

FB Graph API

Outline

- The FB Graph API
- Authorization with oauth
- Our workaround for not running a webserver
- The facebook module
- Installing the facebook module

FB: A REST API

- But have to send an “authorization key” as a POST parameter
 - Details of POST parameters not important; we will use a module that takes care of that for us.
- <https://graph.facebook.com/?<path>>
- Some useful paths
 - /me gets info about current user
 - /me/feed gets their recent posts
 - /me/photos gets their recent photos
 - /245188182322906 gets info about our facebook group
 - 245188182322906/feed gets recent posts to our facebook group

Facebook's Graph API Explorer

Graph API ExplorerApplication: [?] Graph API Explorer ▾

Access Token: CAACEdEose0cBAHhKwU99kRNU4yblLOXvGaH5kXD3WzZCaDvhvJQ5mqyefJz71hQoka9Eo7dTrefdiVxxMoLgmcHqk0A1C Get Token ▾

Graph API FQL Query

GET ▾ → /v2.3 ▾ /1683212485235313/ ★ Debug Enabled ▾ Submit ▶

[Learn more about the Graph API syntax](#)

Edge: 1683212485235313/

·

```
{
  "description": "This Facebook group is for students in UMSI 106, Fall 2015. Please use it to ask questions, share interesting links, etc. Later in the semester, we will be writing programs that automatically download data from this group and analyze it (e.g., to find whose posts get the most likes).",
  "email": "1683212485235313@groups.facebook.com",
  "icon": "https://static.xx.fbcdn.net/rsrsrc.php/v2/yN/r/068fy4gIIq3.png",
  "name": "SI 106 - Fall 2015",
  "owner": {
    "name": "Jackie Cohen",
    "id": "10203048634357245"
  },
  "privacy": "CLOSED",
  "updated_time": "2015-11-29T14:53:44+0000",
  "id": "1683212485235313"
}
```

