DEPLOYMENT MANUAL

SEMANTIC PHI-BASE WEB INTERFACE

Developer:
Juan Camilo Mesa Polo

Deployment manager:
Gerardo Lagunes García

Principal Investigator:
Alejandro Rodríguez González

Documentation made by: Gerardo Lagunes García

Index

Figures index

Table index

Repository

In this section we introduce the online repository where it is stored the application, deployment manual, database scripts and other necessary files.

The application source code is available in the following link:

https://github.com/GerardoUPM/Semantic-PHI-BASE-Web-Interface

It is important to know the file structure showed in Figure 1. This structure will allow a correct deployment.

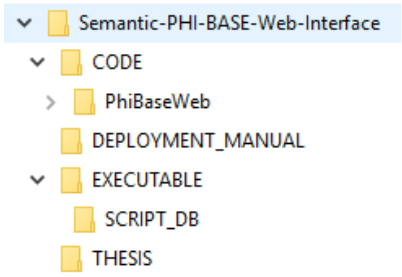

Figure 1. Repository folders structure

In the next table is showed the description of the content of each folder.

Tabl2 1 Description of the folders structure

Folder	**Description**
Semantic-PHI-BASE-Web-Interface	Main project folder. It contains the folder that has the code, executables and README file.
CODE	It contains the source code of the web application.
PhiBaseWeb	It contains all the structure of the source code to be imported by Eclipse. The version used during the deployment stage was Eclipse Kepler.
DEPLOYMENT_MANUAL	It contains the deployment manual.
EXECUTABLE	It contains the necessary executable files for the correct deployment: the file *dbkeygen.jar* is in charge of generating the cyphered keys and the file *download_link* contains a link to download the application executable (.war).
SCRIPT_DB	It contains the scripts to create and put the data in the database: *ScriptCreationDB.sql* is used to create the structure of the database, *ScriptPopulationDB.sql* is used to put the data in the database and *SystemParametrizations.xlsx* describes the configuration parameters used by the application.
THESIS	It contains the master thesis document about the project.

Pre-Requisites

In this section are showed the pre-requisites needed to deploy the application.

Before the deployment it is necessary to have installed the following elements:

1) **Ambiente Linux** (Ubuntu 14.04 LTS). This is the current environment used for the deployment, but any other Linux version should work.
2) **MySQL** as DBMS.
3) **Apache Tomcat 7** as Application server.

Next it is showed the explanation of the installation and the configuration of the pre-requisites.

Installation and pre-requisites configuration

1. **Linux**. We need to install a Linux OS. We have used (and for hence we recommend Ubuntu 14.04 o 16.04 LTS). It is recommended to have the system updated, something that can be done by using the following commands:
    a. sudo apt-get update (It obtains the list of packages to check if there are updated versions to install)
    b. sudo apt-get dist-upgrade (It is in charge of update the distribution)
2. **Java**. It is recommended to install the official version from Oracle. To install this version you have to follow the following steps:
    a. Download Java Server JRE para Linux x86 o x64 depending on the architecture. Download link: http://www.oracle.com/technetwork/java/javase/downloads/index.html
    b. Unzip the file and move the created folder to a known path or follow the installation instructions available at: https://www.java.com/en/download/help/linux_x64_install.xml to also configure the environment variables JAVA_HOME and JRE_HOME.
3. **MySQL**. To install and configure MySQL follow these steps:
    a. Execute: sudo apt-get install mysql-server
    b. Introduce root password
    c. Connect to the database: mysql -u root -p
    d. We have to create a new user and give him privileges over the schema named "*PHIBASEDB*" that will be created later. The following command shows how to create a user named *PHIBASEDB* with password *PHIBASEDB123* and the assignation of privileges. As a security measure it is recommended that the password doesn't match with the username:
        i. CREATE USER 'PHIBASEDB'@'%' IDENTIFIED BY 'PHIBASEDB123';
        ii. GRANT ALL PRIVILEGES ON PHIBASEDB.* TO 'PHIBASEDB'@'%';
    e. Create the database structure by executing the script file *ScriptCreationDB.sql* available in the following folder: EXECUTABLE -> SCRIPT_DB.
    f. Populate the database executing the script file *ScriptPopulationDB.sql* available in the following folder: EXECUTABLE -> SCRIPT_DB.
4. **Apache Tomcat 7**. Download and unzip the application server Apache Tomcat. It can be downloaded from: http://tomcat.apache.org/download-70.cgi. It is recommended to use the last core version of Tomcat 7 because the application executables have been optimized to run over this version. Once you unzip the .tar.gz file it is recommended to rename the folder just to "Tomcat" to allow an easier configuration in the next steps.
    a. Edit the file ***Tomcat/conf/server.xml*** to change the listening port to 80 as is shown in Figure 2:

```
<Connector port="80" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
```

