Università degli Studi di Trento

Web Architectures

Accessing a DB through an EJB layer

Gianvito Taneburgo

October 20, 2016

1 Introduction

The following report describes the procedure that has been followed to develop a client-server application using Enterprise JavaBeans (EJB) and Hibernate to store data on an Apache Derby database via Java Persistence API (JPA). In particular, the program shows how different clients can contact and use EJBs deployed and running inside the EJB container of a Wildlfy application server. For the purpose of this demonstration, a rudimentary online bookshop has been developed. The server provides some functionalities to register users, add books to the catalogue and support the shopping experience of customers with online carts. This business logic is encoded into some EJBs and all the objects are encapsulated into Java EE Entities and persisted via Java DataBase Connectivity (JDBC) on a the database. Two command-line clients are also distributed, one is designed for customers, the other for application administrators.
The Implementation paragraph provides details both on the server and on the clients code. Some details about Java EE, EJB, JPA and JDBC have been omitted: the reader is supposed to know the benefits of the technologies, the most import parts of the specifications and the basic mechanisms that make them possible.
The Explanation paragraph, on the contrary, describes the server EJBs. Additional information about the persistency layer is also given here.
Finally, the Deployment paragraph, provides details on how to configure and run a Wildfly application server, how to configure the database, how to compile and deploy the server EAR containing the EJBs and, finally, how to use it from the two client applications.
Java EE, EJB and JPA 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 Persistence API: *2.1*
- Wildfly: *10.1.0.Final*
- 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 application code is organised into three packages, one for the clients, one for the server and the last for common classes. An overview of how classes and interfaces are subdivided follows.
The *client* package contains two main classes: *AdminClient* and *UserClient*. These classes are supposed to be run from command-line by, respectively, application administrators and customers. They exhibit a shell-like behaviour, allowing users to input as many commands as they want among those listed. The logic in common between the two clients has been moved into the abstract class *ApplicationClient* they both extend. A group of classes and interfaces (*OperationI*, *AdminOperation*, *ClientOperation*, *Command*) constitutes a wrapper for the commands inputted by clients. These classes are flexible and easily extendible to support new operations. Finally, the *Constants* class holds the default values for some clients parameters.
The *common* package contains three classes required to transfer POJOs between clients and server (*BookTO*, *CartTO*, *OperationTO*) and two remote interfaces of EJBs (*CartBeanI*, *ShopBeanI*). *CartBeanI* exposes all the server methods related to users carts (*getCart*, *addToCart*, *buy*, *leave*), while *ShopBeanI* all the other server functionalities (*addBook*, *listBooks*, *listOperations*, *registerUser*).
Finally, in the *server* package there are three classes handled by Hibernate to persist data on the database (*Book*, *Operation*, *User*), two EJBs implementing the interface just described (*ShopBean*, *CartBean*) and the *Constants* class for some parameters.
Most of the methods implementations, as well as the business logic of the application, are pretty straightforward. The main design choices to be discussed involve EJBs and Entities and are examined in depth in the following paragraph. Nevertheless, some details on the clients are worth noting. Clients communicate with the server by using JNDI,

whose parameters are provided with the *jndi.properties* configuration file [1] . It contains the following key-value pairs:

- *java.naming.factory.initial=org.jboss.naming.remote.client.InitialContextFactory*
  the Context [2] class to be used by the JNDI;
- *java.naming.provider.url=http-remoting://127.0.0.1:8080*
  the address of the EJB directory service;
- *java.naming.factory.url.pkgs=org.jboss.ejb.client.naming*
  the type of the naming service;
- *jboss.naming.client.ejb.context=true*
  a flag to be used when dealing with EJB;
- *java.naming.security.principal=myuser*
  the username of an account authorized on the server to interact with the EJBs;
- *java.naming.security.credentials=mypass*
  the password of the account.

In addition, the key bound to the remote EJBs are supplied with command line arguments. If not, the default values are used:

- *server-ear/server-ejb/CartBean!bookshop.common.CartBeanI*
- *server-ear/server-ejb/ShopBean!bookshop.common.ShopBeanI*

AdminClient and *UserClient main* methods can be run once the EJBs are deployed inside the container of the application server.