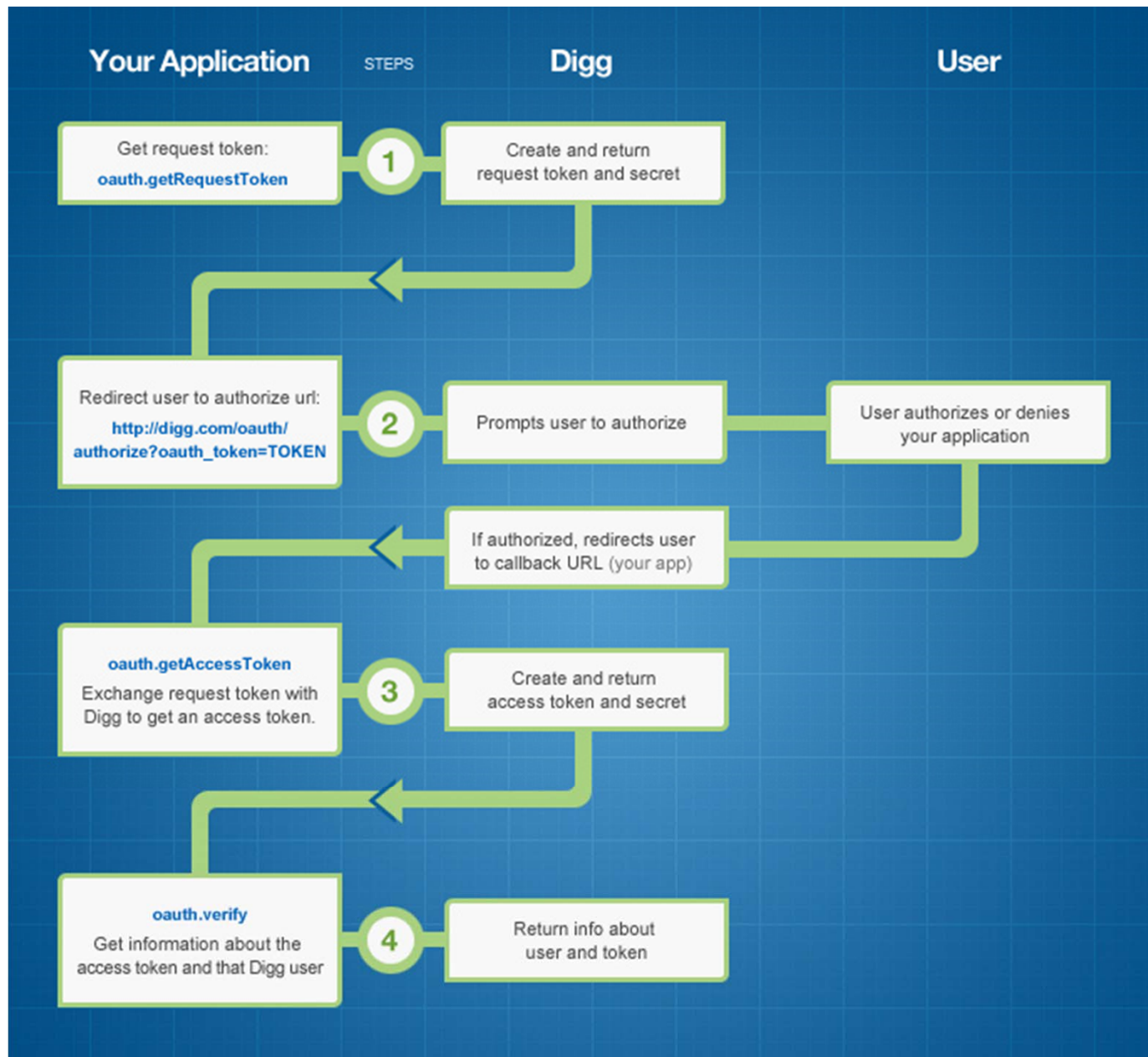
OAUTH

Purpose

- User interacts with site A
 - (or our python program)
- Authentication
 - Prove you own an account on site B
- (limited) Authorization
 - Give site A ability to access things on site B, acting on behalf of user
- Don't reveal your site B password to site A
 - maybe not even username

oAuth Terminology

- User: the person who has to log in
- Consumer (site A)
 - the application that will access data from elsewhere
 - e.g., your code running on your desktop
- Service Provider (site B)
 - The place the user has to log in to
 - e.g., FaceBook, Digg, Twitter



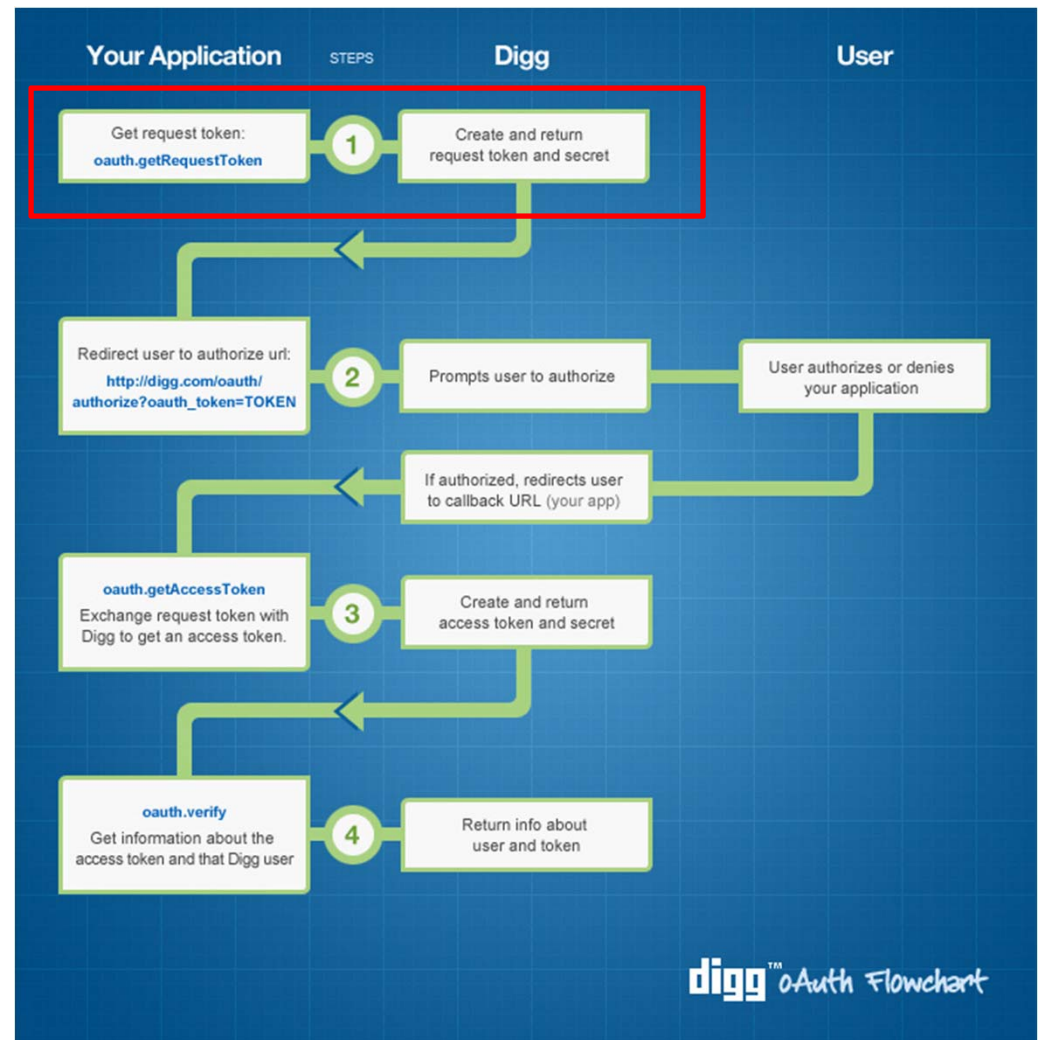
from <http://developers.digg.com/authentication>;
similarly <https://developer.yahoo.com/oauth/guide/oauth-auth-flow.html>

