Facebook Friends Prototype
Graph API v5

October 16, 2018

Table of Contents

1. Introduction

This prototype uses the Facebook Graph API to do the following:
- Allow a user to login using Facebook credentials.
- Find the friends of the current logged in user that also installed this App.
- Finds mutual friends between the current user and a fixed reference base user.

The Reference Base User is an internal test user called "Open Graph Test User". The name, id, and access token information is stored in a SQL Database. The access token is a long-lived token, which means it will expires in December.

2. Development environment

The prototype was developed in a local environment with the following configuration:
- PHP v7.2.9
- Facebook SDK v5 for PHP
- Nginx web server v1.15
- HTTPS enabled required by FB
- MySQL Community Server v8.0.12
- Visual Studio Code

3. Folder structure

The folder structure can be seen in Fig. 1. Also, files used in this project are shown.

.ebextensions
Hidden folder for configuration files used for customized deployment to AWS Elastic Beanstalk (EB) or Elastic Computing Cloud (EC) environments.

.vscode
Hidden folder for configurations files used by Visual Studio Code (VSC).

Certs
Folder that stores the private and public keys certificates to enable HTTPS protocol for the web application. This certificates were generated for a localhost deployment. For other environments the certificates should be generated again.

Public

Folder for resources that have public access. In a typical web application, this folder would be used to store images, javascripts, styling and other public files. For this project, only the index.php is contained in this folder
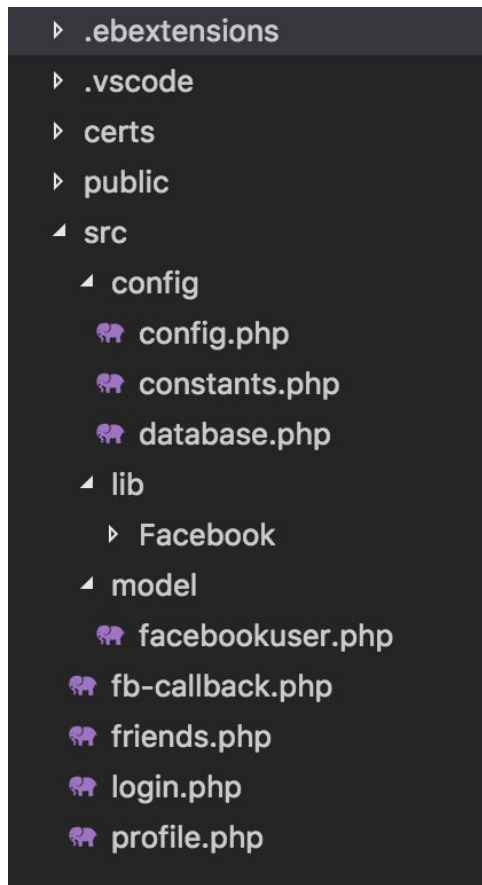

Fig. 1: Folder structure and files for this project

Lib

This folder stores the libraries used in the project. At this moment, only the Facebook SDK for PHP library is included in this prototype, under the Facebook subfolder.

Src

This folder contains the code for the project. The following sections will detail about:

- Configuration files
- Model and data access
- Page code
- Mutual Friends high level algorithm

4. Configuration files

a. Constants.php

It defines the following constants:
- The application ID.
- The application secret.
- Graph API version.
- URI for the callback method for the login process.
- The user ID for the reference base user.
- Number of records per page that the API will return.

b. Database.php

It defines the Database class. This class contains the method for connecting to a MySQL database, using the PHP Database Object (PDO) library, and the following constants:
- Host name
- Database name
- Database user
- User password

c. Config.php

This file define and create the variables that will be used in many other files:
- Defines the constants mentioned in constants.php.
- Initiates a web session for the connecting user.
- Includes support for the Facebook SDK files.
- Creates the Facebook object to the used in the project.

5. Model and Data Access

This folder contains the facebookuser.php file. This file defines the FacebookUser class, which is used to store user ID, name, and access token information. It also defines two methods:
- Get all Facebook user saved in the database
- Get one Facebook user identified by user ID

Both methods use configurations in database.php and the PDO library as a Data Access Layer to interface with MySQL database.