3 Explanation

Application data is persisted on Derby with Hibernate. A *persistence.xml* file provides the information required by the JPA specification, such as the name of the persistence unit, the database connection URL, the driver class and the credentials. Further details related to tables schemas are expressed with Java annotations on the Entities. *Book*, *Operation* and *User* classes are annotated with *@Entity* and with *@Table(name="table_name")*. Each attribute specifies its own table column with the *@Column(name="col_name")* annotation. Furthermore, one attribute per class is also tagged with *@Id* to define the table primary key. The annotation *@GeneratedValue(strategy=GenerationType.AUTO)* is sometimes used to let Hibernate generate ids automatically.
The server functionalities have been split into two EJBs: *CartBean* and *ShopBean* (annotated, respectively, with *@Remote(CartBeanI.class)* and *@Remote(ShopBeanI.class)* accordingly to the EJB 3.2 specifications). *CartBean* is a stateful bean since it has to

[1] http://docs.oracle.com/javase/jndi/tutorial/beyond/env/source.html
[2] https://docs.oracle.com/javase/8/docs/api/javax/naming/Context.html

keep session information about the user cart. Each user will thus connect to one EJB that will also store in its own state the username, required by the *buy* method.
On the contrary, *ShopBean* is a stateless bean, since it only has to store and load data from the database and return them immediately to clients.
The EJBs have an attribute of class *EntityManager*, annotated with *@PersistenceContext(unitName="BookshopPU")*, to interact with the database via JPA.

4 Deployment

The *src.zip* file contains three configuration files for Widlfy, the Apache Derby driver jar and the source code of both the clients and the server. The Apache Maven framework is required to easily compile and package the client JAR applications and the EAR to be deployed on the application server, thus it has to be properly installed on the system.
Download the *src.zip* file and the Wildfly binary tarball [3] into a folder. All the following commands have to be executed from the download directory.
To unzip both the archives just type:

```
$ unzip src; tar -xvzf wildfly-10.1.0.Final.tar.gz
```

To temporarily set the shell variable *JBOSS_HOME*, required by Wildfly, type:

```
$ export JBOSS_HOME=./wildfly-10.1.0.Final
```

An authorized user is required to invoke the EJBs. It can be created by running the Wildfly *add-user.sh* script. In order to simplify the configuration process, some files including a test user have already been provided. It is sufficient to copy them inside the Wildfly folder with the command:

```
$ cp mgmt-* $JBOSS_HOME/standalone/configuration
```

To install Apache Derby on Wildfly copy the driver jar with the command:

```
$ mv derby/ $JBOSS_HOME/modules/system/layers/base/org/apache/
```

To make Widlfy aware of the new database a new *datasource* has to be added into the deployment descriptor file. To simplify the process the new *standalone.xml* file is already provided and can be easily copied with:

```
$ cp standalone.xml $JBOSS_HOME/standalone/configuration
```

The file contains a tag with JDBC information, credentials and pool parameters:

```
<?xml version="1.0" encoding="UTF-8"?>
<datasource jndi-name="java:/DerbyDS" pool-name="DerbyDS" enabled="true" use-ccm="false">
    <connection-url>jdbc:derby://localhost:1527/jhDB;create=true</connection-url>
    <driver>org.apache.derby</driver>
    <pool>
        <min-pool-size>1</min-pool-size>
        <max-pool-size>5</max-pool-size>
```

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

```
        <prefill>true</prefill>
    </pool>
    <security>
        <user-name>myuser</user-name>
        <password>mypass</password>
    </security>
    <validation>
        <validate-on-match>false</validate-on-match>
        <background-validation>false</background-validation>
    </validation>
    <statement>
        <share-prepared-statements>false</share-prepared-statements>
    </statement>
</datasource>
```

Wildlfy is now properly configured and can be executed by typing:

```
$ $JBOSS_HOME/bin/standalone.sh
```

