Начало работы с Polymaps

Обсудить в форуме Комментариев — 11

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

Знакомство с библиотекой Polymaps

Библиотека OpenLayers благодаря своим широким возможностям стала стандартом де-факто при разработке свободных картографических web-приложений. Однако, недавний скачок роста использования Интернета на мобильных устройствах привёл к появлению новых библиотек, демонстрирующих совершенно иной подход. Хотя OpenLayers, начиная с версии 2.11, добился значительного прогресса в вопросе поддержки мобильных устройств, новые библиотеки, такие как Polymaps, Tile5 и Leaflet изначально разрабатывались с ориентацией на современные технологии, в том числе и на работу в мобильных приложениях.

В данной статье мы рассмотрим одну из этих новых библиотек, а именно продукт совместной работы Stamen Design и SimpleGeo (в октябре 2011 компания SimpleGeo была приобретена Urban Airship) - Polymaps.

Содержание

Особенности библиотеки

Polymaps - это небольшая (32 Кб, что в 30 раз меньше OpenLayers в оригинальной сборке) JavaScript библиотека с открытым исходным кодом, предназначенная для создания интерактивных web-карт. Главной особенностью библиотеки является возможность работы с векторными данными больших объёмов. Это достигается за счёт применения тайлового подхода к векторным SVG-изображениям и автоматического изменения детализации геометрий при переходе между масштабными уровнями. В качестве единственного поддерживаемого формата векторных данных в Polymaps используется GeoJSON.

Как уже было отмечено выше, Polymaps ориентирован на работу с масштабируемой векторной графикой (Scalable Vector Graphics, SVG). Многие из существующих библиотек web-картографии обременены поддержкой совместимости с устаревшими браузерами в которых отсутствует поддержка SVG (здесь важно отметить, что поскольку поддержка SVG появилась только в IE9, то в предыдущих версиях указанного браузера Polymaps работать не будет). Выбор же в качестве рендера SVG позволяет Polymaps выполнять широкий спектр графических операций, а также даёт возможность настраивать внешний вид объектов с помощью CSS.

Для обеспечения высокой производительности при работе с большими объёмами данных в Polymaps используется ряд специальных стратегий. Во-первых, все тайлы, и растровые и векторные, кэшируются, поэтому если потребуется какой-то из уже использованных ранее тайлов, он будет получен из кэша, а не через запрос к серверу. Во-вторых, если нужный тайл недоступен в кэше, но доступны тайлы соседних масштабных уровней на этот участок, то они могут быть временно использованы до тех пор, пока нужный тайл не придёт с сервера. Такой подход позволяет обеспечить плавность сдвига и изменения масштаба.

Подключение библиотеки и основные подходы к использованию

Для получения библиотеки необходимо перейти на главную страницу и скачать последнюю версию, на

настоящий момент это 2.5.0. Для подключения Polymaps, необходимо в разделе head html-страницы указать путь до файла библиотеки polymaps.js или polymaps.min.js (минифицированный вариант).

Каждое приложение Polymaps начинается с создания объекта map. Но прежде чем его создавать, необходимо импортировать пространство имён Polymaps в какую-нибудь локальную переменную, например:

```
var po = org.polymaps;
```

После чего можно переходить непосредственно к созданию объекта map:

```
var map = po.map();
```

Отметим, что библиотека не использует традиционный JavaScript конструктор new, вместо этого Polymaps предоставляет фабричные методы, автоматически создающие необходимые объекты.

