

# 2

## Facebook: анализ фан-страниц, исследование дружественных связей и многое другое

В этой главе мы подключимся к платформе Facebook через ее (Social) Graph API и рассмотрим некоторые из многочисленных возможностей. Facebook — одна из самых обширных социальных сетей и является чем-то вроде швейцарского армейского ножа. Кроме того, более половины из двух миллиардов пользователей этой сети<sup>1</sup> проявляют активность каждый день, обновляя статусы, размещая фотографии, обмениваясь сообщениями, беседа в режиме реального времени, играя в игры, совершая покупки и почти все, что только можно представить. С точки зрения анализа социальных сетей данные, которые Facebook хранит о людях, группах и продуктах, представляют очень большой интерес, потому что Facebook API предлагает невероятные возможности для получения информации (самый ценный товар в мире) и сбора ценных идей. С другой стороны, большие возможности накладывают большую ответственность, поэтому в Facebook реализован самый сложный онлайн-контроль конфиденциальности<sup>2</sup>, который когда-либо видел мир, чтобы помочь защитить своих пользователей от посягательств.

Стоит отметить, что, хотя социальная сеть Facebook провозгласила себя социальным графом, она постепенно превращается также в ценный граф интересов, потому что позволяет устанавливать отношения между людьми и предметами,

---

<sup>1</sup> Статистика использования интернета (<http://bit.ly/1a1ljf8>) показывает, что в 2017 году население земного шара начитывало примерно 7,5 миллиарда человек, а интернетом пользовалось примерно 3,9 миллиарда.

<sup>2</sup> <http://on.fb.me/1a1llg9>.

которые их интересуют, и выражать реакцию (например, кликнув Like (Нравится)). В связи с этим все чаще можно услышать, что Facebook позиционируется как «социальный граф интересов». В целом можно смело утверждать, что графы интересов неявно присутствуют и доступны для извлечения в большинстве источников социальных данных. Например, в главе 1 было показано, что Twitter действительно является графом интересов благодаря его асимметричной модели «следования» (или, другими словами, «заинтересованности») в отношениях между людьми и другими людьми, местами или вещами. Идея представления Facebook как графа интересов протянется красной нитью через эту главу, а подробнее об извлечении графа интересов из социальных данных мы поговорим в главе 8.

Далее в этой главе предполагается, что у вас есть активный аккаунт в Facebook<sup>1</sup>, который необходим для получения доступа к Facebook API. В 2015 году в Facebook API были внесены изменения<sup>2</sup>, ограничившие доступ к данным третьим лицам. Например, вы больше не сможете получить доступ к обновлениям статуса или списку интересов вашего друга через программный интерфейс. Эти изменения обусловлены соображениями конфиденциальности. Из-за них в этой главе больше внимания будет уделено использованию Facebook API для измерения вовлеченности пользователей на публичных страницах<sup>3</sup>, например, созданных компаниями или знаменитостями. Но для доступа даже к этим данным требуется все больше привилегий, и вам, возможно, придется запросить одобрение в Facebook, чтобы получить доступ к некоторым из имеющихся возможностей.



Загрузите последний исправленный исходный код примеров для этой (и любой другой) главы, доступный по адресу: <http://bit.ly/Mining-the-Social-Web-3E>. Воспользуйтесь также преимуществами виртуальной машины, описанными в приложении А, чтобы получить максимальное удовольствие от опробования примеров кода.

## 2.1. Обзор

Так как это уже вторая глава в книге, здесь мы рассмотрим вещи, более сложные, чем в главе 1, но все еще доступные для широкой аудитории. В этой главе вы познакомитесь с:

<sup>1</sup> <http://on.fb.me/1a1lkcd>.

<sup>2</sup> <http://tcrn.ch/2zFetfo>.