If everything goes fine the console should print something similar to this (some lines have been omitted for clarity):

```
=========================================================================

  JBoss Bootstrap Environment

  JBOSS_HOME: ./wildfly-10.1.0.Final

  JAVA: /usr/lib/jvm/java-8-openjdk/bin/java

  JAVA_OPTS:  -server -Xms64m -Xmx512m -XX:MetaspaceSize=96M -XX:MaxMetaspaceSize=256m
    -Djava.net.preferIPv4Stack=true -Djboss.modules.system.pkgs=org.jboss.byteman
    -Djava.awt.headless=true

=========================================================================

11:18:16,560 INFO  [org.jboss.modules] (main) JBoss Modules version 1.5.2.Final
11:18:16,965 INFO  [org.jboss.msc] (main) JBoss MSC version 1.2.6.Final
11:18:17,038 INFO  [org.jboss.as] (MSC service thread 1-6) WFLYSRV0049:
  WildFly Full 10.1.0.Final (WildFly Core 2.2.0.Final) starting
11:18:18,283 INFO  [org.jboss.as.server] (Controller Boot Thread) WFLYSRV0039:
  Creating http management service using socket-binding (management-http)
11:18:18,311 INFO  [org.xnio] (MSC service thread 1-1) XNIO version 3.4.0.Final
11:18:18,326 INFO  [org.xnio.nio] (MSC service thread 1-1) XNIO NIO Implementation
  Version 3.4.0.Final
...
11:18:18,533 INFO  [org.jboss.remoting] (MSC service thread 1-8) JBoss Remoting
  version 4.0.21.Final
...
11:18:18,555 INFO  [org.jboss.as.naming] (MSC service thread 1-7) WFLYNAM0003:
  Starting Naming Service
11:18:18,557 INFO  [org.jboss.as.mail.extension] (MSC service thread 1-5) WFLYMAIL0001:
  Bound mail session [java:jboss/mail/Default]
...
```

```
11:18:18,623 INFO  [org.jboss.as.connector.deployers.jdbc] (MSC service thread 1-1)
  WFLYJCA0018: Started Driver service with driver-name = org.apache.derby
11:18:18,635 INFO  [org.jboss.as.connector.subsystems.datasources] (ServerService
  Thread Pool -- 33) WFLYJCA0004: Deploying JDBC-compliant driver class org.h2.Driver
  (version 1.3)
11:18:18,642 INFO  [org.jboss.as.connector.deployers.jdbc] (MSC service thread 1-7)
  WFLYJCA0018: Started Driver service with driver-name = h2
...
11:18:18,816 INFO  [org.wildfly.extension.undertow] (MSC service thread 1-5) WFLYUT0012:
  Started server default-server.
11:18:18,819 INFO  [org.wildfly.extension.undertow] (MSC service thread 1-7) WFLYUT0018:
  Host default-host starting
11:18:18,907 INFO  [org.jboss.as.ejb3] (MSC service thread 1-1) WFLYEJB0482:
  Strict pool mdb-strict-max-pool is using a max instance size of 16 (per class),
  which is derived from the number of CPUs on this host.
11:18:18,911 INFO  [org.jboss.as.ejb3] (MSC service thread 1-2) WFLYEJB0481:
  Strict pool slsb-strict-max-pool is using a max instance size of 64 (per class),
  which is derived from thread worker pool sizing.
11:18:18,974 INFO  [org.wildfly.extension.undertow] (MSC service thread 1-4) WFLYUT0006:
  Undertow HTTP listener default listening on 127.0.0.1:8080
11:18:19,234 WARN  [org.jboss.as.domain.management.security] (MSC service thread 1-1)
  WFLYDM0111: Keystore /home/gianvito/Desktop/./wildfly-10.1.0.Final/standalone/
  configuration/application.keystore not found, it will be auto generated on first use
  with a self signed certificate for host localhost
11:18:19,414 INFO  [org.jboss.as.server.deployment.scanner] (MSC service thread 1-5)
  WFLYDS0013: Started FileSystemDeploymentService for directory /home/gianvito/Desktop/
  ./wildfly-10.1.0.Final/standalone/deployments
11:18:19,564 INFO  [org.wildfly.extension.undertow] (MSC service thread 1-5) WFLYUT0006:
  Undertow HTTPS listener https listening on 127.0.0.1:8443
...
11:18:19,755 INFO  [org.jboss.ws.common.management] (MSC service thread 1-4) JBWS022052:
  Starting JBossWS 5.1.5.Final (Apache CXF 3.1.6)
11:18:19,757 INFO  [org.jboss.as.connector.subsystems.datasources] (MSC service thread
  1-5) WFLYJCA0001: Bound data source [java:/DerbyDS]
...
11:18:19,928 INFO  [org.jboss.as] (Controller Boot Thread) WFLYSRV0060: Http management
  interface listening on http://127.0.0.1:9990/management
11:18:19,929 INFO  [org.jboss.as] (Controller Boot Thread) WFLYSRV0051: Admin console
  listening on http://127.0.0.1:9990
11:18:19,929 INFO  [org.jboss.as] (Controller Boot Thread) WFLYSRV0025: WildFly Full
  10.1.0.Final (WildFly Core 2.2.0.Final) started in 3659ms - Started 338 of 584
  services (394 services are lazy, passive or on-demand)
```

