Decorators

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

Help us document this example! Click the blue pencil icon in the upper right to edit this page.

AccessDeniedException

```
package org.superbiz.cdi.decorators;

import javax.ejb.ApplicationException;

/**
 * @version $Revision$ $Date$
 */
@ApplicationException
public class AccessDeniedException extends RuntimeException {
    public AccessDeniedException(String s) {
        super(s);
    }
}
```

Calculator

```
package org.superbiz.cdi.decorators;

/**
 * @version $Revision$ $Date$
 */
public interface Calculator {

    public int add(int a, int b);

    public int subtract(int a, int b);

    public int multiply(int a, int b);

    public int divide(int a, int b);

    public int remainder(int a, int b);
}
```

CalculatorBean

```
package org.superbiz.cdi.decorators;

import javax.annotation.Resource;
import javax.ejb.SessionContext;
import javax.ejb.Stateless;
import javax.enterprise.inject.Produces;

@Stateless
public class CalculatorBean implements Calculator {

    @Produces
    @Resource
    private SessionContext sessionContext;

    public int add(int a, int b) {
        return a + b;
    }

    public int subtract(int a, int b) {
        return a - b;
    }

    public int multiply(int a, int b) {
        return a * b;
    }

    public int divide(int a, int b) {
        return a / b;
    }

    public int remainder(int a, int b) {
        return a % b;
    }
}
```

CalculatorLogging

```
package org.superbiz.cdi.decorators;

import javax.decorator.Decorator;
import javax.decorator.Delegate;
import javax.inject.Inject;
import java.util.logging.Logger;

@Decorator
public class CalculatorLogging implements Calculator {

    private Logger logger = Logger.getLogger("Calculator");

    @Inject
    @Delegate
    private Calculator calculator;

    @Override
    public int add(int a, int b) {
        logger.fine(String.format("add(%s, %s)", a, b));
        return calculator.add(a, b);
    }

    @Override
    public int subtract(int a, int b) {
        return calculator.subtract(a, b);
    }

    @Override
    public int multiply(int a, int b) {
        logger.finest(String.format("multiply(%s, %s)", a, b));
        return calculator.multiply(a, b);
    }

    @Override
    public int divide(int a, int b) {
        return calculator.divide(a, b);
    }

    @Override
    public int remainder(int a, int b) {
        logger.info(String.format("remainder(%s, %s)", a, b));
        return calculator.remainder(a, b);
    }
}
```

CalculatorSecurity

```
package org.superbiz.cdi.decorators;

import javax.decorator.Decorator;
import javax.decorator.Delegate;
import javax.ejb.SessionContext;
import javax.inject.Inject;

@Decorator
public class CalculatorSecurity implements Calculator {

    @Inject
    @Delegate
    private Calculator calculator;

    @Inject
    private SessionContext sessionContext;

    @Override
    public int add(int a, int b) {
        return calculator.add(a, b);
    }

    @Override
    public int subtract(int a, int b) {
        // Caller must pass a security check to call subtract
        if (!sessionContext.isCallerInRole("Manager")) throw new
AccessDeniedException(sessionContext.getCallerPrincipal().getName());

        return calculator.subtract(a, b);
    }

    @Override
    public int multiply(int a, int b) {
        return calculator.multiply(a, b);
    }

    @Override
    public int divide(int a, int b) {
        return calculator.divide(a, b);
    }

    @Override
    public int remainder(int a, int b) {
        return calculator.remainder(a, b);
    }
}
```

beans.xml

```
<beans>
  <!--
  Explicitly declaring decorators is required by the CDI specification.
  The order decorators are listed in the xml is the order in which they are invoked.
  -->
  <decorators>
    <class>org.superbiz.cdi.decorators.CalculatorSecurity</class>
    <class>org.superbiz.cdi.decorators.CalculatorLogging</class>
  </decorators>
</beans>
```

