

Functional Requirements Specification (FRS)

**Author: Rakesh Ranjan**

**Filename:** WildFly Server Migration

**Version:** 1.0

**Revision Date:**

**Company:** ©ITS-GBS Egypt

**Confidentiality:** Internal Use Only

Contents

[1. Overview 2](#_Toc5265580)

[1.1. Installation: 2](#_Toc5265581)

[1.2. Configuration: 2](#_Toc5265582)

1.2.1 Standalone server configuration : ……………………………………………………………………………………………6

1.2.2 Domain Server configuration : …………………………………………………………………………………………………6

[1.3. Starting Wildfly: 3](#_Toc5265583)

1.3.1 Alternative Starting WildFly Configuration: …………………………………………………………………………6

[1.4. Test Installation: 4](#_Toc5265584)

[2. Authentication and Management: 5](#_Toc5265585)**Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.**

[2.1 Mangement : 9](#_Toc5265588)

[2.2 Authentication 19](#_Toc5265589)

3 . Administration Console : ………………………………………………………………………………………………………………………20

4 . Command Line Interface : ……………………………………………………………………………………………………………………20

5 . Adding Driver and DataSoure : ……………………………………………………………………………………………………………20

6 . Creating DataSoure From WildFly Console : …………………………………………………………………………………………20

[7.Source Code Changes: 5](#_Toc5265585)

**Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.**

**Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.**

# Overview

## Installation of WildFly 16:



WildFly 16 distributions can be obtained from:

[wildfly.org/downloads](http://www.wildfly.org/downloads/)

WildFly 16 provides a single distribution available in zip or tar file formats.

* **wildfly-16.0.0.Final.zip**
* **wildfly-16.0.0.Final.tar.gz**

Simply extract your chosen download to the directory of your choice. You can install WildFly 16 on any operating system that supports the zip or tar formats

**Requirements :**

* Java SE 8 or later.

## Configuration:

* + 1. **Standalone Server Configurations :**
* standalone.xml (*default*)
* Java Enterprise Edition 8 web profile certified configuration with the required technologies plus those noted in the table above.
* standalone-ha.xml
* Java Enterprise Edition 8 web profile certified configuration with high availability
* standalone-full.xml
* Java Enterprise Edition 8 full profile certified configuration including all the required EE 8 technologies
* standalone-full-ha.xml
* Java Enterprise Edition 8 full profile certified configuration with high availability

**1.2.2.** Domain Server Configurations :

* domain.xml
* Java Enterprise Edition 8 full and web profiles available with or without high availability

Important to note is that the ***domain*** and ***standalone*** modes determine how the servers are managed not what capabilities they provide



## Starting WildFly:

To start WildFly 16 using the default web profile configuration in " standalone" mode, change directory to $JBOSS\_HOME/bin.

./standalone.sh

To start the default web profile configuration using domain management capabilities,

./domain.sh

**1.3.1 . Starting WildFly Alternative Configuration**

If you choose to start your server with one of the other provided configurations, they can be accessed by passing the --server-config argument with the server-config file to be used.

To use the full profile with clustering capabilities, use the following syntax from $JBOSS\_HOME/bin:

./standalone.sh --server-config=standalone-full-ha.xml

Similarly, to start an alternate configuration in *domain* mode:

./domain.sh --domain-config=my-domain-configuration.xml

Alternatively, you can create your own selecting the additional subsystems you want to add, remove, or modify.



## Test Installation:

After executing one of the above commands, you should see output like what’s shown below.

=========================================================================

JBoss Bootstrap Environment

JBOSS\_HOME: /opt/wildfly-10.0.0.Final

JAVA: java

JAVA\_OPTS: -server -Xms64m -Xmx512m -XX:MetaspaceSize=96M -XX:MaxMetaspaceSize=256m -Djava.net.preferIPv4Stack=true -Djboss.modules.system.pkgs=com.yourkit,org.jboss.byteman -Djava.awt.headless=true

=========================================================================

11:46:11,161 INFO [org.jboss.modules] (main) JBoss Modules version 1.5.1.Final

11:46:11,331 INFO [org.jboss.msc] (main) JBoss MSC version 1.2.6.Final

11:46:11,391 INFO [org.jboss.as] (MSC service thread 1-6) WFLYSRV0049: WildFly Full 10.0.0.Final (WildFly Core 2.0.10.Final) starting