<sup>3</sup> В терминологии Facebook — фан-страницах, или пабликах. — *Примеч. пер.*

- Facebook Graph API и узнаете, как посылать запросы этому программному интерфейсу;
- протоколом Open Graph и узнаете, как он связан с социальным графом в Facebook;
- возможностью программного доступа к публичным страницам, созданным, например, компаниями или знаменитостями;
- приемами извлечения основных социальных показателей, таких как количество лайков, комментариев и ссылок для оценки вовлеченности аудитории;
- управлением данными с использованием pandas DataFrames и последующей визуализацией результатов.

## 2.2. Facebook Graph API

Платформа Facebook — зрелый, надежный и хорошо документированный интерфейс к наиболее полному как с точки зрения широты, так и глубины и хорошо организованному хранилищу информации. В ширину она охватывает около четверти всего населения мира, а в глубину включает большой объем сведений о любом из ее пользователей. В отличие от Twitter, где используется асимметричная модель дружбы, которая открыта и основана на следовании за другими пользователями без всякого согласия, в Facebook модель дружбы симметрична и требует взаимного согласия пользователей, чтобы они могли видеть взаимодействия и действия друг друга.

Кроме того, в отличие от Twitter, где практически все общение, кроме личных сообщений, происходит посредством публичных статусов, Facebook обеспечивает гораздо более тонкое управление конфиденциальностью, и дружественные отношения могут организовываться и поддерживаться в виде списков с различными уровнями видимости, доступными другу по любому конкретному виду деятельности. Например, вы можете поделиться ссылкой или фотографией только с определенным списком друзей, а не со всей вашей социальной сетью.

Извлекать данные из Facebook для анализа можно, только зарегистрировав приложение и используя его как точку входа в платформу разработчиков Facebook. Кроме того, такому приложению будут доступны только те данные, доступ к которым явно разрешен пользователем. Например, как разработчик приложения для Facebook, вы также будете выступать в роли пользователя, входящего в приложение, и приложение будет иметь доступ только к тем данным, доступ

к которым вы разрешите явно. Как пользователь, вы можете думать о приложении как об одном из ваших друзей в Facebook: вы сможете полностью контролировать доступность тех или иных данных для приложения и закрыть доступ к ним в любой момент. Документ с описанием политики платформы Facebook<sup>1</sup> должен прочитать любой разработчик Facebook, потому что в нем исчерпывающе описываются права и обязанности всех пользователей Facebook, а также дух и буква закона для разработчиков Facebook. Если вы еще не сделали этого, воспользуйтесь моментом, ознакомьтесь с политикой разработчиков Facebook и добавьте закладку на домашнюю страницу разработчиков Facebook<sup>2</sup> — единую точку доступа к платформе Facebook и ее документации. Кроме того, имейте в виду, что программный интерфейс может меняться. Вследствие неумолимой заботы о безопасности и конфиденциальности ваши привилегии, доступные на время экспериментов с платформой Facebook, будут ограничены. Чтобы получить доступ к некоторым возможностям и конечным точкам API, может потребоваться отправить приложение на проверку и утверждение<sup>3</sup>. Пока ваше приложение будет соблюдать условия обслуживания<sup>4</sup>, у вас все должно быть в порядке.



Как у разработчика, анализирующего собственный аккаунт, у вас не должно возникнуть проблем с доступом к данным вашего аккаунта. Но будьте осторожны и не стремитесь создать *приложение*, которое пытается получить больше данных, чем необходимо для выполнения его задачи, потому что пользователь может не доверить вашему приложению такой уровень привилегий (и правильно сделает).

Далее в этой главе мы будем обращаться к платформе Facebook программным способом, однако Facebook предоставляет ряд полезных инструментов для разработчиков<sup>5</sup>, включая приложение Graph API Explorer<sup>6</sup>, которое мы используем для начального знакомства с социальным графом. Приложение предлагает простой и понятный способ опроса социального графа, и как только вы усвоите особенности его работы, мы переведем запросы на язык Python для их автоматизации, и дальнейшая обработка будет осуществляться естественным путем. Несмотря на то что в рамках обсуждения мы будем использовать Graph API, вы можете извлечь дополнительную выгоду, прочитав исчерпывающее введение в API Graph<sup>7</sup>.

<sup>1</sup> <http://bit.ly/1a1lm3C>.

