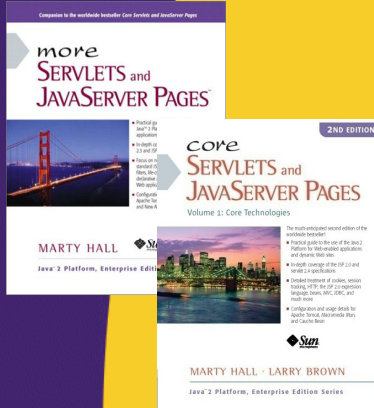




Spring Integration Part 2

Originals of Slides and Source Code for Examples:
<http://courses.coreservlets.com/Course-Materials/spring.html>

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



**For live Spring & Hibernate training, see
courses at <http://courses.coreservlets.com/>.**



**Taught by the experts that brought you this tutorial.
Available at public venues, or customized versions
can be held on-site at your organization.**

- Courses developed and taught by Marty Hall
 - Java 5, Java 6, intermediate/beginning servlets/JSP, advanced servlets/JSP, Struts, JSF, Ajax, GWT, custom mix of topics
- Courses developed and taught by coreservlets.com experts (edited by Marty)
 - Spring, Hibernate/JPA, EJB3, Ruby/Rails

Contact hall@coreservlets.com for details

Topics in This Section

- **Managing transactions with Spring**
- **Creating a transactional test environment**
- **Spring and JPA integration**
- **Spring, JPA, and JSF integration**

4

Java EE training: <http://courses.coreservlets.com>

© 2008 coreservlets.com



Managing Transactions with Spring

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Objectives

- Transaction instance
- Read consistency
- Error recovery

6

Java EE training: <http://courses.coreservlets.com>

Spring AOP Advice Transaction Example

```
<tx:advice id="transactionAdvice"
           transaction-manager="transactionManager">
  <tx:attributes>
    <tx:method name="*"
               propagation="REQUIRED"
               isolation="READ_COMMITTED"
               timeout="-1"
               read-only="false"
               rollback-for="java.lang.Throwable"
               no-rollback-for="NoTransactionException"/>
  </tx:attributes>
</tx:advice>
```

7

Java EE training: <http://courses.coreservlets.com>

Propagation Options

- **REQUIRED (default)**
 - Adopts existing transactions
 - Creates new transactions
- **SUPPORTS**
 - Does **not** create new transactions
 - If a transaction is present, adopts existing transactions
- **MANDATORY**
 - Does not create new transactions
 - Adopts existing transactions
 - Throws an error, if a transaction is not present
- **NEVER**
 - Throws an error, if a transaction is present

8

Java EE training: <http://courses.coreservlets.com>

Propagation Options Continued

- **NESTED**
 - Adopts existing transactions
 - Creates new transactions
 - Commits to savepoints
 - **REQUIRES_NEW ***
 - Creates a new transaction for each execution scope
 - **Suspends** existing transactions
 - **NOT_SUPPORTED ***
 - **Suspends** existing transactions
- * Requires JTA integration**
- See `org.springframework.transaction.jta.JtaTransactionManager`

9

Java EE training: <http://courses.coreservlets.com>

Isolation

- **DEFAULT**
 - Driver dependent
- **READ_UNCOMMITTED**
 - Read uncommitted information
- **READ_COMMITTED**
 - Exposes committed information
- **REPEATABLE_READ**
 - Provides consistent row information for each read
- **SERIALIZABLE**
 - Performs a full table lock
 - Guaranteed to deliver consistent query results
 - Two queries executed at different times, sharing the same transaction scope

10

Java EE training: <http://courses.coreservlets.com>

Error Recovery

- **Recovery options**
 - Rollback
 - No rollback
- **Mapped to error type**
 - Defaults to `java.lang.RuntimeException`

```
<tx:method name="*"
    rollback-for="java.lang.Throwable"
    no-rollback-for="NoTransactionException"/>
```

11

Java EE training: <http://courses.coreservlets.com>

Other Options

- **Timeout**
 - Transaction time ceiling in seconds
 - Defaults to none; i.e., **-1**
- **Read-only**
 - Determines if changes to the database are permitted
 - Defaults to **false**