The output shows the URLs of the running services, the versions of the main technologies in use, Apache Derby database information and the deployment folder of the application server. Wildfly will now keep running in the console, so another terminal has to been opened in the same working directory to run new commands. The shell variable *JBOSS_HOME* has to be set again:

```
$ export JBOSS_HOME=./wildfly-10.1.0.Final
```

To compile and package the server EAR containing the EJB just type:

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

Maven will download the required dependencies, will compile the code and will build the EAR automatically by using some plugins.
In a similar way, to compile and package the JAR containing the clients application just type:

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

By default, Wildfly automatically deploys new EARs present in a specific folder (the full path can be read in the previous console log). So, to deploy the application, just copy the EAR built by Maven with the command:

```
$ cp server/server-ear/target/server-ear.ear $JBOSS_HOME/
standalone/deployments
```

The previous terminal, with Wildfly still running, will automatically output:

```
11:19:39,470 INFO  [org.jboss.as.repository] (DeploymentScanner-threads - 2) WFLYDR0001:
  Content added at location /home/gianvito/Desktop/./wildfly-10.1.0.Final/
  standalone/data/content/f5/609e897ad02ab34dfd62c67ba96ded1fb844b7/content
11:19:39,502 INFO  [org.jboss.as.server.deployment] (MSC service thread 1-5)
  WFLYSRV0027: Starting deployment of "server-ear.ear" (runtime-name: "server-ear.ear")
...
11:19:39,697 INFO  [org.jboss.as.server.deployment] (MSC service thread 1-5)
  WFLYSRV0207: Starting subdeployment (runtime-name: "server-ejb.jar")
11:19:39,797 INFO  [org.jboss.as.jpa] (MSC service thread 1-6) WFLYJPA0002: Read
  persistence.xml for BookshopPU
11:19:39,916 INFO  [org.jboss.weld.deployer] (MSC service thread 1-3) WFLYWELD0003:
  Processing weld deployment server-ear.ear
11:19:39,929 INFO  [org.jboss.as.jpa] (ServerService Thread Pool -- 66) WFLYJPA0010:
  Starting Persistence Unit (phase 1 of 2) Service 'server-ear.ear/server-ejb.jar
  #BookshopPU'
11:19:39,969 INFO  [org.hibernate.jpa.internal.util.LogHelper] (ServerService Thread
  Pool -- 66) HHH000204: Processing PersistenceUnitInfo [
        name: BookshopPU
        ...]
11:19:40,039 INFO  [org.hibernate.validator.internal.util.Version] (MSC service thread
  1-3) HV000001: Hibernate Validator 5.2.4.Final
11:19:40,090 INFO  [org.hibernate.Version] (ServerService Thread Pool -- 66) HHH000412:
  Hibernate Core {5.0.10.Final}
11:19:40,095 INFO  [org.hibernate.cfg.Environment] (ServerService Thread Pool -- 66)
  HHH000206: hibernate.properties not found
11:19:40,097 INFO  [org.hibernate.cfg.Environment] (ServerService Thread Pool -- 66)
  HHH000021: Bytecode provider name : javassist
11:19:40,168 INFO  [org.hibernate.annotations.common.Version] (ServerService Thread Pool
  -- 66) HCANN000001: Hibernate Commons Annotations {5.0.1.Final}
11:19:40,269 INFO  [org.jboss.weld.deployer] (MSC service thread 1-5) WFLYWELD0003:
  Processing weld deployment server-ejb.jar
11:19:40,280 INFO  [org.jboss.as.ejb3.deployment] (MSC service thread 1-5) WFLYEJB0473:
  JNDI bindings for session bean named 'CartBean' in deployment unit 'subdeployment
  "server-ejb.jar" of deployment "server-ear.ear"' are as follows:

        java:global/server-ear/server-ejb/CartBean!bookshop.common.CartBeanI
        java:app/server-ejb/CartBean!bookshop.common.CartBeanI
        java:module/CartBean!bookshop.common.CartBeanI
```

```
        java:jboss/exported/server-ear/server-ejb/CartBean!bookshop.common.CartBeanI
        java:global/server-ear/server-ejb/CartBean
        java:app/server-ejb/CartBean
        java:module/CartBean

11:19:40,281 INFO  [org.jboss.as.ejb3.deployment] (MSC service thread 1-5) WFLYEJB0473:
  JNDI bindings for session bean named 'ShopBean' in deployment unit 'subdeployment
  "server-ejb.jar" of deployment "server-ear.ear"' are as follows:

        java:global/server-ear/server-ejb/ShopBean!bookshop.common.ShopBeanI
