Accessing an enterprise bean

Michael Dallago

October 17, 2016

Chapter 1

Introduction

This project consists of an **enterprise bean** which exposes methods to get the current time and date and a client which connects to the bean.

The enterprise bean is deployed on the *Wildfly* server.

Chapter 2

Implementation

2.1 Bean

The following box shows the source code for the Bean, called `DateBean` since it is used to get the date.

```
 1 package server;
 2
 3 import java.util.Date;
 4 import javax.ejb.Stateless;
 5
 6 @Stateless
 7 public class DateBean implements DateBeanRemote
 8 {
 9   @Override
10   public String getTimeAndDate()
11   {
12     Date date = new Date();
13     String str = String.format("%1$tT:%1$tL, %1$td %1$tB %1$tY", date);
14     return str;
15   }
16 }
```

The `Bean` is inside package `server` and, since we use Java-Data utilities, we import the `java.util.Date` package. This `bean` is `Stateless`, that is, it has no member variable and can be destroyed as soon as it is not used anymore, so we import the `javax.ejb.Stateless`.

The method used to return the date to the client is straightforward. We first get the date on line 12, then we construct a `String` formatted according to how we want to display the date and the time. In fact, we specify we want the time of the day, followed by a comma, followed b y the number of the day of the month, the name of the month and finally the year.

So, for example, we will get a `String` such as the following

```
23:01:43:194, 16 October 2016
```

This `String` is the return value of the method, that is, it is the value the client gets when it calls this method.

2.2 Client

The `Client` consists of two packages:

- `client` package contains the client code, that is, the Java class `Client.java` code.

- `server` package contains the remote bean code, that is, the code of the interface exposing the bean method code.

```
 1 package client;
 2 
 3 import java.util.Properties;
 4 import javax.naming.Context;
 5 import javax.naming.InitialContext;
 6 import javax.naming.NamingException;
 7 import server.DateBeanRemote;
 8 
 9 public class Client
10 {
11   public static void main(String[] args)
12   {
13     Properties jndiProps = new Properties();
14     jndiProps.put(Context.INITIAL_CONTEXT_FACTORY,
15                   "org.jboss.naming.remote.client.InitialContextFactory");
16     jndiProps.put(Context.PROVIDER_URL, "http-remoting://127.0.0.1:8080");
17     jndiProps.put(Context.SECURITY_PRINCIPAL, "user");
18     jndiProps.put(Context.SECURITY_CREDENTIALS, "password");
19     jndiProps.put("jboss.naming.client.ejb.context", true);
20 
21     try
22     {
23       InitialContext context = new InitialContext(jndiProps);
24       DateBeanRemote myDateBeanRemote = (DateBeanRemote) context.lookup(
25 "/Server/Server-ejb/DateBean!server.DateBeanRemote");
26       System.out.println(myDateBeanRemote.getTimeAndDate());
27       System.out.println(myDateBeanRemote.getTimeAndDate());
28     }
29     catch (NamingException ex)
30     {
31       System.out.println(ex.getMessage());
32     }
33   }
34 }
```

As we can see in the above box, the `Main-Class` for the client consists of a single method, the `main()`.

In the `main()` we first initialize some properties, used to search and find the bean on the already-running Wildfly server.

In line 16, we specify the host on which the Wildfly server is running (in this case `localhost`) followed by the port number.

In line 17 and 18 respectively, we define the user name and the corresponding password in order to authenticate on the server.

In line 19 we specify an important property to set if we want to do EJB invocations via the remote-naming project. Since this property is set to true and passed to the `InitialContext` creation, the remote-naming project internally will do whatever is necessary to setup a `EJBClientContext`. The InitialContext creation done in line 23 via the remote-naming project has now internally triggered the creation of a `EJBClientContext` containing a EJBReceiver capable of handling the EJB invocations.

After the creation of the `InitialContext`, we look up for the `Bean` in the context. The specified name matches the one returned by the server when the bean is deployed.

On lines 26 and 27, we simply call the method: since it returns a `String`, we can print it through the method `System.out.println()`.

The following is the remote interface code placed inside the `server` package but on the client side.

```java
package server;

import javax.ejb.Remote;

@Remote
public interface DateBeanRemote
{
  String getTimeAndDate();
}
```

As you can see, it is a Plain Old Java Interface exposing the name and the return value for the `getTimeAndDate()` server method.

