CDI @Inject

Example cdi-basic can be browsed at https://github.com/apache/tomee/tree/master/examples/cdi-basic

To use `@Inject`, the first thing you need is a `META-INF/beans.xml` file in the module or jar. This effectively turns on CDI and allows the `@Inject` references to work. No `META-INF/beans.xml` no injection, period. This may seem overly strict, but it is not without reason. The CDI API is a bit greedy and does consume a fair about of resources by design.

When the container constructs a bean with an `@Inject` reference, it will first find or create the object that will be injected. For the sake of simplicity, the is example has a basic `Faculty` pojo with a no-arg constructor. Anyone referencing `@Inject Faculty` will get their own instance of `Faculty`. If the desire is to share the same instance of `Faculty`, see the concept of `scopes` — this is exactly what scopes are for.

Example

In this example we have an `@Stateless` bean `Course` with an `@Inject` reference to an object of type `Faculty`. When `Course` is created, the container will also create an instance of `Faculty`. The `@PostConstruct` will be called on the `Faculty`, then the `Faculty` instance will be injected into the `Course` bean. Finally, the `@PostConstruct` will be invoked on `Course` and then we're done. All instances will have been created.

The `CourseTest` test case drives this creation process by having `Course` injected into it in its `@Setup` method. By the time our `@Test` method is invoked, all the real work should be done and we should be ready to go. In the test case we do some basic asserts to ensure everything was constructed, all `@PostConstruct` methods called and everyting injected.

Faculty <small>a basic injectable pojo</small>

```
public class Faculty {

    private List<String> facultyMembers;

    private String facultyName;

    @PostConstruct
    public void initialize() {
        this.facultyMembers = new ArrayList<String>();
        facultyMembers.add("Ian Schultz");
        facultyMembers.add("Diane Reyes");
        facultyName = "Computer Science";
    }

    public List<String> getFacultyMembers() {
        return facultyMembers;
    }

    public String getFacultyName() {
        return facultyName;
    }

}
```

Course <small>a simple session bean</small>

```
@Stateless
public class Course {

    @Inject
    private Faculty faculty;

    private String courseName;

    private int capacity;

    @PostConstruct
    private void init() {
        assert faculty != null;

        // These strings can be externalized
        // We'll see how to do that later
        this.courseName = "CDI 101 - Introduction to CDI";
        this.capacity = 100;
    }

    public String getCourseName() {
        return courseName;
    }

    public int getCapacity() {
        return capacity;
    }

    public Faculty getFaculty() {
        return faculty;
    }
}
```

Test Case

```
public class CourseTest extends TestCase {

    @EJB
    private Course course;

    @Before
    public void setUp() throws Exception {
        EJBContainer.createEJBContainer().getContext().bind("inject", this);
    }

    @Test
    public void test() {

        // Was the EJB injected?
        assertTrue(course != null);

        // Was the Course @PostConstruct called?
        assertNotNull(course.getCourseName());
        assertTrue(course.getCapacity() > 0);

        // Was a Faculty instance injected into Course?
        final Faculty faculty = course.getFaculty();
        assertTrue(faculty != null);

        // Was the @PostConstruct called on Faculty?
        assertEquals(faculty.getFacultyName(), "Computer Science");
        assertEquals(faculty.getFacultyMembers().size(), 2);
    }
}
```

Running

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running org.superbiz.cdi.basic.CourseTest
Apache OpenEJB 4.0.0-beta-1    build: 20111002-04:06
http://tomee.apache.org/
INFO - openejb.home = /Users/dblevins/examples/cdi-basic
INFO - openejb.base = /Users/dblevins/examples/cdi-basic
INFO - Using 'javax.ejb.embeddable.EJBContainer=true'
INFO - Configuring Service(id=Default Security Service, type=SecurityService,
provider-id=Default Security Service)
INFO - Configuring Service(id=Default Transaction Manager, type=TransactionManager,
provider-id=Default Transaction Manager)
INFO - Found EjbModule in classpath: /Users/dblevins/examples/cdi-basic/target/classes
INFO - Beginning load: /Users/dblevins/examples/cdi-basic/target/classes
```

```
INFO - Configuring enterprise application: /Users/dblevins/examples/cdi-basic
INFO - Configuring Service(id=Default Managed Container, type=Container, provider-
id=Default Managed Container)
INFO - Auto-creating a container for bean cdi-basic.Comp: Container(type=MANAGED,
id=Default Managed Container)
INFO - Configuring Service(id=Default Stateless Container, type=Container, provider-
id=Default Stateless Container)
INFO - Auto-creating a container for bean Course: Container(type=STATELESS, id=Default
Stateless Container)
INFO - Enterprise application "/Users/dblevins/examples/cdi-basic" loaded.
INFO - Assembling app: /Users/dblevins/examples/cdi-basic
INFO - Jndi(name="java:global/cdi-basic/cdi-
basic.Comp!org.apache.openejb.BeanContext$Comp")
INFO - Jndi(name="java:global/cdi-basic/cdi-basic.Comp")
INFO - Jndi(name="java:global/cdi-basic/Course!org.superbiz.cdi.basic.Course")
INFO - Jndi(name="java:global/cdi-basic/Course")
INFO -
Jndi(name="java:global/EjbModule1833350875/org.superbiz.cdi.basic.CourseTest!org.super
biz.cdi.basic.CourseTest")
INFO - Jndi(name="java:global/EjbModule1833350875/org.superbiz.cdi.basic.CourseTest")
INFO - Created Ejb(deployment-id=Course, ejb-name=Course, container=Default Stateless
Container)
INFO - Created Ejb(deployment-id=cdi-basic.Comp, ejb-name=cdi-basic.Comp,
container=Default Managed Container)
INFO - Created Ejb(deployment-id=org.superbiz.cdi.basic.CourseTest, ejb-
name=org.superbiz.cdi.basic.CourseTest, container=Default Managed Container)
INFO - Started Ejb(deployment-id=Course, ejb-name=Course, container=Default Stateless
Container)
INFO - Started Ejb(deployment-id=cdi-basic.Comp, ejb-name=cdi-basic.Comp,
container=Default Managed Container)
INFO - Started Ejb(deployment-id=org.superbiz.cdi.basic.CourseTest, ejb-
name=org.superbiz.cdi.basic.CourseTest, container=Default Managed Container)
INFO - Deployed Application(path=/Users/dblevins/examples/cdi-basic)
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.126 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
```