Figure 2. Change listening port in Apache Tomcat

b. Export the variables **JAVA_HOME** y **JRE_HOME** in the script ***Tomcat/bin/catalina.sh***. The new values will be the path where Java Server JRE has been unzipped as is shown in Figure 3:

```
export JAVA_HOME=/opt/jdk1.8.0_77/
export JRE_HOME=/opt/jdk1.8.0_77/jre/
```

Figure 3. Export Java variables

Application installation and configuration

In this section are explained the steps required for the installation and configuration of **Semantic PHI-BASE Web Interface** application.

Installation

1. Copy the executable (*PhiBaseWeb.war*) into the Tomcat server applications folder. The path is: **Tomcat/webapps**. It is important that when you copy the .war file Tomcat server is not running.
    a. To generate the executable it is necessary to open Eclipse, go to the project and in the main application folder, unfold the menu and in "Export" select the option WAR file as in shown in Figure 4. Or, you can directly download the WAR file from the link provided in the file *download_link.RTF*.

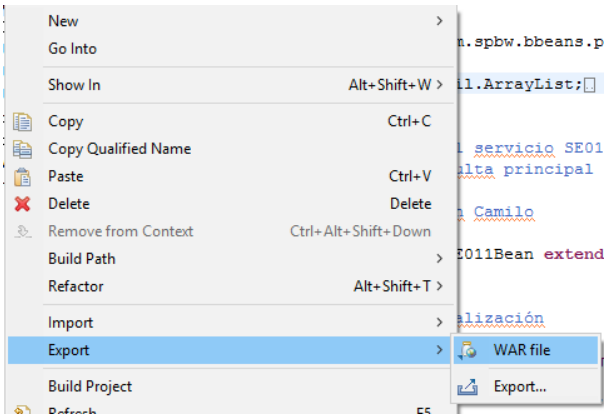

Figure 4. Generate an application executable (.war)

2. Execute the Tomcat startup script. It is available in the path: **Tomcat/bin/startup.sh**. In this momento the deployment of the application will start. You need to wait a few seconds until Tomcat finish to create and unzip all the content inside the folder **Tomcat/webapps/PhiBaseWeb**.
3. Once the deployment has finished, you need to execute the script to shutdown Tomcat and proceed to configure the application. To shutdown Tomcat you should execute the following script: **Tomcat/bin/shutdown.sh**.

In the next section will be shown the configuration that need to be performed over the files of **Semantic PHI-BASE Web Interface** to allow a correct behaviour of the application.

Configuration

1. As a security measure we have a cypher process that should be done using the program contained in the file **dbkeygen.jar.** Steps needed:
    a. Open a terminal and go to the path where you have located the file **dbkeygen.jar**.
    b. Execute the following command: **java -jar dbkeygen.jar <text-to-cypher>**. An execution example is the following:

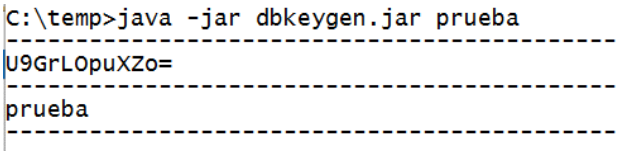

Figure 5. Creating a cypher text using dbkeygen.jar

c. Copy the text that it is shown in the first row (the output) and copy it in the corresponding property. In the example it will be **U9GrLOpuXZo=**.