@Transactional

- **Class or method annotation**
 - **@Transactional**
- **Annotation properties**
 - Fine-grained settings

```
@Transactional(  
    propagation=Propagation.REQUIRES_NEW,  
    isolation=Isolation.READ_COMMITTED)
```

@Transactional Example

- Annotated source

```
@Transactional(propagation=Propagation.REQUIRES_NEW,  
               isolation=Isolation.READ_COMMITTED)  
public class DAO{  
}
```

- Bean definition

```
<beans>  
  <tx:annotation-driven  
    transactionManager="transactionManager"/>  
</beans>
```

Spring Transaction Implication

- Coordinate direct **DataSource** access with transaction management
 - Acquire connections via **DataSourceUtils** API
 - e.g., static method **getConnection(dataSource)**



Creating a Transactional Test Environment

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

General Approach

- **Provide testing bean definitions**
 - Mock beans
 - Data source
 - Transaction manager
 - Test support beans
 - JDBC template
- **Create test class**
- **Integrate transaction management**
 - `@TransactionConfiguration`
 - `@Transactional`
- **Integrate transaction management support**

Dependencies

- **JDBC template support**
 - spring-jdbc, spring-tx, spring-core
- **AOP support**
 - spring-aop, aspectjweaver, aopalliance
- **IoC integration testing**
 - spring-test, spring-core, spring-beans, spring-context

Test Bean Definitions

- **classpath:/coreservletsTestContext.xml**

```
<bean id="dataSource"
      class="coreservlets.EmbeddedDerbyDataSource">
  <constructor-arg value="target/ngcdb" />
  <constructor-arg>
    <list><value>/setup.sql</value></list>
  </constructor-arg>
</bean>
<bean id="simpleJdbcTemplate"
      class="org.springframework.jdbc.core.simple...
      SimpleJdbcTemplate">
  <constructor-arg ref="dataSource" />
</bean>
```

Library Bean Definition

- `classpath:/coreservletsContext.xml`

```
<bean id="customerDao"
      class="coreservlets.SpringJdbcCustomerDao">
  <constructor-arg ref="dataSource" />
</bean>

<bean id="transactionManager"
      class="org.springframework.jdbc.datasource...
          DataSourceTransactionManager">
  <property name="dataSource" ref="dataSource" />
</bean>
```

Library Bean Definitions Continued

```
<aop:config>
  <aop:pointcut id="customerDaoOperations"
               expression="target (coreservlets.CustomerDao)" />
  <aop:advisor advice-ref="transactionAdvice"
              pointcut-ref="customerDaoOperations" />
</aop:config>

<tx:advice id="transactionAdvice"
           transaction-manager="transactionManager">
  <tx:attributes>
    <tx:method name="get*" propagation="REQUIRED"
              read-only="true" />
    <tx:method name="*" propagation="REQUIRED" />
  </tx:attributes>
</tx:advice>
```

Test Class

- **coreservlets.CustomerDaoIntegrationTest**

```
public class CustomerDaoIntegrationTest {  
}
```

- **Integrate Spring container**

```
@RunWith(SpringJUnit4ClassRunner.class)  
public class CustomerDaoIntegrationTest {  
}
```

- **Integrate Spring bean definitions**

```
@ContextConfiguration(locations={  
    "/coreservletsContext.xml", "/coreservletsTestContext.xml"})  
@RunWith(SpringJUnit4ClassRunner.class)  
public class CustomerDaoIntegrationTest {  
}
```

22

Java EE training: <http://courses.coreservlets.com>

Integrate Transaction Management

```
@ContextConfiguration(locations={  
    "/coreservletsContext.xml", "/coreservletsTestContext.xml"})  
@RunWith(SpringJUnit4ClassRunner.class)  
@TransactionConfiguration(  
    transactionManager="transactionManager", defaultRollback=true)  
@Transactional  
public class CustomerDaoIntegrationTest {  
}
```