<sup>2</sup> <http://bit.ly/1a1lm3Q>.

<sup>3</sup> <http://bit.ly/2vDb2B1>.

<sup>4</sup> <http://on.fb.me/1a1lMXM>.

<sup>5</sup> <http://bit.ly/1a1lnVf>.

<sup>6</sup> <http://bit.ly/2jd5Xdq>.

<sup>7</sup> <http://bit.ly/1a1lobU>.



Обратите внимание, что Facebook прекратил поддержку языка запросов Facebook Query Language<sup>1</sup> (FQL). Начиная с 8 августа 2016 года запросы FQL больше не обрабатываются. Теперь разработчики должны выполнять вызовы к Graph API. Для тех, кто раньше уже создал приложение, использующее FQL, Facebook выпустил инструмент обновления API Upgrade Tool<sup>2</sup>, который можно использовать для обновления таких приложений.

### 2.2.1. Знакомство с Graph API

Как нетрудно догадаться, социальный граф Facebook — это обширный граф<sup>3</sup> — структура данных, представляющая социальные связи и состоящая из узлов и связывающих их ребер. Graph API предоставляет основные средства взаимодействия с социальным графом, и лучший способ познакомиться с этим программным интерфейсом — потратить несколько минут на эксперименты с приложением Graph API Explorer<sup>4</sup>.

Важно отметить, что Graph API Explorer не отличается ничем особенным. Кроме возможности ввода и проверки токена доступа, это самое обычное приложение для Facebook, использующее тот же программный интерфейс, что и любое другое приложение. С другой стороны, приложение Graph API Explorer удобно использовать, когда у вас есть определенный токен OAuth, связанный с определенным набором разрешений для разрабатываемого приложения, и вы хотите выполнить несколько запросов для проверки своих предположений или для отладки. Мы еще вернемся к этой идее чуть позже, когда займемся организацией программного доступа к Graph API. На рис. 2.1, 2.2 и 2.3 показана последовательность запросов к Graph API, выполняемых в результате щелчка на значке с символом «плюс» (+) и добавления связей и полей. Отметим некоторые элементы на этих рисунках:

#### *Токен доступа*

Токен доступа (access token), отображаемый в приложении, — это токен OAuth<sup>5</sup>, предоставляемый зарегистрированному пользователю для удобства; именно этот токен OAuth потребует приложение для доступа к данным. Мы будем использовать его на протяжении всей этой главы, но вы можете

<sup>1</sup> <http://bit.ly/1a1lmRd>.

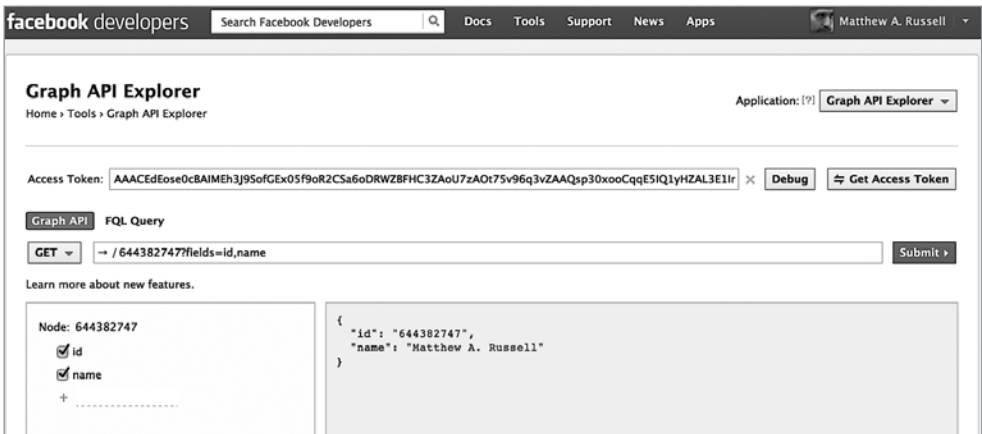
<sup>2</sup> <http://bit.ly/2MGU7Z9>.