Step 0: User Connects to Consumer

- Consumer decides it needs user to authenticate to Service Provider

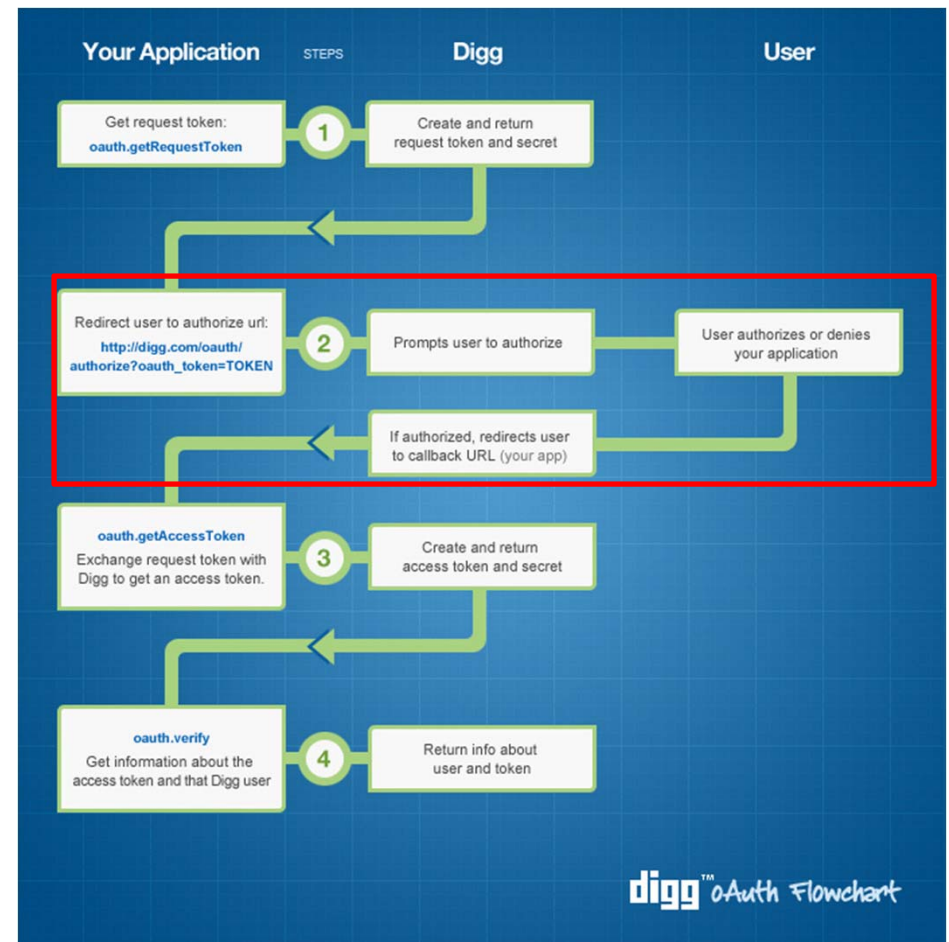
Step 1: Consumer Connects to Service Provider

