Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Московский государственный технический университет имени Н.Э. Баумана

(национальный исследовательский университет)»

**(МГТУ им. Н.Э.Баумана)**

ФАКУЛЬТЕТ \_\_«Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА\_\_\_\_\_\_\_«Защита информации»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ***

***НА ТЕМУ:***

Автоматическое заполнение профиля социальной сети Facebook.

Студент \_\_ИУ10-81\_\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Анисимов М.С.** (Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель практики **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Астрахов А.В.**  (Подпись, дата) (И.О.Фамилия)

*Москва, 2017 г.*

**Содержание**

[Цель и задачи 2](#_Toc488070574)

[Введение 4](#_Toc488070575)

[Исследовательская часть 5](#_Toc488070576)

[Практическая часть 6](#_Toc488070577)

[Начало работы 6](#_Toc488070578)

[Настройка Apache 2.4 6](#_Toc488070579)

[Настройка PHP 7](#_Toc488070580)

[Получение ID приложения, секрета и маркера доступа к странице 8](#_Toc488070581)

[Выводы 11](#_Toc488070582)

[Литература 12](#_Toc488070583)

[Приложение 1 13](#_Toc488070584)

[Приложение 2 17](#_Toc488070585)

# **Цель и задачи**

**Цель практики**: Реализовать автоматическое заполнение страницы профиля в социальной сети Facebook.

**Задачи практики**:

* Изучить взаимодействие API Facebook и PHP;
* Создать «скрипты», выполняющие заполнение профиля по следующим ключевым параметрам:
  + Ссылки на странице профиля;
  + Фотографии;
  + Видео;
  + Сообщения для других пользователей;
* Написать инструкцию по использованию скрипта, включающего методы, позволяющие реализовать заполнение внутреннего содержания страницы пользователя социальной сети Facebook.

# **Введение**

Facebook— крупнейшая социальная сеть в мире. Facebook позволяет создать профиль с фотографией и информацией о себе, приглашать друзей, обмениваться с ними сообщениями, изменять свой статус, оставлять сообщения на своей и чужой «стенах», загружать фотографии и видеозаписи, создавать группы (сообщества по интересам). В 2007 году Facebook предложил сторонним программистам создавать приложения.

Facebook предлагает множество функций, с помощью которых пользователи могут взаимодействовать между собой. Среди наиболее популярных — виртуальное подмигивание, фотоальбомы и «стена», на которой знакомые пользователя могут оставлять *текстовые* и *аудиовизуальные сообщения*. И именно эти варианты взаимодействия с пользователями и будет интересовать нас в практике.

# **Исследовательская часть**

Для взаимодействия со страницами Facebook используется API Graph. **API Graph** — это основной инструмент для получения и ввода данных на платформе Facebook. Это низкоуровневый API на основе HTTP, с помощью которого можно программно запрашивать данные, создавать публикации, управлять рекламой, загружать фото и выполнять множество других задач в приложении. API Graph основан на протоколе HTTP и работает с любым языком, в котором есть библиотека HTTP (например, cURL, urllib). Общий принцип: для чтения данных из API необходимо отправлять HTTP-запросы GET на узлы или границы между этими узлами.

Запросы, отправленные в API, передаются по адресу graph.facebook.com — кроме запросов на загрузку видео, для которых используется адрес graph-video.facebook.com.

Первое, с чего нужно начинать при разработке приложения для Facebook такого рода – получение маркера доступа к странице (Access Page Token) с помощью API Graph Explorer. В дальнейшем, полученный маркер доступа, вместе с ID приложения и его секретом, интегрируется в код самого приложения и только после этого пишутся основные скрипты взаимодействия со страницей пользователя Facebook. Из-за нелегальности приложения и из-за того, что прошло очень большое количество запросов через graph.facebook.com, страница пользователя может быть «забанена» до разбора обстоятельств => требуется осторожность при выполнении скрипта. Разберем последовательность действий и код скрипта.

# **Практическая часть**

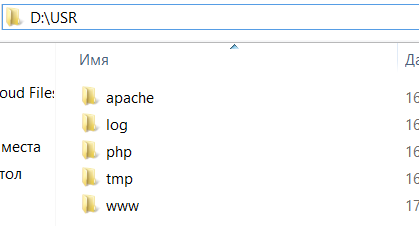
Практическая часть будет описана в виде пошаговой инструкции по настройке устройства реализации скрипта автоматического заполнения профиля Facebook. Скрипт проверялся под ОС Windows 8.1 и Windows 7 Professional.

## **Начало работы**

Для создания скрипта использовался сервер Apache 2.4 (bit.ly/APACHE24) и PHP 5.6 (bit.ly/PHP5600).

