

Change JAXWS URL

Example [change-jaxws-url](https://github.com/apache/tomee/tree/master/examples/change-jaxws-url) can be browsed at <https://github.com/apache/tomee/tree/master/examples/change-jaxws-url>

**Help us document this example! Click the blue pencil icon in the upper right to edit this page.**

To change a webservice deployment URI one solution is to use `openejb-jar.xml`.

In this sample we have a webservice though the class `Rot13`:

```
package org.superbiz.jaxws;

import javax.ejb.Lock;
import javax.ejb.LockType;
import javax.ejb.Singleton;
import javax.jws.WebService;

@Lock(LockType.READ)
@Singleton
@WebService
public class Rot13 {
    public String rot13(final String in) {
        final StringBuilder builder = new StringBuilder(in.length());
        for (int b : in.toCharArray()) {
            int cap = b & 32;
            b &= ~cap;
            if (Character.isUpperCase(b)) {
                b = (b - 'A' + 13) % 26 + 'A';
            } else {
                b = cap;
            }
            b |= cap;
            builder.append((char) b);
        }
        return builder.toString();
    }
}
```

We decide to deploy to `/tool/rot13` url.

To do so we first define it in `openejb-jar.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<openejb-jar xmlns="http://www.openejb.org/xml/ns/openejb-jar-2.1">
  <enterprise-beans>
    <session>
      <ejb-name>Rot13</ejb-name>
      <web-service-address>/tool/rot13</web-service-address>
    </session>
  </enterprise-beans>
</openejb-jar>
```

It is not enough since by default TomEE deploys the webservises in a subcontext called webservises. To skip it simply set the system property tomee.jaxws.subcontext to / (done in arquillian.xml for our test).

Then now our Rot13 webservice is deployed as expected to /tool/rot13 and we check it with arquillian and tomee embedded:

```
package org.superbiz.jaxws;
```

```
import org.apache.ziplock.IO;
import org.jboss.arquillian.container.test.api.Deployment;
import org.jboss.arquillian.junit.Arquillian;
import org.jboss.arquillian.test.api.ArquillianResource;
import org.jboss.shrinkwrap.api.ArchivePaths;
import org.jboss.shrinkwrap.api.ShrinkWrap;
import org.jboss.shrinkwrap.api.asset.ClassLoaderAsset;
import org.jboss.shrinkwrap.api.spec.WebArchive;
import org.junit.AfterClass;
import org.junit.BeforeClass;
import org.junit.Test;
import org.junit.runner.RunWith;
```

```
import java.net.URL;
```

```
import static org.junit.Assert.assertThat;
import static org.junit.internal.matchers.StringContains.containsString;
```

```
@RunWith(Arquillian.class)
public class Rot13Test {
    @ArquillianResource
    private URL url;
```

```
@Deployment(testable = false)
public static WebArchive war() {
    return ShrinkWrap.create(WebArchive.class)
        .addClass(Rot13.class)
        .addAsWebInfResource(new ClassLoaderAsset("META-INF/openejb-jar.xml"),
ArchivePaths.create("openejb-jar.xml"));
}
```

```
@Test
public void checkWSDLIsDeployedWhereItIsConfigured() throws Exception {
    final String wsdl = IO.slurp(new URL(url.toExternalForm() +
"tool/rot13?wsdl"));
    assertThat(wsdl, containsString("Rot13"));
}
}
```