

Hello World



This page shows the basic steps required to create, build, and run an EJB and EJB client in its most minimum form. It does not hide steps or rely on special build tools or IDEs and is about the most stripped down you can get.

See the [Examples](#) page for a full list of examples that range from *[@Stateless|Simple Stateless Example]* and *[@Stateful|Simple Stateful Example]* beans, to *[Dependency Injection|Injection of env-entry Example]* , *JDBC [DataSources|Injection of DataSource Example]* , *JPA [EntityManagers|Injection of EntityManager Example]* and more.

A basic EJB example

Here are some basic steps you need to perform to get started with OpenEJB

1. Download and install OpenEJB
2. Setup your development environment
3. Write an EJB
4. Write an EJB client
5. Start the server
6. Deploy the EJB
7. Run the client
8. Stop the server

Download and install OpenEJB

This example pertains to OpenEJB 3.0 which can be [downloaded here](#) . Once you have downloaded OpenEJB, you can then simply extract the contents of the downloaded file to whichever directory you want to install OpenEJB in.

After extracting the file contents, you should now see a directory named `openejb-3.0`. If you look under this directory, you will find a few more directories: - *bin*: Contains commands to start/stop the server (You can also do a lot of other stuff like deploy/undeploy, but we will just talk about things needed to get you started) - *lib*: Contains several jar files (you only need of few of these jars in your classpath to do EJB development) - *apps*: Once you create your EJB's and jar them up, you can place your jar file in this directory and start the server. The server will automatically deploy all the EJB's contained in this JAR. - *conf*: This directory contains all the configuration files. Although you may not see any file except for a README.txt file right now, but after you start the server, the required configuration files

will be automatically created. It is highly recommended to read the README.txt file under this directory - /logs: Contains log files.

Setup your development environment

Create a working directory

Assuming you are in your home directory, create a directory named projects

```
karan@poweredge:~$ mkdir projects
```

Go to the projects directory

```
karan@poweredge:~$ cd projects
```

We will do all our work in this directory.

Install Java

Download and install Java (version 5 or higher). Also set it up so that you can run the java and javac commands from any directory

Set OPENEJB_HOME

We will setup this variable to refer to the openejb install location.

```
karan@poweredge:~/projects$ export
```

```
OPENEJB_HOME=/home/karan/install/openejb-3.0
```

Write an EJB

Whatever files you create should be placed under the projects directory

Create the Remote Interface

Using your favorite editor, create a file named Hello.java (shown below)

```
package org.acme;
```

```
import javax.ejb.Remote;
@Remote
public interface Hello{
    public String sayHello();
}
```

Create the Bean Class

Now create a file named HelloBean.java (shown below)

```
package org.acme;
import javax.ejb.Stateless;
@Stateless
public class HelloBean implements Hello{
    public String sayHello(){
        return "Hello World!!!!";
    }
}
```

Compile the source code

Since we have imported the javax.ejb.Stateless and javax.ejb.Remote annotations, we need these in our classpath to compile our source code. These annotations can be found in the \$OPENEJB_HOME/lib/javaee-5.0-1.jar. Lets compile our source (make sure you are in the projects directory)

```
karan@poweredge:~/projects$ javac -cp $OPENEJB_HOME/lib/javaee-5.0-1.jar -d
```

```
. *.java
```

The above will compile all the .java files and also create the required packages. You should now see a package named org under the projects directory. All class files should be under org/acme directory.