CalculatorTest

```
package org.superbiz.cdi.decorators;

import junit.framework.TestCase;

import javax.annotation.security.RunAs;
import javax.ejb.EJB;
import javax.ejb.Stateless;
import javax.ejb.embeddable.EJBContainer;
import java.util.concurrent.Callable;

public class CalculatorTest extends TestCase {

    @EJB
    private Calculator calculator;

    @EJB
    private ManagerBean manager;

    /**
     * Bootstrap the Embedded EJB Container
     *
     * @throws Exception
     */
    protected void setUp() throws Exception {
        EJBContainer.createEJBContainer().getContext().bind("inject", this);
    }

    /**
     * Test Add method
     */
    public void testAdd() {
```

```
        assertEquals(10, calculator.add(4, 6));
    }

    /**
     * Test Subtract method
     */
    public void testSubtract() {

        try {
            calculator.subtract(4, 6);

            fail("AccessDeniedException should have been thrown for unauthenticated
access");
        } catch (AccessDeniedException expected) {
            // pass
        }

        final int result = manager.call(new Callable<Integer>() {
            public Integer call() {
                return calculator.subtract(4, 6);
            }
        });

        assertEquals(-2, result);
    }

    /**
     * Test Multiply method
     */
    public void testMultiply() {

        assertEquals(24, calculator.multiply(4, 6));
    }

    /**
     * Test Divide method
     */
    public void testDivide() {

        assertEquals(2, calculator.divide(12, 6));
    }

    /**
     * Test Remainder method
     */
    public void testRemainder() {

        assertEquals(4, calculator.remainder(46, 6));
    }

    @Stateless
```

```
    @RunAs("Manager")
    public static class ManagerBean {

        public <V> V call(Callable<V> callable) {
            try {
                return callable.call();
            } catch (Exception e) {
                throw new RuntimeException(e);
            }
        }
    }
}
```

Running

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running org.superbiz.cdi.decorators.CalculatorTest
Apache OpenEJB 4.0.0-beta-1    build: 20111002-04:06
http://tomee.apache.org/
INFO - openejb.home = /Users/dblevins/examples/decorators
INFO - openejb.base = /Users/dblevins/examples/decorators
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/decorators/target/classes
INFO - Found EjbModule in classpath: /Users/dblevins/examples/decorators/target/test-
classes
INFO - Beginning load: /Users/dblevins/examples/decorators/target/classes
INFO - Beginning load: /Users/dblevins/examples/decorators/target/test-classes
INFO - Configuring enterprise application: /Users/dblevins/examples/decorators
WARN - Method 'lookup' is not available for 'javax.annotation.Resource'. Probably
using an older Runtime.
INFO - Configuring Service(id=Default Managed Container, type=Container, provider-
id=Default Managed Container)
INFO - Auto-creating a container for bean decorators.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 CalculatorBean: Container(type=STATELESS,
id=Default Stateless Container)
INFO - Enterprise application "/Users/dblevins/examples/decorators" loaded.
INFO - Assembling app: /Users/dblevins/examples/decorators
INFO -
```

```
Jndi(name="java:global/decorators/decorators.Comp!org.apache.openejb.BeanContext$Comp"
)
INFO - Jndi(name="java:global/decorators/decorators.Comp")
INFO -
Jndi(name="java:global/decorators/CalculatorBean!org.superbiz.cdi.decorators.Calculato
r")
INFO - Jndi(name="java:global/decorators/CalculatorBean")
INFO -
Jndi(name="java:global/decorators/ManagerBean!org.superbiz.cdi.decorators.CalculatorTe
st$ManagerBean")
INFO - Jndi(name="java:global/decorators/ManagerBean")
INFO -
Jndi(name="java:global/EjbModule628834558/org.superbiz.cdi.decorators.CalculatorTest!o
rg.superbiz.cdi.decorators.CalculatorTest")
INFO -
Jndi(name="java:global/EjbModule628834558/org.superbiz.cdi.decorators.CalculatorTest")
INFO - Created Ejb(deployment-id=CalculatorBean, ejb-name=CalculatorBean,
container=Default Stateless Container)
INFO - Created Ejb(deployment-id=decorators.Comp, ejb-name=decorators.Comp,
container=Default Managed Container)
INFO - Created Ejb(deployment-id=ManagerBean, ejb-name=ManagerBean, container=Default
Stateless Container)
INFO - Created Ejb(deployment-id=org.superbiz.cdi.decorators.CalculatorTest, ejb-
name=org.superbiz.cdi.decorators.CalculatorTest, container=Default Managed Container)
INFO - Started Ejb(deployment-id=CalculatorBean, ejb-name=CalculatorBean,
container=Default Stateless Container)
INFO - Started Ejb(deployment-id=decorators.Comp, ejb-name=decorators.Comp,
container=Default Managed Container)
INFO - Started Ejb(deployment-id=ManagerBean, ejb-name=ManagerBean, container=Default
Stateless Container)
INFO - Started Ejb(deployment-id=org.superbiz.cdi.decorators.CalculatorTest, ejb-
name=org.superbiz.cdi.decorators.CalculatorTest, container=Default Managed Container)
INFO - Deployed Application(path=/Users/dblevins/examples/decorators)
INFO - EJBContainer already initialized.  Call ejbContainer.close() to allow
reinitialization
Oct 29, 2011 11:41:04 AM org.apache.webbeans.decorator.DelegateHandler invoke
SEVERE: Exception in calling method : [subtract] in decorator class :
[org.superbiz.cdi.decorators.CalculatorSecurity]. Look in the log for target checked
exception.
org.superbiz.cdi.decorators.AccessDeniedException: guest
    at
org.superbiz.cdi.decorators.CalculatorSecurity.subtract(CalculatorSecurity.java:43)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
    at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
    at java.lang.reflect.Method.invoke(Method.java:597)
    at org.apache.webbeans.decorator.DelegateHandler.invoke(DelegateHandler.java:98)
    at org.apache.openejb.cdi.CdiInterceptor.invoke(CdiInterceptor.java:127)
    at org.apache.openejb.cdi.CdiInterceptor.access$000(CdiInterceptor.java:45)
    at org.apache.openejb.cdi.CdiInterceptor$1.call(CdiInterceptor.java:66)
```

