

Выгрузка фотографий с личной страницы в социальной сети «ВКонтакте» с помощью VK API

Маринчук Александр Сергеевич

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

Web API это довольно полезный способ взаимодействия с той или иной системой, который позволяет программистам без труда работать с нужными методами. Большинство популярных сервисов давно реализовало и использует в своем арсенале API, упрощая жизнь программистам, которые вплотную работают с данными системами. В данной статье описан способ выгрузки фотографий с личной страницы в социальной сети «ВКонтакте» с помощью VK API.

Ключевые слова: социальная сеть, фотографии, выгрузка, VK API, Python.

Uploading photos from a personal page on the VKontakte social network using the VK API

Marinchuk Alexander Sergeevich

Sholom-Aleichem Priamursky State University

Student

Abstract

Web API is a pretty useful way of interacting with a particular system, which allows programmers to easily work with the desired methods. Most of the popular services have long been implemented and are using APIs in their arsenal, making life easier for programmers who work closely with these systems. This article describes a method for uploading photos from a personal page on the VKontakte social network using the VK API.

Keywords: social network, photos, upload, VK API, Python.

1. Введение

1.1 Актуальность исследования

Web API это довольно полезный способ взаимодействия с той или иной системой, который позволяет программистам без труда работать с нужными методами. Большинство популярных сервисов давно реализовало и использует в своем арсенале API, упрощая жизнь программистам, которые вплотную работают с данными системами.

У каждого сервиса, имеющего API, существует необходимая документация, описывающая правила взаимодействия с данной системой.

API может использоваться по разному: выгрузка фотографий, возврат списка друзей, добавление поста, просмотра аудиозаписей и многое другое. В данной статье описан способ выгрузки фотографий с личной страницы в социальной сети «ВКонтакте» с помощью VK API и языка программирования Python.

1.2 Обзор исследований

В статье С. С. Щедрина описано создание приложения, которое реализовывает синхронизацию баз данных университета, используя API информационного портала [1]. Создала генератор отчетов о посещаемости сайта в формате Microsoft Office Excel с помощью API Яндекс.Метрика в своем исследовании И. Чернова [2]. Рассмотрены способы обработки запросов для чат-бота при помощи инструментов VK API в работе Д. И. Бикова [3]. В статье А. Н. Крайникова и А. А. Банникова рассматривается пример того, каким образом можно управлять пинами ввода/вывода на микрокомпьютере raspberry pi с помощью VK API, тем самым имитирую концепт «интернет вещей» [4]. А. Ю. Баранов разработал мобильное приложение для ведения заметок с интеграцией VK API в своем исследовании [5]. Г. С. Сейдаметов и Ш. М. Усеинов рассмотрели взаимодействие с API VK для загрузки медиа-контента на сервера VK (ВКонтакте - социальная сеть). В статье также представлена последовательность реализации работающего кода, который можно использовать для добавления фотографий на стену, в альбомы или же в диалоги [6]. В работе Н. С. Ампилогова рассмотрена структура API социальных сетей, на примере vk.com и facebook.com. Описаны основные правила запросов и получаемых ответов от сервисов [7].

1.3 Цель исследования

Целью данной статьи является описание способа выгрузки фотографий с личной страницы в социальной сети «ВКонтакте» с помощью VK API и языка программирования Python.

2. Методы исследования

В данной статье будет рассмотрен способ выгрузки фотографий с личной страницы в социальной сети «ВКонтакте» с помощью VK API и языка программирования Python [8-9].

API (программный интерфейс приложения) - описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой [10].

В рамках данного исследования будут выгружаться все фотографии с личной страницы. Для начала следует создать Standalone-приложение в ВК для чего надо перейти во вкладку «Управление» и нажать кнопку «Создать» (Рис. 1).