        java:app/server-ejb/ShopBean!bookshop.common.ShopBeanI
        java:module/ShopBean!bookshop.common.ShopBeanI
        java:jboss/exported/server-ear/server-ejb/ShopBean!bookshop.common.ShopBeanI
        java:global/server-ear/server-ejb/ShopBean
        java:app/server-ejb/ShopBean
        java:module/ShopBean

11:19:40,362 INFO  [org.jboss.weld.Version] (MSC service thread 1-8) WELD-000900: 2.3.5
  (Final)
11:19:40,700 INFO  [org.infinispan.configuration.cache.EvictionConfigurationBuilder]
  (ServerService Thread Pool -- 66) ISPN000152: Passivation configured without an
  eviction policy being selected. Only manually evicted entities will be passivated.
11:19:40,701 INFO  [org.infinispan.configuration.cache.EvictionConfigurationBuilder]
  (ServerService Thread Pool -- 66) ISPN000152: Passivation configured without an
  eviction policy being selected. Only manually evicted entities will be passivated.
11:19:40,787 INFO  [org.jboss.as.jpa] (ServerService Thread Pool -- 67) WFLYJPA0010:
  Starting Persistence Unit (phase 2 of 2) Service 'server-ear.ear/server-ejb.jar
  #BookshopPU'
11:19:41,071 INFO  [org.hibernate.dialect.Dialect] (ServerService Thread Pool -- 67)
  HHH000400: Using dialect: org.hibernate.dialect.H2Dialect
11:19:41,079 WARN  [org.hibernate.dialect.H2Dialect] (ServerService Thread Pool -- 67)
  HHH000431: Unable to determine H2 database version, certain features may not work
11:19:41,105 INFO  [org.jboss.as.clustering.infinispan] (ServerService Thread Pool --
  66) WFLYCLINF0002: Started client-mappings cache from ejb container
11:19:41,165 INFO  [org.hibernate.envers.boot.internal.EnversServiceImpl] (ServerService
  Thread Pool -- 67) Envers integration enabled? : true
11:19:41,627 INFO  [...] (ServerService) Hibernate:
11:19:41,627 INFO  [...] (ServerService)     drop table BOOKS if exists
11:19:41,628 INFO  [...] (ServerService) Hibernate:
11:19:41,628 INFO  [...] (ServerService)     drop table OPERATIONS if exists
11:19:41,628 INFO  [...] (ServerService) Hibernate:
11:19:41,629 INFO  [...] (ServerService)     drop table USERS if exists
11:19:41,629 INFO  [...] (ServerService) Hibernate:
11:19:41,629 INFO  [...] (ServerService)     drop sequence if exists hibernate_sequence
11:19:41,630 INFO  [...] (ServerService) Hibernate:
11:19:41,630 INFO  [...] (ServerService)     create sequence hibernate_sequence start
                                                    with 1 increment by 1
11:19:41,631 INFO  [...] (ServerService) Hibernate:
11:19:41,632 INFO  [...] (ServerService)     create table BOOKS (
11:19:41,632 INFO  [...] (ServerService)         id integer not null,
11:19:41,632 INFO  [...] (ServerService)         price integer,
11:19:41,632 INFO  [...] (ServerService)         title varchar(255),
11:19:41,632 INFO  [...] (ServerService)         primary key (id)
11:19:41,632 INFO  [...] (ServerService)     )
```

```
11:19:41,633 INFO  [...] (ServerService) Hibernate:
11:19:41,633 INFO  [...] (ServerService)     create table OPERATIONS (
11:19:41,633 INFO  [...] (ServerService)         id integer not null,
11:19:41,633 INFO  [...] (ServerService)         booksIds varchar(255),
11:19:41,634 INFO  [...] (ServerService)         username varchar(255),
11:19:41,634 INFO  [...] (ServerService)         primary key (id)
11:19:41,634 INFO  [...] (ServerService)     )
11:19:41,635 INFO  [...] (ServerService) Hibernate:
11:19:41,635 INFO  [...] (ServerService)     create table USERS (
11:19:41,635 INFO  [...] (ServerService)         username varchar(255) not null,
11:19:41,635 INFO  [...] (ServerService)         password varchar(255),
11:19:41,635 INFO  [...] (ServerService)         primary key (username)
11:19:41,635 INFO  [...] (ServerService)     )
11:19:41,636 INFO  [org.hibernate.tool.hbm2ddl.SchemaExport] (ServerService Thread Pool
  -- 67) HHH000230: Schema export complete
11:19:42,217 INFO  [org.jboss.as.server] (DeploymentScanner-threads - 2) WFLYSRV0010:
  Deployed "server-ear.ear" (runtime-name : "server-ear.ear")
11:19:51,188 INFO  [org.jboss.ejb.client] (pool-1-thread-1) JBoss EJB Client version
  2.1.4.Final
```