<sup>3</sup> <http://bit.ly/1a1loIX>.

<sup>4</sup> <http://bit.ly/2jd5Xdq>.

<sup>5</sup> <http://bit.ly/1a1kZWN>.

также ознакомиться с приложением Б, где дается краткий обзор OAuth, включая сведения об особенностях реализации OAuth в Facebook и приемах получения токена доступа. Как уже упоминалось в главе 1, если это ваше первое знакомство с OAuth, на данный момент вам достаточно знать, что это протокол открытой авторизации, ставший стандартом для социальных сетей. Проще говоря, OAuth — это средство, с помощью которого пользователи могут разрешать сторонним приложениям получать доступ к данным в их учетных записях без необходимости передавать конфиденциальную информацию, например пароль.



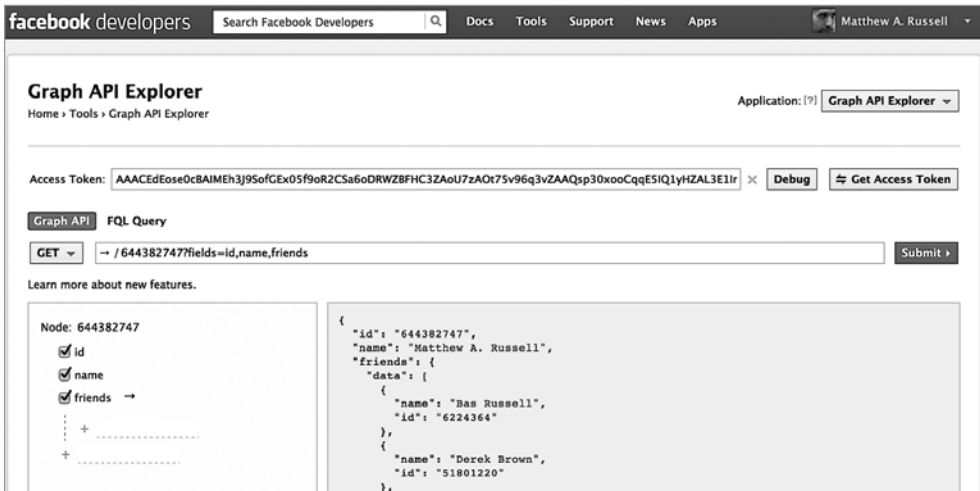
**Рис. 2.1.** Использование приложения Graph API Explorer для получения узла в социальном графе



Дополнительные сведения об использовании протокола OAuth 2.0, которые понадобятся при создании приложения, требующего авторизации произвольного пользователя для доступа к данным учетной записи, вы найдете в приложении Б.

### Идентификаторы узлов

Основой запроса является узел с идентификатором (ID) «644382747», который соответствует человеку с именем «Matthew A. Russell» (Мэтью А. Рассел) и был загружен для текущего пользователя Graph Explorer, выполнившего вход. Значения «id» и «name» называются *полями* узла. Основой запроса мог бы быть любой другой узел, и, как мы скоро увидим, без всякого труда можно выполнять обход графа и запрашивать другие узлы (которые могут представлять людей или неодушевленные сущности, такие как книги или телешоу).



