

# Injection Of Ejbs

Example `injection-of-ejbs` can be browsed at <https://github.com/apache/tomee/tree/master/examples/injection-of-ejbs>

This example shows how to use the `@EJB` annotation on a bean class to refer to other beans.

This functionality is often referred as dependency injection (see <http://www.martinfowler.com/articles/injection.html>), and has been recently introduced in Java EE 5.

In this particular example, we will create two session stateless beans

- a `DataStore` session bean
- a `DataReader` session bean

The `DataReader` bean uses the `DataStore` to retrieve some informations, and we will see how we can, inside the `DataReader` bean, get a reference to the `DataStore` bean using the `@EJB` annotation, thus avoiding the use of the JNDI API.

## **DataReader**

```

package org.superbiz.injection;

import javax.ejb.EJB;
import javax.ejb.Stateless;

/**
 * This is an EJB 3.1 style pojo stateless session bean
 * Every stateless session bean implementation must be annotated
 * using the annotation @Stateless
 * This EJB has 2 business interfaces: DataReaderRemote, a remote business
 * interface, and DataReaderLocal, a local business interface
 * <p/>
 * The instance variables 'dataStoreRemote' is annotated with the @EJB annotation:
 * this means that the application server, at runtime, will inject in this instance
 * variable a reference to the EJB DataStoreRemote
 * <p/>
 * The instance variables 'dataStoreLocal' is annotated with the @EJB annotation:
 * this means that the application server, at runtime, will inject in this instance
 * variable a reference to the EJB DataStoreLocal
 */
//START SNIPPET: code
@Stateless
public class DataReader {

    @EJB
    private DataStoreRemote dataStoreRemote;
    @EJB
    private DataStoreLocal dataStoreLocal;
    @EJB
    private DataStore dataStore;

    public String readDataFromLocalStore() {
        return "LOCAL:" + dataStoreLocal.getData();
    }

    public String readDataFromLocalBeanStore() {
        return "LOCALBEAN:" + dataStore.getData();
    }

    public String readDataFromRemoteStore() {
        return "REMOTE:" + dataStoreRemote.getData();
    }
}

```

## DataStore

```

package org.superbiz.injection;

import javax.ejb.LocalBean;
import javax.ejb.Stateless;

/**
 * This is an EJB 3 style pojo stateless session bean
 * Every stateless session bean implementation must be annotated
 * using the annotation @Stateless
 * This EJB has 2 business interfaces: DataStoreRemote, a remote business
 * interface, and DataStoreLocal, a local business interface
 */
//START SNIPPET: code
@Stateless
@LocalBean
public class DataStore implements DataStoreLocal, DataStoreRemote {

    public String getData() {
        return "42";
    }
}

```

## DataStoreLocal

```

package org.superbiz.injection;

import javax.ejb.Local;

/**
 * This is an EJB 3 local business interface
 * A local business interface may be annotated with the @Local
 * annotation, but it's optional. A business interface which is
 * not annotated with @Local or @Remote is assumed to be Local
 */
//START SNIPPET: code
@Local
public interface DataStoreLocal {

    public String getData();
}

```

## DataStoreRemote

```

package org.superbiz.injection;

import javax.ejb.Remote;

/**
 * This is an EJB 3 remote business interface
 * A remote business interface must be annotated with the @Remote
 * annotation
 */
//START SNIPPET: code
@Remote
public interface DataStoreRemote {

    public String getData();
}

```

## EjbDependencyTest

```

package org.superbiz.injection;

import junit.framework.TestCase;

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

/**
 * A test case for DataReaderImpl ejb, testing both the remote and local interface
 */
//START SNIPPET: code
public class EjbDependencyTest extends TestCase {

    public void test() throws Exception {
        final Context context = EJBContainer.createEJBContainer().getContext();

        DataReader dataReader = (DataReader) context.lookup("java:global/injection-of-
ejbs/DataReader");

        assertNotNull(dataReader);

        assertEquals("LOCAL:42", dataReader.readDataFromLocalStore());
        assertEquals("REMOTE:42", dataReader.readDataFromRemoteStore());
        assertEquals("LOCALBEAN:42", dataReader.readDataFromLocalBeanStore());
    }
}

```

# Running

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

```
Running org.superbiz.injection.EjbDependencyTest
Apache OpenEJB 4.0.0-beta-1    build: 20111002-04:06
http://tomee.apache.org/
INFO - openejb.home = /Users/dblevins/examples/injection-of-ejbs
INFO - openejb.base = /Users/dblevins/examples/injection-of-ejbs
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/injection-of-
ejbs/target/classes
INFO - Beginning load: /Users/dblevins/examples/injection-of-ejbs/target/classes
INFO - Configuring enterprise application: /Users/dblevins/examples/injection-of-ejbs
INFO - Configuring Service(id=Default Stateless Container, type=Container, provider-
id=Default Stateless Container)
INFO - Auto-creating a container for bean DataReader: Container(type=STATELESS,
id=Default Stateless 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.EjbDependencyTest:
Container(type=MANAGED, id=Default Managed Container)
INFO - Enterprise application "/Users/dblevins/examples/injection-of-ejbs" loaded.
INFO - Assembling app: /Users/dblevins/examples/injection-of-ejbs
INFO - Jndi(name="java:global/injection-of-
ejbs/DataReader!org.superbiz.injection.DataReader")
INFO - Jndi(name="java:global/injection-of-ejbs/DataReader")
INFO - Jndi(name="java:global/injection-of-
ejbs/DataStore!org.superbiz.injection.DataStore")
INFO - Jndi(name="java:global/injection-of-
ejbs/DataStore!org.superbiz.injection.DataStoreLocal")
INFO - Jndi(name="java:global/injection-of-
ejbs/DataStore!org.superbiz.injection.DataStoreRemote")
INFO - Jndi(name="java:global/injection-of-ejbs/DataStore")
INFO -
Jndi(name="java:global/EjbModule355598874/org.superbiz.injection.EjbDependencyTest!org
.superbiz.injection.EjbDependencyTest")
INFO -
Jndi(name="java:global/EjbModule355598874/org.superbiz.injection.EjbDependencyTest")
INFO - Created Ejb(deployment-id=DataReader, ejb-name=DataReader, container=Default
Stateless Container)
INFO - Created Ejb(deployment-id=DataStore, ejb-name=DataStore, container=Default
Stateless Container)
INFO - Created Ejb(deployment-id=org.superbiz.injection.EjbDependencyTest, ejb-
```

```
name=org.superbiz.injection.EjbDependencyTest, container=Default Managed Container)
INFO - Started Ejb(deployment-id=DataReader, ejb-name=DataReader, container=Default
Stateless Container)
INFO - Started Ejb(deployment-id=DataStore, ejb-name=DataStore, container=Default
Stateless Container)
INFO - Started Ejb(deployment-id=org.superbiz.injection.EjbDependencyTest, ejb-
name=org.superbiz.injection.EjbDependencyTest, container=Default Managed Container)
INFO - Deployed Application(path=/Users/dblevins/examples/injection-of-ejbs)
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.225 sec
```

Results :

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