

Рассматриваются вопросы динамической выборки из массива данных определенной части. Описывается решение, стандартизированное в спецификации OGC WMS, и частные реализации в программах GeoServer, UMN MapServer, QGIS MapServer. Не затрагивается раздел работы с временными отрезками. Приводятся методы работы с фильтрами в javascript-библиотеке OpenLayers.

[Обсудить в форуме](#) Комментариев — 11

Оглавление

1. [Спецификация OGC WMS](#)
2. [Использование](#)
3. [GeoServer](#)
4. [UMN MapServer](#)
5. [OpenLayers](#)

1. Спецификация OGC WMS

Текущая редакция стандарта [OGC Web Map Service Interface](#) 1.3.0 (20.01.2004) предполагает настройку сервиса в реальном времени посредством отправки документов SLD. Фильтры можно задать в двух случаях: LayerFeatureConstraints для всего слоя (SLD 1.1.0, раздел 11.2 «Named layers») и для каждого правила (Rule) в стилях (SE 1.1.0, раздел 10 «Rules»).

Примечание: правила составления фильтров описываются в спецификации [Filter Encoding Implementation](#).

Во втором случае данные из хранилища (например, БД или WFS) запрашиваются в полном объеме. А после применения стилей часть объектов может быть исключена, т.к. они не вошли ни в одно из правил. Накладные расходы на подготовку и транспортировку данных очевидны, но проект **UMN MapServer** пошел именно по этому пути (см. ниже).

Если же задать LayerFeatureConstraints для слоя, то запрос в хранилище будет производиться с учетом фильтра и лишние данные не будут включены в результат.

Обратимся к профилю [Styled Layer Descriptor](#) (версия 1.1.0 от 29.06.2007). Пример фильтра слоя выглядит так (описание namespace опущено):

```
<sld:StyledLayerDescriptor version="1.1.0">
  <sld:NamedLayer>
    <se:Name>mylayer</se:Name>

    <!-- фильтр слоя -->
    <sld:LayerFeatureConstraints>
      <sld:FeatureTypeConstraint>
        <ogc:Filter>
          <ogc:PropertyIsEqualTo>
            <ogc:PropertyName>country</ogc:PropertyName>
            <ogc:Literal>UKRAINE</ogc:Literal>
          </ogc:PropertyIsEqualTo>
        </ogc:Filter>
      </sld:FeatureTypeConstraint>
    </sld:LayerFeatureConstraints>

    <sld:UserStyle>
      <!-- описание стилей (rules) -->
    </sld:UserStyle>
  </sld:NamedLayer>
</sld:StyledLayerDescriptor>
```

Полная реализация LayerFeatureConstraints есть в **QGIS MapServer**.

В программе **UMN MapServer** фильтр слоя напрямую не поддерживается, но если источником данных является БД, то выполняется довольно спорная оптимизация ([ticket #2840](#)): фильтры из всех rules объединяются в один общий, который используется при обращении к источникам данных. Если же вы хотите установить фильтр слоя другими методами (см. ниже), то такое поведение будет сильно мешать. Отключить эту оптимизацию можно только в исходном коде с перекомпиляцией (или применить бинарный патч).

GeoServer использует библиотеку GeoTools, в которой реализация LayerFeatureConstraints присутствует. (В реальной жизни не проверял.)

2. Использование

Подготовленный документ SLD сохраняется в файл, доступный для сервера WMS по протоколу HTTP. В строке запроса указывается полная ссылка в параметре SLD (одной строкой):

```
http://wms-server/wms?SERVICE=WMS&
LAYERS=mylayer&
SLD=http://other-server/layer_with_filter.sld
```

Заметьте, что имя слоя в параметре LAYERS и в документе SLD должно совпадать (в примере и там и там 'mylayer').

При работе в OpenLayers составляют условие фильтра и передают серверному скрипту (например, php или python) который оформляет документ SLD и сохраняет в виде файла. Как вариант, можно на результат сразу на сервер WMS:

```
http://wms-server/wms?SERVICE=WMS&
LAYERS=mylayer&
SLD=http://other-server/style_filter.php?country#UKRAINE
```

Если отбросить решаемые трудности с кодированием строки, то нужно отметить только увеличение длины строки запроса и накладные расходы на постоянную работу скрипта (будет выполняться при рендеринге каждого тайла). Для несложных фильтров этот метод может быть удобен.

Применение фильтра к слою:

```
lay_points.mergeNewParams({
  sld: 'http://other-server/layer_with_filter.sld'
});
```

При добавлении (обновлении) параметров слоя он автоматически перерисовывается.

