

周报

一、百度地图POI分类标准与高德地图分类标准

百度地图POI行业分类 (tag) : <http://lbsyun.baidu.com/index.php?title=lbscloud/poitags>

高德地图POI行业分类: <https://lbs.amap.com/api/webservice/download>

百度地图POI包含21个一级行业分类, 83个二级行业分类, 其中, 居民小区所属一级行业分类为房地产, 二级行业分类为住宅区。高德地图包含23个大类, 265个中类, 869个小类, 其中, 居民小区所属大类为商务住宅, 中类为住宅区, 小类为住宅小区或别墅。

二、百度地图搜索方式

LocalSearch: 本地搜索, 提供某一特定地区的位置搜索服务, 比如在北京市搜索“公园”。

城市检索

BMap.LocalSearch提供本地搜索服务, 在使用本地搜索时需要为其设置一个检索区域, 检索区域可以是BMap.Map对象、BMap.Point对象或者是省市名称 (比如: “北京市”) 的字符串。

BMap.LocalSearch构造函数的第二个参数是可选的, 您可以在其中指定结果的呈现。

BMap.RenderOptions类提供了若干控制呈现的属性, 其中map指定了结果所展现的地图实例, panel指定了结果列表的容器元素。

```
var map = new BMap.Map("container");
map.centerAndZoom(new BMap.Point(116.404, 39.915), 11);
var local = new BMap.LocalSearch(map, {
    renderOptions:{map: map}
});
local.search("天安门");
```

另外, BMap.LocalSearch还提包含searchNearby和searchInBounds方法, 为您提供周边搜索和范围搜索服务。

周边搜索

通过周边搜索服务, 您可以在某个地点附近进行搜索, 也可以在某一个特定结果点周围进行搜索。

```
var map = new BMap.Map("container");
map.centerAndZoom(new BMap.Point(116.404, 39.915), 11);
var local = new BMap.LocalSearch(map,
    { renderOptions:{map: map, autoViewport: true}});
local.searchNearby("小吃", "前门");
```

矩形搜索

矩形范围搜索将根据您提供的视野范围提供搜索结果。注意: 当搜索范围过大时可能会出现无结果的情况。

```
var map = new BMap.Map("container");
map.centerAndZoom(new BMap.Point(116.404, 39.915), 14);
var local = new BMap.LocalSearch(map,
    { renderOptions:{map: map}});
local.searchInBounds("银行", map.getBounds());
```

数据接口

BMap.LocalSearch和BMap.LocalSearchOptions类提供了若干设置回调函数的接口，通过它们可得到搜索结果的数据信息。例如，通过onSearchComplete回调函数参数可以获得BMap.LocalResult对象实例，它包含了每一次搜索结果的数据信息。当回调函数被执行时，您可以使用BMap.LocalSearch.getStatus()方法来确认搜索是否成功或者得到错误的详细信息。

LocalSearchOptions

此类表示LocalSearch构造函数的可选参数。

属性	类型	描述
renderOptions	LocalRenderOptions	结果呈现设置
onMarkersSet	Function	标注添加完成后的回调函数。参数： pois: Array，通过marker属性可得到其对应的标注
onInfoHtmlSet	Function	标注气泡内容创建后的回调函数。参数： poi: LocalResultPoi，通过其marker属性可得到当前的标注。html: HTMLElement，气泡内的Dom元素
onResultsHtmlSet	Function	结果列表添加完成后的回调函数。参数： container: HTMLElement，结果列表所用的HTML元素
pageCapacity	Number	设置每页容量，取值范围：1 - 100，对于多关键字检索，容量表示每个关键字的数量，如果有2个关键字，则实际检索结果数量范围为：2 - 200
onSearchComplete	Function	检索完成后的回调函数。参数： results: LocalResult或Array 如果是多关键字检索，回调函数参数返回一个LocalResult的数组，数组中的结果顺序和检索中多关键字数组中顺序一致

三、矩形范围内所有小区边界

1、初始化百度地图

```
var map = new BMap.Map("container");
// 创建地图实例
var point = new BMap.Point(116.404, 39.915);
// 创建点坐标
map.centerAndZoom(point, 15);
// 初始化地图，设置中心点坐标和地图级别
map.enableScrollWheelZoom();
```

