
Использование сторонних интерфейсов программирования приложений на примере интерфейсов прикладного программирования социальных сетей Facebook и Twitter

А.Г. Янишевская, М.А. Чурсин

Омский Государственный Технический Университет

Аннотация: Проблемы работы с различными типами API на примере социальных сетевых сервисов Facebook и Twitter. Использование Facebook Graph API для получения данных из социального графа Facebook. Схемы работы Streaming и REST типов API.

Ключевые слова: Интерфейсы программирования приложений, сетевые социальные сервисы, передача репрезентативного состояния, потоковая передача данных, API, application programming interface, Facebook, FQL, Facebook Graph API, Twitter, twitter streaming api, rest api, streaming api.

В современном обществе важную роль играют сетевые социальные сервисы. Крупнейшие корпорации мира считают неотъемлемой частью своей внешней политики использование twitter. Не редки случаи, когда районные отделения полиции заводили себе свой instagram. При появлении всего этого разнообразия сервисов закономерным стало появление людей, желающих воспользоваться данными в своем приложении или же просто размещать свою ленту социальной сети у себя на странице. Заметив в этом социальную выгоду, разработчики облегчили доступ к своим данным выпуском интерфейсов программирования приложений.

Интерфейсы программирования приложений (Application programming interface, API [эй-пи-ай], далее API) социальных сервисов можно разделить на два типа:

1. API, предоставляющие непосредственный доступ к данным на серверах сервиса,
2. Streaming-API, раздающие все новые данные в режиме реального времени.

Рассмотрим особенности работы с обоими типами на примере API таких крупных сервисов как Facebook и Twitter.

Facebook Graph API и FQL

Graph API является основным способом для получения данных из социального графа Facebook. Это API низкого уровня, который можно использовать для запроса данных, создавать новые данные, загружать фотографии и множество других задач.

Graph API назван в честь идеи “социального графа” - представление информации о Facebook, состоящей из:

- узлов (Nodes) - базовые понятия, такие как пользователь, фотография, страница, комментарий,
- ребер (edges) - связи между этими "узлами", например фотографии на странице, или фото автора комментария,
- полей (fields) - информация о параметрах узлов, например, день рождения пользователя, или заголовок страницы).

Facebook Query Language (далее FQL) - позволяет делать запросы в SQL стиле запрашивать данные. Это предоставляет расширенные возможности, недоступные в Graph API.

Запросы имеют вид `SELECT [fields] FROM [table] WHERE [conditions]`. В отличие от SQL, FQL FROM может содержать только одну таблицу. Хотя и возможно использовать ключевые слова в SELECT или WHERE, но подзапросы не могут ссылаться на переменные из области внешнего запроса. Запрос также должен быть индексируемым, это означает, что он должен запрашивать поля, которые помечены как проиндексированные.

FQL может обрабатывать простую математику, основные булевы переменные, логические И ИЛИ НЕ, а так же ORDER BY и LIMIT.

Несмотря на кажущееся удобство использования данных непосредственно с сервера социального сервиса это накладывает ряд существенных ограничений, начиная с элементарной задержки выполнения запросов, связанных с загрузкой сетевого-канала, заканчивая сложностью отладки из-за отсутствия прямого доступа к базам данных. Также если рассматривать этот конкретный пример, важным недостатком является ограничение на связывание таблиц при запросе, что может также стать серьезной проблемой при разработке сложных приложений.

Twitter streaming api

Набор потоковых API, предлагаемых Twitter дают разработчикам низкий уровень задержки в получении глобального потока данных из Tweet. Получение клиентом потоковых данных характеризуется отсутствием каких-либо накладных расходов, связанных с опросом базы данных самого twitter.

Twitter предлагает несколько типов стриминга данных, каждый из которых подходит для конкретных случаев.

- **Public streams** — Потоки открытых данных, проходящих через Twitter. Подходит для отслеживания твитов всех пользователей, а также для простого анализа данных.
- **User streams** — поток данных одного пользователя.
- **Site streams** — многопользовательская версия User streams. Этот тип предназначен для серверов, которые должны подключаться к Twitter от имени многих пользователей.

