Testcase Injection

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

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

Movie

```java
package org.superbiz.testinjection;

import javax.persistence.Entity;

@Entity
public class Movie {

    private String director;
    private String title;
    private int year;

    public Movie() {
    }

    public Movie(String director, String title, int year) {
        this.director = director;
        this.title = title;
        this.year = year;
    }

    public String getDirector() {
        return director;
    }

    public void setDirector(String director) {
        this.director = director;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public int getYear() {
        return year;
    }

    public void setYear(int year) {
        this.year = year;
    }

}
```

Movies

```java
package org.superbiz.testinjection;

import javax.ejb.Stateful;
import javax.ejb.TransactionAttribute;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.PersistenceContextType;
import javax.persistence.Query;
import java.util.List;

import static javax.ejb.TransactionAttributeType.MANDATORY;

//START SNIPPET: code
@Stateful
@TransactionAttribute(MANDATORY)
public class Movies {

    @PersistenceContext(unitName = "movie-unit", type = PersistenceContextType
.TRANSACTION)
    private EntityManager entityManager;

    public void addMovie(Movie movie) throws Exception {
        entityManager.persist(movie);
    }

    public void deleteMovie(Movie movie) throws Exception {
        entityManager.remove(movie);
    }

    public List<Movie> getMovies() throws Exception {
        Query query = entityManager.createQuery("SELECT m from Movie as m");
        return query.getResultList();
    }
}
```

persistence.xml

```
<persistence xmlns="http://java.sun.com/xml/ns/persistence" version="1.0">

  <persistence-unit name="movie-unit">
    <jta-data-source>movieDatabase</jta-data-source>
    <non-jta-data-source>movieDatabaseUnmanaged</non-jta-data-source>
    <class>org.superbiz.testinjection.Movie</class>

    <properties>
      <property name="openjpa.jdbc.SynchronizeMappings" value=
"buildSchema(ForeignKeys=true)"/>
    </properties>
  </persistence-unit>
</persistence>
```

MoviesTest

```
package org.superbiz.testinjection;

import junit.framework.TestCase;

import javax.annotation.Resource;
import javax.ejb.EJB;
import javax.ejb.embeddable.EJBContainer;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.transaction.UserTransaction;
import java.util.List;
import java.util.Properties;

//START SNIPPET: code
public class MoviesTest extends TestCase {

    @EJB
    private Movies movies;

    @Resource
    private UserTransaction userTransaction;

    @PersistenceContext
    private EntityManager entityManager;

    public void setUp() throws Exception {
        Properties p = new Properties();
        p.put("movieDatabase", "new://Resource?type=DataSource");
        p.put("movieDatabase.JdbcDriver", "org.hsqldb.jdbcDriver");
        p.put("movieDatabase.JdbcUrl", "jdbc:hsqldb:mem:moviedb");

        EJBContainer.createEJBContainer(p).getContext().bind("inject", this);
```

```
    }

    public void test() throws Exception {

        userTransaction.begin();

        try {
            entityManager.persist(new Movie("Quentin Tarantino", "Reservoir Dogs",
1992));
            entityManager.persist(new Movie("Joel Coen", "Fargo", 1996));
            entityManager.persist(new Movie("Joel Coen", "The Big Lebowski", 1998));

            List<Movie> list = movies.getMovies();
            assertEquals("List.size()", 3, list.size());

            for (Movie movie : list) {
                movies.deleteMovie(movie);
            }

            assertEquals("Movies.getMovies()", 0, movies.getMovies().size());
        } finally {
            userTransaction.commit();
        }

    }
}
```

Running

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running org.superbiz.testinjection.MoviesTest
Apache OpenEJB 4.0.0-beta-1    build: 20111002-04:06
http://tomee.apache.org/
INFO - openejb.home = /Users/dblevins/examples/testcase-injection
INFO - openejb.base = /Users/dblevins/examples/testcase-injection
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 - Configuring Service(id=movieDatabase, type=Resource, provider-id=Default JDBC
Database)
INFO - Found EjbModule in classpath: /Users/dblevins/examples/testcase-
injection/target/classes
INFO - Beginning load: /Users/dblevins/examples/testcase-injection/target/classes
INFO - Configuring enterprise application: /Users/dblevins/examples/testcase-injection
```

```
WARN - Method 'lookup' is not available for 'javax.annotation.Resource'. Probably
using an older Runtime.
INFO - Configuring Service(id=Default Stateful Container, type=Container, provider-
id=Default Stateful Container)
INFO - Auto-creating a container for bean Movies: Container(type=STATEFUL, id=Default
Stateful Container)
INFO - Configuring Service(id=Default Managed Container, type=Container, provider-
id=Default Managed Container)
INFO - Auto-creating a container for bean org.superbiz.testinjection.MoviesTest:
Container(type=MANAGED, id=Default Managed Container)
INFO - Configuring PersistenceUnit(name=movie-unit)
INFO - Auto-creating a Resource with id 'movieDatabaseNonJta' of type 'DataSource for
'movie-unit'.
INFO - Configuring Service(id=movieDatabaseNonJta, type=Resource, provider-
id=movieDatabase)
INFO - Adjusting PersistenceUnit movie-unit <non-jta-data-source> to Resource ID
'movieDatabaseNonJta' from 'movieDatabaseUnmanaged'
INFO - Enterprise application "/Users/dblevins/examples/testcase-injection" loaded.
INFO - Assembling app: /Users/dblevins/examples/testcase-injection
INFO - PersistenceUnit(name=movie-unit,
provider=org.apache.openjpa.persistence.PersistenceProviderImpl) - provider time 408ms
INFO - Jndi(name="java:global/testcase-
injection/Movies!org.superbiz.testinjection.Movies")
INFO - Jndi(name="java:global/testcase-injection/Movies")
INFO -
Jndi(name="java:global/EjbModule1583515396/org.superbiz.testinjection.MoviesTest!org.s
uperbiz.testinjection.MoviesTest")
INFO -
Jndi(name="java:global/EjbModule1583515396/org.superbiz.testinjection.MoviesTest")
INFO - Created Ejb(deployment-id=Movies, ejb-name=Movies, container=Default Stateful
Container)
INFO - Created Ejb(deployment-id=org.superbiz.testinjection.MoviesTest, ejb-
name=org.superbiz.testinjection.MoviesTest, container=Default Managed Container)
INFO - Started Ejb(deployment-id=Movies, ejb-name=Movies, container=Default Stateful
Container)
INFO - Started Ejb(deployment-id=org.superbiz.testinjection.MoviesTest, ejb-
name=org.superbiz.testinjection.MoviesTest, container=Default Managed Container)
INFO - Deployed Application(path=/Users/dblevins/examples/testcase-injection)
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.24 sec

Results :

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