

Поиск объектов на карте с помощью OpenLayers

Рассмотрен вариант организации поиска и подсветки найденных объектов с помощью OpenLayers.

[Обсудить в форуме](#) Комментариев — 32

Поиск объектов по атрибутике и масштабирование на них является распространенной задачей при создании как настольных ГИС, так и веб-сервисов. Эта статья рассматривает одно из возможных решений для веб-приложения.

Оглавление

1. [Введение](#)
2. [Настройка WMS сервера](#)
3. [Размещение основных элементов на странице](#)
4. [Определение охвата найденных объектов](#)
5. [Результат](#)

1. Введение

Предположим, что у нас есть векторный слой данных, который бы мы хотели опубликовать в Интернет и предоставить пользователю возможность осуществлять поиск объектов по его атрибутике. Для примера воспользуемся слоем населенных пунктов **settlements** из открытого набора данных [Geosample](#), который хранится в базе данных PostGIS. Будем осуществлять поиск объектов по полю "name".

2. Настройка WMS сервера

OpenLayers не позволяет визуализировать данные непосредственно из PostGIS, выходом из этой ситуации является опубликование слоя в качестве WMS-сервиса. Воспользуемся таким сервисом, созданным в ходе работы над проектом Geosample: <http://gis-lab.info:8180/geoserver/wms>, среди слоёв которого присутствует и необходимый нам **settlements**. В нашем примере мы будем использовать его в качестве подложки, также подключенной как WMS, которая изменяться никоим образом не будет. Вы можете использовать свою подложку.

Для подсветки найденных объектов создадим собственный WMS-сервис на базе MapServer, содержащий один слой, который будет формироваться на основе тех же данных (PostGIS), что и WMS-сервис из геосемпла, но динамически, в ходе выполнения запроса, и содержать контура найденных объектов, обведенные красным цветом.

Содержимое map-файла **layer.map**:

```
MAP
  STATUS          ON
  IMAGETYPE       PNG
  EXTENT          82 54 84 56
  SIZE            400 300
  FONTSET         "fonts/fonts.list"

  WEB
    IMAGEPATH     "/usr/local/www/gis-lab/data/tmp/"
    IMAGEURL      "/tmp/"
    METADATA
      wms_title           "OpenLayers search example"
      wms_abstract        "OpenLayers search example"
      wms_srs             "EPSG:4326"
      wms_feature_info_mime_type "text/html"
    END
  END
  PROJECTION
    "init=epsg:4326"
  END
```

```

LAYER
  NAME "selected"
  CONNECTIONTYPE postgis
  CONNECTION "user=guest password=guest dbname=geosample host=gis-lab.info"
  DATA "the_geom from (%SQL%) as newtable using unique gid using srid=4326"
  TYPE POLYGON
  CLASS
    STYLE
      OUTLINECOLOR 255 0 0
      ANTIALIAS TRUE
      WIDTH 2
    END
  END
  METADATA
    wms_srs "EPSG:4326"
  END
END
END

```

Более подробное описание работы MapServer с данными PosGIS можно найти в статье [Визуализация данных PostGIS в MapServer](#).

3. Размещение основных элементов

После того, как завершена настройка WMS сервера, переходим к настройке пользовательского интерфейса, представляющего собой HTML-страницу. Возможность выбора атрибута, в нашем случае названия населенного пункта (поле "name"), может предоставляться пользователю, например, посредством выпадающего списка, организованного следующим образом:

```

<form name='form'><select name='name'>
<option value=''>Select one</option>
<option value=2200800000200 >Акутиха</option>
<option value=22000000200000 >Алейск</option>
<option value=2200300000100 >Алтайский</option>
<option value=2204700000300 >Алтайское</option>
...

```

где "value" - КЛАДР код населенного пункта (поле "code" слоя settlements). Построение подобного списка легко автоматизировать, например, с помощью PHP. После размещения на странице выпадающего списка, добавим кнопку по нажатию которой будет осуществляться поиск и непосредственно карту:

```

<html>
<head>
  <script src="http://openlayers.org/api/OpenLayers.js"></script>
  <script src="http://gis-lab.info/programs/js/ol-search.js"></script>
  ...
</head>
<body onLoad="init()">
  <div id="map"></div>
  <input type="button" value="Search" onclick="gettext()">
  ...

```

<http://openlayers.org/api/OpenLayers.js> - ссылка на библиотеку OpenLayers, **ol-search.js** - файл, содержащий 3 функции:

- **init()** - инициализации карты;
- **init AJAX()** - создание экземпляра объекта XMLHttpRequest;
- **gettext()** - функция, отвечающая за поиск объектов.