2、确定矩形区域

```

var pStart = new BMap.Point(116.274625,39.958527);
var pEnd = new BMap.Point(116.307474,39.988409);
var bs = new BMap.Bounds(pStart,pEnd); //自己规定范围
var polygon = new BMap.Polygon([
    new BMap.Point(pStart.lng,pStart.lat),
    new BMap.Point(pEnd.lng,pStart.lat),
    new BMap.Point(pEnd.lng,pEnd.lat),
    new BMap.Point(pStart.lng,pEnd.lat)
], {strokeColor:"blue", strokeWeight:6, strokeOpacity:0.2});
map.addOverlay(polygon);

```

3、获取矩形范围内住宅小区UID

```

var local = new BMap.LocalSearch(
    map,
    {renderOptions: {map: map, autoViewport: true},
      pageCapacity: 50,
      onSearchComplete: function (results) {
        for (var i = 0; i < results.Ir.length; i++){
          ResultArray.push(results.Ir[i].uid)
        }
        var totalPages = results.getNumPages();
        var currPage = results.getPageIndex();
        if (currPage < totalPages-1){
          local.gotoPage(currPage + 1);
        }else {
          for (var j = 0; j < ResultArray.length; j++){
            queryuid(ResultArray[j]);
          };
        }
      }
    }
  );

```

其中，onSearchComplete用于获取全部信息。否则只能获取一页信息，默认10条。pageCapacity用于设置每一页结果数量，最好设置为50。百度地图API对一次性返回的搜索结果进行了限制，一次性最多只能返回50条搜索结果，默认情况下一性返回10条搜索结果。

4、使用queryuid (uid) 画出边界

```

function queryuid(uid){
    $.ajax({
        async: false,
        url: "http://map.baidu.com/?
pcevaname=pc4.1&qt=ext&ext_ver=new&l=12&uid="+uid,
        dataType: 'jsonp',
        jsonp: 'callback',
        success: function (result) {
            var content = result.content;
            if (null != content.geo && content.geo != undefined){
                var geo = content.geo;
                var points = coordinateToPoints(geo);
                if (points && points.indexOf(";") >= 0){
                    points = points.split(";");
                };
            };
        }
    });
}

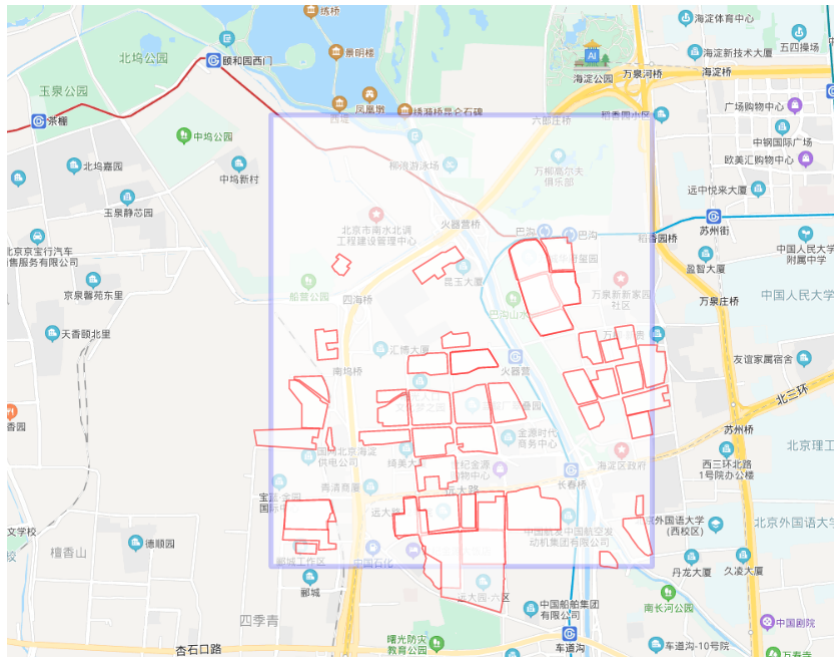
```

```

var arr = [];
for(var i = 0; i < points.length-1; i++){
    var tem = points[i].split(",");
    arr.push(new BMap.Point(parseFloat(tem[0]),
parseFloat(tem[1])));
};
var polygon = new BMap.Polygon(arr, {strokeColor:"red",
strokeWeight:2, strokeOpacity:0.5});
map.addOverlay(polygon);
}
}
});
};

