Università degli Studi di Trento

Web Architectures

Application front end on top of an EJB layer

Gianvito Taneburgo

December 2, 2016

1 Introduction

The following report describes the procedure that has been followed to develop a front end for the client-server application described in the 4th assignment, *Accessing a DB through an EJB layer*. The web application running on Apache Tomcat provides a simple interface for clients and administrators to interact with the application back end. All the previous command line operations are now executed by browsing through the pages and interacting with them.

Some minor changes have been applied to the application back end. The differences between the current version and the original one are discussed in the Implementation paragraph, together with some details about the front end.

On the contrary, the Deployment paragraph provides details on how to configure and run Wildfly and Tomcat, how to compile and deploy the server EAR containing the EJBs on Wildfly and, finally, how to deploy the web application on Tomcat.

Java EE specifications have been modified over the years and the process to write and deploy client-server applications relying on EJBs has changed as well. For the remainder of this report the following versions of specifications, applications and frameworks will be considered:

- Java Platform, Standard Edition: *8*
- Java Platform, Enterprise Edition: *7*
- Enterprise JavaBean: *3.2*
- Java Servlet: *3.1*

- Java Persistence API: *2.1*
- Wildfly: *10.1.0.Final*
- Apache Tomcat: *8.0.39*
- Hibernate: *5.2.3.Final*
- Apache Derby: *10.12.1.1*
- Apache Maven: *3.3.9*

Further details can be found inside Maven *pom.xml* files.

2 Implementation

The implementation details of the application back end running on Wilfdly are not going to be repeated in this paragraph. On the contrary, only the differences with the new version are going to be highlighted.

The previous server version provided both stateful and stateless EJBs, with the formers required to keep track of the user interaction in the bookshop. In the current version stateful EJBs have been replaced by stateless ones, since the storage of user information is now performed in HTTP session objects. While clients interact with the web application, in fact, the HTTP session gets filled with all the data previously stored by the Widlfy server. Data is later sent, if required, to the back end for every request.

For such reason, the current version of the application only provides one stateless EJB. This changing is also reflected in the *common* package, that now contains only one remote interface. The other layers of the application (e.g. persistence) are unchanged.

The web application running on Tomcat takes advantage of both Servlets and JSP. Servlets are used to implement authentication filters (for administrators and clients) and more complex tasks. JSP pages, on the contrary, are used to accomplish simpler tasks, usually related to the presentation layer, such as hiding/showing page components. Other static web pages, written in simple HTML, complete the application.

A *SessionManager* class is responsible of storing and retrieving items from the HTTP session object. This class is uniformly used both by the JSP and by the Servlets. Their code is really straightforward and usually involves some kind of controls on session values before showing items in the UI or sending requests to the back end. When some conditions are not satisfied the user is typically redirected to login pages in order to authenticate himself.

All the application requests are executed by connecting to the EJB via JNDI. Its parameters are defined in the *jboss-ejb-client.properties* file and are very similar to those of the desktop client. Some subtle changes were required to make Tomcat able to communicate with Widlfy.

Further implementation details can be found in the report *Accessing a DB through an EJB layer.*

3 Deployment

The *src.zip* file contains some configuration files for Widlfy and Tomcat, the Apache Derby driver jar, a library required for the application and, finally, the source code of both the web application and the server. The Apache Maven framework is required to easily compile and package the application WAR for Tomcat and the server EAR for Wildfly, thus it has to be properly installed on the system.
Download the *src.zip* file, the Wildfly binary tarball [1] and the Apache Tomcat binary tarball [2] into a folder. All the following commands have to be executed from the download directory.
To run the Wildfly application server with the back end please refer to the detailed instructions in the *Accessing a DB through an EJB layer* report. Logs will not be reported for this part and the commands will only be listed below:

```
$ unzip src.zip
$ tar xzvf wildfly-10.1.0.Final.tar.gz
$ export JBOSS_HOME=./wildfly-10.1.0.Final
$ cp mgmt-* $JBOSS_HOME/standalone/configuration
$ mv derby/ $JBOSS_HOME/modules/system/layers/base/org/apache
$ cp standalone.xml $JBOSS_HOME/standalone/configuration
$ $JBOSS_HOME/bin/standalone.sh
```

