

# Способы отображения данных с помощью OpenLayers

Несколько способов визуализации пространственных данных с помощью OpenLayers, плюсы и минусы.

[Обсудить в форуме](#) Комментариев — 11

[OpenLayers](#) - свободная библиотека JavaScript для отображения картографических данных на веб-страницах с API сходным с API Google Maps/Virtual Earth.

В зависимости от того, как хранятся исходные данные и требований к создаваемому картографическому представлению в веб, возможно несколько подходов к отображению данных на веб-странице. Эта статья иллюстрирует эти подходы и дает примеры программной реализации каждого из них.

## Оглавление

1. Прямое отображение данных
2. Отображение слоя WMS
3. Загрузка специализированных данных: WMS и Google
4. Загрузка специализированных данных: вектор и Google

## 1. Прямое отображение данных

Способ самый быстрый в развертывании. Заключается в прямом, без посредников, доступе к пространственным данным. OpenLayers позволяет напрямую показывать ограниченное количество векторных и растровых форматов, что является главным недостатком этого метода. К сожалению, на момент написания статьи невозможно работать напрямую с форматом shape, но можно работать с KML, JSON и некоторыми другими. Вторым большим недостатком является невозможность отображения больших наборов данных.

На следующем примере осуществляется прямое подключение к набору данных (граница Республики Калмыкия) в формате KML, хранящемся на сервере и его визуализация средствами OpenLayers.

```
<script type="text/javascript">
map = new OpenLayers.Map('map');
var bounds = new OpenLayers.Bounds(41, 44, 48, 48);
var graphic = new OpenLayers.Layer.GML(
    "OpenLayers KML",
    "../other/kalm.kml",
    {format: OpenLayers.Format.KML};
graphic.setIsBaseLayer(true);
map.addLayers(graphic);
map.zoomToExtent(bounds);
</script>
```

Разберем ключевые фрагменты кода. API OpenLayers очень похож по структуре на аналогичный API Google. Для масштабирования карты на определенном фрагменте, зададим охват переменной bounds. Затем создадим новый слой graphics, для которого будут использоваться данные в формате KML, являющегося разновидностью GML, поэтому [OpenLayers.Layer.GML](#). При отображении данных напрямую, без растровой подложки, нужно указать, что единственный наш слой является базовым (подложкой) - [graphic.setIsBaseLayer\(true\)](#).

## 2. Отображение слоя WMS

На следующем примере осуществляется визуализация данных сервиса WMS, запущенного с помощью MapServer, для сервиса WMS используются данные в формате shape (границы стран). Такой способ отображения данных позволяет работать с большим перечнем векторных форматов, не ограниченным XML/JSON представлением данных, позволяет использовать сложные условные обозначения и осуществлять дополнительную конфигурацию картографического представления данных через редакцию map-файла

MapServer. Так же, используя такой подход, кэши фрагментов (tilecache) и грамотно построенную детализацию зависящую от масштаба практически нет ограничений на объемы отображаемых данных.

К недостаткам данного способа относится относительная сложность настройки, если загрузку необходимо производить с Вашего же сервера WMS, которая включает установку и запуск MapServer, конфигурацию сервиса и т.д. С другой стороны, настроенный сервер WMS позволяет использовать свои данные любому клиентском ПО, как через специализированное ПО ГИС, так и через браузер, как в этом примере.

```
<script type="text/javascript">
var lon = 44;
var lat = 46;
var zoom = 5;
var map, layer;
map = new OpenLayers.Map('map2');
layer = new OpenLayers.Layer.WMS(
    "GIS-Lab Test WMS",
    "http://gis-lab.info/cgi-bin/wmsworld?",
    {layers: 'world'}
);
map.addLayer(layer);
map.setCenter(new OpenLayers.LonLat(lon, lat), zoom);
map.addControl(new OpenLayers.Control.LayerSwitcher());
</script>
```

Разберем ключевые фрагменты кода. Новый объект назовем map2, чтобы он не конфликтовал с уже существующим объектом map, созданным в первом примере. В этом случае слой будет образован сервисом WMS [OpenLayers.Layer.WMS](#). Используем один из наших сервисов WMS с адресом <http://gis-lab.info/cgi-bin/wmsworld?>, зададим ему имя "GIS-Lab Test WMS" и перечислим используемые слои, перечень которых нужно узнать предварительно. Добавим так же для демонстрации новый контрол - переключатель слоев, хотя слой у нас только один, с помощью [OpenLayers.Control.LayerSwitcher](#). Показать подобный набор данных помощью способа №1 можно, однако загрузка слоев займет очень большое время.

