

Openlayers超级简单实现流向动画例子！

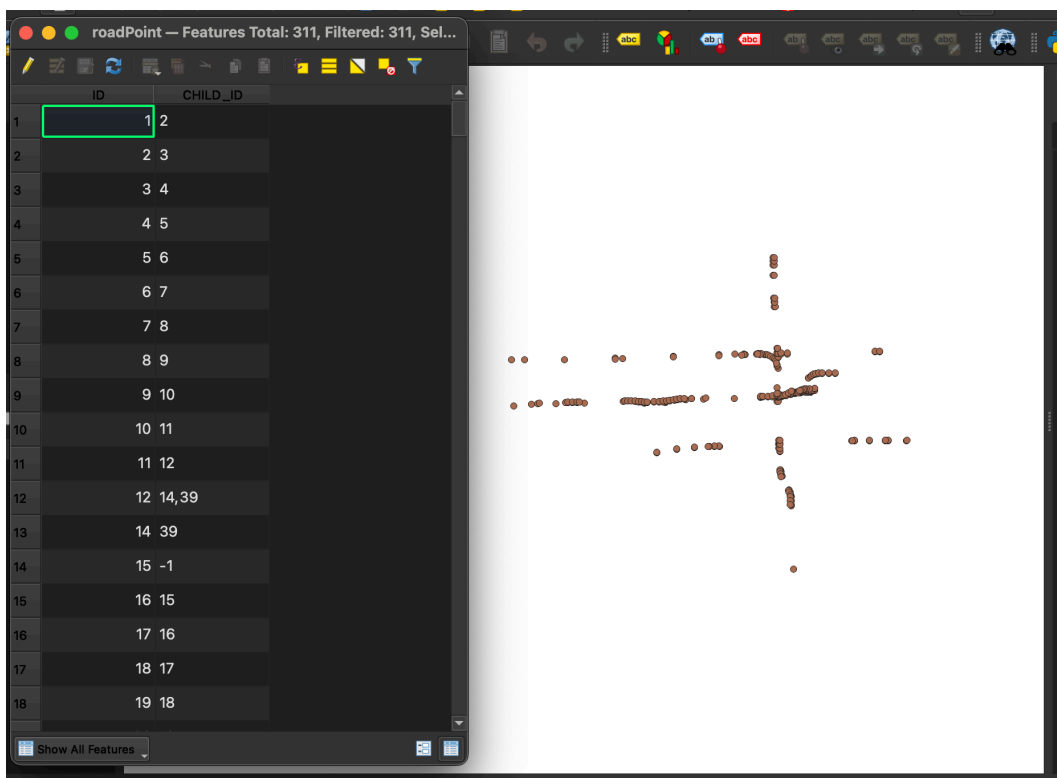


1、数据格式

地图上：html/data/road.js 绿色线数据；html/data/roadPoint.js 紫色点数据；

后台：WebSockProject/data/roadPoint.shp。

roadPoint.shp和roadPoint.js是完全一致的就是数据格式不通一个是esri shp文件一个是geojson，之所以这样是浏览器无法直接加载shp数据。road.js只做显示作用。

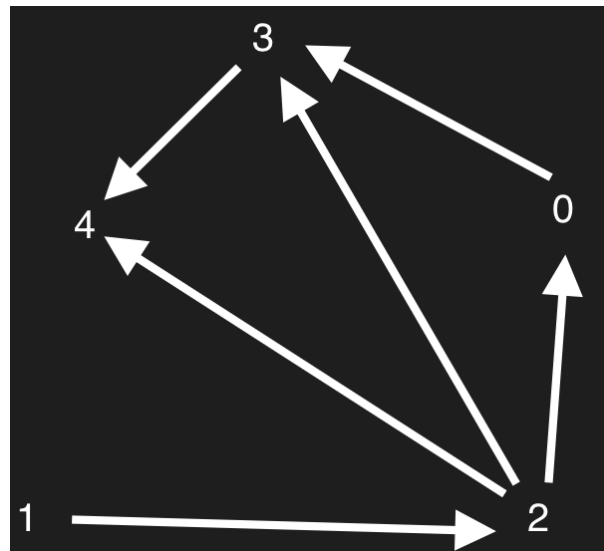


roadPoint.shp属性表存储当前点的ID和他子节点的CHILD_ID，如果他的子节点CHILD_ID有多个说明这是个分叉点，如果为-1则为终点无子节点。

2、后台实现思路 (WebSockProject)

步骤一：依据属性表生成一个二维数组来描述点之间连通关系

	0	1	2	3	4
0	0	0	0	1	0
1	0	0	1	0	0
2	1	0	0	1	1
3	0	0	0	0	1
4	0	0	0	0	0



通过0或者1来描述连通的关系，0不通，1通。对角线都是0，节点自身不通，红框所示节点2到0是通。（以竖着的0，1，2，3，4为基准）这相当于有向图数据结构。

选点遍历递归

以选择节点1为例遍历结果为：1、2、0、3、4； 2、3、4； 2、4； 三条线。其中1、2、0、3、4是一条完整的线，从选择的点到4叶子节点。

对递归的结果还要进行去重的操作，有些情况是分支线路是一条线路的一段，那这种情况就不需要保留了。

gdalshap.cpp是读取原始数据；

pipeConnectivity.cpp生成连通关系线路；

3、前端实现，获取后端传来的点编号，按照点编号渲染，两点之间通过经度纬度差，算要插入几个点来计算动画节点。如 $(lon1, lat1)$, $(lon2, lat2)$, 之间的第N个点为 $lonN = lon1 + N * daltLon$; $latN = lat1 + N * daltLat$, 直接当直角坐标系处理。插值越多越平滑。具体代码实现html/map.js。