2. Now it is showd a table with the propertys that can be configured in the properties file located in **Tomcat/webapps/PhiBaseWeb/WEB-INF/conf/config.properties**. Those properties that are required for the correct behaviour of the platform are highlighted.

Table 2. Configuration properties

Property	**Configuration / Description**
edu.upm.spbw.db.driver	Database driver. In this case the default value is the one for MySQL and it is correct.
edu.upm.spbw.db.url	It is the URL to Access the database. It should be put in JDBC address format. In the configuration example we have a local address: **jdbc:mysql://127.0.0.1:3306/PHIBASEDB**
edu.upm.spbw.db.username	Database username. It is the name created in the pre-requisites when the database was installed. This value should be cypher by using **dbkeygen.jar**.
edu.upm.spbw.db.passwd	Password of the user. This value should be cypher by using **dbkeygen.jar**.
edu.upm.spbw.db.dialect	Database dialect. In MySQL case current value should be used.
Filecfg	Location of the log configuration file. This file it is located in the same path of the configuration file.
edu.upm.spbw.sparql.prefix	Location of the file that contains the prefixes used in the SPARQL queries against Virtuoso endpoint. This file is also located in the same path of the configuration file.
edu.upm.spbw.login.link	It is the link to enter into the application. This value will be used in the emails that are sent automatically when a new user is created.
edu.upm.spbw.mail.logo	Location of the logo to be attached in the emails.
edu.upm.spbw.mail.account	Email account used to send the emails from the system. This value should be cypher by using **dbkeygen.jar**.
edu.upm.spbw.mail.password	Password of the email account. This value should be cypher by using **dbkeygen.jar**.
edu.upm.spbw.mail.host	SMTP server address of the email server used to send the emails. This value should be cypher by using **dbkeygen.jar**.
edu.upm.spbw.mail.port	SMTP server port of the email server used to send the emails. The value should be 1 or 0. This value should be cypher by using **dbkeygen.jar**.

edu.upm.spbw.mail.auth	Boolean value to indicate if authentication is required when the emails are sent. This value should be cypher by using **dbkeygen.jar**.
edu.upm.spbw.mail.tls	Boolean value to indicate if the email should be sent using TLS. The value should be 1 or 0. This value should be cypher by using **dbkeygen.jar**.

3. The next file to configure is **Tomcat/webapps/PhiBaseWeb/WEB-INF/conf/logger.properties**. Most of the parameters of this file are by default and are not needed to be changed. The only parameter to be modified is **log4j.appender.fileappender.File**. The value should be the path where the user wants to generate the application log files.
4. Finally, once all the configuration files are correctly configured, we can start again the Tomcat server to check that everything is working properly. The script to start the server is **Tomcat/bin/startup.sh**.
5. To verify that the application has been deployed, open the Tomcat configuration panel with a Web browser; If you have local go to: http://127.0.0.1:8080 or to http: // localhost: 8080, the result is shown in Figure 6.

