Знакомство с Web Feature Service

Обсудить в форуме Комментариев — 0

Данная статья представляет собой введение в протокол OGC Web Feature Service. Рассматривается версия протокола 1.1.0. Статья написана по мотивам руководства, расположенного на сайте OGC.

Содержание

- 1 Введение
- 2 GET и POST запросы
- 3 Доступные методы
  - 3.1 GetCapabilities
  - 3.2 DescribeFeatureType
  - 3.3 GetFeature
- 4 WFS-серверы и клиенты
- 5 Заключение

Введение

Предположим, у нас имеется большой объём обновляемых данных, например, данные по тепловым аномалиям на территорию Российской Федерации, хранящиеся в пространственной базе данных PostGIS и какой-нибудь сторонний разработчик принимает решение использовать их в своем приложении. Прежде всего он вынужден будет реализовать все необходимые операции (извлечение, фильтрацию и т.п.), в терминах того хранилища к которому он подключен (в нашем случае в качестве таких терминов выступают операторы языка SQL). Если мы вдруг решим сменить способ хранения (например, перейдем на файловый вариант) или появится более качественный источник аналогичных данных, но в другом формате и разработчику нужно будет переключиться на него, то он будет вынужден полностью переписывать ту часть своего приложения, которая отвечает за извлечение данных из хранилища. Так вот основная задача WFS-сервисов - это предоставление разработчикам (или конечным потребителям данных) универсального интерфейса доступа к пространственным данным, убирающего необходимость прямого доступа к хранилищу и таким образом делающего его прозрачным для пользователя. В этом случае задача по реализации доступа к различным хранилищам данных ложится на плечи WFS-сервера, то есть WFS сервер выступает в качестве прослойки между хранилищем данных и пользователем данных (к пользователям данных могут относится и в том числе и программы). Стандарт предполагает, что на выходе WFS-сервис отдает данные в формате Geography Markup Language (GML)[1], но это нисколько не запрещает на уровне WFS-сервера реализовать поддержку любого другого выходного формата.

Важно отличать Web Feature Service (WFS) от Web Map Service (WMS). В то время как WMS предназначен для передачи уже отрендеренных данных, WFS отдаёт данные в оригинальном виде. Можно провести аналогию: скомпилированная программа (WMS) и исходный код (WFS).

На сегодняшний день существует 2 версии WFS - 1.1.0 и 2.0. Обе версии считаются действующими, но несколько отличаются друг от друга. В рамках данной статьи будет рассмотрена работа с WFS 1.1.0.

GET и POST запросы

Здесь и далее мы не будем приводить ответы WFS-сервера, а будем использовать для этого онлайн-сервис hurl, позволяющий выполнять HTTP-запросы различными методами и делиться результатами ответов. Подробнее о сервисе можно прочитать на хабре.

Существует два способа получения данных с WFS-сервера: HTTP GET и HTTP POST. При GET-запросе данные запроса передаются с помощью строки URL:

```
http://wfsserver?key1=value1&ke2=value2
```

Представленный запрос формируется путём добавления параметров запроса в виде *?ключ1=значение1&ключ2=значение2&...* к адресу WFS-сервера. Часть этих самых ключей и допустимых значений определены в стандарте WFS[2], а часть, так называемая vendor-specific, реализована в рамках того или иного WFS-сервера, но при этом отсутствует в стандарте. Поскольку официально на длину URL не накладывается никаких ограничений, то с помощью метода GET можно выполнять WFS-запросы любой длины. К преимуществам такого способа взаимодействия с WFS-сервером можно отнести такой момент: в этом случае пользователь может вставить строку запроса непосредственно в адресную строку браузера или поделиться ей с кем-нибудь, кому будут нужны те же самые данные (например, отфильтрованные по тому или иному признаку).

POST-запросы - это более мощный инструмент, позволяющий передавать параметры запроса в формате XML в теле самого запроса. Любой GET-запрос к WFS-серверу можно описать в формате POST (обратное утверждение не верно). Значение заголовка Content-Type при таком запросе должен быть согласно стандарту установлен в значение *text/xml*.

