App Transformers

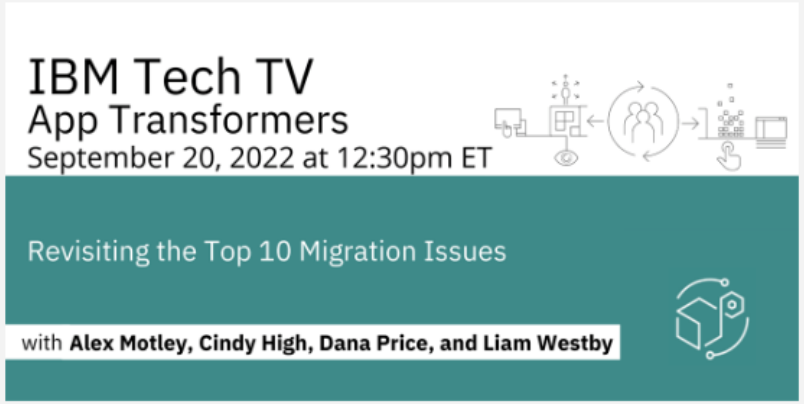
Revisiting the Top 10 Migration Issues Webinar

View Only

Group Home | Discussion 8.5K | Library 381 | Blogs 534 | Events 15 | Members 7.4K

When: Sep 20, 2022 from 12:30 PM to 01:30 PM (ET)

share:



Summary

We have covered the top Liberty migration issues in previous App Transformers sessions. In this session we will revisit the list to see what has changed - either because Liberty has included new capability or the migration tools have been improved. We also will dive in deeper and discuss how to resolve each issue in the list.

