THE ANALYSYS OF WAYS OF GATHERING SOCIAL DATA FROM THE INTERNET

Sukhanov A.[1], Maratkanov A.[2] (Russian Federation)

АНАЛИЗ СПОСОБОВ СБОРА СОЦИАЛЬНЫХ ДАННЫХ ИЗ СЕТИ ИНТЕРНЕТ

Суханов А. А.[1], Маратканов А. С.[2] (Российская Федерация)

[1]Суханов Александр Александрович / Sukhanov Aleksandr – магистрант;
[2]Маратканов Александр Сергеевич / Maratkanov Aleksandr – магистрант,
кафедра компьютерных систем и сетей, факультет информатики и систем управления,
Московский государственный технический университет им. Н. Э. Баумана, г. Москва

Abstract: *users actively share their data. The problem of collecting such information becomes relevant. First of all, there is no universal way to collect that data. Also, many of the resources do not allow to collect information, so you have to use a variety of ways. Using API is the main way of gathering social data. But there are some other ways like semantic content analysis (parsing) web pages and collect data using the simulation of real user behavior. That ways of collecting social data have been considered in the article too.*

Аннотация: *сейчас пользователи активно делятся своими данными. В связи с этим возникает необходимость контролировать и анализировать эти данные. Из-за этого приобретает актуальность проблема сбора подобной информации. Так как не все ресурсы предоставляют возможность собирать информацию, приходится использовать множество способов. Основным способом является использование API. Но кроме этого в статье рассмотрены и такие способы, как семантический анализ содержимого (парсинг) веб-страниц, так и сбор данных с помощью эмуляции поведения пользователя.*

Keywords: *social data, API, parsing, Selenium, data crawling, social networks.*
Ключевые слова: *социальные данные, API, парсинг, Selenium, сбор данных, социальные сети.*

1. Сбор данных с помощью API

API (от англ. application programming interface) – это интерфейс взаимодействия между сайтом и сторонними программами и серверами, набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) или операционной системой для использования во внешних программных продуктах [1].

Все мы привыкли к тому, что на разных онлайн-сервисах или платформах мы можем вместо регистрации войти через свои аккаунты в социальных сетях. Именно это и является использованием API, когда сервисы или приложения используют базы данных социальных сетей. При этом сервис может получать информацию о пользователе и использовать ее в своих целях. Еще один пример: Amazon предлагает пользователю книги, основанные на выборе книг его друзей в facebook.

У самой популярной социальной сети на территории СНГ ВКонтакте тоже есть API. Open API — система для разработчиков сторонних сайтов, которая предоставляет возможность легко авторизовывать пользователей ВКонтакте на Вашем сайте. Кроме этого, с согласия пользователей, вы сможете получить доступ к информации об их друзьях, фотографиях, аудиозаписях, видеороликах и прочих данных ВКонтакте для более глубокой интеграции с Вашим проектом.

В рамках подключения к Open API создается специальное приложение, которое позволяет использовать на Вашем сайте все текущие методы ВКонтакте API. Помимо этого, Open API предоставляет возможность упростить процесс регистрации новых пользователей на Вашем сайте, если у них уже есть учетная запись ВКонтакте [2].

API ВКонтакте — это интерфейс, который позволяет получать информацию из базы данных vk.com с помощью http-запросов к специальному серверу. Вам не нужно знать в подробностях, как устроена база, из каких таблиц и полей каких типов она состоит — достаточно того, что API-запрос об этом «знает». Синтаксис запросов и тип возвращаемых ими данных строго определены на стороне самого сервиса.

Несмотря на все удобство использования API, существует одно ограничение - социальная сеть не может отдавать все данные, которые видны пользователям в интерфейсе. Связано это, во-первых, с тем, что социальные сети стараются сохранять приватность своих пользователей, а во-вторых, с тем, что некоторые функции слишком сильно нагружают серверную часть приложения. Для того чтобы преодолеть это ограничение следует использовать такой механизм как парсинг веб-сайта.

