**Trello Implementation**

Steps to get client ID and secret for Oauth:

1. After logging in visit this link <https://trello.com/app-key>
2. Now you got client-id and client-secret to perform Oauth.
3. To get URLs for Oauth visit <https://trello.com/docs/gettingstarted/oauth.html>
4. Anyway, i have used Scibe API for Oauth is used. This will generate access token provided client ID and Secret.

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, room-id, request-token, request-token-secret and null(for future use(accesstoken)) in the database.
4. Redirect the user to the authorization-url once the user accepts, response will be redirected to the call-back-url with a parameter named 'oauth\_verifier' and request-token as 'oauth\_token'.
5. This oauth\_token is same one which we stored earlier in database.
6. Now retrieve the token secret and regenerate the request\_token.
7. Use this generated request\_token and 'oauth\_verifier' to trade for access-token.
8. {service.getAccessToken(requestToken, verifier)} gives the accesstoken.
9. Update the database by adding this accesstoken.
10. Get the room-id from database and post a message as the user is connected.

Functionalities(keywords) providing here:

1. organizations – It displays the names of organizations present in the user's account.
2. Boards – It displays the names of the boards present in the user's account with links.
3. Notify – It manages the webhooks (create and delete)

Organizations:

1. Here i am going to retrieve the access token from the database and makes GET request to <https://trello.com/1/members/my/organizations?key>[=](https://trello.com/1/members/my/organizations?key=PropertiesFile){api-key}& token={access-token}
2. Retrieve the list from the resulting JSON and display.

Boards:

1. Here i am going to retrieve the access token from the database and makes GET request to [https://trello.com/1/members/me/boards?key](https://trello.com/1/members/my/organizations?key)[=](https://trello.com/1/members/my/organizations?key=PropertiesFile){api-key}& token={access-token}
2. Retrieve the list from the resulting JSON and display.

Notify:

1. If user selects ON in the drop down provided to him then we have to create webhooks.

* We can create webhook to any item in the trello account.
* Here i'm creating webhook for member so that single webhook is enough to get all the notifications in the user's account.
* Send a POST request to <https://trello.com/1/tokens/>[{access-token}/ webhooks ?key](https://trello.com/1/members/my/organizations?key)[=](https://trello.com/1/members/my/organizations?key=PropertiesFile){api-key}
* Send JSON body with call-back url(for recieving notifications) and idMember(to which we are creating webhook)
* On success we'll get response code as 200.
* Then notify the user regarding this success.
* Whenever there's an update in user's account, it'll hit the URL provided by us while creating webhook.
* Parse the JSON recieved in servlet and redirect it to corresponding user

1. If user selects OFF in the drop down provided to him then we have to delete webhooks and nullify the webhook-id field in the database

* Send http DELETE request to <https://trello.com/1/webhooks/webhook_id>?key={api-key}&token={access-token}
* Response code 200 resembles success.

1. Extra Details:
2. Using \*\*\*.properties file to store all the details.
3. Using embedded url so that everything will be opened in Teamchat app itself.