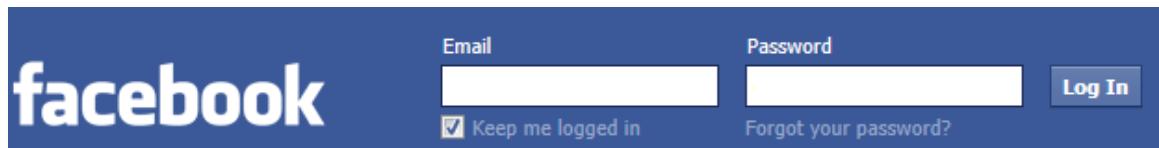


Creating a Facebook desktop application using Facebook SDK and FbGraphApiWrapper (.NET Framework 4, WinForms, Visual Studio)

1. Creating an application-account in Facebook

In order to create an application that communicates with Facebook and acts on behalf of your user, you must first create an application-entity in Facebook:

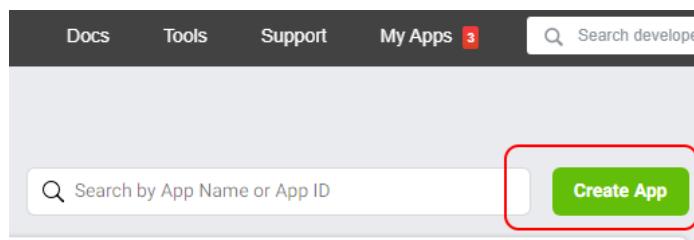
- You must have a Facebook account in order to create an application account which will be



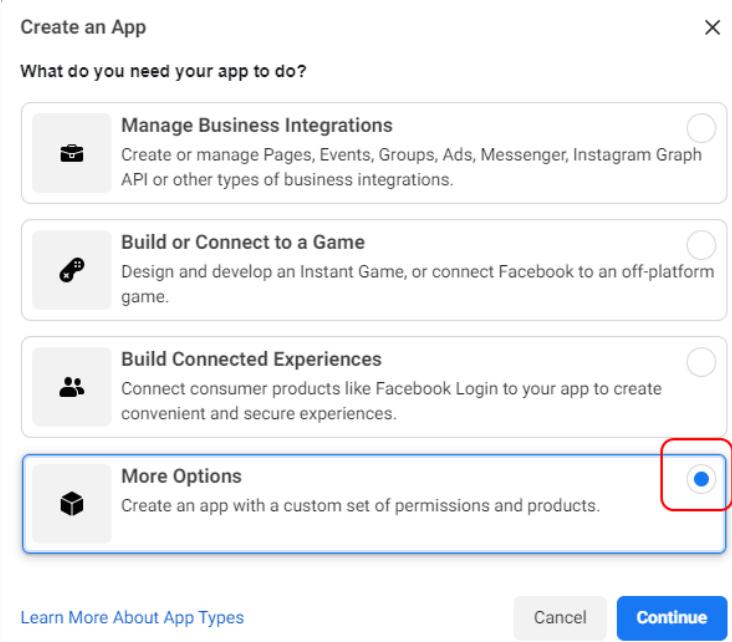
created under your Facebook account.

- Create an Application Account:

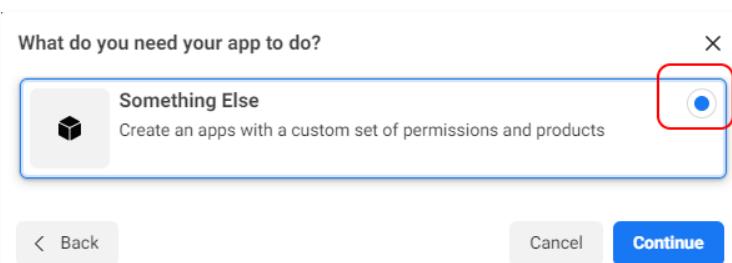
- Go to <https://developers.facebook.com/apps>
- Click the "Create App" button to create a new application account



- Out of the options, select 'More Options':



- Select the 'Something Else' (only) option:



v. Give the application a name using the following format:

DP.011111111.022222222

replace 011111111 with the first student's ID number (9 digits!)

replace 022222222 with the second student's ID number (9 digits!)

and fill in an email address!

Create a New App ID

Get started integrating Facebook into your app or website

Display Name

DP.011111111.022222222

Contact Email

yourEmail@mta.ac.il

By proceeding, you agree to the [Facebook Platform Policies](#)

[Cancel](#)

[Create App ID](#)

c. Hit the '**Set Up**' button in the 'Facebook Login' section:

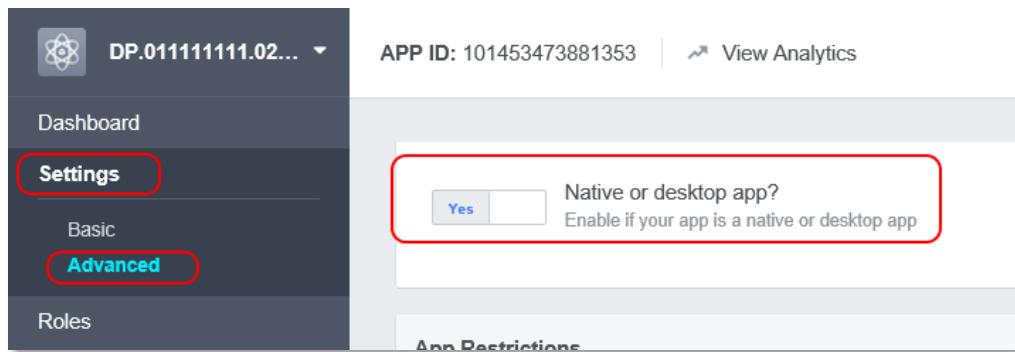
The screenshot shows the Facebook App Dashboard for an app with the ID DP.011111111.02... . The left sidebar has 'Facebook Login' selected under 'PRODUCTS'. A modal window titled 'Select a product' is open, showing the 'Facebook Login' product with a blue icon. Below it, the text reads: 'The world's number one social login product.' There are 'Read Docs' and 'Set Up' buttons at the bottom of the modal.

