Errai

Errai Quickstart Guide

Preface

1. Document Conventions

2. Feedback

Errai Maven Archetype

In order to get you going quickly, we've provided a project archetype, that allows you to create a project skeleton similiar to the one we use for building the examples. It's based on the maven archetype plugin *http://maven.apache.org/plugins/maven-archetype-plugin/* and needs to be invoked from the command line:

```
mvn archetype:generate \
-DarchetypeGroupId=org.jboss.errai.archetypes \
-DarchetypeArtifactId=bus-quickstart \
-DarchetypeVersion=2.2.0.CR2 \
-DarchetypeRepository=https://repository.jboss.org/nexus/content/groups/
public/
```

When invoking the archetype build you will be asked to provide the maven groupId, artifactId and package name your GWT application should use:

```
Define value for groupId: : foo.bar
Define value for artifactId: : gwt-app
Define value for version: 1.0-SNAPSHOT: :
Define value for package: foo.bar: : foo.bar.ui
Confirm properties configuration:
groupId: foo.bar
artifactId: gwt-app
version: 1.0-SNAPSHOT
package: foo.bar.ui
Y: :
```

You will be left with a maven build structure, including references to the GWT SDK and the Errai dependencies necessary to build, test, package, and launch a simple application.

To launch the GWT development mode, change into the project directory (name corresponding to the provided artifactId) and type:

```
mvn gwt:run
```

The example application also comes with an integration test suite that exercises most of its client-side and server-side code. To run the test suite, type:

```
mvn test -Pintegration-test
```

To generate a set of HTML documents under target/site/jacoco/ detailing code coverage of the most recent test run, type:

```
mvn site
```

By default the archetype does package the web application for Development Mode execution. To deploy your application to JBoss AS 7, you need to execute a clean rebuild using the JBoss profile (e.g. `-Pjboss7`).

```
mvn -Pjboss7 clean install
cp target/gwt-app.war $JBOSS_HOME/standalone/deployments
```

Errai CDI Maven Archetype

The different runtime models explained *here* [https://docs.jboss.org/author/pages/viewpage.action?pageId=5931501] are all incorporated into this maven archetype using profiles. It enables execution in Development Mode and supports both packaging for deployment to a Servlet Engine and the JBoss Application Server.

To begin with we'll create a project layout using a maven build structure, which will provide us with a bare bones project, including all dependencies, which can later on be imported in your IDE of choice.

```
mvn archetype:generate \
-DarchetypeGroupId=org.jboss.errai.archetypes \
-DarchetypeArtifactId=cdi-quickstart \
-DarchetypeVersion=2.2.0.CR2 \
-DarchetypeRepository=https://repository.jboss.org/nexus/content/groups/
public/
```

Customize the build properties according to your needs.

```
Define value for property 'groupId': : foo.bar
Define value for property 'artifactId': : gwt-app
Define value for property 'version': 1.0-SNAPSHOT:
Define value for property 'package': foo.bar: com.foo.bar
Confirm properties configuration:
groupId: foo.bar
artifactId: gwt-app
version: 1.0-SNAPSHOT
package: com.foo.bar
Y:
```

The project will be created in a directory that corresponds to the provided artifactId.

In a few simple steps, you have created a build environment that can build, test, package, and launch a simple application. You can now launch GWT development mode, run integration tests with coverage reporting, and package your web application for deployment.

To launch the GWT development mode, change into the project directory (name corresponding to the provided artifactId) and type:

```
```

```
mvn gwt:run (launch hosted mode)
mvn gwt:debug (launch hosted with debug settings)
```

The example application comes with an integration test suite that exercises most of its client-side and server-side code. To run the test suite, type:

```
mvn test -Pintegration-test
```

To generate a set of HTML documents under target/site/jacoco/ detailing code coverage of the most recent test run, type:

```
mvn site
```

By default the archetype does package the web application for Development Mode execution. To deploy your application to JBoss AS 7, you need to execute a clean rebuild using the JBoss profile (e.g. `-Pjboss7`).

```
mvn -Pjboss7 clean install
cp target/gwt-app.war $JBOSS_HOME/standalone/deployments
```

Importing the project into eclipse

The setup instuctions for eclipse can be found in the Errai *WIKI* [http://community.jboss.org/wiki/WorkingwithGWTCDIandErrai] .

