



# **EXTENSIONS, WHICH YOU HAVE TO TAKE WITH YOU TO A DESERTED ISLAND**

Michał Matłoka / @mmatloka

# ABOUT ME

**POZnan\***

\*Miasto know-how



**ShrinkWrap**



**SOFTWAREMILL**

JBoss Community  
**RECOGNITION AWARDS**



JBoss Community Leader

**Michal Matloka** SHIRINKWRAP  
ISSUES / JIRA

# EXAMPLE SITUATION



Kajtek

Company with traditions

- JPA
- JSF
- No tests

# **JBoss Arquillian**

- Manages the lifecycle of the container (or containers)
- Deploys archive (or archives) to the container (or containers)
- Executes the tests inside (or against) the container
- Captures the results and returns them to the test runner for reporting

# CONTAINER ADAPTERS

- JBoss AS/WildFly
- GlassFish
- WebLogic
- WebSphere
- Jetty
- Tomcat
- OSGi
- Resin
- OpenShift
- Cloudbees
- ...

# CONTAINER TYPES

- Embedded
- Managed
- Remote

# JBOSS ARQUILLIAN

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

    @Deployment
    public static Archive<?> createDeployment() { return ... }

    @Inject
    private Something something;

    @Test
    public void should..() {
        ...
    }
}
```

# SHRINKWRAP

- Bundle the dependent classes and resources into an archive

```
WebArchive webArchive = ShrinkWrap.create(WebArchive.class)
    .addClass(User.class)
    .addAsManifestResource(EmptyAsset.INSTANCE, "beans.xml");
```

```
WebArchive webArchive = ShrinkWrap.create(WebArchive.class)
    .addClasses(User.class, ...)
    .addAsManifestResource(EmptyAsset.INSTANCE, "beans.xml");
```

```
WebArchive webArchive = ShrinkWrap.create(WebArchive.class)
    .addPackage(User.class.getPackage())
    .addAsManifestResource(EmptyAsset.INSTANCE, "beans.xml");
```



# SHRINKWRAP RESOLVER

```
File[] files = Maven.resolver()  
    .resolve("G:A:V")  
    .withTransitivity()  
    .asFile();
```

```
File[] files = Maven.resolver()  
    .loadPomFromFile("/path/to/pom.xml")  
    .resolve("G:A")  
    .withTransitivity()  
    .asFile();
```

```
File[] files = Maven.resolver()  
    .loadPomFromFile("/path/to/pom.xml")  
    .importRuntimeDependencies()  
    .resolve().withTransitivity()  
    .asFile();
```

```
WebArchive webArchive = ShrinkWrap.create(MavenImporter.class)  
    .loadPomFromFile("/path/to/pom.xml")  
    .importBuildOutput()  
    .as(WebArchive.class);
```

```
WebArchive webArchive = ShrinkWrap  
    .create(EmbeddedGradleImporter.class)  
    .forProjectDirectory(dir)  
    .importBuildOutput()  
    .as(WebArchive.class);
```

# SHRINKWRAP DESCRIPTORS

*The goal is to provide most of the Java EE deployment descriptors. Each descriptor follows the same API style and allows to manipulate all elements as the specification describes.*

# SHRINKWRAP DESCRIPTORS - PERSISTENCE.XML

```
<persistence>
  <persistence-unit name="myapp">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>
    <jta-data-source>java:/DefaultDS</jta-data-source>
    <properties>
      <property name="hibernate.dialect" value="org.hibernate.dialect.
      <property name="hibernate.hbm2ddl.auto" value="create-drop">
    </property></property></properties>
  </persistence-unit>
</persistence>
```

# SHRINKWRAP DESCRIPTORS - CODE

