

Использование контрола WMSGetFeatureInfo в OpenLayers

[Обсудить в форуме](#) Комментариев — 3

Эта страница опубликована в основном списке статей сайта по адресу <http://gis-lab.info/qa/wmsgetfeatureinfo.html>

Приведены основные настройки контрола WMSGetFeatureInfo, использующегося в [OpenLayers](#) и показан базовый пример его использования.

Содержание

- [1 Введение](#)
- [2 Настройки WMSGetFeatureInfo](#)
- [3 Пример использования WMSGetFeatureInfo](#)
- [4 Cross-domain security](#)

Введение

[Web Map Service Interface Standard](#) (WMS) предоставляет простой HTTP-интерфейс для передачи клиенту изображений, имеющих географическую привязку. WMS-запрос обязательно содержит наименования запрашиваемого слоя и требуемый охват, а WMS-ответ представляет собой одно или несколько растровых изображений. Кроме того WMS поддерживает опцию прозрачности, позволяющую использовать WMS-слои совместно с другими слоями, не перекрывая их. Любой сервер WMS в обязательном порядке поддерживает два основных запроса:

- GetCapabilities - возвращает описание конкретного сервиса WMS и перечень доступных слоёв
- GetMap - непосредственно запрос изображений.

остальные запросы, описанные в стандарте, поддерживаются факультативно. Среди них есть и GetFeatureInfo, позволяющий запрашивать атрибутивную информацию об объекте, расположенном по указанным координатам. Именно этот тип запроса используется во многих клиентских приложениях и со стороны пользователя может выглядеть, например, как всплывающее окно с информацией об объекте, появляющееся после щелчка на объекте.

Настройки WMSGetFeatureInfo

Краткий перечень наиболее часто используемых настроек контрола:

Свойство	Тип	Описание
hover	{Boolean}	Запрос GetFeatureInfo выполняется при задержке курсора мыши над объектом. По умолчанию <i>false</i> и запрос выполняется только по щелчку.
maxFeatures	{Integer}	Максимальное количество возвращаемых записей, определяет значение параметра FEATURE_COUNT запроса GetFeatureInfo. По умолчанию 10.
layers	{Array(OpenLayers.Layer.WMS)}	Список слоёв, к которым будет выполнен запрос GetFeatureInfo. Если значение не определено, то будут использованы все WMS-слои карты, url которых совпадает со значением свойства url самого контрола или одним из url из массива layerUrls. По умолчанию не определено.

queryVisible {Boolean}

url {String}

infoFormat {String}

vendorParams {Object}

format {OpenLayers.Format}

Если *true*, исключает скрытые слои из запроса GetFeatureInfo. По умолчанию *false*, то есть, если слой присутствует среди слоёв, описанных в массиве *layers*, но при этом выключен на карте, то запрос GetFeatureInfo к этому слою всё равно будет выполнен.

URL сервиса WMS к которому будет осуществляться запрос. Если *url* не задан при создании контрола WMSGetFeatureInfo, то в качестве значения будет использовано свойство *url* первого подходящего слоя. По умолчанию не определено.

MIME-тип ответа, который должен отдать сервер. По умолчанию *text/html*. При использовании в качестве сервера WMS MapServer может быть использован один из трёх типов: *text/plain*, *text/html* или *application/vnd.ogc.gml*.

Дополнительные параметры, которые будут включены в строку запроса. Например, при таком определении данного свойства: {radius: 10} в запрос добавится подстрока "&RADIUS=10"

Формат, используемый для парсинга ответа. По умолчанию OpenLayers.Format.WMSGetFeatureInfo, поэтому для корректного парсинга значение *infoFormat* контрола должно быть установлено в значение *application/vnd.ogc.gml*.

Полный список доступных свойств и методов доступен в официальной [документации](#). Здесь же хотелось бы остановить внимание еще на одном свойстве контрола WMSGetFeatureInfo, наследуемого от OpenLayers.Control - *eventListeners*. Это объект, который позволяет слушать различные события, происходящие с контролом. Для WMSGetFeatureInfo доступно 3 таких события: *beforegetfeatureinfo*, *nogetfeatureinfo*, *getfeatureinfo*.