Errai JAX-RS Maven Archetype

You can use the Errai JAX-RS maven archetype to get started quickly. It will generate a fully functional CRUD application using JAX-RS.

```
mvn archetype:generate \
-DarchetypeGroupId=org.jboss.errai.archetypes \
-DarchetypeArtifactId=jaxrs-quickstart \
-DarchetypeVersion=2.2.0.CR2 \
-DarchetypeRepository=https://repository.jboss.org/nexus/content/groups/
public/
```

Customize the build properties according to your needs.

```
Define value for property 'groupId': : foo.bar
Define value for property 'artifactId': : rest-app
Define value for property 'version': 1.0-SNAPSHOT:
Define value for property 'package': foo.bar: com.foo.bar
Confirm properties configuration:
groupId: foo.bar
artifactId: rest-app
version: 1.0-SNAPSHOT
package: com.foo.bar
Y:
```

The project will be created in a directory that corresponds to the provided artifactId.

Now we have a fully working build environment set up in a few, simple steps. You can already use it to launch the GWT development mode or to package your web application for deployment.

```
mvn gwt:run (launch hosted mode)
mvn gwt:debug (launch hosted with debug settings)
```

The example application comes with an integration test suite that exercises most of its client-side and server-side code. To run the test suite, type:

```
mvn test -Pintegration-test
```

To generate a set of HTML documents under target/site/jacoco/ detailing code coverage of the most recent test run, type:

```
mvn site -Pintegration-test
```

To deploy your application to JBoss, you need to execute a clean rebuild using the JBoss profile (e.g. -Pjboss7)

```
mvn -Pjboss7 clean install
cp target/rest-app.war $JBOSS_HOME/standalone/deployments
```

Errai Kitchen Sink Maven Archetype

The Errai Kitchen Sink archetype generates a project that uses everything but the kitchen sink: annotation-driven CDI and Java Bean Validation on both the client and server, plus JPA and JAX-RS on the server side. It also demonstrates usage of Errai RPC and GWT UiBinder.

Note: Unlike the other Errai Archetypes, this Kitchen Sink archetype creates a project that only works with JBoss AS 7.1 or newer. Although this project could be made to work on other servers, the POM would be quite unwieldy if it included profiles for Tomcat and Jetty.

Here's how to create a project layout using a maven build structure, which will provide us with a bare bones project, including all dependencies, which you can import into your IDE of choice. You will need to have *Maven 3* [http://maven.apache.org/download.html] installed in order to execute this command successfully:

```
mvn archetype:generate \
-DarchetypeGroupId=org.jboss.errai.archetypes \
-DarchetypeArtifactId=jboss-errai-kitchensink-archetype \
-DarchetypeVersion=2.2.0.CR2 \
-DarchetypeRepository=https://repository.jboss.org/nexus/content/groups/
public/
```

Customize the build properties according to your needs.

```
Define value for property 'groupId': : foo.bar
Define value for property 'artifactId': : my-kitchen-sink
Define value for property 'version': 1.0-SNAPSHOT:
Define value for property 'package': foo.bar: com.foo.bar
Confirm properties configuration:
groupId: foo.bar
artifactId: my-kitchen-sink
version: 1.0-SNAPSHOT
package: com.foo.bar
Y:
```

The project will be created in a directory that corresponds to the provided artifactId.

In a few simple steps, you have created a build environment that can build, test, package, and launch a simple application. You can now launch GWT development mode, run integration tests with coverage reporting, and package your web application for deployment.

To get your app running in JBoss AS 7, ensure there is a local instance of JBoss AS 7.1 or newer running locally, then type:

```
mvn package
mvn jboss-as:deploy
```

You can then try out the app by pointing your web browser at *http://localhost:8080/artifactId/* (where *artifactId* is the artifactId you gave when generating the project.)

To launch the app in GWT development mode, keep JBoss AS running, change into the project directory (name corresponding to the provided artifactId) and type:

```
mvn gwt:run (launch hosted mode)
mvn gwt:debug (launch hosted with debug settings)
```

Please remember that in this project, Dev Mode's built-in Jetty server is disabled. Your app must be up and running on JBoss AS 7 as outlined in the previous step.

Importing the project into eclipse

The setup instructions for Eclipse can be found *here* [https://docs.jboss.org/author/pages/viewpage.action?pageId=54493497] .

Appendix A. Revision History

Revision History

Revision <>