```
final PersistenceDescriptor persistence = Descriptors
    .create(PersistenceDescriptor.class)
    .createPersistenceUnit()
    .name("myapp")
    .provider("org.hibernate.ejb.HibernatePersistence")
    .jtaDataSource("java:/DefaultDS")
    .getOrCreateProperties()
    .createProperty().name("hibernate.dialect")
    .value("org.hibernate.dialect.HSQLDialect").up()
    .createProperty().name("hibernate.hbm2ddl.auto")
    .value("create-drop").up()
    .up().up()
```

# ARQUILLIAN PERSISTENCE EXTENSION

- Wrapping each test in the separated transaction
- Database seeding using: DBUnit (XML, XLS, YAML, JSON), SQL scripts
- DB state comparing with dataset at the end of the test
- Eviction JPA second level cache between test method invocation

# ARQUILLIAN PERSISTENCE EXTENSION - EXAMPLE PART I

```
@Test
@UsingDataSet("datasets/users.yml")
@ShouldMatchDataSet("datasets/expected-users.yml")
public void shouldChangeUserPassword() throws Exception {
    // given
    final String expectedPassword = "lubiePlacki";
    final User user = em.find(User.class, 2L);

    // when
    user.setPassword(expectedPassword);
    em.merge(user);

    // then
    assertThat(user.getPassword()).isEqualTo(expectedPassword);
}
```

# ARQUILLIAN PERSISTENCE EXTENSION - EXAMPLE PART II

```
user:
- id: 1
  firstname: John
  lastname: Smith
  username: doovde
  password: password
- id: 2
  firstname: Clark
  lastname: Kent
  username: superman
  password: kryptonite
```

# ARQUILLIAN DRONE

- Integrates WebDriver with Arquillian
- Interaction with deployments and containers provided by Arquillian
- Configuration kept in arquillian.xml
- Compatible with WebDriver (Selenium 2), Selenium 1 and Selenium Grids



# ARQUILLIAN DRONE - EXAMPLE

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

    @Deployment(testable = false)
    public static WebArchive createDeployment() ...

    @ArquillianResource
    URL contextPath;

    @Drone
    WebDriver driver;

    @Test
    public void shouldNotLogin() {
        driver.findElement(By.id("loginForm:login")).click();
        ...
    }
}
```

# ARQUILLIAN GRAPHENE

- Extensions for WebDriver
- PageObjects and PageFragments
- AJAX support

# ARQUILLIAN GRAPHENE - EXAMPLE PART I

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

    @Drone
    WebDriver browser;

    @Page
    HomePage homePage;

    @Test(expects = LoginFailedException.class)
    public void testLoginFailed() {}
        homePage.login("non-existent", "user");
    }}
}
```

# ARQUILLIAN GRAPHENE - EXAMPLE PART II

```
public class HomePage {  
  
    @FindBy(".login-form")  
    LoginDialog loginDialog;  
  
    @FindBy(".search")  
    AutocompleteComponent fulltextSearch;  
  
    public void login(String user, String password) throws LoginFailedExc  
        loginDialog.setUser(user);  
        loginDialog.setPassword(password);  
        loginDialog.login();  
    }  
    ...  
}
```

# ARQUILLIAN GRAPHENE - EXAMPLE PART III

```
public class LoginDialog {  
  
    @FindBy  
    private WebElement userName;  
  
    @FindBy(id = "login")  
    private WebElement loginButton;  
  
    ....  
}
```

# ARQUILLIAN GRAPHENE - AJAX

```
Graphene.waitAjax().until().element(webElementOrBy).is().  
    .is().not().present();  
    .enabled();  
    .selected();  
    .visible();
```

# ARQUILLIAN WARP

*Using Warp, you can initiate an HTTP request using a client-side testing tool such as WebDriver and, in the same request cycle, execute in-container server-side tests*

# ARQUILLIAN WARP

- Includes support for
  - JSF
  - JAX-RS (REST)
  - Spring MVC



# ARQUILLIAN WARP - EXAMPLE PART I

```
@RunWith(Arquillian.class)
@WarpTest
@RunAsClient
public class BasicTest {
    @Deployment
    ...

