

cdi-alternative-and-stereotypes

Example `cdi-alternative-and-stereotypes` can be browsed at <https://github.com/apache/tomee/tree/master/examples/cdi-alternative-and-stereotypes>

# Introduction

CDI is a revolution for Java EE world. This specification is the best one to avoid coupling between classes.

This example simply aims to override bindings at runtime to simplify mocking work.

It uses two kind of mocks: 1) a mock with no implementation in the classloader 2) a mock with an implementation in the classloader

The mock answer from CDI is called **alternative**.

Annotating `@Alternative` a class will mean it will replace any implementation if there is no other implementation or if it is forced (through `META-INF/beans.xml`).

## Code explanation

### main code

We use an EJB `Journey` to modelize a journey where the vehicle and the society can change. Here an EJB is used simply because it simplifies the test. A journey wraps the vehicle and society information.

We define then two interfaces to inject it in the `Journey` EJB: `Vehicle` and `Society`.

Finally we add an implementation for `Society` interface: `LowCostCompanie`.

If we don't go further `Journey` object will not be able to be created because no `Vehicle` implementation is available.

Note: if we suppose we have a `Vehicle` implementation, the injected Society should be `LowCostCompanie`.

### test code

The goal here is to test our `Journey` EJB. So we have to provide a `Vehicle` implementation: `SuperCar`.

We want to force the `Society` implementation (for any reason) by our test implementation: `AirOpenEJB`.

One solution could simply be to add `@Alternative` annotation on `AirOpenEJB` and activate it through the `META-INF/beans.xml` file.

Here we want to write more explicit code. So we want to replace the `@Alternative` annotation by `@Mock` one.

So we simply define an `@Mock` annotation for classes, resolvable at runtime which is a stereotype (`@Stereotype`) which replace `@Alternative`.

Here is the annotation:

```
@Stereotype // we define a stereotype
@Retention(RUNTIME) // resolvable at runtime
@Target(TYPE) // this annotation is a class level one
@Alternative // it replace @Alternative
public @interface Mock {}
```

Note: you can add more CDI annotations after `@Alternative` and it will get the behavior expected (the scope `for` instance).

So now we have our `@Mock` annotation which is a stereotype able to replace `@Alternative` annotation  
we simply add this annotation to our mocks.

If you run it now you'll have this exception:

```
javax.enterprise.inject.UnsatisfiedResolutionException: Api type
[org.superbiz.cdi.stereotype.Vehicle] is not found with the qualifiers
Qualifiers: [@javax.enterprise.inject.Default()]
for injection into Field Injection Point, field name : vehicle, Bean Owner :
[Journey, Name:null, WebBeans Type:ENTERPRISE, API
Types:[java.lang.Object,org.superbiz.cdi.stereotype.Journey],
Qualifiers:[javax.enterprise.inject.Any,javax.enterprise.inject.Default]]
```

It means the stereotype is not activated. To do it simply add it to your `META-INF/beans.xml`:

```
<alternatives>
  <stereotype>org.superbiz.cdi.stereotype.Mock</stereotype>
</alternatives>
```

Note: if you don't specify `'AirOpenEJB'` as `@Alternative` (done through our mock annotation) you'll get this exception:

```
Caused by: javax.enterprise.inject.AmbiguousResolutionException: There is more than
one api type with : org.superbiz.cdi.stereotype.Society with qualifiers : Qualifiers:
[@javax.enterprise.inject.Default()]
for injection into Field Injection Point, field name : society, Bean Owner :
[Journey, Name:null, WebBeans Type:ENTERPRISE, API
Types:[org.superbiz.cdi.stereotype.Journey,java.lang.Object],
Qualifiers:[javax.enterprise.inject.Any,javax.enterprise.inject.Default]]
found beans:
AirOpenEJB, Name:null, WebBeans Type:MANAGED, API
```

```
Types:[org.superbiz.cdi.stereotype.Society,org.superbiz.cdi.stereotype.AirOpenEJB,java
.lang.Object],
Qualifiers:[javax.enterprise.inject.Any,javax.enterprise.inject.Default]
LowCostCompanie, Name:null, WebBeans Type:MANAGED, API
Types:[org.superbiz.cdi.stereotype.Society,org.superbiz.cdi.stereotype.LowCostCompanie
,java.lang.Object],
Qualifiers:[javax.enterprise.inject.Any,javax.enterprise.inject.Default]
```

which simply means two implementations are available for the same injection point (`Journey.society`).

## # Conclusion

With CDI it is really easy to define annotations with a strong meaning. You can define business annotations or simply technical annotations to simplify your code (as we did with the mock annotation).

Note: if for instance you used qualifiers to inject societies you could have put all these qualifiers on the mock class or defined a `@SocietyMock` annotation to be able to inject the same implementation for all qualifiers in your tests.