Chapter 3

Deployment

3.1 Compile and run on Netbeans

Unzip the source code and open Netbeans. Inside the latter, click `Open Project...` and open both `Client` and `Server` projects.

Double click on the `Server` package (the one with a purple triangle on its left), expand `Java EE Modules` and double click on `Server-ejb.jar`. An EJB module project will be opened as well.

At this time you have all the source code opened in NetBeans, and you can modify it if you want to.

In order to compile the sources:

1. Right click on `Server-ejb` and then click `Clean and build`.

2. Now go to the `Server` project, right click on it, click `Clean and build` and then `Run`. This will compile the project, generate the `.ear` archive, start the WildFly server and deploy it.

3. Right click on the `Client`, cliuck `Properties`, `Libraries` and add both the `Java EE 7 API Library` and the `jboss-client.jar` archive. Then `Clean and build` it and `Run` it.

Note that libraries should already have been included in the project and NetBeans should automatically detect them.

You should get output similar to the following.

```
...
23:43:32,419 INFO  [org.jboss.weld.deployer] (MSC service thread 1-5)
    WFLYWELD0003: Processing weld deployment Server.ear
23:43:32,483 INFO  [org.hibernate.validator.internal.util.Version] (MSC
    service thread 1-5) HV000001: Hibernate Validator 5.2.4.Final
23:43:32,564 INFO  [org.jboss.weld.deployer] (MSC service thread 1-4)
    WFLYWELD0003: Processing weld deployment Server-ejb.jar
23:43:32,567 INFO  [org.jboss.as.ejb3.deployment] (MSC service thread
    1-4) WFLYEJB0473: JNDI bindings for session bean named 'DateBean' in
    deployment unit 'subdeployment "Server-ejb.jar" of deployment
    "Server.ear"' are as follows:

     java:global/Server/Server-ejb/DateBean!server.DateBeanRemote
     java:app/Server-ejb/DateBean!server.DateBeanRemote
     java:module/DateBean!server.DateBeanRemote
     java:jboss/exported/Server/Server-ejb/DateBean!server.DateBeanRemote
     java:global/Server/Server-ejb/DateBean
     java:app/Server-ejb/DateBean
     java:module/DateBean

```

```
15 23:43:32,612 INFO  [org.jboss.weld.Version] (MSC service thread 1-3)
     WELD-000900: 2.3.5 (Final)
16 ...
```

Note that line 10 is partially used to find the bean in the client code.

When running the client, you should get something like the following:

```
 1 Oct 16, 2016 11:48:04 PM org.xnio.Xnio <clinit>
 2 INFO: XNIO version 3.4.0.Final
 3 Oct 16, 2016 11:48:04 PM org.xnio.nio.NioXnio <clinit>
 4 INFO: XNIO NIO Implementation Version 3.4.0.Final
 5 Oct 16, 2016 11:48:04 PM org.jboss.remoting3.EndpointImpl <clinit>
 6 INFO: JBoss Remoting version 4.0.21.Final
 7 Oct 16, 2016 11:48:05 PM org.jboss.ejb.client.remoting.VersionReceiver
     handleMessage
 8 INFO: EJBCLIENT000017: Received server version 2 and marshalling
     strategies [river]
 9 Oct 16, 2016 11:48:05 PM
     org.jboss.ejb.client.remoting.RemotingConnectionEJBReceiver associate
10 INFO: EJBCLIENT000013: Successful version handshake completed for
     receiver context
     EJBReceiverContext{clientContext=org.jboss.ejb.client.EJBClientContext@1e88b3c,
     receiver=Remoting connection EJB receiver [connection=Remoting
     connection <3adb73e3> on endpoint
     "config-based-naming-client-endpoint"
     <42d80b78>,channel=jboss.ejb,nodename=localhost]} on channel Channel
     ID 8df8f850 (outbound) of Remoting connection 589838eb to
     /127.0.0.1:8080 of endpoint "config-based-naming-client-endpoint"
     <42d80b78>
11 Oct 16, 2016 11:48:05 PM org.jboss.ejb.client.EJBClient <clinit>
12 INFO: JBoss EJB Client version 2.1.4.Final
13 23:48:05:294, 16 October 2016
14 23:48:05:298, 16 October 2016
```

The most interesting lines are 13 and 14, since there the date is printed.

Note that you need to add a `user` to your WildFly server, via the `adduser` utility you can find in the `$JBOSS_HOME/bin` directory.