Примечание: можно, также, передать все содержимое документа в параметре SLD_BODY. Но так как длина строки ограничена (зависит от браузера, 500-2000 знаков), то практического применения этот способ не нашел.

3. GeoServer

Использование SLD позволяет создавать веб-приложения, не зависящие от используемого на сервере WMS программного обеспечения. Но иногда бывает удобнее и проще воспользоваться дополнительными функциями (vendor feature) конкретного сервера.

Common Query Language

В GeoServer реализован собственный механизм фильтров слоя — CQL. В новой версии (см. [пресс-релиз GeoServer 2.1 beta 3](#)) обещают его расширенный вариант ECQL. Язык CQL введен в спецификации [OGC Catalogue Services](#) (CS 2.0.2, раздел 6.2.2). В тексте дано исключительно формальное описание, требующее для понимания определенной подготовки. Поэтому рекомендую обратиться к документации [GeoServer](#) и [GeoTools](#).

OGC FE разрабатывается на основе CQL и по возможностям оба языка равны. В CQL присутствуют функции пространственных отношений (BBOX, пересечение и т.п.) и временные выборки.

Примеры фильтров:

```
city_name LIKE 'A%' AND population > 100000  
country IN ('UKRAINE', 'BELARUS')
```

При использовании с сервисами WMS в строку запроса добавляется параметр CQL_FILTER:

```
http://wms-server/wms?SERVICE=WMS&  
LAYERS=mylayer&  
CQL_FILTER="country IN ('UKRAINE', 'BELARUS')"
```

Аналогично для OpenLayers:

```
lay_points.mergeNewParams({  
  CQL_FILTER: "country IN ('UKRAINE', 'BELARUS')"  
});
```

OGC Filter

GeoServer позволяет применять фильтры OGC напрямую. В параметре FILTER запроса передается описание фильтра в формате XML. Пример из документации:

```
http://localhost:8080/geoserver/wms/kml_reflect?  
layers=topp:states&  
FILTER=%3CFilter%3E%3CPropertyIsBetween%3E%3CPropertyName%3Etopp:LAND_KM%3C/PropertyName%3E%3CLowerBoundary%3E%3CLiteral%3E100000%3C/Literal%3E%3C/LowerBoundary%3E%3CUpperBoundary%3E%3CLiteral%3E150000%3C/Literal%3E%3C/UpperBoundary%3E%3C/PropertyIsBetween%3E%3C/Filter%3E
```

Строка закодирована по методу Quoted-printable и эквивалентна:

```
<Filter>  
  <PropertyIsBetween>  
    <PropertyName>topp:LAND_KM</PropertyName>  
    <LowerBoundary>  
      <Literal>100000</Literal>  
    </LowerBoundary>  
    <UpperBoundary>  
      <Literal>150000</Literal>  
    </UpperBoundary>  
  </PropertyIsBetween>  
</Filter>
```

Как правило, кодировать вручную не требуется, браузер выполняет эту работу самостоятельно. Но если понадобится раскодировать, то рекомендую программу Штирлиц. (Последняя версия 4.01 вышла в 2001 году, но легко находится в интернете.)

featureid

Редко используемый фильтр, предназначен для выбора объекта по известному идентификатору:

```
http://localhost:8080/geoserver/wms/kml_reflect?
layers=topp:states&featureid=states.5
```

4. UMN MapServer

Стационарный фильтр слоя задается в файле настроек [ключевым словом FILTER](#):

```
LAYER
  NAME "mylayer"
  FILTER (type='road' and size<2)
  ...
END # layer
```

Полный список возможных логических конструкций приведен в документации в разделе [«MapServer expressions»](#).

Run-time Substitution

Что бы изменять параметры динамически (см. главу [«Run-time Substitution»](#)), введем в фильтр переменные и опишем информацию, для проверки на корректность:

```
LAYER
  ...
  FILTER ( ("region" = %country%) AND ("size" < %size%) )
  ...
  METADATA
    country_validation_pattern '.'
    size_validation_pattern '^0-9{1,}$'
  END
END
```

Две переменные — %country% и %size%. Соответствующие им строки проверки задаются в форме Regular expression (RegExp): country — любое значение, size — только цифры. В документации строго не рекомендуют использовать RegExp '.', так как это создает опасность проведения атаки. Но в принципе, никто не запрещает использовать универсальный фильтр типа:

```
LAYER
  ...
  FILTER (%myfilter%)
  ...
  METADATA
    myfilter_validation_pattern '.'
  END
END
```

Строка запроса к серверу WMS принимает вид:

```
http://wms-server/wms?SERVICE=WMS&
LAYERS=mylayer&
country='UKRAINE'&
size='100000'
```

Есть и более общий механизм изменения любых параметров файла настроек. Он очень похож на описанный выше:

```
LAYER
...
VALIDATION
    'filter' '.'
END
END
```

Используется группа VALIDATION, в которой указывается имя параметра и строка проверки. Как я указал выше изменить можно любой параметр, имя которого внесено в группу. Это позволяет изменять, например, строку подключения к базе данных. На всякий случай приведу познавательную ссылку - [SQL injection](#)).