- Please create and return `oauth_token` and `oauth_secret`



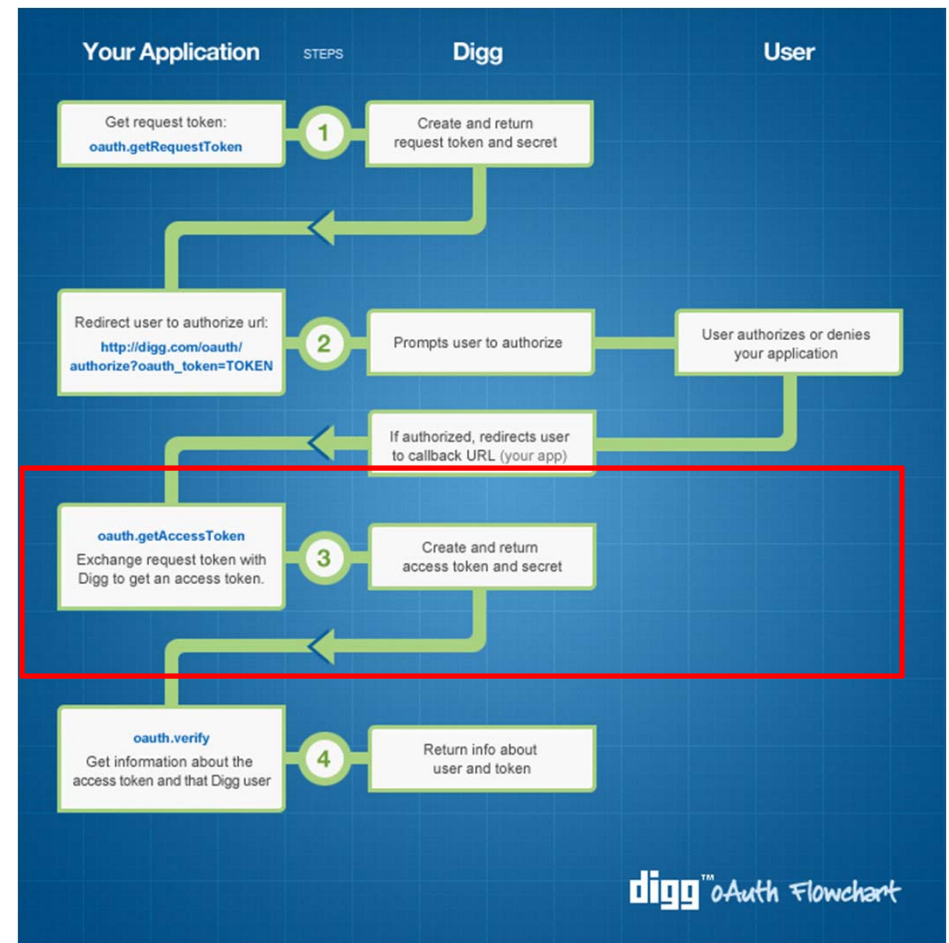
Step 2: User Authenticates to Service Provider

- Redirect User to Digg URL
 - Encode within that Digg URL a URL at your site to redirect to after user authenticates
- When Digg redirects user to your site
 - authorization code also sent as query parameter



