

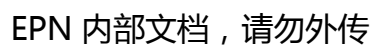
Esri 的四个 JS 开源项目: Esri-Leaflet、 Geoservices.js、Terraformer、 Pushlet

易智瑞（中国）信息技术有限公司

2013 年 8 月

版权声明

本文档版权为 ESRI 中国（北京）有限公司所有。未经本公司书面许可，任何单位和个人不得以任何形式摘抄、复制本文档的部分或全部，并以任何形式传播。

[illegible]

目 录

1.	ESRI-LEAFLET	1
2.	GEOSERVICES.JS	4
3.	TERRAFORMER.....	5
4.	PUSHLET	5

Esri 的四个开源 JS 项目：Esri-Leaflet、Geoservices.js、Terraformer、Pushlet

翻译自 ArcGIS 资源中心博客：[New Esri Open Source Javascript Projects: Leaflet, Geoservices.js, Terraformer, Pushlet](#)

[Aaron Parecki](#) 写于 2013 年 7 月 31 日

我们高兴的宣布，经过我们过去几个月的努力，我们终于可以公布以下四个开源的 Javascript 项目。

- [Esri-Leaflet](#) 项目：是一个帮助开发者使用 Leaflet 构建轻量级应用的 JS 地图组件库
- [Geoservices.js](#) 项目：是一个在浏览器或者 Node.js 操作 ArcGIS Online 服务的 JS 客户端
- [Terraformer](#) 项目：是一个处理不同几何类型并构建空间数据库的几何工具包
- [Pushlet](#) 项目：是一个通过 Apple 和 Google 推送通知服务发送通知的简单 API

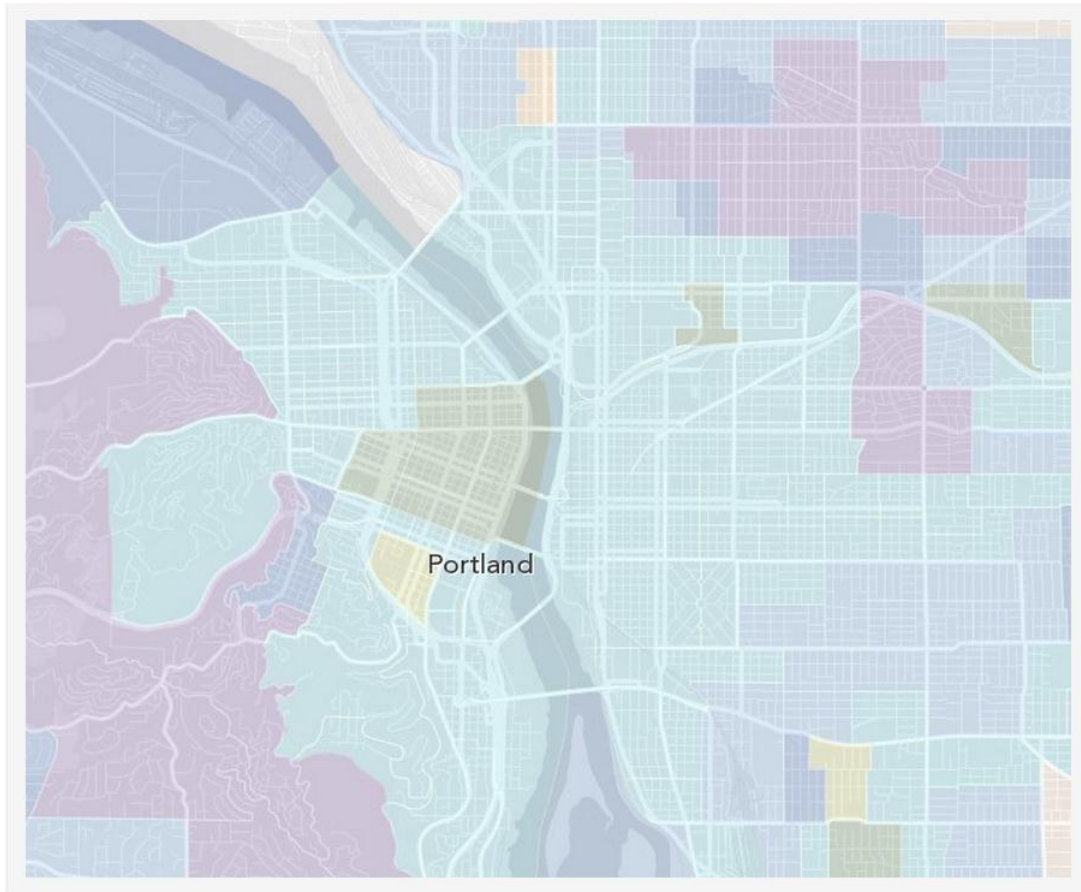
下面分别介绍这四个开源项目的情况：

1. Esri-Leaflet

如上面介绍，Esri-Leaflet 是一个能够构建轻量级应用的 JS 库，它可以和 Terraformer 结合来实现不同空间数据格式的转换，它也可以和 Geoservices 一起构建 ArcGIS REST 服务请求，比如地址查询和反向地理编码。当你熟悉了 Leaflet，你会觉得使用它是如此的轻松自如。下面介绍一下 Leaflet 的优势，Leaflet 可以直接操作 ArcGIS 地图切片，并且会根据地图显示范围在地图上显示当前的数据版权信息，Esri-Leaflet 还支持了显示视网膜底图。