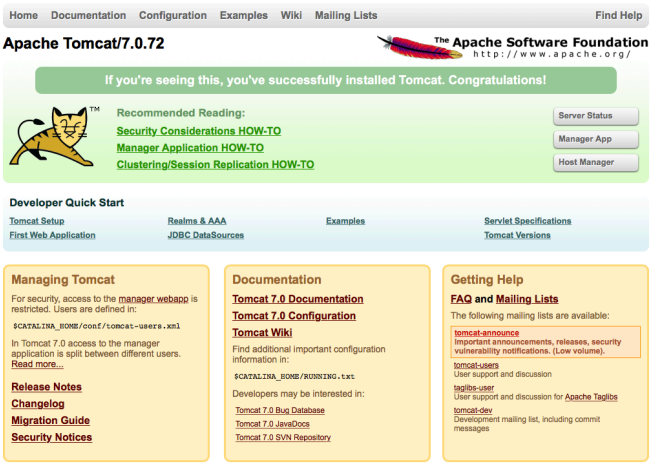

Figure 6 Panel principal del servidor Tomcat

Now go to the **Manager App** option, this option will ask you to sign in to the applications administration panel, the login is shown in Figure 7. The administration screen is shown in Figure 8.

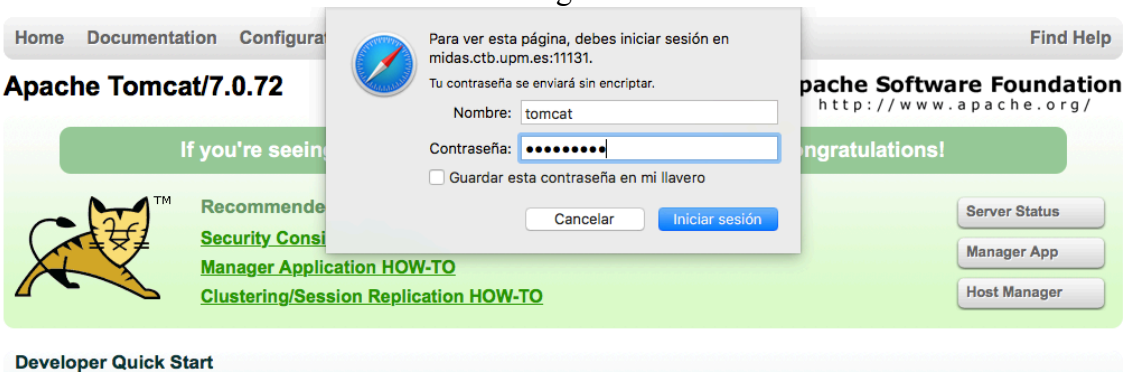

Figure 7 Enter credentials to enter the application management panel

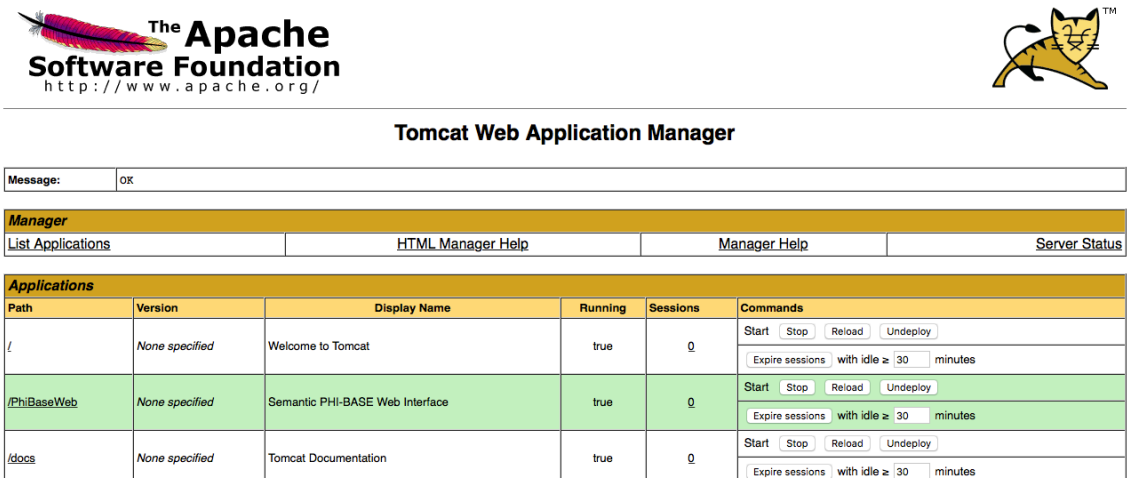

Figure 8 Application Management

Select in the **Path** column the application /**PhiBaseWeb**; Another webpage will open that is the main page of the **Semantic PHI-BASE Web Interface** application, their deployment is shown in Figura 9.