<snip>

11:46:14,300 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0025: WildFly Full 10.0.0.Final (WildFly Core 2.0.10.Final) started in 1909ms - Started 267 of 553 services (371 services are lazy, passive or on-demand)

As with previous WildFly releases, you can point your browser to ***[http://localhost:8080](http://localhost:8080/)*** (if using the default configured http port) which brings you to the Welcome Screen:

From here you can access links to the WildFly community documentation set, stay up-to-date on the latest project information, have a discussion in the user forum and access the enhanced web-based Administration Console. Or, if you uncover a defect while using WildFly, report an issue to inform us (attached patches will be reviewed). This landing page is recommended for convenient access to information about WildFly 16 but can easily be replaced with your own if desired.



# Authentication and Management:

**2.1 Management :**

WildFly 16 offers two administrative mechanisms for managing your running instance:

* command-line interface
* web-based Administration Console

**2.2 Authentication :**

By default, WildFly 16 is now distributed with security enabled for the management interfaces, this means that before you connect using the administration console or remotely using the CLI you will need to add a new user, this can be achieved simply by using the *add-user.sh* script in the bin folder.

After starting the script, you will be guided through the process to add a new user: -

./add-user.sh

What type of user do you wish to add?

a) Management User (mgmt-users.properties)

b) Application User (application-users.properties)

(a):

In this case a new user is being added for the purpose of managing the servers so select option a.

You will then be prompted to enter the details of the new user being added: -

Enter the details of the new user to add.

Realm (ManagementRealm) :

Username :

Password :

Re-enter Password :

It is important to leave the name of the realm as 'ManagementRealm' as this needs to match the name used in the server’s configuration, for the remaining fields enter the new username, password and password confirmation.

Provided there are no errors in the values entered you will then be asked to confirm that you want to add the user, the user will be written to the properties files used for authentication and a confirmation message will be displayed.

The modified time of the properties files are inspected at the time of authentication and the files reloaded if they have changed, for this reason you do not need to re-start the server after adding a new user.

# Administration Console:

To access the web-based Administration Console, simply follow the link from the Welcome Screen. To directly access the Management Console, point your browser at:

***<http://localhost:9990/console>***

|  |  |
| --- | --- |
|  | port 9990 is the default port configured. |

<management-interfaces>

<native-interface security-realm="ManagementRealm">

<socket-binding native="management-native"/>

</native-interface>

<http-interface security-realm="ManagementRealm">

<socket-binding http="management-http"/>

</http-interface>

</management-interfaces>

If you modify the *management-http* socket binding in your running configuration: adjust the above command accordingly. If such modifications are made, then the link from the Welcome Screen will also be inaccessible.

If you have not yet added at least one management user an error page will be displayed asking you to add a new user, after a user has been added you can click on the 'Try Again' link at the bottom of the error page to try connecting to the administration console again.

# 4.Command-Line Interface

If you prefer to manage your server from the command line (or batching), the *jboss-cli.sh* script provides the same capabilities available via the web-based UI. This script is accessed from $JBOSS\_HOME/bin directory; e.g.,

$JBOSS\_HOME/bin/jboss-cli.sh --connect

Connected to standalone controller at localhost:9990

Notice if no host or port information provided, it will default to localhost:9990.

When running locally to the WildFly process the CLI will silently authenticate against the server by exchanging tokens on the file system, the purpose of this exchange is to verify that the client does have access to the local file system. If the CLI is connecting to a remote WildFly installation, then you will be prompted to enter the username and password of a user already added to the realm.

Once connected you can add, modify, remove resources and deploy or undeployed applications. For a complete list of commands and command syntax, type ***help*** once connected.

