SLACK IMPLEMENTATION

SLACK ACCOUNT USED BY DEVELOPER:

email: [ameyambade@gmail.com](mailto:ameyambade@gmail.com)

domain: ameyxd.slack.com

OAUTH

* The Slack integration service uses Oauth 2.0
* simple-slack-api is used to implement real time notifications using webhooks (thanks to ullink from GitHub)
* A database is maintained which securely saves email addresses, corresponding codes and access\_tokens to facilitate speedy login
* A property file named 'slack.properties' is maintained in the src folder. This file has configurable client details and database credential entries, which can be changed (with caution) after creating a new App on Slack or on creation of a new database.
* This property file also has relavant details about the redirect\_uri and any other configurable data that the code can access.
* This enables a user to completely alter the credentials for use without having access to the actual code. Care has been taken to keep the least amount of data hard coded in the project.

SLACK STRUCTURE

* Every user has a team, consisting of an admin, and other team members.
* This team has a domain name, like, for example, xyz.slack.com.
* A user accesses Slack using this team domain.
* The user can create channels, add team members to specific channels, and post messages.
* Simpification of messaging and rich media attachment is therefore possible using Slack.

KEYWORDS

Following are the keywords the bot can respond to:

* **Help** : Diplays help content to users, like information on how to use the bot, and displays a list of keywords supported along with their associated use.
* **Slack** : Displays a link that takes a user to the Slack login page to give authorization access to Teamchat.
* **Notifyme** : This keyword supplies real time notifications of anything happening in Slack
* **Channel** : Helps create new channel
* **Teaminfo** : Gives details of a user's team, and domain.
* **Userlist** : Gives details about a user's team members.
* **Search** : Searches Slack channel/s for specific keyword/s supplied by the user
* **Message** : Helps post messages to a specific channel in Slack using Teamchat.
* **Delete** : Deletes the most recent message posted to Slack through Teamchat.

API DETAILS

The Slack integration service uses Slack's provided API for pulling already present data and metadata from Slack and for additional posting to Slack.

The simple-slack-api is used for the Real Time Messaging notification system triggered by keyword 'notifyme'.

I acknowledge and thank user ullink from GitHub for this Websocket based API.

Link for reference: [*https://github.com/Ullink/simple-slack-api*](https://github.com/Ullink/simple-slack-api)

The code is written entirely in Java, using Eclipse Luna IDE, and requests to the OAuth and the various methods and events provided by the Slack API are made using HTTP Client.

FOR A NEW DEVELOPER WANTING TO EDIT

The code is written entirely in Java, with methods, classes and code formatting pertaining to standard Teamchat SDK and Java, so a new developer wanting to edit the code (with caution) must be fairly accustomed to coding in Java.

Step 1: Slack account creation and API configuration

* Create an account on Slack.com.
* Once logged in, click the downward arrow next to your team name on the sidebar on the far left side of the screen and select 'Team Settings'.
* On the new webpage, click on 'Menu' in the top left corner, then choose 'API' from the sidebar that pops up.
* On the new webpage 'api.slack.com', select 'My Apps' for the top right corner of the screen, and then choose 'Create new Application'.
* Enter the details of your new app on Slack (name, URL, redirect URI) and click 'Create' (NOTE: The redirect URI must match that of the servlet used).
* Format of the redirect URI must be 'http://yourservername:yourport/Slack/slack\_auth'
* Once you have created the app, you have access to the Client ID and Client Secret. These are parameter that would be required.
* Update the values of your redirect URI, client\_id and client\_secret in the /src/slack.properties file.

Configuring the Property file

The property file has the following fields:

* client\_id : The client\_id of the app you just created
* client\_secret : The client\_secret of the app you just created
* redirect\_uri : The URI of the format mentioned above
* DB\_URL : The URL of the database jdbc:mysql://localhost/databasename
* JDBC\_DRIVER : com.mysql.jdbc.Driver
* USER : username of the database user
* PASS : password of the database user

NOTE: The database of the name databasename must have a table named Slack with three columns : email (varchar(50)), code (varchar(50)), access\_token (varchar(50)).

That is it! Unless the code needs changes, you do not need to edit the code at all.

-Amey Ambade.