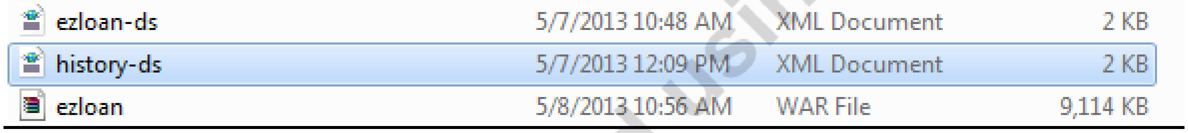
XA Transaction implementation using spring framework with JBoss

1. Create two different XA datasource



ezloan-ds	5/7/2013 10:48 AM	XML Document	2 KB
history-ds	5/7/2013 12:09 PM	XML Document	2 KB
ezloan	5/8/2013 10:56 AM	WAR File	9,114 KB

history-ds.xml

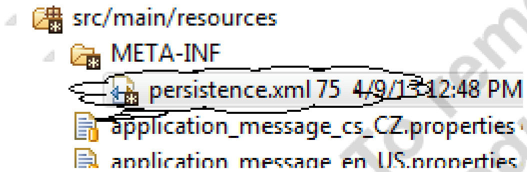
```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
   <xa-datasource>
      <jndi-name>JPAHistoryDS</jndi-name>
      <xa-datasource-class>com.mysql.jdbc.jdbc2.optional.MysqlXADataSource</xa-datasource-class>
      <xa-datasource-property name="URL">jdbc:mysql://localhost:3306/history_db</xa-datasource-property>
      <user-name>root</user-name>
      <password>root</password>
      <transaction-isolation>TRANSACTION_READ_COMMITTED</transaction-isolation>
      <min-pool-size>5</min-pool-size>
      <max-pool-size>100</max-pool-size>
      <blocking-timeout-millis>2000</blocking-timeout-millis>
      <idle-timeout-minutes>2</idle-timeout-minutes>
      <track-connection-by-tx />
      <valid-connection-checker-class-name>org.jboss.resource.adapter.jdbc.vendor.MySQLValidConnectionChecker</valid-connection-checker-class-name>
      <exception-sorter-class-name>org.jboss.resource.adapter.jdbc.vendor.MySQLExceptionSorter</exception-sorter-class-name>
      <!--new-connection-sql></new-connection-sql-->
      <!--check-valid-connection-sql></check-valid-connection-sql-->
      <metadata>
         <type-mapping>mySQL</type-mapping>
      </metadata>
   </xa-datasource>
        </datasources>
```

ezloann-ds.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
   <xa-datasource>
      <jndi-name>LoanAppDS</jndi-name>
      <xa-datasource-class>com.mysql.jdbc.jdbc2.optional.MysqlXADataSource</xa-datasource-class>
      <xa-datasource-property name="URL">jdbc:mysql://localhost:3306/ezloan</xa-datasource-property>
      <user-name>root</user-name>
      <password>root</password>
      <transaction-isolation>TRANSACTION_READ_COMMITTED</transaction-isolation>
      <min-pool-size>5</min-pool-size>
      <max-pool-size>100</max-pool-size>
      <blocking-timeout-millis>2000</blocking-timeout-millis>
      <idle-timeout-minutes>2</idle-timeout-minutes>
      <track-connection-by-tx />
      <valid-connection-checker-class-name>org.jboss.resource.adapter.jdbc.vendor.MySQLValidConnectionChecker</valid-connection-checker-class-name>
      <exception-sorter-class-name>org.jboss.resource.adapter.jdbc.vendor.MySQLExceptionSorter</exception-sorter-class-name>
      <!--new-connection-sql></new-connection-sql-->
      <!--check-valid-connection-sql></check-valid-connection-sql-->
      <metadata>
         <type-mapping>mySQL</type-mapping>
      </metadata>
   </xa-datasource>
        </datasources>
```

Step-2

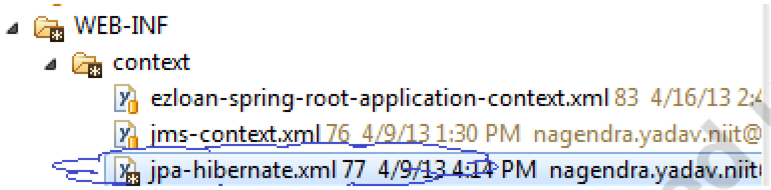
Create two persistence-unit inside the persistence.xml file.


src/main/resources
META-INF
persistence.xml 75 4/9/13 12:48 PM
application_message_cs_CZ.properties
application_message_en_US.properties