## **Настройка Apache 2.4**

* Переходим по ссылке и загружаем архив с Apache 2.4;
* Распаковываем содержимое папки Apache 2.4 в заранее заготовленную папку для сервера (В нашем случае в папку D:\USR\apache)



* Редактируем файл \conf\httpd.conf:
* ServerRoot "d:/USR/apache"
* DocumentRoot "d:/USR/www"
* Direcroty "d:/USR/www"
* ServerName localhost:80
* ErrorLog "D:/USR/log/apache-error.log"
* CustomLog "D:/USR/log/apache-access.log" common
* От Администратора запускаем cmd и вводим D:\USR\apache\bin\httpd.exe -k install
* Запускаем ApacheMonitor. Apache2.4 -> Start.

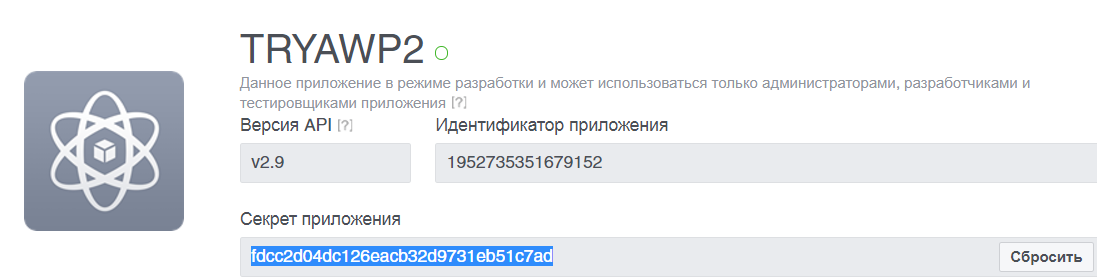
## **Настройка PHP**

* + Переходим по ссылке и загружаем архив с PHP 5.6;
  + Извлекаем содержимое архива в \php;
  + В файл apache\conf\httpd.conf добавляем строки:
  + LoadModule php5\_module "d:/USR/php/php5apache2\_4.dll";
  + AddHandler application/x-httpd-php .php;
  + # Путь к файлу php.ini;
  + PHPIniDir "D:/USR/php";
  + DirectoryIndex index.php ;
  + Используя ApacheMonitor: Apache24 -> Restart;
  + В браузере вводим http://localhost/index.php и убеждаемся, что PHP работает;
  + Делаем копию файла php\php.ini-development с именем php\php.ini;
  + Пользуясь поиском находим, раскомментируем и изменяем параметры:
  + extension\_dir = "D:/USR/php/ext"
  + sys\_temp\_dir = "D:/USR/tmp"
  + extension=php\_mysql.dll
  + extension=php\_mysqli.dll
  + extension=php\_openssl.dll
  + date.timezone = Europe/Moscow
  + ApacheMonitor: Apache24 -> Restart;
  + Создаем папку fbsdk в www;
  + Скачиваем <https://github.com/facebook/php-graph-sdk> и распаковываем туда содержимое папки Facebook в архиве;
  + Переходим по ссылке <https://getcomposer.org/Composer-Setup.exe> и устанавливаем приложение;
  + В cmd выполняем: composer require facebook/graph-sdk;

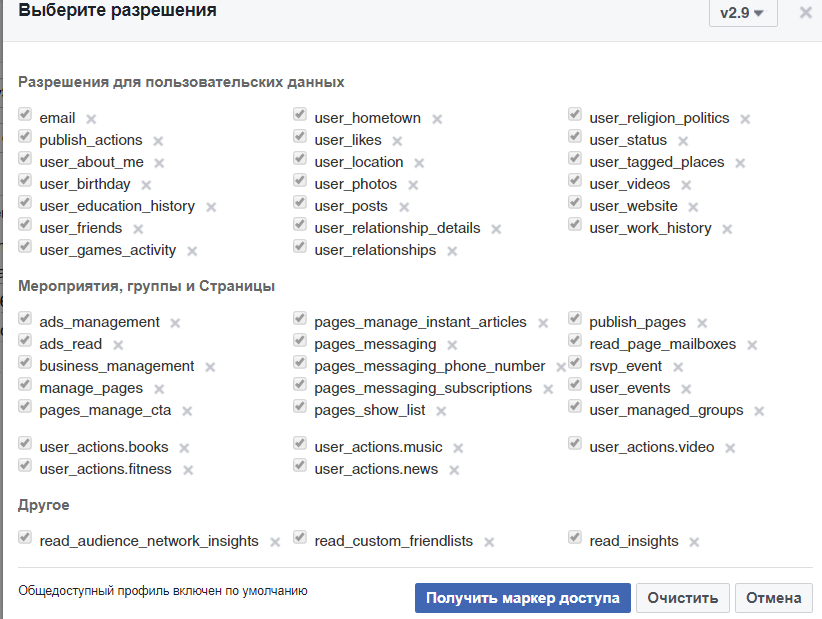
Настройка PHP для взаимодействия с Facebook API завершено. Теперь можно писать приложение, но для начала нужно получить маркеры доступа к странице у уже имеющегося приложения.