23

Java EE training: <http://courses.coreservlets.com>

Inject Test Fixtures

```
@ContextConfiguration(locations={
    "/coreservletsContext.xml", "/coreservletsTestContext.xml"})
@RunWith(SpringJUnit4ClassRunner.class)
@TransactionalConfiguration(
    transactionManager="transactionManager", defaultRollback=true)
@Transactional
public class CustomerDaoIntegrationTest {

    @Resource
    private CustomerDao service;

    @Resource
    private SimpleJdbcTemplate jdbc;
}
```

24

Java EE training: <http://courses.coreservlets.com>

Leverage Transaction Support

```
public class CustomerDaoIntegrationTest {
    ...
    @BeforeTransaction
    @AfterTransaction
    public void beforeAndAfterTx() {
        assertCount(2);
    }

    private void assertCustomerCount(int expectedCount){
        int storedCount = jdbc.queryForInt("select count(*) from
        customer");
        assertEquals(expectedCount, storedCount);
    }
}
```

25

Java EE training: <http://courses.coreservlets.com>

Develop Test Cases

```
@Test(expected=IllegalArgumentException.class)
public void verifyUniqueConstraint(){
    service.insert(new Customer("t-id","t-name"));
    service.insert(new Customer("t-id","t-name"));
}
```

26

Java EE training: <http://courses.coreservlets.com>

© 2008 coreservlets.com



Spring and JPA

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Summary

- Leverage spring-orm
- Configure EntityManager
- Integrate JPA persistence unit
- Integrate JPA vendor; e.g. Hibernate
- Inject EntityManager into business beans
- Integrate Spring transaction management

Process

- **Map class to table**
 - JPA annotations; e.g., **@Entity**, **@Table**, **@Id**, **@Column**,
- **Implement DAO**
 - Define and/or inject **javax.persistence.EntityManager**
 - `@PersistenceContext(unitName="unitName")`
 - Implement business methods
 - Use `javax.persistence.EntityManager`
- **Create DAO bean definition**
 - `<bean/>`
- **Add Spring ORM JAR**
- **Create persistence.xml**
 - `classpath:/META-INF/persistence.xml`

Process Continued

- **Configure JPA EntityManager**
 - Persistence unit name
 - JPA vendor
 - e.g., Hibernate with RDBMS client dialect
- **Integrate transaction**
 - Transaction advice
 - `<tx:advice/>`
 - Transaction manager
 - `PlatformTransactionManager`
 - Aspect
 - Define pointcut referencing application
 - Define aspect applying TX advice to application
- **Register bean post processor**
 - `PersistenceAnnotationBeanPostProcessor`

30

Java EE training: <http://courses.coreservlets.com>

Map Database Table

```
import javax.persistence.*;

@Entity @Table
public class Customer {

    @Id @Column
    private String id;

    @Column(nullable=false,unique=true,length=64)
    private String name;

    ...
}
```

31

Java EE training: <http://courses.coreservlets.com>

Implement DAO

```
import javax.persistence.*;

public class JpaCustomerDao implements CustomerDao {

    @PersistenceContext(unitName="coreservlets")
    private EntityManager entityManager;

    public Customer getCustomerByName(String customerName) {
        try{
            return (Customer) this.entityManager.createQuery(
                "select c from Customer c"
                + " where c.name = :customerName")
                .setParameter("customerName", customerName)
                .getSingleResult();
        }
        catch(NoResultException e){
            return null;
        }
    }
}
```

32

Java EE training: <http://courses.coreservlets.com>

Create DAO Bean Definition

- **src/main/resources**
 - **coreservletsContext.xml**

```
<bean id="customerDao"
      class="coreservlets.JpaCustomerDao" />
```

33

Java EE training: <http://courses.coreservlets.com>

Register Spring ORM Dependency

- **pom.xml**

```
<dependency>  
  <groupId>org.springframework</groupId>  
  <artifactId>spring-orm</artifactId>  
  <version>2.5.5</version>  
</dependency>
```

Create persistence.xml

- **src/main/resources**
 - **coreservletsContext.xml**

```
<persistence xmlns="http://java.sun.com/xml/ns/persistence"  
  version="1.0">  
  <persistence-unit name="coreservlets"  
    transaction-type="RESOURCE_LOCAL">  
    <class>coreservlets.Customer</class>  
  </persistence-unit>  
</persistence>
```