Различия между Streaming и REST.

Передача репрезентативного состояния (англ. Representational State Transfer, далее REST) — стиль построения архитектуры распределенного приложения.

Подключение к streaming API требует поддержания постоянного соединения HTTP открытым. Во многих случаях это заставляет смотреть на работу вашего приложения иначе, чем в случае, если бы это было взаимодействие с API REST. Для примера рассмотрим веб-приложение, которое принимает запросы пользователей, делает один или несколько запросов в twitter API, и печатает результат для пользователя, в качестве ответа на первоначальный запрос. Схема работы такого приложения изображена на рис.1.

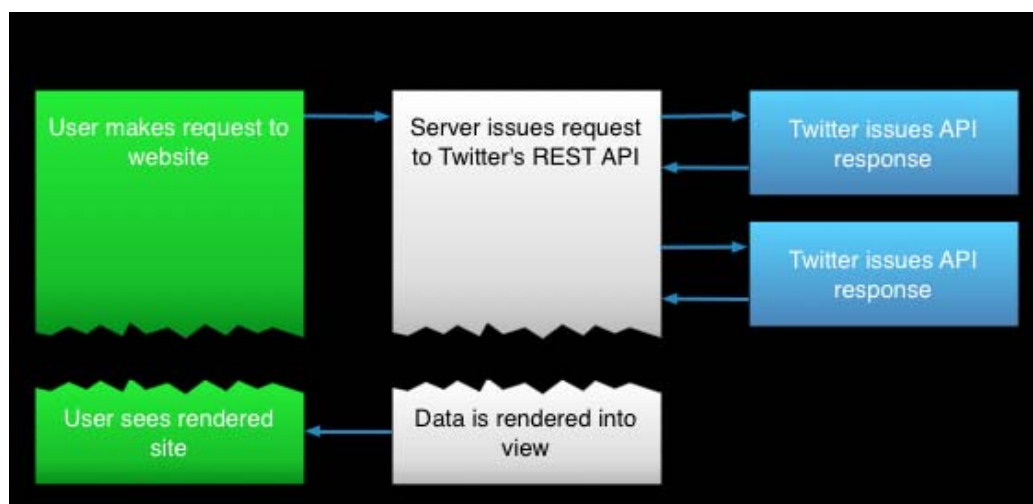


Рис. 1. – Схема работы с REST API

Приложения, которое подключается к Streaming API, не смогут установить соединение в ответ на запрос пользователя, как показано в приведенном выше примере. Вместо этого код для поддержания соединения Streaming, как правило, выполняются в процессе отдельно от процесса, который обрабатывает HTTP запросы. Схема работы Streaming приложения изображена на рис.2.

