

Persistence Fragment

Example persistence-fragment can be browsed at <https://github.com/apache/tomee/tree/master/examples/persistence-fragment>

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

Movie

```
package org.superbiz.injection.jpa;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;

@Entity
public class Movie {
    @Id
    @GeneratedValue
    private long id;
    private String director;
    private String title;
    private int year;

    public Movie() {
        // no-op
    }

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

    public long getId() {
        return id;
    }

    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;
}
}
```

persistence-fragment.xml

```
<persistence-fragment version="2.0">
  <persistence-unit-fragment name="movie-unit">
    <class>org.superbiz.injection.jpa.Movie</class>
    <exclude-unlisted-classes>true</exclude-unlisted-classes>
  </persistence-unit-fragment>
</persistence-fragment>
```

MoviesTest

```

package org.superbiz.injection.jpa;

import org.apache.openejb.assembler.classic.ReloadableEntityManagerFactory;
import org.junit.Test;

import javax.ejb.embeddable.EJBContainer;
import javax.naming.Context;
import javax.persistence.EntityManagerFactory;
import javax.persistence.PersistenceUnit;
import java.util.Properties;

import static org.junit.Assert.assertTrue;

public class MoviesTest {
    @PersistenceUnit
    private EntityManagerFactory emf;

    @Test
    public void test() throws Exception {
        final 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");

        final EJBContainer container = EJBContainer.createEJBContainer(p);
        final Context context = container.getContext();
        context.bind("inject", this);

        assertTrue(((ReloadableEntityManagerFactory) emf).getManagedClasses().
contains(Movie.class.getName()));

        container.close();
    }
}

```

persistence.xml

```
<persistence version="2.0"
  xmlns="http://java.sun.com/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
    http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd">
  <persistence-unit name="movie-unit">
    <jta-data-source>movieDatabase</jta-data-source>
    <non-jta-data-source>movieDatabaseUnmanaged</non-jta-data-source>
    <properties>
      <property name="openjpa.jdbc.SynchronizeMappings" value=
"buildSchema(ForeignKeys=true)"/>
    </properties>
  </persistence-unit>
</persistence>
```