Configure JPA EntityManager

```
<bean id="entityManagerFactory"
  class="org.springframework.orm.jpa...
    LocalContainerEntityManagerFactoryBean">
  <property name="persistenceUnitName"
    value="coreservlets"/>
  <property name="jpaVendorAdapter">
    <bean class="org.springframework.orm.jpa.vendor...
      HibernateJpaVendorAdapter">
      <property name="databasePlatform"
        value="org.hibernate.dialect.DerbyDialect"/>
    </bean>
  </property>
  <property name="dataSource" ref="dataSource"/>
  <property name="loadTimeWeaver">
    <bean class="org.springframework.instrument.classloading...
      InstrumentationLoadTimeWeaver"/>
  </property>
</bean>
```

36

Java EE training: <http://courses.coreservlets.com>

Integrate Transaction Management

- Transaction manager

```
<bean id="transactionManager"
  class="org.springframework.orm.jpa.
    JpaTransactionManager">
  <property name="entityManagerFactory"
    ref="entityManagerFactory"/>
  <property name="dataSource" ref="dataSource"/>
</bean>
```

37

Java EE training: <http://courses.coreservlets.com>

Integrate Transaction Management

- Transaction advice

```
<tx:advice id="transactionAdvice"
           transaction-manager="transactionManager">
  <tx:attributes>
    <tx:method name="get*"
               propagation="REQUIRED" read-only="true" />
    <tx:method name="*" propagation="REQUIRED" />
  </tx:attributes>
</tx:advice>
```

Integrate Transaction Management

- Transaction aspects

```
<aop:config>
  <aop:pointcut id="customerDaoOperations"
               expression="target(coreservlets.CustomerDao)" />
  <aop:advisor advice-ref="transactionAdvice"
               pointcut-ref="customerDaoOperations" />
</aop:config>
```

Process Bean Annotations

- **Bean post processor**

```
<bean class="org.springframework.orm.jpa.support.  
    PersistenceAnnotationBeanPostProcessor"/>
```

40

Java EE training: <http://courses.coreservlets.com>

Execute Integration Tests

- **Compile, test, package, and install using Maven 2**

Standard output

```
coreservlets-jpa> mvn install  
-----  
  T E S T S  
-----  
Running coreservlets.CustomerDaoIntegrationTest  
Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed:  
9.831 sec  
Running coreservlets.CustomerDaoTransactionalTest  
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed:  
0.021 sec  
Results :  
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0  
  
[INFO] [install:install]  
[INFO] Installing coreservlets-jpa-1.0-SNAPSHOT.jar
```

41



Spring, JPA and JSF

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

General Approach

- **Create coreservlets-web**
 - Compile with **coreservlets-api**
 - Inject **coreservlets-jpa** at runtime
- **Integrate Spring into the web application**
- **Integrate Spring into JSF**

Process

- **Reference**
 - Compile-time dependency: `coreservlets-api`
 - Runtime dependency: `coreservlets-jpa`
- **Add Spring JARs**
 - `spring-web`
- **Edit web.xml**
 - Add servlet context parameters
 - Identifying bean definition paths
 - `<context-param/>`
 - Add web application listener callback
 - Initialize Spring context
 - `<listener/>`
- **Edit faces-config.xml**
 - Add Spring/JSF variable resolver

44

Java EE training: <http://courses.coreservlets.com>

Create Library References

```
<dependency>
  <groupId>${project.groupId}</groupId>
  <artifactId>coreservlets-api</artifactId>
  <version>${project.version}</version>
</dependency>
<dependency>
  <groupId>${project.groupId}</groupId>
  <artifactId>coreservlets-jpa</artifactId>
  <version>${project.version}</version>
</dependency>
```

45

Java EE training: <http://courses.coreservlets.com>

Edit web.xml

- **Create servlet context parameters**

- Define bean definition resource paths

```
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>
    WEB-INF/applicationContext.xml
    classpath:/coreservletsContext.xml
  </param-value>
</context-param>
```

- **Register application callback**

```
<listener><listener-class>
  org.springframework.web.context.ContextLoaderListener
</listener-class></listener>
```