Все методы объекта map возвращают сам объект, поэтому работа с Polymaps предполагает активное использование цепочек методов. Например:

```
var map =
po.map().container(document.body.appendChild(po.svg("svg"))).add(po.image().url(…));
```

Это одна команда, которая создаёт объект map, выделяет под него контейнер и добавляет растровый слой. В случае, если карта содержит несколько слоёв, они будут отрисовываны снизу вверх (алгоритм художника):

```
var map = po.map()
    .container(document.body.appendChild(po.svg("svg")))
    .add(po.image().url("grid.png")) //Нижний слой
    .add(po.image().url(…)); //Верхний слой
```

При работе со слоями также используются цепочки методов, позволяющие конфигурировать слои одновременно с добавлением их на карту. URL слоя может быть либо статическим (например, в случае использования растрового файла), либо динамическим. В последнем случае составляется шаблон URL с использованием подстановочных переменных:

- {X} - номер столбца тайловой разбивки;
- {Y} - номер строки тайловой разбивки;
- {B} - охват;
- {Z} - масштабный уровень;
- {S} - хост.

Синтаксис добавления векторного GeoJSON слоя похож на синтаксис добавления растровой подложки:

```
var map = po.map()
    .container(document.body.appendChild(po.svg("svg")))
    .add(po.image().url(…)) //Растровый слой
    .add(po.geoJson().url(…)); //Векторный слой
```

По умолчанию создаваемая карта неинтерактивна. Для придания ей элементов интерактивности необходимо добавить контролы взаимодействия с пользователем:

```
var map = po.map()
    .container(document.body.appendChild(po.svg("svg")))
    .add(po.image())
    .add(po.interact());
```

Контрол interact представляет собой обобщенный вызов пяти различных контролов:

```
var map = po.map()
    .container(document.body.appendChild(po.svg("svg")))
    .add(po.image())
    .add(po.arrow())
    .add(po.drag())
```

```
    .add(po.dblclick())
    .add(po.touch())
```

Контрол arrow отвечает за сдвиг карты с помощью клавиш стрелок и изменение масштаба с помощью клавиш "+" и "-", контрол drag - за сдвиг карты с помощью мыши, контрол dblclick позволяет приблизить карту по двойному клику, контрол wheel отвечает за изменение масштаба карты при вращении колеса мыши и, наконец, контрол touch отвечает за работу с мобильными устройствами.

Практическое применение

На официальном сайте Polymaps представлен ряд примеров использования библиотеки. Кроме того, в архиве с библиотекой представлено еще около 30 примеров, некоторые из которых дублируют примеры с сайта. Рассмотрим пару простых примеров, наглядно иллюстрирующих использование Polymaps для создания интерактивных карт.

Подключение подложки CloudMade

Подложка CloudMade, представляет собой различные стили оформления данных проекта OpenStreetMap.

Код представленной карты:

```
<html>
  <head>
    <style type="text/css">
      #map {height: 300px; width: 700px;}
    </style>
    <script type="text/javascript" src="polymaps/polymaps.js"></script>
  </head>
  <body>
    <div id="map"></div>
    <script type="text/javascript">
      var po = org.polymaps;
      var map = po.map()
        .container(document.getElementById("map").appendChild(po.svg("svg")))
        .center({lat: 55.0276, lon:82.9241})
        .zoom(13)
        .add(po.interact())
        .add(po.hash())
        .add(po.image()
        .url(po.url("http://{S}tile.cloudmade.com"
        + "/1a1b06b230af4efdbb989ea99e9841af" // http://cloudmade.com/register
        + "/999/256/{Z}/{X}/{Y}.png")
        .hosts(["a.", "b.", "c.", ""])));
    </script>
  </body>
</html>
```

Рассмотрим представленный код более подробно. Импорт пространства имён Polymaps:

```
var po = org.polymaps;
```

Создание объекта map:

```
var map = po.map();
```

Создание контейнера в который будет добавлена карта:

```
.container(document.getElementById("map").appendChild(po.svg("svg")))
```

Установка центра карты:

```
.center({lat: 55.0276, lon: 82.9241});
```

Задание масштабного уровня:

```
.zoom(13)
```

Добавление контролов взаимодействия с пользователем:

```
.add(po.interact())
```

Добавление контрола hash:

```
add(po.hash())
```

Данный контрол позволяет автоматически изменять URL карты при сдвиге и изменении масштаба, тем самым даёт возможность легкого создания постоянных ссылок на участки карты (Permalink).