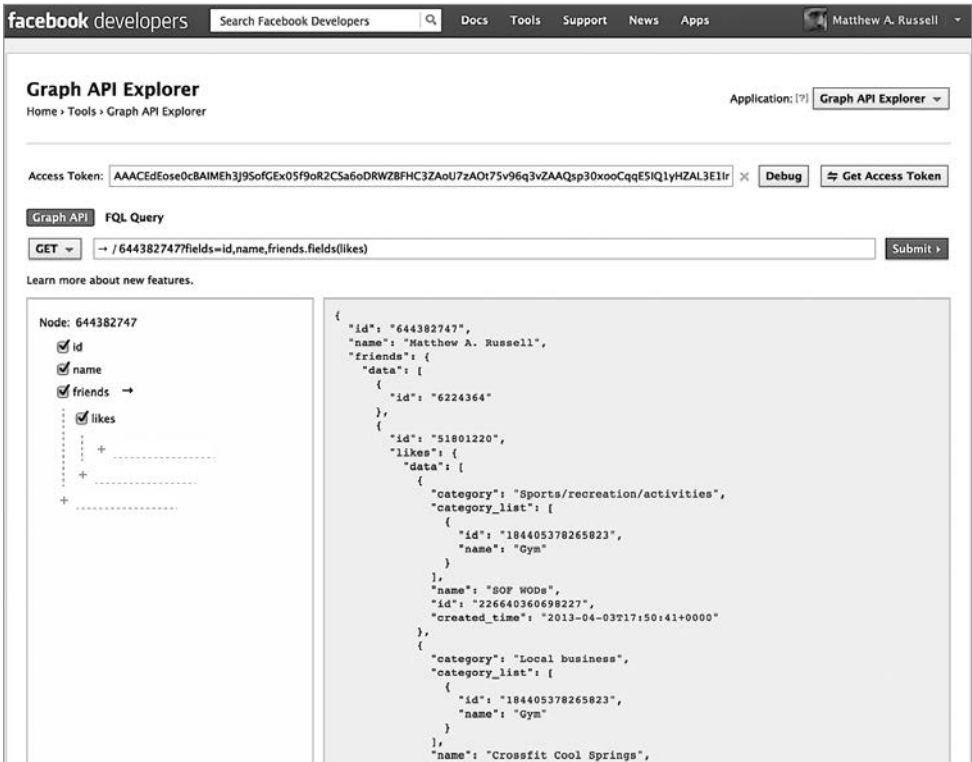
**Рис. 2.2.** Использование приложения Graph API Explorer для последовательного получения узла и его связей с друзьями. Имейте в виду, что для доступа к некоторым данным могут потребоваться дополнительные привилегии, а изолированным приложениям доступен очень ограниченный набор данных. За годы существования в Facebook внесли много изменений в их политику конфиденциальности

### Связи

Запрос можно изменить с помощью связи «friends» (друзья), как показано на рис. 2.2, щелкнув на значке с символом + и выбрав в контекстном меню «connections» (связи) пункт «friends» (друзья). В разделе «friends», отображаемом в консоли, перечислены узлы, связанные с исходным узлом запроса. В данный момент вы можете щелкнуть на значении в поле «id» любого из этих узлов и инициировать запрос с этим конкретным узлом в качестве основы. Выражаясь языком теории графов, вы получили *эго-граф* (ego graph), с субъектом (*эго*) в его логическом центре, который связан с окружающими его узлами. Если нарисовать эго-граф, он будет напоминать колесо со спицами.

### «Лайки»

Еще больше изменить запрос можно, добавив связи с «лайками» каждого из ваших друзей, как показано на рис. 2.3. Однако теперь эти данные считаются конфиденциальными и приложения больше не имеют доступа к «лайкам» ваших друзей. В Facebook были внесены изменения в API, ограничивающие круг информации, к которой приложения могут получить доступ.



**Рис. 2.3.** Использование приложения Graph API Explorer для последовательного получения информации о заинтересованности друзей: запрашивается узел, связи с друзьями и «лайки» этих друзей. Этот пример наглядно показывает, насколько строже и сосредоточеннее на конфиденциальности пользователей стала политика доступа к данным в Facebook

*Отладка*

Кнопка Debug (Отладка) может пригодиться для поиска и устранения проблем в запросах, которые, по вашему мнению, должны возвращать данные, с учетом разрешений, связанных с токеном доступа, но ничего не возвращают.

*Формат ответов JSON*

В ответ на запросы Graph API возвращает результаты в удобном формате JSON, который легко поддается обработке.

