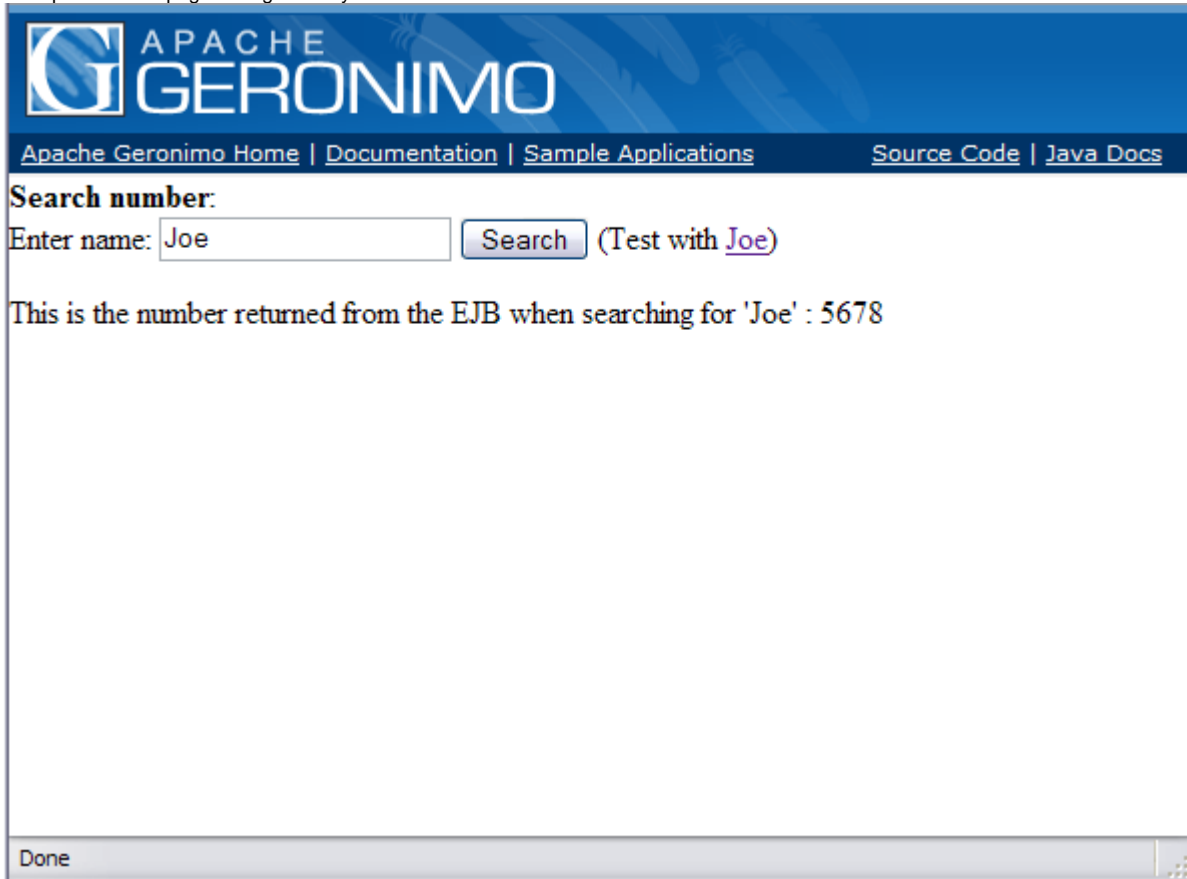


# Very simple Entity EJB example

{scrollbar}

## The Phone Book Bean Example

This is an example of a JSP-page calling an Entity Bean that uses annotations. The result looks like this:



## Application Contents

First, let us take a look at the **PhoneBook** Entity Bean that represents a table in the database. Each instance of **PhoneBook** is a record of the table. **PhoneBook.java** uses the

1. `@Entity` annotation to mark this class as an Entity Bean
2. `@Table` annotation to map the table name that is being represented by the Entity Bean.
3. `@Id` annotation to specify the primary key of the table.