```

5、效果



6、数据分析

```

▼ 0:
  address: "北京市海淀区远大路19号附近"
  city: "北京市"
  detailUrl: "http://api.map.baidu.com/place/detail?uid=d8b36c5414b140526f47609f&output=html&source=jsapi"
  isAccurate: false
  ▶ marker: V {ba: "TANGRAM_55", Lb: "overlay", map: null, Pa: true, Ab: null, ...}
    phoneNumber: undefined
  ▶ point: N {lng: 116.288577, lat: 39.963826}
  ▶ pointN: H {lng: 116.288577, lat: 39.963826, of: "inner"}
    postcode: undefined
    province: "北京市"
    title: "远大路19号院"
    type: 0
    uid: "d8b36c5414b140526f47609f"
    url: "http://map.baidu.com/?s=inf%26uid%3Dd8b36c5414b140526f47609f%26c%3D131&i=0&sr=1"
  ▶ __proto__: Object

```

通过百度地图的localsearch本地搜索，以小区为关键词共获得134个POI点，其中包含边界坐标的有51个。获取的POI信息包括地址、坐标、城市、UID、名称等，部分POI点信息包含tag，即行业分类。高德地图的POI信息则包含分类。

```

▼ 9:
  address: "北京市海淀区四季青桥常青园路"
  city: "北京市"
  detailUrl: "http://api.map.baidu.com/place/detail?uid=08f1f1f5a2f139ca89db73ac&output=html&source=jsapi"
  isAccurate: false
  ▶ marker: V {ba: "TANGRAM__2p", Lb: "overlay", map: null, Pa: true, Ab: null, ...}
  phoneNumber: undefined
  ▶ point: N {lng: 116.277881, lat: 39.962019}
  ▶ pointN: H {lng: 116.277881, lat: 39.962019, of: "inner"}
  postcode: undefined
  province: "北京市"
  ▶ tags: (2) ["地产小区", "小区/楼盘"]
  title: "常青园一区"
  type: 0
  uid: "08f1f1f5a2f139ca89db73ac"
  url: "http://map.baidu.com/?s=inf%26uid%3D08f1f1f5a2f139ca89db73ac%26c%3D131&i=0&sr=1"
  ▶ __proto__: Object

```

提取poi中的信息，如下：

```

    ▼ 0:
      UID: "348dc170753e8a8b601809b6"
      address: "万柳光大西园6"
      city: "北京市"
      ▶ point: N {lng: 116.301871, lat: 39.970491}
      province: "北京市"
      ▶ tag: (2) ["地产小区", "小区/楼盘"]
      title: "光大水墨风景小区-东门"
      ▶ __proto__: Object

```

获取具有边界坐标的uid，如下：

```

▼ 0:
  UID: "fa7da3e88129f06243d63eaf"
  ▼ boundary: Array(27)
    ► 0: N {lng: 116.289317, lat: 39.957167}
    ► 1: N {lng: 116.289216, lat: 39.957122}
    ► 2: N {lng: 116.289095, lat: 39.957063}
    ► 3: N {lng: 116.288776, lat: 39.957184}
    ► 4: N {lng: 116.28861, lat: 39.957311}
    ► 5: N {lng: 116.28859, lat: 39.957371}
    ► 6: N {lng: 116.288574, lat: 39.957401}
    ► 7: N {lng: 116.288544, lat: 39.957467}
    ► 8: N {lng: 116.28834, lat: 39.95768}
    ► 9: N {lng: 116.287975, lat: 39.957898}
    ► 10: N {lng: 116.287737, lat: 39.958119}
    ► 11: N {lng: 116.287828, lat: 39.958437}
    ► 12: N {lng: 116.287751, lat: 39.958464}
    ► 13: N {lng: 116.287738, lat: 39.958479}
    ► 14: N {lng: 116.287797, lat: 39.958671}
    ► 15: N {lng: 116.287954, lat: 39.959095}
    ► 16: N {lng: 116.287855, lat: 39.959392}
    ► 17: N {lng: 116.287644, lat: 39.959741}
    ► 18: N {lng: 116.287725, lat: 39.9599}
    ► 19: N {lng: 116.287586, lat: 39.96017}
    ► 20: N {lng: 116.287527, lat: 39.960371}
    ► 21: N {lng: 116.287554, lat: 39.960578}
    ► 22: N {lng: 116.29068, lat: 39.960899}
    ► 23: N {lng: 116.290801, lat: 39.960903}
    ► 24: N {lng: 116.290851, lat: 39.960816}
    ► 25: N {lng: 116.290909, lat: 39.957087}
    ► 26: N {lng: 116.289317, lat: 39.957167}
    length: 27
    ► __proto__: Array(0)
  ► __proto__: Object