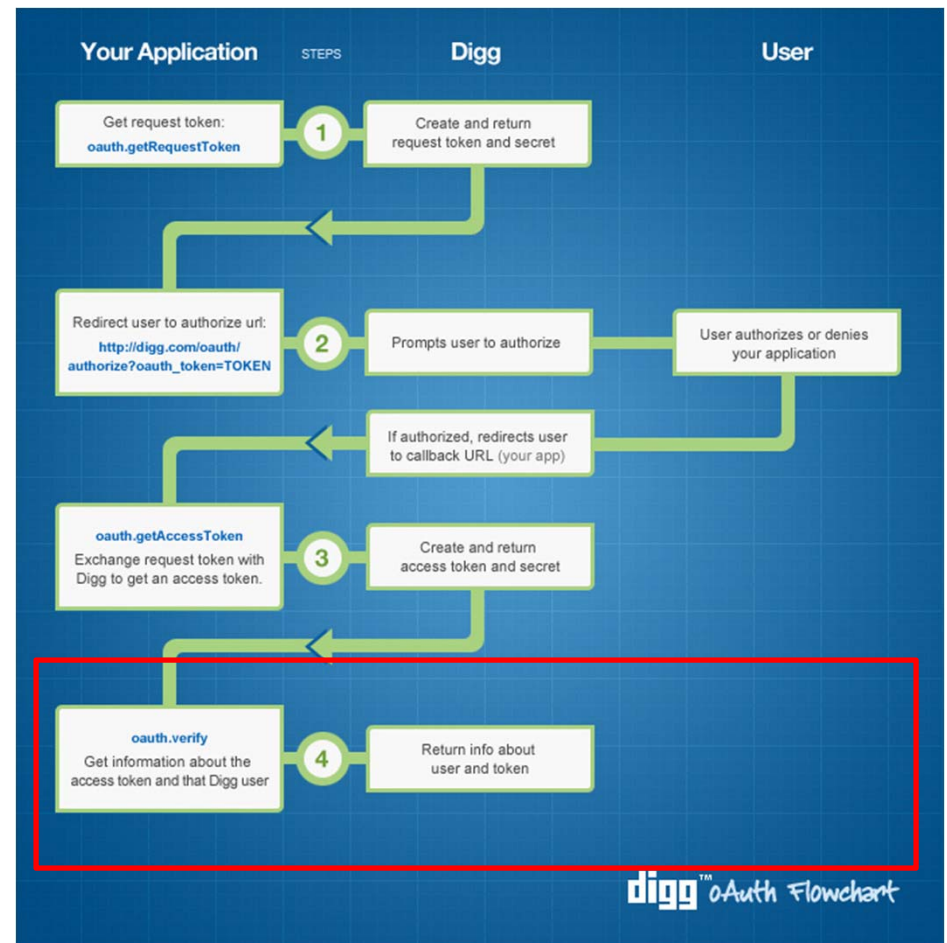
Step 3: Consumer gets a token

- Consumer sends auth code to Service Provider
 - With some info about the Consumer (application)
- Get back an access token



Step 4: use access token

- Consumer sends access token along with requests for data
 - Digg keeps track of access tokens it has issued
 - Digg decides permissions based on who the access token was for



The Problem For Us

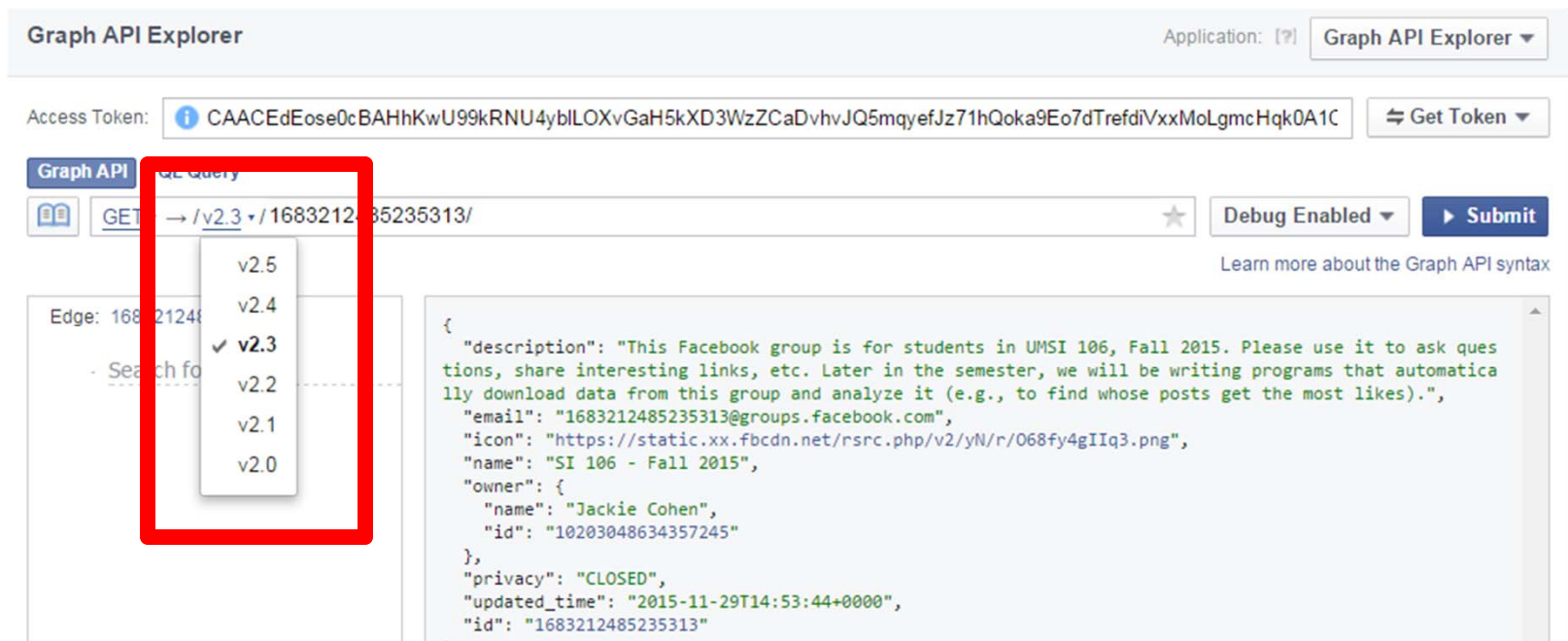
- We are not running a web server
- The step where service redirects user to our site, passing a token, won't work for us

