Data collection – historic setup

Author: Matsobane Khwinana

The following tools are required to run the historic data extraction:

1. Java SE 8 – Oracle version was used
2. Apache Maven 3.2.5 or higher
3. Git and GitHub account (for cloning the code repository)
4. Netbeans 8.1
5. Java EE 7 – Jboss Wildfly 10
6. MongoDB

1. Install Oracle Java 8 in Ubuntu via PPA

Add "webupd8team" PPA on Linux terminal (Ctrl+Alt+T):

```
sudo add-apt-repository ppa:webupd8team/java
sudo apt-get update
sudo apt-get install oracle-java8-installer
```

Accept Oracle licenses and once the installation has completed, execute the following (in Linux terminal) to check if Java has been installed properly:

```
java -version
```

1.1 Configure Oracle Java 8

Add JAVA_HOME to the PATH environment. This will add the variable system-wide; on Linux terminal (Ctrl+Alt+T):

```
sudo gedit /etc/profile.d/my_env.sh
```

This will open the text editor. Add the following lines, save (Ctrl+S) and close the editor (Ctrl+Q):

```
#Java
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
export PATH=$PATH:$JAVA_HOME/bin
```

2. Install and Configure Apache Maven

Download Apache Maven 3.2.5 or later.

Create Apache folder in /opt; on Linux terminal (Ctrl+Alt+T):

```
sudo mkdir /opt/apache
```

Extract the downloaded tar ball into "/opt/apache" (where 3.x.x is version of your maven).

On Linux terminal (Ctrl+Alt+T):

```
cd /opt/apache
sudo tar -xvf ~/Downloads/apache-maven-3.x.x-tar.gz .
```

Add M2_HOME and M2 to the PATH environment;on Linux terminal (Ctrl+Alt+T):

```
sudo gedit /etc/profile.d/my_env.sh
```

Add the following lines above the #Java line (where 3.x.x is version of your maven):

```
#Maven
export M2_HOME=/opt/apache/apache-maven-3.x.x
export M2=/$M2_HOME/bin
export PATH=$PATH:$M2
```

Restart your OS.

Test that maven was installed and configured correctly; on Linux terminal (Ctrl+Alt+T):

```
mvn --version
```

You should get an output similar to this below:

```
Apache Maven 3.2.5 (12a6b3acb947671f09b81f49094c53f426d8cea1;
2016-06-30T19:29:23+02:00)
Maven home: /opt/apache/apache-maven-3.2.5
Java version: 1.8.0_76, vendor: Oracle Corporation
Java home: /usr/lib/jvm/java-8-oracle/jre
Default locale: en_ZA, platform encoding: UTF-8
OS name: "linux", version: "3.13.0-48-generic", arch: "amd64",
family: "unix"
```

3. Install and configure git

Install git from the Ubuntu software repository; on Linux terminal (Ctrl+Alt+T):

```
sudo apt-get install git
```

Add your user details for the "gitconfig" file; on Linux terminal (Ctrl+Alt+T):

```
git config --global user.name "John Doe"
git config --global user.email John.Doe@gmail.com
```

You should be able to view your git user information on your "`~/.gitconfig`" file.

Create an account at GitHub if you don't have only already and clone the following project:

```
https://github.com/garethstephenson/ELEN7046_Group2_2016
```

Download and install Netbeans.

Follow instructions from Netbeans website to install.

Open the netbeans project from:

```
.../ELEN7046_Group2_2016/twitter/java-data-extraction-service
```

Download and install Jboss Wildfly 10.

Follow instructions from the website to install.

Download and install MongoDB.

Follow instructions from the website to install.

The setup is completed!

Running the data collection.

Make sure that **Wildfly** and **MongoDB** are both running.

In Netbeans, Right-click on the project and select run…

Using a REST client

http://localhost:8080/historic-data-extraction-service/rest/extract-data/byId/729674502339055617

To run a scheduled data collection, the java class “” and change the cron expression as desired. Clean and Build, and Run...when the time is reached the collection will start.

To distribute the data via REST service. Using a REST client :

http://localhost:8080/historic-data-extraction-service/rest/distribute-data

Alternatively, data collection can be triggered via test cases in project.

We hope you found this useful!