Deployment to an application server

For deployment to an application server the required elements are a properly configured Java application server and a database system capable of communicating with it, in other words, for which the appropriate driver exists.

The database system only needs to contain a corresponding database to the one defined in server configuration. Tables are created and filled during the deployment of the application.

The configuration of the application server must set the database as data source and establish the security domain. The guide how to install the application to the JBoss Application Server.:

1. Download the latest final version of JBoss Application Server 7 (currently it is 7.1.1 from http://www.jboss.org/jbossas/downloads).
2. Extract the downloaded archive to the destined folder.
3. Create the database for the application in the database system.
4. Copy the driver for the database to the `standalone/deployments` folder (for example the apropriate JDBC Driver for MySQL is called Connector/J and can be obtained at http://www.mysql.com/products/connector/.
The file `mysql-connector-java-5.1.22-bin.jar` would then be copied to `server-folder/standalone/deployments`)
5. Configure the server:
    a) Open file `server-folder/standalone/configuration/standalone.xml`
    b) Add the appropriate data source (inside the `datasources` xml element), example:

```
<datasource jta="false"
     jndi-name="java:jboss/datasources/mysqlfusionweb"
     pool-name="FusionWeb" enabled="true" use-ccm="false">
     <connection-url>jdbc:mysql://localhost:3306/fusionweb?
     zeroDateTimeBehavior=convertToNull</connection-url>
     <driver-class>com.mysql.jdbc.Driver</driver-class>
     <driver>mysql-connector-java-5.1.22-bin.jar</driver>
     <security>
          <user-name>root</user-name>
          <password>1234</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>
```

The **jndi-name** must correspond to the one in persistence.xml file (default is this). The **connection-url** must correspond to the relevant url for connecting to the database. The **driver** must be the same as the defined driver (in the next step). The **security** element represents the credentials for logging into the database.

c) Add the corresponding driver (inside the drivers xml element), for example:

```
<driver name="mysql-connector-java-5.1.22-bin.jar"
module="com.mysql">
     <driver-class>com.mysql.jdbc.Driver</driver-
class>
</driver>
```

d) Add the security domain (inside the security-domains xml element). Example:

```
<security-domain name="fusionweb">
     <authentication>
          <login-module code="Database" flag="required">
               <module-option name="dsJndiName"
               value="java:jboss/datasources/mysqlfusionweb
               "/>
               <module-option name="principalsQuery"
               value="select password from User where
               username=?"/>
               <module-option name="rolesQuery"
               value="select role,'Roles' from UserRoles
               where User_username=?"/>
               <module-option name="hashAlgorithm"
               value="SHA-256"/>
               <module-option name="hashEncoding"
               value="base64"/>
               <module-option
               name="unauthenticatedIdentity"
               value="guest"/>
               <module-option name="allowEmptyPasswords"
               value="false"/>
          </login-module>
     </authentication>
</security-domain>
```

The security domain specifies the authentication of the users. The value of **dsJndiName** must be same as the datasource.

e) Increase the timeout for the deployment - as during the time the deployment is created, the tables filled and the session constructed, it might not meet the default timeout for the deployment. To the deployment-scanner xml element add attribute `deployment-timeout="600"`.

f) Save and close the file. The configuration can also be managed through the administration console of the server.

6. Start the server. It can be achieved through IDE or the execution of the appropriate file in the `server-folder/bin` folder (for Linux based operating systems `standalone.sh`).

7. Deploy the application - either through IDE, console or copy the `FusionWeb.war` file into `server-folder/standalone/deployments`.

8. Explore the application on: http://localhost:8080/FusionWeb/