**Google Analytics Implementation**

Steps to get client ID and secret for Oauth:

1. Firstly go to <https://console.developers.google.com/project> and create a project and enable Analytics API which is listed in Advertising APIs.
2. Click the credentials tab present in APIs and auth. Click the button Create new client ID.
3. Select the option 'web application' in the pop-up screen and click the button configure consent screen.
4. In consent screen, enter the product name and save.
5. Again select web application and enter your 'call back URIs' in the below textbox.
6. Now you got all the required credentials you need. You can edit your callback Uris whenever you want by clicking 'Edit Settings' in the in the credentials page itself.

Steps to get Access Token of the user using the above credentials:

1. Firstly, redirect the user along with user's teamchat mail-id and room-id to 'servlet\_url' noted in properies file.
2. Rerieve the mail-id and room-id in servlet to which the user is redirected.
3. Store the mail-id and room-id along with two null values for access-token and refresh-token in the database.
4. Redirect the user to the authorization-url with state parameter as mail-id and once the user accepts, response will be redirected to the call-back-url noted in properties file.
5. In call back servlet, retrieve the 'state' parameter(mail-id) and 'code' to trade for access token and forward these parameters to the class where you are going to trade for access-token.
6. In the redirected class, send a POST request with the 'code' recieved above and other necessary parameters client\_id, client\_secret, grant\_type, redirect\_uri.
7. Now you'll get json response with access\_token, refresh\_token and other details.
8. Now store these tokens by updating the tuple which is created earlier using the user's mail-id stored in 'state' parameter.
9. Get the room-id of the user from the database and post a simple chatlet mentioning that he's connected now.

Functionalities(keywords) providing here:

1. current – It gives current details related to your configured URL.
2. Report – It gives the report of the URL in duration provided by the user.

Current:

1. Retrieve all the 'Accounts'(different from google accounts) user is having in his google analytics account by sending a GET request with users accesstoken to <https://www.googleapis.com/analytics/v3/management/accounts> along with the parametes(access\_token, client\_id, scope).
2. Now show a drop down list to the user to select his desired account.
3. Get the account name chosen by user and make above GET request again to get the ID of the account selected by the user.
4. Using the account\_id, get all the properties the user's having by sending a GET to <https://www.googleapis.com/analytics/v3/management/accounts/>{account\_id}/ webproperties along with the same parameters which we used in above request.
5. Now show adropdown list to the user to select his desired property.
6. Get the property name selected by the user and get its ID by sending the same GET request again.
7. Use the property ID in the below URL to get the current details https://www.googleapis.com/analytics/v3/data/realtime?ids=ga:{property\_id}&metrics=rt:activeusers&dimensions=rt:medium,rt:city,rt:browser&client\_id={client-id}&access\_token={access-token}&scope={scope}
8. From the resulting JSON retrieve the necessary details and display in the form of a HTML table.

Report:

1. Follow the above step-1 to step-4.
2. While asking the your to select property from drop down ask him to enter 'start-date' and 'end-date' in (YYYY-MM-DD) formatto get details of that particular duration.
3. Get the property name selected by the user and get its ID by sending the same GET request again.
4. Use the property ID in the below URL to get the current details https://www.googleapis.com/analytics/v3/data/realtime?ids=ga:{property\_id}&start\_date={start-date}&end\_date={end-date}&metrics=ga:pageviews,ga:sessions,ga:users,ga:newUsers,gs:avgSessionDuration&dimensions=rt:city,rt:browser&client\_id={client-id}&access\_token={access-token}&scope={scope}
5. From the resulting JSON retrieve the necessary details and display in the form of a HTML table.

Extra Details:

1. Using \*\*\*.properties file to store all the details.
2. Using embedded url so that everything will be opened in Teamchat app itself.
3. Access token will be expired frequently resulting in 401 error. At that time getting refresh token from database to get new access token.