The output shows the EJB bindings required by the client JNDI and some Hibernate log produced during tables creation.
Clients can now be executed. To run the administration client execute:

```
$ java -jar client/target/AdminClient.jar
```

```
Trying to connect to EJB located at server-ear/server-ejb/ShopBean!bookshop.common.
  ShopBeanI

Oct 20, 2016 1:55:15 PM org.xnio.Xnio <clinit>
INFO: XNIO version 3.4.0.Final
Oct 20, 2016 1:55:15 PM org.xnio.nio.NioXnio <clinit>
INFO: XNIO NIO Implementation Version 3.4.0.Final
Oct 20, 2016 1:55:15 PM org.jboss.remoting3.EndpointImpl <clinit>
INFO: JBoss Remoting version 4.0.21.Final
Oct 20, 2016 1:55:15 PM org.jboss.ejb.client.remoting.VersionReceiver handleMessage
INFO: EJBCLIENT000017: Received server version 2 and marshalling strategies [river]
Oct 20, 2016 1:55:15 PM org.jboss.ejb.client.remoting.RemotingConnectionEJBReceiver
  associate
INFO: EJBCLIENT000013: Successful version handshake completed for receiver context
  EJBReceiverContext{clientContext=org.jboss.ejb.client.EJBClientContext@685cb137,
  receiver=Remoting connection EJB receiver [connection=Remoting connection <7b875d2c>
  on endpoint "config-based-naming-client-endpoint" <6a41eaa2>,channel=jboss.ejb,
  nodename=xps13]} on channel Channel ID de371119 (outbound) of Remoting connection
  1ce92674 to /127.0.0.1:8080 of endpoint "config-based-naming-client-endpoint" <6a41a2>
Oct 20, 2016 1:55:16 PM org.jboss.ejb.client.EJBClient <clinit>
INFO: JBoss EJB Client version 2.1.4.Final
Please specify one of the following operations:
addBook
listOperations
>
```