Содержимое файла **ol-search.js**:

```
function init() {
```

```

map = new OpenLayers.Map("map");

sel = new OpenLayers.Layer.WMS(
    "Selected",
    "http://gis-lab.info/cgi-bin/mapserv?",
    {map: "/usr/local/www/gis-lab/data/programs/mapserver/ol-
search/layer.map", layers: 'selected', sql: "select * from settlements where name='-
1'", transparent: "true"}, {singleTile: true});

geosample = new OpenLayers.Layer.WMS(
    "Selected",
    "http://gis-lab.info:8180/geoserver/wms",
    {layers: "geosample:ecoregions,geosample:road-1-
osm,geosample:settlements", transparent: "false"});

map.addLayers(geosample, sel);

var bounds = new OpenLayers.Bounds.fromString('82,54,84,56');

map.zoomToExtent(bounds);

}

function init_AJAX() {
try { return new ActiveXObject("Msxml2.XMLHTTP"); } catch(e) {} //IE
try { return new ActiveXObject("Microsoft.XMLHTTP"); } catch(e) {} //IE
try { return new XMLHttpRequest(); } catch(e) {} //Native Javascript
alert("XMLHttpRequest not supported");
return null;
};

function gettext() {
var req = init_AJAX();

req.open("GET", "http://gis-lab.info/programs/php/ol-search-
gettext.php?name="+encodeURIComponent(document.form.name.optionsdocument.form.name.optio
ns.selectedIndex.value));

req.onreadystatechange = function () {

if (req.readyState==4) {

```

```

        if (req.status==200) {

            var response = req.responseText.split("|");

            sel.mergeNewParams({sql:response0});

            var bounds = new OpenLayers.Bounds.fromString(response1);

            map.zoomToExtent(bounds);

        }

    }

};

req.send(null);

}

```

Рассмотрим содержимое данного файла более подробно.

init()

Создаём экземпляр карты:

```
map = new OpenLayers.Map("map");
```

Затем инициализируем слой населенных пунктов (переменная **geosample**) и слой обводки объектов (переменная **sel**) изначально пустой, поскольку в качестве подстановочного поля %SQL% серверу передается выражение *"select * from settlements where name='-1'"*, которому, очевидно, в таблице **settlements** не удовлетворяет ни одна запись:

```

geosample = new OpenLayers.Layer.WMS(...);
sel = new OpenLayers.Layer.WMS(...,{...,sql:"select * from settlements where name='-1'"},...);

```

Важное замечание: при пересылке параметров серверу OpenLayers переводит все параметры в верхний регистр (map, layers, sql, transparent будут переданы как MAP, LAYERS, SQL, TRANSPARENT), поэтому в качестве подстановочного поля в map-файле используется %SQL%, а не %sql%.

Добавляем слои на карту и устанавливаем нужный охват:

```

map.addLayers(geosample, sel);
var bounds = new OpenLayers.Bounds.fromString('82,54,84,56');
map.zoomToExtent(bounds);

```

init AJAX()

Данная функция отвечает за создание экземпляра объекта XMLHttpRequest способом, определяемым типом браузера. XMLHttpRequest — набор API, используемый в языках JavaScript, JScript, VBScript и им подобных для пересылки различных данных по HTTP-протоколу между браузером и веб-сервером. Позволяет осуществлять HTTP-запросы к удаленному серверу без необходимости перезагружать страницу.

gettext()

Создаём экземпляр объекта XMLHttpRequest:

```
var req = init_AJAX();
```

и передаём асинхронный запрос методом GET серверному скрипту **gettext.php**:

```
req.open("GET", "http://gis-lab.info/programs/php/ol-search-  
gettext.php?name="+encodeURIComponent(document.form.name.optionsdocument.form.name.optio  
ns.selectedIndex.value));
```

Асинхронность заключается в том, что клиентское приложение не дожидается ответа от сервера, а продолжает свою работу. Как только с сервера приходит ответ, он заносится в массив **response**. Серверный скрипт написан таким образом, что результатом его выполнения является строка SQL-запроса на поиск выбранного пользователем объекта и пространственный охват этого объекта, разделенные символом "|":

```
if (req.readyState==4) {  
    if (req.status==200) {  
        var response = req.responseText.split("|");  
        ...  
    }  
}  
};
```

После чего слою **sel** передаётся подстановочное поле (которое затем отправляется на WMS сервер, который в свою очередь рендерит слой в соответствии с этим полем, в нашем случае - контура объектов) и устанавливается новый охват, соответствующий охвату найденных объектов:

```
sel.mergeNewParams({sql:response0});  
var bounds = new OpenLayers.Bounds.fromString(response1);  
map.zoomToExtent(bounds);
```