Key Speakers

Alex Motley - WebSphere Migration Tools Development,

Cindy High - Software Engineer, WebSphere Development

Dana Price - Architect for IBM Application Modernization, Master Inventor

Liam Westby - WebSphere Migration Development

#ibmtechtv

Register Now

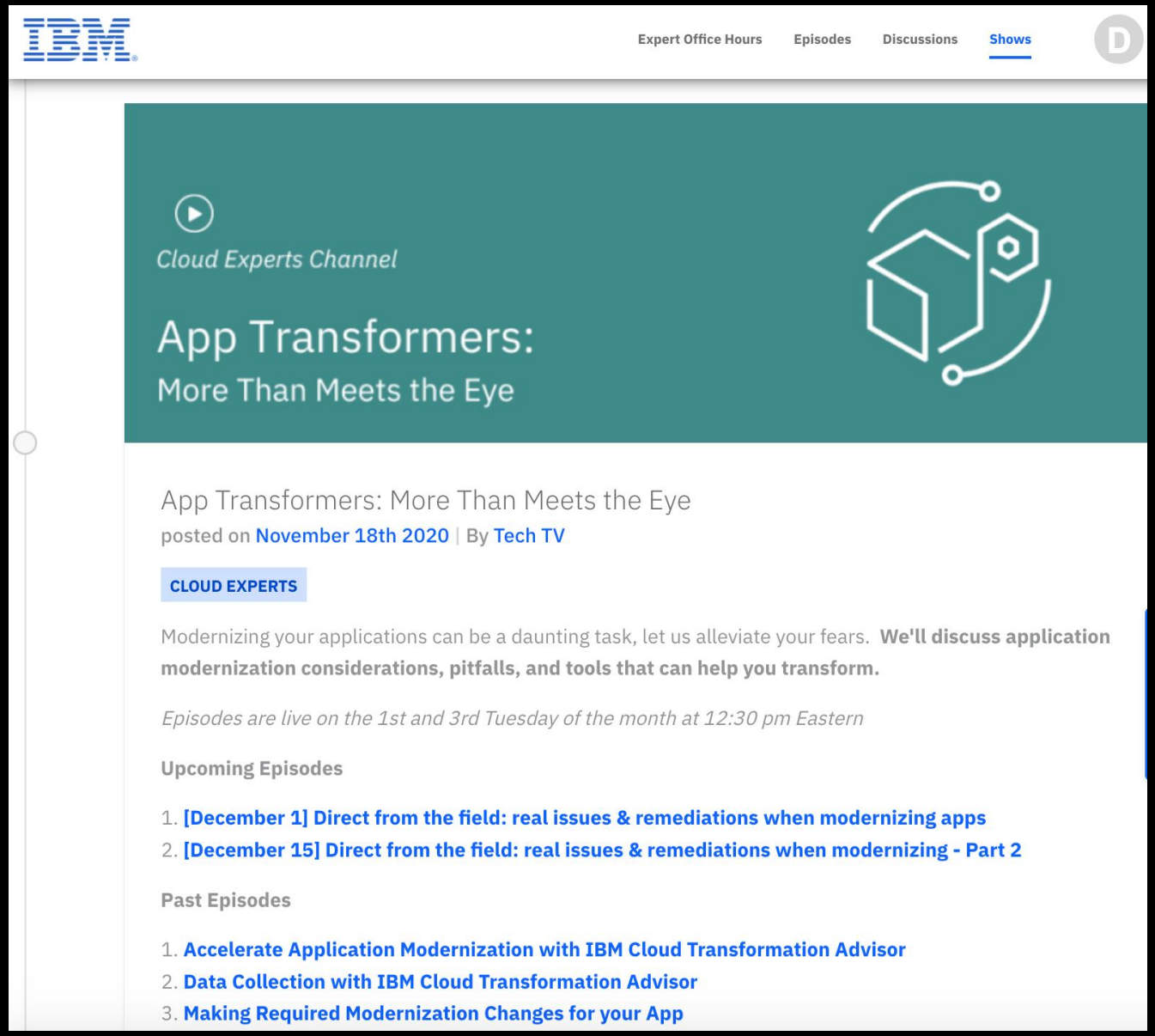
Download to Your Calendar

Contact

Michael Williams

mwilliams@higherlogic.com

App Transformers



Direct from the field:

Real issues & remediations when modernizing apps

David Mulley

David Hodges

David Vandepol

Our App Mod Top Ten – Dec 2020

Rule
The InitialContext lookup method can return primitive types
The use of java.sql.Driver and java.sql.DriverManager interfaces requires configuration
Java API for XML-based RPC (JAX-RPC) is unavailable
org.apache third-party APIs are unavailable on Liberty
The WebSphere web services APIs and SPIs are unavailable
Review differences in WebSphere MBeans
Behavior difference for web service host name validation
The getRealPath method returns null from non-expanded WAR file
Ported locally transacted JMS sessions do not work in Liberty
The WebSphere Common Exception APIs are unavailable

- **JAX-RPC –** Best approach is to replace with JAX-WS Web Services.
- **Remote EJB Lookups –** have to make a fairly simple code change (deploy the app, start it, get the new URL, go back to the code and put it in)
- **WebSphere runtime APIs** (com.ibm.websphere.runtime.*) calls – most have open source alternatives
- **3rd party APIs –** can be included in app or converted
- **Start up beans –** these APIs can be replaced by JEE (or Spring) annotations
- **Asynch Beans API, CommonJ Time and Work Manager APIs** - can be remediated to open source concurrency APIs
  https://www.ibm.com/support/knowledgecenter/en/SSEQTP_liberty/com.ibm.websphere.wlp.doc/ae/rwlp_mig_workmgr.html

