CDI Produces Disposes

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

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

ConsoleHandler

```java
package org.superbiz.cdi.produces.disposes;

public class ConsoleHandler implements LogHandler {

    private String name;

    public ConsoleHandler(String name) {
        this.name = name;
    }

    @Override
    public String getName() {
        return name;
    }

    @Override
    public void writeLog(String s) {
        System.out.printf("##### Handler: %s, Writing to the console!\n", getName());
    }
}
```

DatabaseHandler

```
package org.superbiz.cdi.produces.disposes;

public class DatabaseHandler implements LogHandler {

    private String name;

    public DatabaseHandler(String name) {
        this.name = name;
    }

    @Override
    public String getName() {
        return name;
    }

    @Override
    public void writeLog(String s) {
        System.out.printf("##### Handler: %s, Writing to the database!\n", getName());
        // Use connection to write log to database
    }
}
```

FileHandler

```
package org.superbiz.cdi.produces.disposes;

public class FileHandler implements LogHandler {

    private String name;

    public FileHandler(String name) {
        this.name = name;
    }

    @Override
    public String getName() {
        return name;
    }

    @Override
    public void writeLog(String s) {
        System.out.printf("##### Handler: %s, Writing to the file!\n", getName());
        // Write to log file
    }
}
```

LogFactory

```java
package org.superbiz.cdi.produces.disposes;

import javax.enterprise.inject.Disposes;
import javax.enterprise.inject.Produces;

public class LogFactory {

    private int type = 2;

    @Produces
    public LogHandler getLogHandler() {
        switch (type) {
            case 1:
                return new FileHandler("@Produces created FileHandler!");
            case 2:
                return new DatabaseHandler("@Produces created DatabaseHandler!");
            case 3:
            default:
                return new ConsoleHandler("@Produces created ConsoleHandler!");
        }
    }

    public void closeLogHandler(@Disposes LogHandler handler) {
        switch (type) {
            case 1:
                System.out.println("Closing File handler!");
                break;
            case 2:
                System.out.println("Closing DB handler!");
                break;
            case 3:
            default:
                System.out.println("Closing Console handler!");
        }
    }
}
```

Logger

```
package org.superbiz.cdi.produces.disposes;

public interface Logger {

    public void log(String s);

    public LogHandler getHandler();
}
```

LoggerImpl

```
package org.superbiz.cdi.produces.disposes;

import javax.inject.Inject;
import javax.inject.Named;

@Named("logger")
public class LoggerImpl implements Logger {

    @Inject
    private LogHandler handler;

    @Override
    public void log(String s) {
        getHandler().writeLog(s);
    }

    public LogHandler getHandler() {
        return handler;
    }
}
```

LogHandler

```
package org.superbiz.cdi.produces.disposes;

public interface LogHandler {

    public String getName();

    public void writeLog(String s);
}
```

beans.xml

```xml
<beans xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
                           http://java.sun.com/xml/ns/javaee/beans_1_0.xsd">

</beans>
```

LoggerTest

```java
package org.superbiz.cdi.produces.disposes;

import org.junit.After;
import org.junit.Before;
import org.junit.Test;

import javax.ejb.embeddable.EJBContainer;
import javax.inject.Inject;
import javax.naming.Context;

import static junit.framework.Assert.assertNotNull;
import static org.junit.Assert.assertFalse;
import static org.junit.Assert.assertTrue;

public class LoggerTest {

    @Inject
    Logger logger;

    private Context ctxt;

    @Before
    public void setUp() {
        try {
            ctxt = EJBContainer.createEJBContainer().getContext();
            ctxt.bind("inject", this);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    @After
    public void cleanUp() {
        try {
            ctxt.unbind("inject");
            ctxt.close();
```

```
        } catch (Exception e) {
            e.printStackTrace();
        }
    }


    @Test
    public void testLogHandler() {
        assertNotNull(logger);
        assertFalse("Handler should not be a ConsoleHandler", logger.getHandler()
instanceof ConsoleHandler);
        assertFalse("Handler should not be a FileHandler", logger.getHandler()
instanceof FileHandler);
        assertTrue("Handler should be a DatabaseHandler", logger.getHandler()
instanceof DatabaseHandler);
        logger.log("##### Testing write\n");
        logger = null;
    }


}
```

Running

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running org.superbiz.cdi.produces.disposes.LoggerTest
Apache OpenEJB 4.0.0-beta-1    build: 20111002-04:06
http://tomee.apache.org/
INFO - openejb.home = /Users/dblevins/examples/cdi-produces-disposes
INFO - openejb.base = /Users/dblevins/examples/cdi-produces-disposes
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-produces-
disposes/target/classes
INFO - Beginning load: /Users/dblevins/examples/cdi-produces-disposes/target/classes
INFO - Configuring enterprise application: /Users/dblevins/examples/cdi-produces-
disposes
INFO - Configuring Service(id=Default Managed Container, type=Container, provider-
id=Default Managed Container)
INFO - Auto-creating a container for bean cdi-produces-disposes.Comp:
Container(type=MANAGED, id=Default Managed Container)
INFO - Enterprise application "/Users/dblevins/examples/cdi-produces-disposes" loaded.
INFO - Assembling app: /Users/dblevins/examples/cdi-produces-disposes
INFO - Jndi(name="java:global/cdi-produces-disposes/cdi-produces-
disposes.Comp!org.apache.openejb.BeanContext$Comp")
```

```
INFO - Jndi(name="java:global/cdi-produces-disposes/cdi-produces-disposes.Comp")
INFO -
Jndi(name="java:global/EjbModule10202458/org.superbiz.cdi.produces.disposes.LoggerTest
!org.superbiz.cdi.produces.disposes.LoggerTest")
INFO -
Jndi(name="java:global/EjbModule10202458/org.superbiz.cdi.produces.disposes.LoggerTest
")
INFO - Created Ejb(deployment-id=cdi-produces-disposes.Comp, ejb-name=cdi-produces-
disposes.Comp, container=Default Managed Container)
INFO - Created Ejb(deployment-id=org.superbiz.cdi.produces.disposes.LoggerTest, ejb-
name=org.superbiz.cdi.produces.disposes.LoggerTest, container=Default Managed
Container)
INFO - Started Ejb(deployment-id=cdi-produces-disposes.Comp, ejb-name=cdi-produces-
disposes.Comp, container=Default Managed Container)
INFO - Started Ejb(deployment-id=org.superbiz.cdi.produces.disposes.LoggerTest, ejb-
name=org.superbiz.cdi.produces.disposes.LoggerTest, container=Default Managed
Container)
INFO - Deployed Application(path=/Users/dblevins/examples/cdi-produces-disposes)
##### Handler: @Produces created DatabaseHandler!, Writing to the database!
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.02 sec

Results :

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