Далее в этой главе мы будем исследовать Graph API программным способом, используя специализированный пакет на языке Python, однако есть возмож-



ность выполнять запросы к Graph API более непосредственно — через HTTP — например, сымитировав запрос, который мы видели в Graph API Explorer. Код в примере 2.1 использует пакет `requests`<sup>1</sup> (вместо более громоздкого пакета из стандартной библиотеки Python, такого как `urllib`), чтобы упростить создание HTTP-запроса для получения списка ваших друзей и их лайков. Этот пакет можно установить, выполнив в терминале команду `pip install requests`. Запрос формируется на основе значений в параметре `fields` и идентичен запросу, который мы сконструировали в API Graph Explorer в интерактивном режиме. Особое внимание обратите на синтаксис `likes.limit(10){about}` — здесь используется механизм расширения полей<sup>2</sup> в Graph API, позволяющий выполнять вложенные подзапросы в одном вызове API.

### Пример 2.1. Выполнение запросов к Graph API через HTTP

```
import requests # pip install requests
import json

base_url = 'https://graph.facebook.com/me'

# Определить извлекаемые поля
fields = 'id,name,likes.limit(10){about}'

url = '{0}?fields={1}&access_token={2}'.format(base_url, fields, ACCESS_TOKEN)

# Этот API доступен через HTTP, и запрос к нему можно отправить из браузера,
# с помощью утилиты командной строки, такой как curl, или из программы на
# любом языке, просто пошлав запрос по адресу URL.
# Чтобы убедиться в этом, щелкните по гиперссылке, которая появится в
# блокноте Jupiter Notebook после выполнения этого кода
print(url)

# Преобразовать ответ в формате JSON обратно в структуры
# данных языка Python
content = requests.get(url).json()

# Вывести содержимое ответа в отформатированном виде
print(json.dumps(content, indent=1))
```

Используя синтаксис расширения полей, в запросе к API можно устанавливать ограничения и смещения. Этот начальный пример просто показывает, что Facebook API реализован поверх HTTP. Ниже приводится несколько при-

<sup>1</sup> <http://bit.ly/1a1lrEt>.

<sup>2</sup> <http://bit.ly/1a1lsIE>.

меров использования ограничений/смещений, иллюстрирующих возможности селекторов полей:

```
# Получить 10 моих лайков
fields = 'id,name,likes.limit(10)'

# Получить следующие 10 моих лайков
fields = 'id,name,likes.offset(10).limit(10)'
```

Facebook API автоматически разбивает возвращаемые результаты на страницы, то есть если в ответ на ваш запрос будет найдено большое количество результатов, API не будет передавать их все сразу. Вместо этого результаты будут разбиты на фрагменты (*страницы*) и вы получите *курсор*, ссылающийся на следующую страницу с результатами. Дополнительные сведения о *разбиении результатов на страницы* смотрите в документации<sup>1</sup>.

## 2.2.2. Знакомство с Open Graph Protocol

Вы должны знать, что кроме мощного программного интерфейса Graph API, который позволяет исследовать социальный граф и запрашивать известные объекты, платформа Facebook поддерживает протокол под названием Open Graph Protocol<sup>2</sup> (OGP), который был представлен еще в апреле 2010 года, на той же конференции F8, где был представлен социальный граф. OGP — это механизм, позволяющий разработчикам сделать любую веб-страницу объектом в социальном графе Facebook, внедрив в нее некоторые метаданные RDFa<sup>3</sup>. То есть кроме десятков объектов из «закрытой экосистемы» Facebook, описанных в справочнике<sup>4</sup> Graph API (пользователи, фотографии, видео, отметки о посещении, ссылки, посты и т. д.), можно также встретить страницы из интернета, представляющие значимые понятия, которые были внедрены в социальный граф. Иначе говоря, OGP — это средство «расширения» социального графа, и в документации разработчика Facebook эта идея описывается в разделе «Open Graph»<sup>5</sup>.

---

<sup>1</sup> <http://bit.ly/1a1ltMP>.

<sup>2</sup> <http://bit.ly/1a1lu3m>.

<sup>3</sup> <http://bit.ly/1a1lujR>.

<sup>4</sup> <http://bit.ly/1a1lvEr>.

<sup>5</sup> В этом разделе, описывающем реализацию OGP, термин *социальный граф* используется для обозначения обоих понятий — социального графа и открытого графа, если явно не оговаривается иное.