

JPA Hibernate

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

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

## Movie

```
package org.superbiz.injection.h3jpa;

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

@Entity
public class Movie {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private long id;

    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

```

package org.superbiz.injection.h3jpa;

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

@Stateful
public class Movies {

    @PersistenceContext(unitName = "movie-unit", type = PersistenceContextType
.EXTENDED)
    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 version="1.0"
    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_1_0.xsd">
    <persistence-unit name="movie-unit">
        <provider>org.hibernate.ejb.HibernatePersistence</provider>
        <jta-data-source>movieDatabase</jta-data-source>
        <non-jta-data-source>movieDatabaseUnmanaged</non-jta-data-source>
        <properties>
            <property name="hibernate.hbm2ddl.auto" value="create-drop"/>
            <property name="hibernate.dialect" value="org.hibernate.dialect.HSQLDialect"/>
        </properties>
    </persistence-unit>
</persistence>
```

## MoviesTest

```

package org.superbiz.injection.h3jpa;

import junit.framework.TestCase;

import javax.ejb.embeddable.EJBContainer;
import javax.naming.Context;
import java.util.List;
import java.util.Properties;

/**
 * @version $Revision: 607077 $ $Date: 2007-12-27 06:55:23 -0800 (Thu, 27 Dec 2007) $
 */
public class MoviesTest extends TestCase {

    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 Context context = EJBContainer.createEJBContainer(p).getContext();
        Movies movies = (Movies) context.lookup("java:global/jpa-hibernate/Movies");

        movies.addMovie(new Movie("Quentin Tarantino", "Reservoir Dogs", 1992));
        movies.addMovie(new Movie("Joel Coen", "Fargo", 1996));
        movies.addMovie(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());
    }
}

```

## Running

-----  
T E S T S  
-----

```

Running org.superbiz.injection.h3jpa.MoviesTest
Apache OpenEJB 4.0.0-beta-1    build: 20111002-04:06
http://tomee.apache.org/
INFO - openejb.home = /Users/dblevins/examples/jpa-hibernate

```

```
INFO - openejb.base = /Users/dblevins/examples/jpa-hibernate
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/jpa-
hibernate/target/classes
INFO - Beginning load: /Users/dblevins/examples/jpa-hibernate/target/classes
INFO - Configuring enterprise application: /Users/dblevins/examples/jpa-hibernate
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.injection.h3jpa.MoviesTest:
Container(type=MANAGED, id=Default Managed Container)
INFO - Configuring PersistenceUnit(name=movie-unit,
provider=org.hibernate.ejb.HibernatePersistence)
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/jpa-hibernate" loaded.
INFO - Assembling app: /Users/dblevins/examples/jpa-hibernate
INFO - PersistenceUnit(name=movie-unit,
provider=org.hibernate.ejb.HibernatePersistence) - provider time 631ms
INFO - Jndi(name="java:global/jpa-
hibernate/Movies!org.superbiz.injection.h3jpa.Movies")
INFO - Jndi(name="java:global/jpa-hibernate/Movies")
INFO -
Jndi(name="java:global/EjbModule1235930463/org.superbiz.injection.h3jpa.MoviesTest!org
.superbiz.injection.h3jpa.MoviesTest")
INFO -
Jndi(name="java:global/EjbModule1235930463/org.superbiz.injection.h3jpa.MoviesTest")
INFO - Created Ejb(deployment-id=Movies, ejb-name=Movies, container=Default Stateful
Container)
INFO - Created Ejb(deployment-id=org.superbiz.injection.h3jpa.MoviesTest, ejb-
name=org.superbiz.injection.h3jpa.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.injection.h3jpa.MoviesTest, ejb-
name=org.superbiz.injection.h3jpa.MoviesTest, container=Default Managed Container)
INFO - Deployed Application(path=/Users/dblevins/examples/jpa-hibernate)
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.22 sec
```

Results :

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