**5.Adding Driver and Data Source**

Extract the appropriate JDBC jar file from within the archive and place it in that folder then create a module.xml file with the following content

<module xmlns="urn:jboss:module:1.3" name="com.microsoft.sqlserver">

    <resources>

        <resource-root path="mssql-jdbc-6.4.0.jre[version].jar" />

    </resources>

    <dependencies>

        <module name="javax.api"/>

        <module name="javax.transaction.api"/>

    </dependencies>

</module>

Next, we need to tell Wildfly about the new JDBC driver. Open the configuration file you’re using (e.g. WILDFLY\_HOME/standalone/configuration/standalone.xml), and navigate to the “<drivers>” section. You should see a reference to the H2 JDBC driver here. Add the following block after the “<driver>” entry for H2:

<driver name="sqlserver" module="com.microsoft.sqlserver">

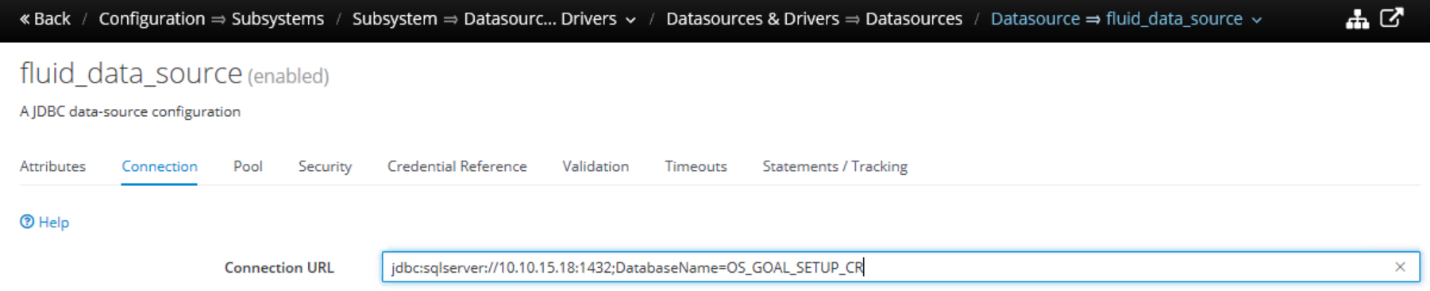
    <driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver-class>

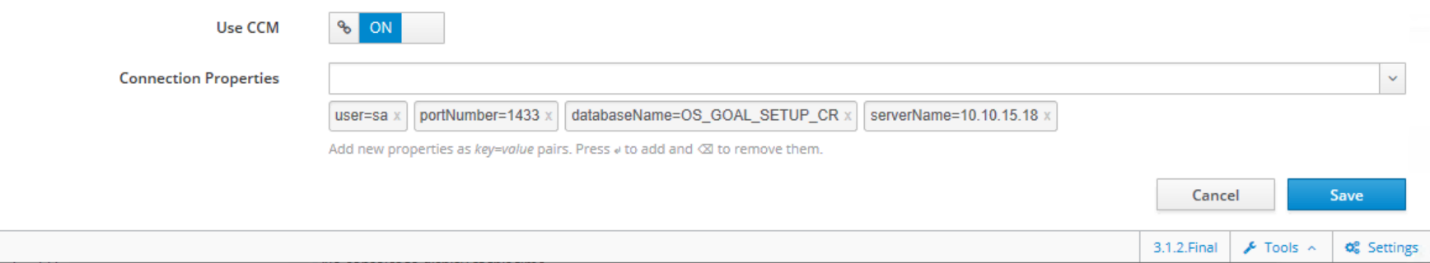
    <xa-datasource-class>com.microsoft.sqlserver.jdbc.SQLServerXADataSource</xa-datasource-class>