Click on 'settings' under 'Facebook Login' on the left panel,
and turn on the 'Embedded browser OAuth Login option (choose 'yes')
(And then hit '**Save Changes**')

The screenshot shows the Facebook App Dashboard for the same app. The left sidebar has 'Facebook Login' selected under 'PRODUCTS' and 'Settings' highlighted with a red oval. The right panel shows the 'Client OAuth Settings' section. Under 'Client OAuth Login', there is a 'Yes' button next to a checkbox. A tooltip explains: 'Enables the standard OAuth client token flow. Secure your app by specifying which token redirect URLs are allowed with the options below.' Under 'Web OAuth Login', there is another 'Yes' button next to a checkbox. A tooltip explains: 'Enables web based OAuth client login for building custom login flows. [?]' At the bottom, the 'Embedded Browser OAuth Login' section is shown with a 'Yes' button next to a checkbox, which is highlighted with a red rectangle. A tooltip explains: 'Enables browser control redirect uri for OAuth client login. [?]'.

d. Go to 'Settings' tab (left menu):

- i. In the 'Settings > Advanced' section, select the Native/Desktop application type



- ii. Hit "Save Changes" and go to the 'Roles' tab.

"Roles" is where you can define the "Administrators", "Developers" and "Testers" of your app. Only these users can login through your app until you'll make it available for the general public use:

Add any facebook account to the Administrators/developers/testers list which you want to be able to test your app. Specifically, add the <http://www.facebook.com/design.patterns> user as a tester (you'll need to be 'friends' with desig patter in order to add him as tester).

The screenshot shows the Facebook App Roles interface. The sidebar on the left has 'Dashboard', 'Settings', 'Roles' (which is highlighted), 'Test Users', 'Alerts', and 'App Review'. Under 'PRODUCTS', it shows 'Facebook Login' and '+ Add Product'. The main area has three sections: 'Administrators [?]' with a placeholder for Desig Patter, 'Developers [?]' with a message 'There are no developers for this app.', and 'Testers [?]' with a message 'There are no testers for this app.'

Creating a .NET 4 WinForms application using the FbGraphApiWrapper.dll assembly

- e. In Visual Studio, create a new **.NET Framework 4.0** WinForms project.
- f. From the reference folder of your project, add a reference to the .dll files included in the .zip file of the exercise (FbGraphApiWrapper.dll, Facebook.dll).
- g. Use the static login method

```
LoginResult result = FacebookWrapper.FBService.Login("1450160541956417",
```

providing your AppID and the permissions required from your app's user to display a login form to your user.
If this is the first time your user (a facebook account owner) is using your app, he/she will be prompted to approve the permissions requested by your application.
For the list of permission, see this [link](#).
- h. The return value of the Login method (LoginResult) has a LoggedInUser property (of type FacebookWrapper.ObjectModel.User) which you should use in order to utilize your user's data and actions, in an object-oriented fashion, for example:
 - i. Data:
user.FirstName, user.LastName, user.Birthday, user.RelationshipStatus, etc.
 - ii. Relations to facebook objects:
user.Friends, user.FriendLists, user.Checkins, user.WallPosts, user.Events, user.Albums, user.Pokes, user.Videos, etc.
friend.FirstName, friend.LastName, friend.Albums, friend.Checkins, etc.
album.Photos, checkin.Comments, photo.Comments, photo.Tagged, photo.LikedBy, etc.
 - iii. Actions:
user.PostStatus(), user.PostPhoto(), user.CreateAlbum(), user.CreateFriendList(), etc.
album.UploadPhoto(), photo.Comment(), photo.Like(), status.Comment(), etc.
- i. If the user failed to login or simply closed/canceled the login dialog, the result object will indicate the error with the ErrorMessage property of the LoginResult object.
- j. The return value of the Login method (**LoginResult**) also has a AccessToken property which holds the AccessToken your app got in the Login process. You can save this accessToken for future use (save it to a file/DB) for connecting with facebook in regards to the logged-in user.
Use the static 'Connect' method, the AccessToken you got in the Login process, like such:

```
LoginResult result = FacebookWrapper.FBService.Connect(theAccessToken);
```

result.LoggedInUser will hold the User object with the logged in use data.

2. Resources:

- a. Visit <https://developers.facebook.com/docs/reference/api/> to understand more and get all the information about the Facebook Graph API
- b. Use the <https://developers.facebook.com/tools/explorer> application to browse data on facebook using the Graph API and understanding Jason
- c. The .zip file contains Class Diagrams of the object-oriented wrapper API (.png image files and .cd files which should be viewed in Visual Studio). They are also here on the next pages.
Use them to learn more about the structure of the API (note: These class diagrams are not complete)
- d. The **ReleaseNote - READ ME!!!.txt** file contains interesting information regarding the changes made throughout the different versions of the API. You may find this information useful