## **Получение ID приложения, секрета и маркера доступа к странице**

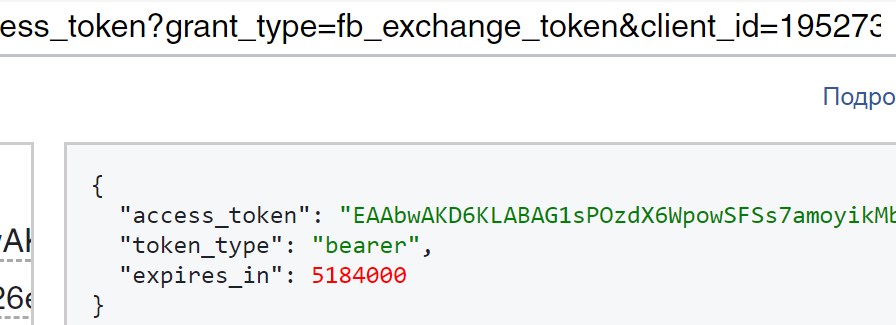
После создания приложения в developers.facebook открывается панель управления приложением, где находятся ID приложения и его секрет. Оба параметра пригодятся для получения маркера доступа к странице для приложения.



Для получения маркера доступа необходимо открыть API Graph Explorer (<https://developers.facebook.com/tools/explorer/>), в графе «Получить маркер» выбрать «Получить маркер доступа пользователя», в выпадающем меню указать те пункты, с которыми приложение будет взаимодействовать и нажать OK.



Затем выбрать «Получить маркер доступа к Странице», скопировать его и в строке запроса GET ввести команду oauth/access\_token?grant\_type=fb\_exchange\_token&client\_id=[ID\_ПРИЛОЖЕНИЯ]&client\_secret=[СЕКРЕТ\_ПРИЛОЖЕНИЯ]&fb\_exchange\_token=[Маркер\_доступа\_к\_странице], заменив [ID\_ПРИЛОЖЕНИЯ],[СЕКРЕТ\_ПРИЛОЖЕНИЯ],[Маркер\_доступа\_к\_странице] на соответствующие приложению параметры. Затем нажать отправить и в окне обработки запроса будет необходимый маркер доступа, который и интегрируется в код скрипта.



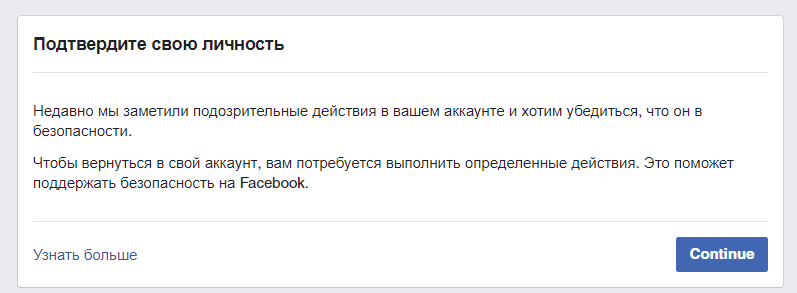
Далее можно писать сам скрипт на PHP, используя стандартные методы и запросы API Graph. Через API Graph можно реализовать следующее заполнение профиля:

* Получить полное имя пользователя (name) и его ID (id);
* Получить то, что пользователь отметил как понравившееся (узел Страницы);
* Получить два события, на которые пользователь был приглашен (узлы События);
* Создать сообщение на основе данных, полученных из 3 запросов выше, и опубликовать его в Хронике пользователя;
* Получить два фото от пользователя;
* Публикация ссылок и обычных сообщений (постов);
* Получение профиля пользователя;
* Загрузка фото в профиль пользователя;
* Загрузка видео в профиль пользователя;
* Загрузка «Аватар» в профиль пользователя;

В ***Приложении 1*** приведен пример кода скрипта для ускоренного заполнения профиля facebook, а в ***Приложении 2*** показан результат работы скрипта на самой странице пользователя.

# **Выводы**

В практической работе реализован скрипт по автоматизированному заполнению профиля Facebook, однако при его использовании следует учитывать, что при исполнении, так как приложение не авторизовано официально, страница пользователя может быть «забанена» на неопределенный срок до проверки обстоятельств.



