

Testing with JUnit4

JUnit 4 for CDI Java-SE tests

If just plain unit testing is needed without JEE APIs it's possible to use the `AbstractCdiAwareTest`

Simple JUnit 4 test with dependency injection

```
@RunWith(JUnit4.class)
public class SimpleJUnitTest extends org.apache.myfaces.extensions.cdi.test.junit4.AbstractCdiAwareTest
{
    @Inject
    private Logger logger;

    @Test
    public void testLogger()
    {
        assertNotNull(this.logger);
        this.logger.info("I'm a JUnit test-case and I can use dependency injection!");
    }
}
```

JUnit 4 for CDI Java-EE tests

If it's required to test e.g. request scoped beans it's possible to use the `AbstractServletAwareTest`

Simple JUnit 4 test with dependency injection for the Servlet API

```
@RunWith(JUnit4.class)
public class SimpleJUnitTest extends org.apache.myfaces.extensions.cdi.test.junit4.AbstractServletAwareTest
{
    @Inject
    private RequestScopedBean requestScopedBean;

    @Test
    public void testLogger()
    {
        assertNotNull(this.requestScopedBean.getValue());
    }
}
```

If it's required to test e.g. grouped conversation scoped beans it's possible to use the `AbstractServletAwareTest`

Simple JUnit 4 test with dependency injection for the JSF API

```
@RunWith(JUnit4.class)
public class SimpleJUnitTest extends org.apache.myfaces.extensions.cdi.test.junit4.AbstractJsfAwareTest
{
    @Inject
    private ViewAccessScopedBean viewAccessScopedBean;

    @Test
    public void testLogger()
    {
        assertNotNull(this.viewAccessScopedBean.getValue());
    }
}
```

Needed modules and dependencies

Because the unit-test modules will never be needed in production builds, you have to add the CODI-test modules manually, even if you are using one of the all-in-one packages. You'll need at least the `myfaces-extcdi-junit-support-module` for general JUnit support. And depending on your needs, you'll also need the `myfaces-extcdi-owb-support-module` and `myfaces-extcdi-jsf-support-module`. Don't forget to also include the needed (test)modules of the CDI implementation. In case of OWB, you'll need `cditest` and `cditest-owb` from the `org.apache.openwebbeans.test` group. For convenience, here's a pom-snippet to pick the needed dependencies from:

Snippet from pom.xml with example CODI-test-dependencies.

```
<!-- JUnit -->
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <scope>test</scope>
  <version>${junit-version}</version>
</dependency>

<!-- CODI test modules -->
<dependency>
  <groupId>org.apache.myfaces.extensions.cdi.test</groupId>
  <artifactId>myfaces-extcdi-junit-support-module</artifactId>
  <version>${myfaces.cdi.version}</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.apache.myfaces.extensions.cdi.test</groupId>
  <artifactId>myfaces-extcdi-owb-support-module</artifactId>
  <version>${myfaces.cdi.version}</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.apache.myfaces.extensions.cdi.test</groupId>
  <artifactId>myfaces-extcdi-jsf-support-module</artifactId>
  <version>${myfaces.cdi.version}</version>
  <scope>test</scope>
</dependency>

<!-- the CODI myfaces-extcdi-jsf-support-module depends on myfaces-test -->
<dependency>
  <groupId>org.apache.myfaces.test</groupId>
  <artifactId>myfaces-test12</artifactId>
  <version>1.0.0</version>
  <scope>test</scope>
</dependency>

<!-- needed by myfaces test -->
<dependency>
  <groupId>commons-logging</groupId>
  <artifactId>commons-logging</artifactId>
  <version>1.1.1</version>
  <scope>test</scope>
</dependency>

<dependency>
  <groupId>org.apache.geronimo.specs</groupId>
  <artifactId>geronimo-el_1.0_spec</artifactId>
  <version>1.0.2</version>
  <scope>test</scope>
</dependency>

<!-- OWB test -->
<dependency>
  <groupId>org.apache.openwebbeans.test</groupId>
  <artifactId>cditest</artifactId>
  <version>${owb.version}</version>
  <scope>test</scope>
</dependency>

<dependency>
  <groupId>org.apache.openwebbeans.test</groupId>
  <artifactId>cditest-owb</artifactId>
  <version>${owb.version}</version>
  <scope>test</scope>
</dependency>
```

Examples

An example is available [here](#).