And, as usual there is an empty constructor for the Entity Bean

```
javasolidPhoneBook.java package org.apache.geronimo.samples.myphonebookpak; import java.io.Serializable; import javax.persistence.Id; import javax.persistence.Entity; import javax.persistence.Table; @Entity @Table(name = "phonebook") public class PhoneBook implements Serializable { private String number; private String name; public PhoneBook() { } public PhoneBook(String name, String number) { this.name = name; this.number = number; } @Id public String getName() { return name; } public void setName(String name) { this.name = name; } public String getNumber() { return number; } public void setNumber(String number) { this.number = number; } }
```

**MyPhonebookLocal.java** is the business interface that drives the above mentioned Entity Bean.

```
javasolidMyPhonebookLocal.java package org.apache.geronimo.samples.myphonebookpak; import org.apache.geronimo.samples.myphonebookpak.PhoneBook; public interface MyPhonebookLocal { public PhoneBook findByPrimaryKey(String name); }
```

**MyPhonebookBean.java** is where the implementation of the local (and if there is, a the remote) interface. To explain what the annotations in this Stateless Session Bean means I will enumerate them:

1. `@Stateless` - tells Geronimo that this is a stateless session bean
2. `@PersistenceUnit` - tells Geronimo to retrieve a persistence unit defined in the **persistence.xml** and place it in the EntityManagerFactory. Note that PersistenceContext is used when you are directly obtaining a EntityManager. For an EntityManagerFactory use PersistenceUnit.

```

javasolidMyPhonebookBean.java package org.apache.geronimo.samples.myphonebookpak; import javax.ejb.Stateless; import javax.persistence.
PersistenceUnit; import javax.persistence.EntityManager; import javax.persistence.EntityManagerFactory; import org.apache.geronimo.samples.
myphonebookpak.PhoneBook; @Stateless public class MyPhonebookBean implements MyPhonebookLocal { @PersistenceUnit(unitName="PhonePU")
protected EntityManagerFactory emf; public MyPhonebookBean() {} public PhoneBook findByPrimaryKey(String name) { EntityManager em = emf.
createEntityManager(); PhoneBook phonebook = (PhoneBook)em.find(PhoneBook.class, name); em.close(); return phonebook; }

```

**index.jsp** is the JSP page that uses the EJB to access the database.

```

javasolidindex.jsp <%@ page contentType="text/html" import="org.apache.geronimo.samples.myphonebookpak.* , javax.naming.* " %> <% String
searchName = ""; if (request.getParameter("searchname") != null) { searchName=request.getParameter("searchname"); } %>
<html><head><title>Phonebook</title></head><body> <form action="index.jsp" > <b>Search number</b><br> Enter name: <input type="text" name="
searchname" value="<%=searchName%>"> <input type="submit" value="Search"> (Test with <a href="index.jsp?searchname=Joe">Joe</a>) </form> <%
if (! searchName.equals("")) { String number=""; try { Context context = new InitialContext(); MyPhonebookLocal myPhonebookLocal =
(MyPhonebookLocal)context.lookup("java:comp/env/ejb/MyPhonebookBean"); PhoneBook phonebook = myPhonebookLocal.findByPrimaryKey
(searchName); if(phonebook != null) { number = phonebook.getNumber(); } } catch (Exception e) { number=e.toString(); } out.println("This is the number
returned from the EJB when searching for "+searchName+" : " + number); } %> </body></html>

```

## Deployment Plans for EJB

**openejb-jar.xml** just specifies the module's information.

```

xmllsolidopenejb-jar.xml <?xml version="1.0" encoding="UTF-8"?> <openejb-jar xmlns="http://www.openejb.org/xml/ns/openejb-jar-2.1" xmlns:nam="
http://geronimo.apache.org/xml/ns/naming-1.1" xmlns:pkggen="http://www.openejb.org/xml/ns/pkggen-2.0" xmlns:sec="http://geronimo.apache.org/xml/ns
/security-1.1" xmlns:sys="http://geronimo.apache.org/xml/ns/deployment-1.2"> <sys:environment> <sys:moduleid> <sys:groupid>org.apache.geronimo.
samples/</sys:groupid> <sys:artifactid>MyPhonebookBean</sys:artifactid> <sys:version>1.0</sys:version> <sys:type>car</sys:type> </sys:moduleid> <
/sys:environment> </openejb-jar>

```

**persistence.xml** will specify the name of the PersistenceUnit. This name is used when referencing for the EntityManagerFactory. I have denoted it as **PhonePU**. For some reason I could not get it to reference with jta-data-source. So the alternative method is to explicitly specify the ConnectionURL, ConnectionDriverName, and ConnectionUserName. I added an extra property called SynchronizeMappings so that the data in the database will not be overwritten.