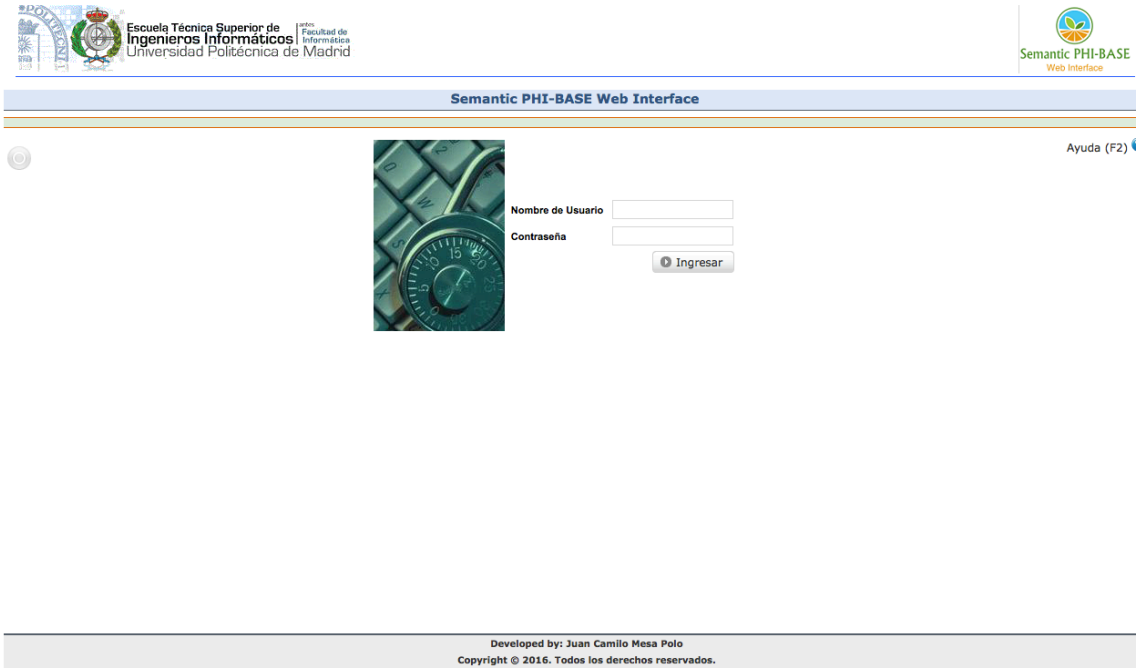

Figura 9 Main page for Semantic PHI-BASE Interface Web

6. To enter the Semantic PHI-BASE Web Interface system type user name: admin@phibaseweb.com and password: sotileza.

Maintenance and revision

In this section is explained the maintenance of the application and how to check that it is working properly. For this, it is necessary to check the work available in the following link: https://github.com/GerardoUPM/Semantic-PHI-BASE-Web-Interface/tree/master/THESIS .

Developer manual

In this section are explained how to make modifications to the application. For this, it is necessary to check the work available in the following link: https://github.com/GerardoUPM/Semantic-PHI-BASE-Web-Interface/tree/master/THESIS .

Last updates

In this section are explained in a general way the last updates done in the application.

1. Modification of the original database script.:
    a. The tables SEPCONCE, SEPRELCO y USDLOGQU were modified to store more relevant information. It was also modified the INSERT operations that have to be applied in the tables with the new data.
2. Update of JavaScript method “inner_showPhiBaseGraph” to allow the sending to the web application valid links.
3. Change the language the application, from spanish to english.