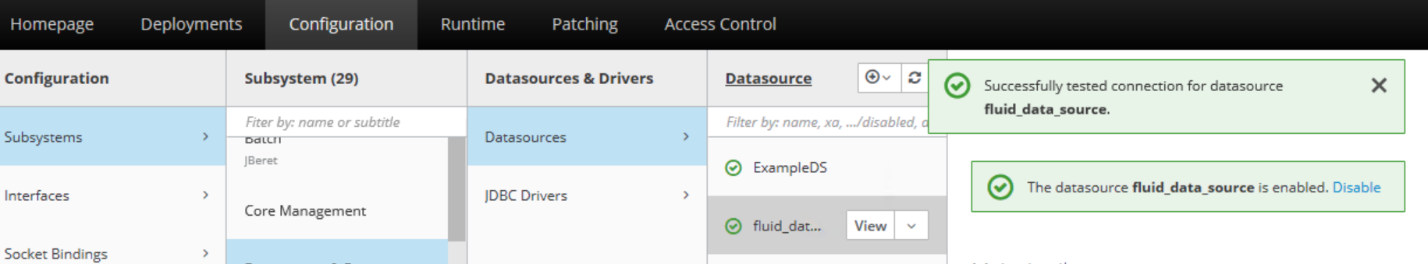
</driver>

**6.Create Data-Source from WildFly console:**

Once Driver is loaded, create data source connection as below:









**7.code change:**

Following are the changes that are performed and in progress to deploy the FluidCebApp.ear on WildFly server.