SEE BELOW FOR POSSIBLE SOLUTION

```

xmllsolidpersistence.xml <?xml version="1.0" encoding="UTF-8"?> <persistence xmlns="http://java.sun.com/xml/ns/persistence" xmlns:xsi="http://www.w3.
org/2001/XMLSchema-instance" version="1.0" xsi:schemaLocation="http://java.sun.com/xml/ns/persistence http://java.sun.com/xml/ns/persistence
/persistence_1_0.xsd"> <persistence-unit name="PhonePU"> <description>Phone Book</description> <provider>org.apache.openjpa.persistence.
PersistenceProviderImpl</provider> <class>org.apache.geronimo.samples.myphonebookpak.PhoneBook</class> <jta-data-source>PhoneBookPool</jta-
data-source> <non-jta-data-source>PhoneBookPool</non-jta-data-source> </persistence-unit> </persistence>

```

## Deployment Plans for Web-App

**web.xml** references the EJB by specifying the package to which the **MyPhonebookLocal** belongs to.

```

xmllsolidweb.xml <?xml version="1.0" encoding="UTF-8"?> <web-app id="WebApp_ID" version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
<display-name>MyPhonebookWeb</display-name> <welcome-file-list> <welcome-file>index.jsp</welcome-file> </welcome-file-list> <ejb-local-ref> <ejb-
ref-name>ejb/MyPhonebookBean</ejb-ref-name> <ejb-ref-type>Entity</ejb-ref-type> <local>org.apache.geronimo.samples.myphonebookpak.
MyPhonebookLocal</local> </ejb-local-ref> </web-app>

```

**geronimo-web.xml** specifies the module's information and the context-root for the web-app.

```

xmllsolidgeronimo-web.xml <?xml version="1.0" encoding="UTF-8"?> <web-app xmlns="http://geronimo.apache.org/xml/ns/j2ee/web-1.1" xmlns:nam="
http://geronimo.apache.org/xml/ns/naming-1.1" xmlns:sec="http://geronimo.apache.org/xml/ns/security-1.1" xmlns:sys="http://geronimo.apache.org/xml/ns
/deployment-1.1"> <sys:environment> <sys:moduleid> <sys:groupid>${pom.groupId}</sys:groupid> <sys:artifactid>${pom.artifactId}</sys:artifactid> <sys:
version>${version}</sys:version> <sys:type>war</sys:type> </sys:moduleid> </sys:environment> <context-root>/myphonebook</context-root> </web-
app>

```

## Deployment Plan for the application

**geronimo-application.xml** tells the application that there is a database pool that needs to be deployed as well. The db pool is defined in **PhoneBookPool.xml** and the driver that is needed in order to be deployed is the **tranql-connector-ra-1.3.rar** file--these two files will reside on the top level layer of the resultant EAR file.

```

xmllsolidgeronimo-application.xml <?xml version="1.0" encoding="UTF-8"?> <application xmlns="http://geronimo.apache.org/xml/ns/j2ee/application-1.1"
xmlns:sec="http://geronimo.apache.org/xml/ns/security-1.1" xmlns:sys="http://geronimo.apache.org/xml/ns/deployment-1.1" application-name="t6"> <sys:
environment> <sys:moduleid> <sys:groupid>${pom.groupId}</sys:groupid> <sys:artifactid>${pom.artifactId}</sys:artifactid> <sys:version>${version}</sys:
version> <sys:type>ear</sys:type> </sys:moduleid> </sys:environment> <module> <connector>tranql-connector-ra-1.3.rar</connector> <alt-
dd>PhoneBookPool.xml</alt-dd> </module> </application>

```

# Configuring, Building, and Deploying the Application

Download the **MyPhoneBook** application from the following link:  
[MyPhoneBook](#)

After decompressing the given file, the **myphonebook** directory will be created.