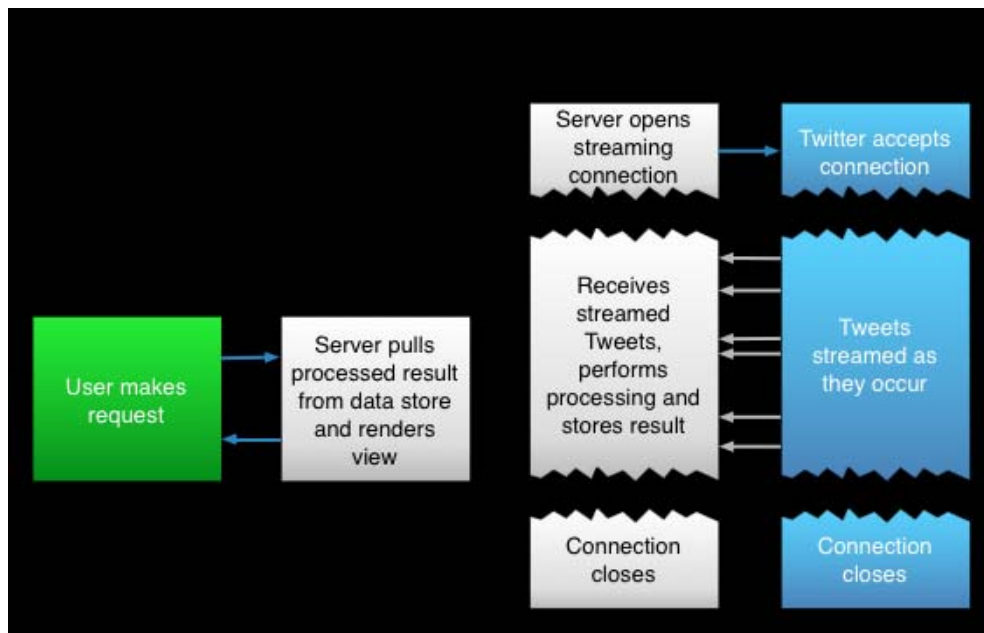


Рис. 2. – Схема работы с STREAMING API

В итоге сервис получает все входные твиты и сам выполняет любой их анализ, фильтрацию, и/или агрегацию необходимую перед сохранением результата в хранилище данных. Хотя эта модель является более сложной, чем в первом примере, преимущество от того, что поток идет в реальном времени упрощает интеграцию twitter в любое приложение.

Различие в способе предоставления данных целевыми сервисами приводит к тому, что для каждого сервиса приходится использовать отдельный метод получения данных. Это замедляет работу, увеличивает стоимость разработки, так как требуются специалисты, разбирающиеся в особенностях каждого сервиса, а также способные построить эффективную модель соотношения данных поступающих разными способами.

Литература

1. Литвинов С.В. Профессиональная социальная сеть «Сообщество пользователей уникальным научным оборудованием». Инженерный вестник Дона, 2015, №2 URL: ivdon.ru/ru/magazine/archive/n1e2010/172.



2. Литвинов С.В., Носко В.И. Модели инсорсинга уникального научного оборудования на базе профессиональных социальных сетей // Инженерный вестник Дона, 2015, №2 URL: ivdon.ru/ru/magazine/archive/n3y2010/219.

3. Facebook developers. Facebook Query Language (FQL) Reference, 2012, URL: developers.facebook.com/docs/reference/fql/.

4. Twitter developers. Twitter The Streaming APIs, 2014, URL: dev.twitter.com/docs/api/streaming.

5. Jesse Russel, Ronald Kohan Web API. Bookvika publishing, 2012. 140 p.

6. Leonard Richardson, Mike Amundsen, Sam Ruby RESTful Web APIs. O'Reilly Media, 2013. 406 p.

7. Kevin Makice Twitter API: Up and Running. O'Reilly Media, 2009. 414 p.

8. Matthew A. Russell 21 Recipes for Mining Twitter. O'Reilly, 2011. 72 p.

9. Mario Scheliga Facebook-Anwendungen programmieren. O'Reilly, 2011. 464 p.

10. Ido Green Web Workers. O'Reilly, 2012. 58 p.

References

1. Litvinov S.V. Inzhenernyj vestnik Dona (Rus), 2015, №2 URL: ivdon.ru/ru/magazine/archive/n1e2010/172.

2. Litvinov S.V., Nosko V.I. Inzhenernyj vestnik Dona (Rus), 2015, №2 URL: ivdon.ru/ru/magazine/archive/n3y2010/219.

3. Facebook developers. Facebook Query Language (FQL) Reference, 2012, URL: developers.facebook.com/docs/reference/fql/.

4. Twitter developers. Twitter The Streaming APIs, 2014, URL: dev.twitter.com/docs/api/streaming.



5. Jesse Russel, Ronald Kohan Web API. Bookvika publishing, 2012. 140 p.
6. Leonard Richardson, Mike Amundsen, Sam Ruby RESTful Web APIs. O'Reilly Media, 2013. 406 p.
7. Kevin Makice Twitter API: Up and Running. O'Reilly Media, 2009. 414 p.
8. Matthew A. Russell 21 Recipes for Mining Twitter. O'Reilly, 2011. 72 p.
9. Mario Scheliga Facebook-Anwendungen programmieren. O'Reilly, 2011. 464 с.
10. Ido Green Web Workers. O'Reilly, 2012. 58 p.