Dropbox Integration

**App Type:**

The type of app used was full user dropbox access account. In this we can manage user dropbox content too. Other option was to access just 1 app folder inside user’s dropbox to keep company data. Scopes can be selected as per need.

**For making app:**

For this we just have to select type of app and select the scopes for accessing user contents.

**For publishing app:**

Details required are very basic, Publisher name, Description for the app, Application website, Icon for displaying dropbox app in app market.

**Authentication:**

For accessing a user dropbox account, first the user must login to his dropbox account, then he will be redirected to our application page and receive an secret code from which we can get access code. Then we can make changes to user’s account as per the scope.

**Apis added in our app:**

* Upload File: In this I have created two options. One to save a file at a location directly by our app (will be used when adding meeting files) and other is user chooses where to save the file. For later option dropbox saver is used. This will pop up a dropbox window showing users dropbox.

<a data-filename="" id="dropbox\_saver" class="dropbox-saver">Choose Upload Location</a>

document.getElementById("dropbox\_saver").href=this.result;

document.getElementById("dropbox\_saver").setAttribute('data-filename',document.getElementById("myfile").files[0].name);

The content of file to be uploaded is first read by file reader from local machine and the that is passed as attributes in anchor tag.

* Downloading file: This has same two options as saving the file. But in this the dropbox chooser provides two options which are to directly download the file to users local machine, or to get the file as a link.

var options =

            {

                success: function(files) {

                },

                cancel: function() {

                },

                linkType: "preview", // or "direct"

                multiselect: false, // or true

                //extensions: ['.pdf', '.doc', '.docx'],

                folderselect: false, // or true

                sizeLimit: 10240000000,

            }

* Create folders: In this a group of folder can be mentioned in a single call and accordingly folders will be made.
* body=json.dumps({"paths":["/Workpeer","/Workpeer/Aztlan","/Workpeer/Aztlan/Projects","/Workpeer/Aztlan/Teams","/Workpeer/Aztlan/Misc","/Workpeer/Aztlan/Projects/Meetings","/Workpeer/Aztlan/Teams/Meetings"],"autorename": False,"force\_async": False})
* header={ 'Authorization' : 'Bearer '+ accessToken ,"Content-Type": "application/json"}
* response=rq.post('https://api.dropboxapi.com/2/files/create\_folder\_batch',data = body, headers= header)
* Get all files and folders: This simply returns all the files and folders in a user’s dropbox. The root folder can be set in this query.

body=json.dumps({"path":"/Workpeer","recursive": True,"include\_mounted\_folders": True})

    header={"Content-Type":"application/json" ,'Authorization' : 'Bearer '+ accessToken }

    response=rq.post('https://api.dropboxapi.com/2/files/list\_folder',data=body, headers= header)

**Inviting a member:**

@meetings\_router.post("/dropbox\_add\_member")

async def dropbox\_add\_member(request:Request):

    data = await request.form()

    formdata = jsonable\_encoder(data)

    dat=get\_third\_party\_creds("dropbox")

    accessToken=dat["accessToken"]

    body=json.dumps({"new\_members":[{"member\_email": formdata["member\_name"]}]})

    header={ 'Authorization' : 'Bearer '+ accessToken ,"Content-Type": "application/json"}

    response=rq.post('https://api.dropboxapi.com/2/team/members/add',data = body, headers= header)

    print(response.\_\_dict\_\_)

    return templates.TemplateResponse("dropbox.html",{"request": request})

In this we just have to pass the users email id. After submission, user will have option to merge their personal account and team folder or use separate team and personal account.