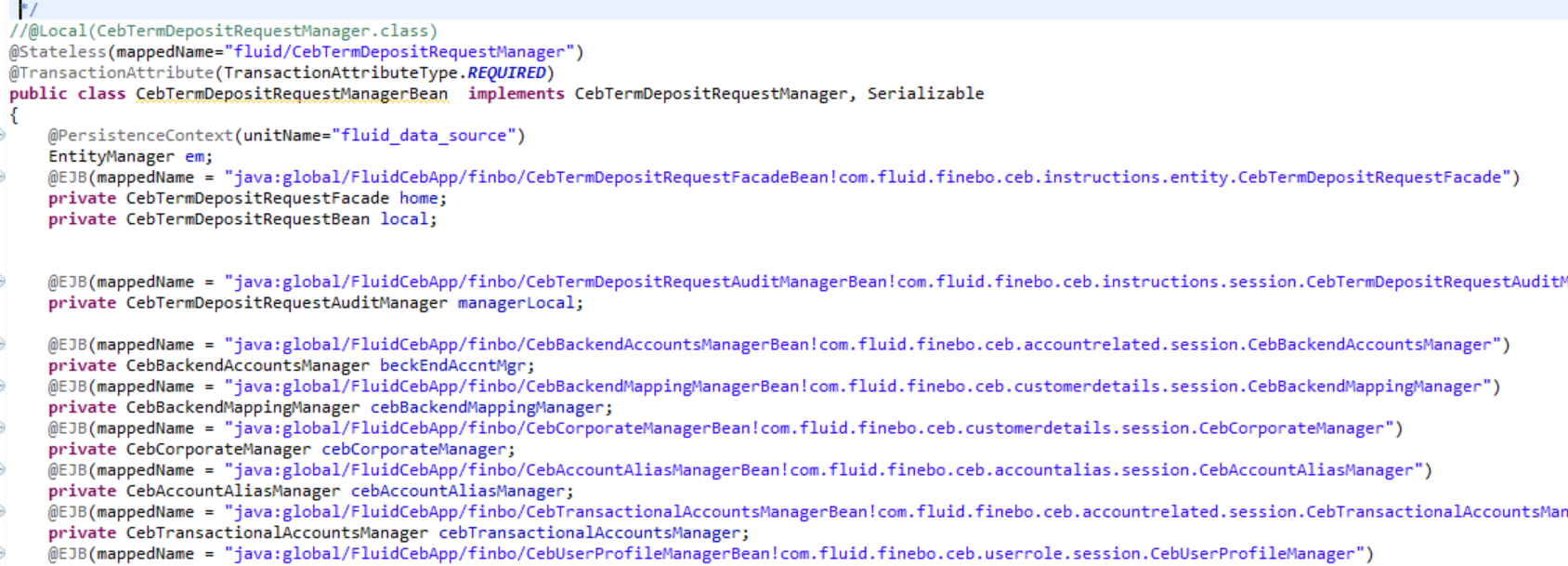
**Background of the changes:**

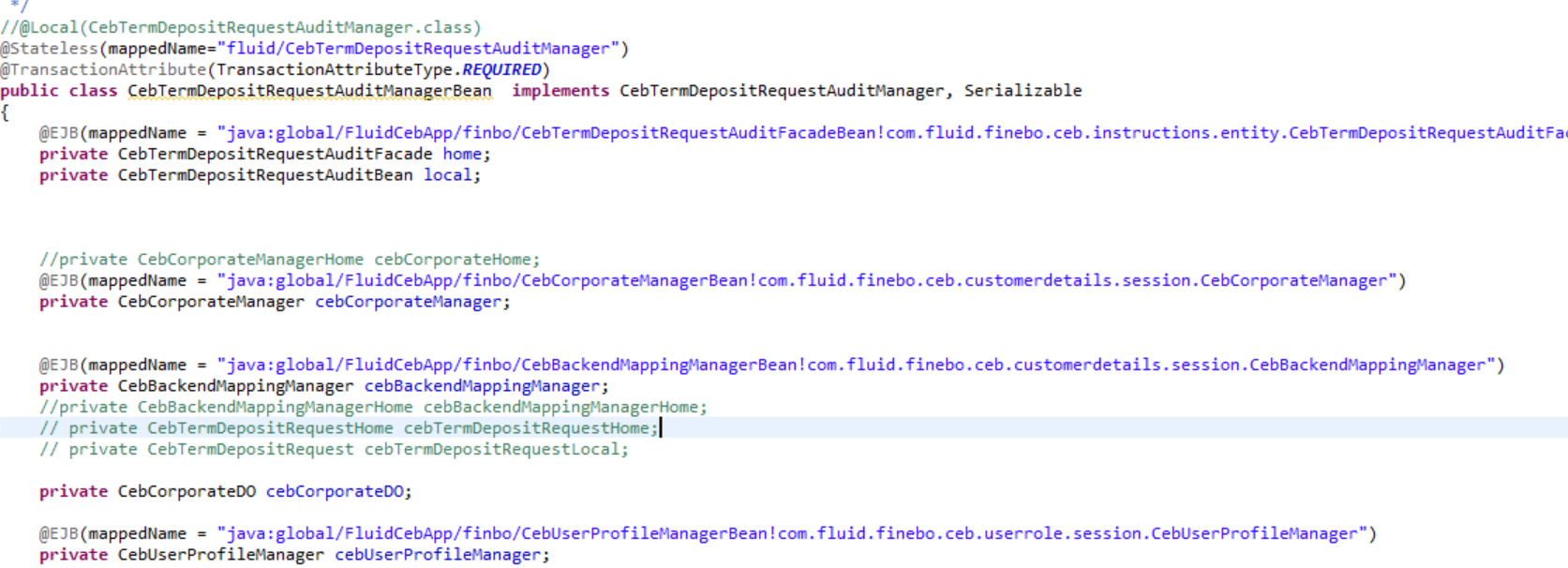
1. the existing code is implemented by referring the local view and remote more than once. That is, it is referring from @Local or @Remote as well as is referred by implementing the Remote or Local interface. Hence during the deployment on WildFly the server gives error stating bean is already registered as Remote or Local view.
2. After the above issue was resolved, WildFly gave another error for No EJB found with the interface. The EJB beans are getting registered and the log also shows the JNDI lookup URL for the same for each beans . But the bean referring to another bean was not being injected. Hence there was an error.