Our App Mod Top Ten – Dec 2020

Rule	
The InitialContext lookup method can return primitive types	**Fixed with config**
The use of java.sql.Driver and java.sql.DriverManager interfaces requires configuration	**Fixed with improved scanning**
Java API for XML-based RPC (JAX-RPC) is unavailable	
org.apache third-party APIs are unavailable on Liberty	**Many of these were false positives. Test.**
The WebSphere web services APIs and SPIs are unavailable	**JAX-RPC related. Stub class flagged separately now.**
Review differences in WebSphere MBeans	**Fixed with improved scanning**
Behavior difference for web service host name validation	**Liberty more secure by default. Changed with config**
The getRealPath method returns null from non-expanded WAR file	**Fixed with config**
Ported locally transacted JMS sessions do not work in Liberty	
The WebSphere Common Exception APIs are unavailable	**Fixed with heritageAPIs**

Our App Mod Top Ten – Dec 2020

Rule	
The InitialContext lookup method can return primitive types	**Fixed with config**
The use of java.sql.Driver and java.sql.DriverManager interfaces requires configuration	**Fixed with improved scanning**
Java API for XML-based RPC (JAX-RPC) is unavailable	
org.apache third-party APIs are unavailable on Liberty	**Many of these were false positives. Test.**
The WebSphere web services APIs and SPIs are unavailable	**JAX-RPC related. Stub class flagged separately now.**
Review differences in WebSphere MBeans	**Fixed with improved scanning**
Behavior difference for web service host name validation	**Liberty more secure by default. Changed with config**
The getRealPath method returns null from non-expanded WAR file	**Fixed with config**
Ported locally transacted JMS sessions do not work in Liberty	
The WebSphere Common Exception APIs are unavailable	**Fixed with heritageAPIs**

Our App Mod Top Ten – Sept 2022

Rule name
Ported locally transacted JMS sessions do not work in Liberty
Java API for XML-based RPC (JAX-RPC) is unavailable
The WebSphere web services APIs and SPIs are unavailable
Transaction propagation is not supported for Enterprise JavaBeans (EJB) remote interfaces
The WebSphere Common Exception APIs are unavailable
Do not use the HttpSession invalidate method for programmatic security logout in Servlet 3.0
Use the default InitialContext JNDI properties
Getting the server name on Liberty
The WebSphere Resource Adapter APIs are unavailable
The WebSphere Management APIs are unavailable

1. Ported locally transacted JMS sessions do not work in Liberty

- If your code is running in a global transaction, this is a false positive result

- If running outside of a global transaction and using the MQ JMS provider, you should:
  - Move the code to a global transaction
  - Rewrite the code to not rely on the commit/rollback methods