```

从poi集合中提取出具有边界信息的点，并将边界信息添加进poi信息集合中。如下：

```

▼ 0:
  UID: "08f1f1f5a2f139ca89db73ac"
  address: "北京市海淀区四季青桥常青园路"
  ► boundary: (36) [N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N]
  city: "北京市"
  ► point: N {lng: 116.277881, lat: 39.962019}
  province: "北京市"
  ► tag: (2) ["<font color="#c60a00">地产小区</font>", "<font color="#c60a00">小区</font>/楼盘"]
  title: "常青园-一区"
  ► __proto__: Object

```

在实验中发现，本地搜索最多返回150条信息，因此需要划块后搜索。

7、源码

<https://github.com/QYongCN/houseboundary.git>

8、数据库

根据已有信息，可以在数据库建两个表，一个表是包含区域内poi信息的POI表，一个是含有边界集合和uid的边界表，主键为UID

uid,address,city,province,title数据类型可以为char或varchar，point需要转换成字符串，boundary如果太长的话转换成字符串可能会溢出。

使用SQLite数据库保存数据。

(1)引入sql.js

```
<script src="https://cdn.bootcss.com/sql.js/0.5.0/js/sql-optimized.js"></script>
```

(2)创建数据库

```
var db = openDatabase("myDB", "1.0", "test db", 100*1024*100);
```

```
var db = new SQL.Database();
```

(3)创建表

```
db.transaction(function (tx) {  
    tx.executeSql('CREATE TABLE IF NOT EXISTS WIN2 (uid unique, title, address,  
city, province)');  
});
```

```
db.run("CREATE TABLE IF NOT EXISTS WIN3 (uid unique, title, address, city,  
province)");
```

(4)插入数据

```
db.transaction(function (tx) {  
    for (var i = 0; i < Community.length; i++){  
        tx.executeSql("INSERT INTO WIN2 (uid, title, address, city, province)  
VALUES (?, ?, ?, ?, ?)", [Community[i]['UID'], Community[i]['title'],  
Community[i]['address'], Community[i]['city'], Community[i]['province']]);  
    }  
});
```

```
for (var i = 0; i < Community.length; i++){  
    db.run("INSERT INTO WIN3 (uid, title, address, city, province) VALUES (?, ?,  
?, ?, ?)", [Community[i]['UID'], Community[i]['title'], Community[i]['address'],  
Community[i]['city'], Community[i]['province']]);  
}
```

(5)查询数据

```
db.transaction(function (tx) {  
    tx.executeSql('SELECT * FROM WIN2', [], function (tx, results) {  
        var len = results.rows.length;  
        console.log(results)  
        for (var i = 0; i < len; i++){  
            console.log(results.rows.item(i));  
        }  
    }, null);  
});
```

```
// Prepare a statement
var stmt = db.prepare("SELECT * FROM WIN3");
stmt.getAsObject({$start:1, $end:1}); // {col1:1, col2:111}

// Bind new values
stmt.bind({$start:1, $end:2});
while(stmt.step()) { //
    var row = stmt.getAsObject();
}
```

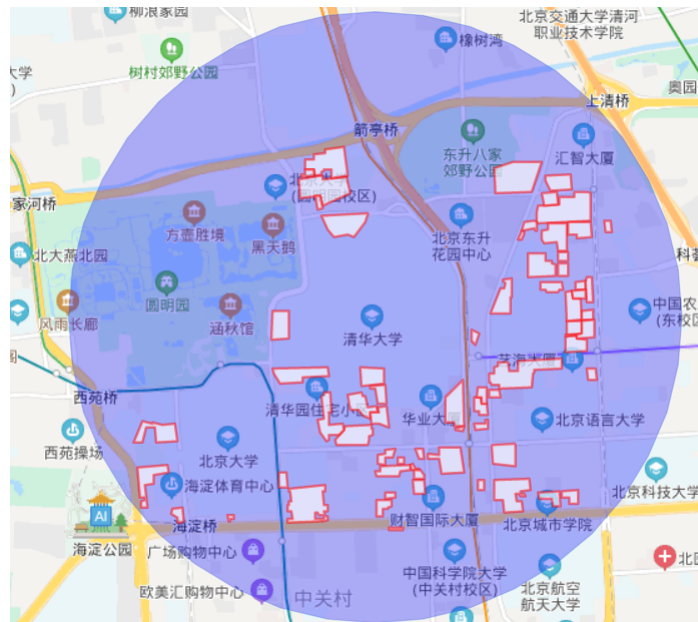
四、圆形范围内所有小区边界

圆形搜索和矩形搜索方法一致，只需要修改查询方式，即将searchInBounds改为searchNearby

```
var circle = new BMap.Circle(point,7000,{fillColor:"blue", strokeweight: 1
,fillOpacity: 0.3, strokeOpacity: 0.3});
map.addOverlay(circle);
```