Our Workarounds

- For FB
 - Use the FB developer tool to get the token manually
 - Paste it into your code (or terminal window)
- For Twitter (Wednesday)
 - Open a web browser for the login; copy and paste token from the browser
- For your project
 - Read the API documentation for the particular service
 - If oAuth required, do something similar to what we've done for FB or Twitter

Getting the user_groups permission

- Now deprecated; but available in version 2.3 of the FB API



The screenshot shows the Graph API Explorer interface. At the top, the title is "Graph API Explorer" and the application is set to "Graph API Explorer". Below this, the "Access Token" field contains a long alphanumeric string, and a "Get Token" button is visible. The main section is titled "Graph API Explorer" and features a "GET" button and a URL input field. The URL is set to "/v2.3/1683212485235313/". A dropdown menu is open, showing the following options: v2.5, v2.4, v2.3 (selected with a checkmark), v2.2, v2.1, and v2.0. The dropdown is highlighted with a red rectangle. To the right of the URL field, there is a "Debug Enabled" button and a "Submit" button. Below the URL field, the "Edge" field is set to "1683212485235313". The "Search for" field is empty. The response area on the right shows a JSON object with the following fields: "description", "email", "icon", "name", "owner", "privacy", "updated_time", and "id".

Graph API Explorer

Application: [?] Graph API Explorer

Access Token: CAACEdEose0cBAHhKwU99kRNU4ybILOXvGaH5kXD3WzZCaDvhvJQ5mqyefJz71hQoka9Eo7dTrefdiVxxMoLgmcHqk0A1C Get Token