46

Java EE training: <http://courses.coreservlets.com>

Edit faces-config.xml

- **Add JSF/Spring variable resolver**

- Resolver implementation depends on JSF version

```
<faces-config>
  <application>
    <variable-resolver>
      org.springframework.web.jsf.SpringBeanVariableResolver
    </variable-resolver>
  </application>
</faces-config>
```

47

Java EE training: <http://courses.coreservlets.com>

JSF Backing Bean Example

```
public class CustomerLookupBean {
    private CustomerDao customerDao;
    private String name;
    private Customer customer;
    public void setCustomerDao(CustomerDao customerDao){
        this.customerDao = customerDao;
    }
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public Customer getCustomer() { return customer; }
    public void setCustomer(Customer customer) {
        this.customer = customer;
    }
    public String lookupCustomer(){
        this.customer =
            this.customerDao.getCustomerByName(this.name);
        return null;
    }
}
```

48

Java EE training: <http://courses.coreservlets.com>

Bean Definitions

- **WEB-INF/applicationContext.xml**

```
<bean id="customerLookupBean"
      class="coreservlets.CustomerLookupBean"
      scope="session">
    <property name="customerDao" ref="customerDao"/>
    <aop:scoped-proxy/>
</bean>
<bean id="dataSource"
      class="coreservlets.dataSource.EmbeddedDerbyDataSource">
    <constructor-arg value="target/ngcdb" />
    <constructor-arg>
        <list>
            <value>/setup.sql</value>
        </list>
    </constructor-arg>
</bean>
</beans>
```

49

Java EE training: <http://courses.coreservlets.com>

JSP Form

```
<h:form>
  <h:outputText value="Name:" />

  <h:inputText value="#{customerLookupBean.name}" />

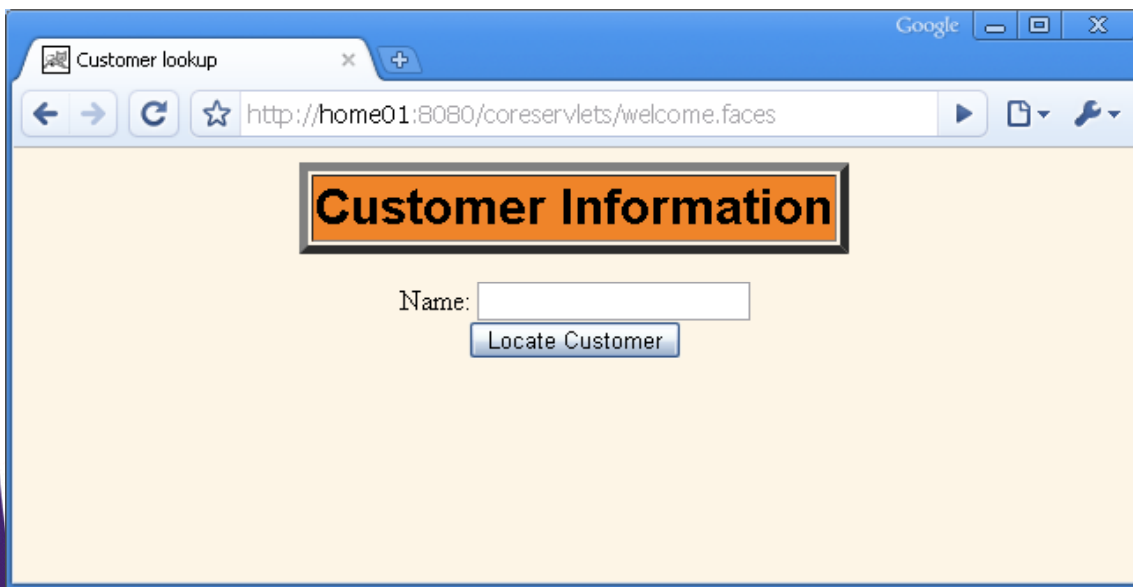
  <BR/>

  <h:commandButton value="Locate Customer"
    action="#{customerLookupBean.lookupCustomer}" />
</h:form>
```