## Source Code

You can checkout the source code of this sample from SVN:

svn checkout <http://svn.apache.org/repos/asf/geronimo/samples/trunk/samples/myphonebook>

## Creating and Populating Database

After starting Apache Geronimo log into the console and follow the given steps to create the PhoneBookDB.

```
xmlsolidPhoneBook.sql CREATE TABLE phonebook ( name VARCHAR(255) PRIMARY KEY, number VARCHAR(255) ); INSERT INTO phonebook VALUES ('John', '1234'); INSERT INTO phonebook VALUES ('Joe', '5678');
```

1. Select **DB Manager** link from the Console Navigation in the left.
2. Give the database name as **PhoneBookDB** and click Create button.
3. Select **PhoneBookDB** to the Use DB field.
4. Open **PhoneBookDB.sql** in the **myphonebook/myphonebook-ear/src/main/resources** directory from a text editor.
5. Paste the content **PhoneBookDB.sql** to the SQL Commands text area and press **Run SQL** button.

## Building

Use a command prompt to navigate into the **myphonebook** directory and just give **mvn install** followed by **mvn site** command to build. It will create the **myphonebook-ear-2.0-SNAPSHOT.ear** under the **myphonebook** folder. Now, you are ready to deploy **myphonebook** application in the Geronimo Application server.

## Deploying the Application

Deploying sample application is pretty straight forward as we are going to use the Geronimo Console.

1. Scroll down to **Deploy New** from the Console Navigation panel.
2. Load **myphonebook-ear-2.0-SNAPSHOT.ear** from **myphonebook** folder in to the **Archive** input box.
3. Press **Install** button to deploy application in the server.

## MyPhoneBook Web Application

To test the sample web application open a browser and type <http://localhost:8080/myphonebook>.

## Untested Instructions for using jta-datasource in persistence.xml

This app does not use openjpa sequences so apparently you can get by with only a jta-datasource. In my experience apps that do use openjpa sequences to supply primary key values also need a non-jta-datasource. When deploying such a non-jta-datasource check the plan and make sure it says `<no-transaction/>` rather than `<local-transaction/>` or `<xa-transaction/>`.

For this app, use the console to deploy a datasource using the database of your choice. You need to supply a name such as "MyDS" for your datasource. At the end you should have a name for the module your datasource is in such as console.dbpool/MyDS/1.0/rar. You need to do two things so geronimo can hook up to your datasource:

1. Include the module name in the app dependencies so geronimo knows to look in the new datasource module for the datasource. So, the openejb-jar.xml should look something like:

```
xmlsolidopenejb-jar.xml <?xml version="1.0" encoding="UTF-8"?> <openejb-jar xmlns="http://www.openejb.org/xml/ns/openejb-jar-2.1" xmlns:nam="http://geronimo.apache.org/xml/ns/naming-1.1" xmlns:pkggen="http://www.openejb.org/xml/ns/pkggen-2.0" xmlns:sec="http://geronimo.apache.org/xml/ns/security-1.1" xmlns:sys="http://geronimo.apache.org/xml/ns/deployment-1.2"> <sys:environment> <sys:moduleId> <sys:groupId>org.apache.geronimo.samples</sys:groupId> <sys:artifactId>MyPhonebookBean</sys:artifactId> <sys:version>1.0</sys:version> <sys:type>car</sys:type> </sys:moduleId> <sys:dependencies> <sys:dependency> <sys:groupId>console.dbpool</sys:groupId> <sys:artifactId>MyDS</sys:artifactId> <sys:version>1.0</sys:version> <sys:type>rar</sys:type> </sys:dependency> </sys:dependencies> </sys:environment> </openejb-jar>
```

2. specify the name of the datasource in persistence.xml, something like

```
xmlsolidpersistence.xml <?xml version="1.0" encoding="UTF-8"?> <persistence xmlns="http://java.sun.com/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0" xsi:schemaLocation="http://java.sun.com/xml/ns/persistence http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd"> <persistence-unit name="PhonePU"> <description>Phone Book</description> <provider>org.apache.openjpa.persistence.PersistenceProviderImpl</provider> <jta-datasource>MyDS</jta-datasource> <class>org.apache.geronimo.samples.myphonebookpak.PhoneBook</class> <properties> <property name="openjpa.jdbc.SynchronizeMappings" value="false" /> </properties> </persistence-unit> <!-- <jta-data-source>PhoneBookPool</jta-data-source> <non-jta-data-source>NoTXPhoneBookPool</non-jta-data-source> --> </persistence>
```