В строку запроса на сервер добавляются конструкция вида (пробелы можно заменять на символ '+', при необходимости):

```
&map.layer0=FILTER("country" = "UKRAINE")
&map.layer0.class0=EXPRESSION("country" = "BELARUS")
&map.layer0=DATA+"the_geom from xxx using unique gid using srid=4326"
```

MapScript

Пожалуй, наиболее удобный вариант изменения фильтра — это использование скрипта на одном из диалектов [MapScript](#) (PHP, Python, Perl и др.). Тем более, что при работе с этим сервером создание вращпера желательно в любом случае.

Пример для PHP/MapScript:

```
<?php
#dl('php_mapscript.so');

# создаем карту
$map = ms_newMapobj('/usr/local/example.map');
# загружаем переданные параметры
$request = ms_newowsrequestobj();
$request->loadparams();
# задаем фильтр
$layer = $map->getLayerByName('mylayer');
$layer->setFilter($request->getValueByName('MY_FILTER'));

# рисуем карту
ms_ioinstallstdouttobuffer();
$map->owsdispatch($request);
$contenttype = ms_iostripstdoutbuffercontenttype();
if ($contenttype == 'image/png')
    header('Content-type: image/png');
ms_iogetStdoutBufferBytes();
ms_ioresethandlers();
?>
```

5. OpenLayers

В библиотеке есть встроенная поддержка работы с фильтрами в формате OGC — набор классов [OpenLayers.Filter](#).

Простое сравнение country = 'UKRAINE':

```
var filter = new OpenLayers.Filter.Comparison({
    type: OpenLayers.Filter.Comparison.EQUAL_TO,
    property: 'country',
    value: 'UKRAINE',
```

```
});
```

Преобразование в XML:

```
var format = new OpenLayers.Format.Filter({version: "1.1.0"});  
var xml = new OpenLayers.Format.XML();  
var text = xml.write(format.write(filter));
```

В версии для разработчиков (trunk) уже присутствует парсер [OpenLayers.Format.CQL](#). Работа с ним аналогична рассмотренному выше.

Стили

Вот так выглядит точечный стиль с логическим условием (country = 'UKRAINE') OR (country = 'BELARUS'):

```
var style = new OpenLayers.Style();  
  
var rule = new OpenLayers.Rule({  
    name: species_selectedi,  
    filter: new OpenLayers.Filter.Logical({  
        type: OpenLayers.Filter.Logical.OR,  
        filters:  
            new OpenLayers.Filter.Comparison({  
                type: OpenLayers.Filter.Comparison.EQUAL_TO,  
                property: 'country',  
                value: 'UKRAINE',  
            }},  
            new OpenLayers.Filter.Comparison({  
                type: OpenLayers.Filter.Comparison.EQUAL_TO,  
                property: 'country',  
                value: 'BELARUS',  
            })  
        },  
    symbolizer: {  
        "Point": {  
            graphicName: "circle", pointRadius: 3.5,  
            strokeWidth: 1, strokeColor: "#000000",  
            fillColor: palettex, fillOpacity: 0.5  
        }  
    }  
});  
  
style.addRules(rule);
```

Преобразование стиля в формат SLD (LayerFeatureConstraints не поддерживается):

```
var text = new OpenLayers.Format.SLD().write({  
    namedLayers: {  
        name: "mylayer",  
        userStyles: style  
    }  
});
```

И в заключение, небольшая справка по подстановочным символам (wild-card):

	один любой символ	любое количество любых символов	экранирование (escape)	Примечание
--	-------------------	---------------------------------	------------------------	------------

SLD (FE)	. (точка)	*	!	Переопределяется в singleChar, wildCard, escapeChar
UMN MapServer	. (точка)	*	\	Про правилам RegExp
CQL	—	%	\	
SQL	—	%		экранирование задается параметром escape

[Обсудить в форуме](#) Комментариев — 11

Ссылки по теме

- [Начало работы с GeoServer](#)
- [Начало работы с MapServer](#)
- [Реализация определяющей выборки в Mapserver](#)
- [Выражения в MapServer](#)

Последнее обновление: February 08 2011

Дата создания: 29.12.2010

Автор(ы): [Mavka](#)