    @ArquillianResource
    URL contextPath;

    @Test
    public void should...() {
        Warp
            .initiate(Activity)
            .observe(Observer)
            .inspect(Inspection);
    }
}
```

# ARQUILLIAN WARP - EXAMPLE PART II

```
@Test
public void shouldMakeProperRequest() {
    Warp.initiate(new Activity() {
        @Override
        public void perform() {
            browser.navigate().to(contextPath + "/cart");
        }
    }).group().observe(request().uri().endsWith("/cart"))
    .inspect(inspection);
}
```

# ARQUILLIAN WARP REST - EXAMPLEPART III - REST

```
Inspection inspection = new Inspection() {  
    private static final long serialVersionUID = 1L;  
  
    @ArquillianResource  
    private RestContext restContext;  
  
    @AfterServlet  
    public void testGetStocks() {  
        assertThat(restContext.getHttpRequest().getMethod()).isEqualTo(HttpMet  
        assertThat(restContext.getHttpResponse().getStatusCode()).isEqualTo(Re  
        assertThat(restContext.getHttpResponse().getContentType()).isEqualTo(''  
  
        List list = (List) restContext.getHttpResponse().getEntity();  
        assertThat(list.size()).isEqualTo(1);  
    }  
}
```

# ARQUILLIAN DROIDIUM

- Includes container adapter
- Device and emulator management
- Can create AVD (Android Virtual Device)
- Arquillian Droidium Web - Selenium integration
- Arquillian Droidium Native - Selendroid integration
- Can take screenshots
- Early Alpha

# ARQUILLIAN DROIDIUM - EXAMPLE

```
@RunWith(Arquillian.class)
@RunWithClient
public class DroidiumWebTestCase {
    @Deployment
    @TargetsContainer("jbossas")
    public static Archive<?> getDeployment() {
        return ShrinkWrap
            .createFromZipFile(WebArchive.class, new File("shop.war"));
    }
    @Test
    @OperateOnDeployment("jbossas")
    public void test01(@Drone AndroidDriver driver,
        @ArquillianResource URL deploymentURL) {
        driver.get(deploymentURL.toString());
        assertTrue(driver.getPageSource().contains("Shop"));
    }
}
```

# ARQUILLIAN DROIDIUM - EXAMPLE

```
@RunWith(Arquillian.class)
@RunWithClient
public class DroidiumSelendroidTestCase {
    @Deployment
    @Instrumentable
    public static Archive<?> getDeployment() {
        return ShrinkWrap
            .createFromZipFile(JavaArchive.class, new File("shop.apk"));
    }
    @Test
    public void example(@ArquillianResource AndroidDevice android,
        @Drone WebDriver driver) {
        android.getActivityManagerProvider().getActivityManager()
            .startActivity("com.shop.HomeScreenActivity")
        WebElement button = driver
            .findElement(By.id("cartButton"));
        ...    }}
```

# ARQUILLIAN PERFORMANCE - EXAMPLE

```
@PerformanceTest(resultsThreshold=2)
// checks if resultsThreshold * newTime < oldTime
@RunWith(Arquillian.class)
public class RecommendationBuildingTest {
    @Deployment
    public static JavaArchive createDeployment() {
        return ...
    }

    @Inject RecommendationService recommendationService;