JSP Results

```
<table border="1">
  <tr>
    <th>ID</th><th>NAME</th>
  </tr><tr>
    <td>
      <h:outputText
        value="#{customerLookupBean.customer.id}" />
    </td><td>
      <h:outputText
        value="#{customerLookupBean.customer.name}" />
    </td>
  </tr>
</table>
```

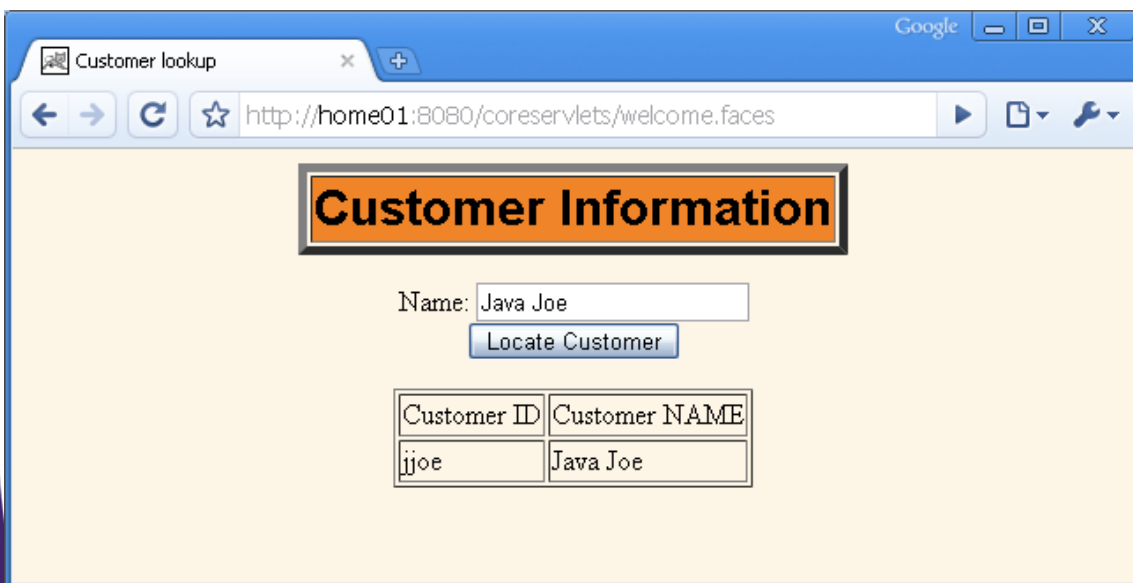
Form



52

Java EE training: <http://courses.coreservlets.com>

Results



53

Java EE training: <http://courses.coreservlets.com>

Special Considerations for Tomcat

- **Libraries**
 - cglib
- **Class loader**

<Loader

```
loaderClass="org.springframework.instrument.classloading.tomcat.TomcatInstrumentableClassLoader"
useSystemClassLoaderAsParent="false"/>
```



Wrap-up

Summary

- **Transaction management**
 - Spring AOP aspects with Spring TX advice implementations
 - `<tx:advice/>`
 - Annotations
 - `@Transactional`
 - `<tx:annotation-driven/>`
- **Transactional test environment**
 - Class annotations
 - `@TransactionalConfiguration`
 - `@Transactional`
 - Method support
 - `@BeforeTransaction`
 - `@AfterTransaction`

56

Java EE training: <http://courses.coreservlets.com>

Summary Continued

- **Spring and JPA integration**
 - Spring ORM module
 - `@PersistenceContext`
 - `EntityManagerFactoryBean`
 - Processes persistence.xml
 - Integrates JPA implementation (Hibernate) and data source
 - Transaction integration via Spring AOP/TX
- **Spring, JPA and JSF integration**
 - JPA integration
 - Runtime API implementation selection using `applicationContext.xml`
 - Integrate IoC container
 - Register bean definition `<context-param/>`
 - Register context loader `<listener/>`
 - Register variable resolver
 - `<variable-resolver/>`

57

Java EE training: <http://courses.coreservlets.com>



Questions?

Customized Java EE Training: <http://courses.coreservlets.com/>

Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.

Developed and taught by well-known author and developer. At public venues or onsite at *your* location.