Добавление подложки CloudMade:

```
add(po.image()
  .url(po.url("http://{S}tile.cloudmade.com"
  + "/1a1b06b230af4efdbb989ea99e9841af" // http://cloudmade.com/register
  + "/999/256/{Z}/{X}/{Y}.png")
  .hosts(["a.", "b.", "c.", ""])));
```

В последней команде интересен метод hosts объекта po.url. Данный метод принимает на вход массив строк, используемых для валидации подстановочной переменной {S} в строке URL. То есть в нашем случае, допустимыми адресами тайлов могут быть:

- http://a.tile.cloudmade.com/1a1b06b230af4efdbb989ea99e9841af/999/256/13/6002/2657.png
- http://b.tile.cloudmade.com/1a1b06b230af4efdbb989ea99e9841af/999/256/13/6003/2657.png
- http://c.tile.cloudmade.com/1a1b06b230af4efdbb989ea99e9841af/999/256/13/6004/2657.png
- http://tile.cloudmade.com/1a1b06b230af4efdbb989ea99e9841af/999/256/13/6001/2657.png

Добавление векторного слоя

Рассмотрим простой пример визуализации статического GeoJSON файла.

Код представленной карты:

```
<html>
  <head>
    <style type="text/css">
      #map {
          height: 400px;
          width: 400px;
        }
      #russia path {
          fill: lightsteelblue;
          fill-opacity: .8;
          stroke: FireBrick;
        }
    </style>
    <script type="text/javascript" src="polymaps/polymaps.js"></script>
  </head>
  <body>
    <div id="map-geojson"></div>
    <script type="text/javascript">
      var po = org.polymaps;
      var map = po.map()
        .container(document.getElementById("map-geojson").appendChild(po.svg("svg")))
        .center({lat: 68, lon: 91.4})
        .zoom(2)
        .add(po.interact());

      map.add(po.image()
```

```
        .url(po.url("http://{S}tile.cloudmade.com"
        + "/1a1b06b230af4efdbb989ea99e9841af" // http://cloudmade.com/register
        + "/7/256/{Z}/{X}/{Y}.png")
        .hosts(["a.", "b.", "c.", ""])));

      map.add(po.geoJson()
        .url("../other/federal-districts.geojson")
        .id("russia"));
    </script>
  </body>
</html>
```

По сравнению с предыдущим примером добавилась одна команда:

```
map.add(po.geoJson()
  .url("federal-districts.geojson")
  .id("russia"));
```

Этой командой мы добавляем на карту слой federal-districts.geojson и присваиваем слою идентификатор "russia". Данный идентификатор используется в качестве CSS-селектора при настройке символики слоя:

```
#russia path {
  fill: lightsteelblue;
  fill-opacity: .8;
  stroke: FireBrick;
}
```

В рассмотренном примере не использовалась техника векторных тайлов, файл загружался целиком. Вопросу разбивки и передачи векторных данных в виде тайлов и анализу увеличения производительности в этом случае мы планируем посвятить одну из следующих статей.

Заключение

Подводя итог, можно сделать вывод о том, что Polymaps - это графический фреймворк, позволяющий работать с большими объемами векторных данных за счет использования тайловой техники и настраивать символику геометрических объектов с помощью каскадных таблиц стилей (CSS). Однако, отсутствие поддержки широкого спектра источников пространственных данных (в том числе WFS, WMS), инструментов редактирования, ограниченный список поддерживаемых браузеров (Polymaps гарантированно работает только в Chrome 6+, Safari 4+ и Firefox 3.6 и гарантированно не работает в версиях MS Internet Explorer младше 9) накладывают некоторые ограничения на создание функциональных картографических приложений.

Ссылки по теме

- Официальный сайт Polymaps
- Примеры использования Polymaps
- Презентация, посвященная Polymaps

Обсудить в форуме Комментариев — 11

Последнее обновление: 2014-05-15 01:14

Дата создания: 15.06.2011
Автор(ы): Денис Рыков