```
<persistence-unit name="JPAezLoan" transaction-type="JTA">
 <provider>org.hibernate.ejb.HibernatePersistence</provider>
<jta-data-source>java:LoanAppDS</jta-data-source>
          <class>com.ez.model.entity.BankEntity</class>
```

```
        <class>com.ez.model.entity.CustomerEntity</class>
        <class>com.ez.model.entity.DepartmentEntity</class>
        <class>com.ez.model.entity.EmployeeEntity</class>
        <properties>
            <property name="hibernate.dialect"
value="org.hibernate.dialect.MySQLDialect" />
            <property name="hibernate.hbm2ddl.auto" value="update" />
            <property name="hibernate.show_sql" value="true" />
            <property name="hibernate.format_sql" value="true" />
            <property name="hibernate.use_sql_comments" value="true" />
            <property name="hibernate.transaction.manager_lookup_class"

    value="org.hibernate.transaction.JBossTransactionManagerLookup" />
        <!--    <property name="hibernate.transaction.factory_class"
                value="org.hibernate.transaction.JTATransactionFactory" /> -->
            <property name="hibernate.current_session_context_class"
                value="jta" />
            <property name="jboss.entity.manager.factory.jndi.name"
                value="java:/JPAezLoan" />
        </properties>
    </persistence-unit>

    <persistence-unit name="JPAHistory" transaction-type="JTA">
     <provider>org.hibernate.ejb.HibernatePersistence</provider>
        <jta-data-source>java:JPAHistoryDS</jta-data-source>
      <class>com.ez.model.entity.BankEntity</class>
        <class>com.ez.model.entity.CustomerEntity</class>
        <class>com.ez.model.entity.DepartmentEntity</class>
        <class>com.ez.model.entity.EmployeeEntity</class>
        <properties>
            <property name="hibernate.dialect"
value="org.hibernate.dialect.MySQLDialect" />
            <property name="hibernate.hbm2ddl.auto" value="update" />
            <property name="hibernate.show_sql" value="true" />
            <property name="hibernate.format_sql" value="true" />
            <property name="hibernate.use_sql_comments" value="true" />
            <property name="hibernate.transaction.manager_lookup_class"

    value="org.hibernate.transaction.JBossTransactionManagerLookup" />

    <!–
        Below entry was showing error so I have commented it

        <property name="hibernate.transaction.factory_class"
                value="org.hibernate.transaction.JTATransactionFactory" /> -->
            <property name="hibernate.current_session_context_class"
                value="jta" />
            <property name="jboss.entity.manager.factory.jndi.name"
                value="java:/JPAHistory" />
        </properties>
        </persistence-unit>
```

Step-3



```
WEB-INF
    context
        ezloan-spring-root-application-context.xml 83  4/16/13 2:4
        jms-context.xml 76  4/9/13 1:30 PM  nagendra.yadav.niit@
        jpa-hibernate.xml 77  4/9/13 4:14 PM  nagendra.yadav.niit
```

Creating entityFactoryManager for different persistsence unit

```
<!-- This is similar to hibernate session factory -->
 <bean id="entityManagerFactory "
        class="org.springframework.orm.jpa.LocalEntityManagerFactoryBean">
                <property name="persistenceUnitName" value="JPAezLoan"/>
</bean>

 <bean id="historyEntityManager "
                class="org.springframework.orm.jpa.LocalEntityManagerFactoryBean">
                <property name="persistenceUnitName" value="JPAHistory"/>
</bean>
```

Note : Configuring spring JTA transaction manager

```
<tx:jta-transaction-manager  />

        <!-- Enabling This transaction manager with annotation -->
        <!-- This code snippet is for Spring Transaction with Annotation -->
<tx:annotation-driven proxy-target-class="true"
                order="100" transaction-manager="transactionManager" mode="proxy"/>
```

Step-4

```
/**
 *
 * @author nagendra.yadav
 *
 */
@Repository("CustomerDAOImpl")
@Transactional
public class CustomerDAOImpl implements CustomerDAO {

	//This is similar to session in hibernate
	@PersistenceContext(unitName="JPAezLoan")
	private EntityManager entityManager;

	@PersistenceContext(unitName="JPAHistory")
	private EntityManager historyEntityManager;

	@Override
	public String addCustomer(CustomerEntity customerEntity) {

	//checking weather spring transaction is active or not
	System.out.println("TransactionSynchronizationManager.isActualTransactionActive()   =
	"+TransactionSynchronizationManager.isActualTransactionActive());
		//persisting data into one database
		entityManager.persist(customerEntity);

		DepartmentEntity department=new DepartmentEntity();
		department.setContact("111");
		department.setCode("122");
		department.setName("loan");
		//persisting data into another database
		historyEntityManager.persist(department);
		return "CustomerSaved";
	}
```