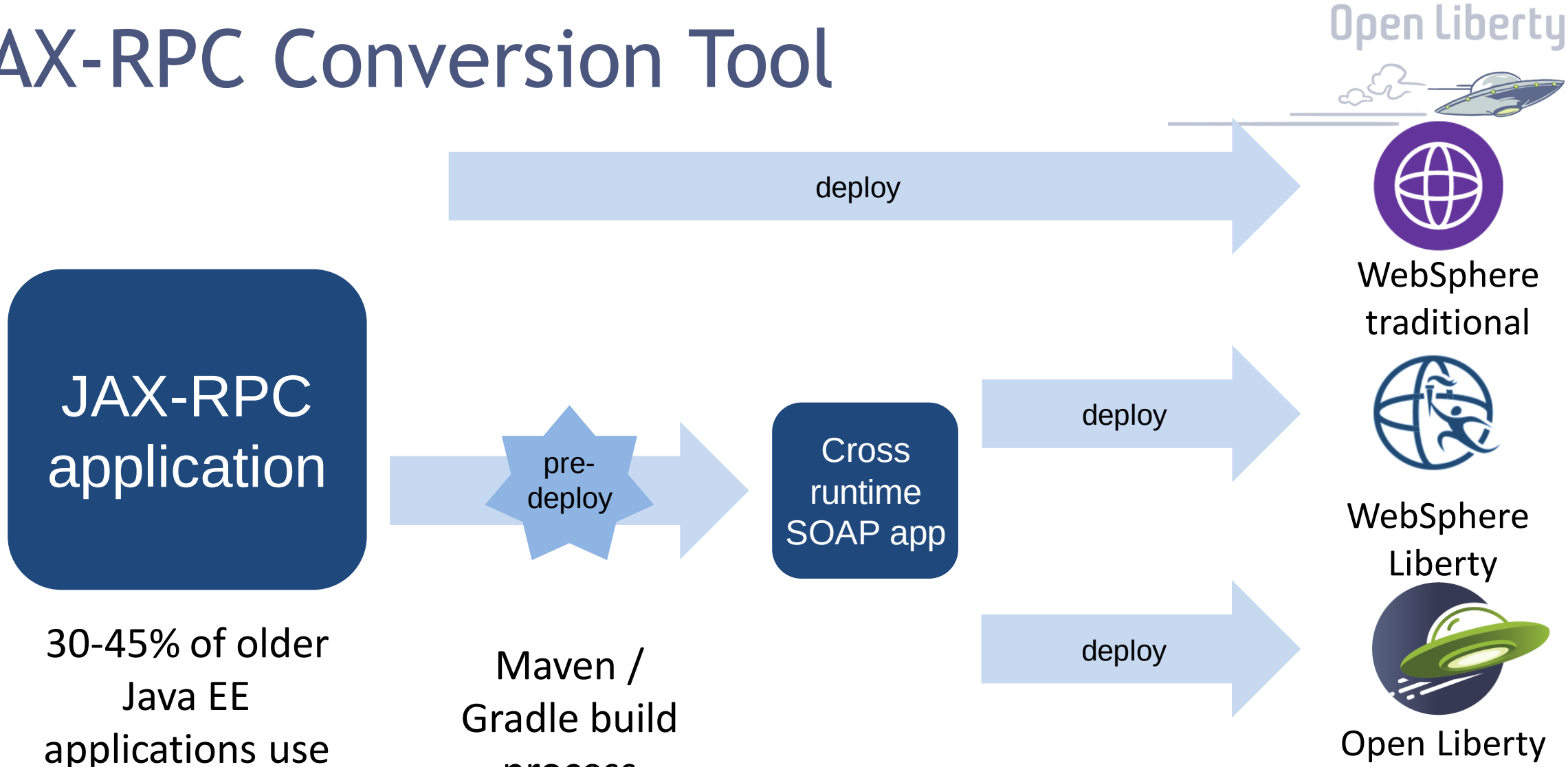
2. Java API for XML-based RPC (JAX-RPC) is unavailable

- Best approach is to replace JAX-RPC with JAX-WS Web Services
- Getting to JAX-WS does not always mean manual code changes

<u>4 Options for migration</u>

- Migrate JAX-RPC to JAX-WS using the JAX-RPC Conversion Tool
- Migrate JAX-RPC to JAX-WS manually
- Use Apache Axis JAX-RPC engine on Liberty
- Leave the application on traditional WebSphere