The code used for the table creation is:

```
CREATE TABLE `facebookuser` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `userid` varchar(256) NOT NULL,
  `accesstoken` varchar(4096) NOT NULL,
  `name` varchar(256) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `userid_UNIQUE` (`userid`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
```

And the contents of this table for this project are:

id	userid	accesstoken	name
▶ 1	120932128882133	EAAKcvqehiQ0BAJPR9fZByfkWvqwYHA7GBZ...	Open Graph Test User
NULL	NULL	NULL	NULL

6. Index.php

It is the root document for the application. It is just a public HTML page that redirects to the Login.php file.

7. Login.php

This code file defines the permissions used during the login. This permissions will establish the access to specific user's information. The access level and accessible data will be coded into the access token hash. For the current prototype, the permissions are:

- Email
- User_friends

It is important to point out that the user_friends permission is commented in the code. To uncomment this permission, the application needs to go through the Facebook App Review process. This happens after March 2018 when a severe security issue was found in Facebook API.

The prototype is able to find friends because the application is using internal test users that are defined in developer.facebook.com website for our application. This internal

users allow to access their friends lists without the user-friends permission and without the Facebook App review process. To be able to use the application with real facebook users, the application must be approved by Facebook using the App Review process.

Also, this code defines the callback routine to be executed after the login process. This routine is the contact point where the Facebook website transfers control to our application.
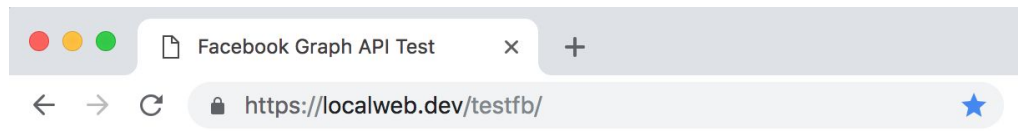Finally, it shows a link for the user to start the login process.

Fig. 2: Page that the user sees after the application is launched

8. Fb-callback.php

This file is executed when the login process has returned from the Facebook website. At this point our project takes control of the user information. It performs the following actions:
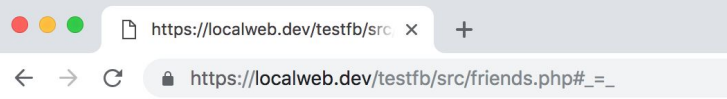- Uses the Facebook object to get a login helper object, to contact Facebook API.
- Uses the login helper object to retrieve the access token for the current logged in user.
- Verifies if the following exceptions events have occured:
  - FacebookResponseException
  - FacebookSDKException
- Transforms the default short-lived access token to a long-lived version.
- Saves the user token information to the web user session.
- Redirects the application to friends.php.

9. Friends.php

This file contains the logic to find the mutual friends between the fixed reference user and the current logged in user. It performs the following actions:

- Connects to MySQL database.

6

- Retrieves id and access token for the fixed reference user from MySQL database.
- Uses the token to get friends for the reference user and stores them in an array.
- Gets current logged in user access token from the user web session.
- Uses the token to get friends for the current user and stores them in an array.
- Verifies if the following exceptions events have occured:
  - FacebookResponseException
  - FacebookSDKException
- Pages results are merged into a single array and sorted in ascendent fashion. A simple algorithm to find intersection of both arrays is used.
- The result is shown in the browser.


```
BASE USER: Open Graph Test User

Array
(
    [0] => Array
        (
            [name] => Linda Albibgfgcejdd Greeneson
            [id] => 102826897369868
        )

    [1] => Array
        (
            [name] => Bill Albiaiehbhbce Moiduman
            [id] => 105154010467793
        )

    [2] => Array
        (
            [name] => Mike Albiagaidhied Bharambeescu
            [id] => 105234350458963
        )

    [3] => Array
        (
            [name] => Nancy Albiadijcjaij Putnamsen
            [id] => 106587440322890
        )

    [4] => Array
        (
            [name] => Mary Albijhfghgbef Laverdetman
            [id] => 112320869747472
        )

)

CURRENT USER: Robert Argume

Array
(
)

NUMBER OF COMMON FRIENDS = 0

Array
(
)
```

Fig. 3: The result that is shown to the current logged in user. In this case, there are no mutual friends.

10. Profile.php

This file was used to test the login process and get the current logged user profile information. It retrieves the following information:

- Id
- Name
- Email
- Cover
- Picture
- Verifies if the following exceptions events have occured:
    - FacebookResponseException
    - FacebookSDKException