LEARN. NETWORK. EXPERIENCE OPEN SOURCE.

June 11-14, 2013
Boston, MA

Testing Persistence with Arquillian

Jim Rigsbee
Curriculum Manager, Red Hat
06.12.13

Testing Persistence with Arquillian

What is Arquillian?

- JUnit based testing framework (and TestNG)
- Runs in the container
- Manages archive building, deployment, and undeployment
- Container features available:
  - CDI
  - Transactions
  - Access to all resources

Testing Persistence with Arquillian

How does it work?

- ShrinkWrap builds the micro-deployment: JAR, WAR, EAR
- Starts an embedded or managed server, if configured (otherwise uses remote server)
- Deploys the micro-deployment
- Executes the tests inside the container
- Captures results and returns them to the test runner for reporting

Testing Persistence with Arquillian

Step 1. Maven POM – Dependency Management

```
<dependency>
    <groupId>org.jboss.bom</groupId>
    <artifactId>jboss-javaee-6.0-with-tools</artifactId>
    <version>1.0.0.M11-redhat-1</version>
    <type>pom</type>
    <scope>import</scope>
</dependency>
```

Testing Persistence with Arquillian

Step 2. Maven POM – Dependencies

```
<dependency>
   <groupId>junit</groupId>
   <artifactId>junit</artifactId
   <scope>test</scope>
</dependency>
<dependency>
   <groupId>org.jboss.arquillian.junit</groupId>
   <artifactId>arquillian-junit-container</artifactId>
   <scope>test</scope>
</dependency>
```

Testing Persistence with Arquillian

Step 3. Substitute Test Runner

```
@RunWith(Arquillian.class)
public class MyTest {
...
```

Testing Persistence with Arquillian

Step 4. ShrinkWrap Deployment Method

```
@Deployment
public static JavaArchive createDeployment() {
 JavaArchive jar = ShrinkWrap
   .create(JavaArchive.class, "test.jar")
   .addPackages(true, "accounting")
   .addAsManifestResource(
     new File("src/main/webapp/WEB-INF", "beans.xml"))
   .addAsManifestResource(
     new File("src/main/resources/META-INF",
       "persistence.xml"));
 System.out.println(jar.toString(true)); // for debugging
return jar;
}
```

Testing Persistence with Arquillian

Step 5. At least one @Test method

```
@Test
public void testSomething {
...
}
```

Testing Persistence with Arquillian

Step 6. Enjoy Container Resources

```
@Inject
private UserTransaction trans;
@PersistenceContext
EntityManager em;
```