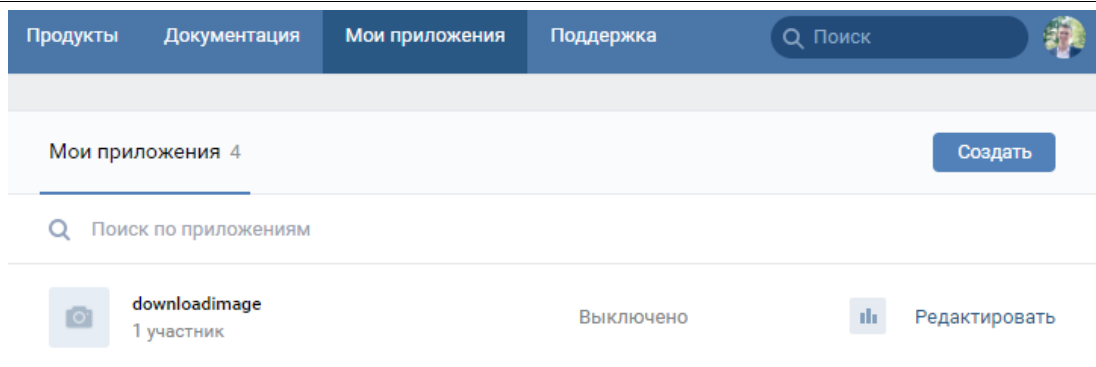


Рисунок 1 – Создание приложения

Далее следует перейти в настройки созданного приложения и скопировать его ID (Рис. 2).

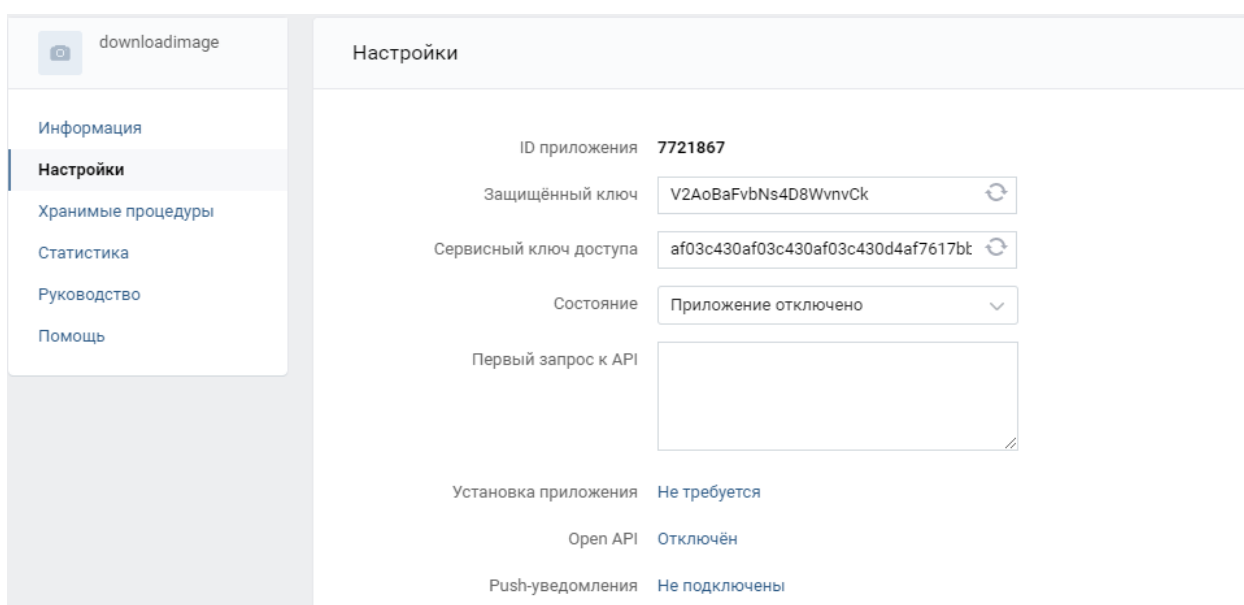


Рисунок 2 – Настройки приложения

Следующим шагом будет выдача прав приложению на выгрузку фотографий. Для этого необходимо скопировать ссылку с документации VK API и подставить в нее ID созданного приложения (Рис. 3-4). Также следует изменить значение переменной `scope` на `photos`.

```
https://oauth.vk.com/authorize?  
client_id=5490057&display=page&redirect_uri=https://oauth.vk.com/blank.html&scope=friends  
&response_type=token&v=5.52
```

Рисунок 3 – Стандартная ссылка для выдачи прав

```
← → ↻ https://oauth.vk.com/authorize?client\_id=7721867&display=page&redirect\_uri=https://oauth.vk.com/blank.html&scope=photos&response\_type=token&v=5.52
```

Рисунок 4 – Измененная ссылка для выдачи прав

Далее пользователя перекинет на страницу, где необходимо подтвердить выдачу прав. После этого в адресной строке появится токен и ID

пользователя, которые нужно скопировать так как они понадобятся в дальнейшем для выгрузки фотографий (Рис. 5).