2. Сбор данных с помощью семантического разбора веб-страниц

Парсинг сайтов – последовательный синтаксический анализ информации, размещённой на интернет-страницах. На человеческом языке предоставлена информация, знания, ради которых, собственно, люди и пользуются Интернетом. Компьютерные языки (html, JavaScript, css) определяют как информация выглядит на мониторе [3].

Для парсинга html наиболее распространены следующие варианты:

- Регулярные выражения. Разумеется, регулярные выражения являются наиболее универсальным и настраиваемым средством семантического разбора. Однако использовать исключительно их – довольно

сложная задача как для разработчика системы, из-за того, что потребуется очень сильно специализировать каждое регулярное выражение, а кроме того, регулярные выражения создают дополнительную нагрузку на ОС.

- BeatifulSoup, lxml. Это две наиболее популярные библиотеки для парсинга html и выбор одной из них, скорее, обусловлен личными предпочтениями. Кроме того, эти библиотеки тесно переплелись: BeautifulSoup стал использовать lxml в качестве внутреннего парсера для ускорения, а в lxml был добавлен модуль soupparser. Данные библиотеки являются наиболее легковесными и, на наш взгляд, являются наиболее оптимальным выбором.

К недостаткам данного метода, кроме, разумеется, долгого времени как разработки, так и работы, можно отнести и то, что не все данные могут быть доступны без какой-либо аутентификации пользователя. На данный момент все социальные сети позволяют скрывать свои страницы от пользователей, не являющихся членами данной социальной сети. Для того чтобы получить доступ к большему числу данных, можно использовать средства, позволяющие эмулировать поведение реального пользователя в браузере.

3. Сбор данных с помощью средств эмуляции поведения пользователя в браузере

Одним из таких средств является Selenium. Selenium – это проект, в рамках которого разрабатывается серия программных продуктов с открытым исходным кодом (open source):

- Selenium WebDriver,
- Selenium RC,
- Selenium Server,
- Selenium Grid,
- Selenium IDE.

Называть просто словом Selenium любой из этих пяти продуктов неправильно, хотя так часто делают, если из контекста понятно, о каком именно из продуктов идёт речь, или если речь идёт о нескольких продуктах одновременно, или обо всех сразу [4].

3.1. Selenium WebDriver

Selenium WebDriver – это программная библиотека для управления браузерами. Часто употребляется также более короткое название WebDriver.

Иногда говорят, что это «драйвер браузера», но на самом деле это целое семейство драйверов для различных браузеров, а также набор клиентских библиотек на разных языках, позволяющих работать с этими драйверами.

Это основной продукт, разрабатываемый в рамках проекта Selenium.

Selenium WebDriver называется также Selenium 2.0, причина этого будет объяснена ниже.

Как уже было сказано, WebDriver представляет собой семейство драйверов для различных браузеров плюс набор клиентских библиотек для этих драйверов на разных языках программирования:

В рамках проекта Selenium разрабатываются драйверы для браузеров Firefox, Internet Explorer и Safari, а также драйверы для мобильных браузеров Android и iOS. Драйвер для браузера Google Chrome разрабатывается в рамках проекта Chromium, а драйвер для браузера Opera (включая мобильные версии) разрабатывается компанией Opera Software. Поэтому они формально не являются частью проекта Selenium, распространяются и поддерживаются независимо. Но логически, конечно, можно считать их частью семейства продуктов Selenium.

Аналогичная ситуация и с клиентскими библиотеками – в рамках проекта Selenium разрабатываются библиотеки для языков Java, .Net (C#), Python, Ruby, JavaScript. Все остальные реализации не имеют отношения к проекту Selenium, хотя, возможно, в будущем, какие-то из них могут влиться в этот проект.

3.2. Selenium RC

Selenium RC – это предыдущая версия библиотеки для управления браузерами. Аббревиатура RC в названии этого продукта расшифровывается как Remote Control, то есть это средство для «удалённого» управления браузером.