Отметим, что в теле POST запроса можно передавать не только XML документ, но и набор вида "ключ1=значение1&ключ2=значение2..." Пример такого запроса доступен здесь. Значение заголовка Content-Type при таком запросе должен быть согласно стандарту установлен в значение *application/x-www-form-urlencoded*.

Доступные методы

Стандарт WFS описывает 6 методов, каждый из которых отвечает за выполнение той или иной операции:

- GetCapabilities
- DescribeFeatureType
- GetFeature
- GetGMLObject
- Transaction
- LockFeature

Поскольку мы рассматриваем базовые возможности WFS, то остановимся на рассмотрении только первых трёх методов: GetCapabilities, DescribeFeatureType и GetFeature. GetGMLObject - довольно сложная штука и вряд ли вы с ней столкнётесь, а последние два метода Transaction и LockFeature предназначены для выполнения операций по редактированию объектов.

GetCapabilities

Для клиентов, в первый раз подключающихся к WFS-серверу, необходимо знать возможности этого сервера (доступные слои, поддерживаемые функции фильтрации и т.п.). Запрос GetCapabilities позволяет осуществить запрос такой информации.

Пример запроса GetCapabilities с помощью метода GET:

```
http://demo.opengeo.org/geoserver/wfs?service=wfs&version=1.1.0&request=GetCapabilities
```

Пример запроса GetCapabilities с помощью метода POST.

В данных примерах WFS-серверу было передано 3 параметра: "service=wfs", "version=1.1.0", and "request=GetCapabilities". Это три обязательных параметра, которые должны присутствовать в любом WFS-запросе (service, version, request). Иногда WFS-сервер ослабляет это требование, но официально - это три обязательных параметра, поэтому следует всегда включать их в свои запросы. Параметр service сообщает WFS-серверу, что выполняется WFS-запрос, version - передаёт информацию о версии сервиса, request - запрашиваемый метод.

Документ, полученный в ответ на запрос GetCapabilities, представляет собой длинный и сложный XML-документ, но он очень важный. Стоит отметить, что GetCapabilities ответ версии 1.0.0 (устаревший вариант) значительно отличается от ответа GetCapabilities версии 1.1.0. Рассмотрим 5 основных разделов этого документа:

- ServiceIdentification - базовая информацию о WFS-сервисе, в частности, поддерживаемые версии;
- ServiceProvider - контактная информация о компании, стоящей за данным WFS-сервисом: телефон, сайт, email;
- OperationsMetadata - описаны поддерживаемые WFS-сервером методы и их параметры, в частности, форматы выходных данных;
- FeatureTypeList - список слоёв, включая используемую проекцию, а также параметры ограничивающего прямоугольника;
- Filter_Capabilities - список поддерживаемых фильтров, доступных при запросе данных.

DescribeFeatureType

В ходе работы может потребоваться узнать дополнительную информацию об отдельных слоях, опубликованных по WFS. Для этого предназначен метод DescribeFeatureType. Данный метод возвращает список доступных слоёв или описание атрибутов, если в запросе передано имя конкретного слоя.

Пример запроса DescribeFeatureType методом GET:

```
http://demo.opengeo.org/geoserver/wfs?service=wfs&version=1.1.0&request=DescribeFeatureType
```

Пример запроса DescribeFeatureType методом POST.

Данный запрос вернёт список слоёв, сгруппированных по значению атрибута namespace. Если мы хотим получить описание конкретного слоя, то запрос будет выглядеть следующим образом.

Пример запроса DescribeFeatureType с указанием конкретного слоя методом GET:

```
http://demo.opengeo.org/geoserver/wfs?service=wfs&version=1.1.0&request=DescribeFeatureType&typeName=topp:states
```

Пример запроса DescribeFeatureType с указанием конкретного слоя методом POST.