    @Test
    @Performance(time=20) // timeout in ms
    public void doHardWork() throws Exception {
        ...
    }
}
```

# OTHER EXTENSIONS

- Extension Byteman
- Google Guice Extension
- GWT Testing extension
- Extension Jacoco
- Extension JRebel
- screenRecorder
- Spring Framework Extension
- arquillian-ios-aggregator
- arquillian-extension-qunit
- Spock Arquillian Extension
- ShrinkWrap DSL

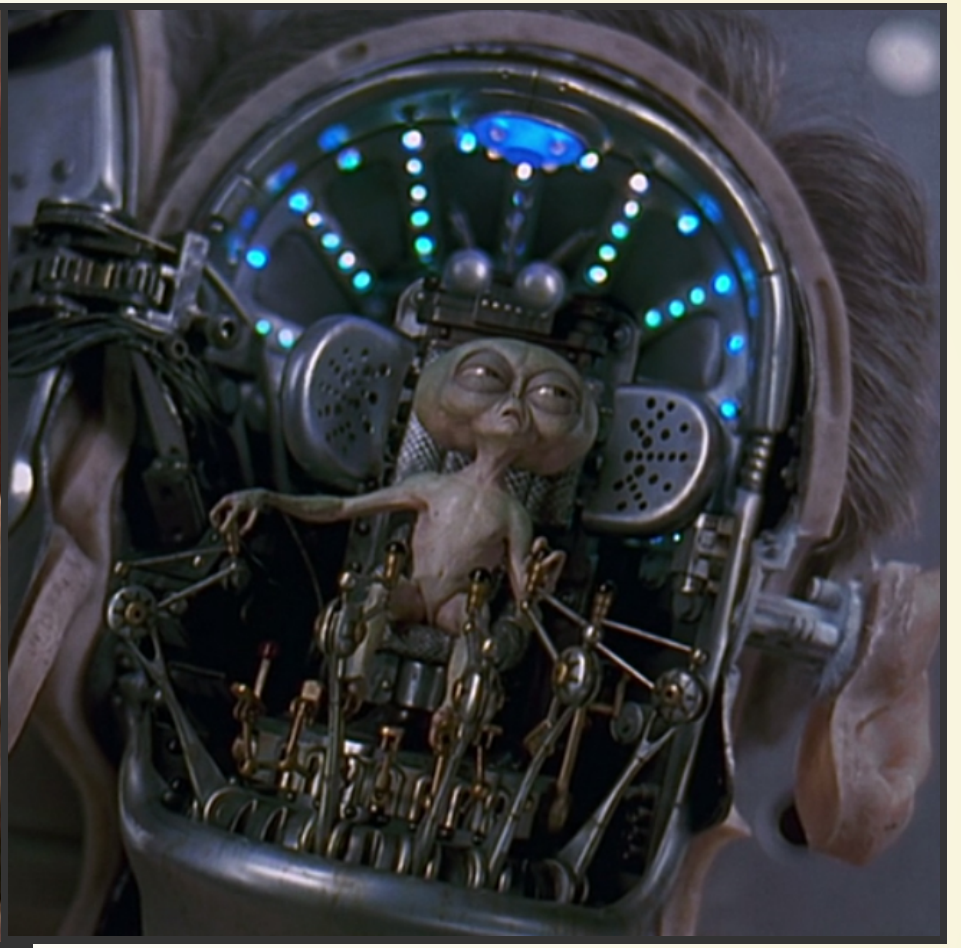


# TAKE AWAYS

- ShrinkWrap - archives
- ShrinkWrap Resolver - maven, gradle
- ShrinkWrap Descriptors - fluent API code generation from XSD
- Arquillian Persistence Extension - DB
- Arquillian Drone - WebDriver integration - UI
- Arquillian Graphene - extended WebDriver - better UI
- Arquillian Warp - client-side request, server-side test - REST, Spring MVC...
- Arquillian Droidium - Android Browser, Android UI
- Arquillian Performance - timeouts, execution time comparison

# MATERIALS & HELP

- <http://arquillian.org>
- <http://arquillian.org/guides/>
- <https://docs.jboss.org/author/display/ARQ/Reference+Guide>
- <https://github.com/arquillian/>
- <https://github.com/shrinkwrap/>
- <https://community.jboss.org/>
- <http://jira.jboss.org/>
- [irc.freenode.net](http://irc.freenode.net) - #arquillian, #jbosstesting



<http://meninblack.wikia.com/>

# THE END

## QUESTIONS?

mmatloka @ gmail.com

<https://github.com/mmatloka/presentations>