## # Output

```
Running org.superbiz.cdi.stereotype.StereotypeTest
Apache OpenEJB 7.0.0-SNAPSHOT    build: 20111030-07:54
http://tomee.apache.org/
INFO - openejb.home = /opt/dev/openejb/openejb-trunk/examples/cdi-alternative-and-
stereotypes
INFO - openejb.base = /opt/dev/openejb/openejb-trunk/examples/cdi-alternative-and-
stereotypes
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: /opt/dev/openejb/openejb-trunk/examples/cdi-
alternative-and-stereotypes/target/test-classes
INFO - Found EjbModule in classpath: /opt/dev/openejb/openejb-trunk/examples/cdi-
alternative-and-stereotypes/target/classes
INFO - Beginning load: /opt/dev/openejb/openejb-trunk/examples/cdi-alternative-and-
stereotypes/target/test-classes
INFO - Beginning load: /opt/dev/openejb/openejb-trunk/examples/cdi-alternative-and-
stereotypes/target/classes
INFO - Configuring enterprise application: /opt/dev/openejb/openejb-
trunk/examples/cdi-alternative-and-stereotypes
INFO - Configuring Service(id=Default Managed Container, type=Container, provider-
id=Default Managed Container)
INFO - Auto-creating a container for bean cdi-alternative-and-stereotypes_test.Comp:
```

```
Container(type=MANAGED, id=Default Managed Container)
INFO - Configuring Service(id=Default Singleton Container, type=Container, provider-
id=Default Singleton Container)
INFO - Auto-creating a container for bean Journey: Container(type=SINGLETON,
id=Default Singleton Container)
INFO - Enterprise application "/opt/dev/openejb/openejb-trunk/examples/cdi-
alternative-and-stereotypes" loaded.
INFO - Assembling app: /opt/dev/openejb/openejb-trunk/examples/cdi-alternative-and-
stereotypes
INFO - Jndi(name="java:global/cdi-alternative-and-stereotypes/cdi-alternative-and-
stereotypes_test.Comp!org.apache.openejb.BeanContext$Comp")
INFO - Jndi(name="java:global/cdi-alternative-and-stereotypes/cdi-alternative-and-
stereotypes_test.Comp")
INFO - Jndi(name="java:global/cdi-alternative-and-stereotypes/cdi-alternative-and-
stereotypes.Comp!org.apache.openejb.BeanContext$Comp")
INFO - Jndi(name="java:global/cdi-alternative-and-stereotypes/cdi-alternative-and-
stereotypes.Comp")
INFO - Jndi(name="java:global/cdi-alternative-and-
stereotypes/Journey!org.superbiz.cdi.stereotype.Journey")
INFO - Jndi(name="java:global/cdi-alternative-and-stereotypes/Journey")
INFO -
Jndi(name="java:global/EjbModule162291475/org.superbiz.cdi.stereotype.StereotypeTest!o
rg.superbiz.cdi.stereotype.StereotypeTest")
INFO -
Jndi(name="java:global/EjbModule162291475/org.superbiz.cdi.stereotype.StereotypeTest")
INFO - Created Ejb(deployment-id=cdi-alternative-and-stereotypes_test.Comp, ejb-
name=cdi-alternative-and-stereotypes_test.Comp, container=Default Managed Container)
INFO - Created Ejb(deployment-id=cdi-alternative-and-stereotypes.Comp, ejb-name=cdi-
alternative-and-stereotypes.Comp, container=Default Managed Container)
INFO - Created Ejb(deployment-id=org.superbiz.cdi.stereotype.StereotypeTest, ejb-
name=org.superbiz.cdi.stereotype.StereotypeTest, container=Default Managed Container)
INFO - Created Ejb(deployment-id=Journey, ejb-name=Journey, container=Default
Singleton Container)
INFO - Started Ejb(deployment-id=cdi-alternative-and-stereotypes_test.Comp, ejb-
name=cdi-alternative-and-stereotypes_test.Comp, container=Default Managed Container)
INFO - Started Ejb(deployment-id=cdi-alternative-and-stereotypes.Comp, ejb-name=cdi-
alternative-and-stereotypes.Comp, container=Default Managed Container)
INFO - Started Ejb(deployment-id=org.superbiz.cdi.stereotype.StereotypeTest, ejb-
name=org.superbiz.cdi.stereotype.StereotypeTest, container=Default Managed Container)
INFO - Started Ejb(deployment-id=Journey, ejb-name=Journey, container=Default
Singleton Container)
INFO - Deployed Application(path=/opt/dev/openejb/openejb-trunk/examples/cdi-
alternative-and-stereotypes)
INFO - Undeploying app: /opt/dev/openejb/openejb-trunk/examples/cdi-alternative-and-
stereotypes
```