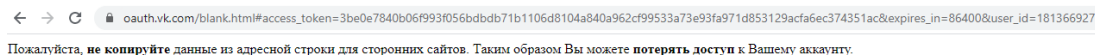


Рисунок 5 – Токен и ID пользователя

Теперь можно приступать к написанию скрипта, выгружающего все фотографии с личной страницы с помощью метода `photos.getAll`. Для начала подключим все необходимые библиотеки (Рис. 6).

```
File Edit Format Run Options Window Help
import requests, json, time, os
```

Рисунок 6 – Подключение библиотек

Далее напишем функцию `get_foto_data` для получения общей информации о фотографиях на личной странице в формате json (Рис. 7).

```
VK_USER_ID = 181366927
VK_TOKEN = "379367dd72d0f9fea2ca37d33b4ed12e99284726838d3f15bd7275df5117f88d232ed770f1d5c75ad4a69"

def get_foto_data(offset=0, count=5):
    api = requests.get("https://api.vk.com/method/photos.getAll", params={
        'owner_id': VK_USER_ID,
        'access_token': VK_TOKEN,
        'offset': offset,
        'count': count,
        'photo_sizes': 0,
        'v': 5.103
    })
    return json.loads(api.text)
```

Рисунок 7 – Получение информации о фотографиях

Теперь опишем основную функцию, которая будет отвечать непосредственно за выгрузку фотографий в локальную папку `images`. В цикле будем перебирать `url`, которые возвращает предыдущая функция и выгружать с данных ссылок фотографии. Количество одновременно выгружаемых картинок будет равно 5, после чего будет произведено смещение и выгрузка продолжится. (Рис. 8).

```
def get_foto():
    data = get_foto_data()
    count_foto = data["response"]["count"]
    i = 0
    count = 5
    fotos = []
    print(count_foto)
    while i <= count_foto:
        if i != 0:
            data = get_foto_data(offset=i, count=count)
        for files in data["response"]["items"]:
            file_url = files["sizes"][-1]["url"]
            filename = file_url.split("/")[-1]
            filename = filename.split("?")[0]
            fotos.append(filename)
            time.sleep(0.1)
            api = requests.get(file_url)

            with open("images/%s" % filename, "wb") as file:
                file.write(api.content)
                print(filename)

        print(i)
        i += count
    print(len(fotos))
```

Рисунок 8 – Код функции get_foto

В конце кода вызовем функцию get_foto и весь код будет выглядеть следующим образом (Рис. 9).

```

vk_save.py - D:\доки\статья VK API\vk_save.py (3.9.1)
File Edit Format Run Options Window Help
import requests, json, time, os

VK_USER_ID = 181366927
VK_TOKEN = "379367dd72d0f9fea2ca37d33b4ed12e99284726838d3f15bd7275df5117f988d232ed770f1d5c75ad4a69"

def get_foto_data(offset=0, count=5):
    api = requests.get("https://api.vk.com/method/photos.getAll", params={
        'owner_id': VK_USER_ID,
        'access_token': VK_TOKEN,
        'offset': offset,
        'count': count,
        'photo_sizes': 0,
        'v': 5.103
    })
    return json.loads(api.text)

def get_foto():
    data = get_foto_data()
    count_foto = data["response"]["count"]
    i = 0
    count = 5
    fotos = []
    print(count_foto)
    while i <= count_foto:
        if i != 0:
            data = get_foto_data(offset=i, count=count)
        for files in data["response"]["items"]:
            file_url = files["sizes"][-1]["url"]
            filename = file_url.split("/")[-1]
            filename = filename.split("?")[0]
            fotos.append(filename)
            time.sleep(0.1)
            api = requests.get(file_url)

            with open("images/%s" % filename, "wb") as file:
                file.write(api.content)
                print(filename)

        print(i)
        i += count
    print(len(fotos))

def main():
    get_foto()

if __name__ == "__main__":
    main()