Graph API Explorer

GET → /v2.3/1683212485235313/ Debug Enabled Submit

Edge: 1683212485235313

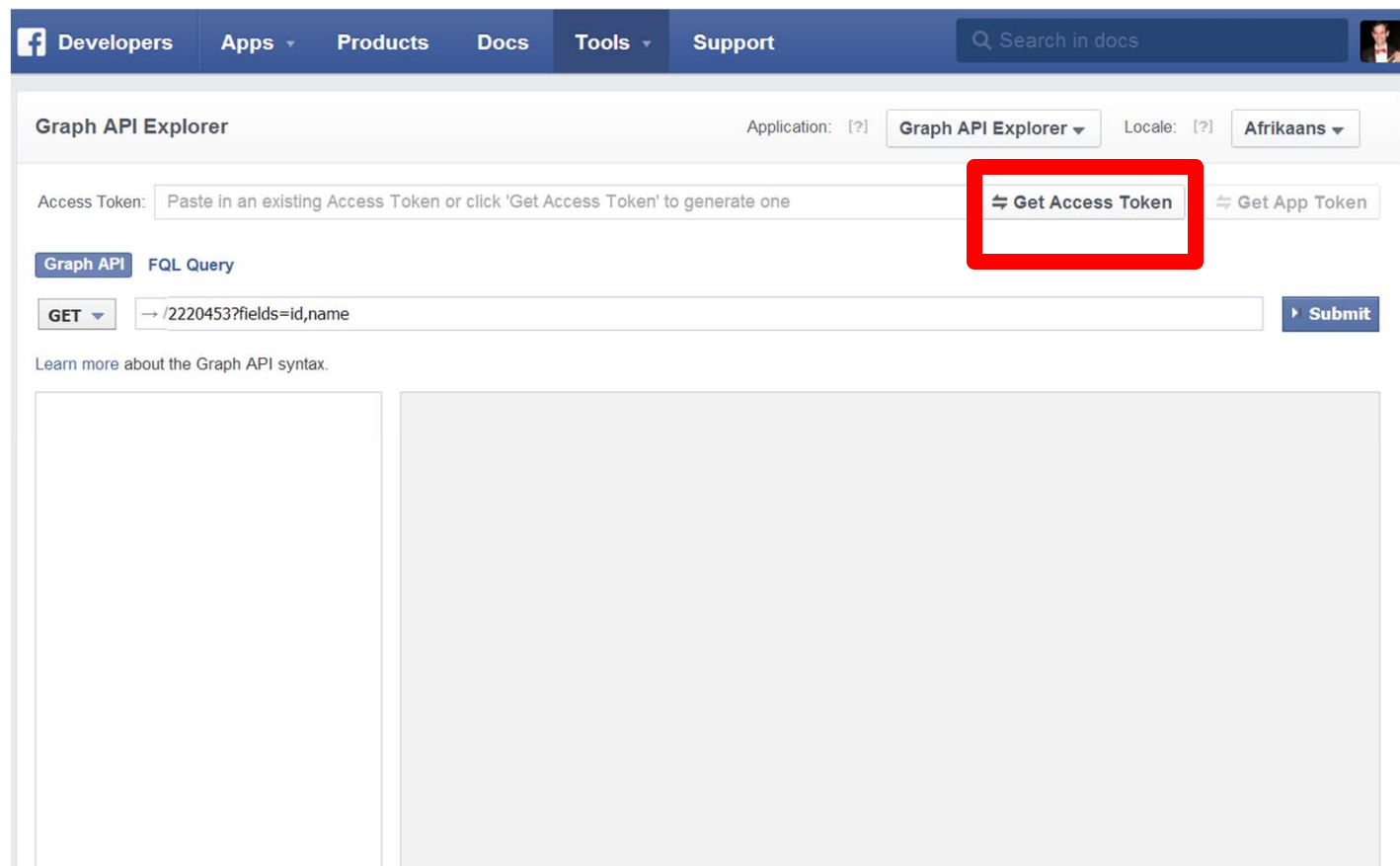
Search for

v2.5
v2.4
✓ v2.3
v2.2
v2.1
v2.0

```
{
  "description": "This Facebook group is for students in UMSI 106, Fall 2015. Please use it to ask questions, share interesting links, etc. Later in the semester, we will be writing programs that automatically download data from this group and analyze it (e.g., to find whose posts get the most likes).",
  "email": "1683212485235313@groups.facebook.com",
  "icon": "https://static.xx.fbcdn.net/rsrc.php/v2/yN/r/068fy4gIIq3.png",
  "name": "SI 106 - Fall 2015",
  "owner": {
    "name": "Jackie Cohen",
    "id": "10203048634357245"
  },
  "privacy": "CLOSED",
  "updated_time": "2015-11-29T14:53:44+0000",
  "id": "1683212485235313"
}
```


Getting the Token

- <https://developers.facebook.com/tools/explorer>



The screenshot shows the Facebook Graph API Explorer interface. At the top, there is a navigation bar with links for Developers, Apps, Products, Docs, Tools, and Support. Below this, the main header includes the application name 'Graph API Explorer', a search bar, and a locale dropdown set to 'Afrikaans'. The 'Access Token' section contains a text input field with the placeholder 'Paste in an existing Access Token or click 'Get Access Token' to generate one'. To the right of this field are two buttons: 'Get Access Token' and 'Get App Token'. The 'Get Access Token' button is highlighted with a red rectangle. Below the 'Access Token' section, there are tabs for 'Graph API' and 'FQL Query'. The 'Graph API' tab is active, showing a 'GET' method and a query path '/2220453?fields=id,name'. A 'Submit' button is located to the right of the query input. At the bottom, there is a link to 'Learn more about the Graph API syntax.' and two large empty rectangular areas for displaying the results of the API call.

