Embedding

The basic process for embedding OpenEJB:

1. Add the OpenEJB libraries to your classpath
2. Ensure your EJB modules are discoverable
3. Use the LocalInitialContextFactory to boot OpenEJB

Important docs

- Application discovery via the classpath
- Embedded Configuration
- Configuring DataSources in Tests
- Configuring PersistenceUnits in Tests
- Configuring Containers in Tests
- Configuring Logging in Tests
- Alternate Descriptors
- Unit Testing Transactions
- TestCase with TestBean inner-class
- TestCase Injection (@LocalClient)

Examples

Unfortunately all the examples are not documented here yet, more examples can be browsed in version control or check it out with subversion (`svn co http://svn.apache.org/repos/asf/openejb/trunk/openejb3/examples`).

title	description	APIs used
Simple Stateless	Simple EJB3 **@Stateless** bean with local and remote business interfaces and unit test.	• javax.ejb.Remote • javax.ejb.Local • javax.ejb.Stateless
Simple Stateful	Simple EJB3 **@Stateful** bean with local and remote business interfaces and unit test.	• javax.ejb.Remote • javax.ejb.Stateful
EJB 3.1 Singleton	Shows two EJB 3.1 **@Singleton** beans. One configured to load on startup via **@Startup** and using Bean-Managed Concurrency and synchronization. Another using Container-Managed Concurrency via **@Lock(READ)** and **@Lock(WRITE)**.	• javax.annotation.PostConstruct • javax.annotation.PreDestroy • javax.ejb.ConcurrencyManagement • javax.ejb.Lock • javax.ejb.Singleton • javax.ejb.Startup • javax.ejb.ConcurrencyManagementType.BEAN • javax.ejb.LockType.READ • javax.ejb.LockType.WRITE

EJB 2.1 Compatibility	Shows an EJB 3.0 Stateful bean with Business interfaces and EJB 2.1 interfaces (now called "component" interfaces) using the @**LocalHome** and @**RemoteHome**. Four interfaces in total.	• javax.ejb.CreateException • javax.ejb.EJBHome • javax.ejb.EJBLocalHome • javax.ejb.EJBLocalObject • javax.ejb.EJBObject • javax.ejb.Init • javax.ejb.Local • javax.ejb.LocalHome • javax.ejb.Remote • javax.ejb.RemoteHome • javax.ejb.Remove • javax.ejb.Stateful
Injection of env-entry	Shows how @**Resource** can be used to inject **`env-entry`** values declared in an ejb-jar.xml file	• javax.annotation.Resource • javax.ejb.Local • javax.ejb.Remote • javax.ejb.Stateful
Injection of env-entry 2	Same as the above example but shows how to use a **properties file** to declare the injection values and demonstrates how to get injection of **more types** such as java.util.Date, java.lang.Class, java.net.URI and more. *OpenEJB specific feature*.	• javax.annotation.Resource • javax.ejb.Stateless
Injection of other EJBs	Shows use of @**EJB** in a Stateless to have another Stateless bean injected into it.	• javax.ejb.EJB • javax.ejb.Local • javax.ejb.Remote • javax.ejb.Stateless
Injection of DataSource	Shows use of @**Resource** to have a JDBC **DataSource** injected into a Stateful bean. The Stateful bean does basic INSERT, SELECT and DELETE SQL operations.	• javax.annotation.PostConstruct • javax.annotation.Resource • javax.ejb.Stateful • javax.sql.DataSource
Injection of EntityManager	Shows use of @**PersistenceContext** to have an **EntityManager** with an **EXTENDED** persistence context injected into a @Stateful bean. An EJB 3 @**Entity** bean is used with the EntityManager to create, persist and merge data to a database.	• javax.ejb.Stateful • javax.persistence.Entity • javax.persistence.EntityManager • javax.persistence.PersistenceContext • javax.persistence.PersistenceContextType.EXTENDED • javax.persistence.Query

Testing Transactions	Shows use of @PersistenceContext to have an **EntityManager** with an **TRANSACTION** persistence context injected into a **@Stateful** bean using the **@TransactionAttribute** annotation and a TestCase that runs test code in a JTA **Transaction**. An EJB 3 @Entity bean is used with the EntityManager to create, persist and merge data to a database.	• javax.ejb.Stateful • javax.ejb.Stateless • javax.ejb.TransactionAttribute • javax.ejb.TransactionAttributeType.MANDATORY • javax.ejb.TransactionAttributeType.REQUIRES_NEW • javax.persistence.Entity • javax.persistence.EntityManager • javax.persistence.PersistenceContext • javax.persistence.PersistenceContextType.TRANSACTION • javax.persistence.Query
Testing Security	Builds upon the Injection of EntityManager Example but adds the use of **@RolesAllowed** and **@PermitAll** in the @Stateful bean to restrict who can perform create, persist and remove operations on the EntityManager. Shows a TestCase using the **@RunAs** annotation to execute and test the bean code as various users.	• javax.annotation.security.PermitAll • javax.annotation.security.RolesAllowed • javax.annotation.security.RunAs • javax.ejb.EJBAccessException • javax.ejb.Stateful • javax.ejb.Stateless • javax.ejb.TransactionAttribute • javax.ejb.TransactionAttributeType.SUPPORTS • javax.persistence.Entity • javax.persistence.EntityManager • javax.persistence.PersistenceContext • javax.persistence.PersistenceContextType.EXTENDED • javax.persistence.Query
Embedded and Remotable	Demonstrates how to use an OpenEJB feature that allows people embedding OpenEJB into their applications to support remote clients in other VMs. This is not required for unit testing.	• javax.ejb.Remote • javax.ejb.Stateful

Helloworld Weblogic	Demonstrates OpenEJBs ability to understand and support the WebLogic deployment descriptors so people using that platform in production can still use OpenEJB in their IDE or build to unit test their EJB applications.	• javax.ejb. CreateException • javax.ejb. EJBLocalHome • javax.ejb. EJBLocalObject • javax.ejb. LocalHome • javax.ejb. Stateless
JSF Injection Support	Demonstrates OpenEJBs ability to inject EJB's into JSF managed beans.	• javax.ejb. Stateless
Struts with OpenEJB and Tomcat	Demonstrates the usage of Struts within an OpenEJB + Tomcat environment.	• javax.ejb. Stateless
Applets with OpenEJB	Demonstrates how an applet can communicate with a remote stateless session bean. The stateless session bean is deployed in an OpenEJB + Tomcat environment	• javax.ejb. Stateless