WildFly is now running. To compile and deploy the application open a new terminal and run the following commands:

```
$ export JBOSS_HOME=./wildfly-10.1.0.Final
$ mvn -f server/ clean install package
$ cp server/server-ear/target/server-ear.ear
  $JBOSS_HOME/standalone/deployments
```

The new EAR will be detected by Wildfly and deployed. It is time to configure and run the web application on Tomcat. In the same terminal extract the application server binaries and configure the environment variables by running:

```
$ tar xzvf apache-tomcat-8.0.39.tar.gz
$ export CATALINA_HOME=./apache-tomcat-8.0.39
$ export TOMCAT_HOME=./apache-tomcat-8.0.39
```

Since Widlfy is running on the port 8080, it is necessary to bind Tomcat on a different one, for instance 8081. The *server.xml* file contains a proper configuration and can be easily used for Tomcat by executing:

```
$ cp server.xml apache-tomcat-8.0.39/conf/
```

Tomcat default web application must be removed with the command:

```
$ rm -r apache-tomcat-8.0.39/webapps/ROOT/
```

[1] Wildfly 10.1.0.Final - Java EE7 Full and Web Distribution, 2016-08-19

[2] Apache Tomcat 8.0.39 - Core

To compile the web application it is necessary to install on the local Maven repository the Wildfly client library by executing:

```
$ mvn install:install-file -Dfile=./jboss-client.jar -DgroupId=jboss
  -DartifactId=ejb-client -Dversion=1.0 -Dpackaging=jar
```

The output should be similar to the following:

```
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building Maven Stub Project (No POM) 1
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-install-plugin:2.4:install-file (default-cli) @ standalone-pom ---
[INFO] Installing /home/[...]/.m2/repository/jboss/ejb-client/1.0/ejb-client-1.0.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 0.743 s
[INFO] Finished at: 2016-12-02T18:48:45+01:00
[INFO] Final Memory: 7M/119M
[INFO] ------------------------------------------------------------------------
```

The application WAR can now be packaged with Maven by executing:

```
$ mvn -f app/ clean install package
```

To deploy and run the application on Tomcat just run:

```
$ cp app/target/ROOT.war apache-tomcat-8.0.39/webapps/
$ $TOMCAT_HOME/bin/catalina.sh run
```

Tomcat will output something similar to this:

```
Using CATALINA_BASE:   ./apache-tomcat-8.0.39
Using CATALINA_HOME:   ./apache-tomcat-8.0.39
Using CATALINA_TMPDIR: ./apache-tomcat-8.0.39/temp
Using JRE_HOME:        /usr/lib/jvm/java-8-openjdk
[...]VersionLoggerListener.log Server version:        Apache Tomcat/8.0.39
[...]VersionLoggerListener.log Server built:          Nov 9 2016 08:48:39 UTC
[...]VersionLoggerListener.log Server number:         8.0.39.0
[...]VersionLoggerListener.log OS Name:               Linux
[...]VersionLoggerListener.log OS Version:            4.8.11-1-ARCH
[...]VersionLoggerListener.log Architecture:          amd64
[...]VersionLoggerListener.log Java Home:             /usr/lib/jvm/java-8-openjdk/jre
[...]VersionLoggerListener.log JVM Version:           1.8.0_112-b15
[...]VersionLoggerListener.log JVM Vendor:            Oracle Corporation
[...]VersionLoggerListener.log CATALINA_BASE:         /home/[...]/apache-tomcat-8.0.39
[...]VersionLoggerListener.log CATALINA_HOME:         /home/[...]/apache-tomcat-8.0.39
[...]INFO [main] org.apache.catalina.core.StandardEngine.startInternal Starting Servlet
  Engine: Apache Tomcat/8.0.39
[...]INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployWAR
  Deploying web application archive /home/[...]/apache-tomcat-8.0.39/webapps/ROOT.war
```

```
[...]HostConfig.deployWAR Deployment of web application archive
  /home/[...]/apache-tomcat-8.0.39/webapps/ROOT.war has finished in 2,559 ms
[...]
```

The bookshop application can be reached at the address *http://localhost:8081*.
It is possible to create users, add books to the catalog, buy them, and so on. These operations will make Wildfly produce some logs similar to those of the previous version.