### 3. Загрузка специализированных данных: WMS и Google

Еще одной возможностью OpenLayers является загрузка данных предоставляемых через свои API различными картографическими сервисами, например OpenStreetMap, Google Maps и многими другими. Так как реализация их протоколов отлична от спецификаций OGC, они выделены в отдельную группу специализированных данных и управляются с помощью семейства объектов [OpenLayers.Layer.Google](#), [OpenLayers.Layer.Yahoo](#) и т.д. Возможности OpenLayers позволяют загружать такие данные совместно со слоями образованными вышеперечисленными способами, создавая гибридные картографические представления. Отображение слоев из разных источников - одна из главных функций OpenLayers, позволяющего быстро настроить любое их количество. В OpenLayers сделать это гораздо легче, чем скажем напрямую показать слой WMS MapServer в Google Maps ([подробнее](#)). Расплачиваться за это приходится переходом на загрузку самого OpenLayers и невозможностью использовать оригинальный API, заменяемый API OpenLayers.

Простейший пример гибридного представления может быть реализован следующим скриптом, использующим в качестве базового слоя данные сервиса Google (топокарту и спутниковые снимки) и слой-оверлей WMS полученный так же, как в пункте 2.

```
<script
src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAAINgesJb4VJ2iYHIcc -
gT7hQGhS5yCmtMNZMUBKqpbJFW2uH_cRSd35D9AE7VSWY096Cd7dLkCk3bRA" type="text/javascript"
encoding="utf-8"></script>
<script type="text/javascript">
var lon = 44;
var lat = 46;
var zoom = 5;
map = new OpenLayers.Map('map3');
map.addControl(new OpenLayers.Control.LayerSwitcher());
var gmap = new OpenLayers.Layer.Google(
    "Google Streets", // the default
    {numZoomLevels: 20}
```

```

);
var gsat = new OpenLayers.Layer.Google(
    "Google Satellite",
    {type: G_SATELLITE_MAP, numZoomLevels: 20}
);
var wms = new OpenLayers.Layer.WMS(
    "World Map",
    "http://gis-lab.info/cgi-bin/mapserv",
    {
        map: '/usr/local/www/gis-lab/data/programs/mapserver/wms-world-
gmap/wms.map',
        transparent: 'TRUE',
        layers: 'world'
    },
    {'reproject': true}
);
map.addLayers(gsat,gmap,wms);
map.setCenter(new OpenLayers.LonLat(lon, lat), zoom);
map.addControl(new OpenLayers.Control.LayerSwitcher());
</script>

```

Параметр {'reproject': true} в настройки слоя получаемого через WMS необходимо добавить для приведения системы координат данных этого слоя с системой координат сервисов Google (Mercator/Sphere). В противном случае, результат будет выглядеть следующим образом:

## 4. Загрузка специализированных данных: вектор и Google

Наложение векторных слоев в форматах GML, KML, GeoJSON на данные Virtual Earth и Google имеют свои особенности, связанные в первую очередь с проекционными преобразованиями. Для правильного преобразования, необходимо указать систему координат карты и каждого из слоев.

```

<script type="text/javascript">
var lon = 135.0658519;
var lat = 48.466571;
var zoom = 16;
var options = {
    projection: new OpenLayers.Projection("EPSG:900913"),
    displayProjection: new OpenLayers.Projection("EPSG:4326"),
    maxExtent: new OpenLayers.Bounds(-20037508.34, -20037508.34, 20037508.34,
20037508.34)
};
map = new OpenLayers.Map('map5', options);

var layer = new OpenLayers.Layer.Google(
    "GoogleSoft", {
        type: G_SATELLITE_MAP,
        numZoomLevels: 20,
        'sphericalMercator': true
    });
map.addLayer(layer);

var myKML = new OpenLayers.Layer.GML(
    "KML",
    "test.kml", {
        format: OpenLayers.Format.KML,
        projection: map.displayProjection
    });
map.addLayer(myKML);

var proj = new OpenLayers.Projection("EPSG:4326");

```

```
var point = new OpenLayers.LonLat(lon, lat);  
map.setCenter(point.transform(proj, map.getProjectionObject()), zoom);  
</script>
```

[Обсудить в форуме](#) Комментариев — 11

## Ссылки по теме

- [Создание WMS-сервиса с помощью Mapserver](#)
- [OpenLayers: начало работы](#)

Последнее обновление: March 02 2011

Дата создания: 19.02.2009

Автор(ы): [Максим Дубинин](#)