Installation guide
Web presentation checker

Jakub Koza

3. listopadu 2014

Obsah

1 Introduction

1.1 Purpose

This document is provided to serve as installation guide for project named Web presentation checker. The document explains how to configure necessary application server with database and how to configure them properly. The document also explains process of building project with Maven tool and deployment to the application server.

2 Pre-requisites

The application was developed on Java EE WildFly 8.0 application server with MySQL database system. This guide covers installation of these servers and necessary JDK but it is also possible to port application on any Java EE application server with EJB and CDI technologies support, and any database system supported by used application server.

2.1 Java environment

Java SE 7 JDK or later version is necessary for compilation of the project with Maven tool and for WildFly application server. It is highly recommended to use the latest update available (project was tested with Java SE JDK version 1.8.0_25).

New Java JDK can be downloaded from Oracle download page.

2.2 Maven

Apache Maven is a software project management and comprehension tool that is used for compilation and building deployable enterprise archive from project source codes. This tool is not necessary if you already own built project enterprise archive (*.ear file).

To install Maven tool please follow instructions on `http://maven.apache.org/download.cgi`. It is recommended to add *bin* directory of installed Maven to your system *PATH* environment variable.

2.3 MySQL database

The project was developed on MySQL database system and so it is recommended to use this database system, although it should not be difficult to port application on any other database system, that is supported by application server.

To install MySQL database please follow the MySQL installation guide. Do not forget to remember root login credentials and the port, that the database server will be listening on.

2.4 WildFly application server

The project was developed and tested on WildFly application server version 8.1.0.Final and this guide covers configuration of application server of this version. It is also possible to use other versions (8+) of WildFly, but the reality between configuration of real application server and this guide could be different. To install WildFly application server simply download archive from `http://wildfly.org/downloads/` and extract in a suitable folder.

3 Configuration

3.1 Database configuration

To configure database first start the MySQL service and then connect to the running database server with your favourite query tool. You can also use command line application *mysql* that comes with MySQL installation. To connect with *mysql* application to MySQL server open shell (or command line in Windows) and type:

```
mysql -u rootUserName -P port -p
```

, where *rootUserName* is the name of the root user created in time of database server installation and the *port* is the TCP port the server is listening on. After executing this command you will be requested to type root's password.

After connecting with your query tool first create new user by executing following query:

```
CREATE USER webchecker IDENTIFIED BY 'secretPassword';
```

where *secretPassword* is password of new user named webChecker.

Then create new database and grant privileges to newly created user:

```
CREATE DATABASE checker CHARACTER SET utf8;
GRANT ALL ON checker.* TO webchecker;
```

Finally use project SQL create script to create required database tables in newly created database schema:

```
source [project_root]/Development/Implementation/create_script.sql
```

where *[project_root]* replace by path to the project root directory.

3.2 WildFly configuration

3.2.1 First start

During first start of WildFly application server it is necessary to create new management user that will be used for signing in to server administration.

Start *add-user.sh* (or *add-user.bat* on Windows)script in WildFly bin directory. Select Management User and add username and password of new management user. Groups selection let empty and on the final question type *no*.

Now you can start your WildFly application server by executing script *standalone.sh* (*standalone.bat* on Windows). If everything goes well, you should see welcome web page `http://127.0.0.1:8080` when you put address into your browser. You can also get to server management by typing address `http://127.0.0.1:9990`. You have to log in by credentials of user created in previous step.

3.2.2 Deployment of JDBC connector

WildFly application server requires JDBC connector to be able to create datasource with your database server. If you follow this tutorial, you use MySQL database server and so you need MySQL JDBC connector. The connector often comes with installation of your database server, but you can also download it from Maven repository. Simply select the latest connector for your version of database system and then download artifact JAR. Now navigate to WildFly server management page (`http://127.0.0.1:9990`), login with your management user, select Runtime ⇒ Manage Deployments. Now select Add and put in path to the MySQL connector JAR file, and select Deploy ⇒ Enable. Do not change name and runtime name.

3.2.3 Creating datasource

Now you are able to create datasource to database schema. In server management navigate to Configuration ⇒ Connector ⇒ Datasources. Now click Add to create new datasource. Type *checkerDB* to Name and *java:jboss/checkerDS*. Now select driver from your MySQL JDBC connector. It should be something like *mysql-connector-java-5.1.31-bin.jar_com.mysql.jdbc.Driver_5_1*. In the final window type *jdbc:mysql://127.0.0.1:3306/checker* to Connection URL where 3306 is the default port of MySQL server and it should be replaced, if your database server uses different one. Type *webchecker* to Username and to

Password type the same password, that was used during creation of database user *webchecker*. Security Domain leave empty and press Done.