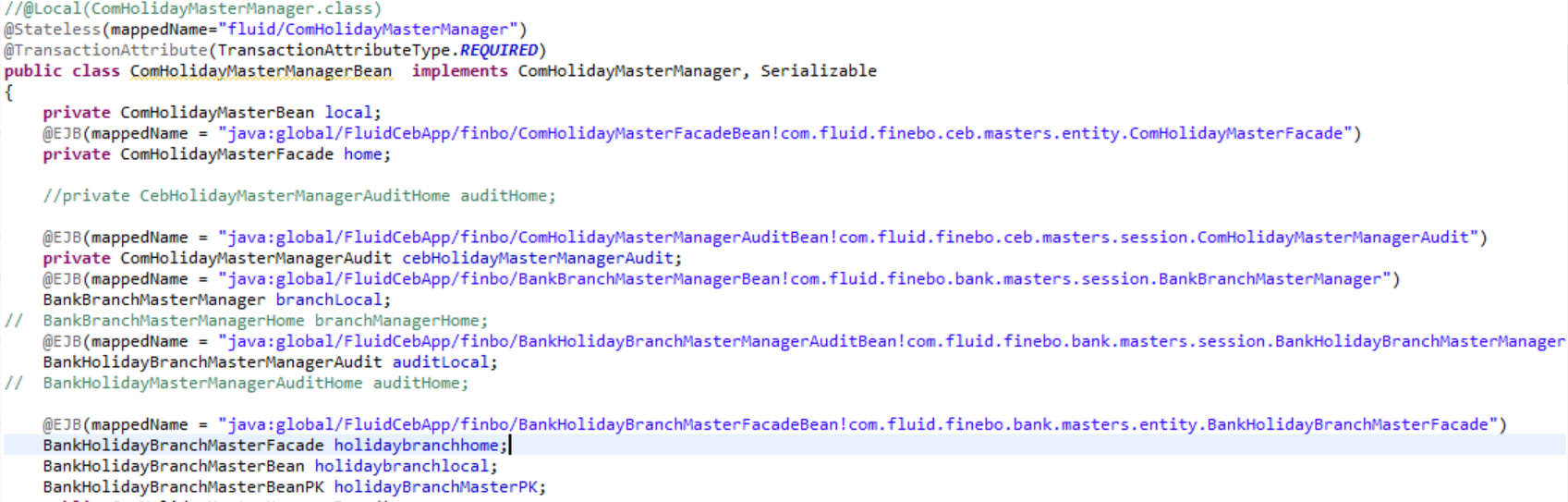
**Fix:**

1. We have currently fixed this issue by removing the interface reference from the @Local and @Remote annotation from around 500 files.
2. We are currently fixing this issue by adding the global JNDI url to the @EJB annotation. There are about 358 files and about 850 lines of code changes.

Lists of file changes :







