

Telephone Stateful

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

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

This example shows how to use OpenEJB's remoting capabilities in an embedded scenario.

The basic recipe is the same for a standard embedded scenario but with these added ingredients:

- `openejb.embedded.remotable` property
- `openejb-ejbd` jar

While creating the `InitialContext`, pass in the `openejb.embedded.remotable` property with the value of `"true"`. When this is seen by the `LocalInitialContextFactory`, it will boot up the `Server ServiceManager` in the VM which will in turn look for `ServerServices` in the classpath.

Provided you have the `openejb-ejbd` jar in your classpath along with its dependencies (`openejb-server`, `openejb-client`, `openejb-core`), then those services will be brought online and remote clients will be able to connect into your vm and invoke beans.

If you want to add more `ServerServices` such as the http version of the `ejbd` protocol you'd simply add the `openejb-http` jar to your classpath. A number of `ServerServices` are available currently:

- `openejb-ejbd`
- `openejb-http`
- `openejb-telnet`
- `openejb-derbynet`
- `openejb-hsql`
- `openejb-activemq`

Telephone

```
package org.superbiz.telephone;  
  
public interface Telephone {  
    void speak(String words);  
    String listen();  
}
```

TelephoneBean

```

package org.superbiz.telephone;

import javax.ejb.Remote;
import javax.ejb.Stateful;
import java.util.ArrayList;
import java.util.List;

@Remote
@Stateful
public class TelephoneBean implements Telephone {

    private static final String[] answers = {
        "How nice.",
        "Oh, of course.",
        "Interesting.",
        "Really?",
        "No.",
        "Definitely.",
        "I wondered about that.",
        "Good idea.",
        "You don't say!",
    };

    private List<String> conversation = new ArrayList<String>();

    public void speak(String words) {
        conversation.add(words);
    }

    public String listen() {
        if (conversation.size() == 0) {
            return "Nothing has been said";
        }

        String lastThingSaid = conversation.get(conversation.size() - 1);
        return answers[Math.abs(lastThingSaid.hashCode()) % answers.length];
    }
}

```

TelephoneTest

```

package org.superbiz.telephone;

import junit.framework.TestCase;

import javax.naming.Context;
import javax.naming.InitialContext;
import java.util.Properties;

```

```

/**
 * @version $Rev: 1090810 $ $Date: 2011-04-10 07:49:26 -0700 (Sun, 10 Apr 2011) $
 */
public class TelephoneTest extends TestCase {

    protected void setUp() throws Exception {
        Properties properties = new Properties();
        properties.setProperty(Context.INITIAL_CONTEXT_FACTORY,
"org.apache.openejb.core.LocalInitialContextFactory");
        properties.setProperty("openejb.embedded.remotable", "true");
        // Uncomment these properties to change the defaults
        //properties.setProperty("ejbd.port", "4202");
        //properties.setProperty("ejbd.bind", "localhost");
        //properties.setProperty("ejbd.threads", "200");
        //properties.setProperty("ejbd.disabled", "false");
        //properties.setProperty("ejbd.only_from", "127.0.0.1,192.168.1.1");

        new InitialContext(properties);
    }

    /**
     * Lookup the Telephone bean via its remote interface but using the
     LocalInitialContextFactory
     *
     * @throws Exception
     */
    public void testTalkOverLocalNetwork() throws Exception {

        Properties properties = new Properties();
        properties.setProperty(Context.INITIAL_CONTEXT_FACTORY,
"org.apache.openejb.core.LocalInitialContextFactory");
        InitialContext localContext = new InitialContext(properties);

        Telephone telephone = (Telephone) localContext.lookup("TelephoneBeanRemote");

        telephone.speak("Did you know I am talking directly through the embedded
container?");

        assertEquals("Interesting.", telephone.listen());

        telephone.speak("Yep, I'm using the bean's remote interface but since the ejb
container is embedded " +
            "in the same vm I'm just using the LocalInitialContextFactory.");

        assertEquals("Really?", telephone.listen());

        telephone.speak("Right, you really only have to use the
RemoteInitialContextFactory if you're in a different vm.");
    }
}

```

```

        assertEquals("Oh, of course.", telephone.listen());
    }

    /**
     * Lookup the Telephone bean via its remote interface using the
     RemoteInitialContextFactory
     *
     * @throws Exception
     */
    public void testTalkOverRemoteNetwork() throws Exception {
        Properties properties = new Properties();
        properties.setProperty(Context.INITIAL_CONTEXT_FACTORY,
"org.apache.openejb.client.RemoteInitialContextFactory");
        properties.setProperty(Context.PROVIDER_URL, "ejbd://localhost:4201");
        InitialContext remoteContext = new InitialContext(properties);

        Telephone telephone = (Telephone) remoteContext.lookup("TelephoneBeanRemote");

        telephone.speak("Is this a local call?");

        assertEquals("No.", telephone.listen());

        telephone.speak("This would be a lot cooler if I was connecting from another
VM then, huh?");

        assertEquals("I wondered about that.", telephone.listen());

        telephone.speak("I suppose I should hangup and call back over the
LocalInitialContextFactory.");

        assertEquals("Good idea.", telephone.listen());

        telephone.speak("I'll remember this though in case I ever have to call you
accross a network.");

        assertEquals("Definitely.", telephone.listen());
    }
}

```

Running

T E S T S

```

Running org.superbiz.telephone.TelephoneTest
Apache OpenEJB 4.0.0-beta-1    build: 20111002-04:06
http://tomee.apache.org/
INFO - openejb.home = /Users/dblevins/examples/telephone-stateful
INFO - openejb.base = /Users/dblevins/examples/telephone-stateful
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/telephone-
stateful/target/classes
INFO - Beginning load: /Users/dblevins/examples/telephone-stateful/target/classes
INFO - Configuring enterprise application: /Users/dblevins/examples/telephone-
stateful/classpath.ear
INFO - Configuring Service(id=Default Stateful Container, type=Container, provider-
id=Default Stateful Container)
INFO - Auto-creating a container for bean TelephoneBean: Container(type=STATEFUL,
id=Default Stateful Container)
INFO - Enterprise application "/Users/dblevins/examples/telephone-
stateful/classpath.ear" loaded.
INFO - Assembling app: /Users/dblevins/examples/telephone-stateful/classpath.ear
INFO - Jndi(name=TelephoneBeanRemote) --> Ejb(deployment-id=TelephoneBean)
INFO - Jndi(name=global/classpath.ear/telephone-
stateful/TelephoneBean!org.superbiz.telephone.Telephone) --> Ejb(deployment-
id=TelephoneBean)
INFO - Jndi(name=global/classpath.ear/telephone-stateful/TelephoneBean) -->
Ejb(deployment-id=TelephoneBean)
INFO - Created Ejb(deployment-id=TelephoneBean, ejb-name=TelephoneBean,
container=Default Stateful Container)
INFO - Started Ejb(deployment-id=TelephoneBean, ejb-name=TelephoneBean,
container=Default Stateful Container)
INFO - Deployed Application(path=/Users/dblevins/examples/telephone-
stateful/classpath.ear)
INFO - Initializing network services
INFO - Creating ServerService(id=admin)
INFO - Creating ServerService(id=ejbd)
INFO - Creating ServerService(id=ejbds)
INFO - Initializing network services
    ** Starting Services **
    NAME                IP            PORT
    admin thread        127.0.0.1    4200
    ejbd                 127.0.0.1    4201
    ejbd                 127.0.0.1    4203
    -----
Ready!
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.448 sec

Results :

Tests run: 2, Failures: 0, Errors: 0, Skipped: 0

```