4. Поиск объектов и определение охвата

Рассмотрим, что из себя представляет серверный скрипт, генерирующий подстановочное поле и вычисляющий охват запрашиваемых объектов **gettext.php**:

```
<?php  
  
$dbh = pg_connect("host=gis-lab.info dbname=geosample user=guest password=guest") or  
die("Error in connection: " . pg_last_error());  
  
$code = $_REQUEST'name';  
  
$str = "SELECT * FROM settlements WHERE code='".$code."'";  
echo $str."|";  
  
$result = pg_query($dbh, "select ST_XMin(ST_Union(the_geom)) as st_xmin,  
ST_YMin(ST_Union(the_geom)) as st_ymin,ST_XMax(ST_Union(the_geom)) as st_xmax,  
ST_YMax(ST_Union(the_geom)) as st_ymax from ($str) AS foo");  
$row = pg_fetch_array($result);  
  
if  
(!empty($row'st_xmin')&!empty($row'st_ymin')&!empty($row'st_xmax')&!empty($row'st_ymax'  
)){  
    echo $row'st_xmin'." ".$row'st_ymin'." ".$row'st_xmax'." ".$row'st_ymax';  
}  
else {  
echo "82,54,84,56";  
}  
?>
```

Соединяемся с базой данных и запрашиваем значение переменной из URL:

```
$dbh = pg_connect("host=gis-lab.info dbname=geosample user=guest password=guest") or  
die("Error in connection: " . pg_last_error());  
  
$code = $_REQUEST'name';
```

В соответствии с названием запрашиваемого объекта генерируем SQL запрос и добавляем после него символ "|":

```
$str = "SELECT * FROM settlements WHERE code='".$code."'";  
echo $str."|";
```

Используя функции PostGIS ST_XMin,ST_YMin,ST_XMax,ST_YMax, вычисляем охват запрашиваемого объекта:

```
$result = pg_query($dbh, "select ST_XMin(ST_Union(the_geom)) as st_xmin,  
ST_YMin(ST_Union(the_geom)) as st_ymin,ST_XMax(ST_Union(the_geom)) as st_xmax,  
ST_YMax(ST_Union(the_geom)) as st_ymax from ($str) AS foo");
```

Выводим рассчитанный охват, либо полный охват карты, если не найдены объекты с выбранным именем (например, выбран элемент выпадающего списка Select one):

```
$row = pg_fetch_array($result);  
  
if  
(!empty($row'st_xmin')&!empty($row'st_ymin')&!empty($row'st_xmax')&!empty($row'st_ymax'  
')){  
    echo $row'st_xmin'."", ".$row'st_ymin'."", ".$row'st_xmax'."", ".$row'st_ymax';  
}  
else {  
    echo "82,54,84,56";  
}
```

Примеры результата выполнения **gettext.php**:

```
SELECT * FROM settlements WHERE  
code='2200000800000'|83.8799743652344,53.3685569763184,84.0150527954102,53.470001220703  
1  
SELECT * FROM settlements WHERE  
code='5400900002500'|78.4868316650391,53.5798873901367,78.5268630981445,53.605888366699  
2  
SELECT * FROM settlements WHERE  
code='2202200001200'|82.6611251831055,51.7036590576172,82.7050018310547,51.730140686035  
2
```

Итак подводя итог всему вышесказанному, представим порядок работы нашего приложения:

1. Выбираем из выпадающего списка название населенного пункта, который мы хотим найти на карте;
2. Нажимаем кнопку Search, вызывая функцию gettext();
3. Функция gettext() фоном отправляет запрос серверному скрипту gettext.php;
4. При возврате ответа от gettext.php найденные объекты выделяются и устанавливается новый охват.

5. Результат

Название населенного пункта:

Select one Акутиха Алейск Алтайский Алтайское Андроновое Анжеро-Судженск Антоновка Ануйское Ая Баган
Баетов Банново Барабинск Барановка Барзас Барнаул Бастан Белово Белое Бердск Береговое Беркульский
Бийск Благовещенка Болотное Большая Покровка Большеникольское Большой Изырак Боровиха Боровское
Боярка Брусенцево Букаевское Букреево Плесо Буланых Буланых Бурла Быструха Быстрый Исток Быстрица
Васильевка Васильчуки Вассино Венгерово Верх-Ануйское Верх-Жилино Верх-Ирмень Верх-Катунское Верх-
Коев Верх-Пайва Верх-Суетка Верх-Урюм Верх-Чик Верх-Чуманка Верхотомское Веселая дубрава Воеводское
Володарка Волчиха Волчно-Бурлинское Воробьево Ворониха Вострово Вылково Вьюны Вяткино Гилев Лог
Глубокое Горно-Алтайск Горняк Грановка Гуниха Гурьевск Гусельниково Довольное Долганка Дресвянка
Дубровино Елтышево Енисейское Журавлиха Заводское Завьялово Зайцево Закладное Залесово Заринск
Зеркалы Змеиногорск Знаменка Зональное Зыково Зятково Ижморский Инской Иня Ирбизино Искитим
Итатский Кабинетное Каз Казачий Мыс Каип Калманка Калманский Калмыцкие мысы Калтан Камень-на-Оби
Камышево Камышловка Кандаурово Карасук Каргат Карпово Второе Касмала Каяушка Кебезень Кемерово

Кирза Кировский Кленовка Клепечиха Кожурла Кожух Колпаково Колыванское Колывань Комарье Комсомольск Корнилово Коротояк Корчино Косиха Костин Лог Коченево Кочки Кочки Крапивинский Краснинское Краснобродский Красногорское Красное Краснозерское Краснощеково Крестьянка Крутологово Кузедеево Куйбышев Кулунда Купино Кучук Куяган Кытманово Кыштовка Ларичиха Лебедевка Лебедево Ленинск-Кузнецкий Лесное Листвянка Луговое Майма Макарово Малиновое Озеро Малиновский Маловолчанка Малый Башелак Малышев лог Мальчиха Мариинск Маслянино Медяково Междуреченск Мельниково Михайловское Моховское Мундыбаш Налобиха Неводное Ненинка Нижнекаменка Нижнеозерное Нижнечуманка Нижний Кучук Нижняя Гусиха Николаевка Никулино Новиково Новичиха Новоалександровка Новоалексеевка Новоалтайск Новогорьевское Новоколпаково Новокормиха Новокрасное Новокузнецк Новолуговое Новопестерево Новопесчаное Новоповалиха Новоподзорново Новопокасьма Новоромановка Новороссийка Новоселье Новосибирск Новошипунovo Новоярки Обь Овечкино Овсянниково Огни Октябрьское Окунево Ордынское Осинники Осинное Плесо Осколково Павловск Паклино Панкрушиха Паново Панфилово Парфеново Пенек Первокаменка Петропавловское Пихтовка Подойниково Подсосново Полуторник Полысаево Посевная Пospelиха Притомский Прокопьевск Промышленная Прыганка Разумовка Рассказиха Ребриха Решеты Рогозиха Рождественка Рубцовск Саввушка Салаир Салтымаково Северное Селиверстово Сертинка Сибирячиха Сидоренково Ситниково Славгород Смоленское Советское Солонешное Солтон Сорокино Сrostки Сrostы Станционно-Ояшинский Старобачаты Старочервово Степной Кучук Сузун Сумы Суслово Сухой Ракит Сычевка Сёйка Тараданово Тарасово Татарск Ташара Таштагол Темиртау Терентьевское Тисуль Титовка Тогул Тогулин Топки Топчиха Травное Травное Трещевский Троицкий Тугозвонovo Тулата Тюменцево Тяжино-Вершинка Тяжинский Убинское Ужаниха Урлапово Урск Усть-Барандат Усть-Калманка Усть-Кан Усть-Кокса Усть-Мосиха Усть-Тарка Усть-Чарыш Усть-Чарышская Пристань Усть-Чем Устюжанино Филиппово Харлово Хорошеборка Хорошее Целинное Чаны Чаузово Чеканиха Чемал Чемское Черга Черемное Черепаново Черная Курья Чик Чикман Чистоозерка Чистоозерное Чистюнька Чоя Чулым Чумай Шалап Шарчино Шахи Шахта Шебалино Шипуново Шурыгино Юдиха Юрга Язово Яконур Ярославцев Лог Ячменюха Яшкино Яя

[Обсудить в форуме](#) Комментариев — 32

Ссылки по теме

- [Создание WMS-сервиса с помощью Mapserver](#)
- [Визуализация данных PostGIS в MapServer](#)
- [OpenLayers: начало работы](#)
- [Способы отображения данных с помощью OpenLayers](#)
- [Освоение Ajax: Часть 2. Выполнение асинхронных запросов с JavaScript и Ajax](#)

Последнее обновление: June 07 2010

Дата создания: 24.04.2010

Автор(ы): [Денис Рыков](#)