Package the EJB

To package the EJB into a JAR, run the following command while you are in the projects directory

```
karan@poweredge:~/projects$ jar cvf hello.jar org
```

The above command will package everything under the org directory (including the org directory itself) into a jar file named hello.jar. Below is the output from running the above command:

```
karan@poweredge:~/projects$ jar cvf hello.jar org
added manifest
adding: org/(in = 0) (out= 0)(stored 0%)
adding: org/acme/(in = 0) (out= 0)(stored 0%)
adding: org/acme/Hello.class(in = 203) (out= 168)(deflated 17%)
adding: org/acme/HelloBean.class(in = 383) (out= 275)(deflated 28%)
```

Write an EJB Client

Now we will write a Client class which will lookup the EJB , invoke the sayHello() business method and print the value returned from the method. While you are in the projects directory, create a new file named HelloClient.java . Add the following to this file:

```
package org.acme;
import java.util.Properties;
import javax.naming.InitialContext;
import javax.naming.Context;
import javax.rmi.PortableRemoteObject;
public class HelloClient{
    public static void main(String[]
```

```
args) throws Exception{ Properties props = new Properties();
```

```
props.put(Context.INITIAL_CONTEXT_FACTORY,"org.apache.openejb.client.RemoteInitialContextFactory");
props.put(Context.PROVIDER_URL,"ejbd://127.0.0.1:4201"); Context ctx = new InitialContext(props); Object ref = ctx.lookup("HelloBeanRemote"); Hello h =
(Hello)PortableRemoteObject.narrow(ref,Hello.class); String result = h.sayHello(); System.out.println(result); } }
```

Compile HelloClient.java

Run the following command:

```
karan@poweredge:~/projects$ javac -d . HelloClient.java
```

Start the Server

Go to the OpenEJB install directory (i.e. OPENEJB_HOME) and run the following command:

```
karan@poweredge:~/install/openejb-3.0$ bin/openejb start
```

Once the Server starts, you will see an output similar to the below in your console:

```
karan@poweredge:~/install/openejb-3.0$ bin/openejb start
Apache OpenEJB 3.0      build: 20070926-12:34
http://openejb.apache.org/
OpenEJB ready.
[OPENEJB:init]
```

```
OpenEJB Remote Server * Starting Services * NAME IP PORT
httpjbd 0.0.0.0 4204
telnet 0.0.0.0 4202
ejbd 0.0.0.0 4201
hsqldb 0.0.0.0 9001
admin thread 0.0.0.0 4200
----- Ready!
```

Take out a minute to browse through the conf and logs directories. You should now see some configuration and log files under the respective directories.

Deploy the EJB

We will now use the deploy command to deploy the EJB in hello.jar. While you are in the projects directory, run the following command:

```
karan@poweredge:~/projects$ $OPENEJB_HOME/bin/openejb deploy hello.jar
```

The above command should give you the following output:

```
karan@poweredge:~/projects$ $OPENEJB_HOME/bin/openejb deploy hello.jar
Application deployed successfully at "hello.jar"
App(id=/home/karan/projects/hello.jar)
```

```
EjbJar(id=hello.jar, path=/home/karan/projects/hello.jar)
Ejb(ejb-name=HelloBean, id=HelloBean)
  Jndi(name=HelloBeanRemote)
```

Notice how the output neatly lays out various deployment details. One thing you might want to note from the output is the JNDI name. This is the JNDI name we used in the client to lookup the EJB

Run the Client

While you are in the projects directory, run the following command to run the client:

```
karan@poweredge:~/projects$ java -cp
```

```
$OPENEJB_HOME/lib/openejb-client-3.0.jar:$OPENEJB_HOME/lib/javaee-5.0-1.jar:. org.acme.HelloClient
```

The above should give you the following output:

```
Hello World!!!!
```

Help! , it didn't work for me!!.

No problem, we are here to help. Just send us an email at users@tomEE.apache.org. If possible, send us the contents of logs/openejb.log file in the email.

Looking for more?

More EJB 3.0 examples, sample applications, tutorials and howtos available [here](#) .

Copyright © 2013 The Apache Software Foundation, Licensed under the Apache License, Version 2.0. Apache TomEE, TomEE, Apache, the Apache feather logo, and the Apache TomEE project logo are trademarks of The Apache Software Foundation. All other marks mentioned may be trademarks or registered trademarks of their respective owners.