PSM SERVER CONFIGURATION DOCUMENTATION

© 2016 Genzet.

Table of Contents

I. Getting All the Required Components

Download all the required components:

1. Github.com/psmessaging
2. Cassandra: http://cassandra.apache.org/download
3. Hector for Cassandra: jar is included with project
4. Oracle's Glassfish Full Platform: http://glassfish.java.net/download.html
   a. You could use tomcat but configuration is not explained in this document.

II. Configuring Cassandra

1. Introduction

Configuring a Cassandra database is well documented elsewhere, but how to configure the database to the default settings, as expected to be configured by the web application, when you downloaded it, is explained in this section.

2. Downloading Cassandra

- Download Cassandra at http://cassandra.apache.org/download.
- Documentation is available at their “Getting Started” resource at http://wiki.apache.org/cassandra/GettingStarted.

Unzip Cassandra to a place of your choice and continue reading to get the right configuration settings to work with this project.

3. Configuring the Cassandra.yaml file

Locate the **cassandra.yaml** file in the 'conf' directory in the root of the cassandra directory. Open this file with a text editor and we will now begin to configure Cassandra's settings for the project to interact with it.

1. cluster_name: 'PSM Cluster'
2. listen_address: 127.0.0.1
3. rpc_port: 9160
4. rpc_server_type: sync
5. initial_token: 1 #you can blank this if you are in a clustered environment

Save & Close and move on to the next step...

4. Setup Cassandra

1. Make sure Cassandra is running by using cassandra.bat for windows or cassandra, if on another platform, from the bin directory.

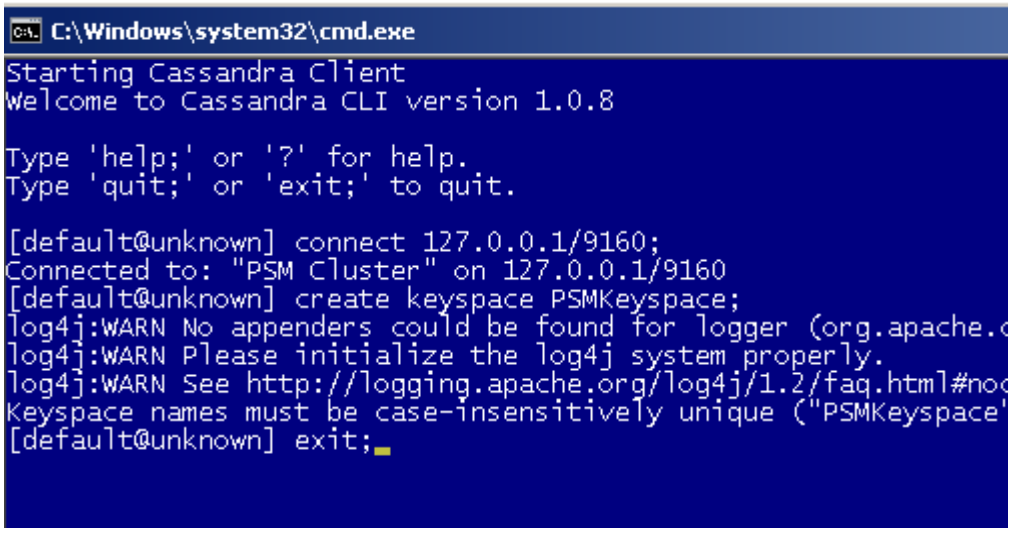
2. Open: Cassandra-cli.bat

3. Connect to the cassandra server by typing: 'connect 127.0.0.1/9160;' then press Enter

4. Begin by creating the keyspace called PSMKeySpace by using the create keyspace command.

5. Type in: 'create keyspace PSMKeySpace;' then press Enter

6. Type in: 'exit;' then Press Enter



III. Configuring Glassfish

1. Download Full Version of Glassfish

Download the full version of Glassfish from http://glassfish.java.net/download.html and unzip it to your folder of choice. *i.e. C:\glassfish-4-1-1*

2. Start Glassfish

Start Glassfish by executing C:/glassfish-4-1-1/glassfish4/bin/asadmin.bat

3. Open the Admin Console

Open your web browser and goto the admin console at http://localhost:4848. Minimize this page and we will come back to it later. Glassfish should now be running and we will come back to the admin console to deploy the psm-server later on in this document; after we build the psm-server.war, psm-ejb.jar, idgenerator.war.

IV. Building the Different Projects

1. Building with Netbeans projects.

It's pretty much self explanatory. The project was original designed in Netbeans, https://netbeans.org/downloads. It uses the jersey REST servlet to serve the web request.

It requires jersey 1.8+ and jersey-gf-server.jar (included in the lib folder). Make sure that the psm-ejb project is linked and packaged with psm.war.

2. Building with Eclipse IDE for Java EE Developers.

This project should also be able to be built using Eclipse IDE for Java EE Developers, http://eclipse.org/downloads. But will not go into detail on how to build it, as it is similar to any other Java EE project, that uses REST endpoints.

V. Deploying on Glassfish 4.1.1

If you still have the admin console open, un-minimize it. Otherwise, goto http://localhost:4848 and do the following steps:

1. Deploy IDGenerator.war

1. Click applications from the left menu
2. Click Deploy
3. Browse for **IDGenerator.war**
4. Set the context root to id (that way it will result in http://localhost:8080/id)
5. Uncheck 'implicit cdi'
6. press ok
7. Should have succeeded

2. Deploy psm.war

1. Click applications from the left menu
2. Click Deploy
3. Browse for **PSM.war**
4. Set the context root to psm (that way it will result in http://localhost:8080/psm)
5. Uncheck 'implicit cdi'
6. press ok
7. Should have succeeded

3. Troubleshooting

If you get "ClassNotFoundException: com.sun.jersey.spi.container.servlet.ServletContainer" error make sure to include *Jersey 1.8+* and *jersey-gf-server.java* library in the web projects (idgenerator and psm)

4. Additional Configuration Suggestions

You can use Glassfish with Apache HTTP Server through an AJP Connector and using mod_rewrite and mod_jk you can forward requests to http://localhost:8080/psm and keep usage statistics and do load balancing and all that stuff. A good guide on how to do this is at: http://community.oracle.com/blogs/amyroh/2012/02/15/running-glassfish-312-apache-http-server

Further information can be found on AJP Connecter at: http://tomcat.apache.org/connectors-doc/ajp/ajpv13a.html

VI. List of Endpoints and Description

1. /server/install

Method: GET
Produces: application/json

After configuring Cassandra and deploying all the web modules, use this endpoint to setup the database with all the columns. This should be commented out and rebuild and redeployed after first use. You don't want to run this more than once.

2. /server/start

Method: GET
Produces: application/json

After starting up the server, execute this endpoint to setup variables so the web service will work properly.

3. /server/stats

Method: GET
Produces: text/html

Returns a count of all the count columns, showing you how many records there are in each column. Such as count of facebook users, foursquare users, deleted items, etc... You should also remove this if you are using this on a production server.