

IRP - TMI Infrastructure documentation

Author: Charles Ofoegbu

Date: 25th August 2014

Contents

1	Codebase.....	2
1.1	Persistence codebase.....	2
1.2	Server codebase.....	2
1.2.1	src - java	2
1.2.2	src - resources	3
2	Deployment.....	4
2.1	System requirements.....	4
2.2	Deployment steps	4
3	Production server.....	5
3.1	Database Credentials	5
3.2	Tomcat Instances	5
3.2.1	Restarting Tomcat Instances.....	5
3.3	Troubleshooting.....	5
4	Functionalities.....	5
5	Pending Tasks.....	5
6	Known bugs.....	5

1 Codebase

The codebase for TMI infrastructure is made up of two major parts as follows: Persistence codebase and the Server codebase. Both are maven based. The Server codebase is dependent on the persistence codebase.

1.1 Persistence codebase

The Persistence codebase contains the main architectural design and implementation for the infrastructure. The codes in this directory should never be modified as they are 100% generated by Velocity and Maven. Modifications are only possible through editing the UML Model in the following directory: `\persistence_code_base\mda\src\main\uml` using Magic Draw UML and then regenerating the codes from the model using maven.

After the codes are generated, they are automatically installed in the maven repository and equally available for use by the Server codebase as far as they both share the same maven repo. If not, the generated jars can be installed manually to the maven repository. The following two generated jar files are the dependencies needed by the Server codebase:

- tweetdeck-common-1.0-SNAPSHOT.jar and
- tweetdeck-core-1.0-SNAPSHOT.jar

They can both be found in the following directory: `\persistence_code_base\app\target\tweetdeck-1.0-SNAPSHOT`

git url - https://github.com/tcofoegbu/irp-ecosystem-social-twitter/persistence_code_base

1.2 Server codebase

The server codebase consist of the integration with the persistence codebase, web service implementation, Twitter fire-hose and User streaming implementation with Twitter4j API,

1.2.1 src - java

Listed below are the packages and classes available in the server codebase

- uk.ac.dotrural.irp.ecosystem.social.twitter
 - ✓ StatusListenerImpl.java
 - ✓ TweetProcessor.java
 - ✓ TwitterComponent.java
 - ✓ UserStreamListenerImpl.java
- uk.ac.dotrural.irp.ecosystem.social.twitter.common
 - ✓ AutoRestore.java
 - ✓ Constants.java
 - ✓ MD5Util.java
 - ✓ PropertyFileUtil.java
- uk.ac.dotrural.irp.ecosystem.social.twitter.dao
 - ✓ DataAccessProxy.java
- uk.ac.dotrural.irp.ecosystem.social.twitter.exception
 - ✓ AuthenticationException.java
- uk.ac.dotrural.irp.ecosystem.social.twitter.listeners
 - ✓ SystemComponentInitializer.java
- uk.ac.dotrural.irp.ecosystem.social.twitter.pojo
 - ✓ AccessPojo.java

- ✓ ApplicationStatePojo.java
- ✓ AuthenticationPojo.java
- ✓ ConversationChainPojo.java
- ✓ ExceptionPojo.java
- ✓ TrackListPojo.java
- ✓ UserPojo.java
- uk.ac.dotrural.irp.ecosystem.social.twitter.quartz
 - ✓ UpdateTweetJob.java
- uk.ac.dotrural.irp.ecosystem.social.twitter.security
 - ✓ ApplicationSecurity.java
- uk.ac.dotrural.irp.ecosystem.social.twitter.service
 - ✓ AccessControlService.java
 - ✓ SystemInfo.java
 - ✓ TrackListService.java
 - ✓ TwitterUserProfileService.java
 - ✓ UserService.java

git url server codebase - https://github.com/tcofoegbu/irp-ecosystem-social-twitter/server_code_base

1.2.2 src - resources

This directory contains various resources used by the code mostly for configurations i.e.

- log4j.properties file
- HibernateConfig.xml
- twitter4j.properties

2 Deployment

This section contains the information necessary for configuring and deploying the system.

2.1 System requirements

The following are required for the deployment of the infrastructure:

1. Minimum of Oracle JDK-7
2. Minimum of Tomcat7
3. Minimum of PostgreSQL 8.1
4. PhpPgAdmin

2.2 Deployment steps

When the above requirements are met, the next steps are as follows:

- Create the appropriate database users
- Create database Schema from PhpPgAdmin interface
- Ensure the database connection setting in the configuration file -TweetDeskHibernateConfig.xml matches the database user/schema created
- Drop the war file in the */webapps* directory of the installed tomcat
- Start the tomcat

The database connection setting and the twitter credential can be found in the following directory of the exploded war file "*webapps/{app-name}/WEB-INF/classes*". However, It is NOT recommendable to tamper with the files there as they are exploded each time Tomcat starts up. Hence any changes made will be overwritten unless the war file is deleted after the files are edited.

Also since the database connection credentials are known ahead of time, it is recommended to create a carbon copy of the known credentials - username, password and database, on the database server. Every other thing is pretty much auto generated when the app runs for the first time, i.e. the database schema(generated by hibernate), properties files(generated by app code) etc.

3 Production server

The production server is available on <http://dtp-24.sncs.abdn.ac.uk>. It can be access via ftp and ssh using Filezilla and Putty respectively. The server authentication is by public/private key - David or Yan can provide this.

3.1 Database Credentials

The database server can be accessed and managed directly using PhpPgAdmin web tool. Below is the credentials needed to access the server:

- PhpPgAdmin server url - <http://dtp-24.sncs.abdn.ac.uk/phpPgAdmin/>
- username: dotrural
- password: y0l0 (Note: 0 = zero)

3.2 Tomcat Instances

The following are the various Tomcat instances running on the production server for TMI on dtp-24.snc:

- /usr/share/apache-tomcat-7.0.53
- /usr/share/apache-tomcat-7.0.53_firehorse_instance1
- /usr/share/apache-tomcat-7.0.53_userstream_instance1

The three instances have the same war file deployed on them. However, three of them perform different task when fired. The task to be performed by an instance depends on the settings configured in EcosystemSocialTwitter.properties found in the bin directory.

3.2.1 Restarting Tomcat Instances

To restart any of the instances all you need to do is to change into the bin directory of the target instance and then execute the command ***sh restart.sh***

3.3 Troubleshooting

4 Functionalities

5 Pending Tasks

6 Known bugs