- *beforegetfeatureinfo* срабатывает перед отправкой запроса на сервер, данное событие удобно использовать, если при каждом запросе к серверу требуется передавать какой-либо изменяющийся параметр. Например:

```
eventListeners: {  
  'beforegetfeatureinfo': function() {  
    this.vendorParams.settings = Math.random();  
  }  
}
```

- *nogetfeatureinfo* срабатывает в случае если не определено ни одного слоя для запроса.
- *getfeatureinfo* наступает в момент получения ответа от сервера.

Пример использования WMSGetFeatureInfo

Создадим слой WMS и добавим контрол WMSGetFeatureInfo для запроса информации об объектах этого слоя:

```
//Создание карты  
map = new OpenLayers.Map('map');  
  
// Создание WMS-слоя  
var report = new OpenLayers.Layer.WMS('report',  
  "http://gis-lab.info/cgi-bin/mapserv?map=/usr/local/www/website/map/wms.map",  
  {layers: "report_1", transparent: true},  
  {isBaseLayer: false, opacity: 1, singleTile: true, visibility: true}  
);  
map.addLayers([report]);  
  
// Создание контрола WMSGetFeatureInfo  
var click_ctrl = new OpenLayers.Control.WMSGetFeatureInfo({  
  url: "http://gis-lab.info/cgi-bin/mapserv?map=/usr/local/www/website/map/wms.map",  
  layers: [report],  
  queryVisible: true,  
  infoFormat: 'application/vnd.ogc.gml',  
  maxFeatures: 50,  
  autoActivate: true,
```

```
// Vendor-specific параметр, используемый в MapServer (радиус поиска объектов в пикселях)
vendorParams: {
    radius: 10
},
eventListeners: {
    'getfeatureinfo': function(e) {
        // Текстовое представление ответа сервера
        console.log(e.text);

        // Координаты клика
        console.log(e.xy);

        // Список объектов OpenLayers.Feature.Vector, содержащих в атрибуте
        информацию, полученную с сервера
        console.log(e.features);
    }
});

// Добавляем контрол на карту
map.addControl(click_ctrl);
```

В данном примере при клике на объекте слоя *report* будет выполнен запрос *GetFeatureInfo* к серверу WMS и при получении ответа (событие *getfeatureinfo*) в консоль будут выведено несколько свойств объекта события. Если с ответом нужно производить какие-то действия (выводить информацию во всплывающем окне, представить в виде таблицы и т.п.), то *console.log()* необходимо заменить на необходимый функционал.

Если в качестве сервера WMS используется MapServer, то настоятельно рекомендуется ознакомиться со статьёй [Особенности работы с запросами GetFeatureInfo для организации сервиса WMS на базе MapServer](#).

Cross-domain security

Политика безопасности JavaScript запрещает обращаться на другие серверы. Поэтому если домен (включая номер порта) на котором расположен сервер WMS отличается от домена на котором расположено само web-приложение, то для функционирования контрола *WMSGetFeatureInfo* потребуется создать проху-скрипт (используя, например, *php* или *python*), который будет обрабатываться тем же сервером, на котором расположено web-приложение. Можно воспользоваться штатным скриптом из состава OpenLayers или [скачать](#) его отдельно. Этот скрипт написан на *python*, поэтому для его функционирования на сервере должен быть установлен соответствующий интерпретатор. Настройка прокси заключается в указании перечня доменов к которым разрешено обращаться. Для наших целей достаточно добавить в массив *allowedHosts* адрес домена по которому расположен сервер WMS. Дополнительно о настройке *ProxyHost* можно прочитать [здесь](#).

[Обсудить в форуме](#) Комментариев — 3

Последнее обновление: 2014-05-15 00:08

Дата создания: 21.03.2012

Автор(ы): [Денис Рыков](#)