人口统计数据服务

下面是在 Esri Leaflet 地图中设置灰色底图的 ArcGIS 切片地图服务的示例

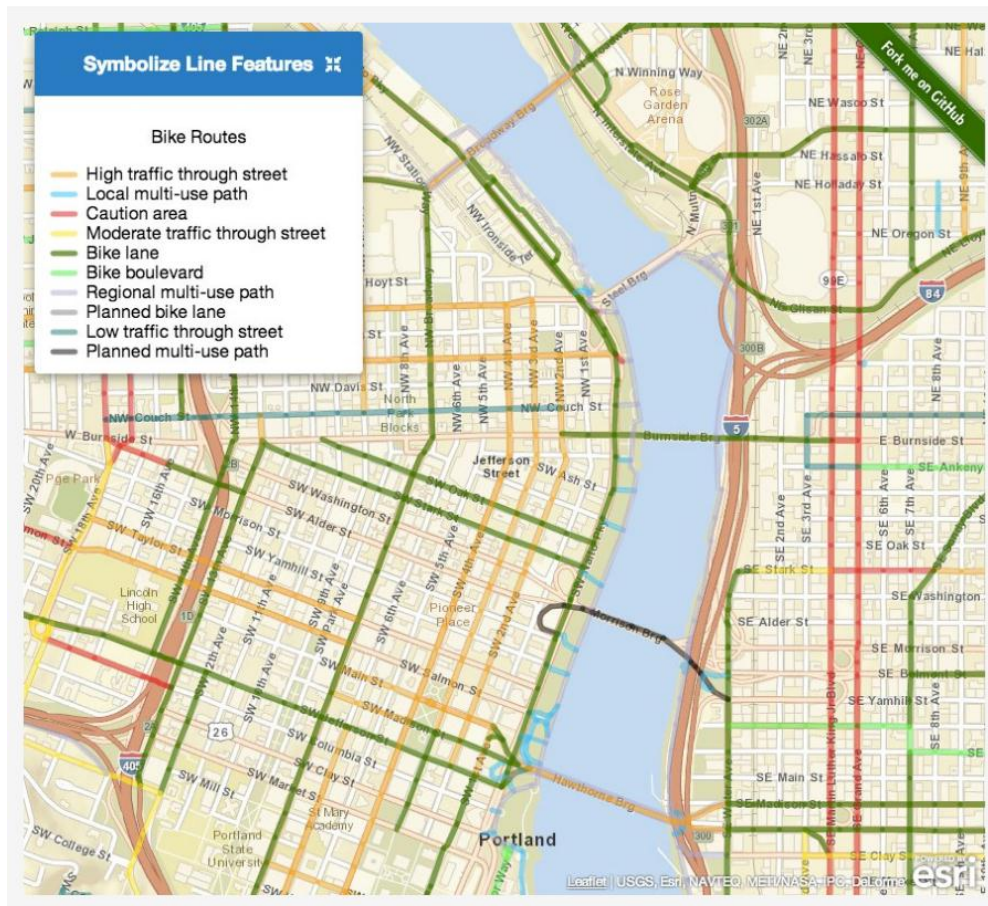


下面是简单的代码片段：

```
1 var baseUrl = "http://services.arcgisonline.com/ArcGIS/rest/services/Demographics/  
2     USA_Tapestry/MapServer";  
3  
4 tiledLayer = L.esri.tiledMapLayer(url, {  
5     opacity: 0.25,  
6     zIndex: 2,  
7     detectRetina: true  
8     }).addTo(map);
```

行车路线

Leaflet 不止可以显示点或多边形，同样可以显示线，下面是在地图上显示 Portland 的行车路线的示例：



下面是代码片段：

```
1 L.esri.featureLayer("http://services.arcgis.com/uCXeTVveQzP4IIcx/arcgis/rest/services/
2 bike_rte/FeatureServer/0", {
3   style: function (feature) {
4     return getStyle(feature);
5   },
6   onEachFeature: function(geojson, layer){
7     layer.bindPopup("<h3>"+geojson.properties.BIKEMODE+"</h3><p>"
8     +(geojson.properties.PREFIX ? geojson.properties.PREFIX : "")+" "
9     +(geojson.properties.STREETNAME ? geojson.properties.STREETNAME : "")+" "
10    +(geojson.properties.FTTYPE ? geojson.properties.FTTYPE : "")+" " ); }).addTo(map);
```