It is now possible to input some commands into the shell, for example:

```
> addBook DB_ADMIN_PASSWORD I_promessi_sposi 10
Book added.
> addBook DB_ADMIN_PASSWORD Il_nome_della_rosa 20
Book added.
```

The server will persist data and log:

```
14:03:04,502 INFO  [stdout] (default task-8) Hibernate:
14:03:04,502 INFO  [stdout] (default task-8)     call next value for hibernate_sequence
14:03:04,567 INFO  [stdout] (default task-8) Hibernate:
14:03:04,567 INFO  [stdout] (default task-8)     insert
14:03:04,567 INFO  [stdout] (default task-8)     into
14:03:04,567 INFO  [stdout] (default task-8)         BOOKS
14:03:04,567 INFO  [stdout] (default task-8)         (price, title, id)
14:03:04,567 INFO  [stdout] (default task-8)     values
14:03:04,567 INFO  [stdout] (default task-8)         (?, ?, ?)
14:03:12,654 INFO  [stdout] (default task-9) Hibernate:
14:03:12,654 INFO  [stdout] (default task-9)     call next value for hibernate_sequence
14:03:12,656 INFO  [stdout] (default task-9) Hibernate:
14:03:12,656 INFO  [stdout] (default task-9)     insert
14:03:12,656 INFO  [stdout] (default task-9)     into
14:03:12,656 INFO  [stdout] (default task-9)         BOOKS
14:03:12,656 INFO  [stdout] (default task-9)         (price, title, id)
14:03:12,656 INFO  [stdout] (default task-9)     values
14:03:12,656 INFO  [stdout] (default task-9)         (?, ?, ?)
```

Finally, run the user client:

```
$ java -jar client/target/UserClient.jar
```

```
Trying to connect to EJB located at server-ear/server-ejb/ShopBean!bookshop.common.
ShopBeanI

Oct 20, 2016 2:06:42 PM org.xnio.Xnio <clinit>
INFO: XNIO version 3.4.0.Final
Oct 20, 2016 2:06:42 PM org.xnio.nio.NioXnio <clinit>
INFO: XNIO NIO Implementation Version 3.4.0.Final
Oct 20, 2016 2:06:42 PM org.jboss.remoting3.EndpointImpl <clinit>
INFO: JBoss Remoting version 4.0.21.Final
Oct 20, 2016 2:06:42 PM org.jboss.ejb.client.remoting.VersionReceiver handleMessage
INFO: EJBCLIENT000017: Received server version 2 and marshalling strategies [river]
Oct 20, 2016 2:06:42 PM org.jboss.ejb.client.remoting.RemotingConnectionEJBReceiver
  associate
INFO: EJBCLIENT000013: Successful version handshake completed for receiver context
  EJBReceiverContext{clientContext=org.jboss.ejb.client.EJBClientContext@685cb137,
  receiver=Remoting connection EJB receiver [connection=Remoting connection <3f6da1ad>
  on endpoint "config-based-naming-client-endpoint" <6a41eaa2>,channel=jboss.ejb,
  nodename=xps13]} on channel Channel ID 837dcca2 (outbound) of Remoting connection
  1ce92674 to /127.0.0.1:8080 of endpoint "config-based-naming-client-endpoint" <6a412>
Oct 20, 2016 2:06:42 PM org.jboss.ejb.client.EJBClient <clinit>
```

```
INFO: JBoss EJB Client version 2.1.4.Final

Trying to connect to EJB located at server-ear/server-ejb/CartBean!bookshop.common.
CartBeanI

Oct 20, 2016 2:06:42 PM org.jboss.ejb.client.remoting.VersionReceiver handleMessage
INFO: EJBCLIENT000017: Received server version 2 and marshalling strategies [river]
Oct 20, 2016 2:06:42 PM org.jboss.ejb.client.remoting.RemotingConnectionEJBReceiver
  associate
INFO: EJBCLIENT000013: Successful version handshake completed for receiver context
  EJBReceiverContext{clientContext=org.jboss.ejb.client.EJBClientContext@5e853265,
  receiver=Remoting connection EJB receiver [connection=Remoting connection <43994371>
  on endpoint "config-based-naming-client-endpoint" <6a41eaa2>,channel=jboss.ejb,
  nodename=xps13]} on channel Channel ID e0573b03 (outbound) of Remoting connection
  5d76b067 to /127.0.0.1:8080 of endpoint "config-based-naming-client-endpoint" <6a412>
Please specify one of the following operations:
registerUser
getCart
listBooks
addToCart
buy
leave
>
```