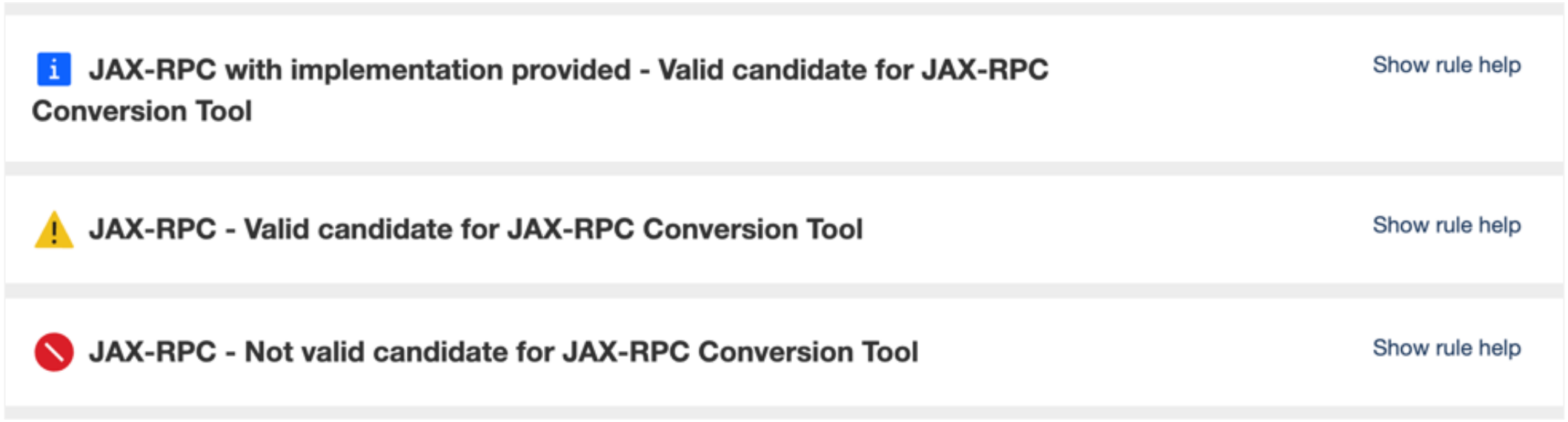
JAX-RPC Conversion Tool




```
<featureManager>
    <feature>jaxws-2.2</feature>
</featureManager>
```

2. Java API for XML-based RPC (JAX-RPC) is unavailable

- App modernization tools guide your choices with pre-validation built in



3. The WebSphere web services APIs and SPIs are unavailable

- For old data, usually due to generated WSDL2Java classes for JAX-RPC
- For applications converted to JAX-WS, these classes can be removed
- The JAX-RPC to JAX-WS Conversion Tool removes the generated classes

4. Transaction propagation is not supported for Enterprise JavaBeans (EJB) remote interfaces

- Are EJB beans called remotely between applications across JVMs?
- Are remote EJB beans called remotely across JVMs?

- If you really need remote interfaces with transaction propagation,
  - Wrap the EJB in a web service
  - Use Web Services Atomic Transactions to enable distributed global transactions

5. The WebSphere Common Exception APIs are unavailable

- Most common exceptions have been added to the Liberty heritage APIs feature (heritageAPIs-1.1)

Package com.ibm.websphere.ce.cm

Exception Summary

Exception	Description
ConnectionWaitTimeoutException	Used as a chained exception when unable to allocate a connection before the connection timeout is reached.
DuplicateKeyException	Indicates when a duplicate key exception is thrown from the database.
ObjectClosedException	A subclass of StaleConnectionException.
PortableSQLException	Abtract base class for Portability Layer generic exception hierarchy; the Portability Layer attempts to map database-specific exceptions into one of the subclasses of PortableSQLException.
StaleConnectionException	Indicates that the connection has somehow become damaged and should be discarded (e.g.
StaleStatementException	StaleStatementException is a special type of StaleConnectionException, used internally, which signals that all cached statements are bad and should be destroyed.

https://www.ibm.com/docs/en/was-liberty/nd?topic=SSAW57_liberty/com.ibm.websphere.javadoc.liberty.doc/api_ndz.html

6. Do not use the HttpSession invalidate method for programmatic security logout in Servlet 3.0

- Before Servlet 3.0, there was the session.invalidate() method to clear out the session data

- Starting with Servlet 3.0, the HttpServletRequest.logout() method invalidates the security context

USE BOTH!!

