POCKET IMPLEMENTATION

[POCKET ACCOUNT USED BY DEVELOPER:](mailto:slack@teamchatbot.com)

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

OAUTH

* The Pocket integration service uses Oauth 2.0
* A database is maintained which saves email addresses, corresponding codes and access\_tokens to facilitate speedy login
* A property file named 'pocket.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 Pocket 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.

POCKET STRUCTURE

* Every user has access to the list of webpages they have added to Pocket.
* These webpages can be archived and maintained in form of lists using tags, etc.
* This allows better bookmarking and maintenance of webpages, and facilitates offline reading.
* The user can save links to Pocket using the Pocket apps or extensions or directly through the website [*https://getpocket.com*](https://getpocket.com/)*.*

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.
* **Add** : Helps a user add web content to their Pocket for reading later. All the user has to do is paste in the link here.
* **Retrieve** : Searches Pocket for links specific keyword/s and other search filters and posts them to Teamchat.

API DETAILS

The Pocket integration service uses Pocket's RESTful API for pulling already present data and metadata from Pocket and for pushing data to Pocket.

Link for reference: [*https://getpocket.com/developer*](https://getpocket.com/developer)*/*

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 Pocket 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: Pocket account creation and API configuration

* Head over to getpocket.com to create an account.
* Once logged in, go to [*https://getpocket.com/developer*](https://getpocket.com/developer)*/*
* On the new webpage, choose 'Create New App' .
* Enter the details of your new app on Slack (name, description, permissions to grant, platforms) and click 'Create' (NOTE: The redirect URI must match that of the servlet used).
* You can edit these app details at any point in time.
* Format of the redirect URI must be 'http://yourservername:yourport/Pocket/Redirect'
* Once you have created the app, you have access to the consumer\_key. This parameter would be required later.
* Update the values of your redirect URI and consumer\_key in the /src/pocket.properties file.

Configuring the Property file

The property file has the following fields:

* consumer\_key : The consumer\_key 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 Pocket 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.