

Simple Stateful

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

This example demonstrates a simple deployment of a Stateful session bean.

NOTE	"As its name suggests, a stateful session bean is similar to an interactive session. A stateful session bean is not shared;
-------------	---

it can have only one client, in the same way that an interactive session can have only one user. When the client terminates, its stateful session bean appears to terminate and is no longer associated with the client."

The **Counter** class is a Stateful session bean that maintains a state in a form of a **count** integer field. It exposes three methods: **count()**, **increment()** and **reset()** to manipulate and view its state.

Typically, Stateful and Stateless beans implement Local and/or Remote interfaces to determine which methods should be exposed. In this case, the bean is using a no-interface view, which means that all public methods are exposed as a local view.

Counter

```

package org.superbiz.counter;

import javax.ejb.Stateful;

/**
 * This is an EJB 3 style pojo stateful session bean
 * Every stateful session bean implementation must be annotated
 * using the annotation @Stateful
 * This EJB has 2 business interfaces: CounterRemote, a remote business
 * interface, and CounterLocal, a local business interface
 * <p/>
 * Per EJB3 rules when the @Remote or @Local annotation isn't present
 * in the bean class (this class), all interfaces are considered
 * local unless explicitly annotated otherwise. If you look
 * in the CounterRemote interface, you'll notice it uses the @Remote
 * annotation while the CounterLocal interface is not annotated relying
 * on the EJB3 default rules to make it a local interface.
 */
//START SNIPPET: code
@Stateful
public class Counter {

    private int count = 0;

    public int count() {
        return count;
    }

    public int increment() {
        return ++count;
    }

    public int reset() {
        return (count = 0);
    }
}

```

CounterTest

The `Counter` class is tested by obtaining a `Context` object and performing a JNDI lookup on it, to retrieve an instance of the `Counter` bean. After some state manipulation, a new instance is fetched from the container and we can see that it's a new instance.

```

package org.superbiz.counter;

import junit.framework.TestCase;

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

public class CounterTest extends TestCase {

    //START SNIPPET: local
    public void test() throws Exception {

        final Context context = EJBContainer.createEJBContainer().getContext();

        Counter counterA = (Counter) context.lookup("java:global/simple-
stateful/Counter");

        assertEquals(0, counterA.count());
        assertEquals(0, counterA.reset());
        assertEquals(1, counterA.increment());
        assertEquals(2, counterA.increment());
        assertEquals(0, counterA.reset());

        counterA.increment();
        counterA.increment();
        counterA.increment();
        counterA.increment();

        assertEquals(4, counterA.count());

        // Get a new counter
        Counter counterB = (Counter) context.lookup("java:global/simple-
stateful/Counter");

        // The new bean instance starts out at 0
        assertEquals(0, counterB.count());
    }
    //END SNIPPET: local
}

```

Running

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

```
Running org.superbiz.counter.CounterTest
Apache OpenEJB 4.0.0-beta-1    build: 20111002-04:06
http://tomee.apache.org/
INFO - openejb.home = /Users/dblevins/examples/simple-stateful
INFO - openejb.base = /Users/dblevins/examples/simple-stateful
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/simple-
stateful/target/classes
INFO - Beginning load: /Users/dblevins/examples/simple-stateful/target/classes
INFO - Configuring enterprise application: /Users/dblevins/examples/simple-stateful
INFO - Configuring Service(id=Default Stateful Container, type=Container, provider-
id=Default Stateful Container)
INFO - Auto-creating a container for bean Counter: Container(type=STATEFUL, id=Default
Stateful Container)
INFO - Configuring Service(id=Default Managed Container, type=Container, provider-
id=Default Managed Container)
INFO - Auto-creating a container for bean org.superbiz.counter.CounterTest:
Container(type=MANAGED, id=Default Managed Container)
INFO - Enterprise application "/Users/dblevins/examples/simple-stateful" loaded.
INFO - Assembling app: /Users/dblevins/examples/simple-stateful
INFO - Jndi(name="java:global/simple-stateful/Counter!org.superbiz.counter.Counter")
INFO - Jndi(name="java:global/simple-stateful/Counter")
INFO -
Jndi(name="java:global/EjbModule309142400/org.superbiz.counter.CounterTest!org.superbi
z.counter.CounterTest")
INFO - Jndi(name="java:global/EjbModule309142400/org.superbiz.counter.CounterTest")
INFO - Created Ejb(deployment-id=Counter, ejb-name=Counter, container=Default Stateful
Container)
INFO - Created Ejb(deployment-id=org.superbiz.counter.CounterTest, ejb-
name=org.superbiz.counter.CounterTest, container=Default Managed Container)
INFO - Started Ejb(deployment-id=Counter, ejb-name=Counter, container=Default Stateful
Container)
INFO - Started Ejb(deployment-id=org.superbiz.counter.CounterTest, ejb-
name=org.superbiz.counter.CounterTest, container=Default Managed Container)
INFO - Deployed Application(path=/Users/dblevins/examples/simple-stateful)
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.098 sec
```

Results :

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