It's time to go shopping:

```
> registerUser user pass
User registered.
> getCart user pass
> listBooks
Books available:
Book{id=1, title='I_promessi_sposi', price=10}
Book{id=2, title='Il_nome_della_rosa', price=20}
> addToCart 1
> buy
```

The server will persist data and log the operation:

```
14:03:12,654 INFO  [stdout] (default task-9) Hibernate:
14:03:12,654 INFO  [stdout] (default task-9)     call next value for hibernate_sequence
14:03:12,656 INFO  [stdout] (default task-9) Hibernate:
14:03:12,656 INFO  [stdout] (default task-9)     insert
14:03:12,656 INFO  [stdout] (default task-9)     into
14:03:12,656 INFO  [stdout] (default task-9)         BOOKS
14:03:12,656 INFO  [stdout] (default task-9)         (price, title, id)
14:03:12,656 INFO  [stdout] (default task-9)     values
14:03:12,656 INFO  [stdout] (default task-9)         (?, ?, ?)
14:10:19,841 INFO  [stdout] (default task-27) Hibernate:
14:10:19,842 INFO  [stdout] (default task-27)     insert
14:10:19,842 INFO  [stdout] (default task-27)     into
14:10:19,842 INFO  [stdout] (default task-27)         USERS
14:10:19,842 INFO  [stdout] (default task-27)         (password, username)
```

```
14:10:19,842 INFO  [stdout] (default task-27)     values
14:10:19,842 INFO  [stdout] (default task-27)         (?, ?)
14:10:27,626 INFO  [org.hibernate.hql.internal.QueryTranslatorFactoryInitiator]
  (default task-28) HHH000397: Using ASTQueryTranslatorFactory
14:10:27,712 INFO  [stdout] (default task-28) Hibernate:
14:10:27,712 INFO  [stdout] (default task-28)     select
14:10:27,713 INFO  [stdout] (default task-28)         user0_.username as username1_2_,
14:10:27,713 INFO  [stdout] (default task-28)         user0_.password as password2_2_
14:10:27,713 INFO  [stdout] (default task-28)     from
14:10:27,713 INFO  [stdout] (default task-28)         USERS user0_
14:10:27,713 INFO  [stdout] (default task-28)     where
14:10:27,713 INFO  [stdout] (default task-28)         user0_.username=?
14:10:27,713 INFO  [stdout] (default task-28)         and user0_.password=?
14:10:41,441 INFO  [stdout] (default task-29) Hibernate:
14:10:41,441 INFO  [stdout] (default task-29)     select
14:10:41,441 INFO  [stdout] (default task-29)         book0_.id as id1_0_,
14:10:41,441 INFO  [stdout] (default task-29)         book0_.price as price2_0_,
14:10:41,441 INFO  [stdout] (default task-29)         book0_.title as title3_0_
14:10:41,442 INFO  [stdout] (default task-29)     from
14:10:41,442 INFO  [stdout] (default task-29)         BOOKS book0_
14:11:18,818 INFO  [stdout] (default task-31) Hibernate:
14:11:18,818 INFO  [stdout] (default task-31)     call next value for hibernate_sequence
14:11:18,824 INFO  [stdout] (default task-31) Hibernate:
14:11:18,824 INFO  [stdout] (default task-31)     insert
14:11:18,824 INFO  [stdout] (default task-31)     into
14:11:18,824 INFO  [stdout] (default task-31)         OPERATIONS
14:11:18,824 INFO  [stdout] (default task-31)         (booksIds, username, id)
14:11:18,824 INFO  [stdout] (default task-31)     values
14:11:18,824 INFO  [stdout] (default task-31)         (?, ?, ?)
```