7. Use the default InitialContext JNDI properties

- Liberty does not use com.ibm.websphere.naming.WsnInitialContextFactory
- Use the empty InitialContext() constructor

```
Hashtable<String, String>ht = new Hashtable<String, String>();
ht.put("java.naming.factory.initial","com.ibm.websphere.naming.WsnInitialContextFactory");
ht.put("java.naming.provider.url", "corbaloc:iiop:localhost:2809");
ht.put(Context.SECURITY_PRINCIPAL, "security_principal");
ht.put(Context.SECURITY_CREDENTIALS, "security_cred");
try {
    Context context = new InitialContext(ht);

    context.close();
}
catch(javax.naming.NamingException ne){}
```

```
try {
    Context context = new InitialContext();

    context.close();
}
catch(javax.naming.NamingException ne){}
```

8. Getting the server name on Liberty

- Simple change to get the server name.
- There is a quick fix in the source scanner

/guiderestintrowar/src/main/java/io/openliberty/guides/rest/HelloServerResource.java

```
1  package io.openliberty.guides.rest;
2
3  import javax.ws.rs.GET;
4  import javax.ws.rs.Produces;
5
6  import javax.ws.rs.Path;
7
8
9  @Path("/helloServer")
10 public class HelloServerResource {
11
12         @GET
13         @Produces("text/plain")
14         public String getClichedMessage() {
15
16           String helloString = java.lang.System.getProperty("wlp.server.name");
17           return "Hello from " + helloString;
18         }
19
20 }
21
22
```

/guiderestintrowar/src/main/java/io/openliberty/guides/rest/HelloServerResource.java

```
1  package io.openliberty.guides.rest;
2
3  import javax.ws.rs.GET;
4  import javax.ws.rs.Produces;
5
6  import com.ibm.websphere.runtime.ServerName;
7
8
9  import javax.ws.rs.Path;
10
11
12 @Path("/helloServer")
13 public class HelloServerResource {
14
15         @GET
16         @Produces("text/plain")
17         public String getClichedMessage() {
18
19           String helloString = ServerName.getFullName();
20           return "Hello from " + helloString;
21         }
22
```

9. The WebSphere Resource Adapter APIs are unavailable

- heritageAPIs-1.1 adds
  - support for CommonJ Timer and Work Manager APIs
  - Extensions to Java Database Connectivity (JDBC)
    - Many classes added

```
<server>
  <featureManager>
    <feature>heritageAPIs-1.1</feature>
    <feature>jdbc-4.2</feature>
    <feature>jndi-1.0</feature>
    ...
  </featureManager>

  <dataSource id="MyDataSource" jndiName="jdbc/myDataSource" type="javax.sql.XADataSource">
    <jdbcDriver libraryRef="MySQL" javax.sql.XADataSource="com.mysql.cj.jdbc.MysqlXADataSource"/>
    <properties databaseName="exampledb" serverName="localhost" portNumber="3306"/>
    <containerAuthData user="dbuser1" password="dbpwd1"/>
    <heritageSettings helperClass="com.ibm.websphere.rsadapter.GenericDataStoreHelper" replaceExceptions="true"/>
    <identifyException sqlState="S1000" as="None"/>
    <identifyException errorCode="1022" as="com.ibm.websphere.ce.cm.DuplicateKeyException"/>
    <identifyException errorCode="1077" as="StaleConnection"/>
    <identifyException errorCode="1079" as="StaleConnection"/>
  </dataSource>
</server>
```

https://www.ibm.com/docs/en/was-liberty/base?topic=liberty-enabling-heritage-programming-models

10. The WebSphere Management APIs are unavailable

- The admin model is different in Liberty

- Look at what you are trying to accomplish to determine the fix.

  - com.ibm.websphere.ant.tasks
  - com.ibm.websphere.hamanager.jmx
  - com.ibm.websphere.interrupt
  - com.ibm.websphere.management
  - com.ibm.websphere.naming
  - com.ibm.websphere.product
  - com.ibm.wsspi.management.metadata

Tips from the field

- Don’t have every developer migrate their own code independently
- Have some experts who can deal with the commonality (instead of everyone doing it themselves)
- There is a learning curve and issues that come up many times.
- If different people have to solve the same issue it will take much longer.
- Solve common code issues first and reuse across applications