# **Литература**

1. <https://developers.facebook.com/docs/php/gettingstarted> - Начало работы с Facebook SDK для PHP;
2. <https://developers.facebook.com/docs/php/howto/example_facebook_login> - «Вход через Facebook»;
3. <https://developers.facebook.com/docs/graph-api/using-graph-api> - API Graph
4. <https://getcomposer.org/doc/01-basic-usage.md> - Использование composer и толкование того, что это такое;

# **Приложение 1**

<?php

include 'fbsdk/autoload.php';

//---------ACCESSTOKENPROVIDE--------------------//

// Как я понял, подразумевается, что уже извесен отсюда https://developers.facebook.com/tools/explorer

// маркер доступа к странице. В дальнейшем банально или кодом можно, или выполняется там же в Explorer команда:

//oauth/access\_token?grant\_type=fb\_exchange\_token&client\_id=[ID ПРИЛОЖЕНИЯ]&client\_secret=

//[секрет приложения]&fb\_exchange\_token=[МАРКЕР ДОСТУПА К СТРАНИЦЕ]. Ну а дальше понятно все.

//---------------------------------------------//

$fb = new Facebook\Facebook([

'app\_id' => '897063643765554',

'app\_secret' => '7f6aa922e415b4968b9b459ab8a4ae98',

'default\_graph\_version' => 'v2.9',

]);

$accessToken = 'EAAMv3ZBfT2zIBADsJQv9fbzM0i0J1Jp7ylXIsDwQfPp3CCPSyXTKkdMZC7qSx6GtGTq1BboI8DkjHZCYtrff94qYNQUzeQJxFVQApzUgAs7JaGW1SURJK1nZAYxhxU5DCPxP5GmNRAuYX8RuZAoO0dv59TOkOgbOzgTB4vZCxZBEUzKGqC4ucUT';

//------------------ССЫЛОЧКИ НА КАНАЛЬЧИКИ-----------------//

$linkData = [

'link' => 'http://www.twitch.tv/happasc2',

'message' => 'HAPPA В ТЕЛЕВИЗОРЕ!!!',

];

try {

// Returns a `Facebook\FacebookResponse` object

$response = $fb->post('/me/feed', $linkData, $accessToken);

} catch(Facebook\Exceptions\FacebookResponseException $e) {

echo 'Graph returned an error: ' . $e->getMessage();

exit;

} catch(Facebook\Exceptions\FacebookSDKException $e) {

echo 'Facebook SDK returned an error: ' . $e->getMessage();

exit;

}

$graphNode = $response->getGraphNode();

echo 'Posted with id: ' . $graphNode['id'];

$linkData = [

'link' => 'https://www.twitch.tv/sinedd92',

'message' => 'А ЭТО СИНЕДД В ТЕЛЕВИЗОРЕ!!!',

];

try {

// Returns a `Facebook\FacebookResponse` object

$response = $fb->post('/me/feed', $linkData, $accessToken);

} catch(Facebook\Exceptions\FacebookResponseException $e) {

echo 'Graph returned an error: ' . $e->getMessage();

exit;

} catch(Facebook\Exceptions\FacebookSDKException $e) {

echo 'Facebook SDK returned an error: ' . $e->getMessage();

exit;

}

$graphNode = $response->getGraphNode();

echo 'Posted with id: ' . $graphNode['id'];

//------------------ФОТКА-----------------//

$data = [

'message' => 'МАМА ЭТО Я',

'source' => $fb->fileToUpload('C:/Users/Lianel/Desktop/Main/All/luls.png'),

];

try {

// Returns a `Facebook\FacebookResponse` object

$response = $fb->post('/me/photos', $data, $accessToken);

} catch(Facebook\Exceptions\FacebookResponseException $e) {

echo 'Graph returned an error: ' . $e->getMessage();

exit;

} catch(Facebook\Exceptions\FacebookSDKException $e) {

echo 'Facebook SDK returned an error: ' . $e->getMessage();

exit;

}

$graphNode = $response->getGraphNode();

echo 'Photo ID: ' . $graphNode['id'];

//------------------ВИДЕО-------------------//

$data = [

'title' => 'ХАЛАТИКИ ПОДЪЕХАЛИ',

'description' => 'БЕСПАЛЕВНАЯ ИДЕЯ',

'source' => $fb->videoToUpload('C:/Users/Lianel/Desktop/vandam.mp4'),

];

try {

$response = $fb->post('/me/videos', $data, $accessToken);

} catch(Facebook\Exceptions\FacebookResponseException $e) {

// When Graph returns an error

echo 'Graph returned an error: ' . $e->getMessage();

exit;

} catch(Facebook\Exceptions\FacebookSDKException $e) {

// When validation fails or other local issues

echo 'Facebook SDK returned an error: ' . $e->getMessage();

exit;

}

$graphNode = $response->getGraphNode();

var\_dump($graphNode);

echo 'Video ID: ' . $graphNode['id'];

?>

# **Приложение 2**