Эта версия с функциональной точки зрения значительно уступает WebDriver. Сейчас она находится в законсервированном состоянии, не развивается и даже известные баги не исправляются. А всем, кто сталкивается с ограничениями Selenium RC, предлагается переходить на использование WebDriver.

Иногда Selenium RC называется также Selenium 1.0, тогда как WebDriver называется Selenium 2.0. Хотя на самом деле дистрибутив версии 2.0 включает в себя одновременно обе реализации – и Selenium RC, и WebDriver. А вот когда выйдет версия 3.0 – в ней останется только WebDriver.

С технической точки зрения WebDriver не является результатом эволюционного развития Selenium RC, они построены на совершенно разных принципах и у них практически нет общего кода. Объединяет их лишь тот факт, что обе реализации были сделаны в рамках проекта Selenium. Ну, или если быть совсем точным, WebDriver сначала был самостоятельным проектом, но в 2008 году произошло слияние и сейчас WebDriver представляет собой основной вектор развития проекта Selenium.

3.3. Selenium Server

Selenium Server – это сервер, который позволяет управлять браузером с удалённой машины, по сети. Сначала на той машине, где должен работать браузер, устанавливается и запускается сервер. Затем на другой машине (технически можно и на той же самой, конечно) запускается программа, которая, используя специальный драйвер RemoteWebDriver, соединяется с сервером и отправляет ему команды. Он в свою

очередь запускает браузер и выполняет в нём эти команды, используя драйвер, соответствующий этому браузеру:

Selenium Server поддерживает одновременно два набора команд – для новой версии (WebDriver) и для старой версии (Selenium RC).

3.4. Selenium Grid

Selenium Grid – это кластер, состоящий из нескольких Selenium-серверов. Он предназначен для организации распределённой сети, позволяющей параллельно запускать много браузеров на большом количестве машин.

Selenium Grid имеет топологию «звезда», то есть в его составе имеется выделенный сервер, который носит название «хаб» или «коммутатор», а остальные сервера называются «ноды» или «узлы». Сеть может быть гетерогенной, то есть коммутатор и узлы могут работать под управлением разных операционных систем, на них могут быть установлены разные браузеры. Одна из задач Selenium Grid заключается в том, чтобы «подбирать» подходящий узел, когда во время старта браузера указываются требования к нему – тип браузера, версия, операционная система, архитектура процессора и ряд других атрибутов.

Ранее Selenium Grid был самостоятельным продуктом. Сейчас физически продукт один – Selenium Server, но у него есть несколько режимов запуска: он может работать как самостоятельный сервер, как коммутатор кластера, либо как узел кластера, это определяется параметрами запуска.

3.5. Selenium IDE

Selenium IDE – плагин к браузеру Firefox, который может записывать действия пользователя, воспроизводить их, а также генерировать код для WebDriver или Selenium RC, в котором выполняются те же самые действия. В общем, это «Selenium-рекордер».

Тестировщики, которые не умеют (или не хотят) программировать, используют Selenium IDE как самостоятельный продукт, без преобразования записанных сценариев в программный код. Это, конечно, не позволяет разрабатывать достаточно сложные тестовые наборы, но некоторым хватает и простых линейных сценариев.

Таким образом, наиболее оптимальным решением является Selenium WebDriver, так как сценарий поведения всех сборщиков данных будет одинаковым.

Литература

1. Википедия. API. [Электронный ресурс]. Режим доступа: https://ru.wikipedia.org/wiki/API/ (дата обращения: 08.12.2016).
2. Веб-сайт ВКонтакте. Работа с API. [Электронный ресурс]. Режим доступа: https://vk.com/dev/apiusage/ (дата обращения: 07.12.2016).
3. Stanford University. Web scraping with Beautiful Soup. [Electronic resource] URL: https://ru.wikipedia.org/wiki/Selenium/ (date of access: 08.12.2016).
4. Википедия. Selenium. [Электронный ресурс]. Режим доступа: https://ru.wikipedia.org/wiki/Selenium/ (дата обращения: 06.12.2016).