AACEdEose0cBANPIHtYcsZCVyEcIqXtTZBF89E0zUKNWHdcPFoZALnbVsdOZAZAsdZB2A0zYNTHdiwerrorPVqQsTO5Sz [Get Token](#)

3 / 1683

35313/

field

Select Permissions

User Data Permissions Extended Permissions

<input type="checkbox"/> user_about_me	<input type="checkbox"/> user_actions.books	<input type="checkbox"/> user_actions.fitness
<input type="checkbox"/> user_actions.music	<input type="checkbox"/> user_actions.news	<input type="checkbox"/> user_actions.video
<input type="checkbox"/> user_birthday	<input type="checkbox"/> user_education_history	<input type="checkbox"/> user_events
<input type="checkbox"/> user_friends	<input type="checkbox"/> user_games_activity	<input checked="" type="checkbox"/> user_groups
<input type="checkbox"/> user_hometown	<input type="checkbox"/> user_likes	<input type="checkbox"/> user_managed_groups
<input checked="" type="checkbox"/> user_managed_groups	<input type="checkbox"/> user_photos	<input checked="" type="checkbox"/> user_posts
<input type="checkbox"/> user_relationship_details	<input type="checkbox"/> user_relationships	<input type="checkbox"/> user_religion_politics
<input checked="" type="checkbox"/> user_status	<input type="checkbox"/> user_tagged_places	<input type="checkbox"/> user_videos
<input type="checkbox"/> user_website	<input type="checkbox"/> user_work_history	

Public profile included by default.

[Get Access Token](#) [Clear](#) [Cancel](#)

ug Enabled [Submit](#)

more about the Graph API syntax

please use it to ask ques
programs that automatica
the most likes).",

Cut and paste the access token

Graph API Explorer Application: [?] Graph API Explorer ▾ Locale: [?]

Access Token: CAACEdEose0cBAAEdQFNKO7Mox4ADyGzzdtnhnIHWtrQSI4RD2Nah51wLN1C1a ✕ Debug ⇌ Get Access Token

Graph API FQL Query

GET ▾ → /me

[Learn more about the Graph API syntax.](#)

See `fboauth.py` for sample code

The facebook-sdk module

- Defines class GraphAPI
- `__init__` takes an auth key
- Method `get_object()` takes a string
 - String is in format that FB defines
 - `"/me"` gets info about current user
 - `"/me/feed"` gets their recent posts
 - `limit` parameter says how many items to retrieve
 - `fields` parameter says which fields of the items to retrieve
 - `get_object` method does a lot of stuff behind the scenes
 - Calls `requests.get`
 - Passes the `auth_key` in the way FB wants
 - Gets the response
 - Calls `json.loads` on it
 - Returns a dictionary

But You Have To Get Your Computer Prepped to Use It

- Install the facebook-sdk module
- Install pip [already done]
- Use pip to install the requests module [already done]

Use pip to Install requests (should have done this already)

- On Mac
 - `pip install requests`
- On Windows
 - `/c/Python27/Scripts/pip install requests`
- Note: if you get permission errors during installation, try running the install commands with a prefix of `sudo` (super user do)
 - E.g., `sudo pip install requests`

Installing the FB module

- `pip install facebook-sdk`
- On Windows
 - `/c/Python27/Scripts/pip install facebook-sdk`
- Note: if you get permission errors during installation, try running the install commands with a prefix of `sudo` (super user do)
 - E.g., `sudo pip install facebook-sdk`

Now fboauth.py Should Run!

- Note that the `get_object()` method does a lot of things that we've done manually before
 - Calls `requests.get()`
 - Reads the response as text
 - Calls `json.loads()`
 - What you get back is a python dictionary

Exercises

- Print out the text of all the messages returned in the feed
- Print out the names of everyone who left a comment on any of the posts