```

Рисунок 9 – Код выгрузки фотографий с личной страницы

Выполним данный код и в выводе увидим количество выгруженных фото и их название (Рис. 10).

```

IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\доки\статья VK API\vk_save.py =====
11
P0tHYW4TsYs.jpg
Am4drlOyxA8.jpg
8r444117BPw.jpg
7UPw3xNj1gc.jpg
8bHuQ61MrO4.jpg
0
CRz9E3pz8FY.jpg
DaF_z64acRs.jpg
YPsZFibdcow.jpg
CRz9E3pz8FY.jpg
irxQbJrxBZo.jpg
5
2Im-VxcRL48.jpg
10
11
>>> |

```

Рисунок 10 – Результат выполнения кода

Как видно из рисунка 10 количество выгруженных фотографий равно 11, но можно заметить, что на странице присутствуют две фотографии с одинаковым названием. Именно поэтому общее число картинок составит 10 штук (Рис. 11).

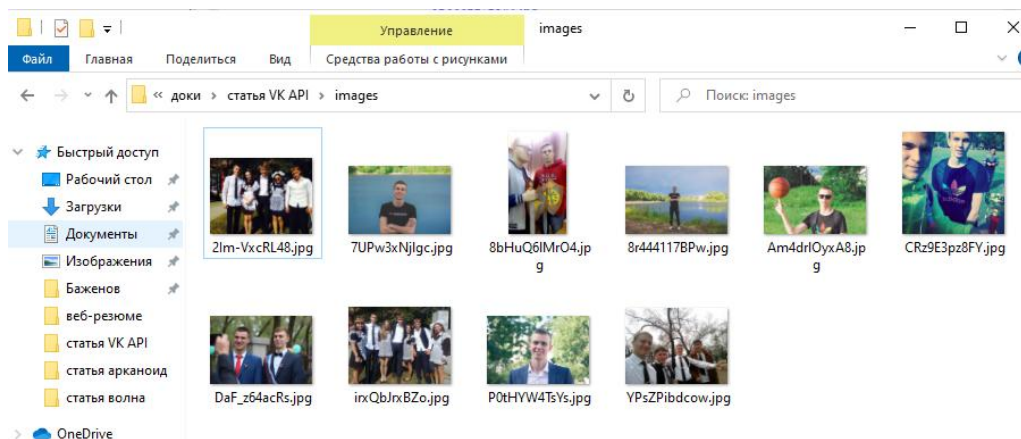


Рисунок 11 – Выгруженные фотографии

3. Выводы

Таким образом, был написан скрипт, выгружающий все фотографии с личной страницы в социальной сети «ВКонтакте» с помощью VK API и языка программирования Python. Умение работать с различными API очень полезный навык в арсенале любого программиста и тем, кто не знаком с данной технологий следует обратить на нее внимание. API значительно облегчает взаимодействие с различными сервисами и помогает быстро выгружать или получать необходимую информацию.

Библиографический список

1. Щедрин С. С. Репликация баз данных с использованием API // Информационно-телекоммуникационные системы и технологии. 2014. С. 346-347.
2. Чернова И. Использование API Яндекс.Метрика // Системный администратор. 2014. №. 7-8. С. 90-94.
3. Биков Д. И. Способы обработки запросов для чат-бота при помощи инструментов VK API // Приоритетные направления инновационной деятельности в промышленности. Москва: Общество с ограниченной ответственностью "КОНВЕРТ", 2020. С. 35-36.
4. Крайников А. Н., Банников А. А. Управление «GPIO» с помощью «VK API» // Международный журнал прикладных наук и технологий «Integral». 2020. №. 4.
5. Баранов А. Ю. Разработка мобильного приложения для ведения заметок с интеграцией VK API // Математическое моделирование и информационные технологии. 2020. С. 49-49.
6. Сейдаметов Г. С., Усеинов Ш. М. Загрузка фотографий через API VK на

- языке программирования PHP // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2018. №. 2. С. 35-41.
7. Ампилогов Н. С. Типовая структура API социальных сетей // Альманах научных работ молодых ученых Университета ИТМО. 2017. С. 4-6.
 8. Знакомство с API ВКонтakte URL: https://vk.com/dev/first_guide (дата обращения: 08.01.2021).
 9. Python URL: <https://ru.wikipedia.org/wiki/Python> (дата обращения: 08.01.2021).
 10. API URL: <https://ru.wikipedia.org/wiki/API> (дата обращения: 08.01.2021).