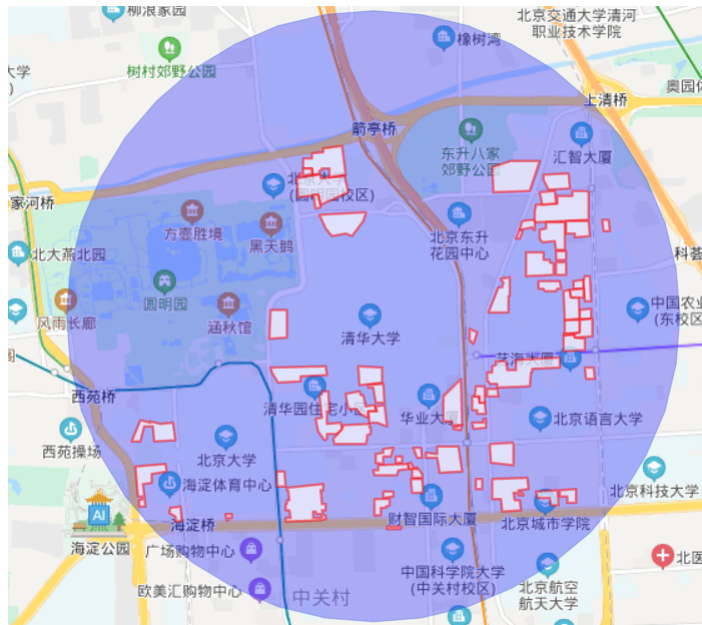
```
local.searchNearby('小区',point,7000);
```

以清华大学坐标 (116.333554,40.010939) 为中心，半径3公里范围内小区边界：

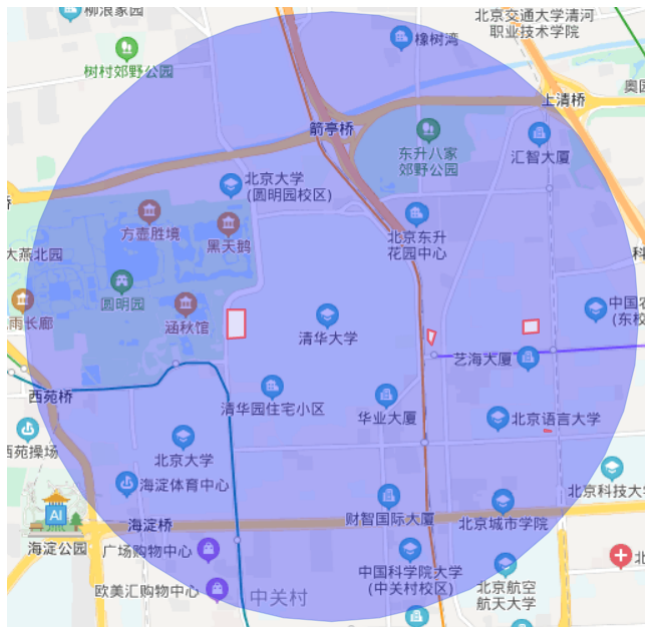


五、已知问题

- 1、使用LocalSearch进行搜索时，最多返回150条POI数据。以圆形搜索为例，当半径为1公里时，可获得40条poi信息，具有边界的poi点有3个，当半径为3公里时，可获得149条poi数据，具有边界的poi点有66个，当半径为7公里时，可获得150条poi信息，当边界为20公里时，返回poi信息仍然只有150条。
- 2、搜索的关键词不同，获得的结果也不同。以圆形搜索为例，当搜索关键词为小区时，结果如下：



当搜索关键词为住宅小区时，结果如下



另外，localSearch支持多关键字搜索，所以需要好好考虑关键词。

3、根据链家网显示的数据，北京目前有11716个小区。用以上方法能不能获取全部小区边界存在疑问。