Now select your newly created datasource and click on Connection ⇒ Edit. Now tick Use JTA option and Transaction Isolation set to TRANSACTION_READ_COMMITED and click Save.

Now it is necessary to reload your WildFly server. You can do it by simply shutting it down (Ctrl+C in window with started *standalone* script) and then starting server again. Log again to server management, navigate back to datasources and select *checkerDB* and click enable.

Now you have successfully created your datasource. You can test it by clicking Test Connection button in Connection tab of your datasource.

3.2.4 Security subsystem

To allow singing in to the application, it is necessary to create new security subsystem domain in WildFly server management.

Log in to server management and navigate to Configuration ⇒ Security ⇒ Security Domains. Click Add and create new security domain named *PresentationCheckerSecurityDomain*. Now select newly created domain and click View. In Authentication tab click Add and type

```
org.presentation.securitydomain.SaltLoginModule
```

to Code and set Flag to required. Click Save and select newly created module. Now select Module Options tab and add new following key-value pairs to the module:

Key	Value
dsJndiName	java:jboss/checkerDS
principalsQuery	select password from user where email=?
rolesQuery	select 'user', 'Roles' from user where email=?
hashAlgorithm	SHA-256
hashEncoding	hex
hashUserPassword	true
saltQuery	select salt from user where email=?

Now it is necessary to reload WildFly server to save the changes.

3.2.5 Certificate and HTTPS settings

It is necessary to configure encrypted connection between clients and server, not to be able to intercept client's username and password for "Man In The

Middle". You will have to enable HTTPS protocol on your server to enable this feature.

At first you will need certificate for your WildFly server. Usually you have to create certificate by help of the Certificate Authority, but you can also use your untrusted self signed certificate for test purposes. Note that in browsers of clients browsing pages of application, the client browser throws warning, that the page is not trusted.

There are many ways to create your own self signed certificate. For example you can use Java keytool utility to create one.

For example to create new self signed certificate saved in Java keystore protected with password *secret*, you can use following command in console:

```
keytool -genkey -keyalg RSA -alias selfsigned
        -keystore keystore.jks -storepass secret
        -validity 360 -keysize 2048
```

You will be requested to type certificate information and optionally to type password to moreover protect certificate in keystore itself.

Now you have to copy newly created *keystore.jks* to *[WildFly_root]/standalone/configuration* directory. Next it is necessary to configure server configuration file. First stop running WildFly server. Next open configuration file *standalone.xml* (*[WildFly_root]/standalone/configuration* directory) in your favourite text editor. Find row with content:

```
<security-realm name="ApplicationRealm">
```

and under this row paste following lines:

```
<server-identities>
        <ssl protocol="TLS">
                <keystore path="keystore.jks"
                 relative-to="jboss.server.config.dir"
                 keystore-password="secret"
                 alias="selfsigned"
                 key-password="certpass"/>
        </ssl>
</server-identities>
```

Replace *secret* by your keystore password and *certpass* by your certificate password.

Now you have to allow https protocol. Find following line:

```
<http-listener name="default" socket-binding="http"/>
```

and right under this line paste new line:

```
<https-listener name="default-https"
          socket-binding="https"
          security-realm="ApplicationRealm"/>
```

Now save the file and start WildFly server. Your server should now listen for https requests on address `https://127.0.0.1:8443`.

4 Project building

This step is not necessary if you do not want to build enterprise archive from sources, but you already have one.

To build whole project start your shell (command line) navigate to *[project_root]/Development/Implementation/checker* and type command:

```
mvn package
```

Project is compiled and built by Maven tool and after this process you can find final deployable enterprise archive (ear file) in folder *[project_root]/Development/Implementation/checker/checker-ear/target*

5 Project deployment

There are many ways to deploy project on WildFly server. The simplest way is to copy built EAR archive to *[wildfly_root/standalone/deployments]* folder.

Another way is to log in to server management and navigate Runtime ⇒ Server ⇒ Manage Deployments and deploy enterprise archive with deployment wizard.

When project is deployed, you can open it in browser on `http://127.0.0.1:8080/presentation`.