```
    at org.apache.openejb.cdi.CdiInterceptor.aroundInvoke(CdiInterceptor.java:72)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
    at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
    at java.lang.reflect.Method.invoke(Method.java:597)
    at
org.apache.openejb.core.interceptor.ReflectionInvocationContext$Invocation.invoke(Refl
ectionInvocationContext.java:181)
    at
org.apache.openejb.core.interceptor.ReflectionInvocationContext.proceed(ReflectionInvo
cationContext.java:163)
    at
org.apache.openejb.core.interceptor.InterceptorStack.invoke(InterceptorStack.java:130)
    at
org.apache.openejb.core.stateless.StatelessContainer._invoke(StatelessContainer.java:2
26)
    at
org.apache.openejb.core.stateless.StatelessContainer.invoke(StatelessContainer.java:17
8)
    at
org.apache.openejb.core.ivm.EjbObjectProxyHandler.synchronizedBusinessMethod(EjbObject
ProxyHandler.java:255)
    at
org.apache.openejb.core.ivm.EjbObjectProxyHandler.businessMethod(EjbObjectProxyHandler
.java:235)
    at
org.apache.openejb.core.ivm.EjbObjectProxyHandler._invoke(EjbObjectProxyHandler.java:9
2)
    at
org.apache.openejb.core.ivm.BaseEjbProxyHandler.invoke(BaseEjbProxyHandler.java:284)
    at $Proxy44.subtract(Unknown Source)
    at org.superbiz.cdi.decorators.CalculatorTest.testSubtract(CalculatorTest.java:59)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
    at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
    at java.lang.reflect.Method.invoke(Method.java:597)
    at junit.framework.TestCase.runTest(TestCase.java:168)
    at junit.framework.TestCase.runBare(TestCase.java:134)
    at junit.framework.TestResult$1.protect(TestResult.java:110)
    at junit.framework.TestResult.runProtected(TestResult.java:128)
    at junit.framework.TestResult.run(TestResult.java:113)
    at junit.framework.TestCase.run(TestCase.java:124)
    at junit.framework.TestSuite.runTest(TestSuite.java:232)
    at junit.framework.TestSuite.run(TestSuite.java:227)
    at org.junit.internal.runners.JUnit38ClassRunner.run(JUnit38ClassRunner.java:83)
    at org.apache.maven.surefire.junit4.JUnit4TestSet.execute(JUnit4TestSet.java:35)
    at
org.apache.maven.surefire.junit4.JUnit4Provider.executeTestSet(JUnit4Provider.java:115
)
```

```
    at org.apache.maven.surefire.junit4.JUnit4Provider.invoke(JUnit4Provider.java:97)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
    at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
    at java.lang.reflect.Method.invoke(Method.java:597)
    at
org.apache.maven.surefire.booter.ProviderFactory$ClassLoaderProxy.invoke(ProviderFacto
ry.java:103)
    at $Proxy0.invoke(Unknown Source)
    at
org.apache.maven.surefire.booter.SurefireStarter.invokeProvider(SurefireStarter.java:1
50)
    at
org.apache.maven.surefire.booter.SurefireStarter.runSuitesInProcess(SurefireStarter.ja
va:91)
    at org.apache.maven.surefire.booter.ForkedBooter.main(ForkedBooter.java:69)
INFO - EJBContainer already initialized.  Call ejbContainer.close() to allow
reinitialization
INFO - EJBContainer already initialized.  Call ejbContainer.close() to allow
reinitialization
INFO - EJBContainer already initialized.  Call ejbContainer.close() to allow
reinitialization
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.338 sec

Results :

Tests run: 5, Failures: 0, Errors: 0, Skipped: 0
```