更多示例：

在 Esri-Leaflet 的 Github 页面上可以看到[更多示例](#)。

2. Geoservices.js

[Geoservices.js](#) 是帮助用户获取和查询 ArcGIS Online 和其他服务的数据的 Javascript 客户端程序。[这里](#)是关于 Geoservices 的说明文档。它也可以在浏览器中以下面的方式运行：

```
1 <script src="browser/geoservices.js"></script>
2 <script> var client=new Geoservices(); </script>
```

或者在 Node.js 上运行：

```
1 var Geoservices = require('geoservices');
2 var client=new Geoservices();
```

目前 Geoservices 可以用于处理免费的服务，可以通过认证后处理收费的服务，现在它已经支持[地理编码](#)和[要素服务](#)，不久以后就会支持[导航和路线规划服务](#)，最终会支持几何服务。下面是简单的使用地理编码服务的示例：

```
1 client.geocode({text: "920 SW 3rd Ave, Portland, OR 97204"}, function (err, result) {
2     if (err) {
3         console.error("ERROR: " + err);
4     } else {
5         console.log("Found it at " + result.locations[0].feature.geometry.y + ",
6             " result.locations[0].feature.geometry.x);
7     }
8 });
```


3. Terraformer

[Terraformer](#)是处理不同的几何格式并创建空间数据库的几何工具包。Terraformer 是一个既可以在浏览器端运行也可以在 Node.js 运行的轻量级的 Javascript 库。Terraformer 包括以下几个功能：

- 在 ArcGIS 地图中浏览 GeoJSON 对象
- 通过 R 树索引可视化 Well-known text
- 通过 Terraformer 的 GeoStore 可视化和搜索美国州数据
- 通过 Terraformer 和 Node.js 建立简单的时区 API

例如，**Terraformer** 可以载入整个美国州的数据文件，建立 R 树索引，那么就可以给定一个经纬度快速的查询所在的州的信息了。

4. Pushlet

[Pushlet](#)是通过 Apple 或者 Google 的推送通知服务来发送通知的简单 API。如果你曾经使用过 Apple 的 APNS API 的话，那你应该知道正确的使用它是多么困难。为了连接到 APNS，你必须要配置好 SSL 证书和认证，然后通过固定的端口连接而不能每次推送消息时新建连接。Pushlet 使用 Redis 作为临时数据存储来保存证书和连接 APNS 服务的端口信息，这样开发者就不用在这方面花费过多的心思了。使用 Pushlet 发送一个通知就像向 Pushlet 终端发送一个 POST 请求一样简单，下面是一个示例：

```
1 {  
2   "appId": "com.example.iphone",  
3   "deviceId": "809f1d3237cd219c1c672bb141f6e18513fd86a073479ef295fd0e1687270853",  
4   "mode": "production",
```

```
5  "notification":{  
6    alert:"The quick brown fox jumps over the lazy dog"  
7  }  
8 }
```

如果 Pushlet 还没有关于这个 appid 的证书，那么会得到下面的响应信息：

```
1 {  
2   "response":"error",  
3   "error":"missing certificate"  
4 }
```

如果发送了包含证书的消息，如下所示：

```
1 {  
2   "appId":"com.example.iphone",  
3   "deviceId": "809f1d3237cd219c1c672bb141f6e18513fd86a073479ef295fd0e1687270853",  
4   "mode": "production",  
5   "notification":{  
6     "alert": "The quick brown fox jumps over the lazy dog"  
7   },  
8   "cert":"-----BEGIN CERTIFICATE-----\nMIIFV...",  
9   "key":"-----BEGIN PRIVATE KEY-----\nMIIEvQI..."  
10 }
```

那么当 Pushlet 成功的连接到了 APNS 并发送了上面的信息后，它需要等待一段时间，验证发送的端口是否处于开启状态，如果端口仍处于开启状态，那么返回 ok。

```
1 {  
2   "response": "ok"  
3 }
```

当然，如果你已经设置好证书并且已经能够成功的向你的设备发送推送信息了，那么你可以不再设置 **timeout** 选项来提高 HTTP 的响应速度

```
1 {  
2   "appId": "com.example.iphone",  
3   "deviceId": "809f1d3237cd219c1c672bb141f6e18513fd86a073479ef295fd0e1687270853",  
4   "mode": "production",  
5   "notification": {  
6     alert: "The quick brown fox jumps over the lazy dog"  
7   },  
8   "timeout": 0  
9 }  
10  
11 {  
12   "response": "sent"  
13 }
```

由于 **Pushlet** 并不会保存设备的 **token**，也不会将推送信息的证书永久保存，所以用户可以通过在集群服务器上建立多个 **Pushlet** 实例的方式来提高吞吐量。