Легко запутаться при чтении XML-документа, но на самом деле здесь не так много информации. Из полученного ответа можно заключить, что слой topp:states содержит 23 атрибута, один из которых представляет описание геометрии, четыре текстовых и восемнадцать числовых атрибутов. Кроме того, в ответе представлены имена атрибутов.

GetFeature

Описаные выше два метода конечно полезны, однако наиболее важный метод ради которого и был разработан стандарт WFS - метод GetFeature, позволяющий запрашивать с сервера непосредственно сами данные. Данный метод необходимо использовать совместно с параметром typeName, определяющем имя слоя, объекты которого мы запрашиваем. Пример запроса GetFeature методом GET:

```
http://demo.opengeo.org/geoserver/wfs?service=wfs&version=1.1.0&request=GetFeature&typeName=topp:states
```

Пример запроса GetFeature методом POST.

Если необходимо запросить информацию об одном объекте, то зная его ID (можно посмотреть в ответе предыдущего запроса), это можно сделать следующим способом (используя опциональный параметр featureID), с помощью метода GET:

```
http://demo.opengeo.org/geoserver/wfs?service=wfs&version=1.1.0&request=GetFeature&type
Name=topp:states&featureID=states.17
```

или с помощью метода POST.

Также, используя параметр maxFeatures можно ограничить количество возвращаемых объектов: POST, GET:

```
http://demo.opengeo.org/geoserver/wfs?service=wfs&version=1.1.0&request=GetFeature&type
Name=topp:states&maxFeatures=2
```

Ограничить список возвращаемых атрибутов можно с помощью параметра propertyName, передавая значения имен атрибутов через запятую, а отсортировать полученный результат с помощью параметра sortBy (sortBy=value+D - по убыванию, sortBy=value+A - по возрастанию). Примеры запроса, комбинирующего эти два параметра: POST, GET:

```
http://demo.opengeo.org/geoserver/wfs?service=wfs&version=1.1.0&request=GetFeature&type
Name=topp:states&sortBy=FAMILIES+D&propertyName=FAMILIES,MALE
```

Вообще для описания формата фильтрации данных существует отдельный стандарт[3], описывающий порядок описания фильтров с использованием XML. В частности в нём описан способ фильтрации данных по охвату (BBOX). Однако такой же параметр введен и в спецификацию WFS для удобства, как сокращенный вариант более общего способа.

Пример запроса данных по охвату: методом GET:

```
http://demo.opengeo.org/geoserver/wfs?service=wfs&version=1.1.0&request=GetFeature&type
Name=topp:states&propertyName=FAMILIES&bbox=40,-79,42,-71
```

WFS-серверы и клиенты

На сегодняшний день существует различные сервера, позволяющие публиковать данные по протоколу WFS. Здесь стоит отметить GeoServer и TinyOWS, влившийся в состав небезызвестного MapServer, как наиболее ярких представителей свободного ПО данного класса.

Поддержка WFS также реализована в различных ГИС, как в настольных, так и в Web. К первым можно отнести, например, QGIS, а к последним OpenLayers. Популярная библиотека OGR также имеет в своём составе драйвер WFS.

Заключение

Мы попытались обосновать необходимость стандарта WFS, а также описали некоторые способы его использования. Естественно, что в рамках данной статьи невозможно рассмотреть все вопросы, связанные с WFS, поэтому в случае возникновения каких-то затруднений, ответы на вопросы следует искать прежде всего в спецификации [2].

Также для более полной картины рекомендуются к прочтению следующие материалы: Сервисы доставки данных OWS и Сервисы OWS. Часть 2. (Или почему WMS = Интернет)

1. ↑ Geography Markup Language
2. ↑ [2,0] [2,1] Web Feature Service 1.1.0 Implementation Specification
3. ↑ OpenGIS Filter Encoding 2.0 Encoding Standard

Обсудить в форуме Комментариев — 0

Последнее обновление: 2014-05-15 01:40

Дата создания: 16.08.2012

Автор(ы): Денис Рыков