểu rõ việc ánh xạ thực thể

#### REVIEW HIBERNATE

- Khi làm việc với Hibernate chúng ta cần
  - \* Hibernate.cfg.xml
  - \*\*Entity class
- Lập trình với Hibernate
  - \*Nạp file cấu hình
  - ★ Tạo SessionFactory
  - **☀Tao Session**
  - Truy vấn và thao tác

### TÍCH HỢP HIBERNATE

- Khai báo thông số kết nối, SessionFactory và Transaction trong file cấu hình
- Tiêm SessionFactory vào Controller để sử dụng
- Lập trình truy vấn và thao tác như cũ
- Điều khiển transaction chung hoặc riêng tùy vào tình huống
- Sử dụng chung một session sẽ tang tính performance của ứng dụng

### Khai báo cấu hình hibernate

```
<!-- DataSource -->
<been id="dataSource"</pre>
     class="org.springframework.jdbc.datasource.DriverManagerDataSource"
           p:driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
           p:url="jdbc:sqlserver://localhost; Database=EShopV10"
           p:username='sa" p:password='songlong'>
</bean>
                                                 Thông số kết nối CSDL
<!-- Hibernate Session Factory -->
<bean id="sessionFactory"</p>
     class="org.springframework.orm.hibernate4.LocalSessionFactoryBean">
 property name="dataSource" ref="dataSource"/>
 property name="hibernateProperties">
  props>
     prop key="hibernate.show_sql">false
   </props>
 </property>
 </bean>
                                                Package chứa entity class
<!-- Transaction Manager -->
<tx:annotation-driven transaction-manager="transactionManager" />
<bean id="transactionManager"</p>
```

class="org.springframework.orm.hibernate4.HibernateTransactionManager"

p:sessionFactory-ref="sessionFactory" />

### Tự ĐIỀU KHIỂN TRANSACTION

```
@Controller
@RequestMapping("account")
public class AccountController {
      @Autowired
                                      Tiêm SessionFactory
      SessionFactory factory;
      @RequestMapping(value="register", method=RequestMethod.POST)
      public String register(ModelMap model,
                  @ModelAttribute("user") Customer user) {
            Session session = factory.openSession();
            Transaction transaction = session.beginTransaction();
            try {
                  session.save(user);
                  transaction.commit();
                  model.addAttribute("message", "Đăng ký thành công!");
            catch (Exception e) {
                  transaction.rollback();
                  model.addAttribute("message", "Đăng ký thất bại !");
                  e.printStackTrace();
            session.close();
            return "register";
      }
```

### TRANSACTION TỰ ĐỘNG THEO ACTION

```
@Controller
@RequestMapping("account")
public class AccountController {
      @Autowired
                                   Tự động điều khiển
     SessionFactory factory;
                               transaction cho action này
      @Transactiona
      @RequestMapping(value="register", method=RequestMethod.POST)
     public String register(ModelMap model,
                  @ModelAttribute("user") Customer user) {
           Session session = factory.getCurrentSession();
           session.save(user);
           return "register";
```

#### Transaction tự động tất cả action

```
@Controller
                                   Tự động điều khiển transaction
@RequestMapping("account")
@Transactional
                                         cho tất cả các action
public class AccountController {
      @Autowired
     SessionFactory factory;
     @RequestMapping(value="register", method=RequestMethod.POST)
     public String register(ModelMap model,
                  @ModelAttribute("user") Customer user) {
           Session session = factory.getCurrentSession();
           session.save(user);
           return "register";
      @RequestMapping(value="update", method=RequestMethod.POST)
     public String update(ModelMap model,
                  @ModelAttribute("user") Customer user) {
           Session session = factory.getCurrentSession();
           session.update(user);
           return "register";
```

# Hoàn thiện AccountController với các action sau

- **☀**Đăng nhập
- \*Đăng ký có upload hình và gửi email kích hoạt tài khoản
- \*Kích hoạt tài khoản sau đăng ký
- \*Quên mật khẩu gửi mật khẩu qua email
- **☀**Đăng xuất
- \*Đổi mật khẩu
- \*Sửa đổi thông tin tài khoản tài khoản

### TÓM TẮT

- Giới thiệu
- Tích hợp Hibernate
- Cơ chế điều khiển transaction
- Hoàn thiện AccountController