**7.Configuration changes on Wildfly:**

|  |  |
| --- | --- |
| **Error** | **Resolution.** |
| 1.No class Def found for BouncyCastIe | Under standalone.xml, to use bouncy from built ear:  <subsystem xmlns="urn:jboss:domain:ee:4.0">  <global-modules>  <module name ="org.bouncycastle" slot="main" />  < / global-modules >  </subsystem> |
| 2.EHIXNET-ws - web.xml-- NoClassDef found for MOXy]sonProvider and eclipse.persistense | <context-param>  <param-name>resteasy.providers</param-name>  <param-value>org.eclipse.persistence.jaxb.rs.MOXy]sonProvider</param-value>  </context-param>  module.xml:  add > <module name="javax.ws.rs.api"/>  <resources>  <resource-root path="jipijapa-eclipselink-16.0.0.Final.jar"/>  <resource-root path="eclipselink-2.4.2.jar"›  <filter>  <exclude path="javax/—"/>  <filter>  </resource-root>  </resources>  under standalone.xml  <module name="org.eclipse.persistence" slot="main"/> |
| 3.ERROR [org.jboss.msc.service.fail] (MSC service thread 1-8) MSC000001: Failed to start service jboss.  deployment.subunit."FluidCebApp.ear"."VIPComponent.jar".metrics: org.jboss.msc.service.StartException in service  jboss.deployment.subunit."FluidCebApp.ear"."VIPComponent.jar".metrics: Failed to start service  at org.jboss.msc.service.ServiceControllerImpl$StartTask.execute(ServiceControllerImpl.java:1730)  at org.jboss.msc.service.ServiceControllerImpl$ControllerTask.run(ServiceControllerImpl.java:1558)  at org.jboss.threads.ContextClassLoaderSavingRunnable.run(ContextClassLoaderSavingRunnable.java:35)  at org.jboss.threads.EnhancedQueueExecutor.safeRun(EnhancedQueueExecutor.java:1982)  at org.jboss.threads.EnhancedQueueExecutor$ThreadBody.doRunTask(EnhancedQueueExecutor.java:1486)  at org.jboss.threads.EnhancedQueueExecutor$ThreadBody.run(EnhancedQueueExecutor.java:1363)  at java.lang.Thread.run(Unknown Source)  Caused by: java.lang.NullPointerException  at java.util.TreeMap.rotateLeft(Unknown Source)  at java.util.TreeMap.fixAfterInsertion(Unknown Source)  at java.util.TreeMap.put(Unknown Source)  at org.wildfly.extension.microprofile.metrics.PrometheusCollector.addMetricFamilySamples(PrometheusCollector.java:23)  at org.wildfly.extension.microprofile.metrics.PrometheusCollector.addMetricFamilySampleSupplier(PrometheusCollector.java:30)  at org.wildfly.extension.microprofile.metrics.MetricCollector.collectResourceMetrics0(MetricCollector.java:157)  at org.wildfly.extension.microprofile.metrics.MetricCollector.collectResourceMetrics0(MetricCollector.java:166)  at org.wildfly.extension.microprofile.metrics.MetricCollector.collectResourceMetrics0(MetricCollector.java:166)  at org.wildfly.extension.microprofile.metrics.MetricCollector.collectResourceMetrics(MetricCollector.java:92)  at org.wildfly.extension.microprofile.metrics.deployment.DeploymentMetricService.start(DeploymentMetricService.java:54)  at org.jboss.msc.service.ServiceControllerImpl$StartTask.startService(ServiceControllerImpl.java:1738)  at org.jboss.msc.service.ServiceControllerImpl$StartTask.execute(ServiceControllerImpl.java:1700)  ... 6 more | upgrade to 17.0.1 |
| 4.timeout >> CorrespondentsService extends AuthorizedService | @E1B(lookup="java:global/FluidCebApp/coarsebo/BankBranchManagerBean!com.fluid.coarsebo.common.bb.maintenance.bankbranchmaster.BankBranchManager")  private BankBranchManager  <ear-subdeployments-isolated>false</ear-subdeployments-isolated> |
| 5.javax.persistence.PersistenceException: [PersistenceUnit: pu] Unable to build Hibernate SessionFactory at org.hibernate.jpa.boot.internal.EntityManagerFactoryBuilderImpl.persistenceException(EntityManagerFactoryBuilderImpl.java:1016) at org.hibernate.jpa.boot.internal.EntityManagerFactoryBuilderImpl.build(EntityManagerFactoryBuilderImpl.java:942) at org.jboss.as.jpa.hibernate5.TwoPhaseBootstrapImpl.build(TwoPhaseBootstrapImpl.java:44) at org.jboss.as.jpa.service.PersistenceUnitServiceImpl$1$1.run(PersistenceUnitServiceImpl.java:170) 9 more Caused by: org.hibernate.MappingException: Could not determine type for: java.util.List, at table: EB\_StatementCycle, for columns: [org.hibernate.mapping.Column(transactionHistory)] at org.hibernate.mapping.SimpleValue.getType(SimpleValue.java:486) | added @ElementCollection(targetClass = EBTransactionResult.class ) |
| 6.2019-11-22 07:11:23,421 ERROR [org.jboss.as.controller.management-operation] (Controller Boot Thread) WFLYCTL0013: Operation ("deploy") failed - address "WFLYCTL0080: Failed services" => {"jboss.persistenceunit.\"FluidCebApp.ear/ETHIX-Net-Services-1.0-SNAPSHOT.jar#pu\"" => "javax.persistence.Persistencel Caused by: javax.persistence.PersistenceException: [PersistenceUnit: pu] Unable to build Hibernate SessionFactory Caused by: org.hibernate.MappingException: Could not determine type for: java.util.List, at table: EBAccountCreationAddress , for columns: [org.hibernate.mapping.Column(addressList)]"1, "WFLYCTL0412: Required services that are not installed:" => rjboss.ra.hornetq-ral, "WFLYCTL0180: Services with missing/unavailable dependencies" => rjboss.deployment.subunit.\"FluidCebApp.ear\".\"coarsebo.jar\".component.Interceptingt |  |
| 7. <class>com.its.ethix.net.shared.entities.EBAccountCreationClass</class> | Remove @ID |
| 8. <class>com.its.ethix.net.shared.entities.EBAccountMarginSearch </class> | EBAccountMarginSearch add @Id |
| 9. |  |



## 

## 































































## 