地图控件使用方法

采用arcgis js api 根据需要使用2D(无显卡电脑,或显卡性能低下)或3D(带显卡电脑)绘制地图

**3D版本请采用webkit内核浏览器运行，并开启webgl**。标题中有标记为3D字样的章节为3D特有功能.

##### 地图控件地址

云版本地址

http://[HOSTNAME\_OR\_SERVERIP]/citygis/areamap/WidgetPages/WidgetGIS.html?code=&token=&maptype=3d

**code参数说明 city为住建委全市，cityga为公安版全市，其他为区县代码 和街道代码** maptype=2d为无显卡电脑使用,

**[HOSTNAME\_OR\_SERVERIP]----控件发布的服务器名称或IP**

##### 地图控件接口API地址

云版本地址

http://[HOSTNAME\_OR\_SERVERIP]/citygis/areamap/CityGis.Bridge.js

##### 三、地图控件测试地址

云版测试

http://[HOSTNAME\_OR\_SERVERIP]/citygis/areamap/Test/index.html

##### 四、socket通讯方式参考

(自行研究或请建坤协助,目前仅限于小木桥场景使用)

云版本地址

http://[HOSTNAME\_OR\_SERVERIP]/citygis/areamap/index.html?code=&token=

**code参数说明 city为住建委全市，cityga为公安版全市，其他为区县代码 和街道代码**

##### 五、IFRAME调用示例(初始化后定位到一个点)

<head>

<script src="http://[HOSTNAME\_OR\_SERVERIP]/citygis/areamap/CityGis.Bridge.js"></script>

<script>

var bridge = new CityGis.Bridge({

id: "map",

url: "http://[HOSTNAME\_OR\_SERVERIP]/citygis/areamap/WidgetPages/WidgetGIS.html?code=city&themeid=Gis&devicetype

=lg",

onReady: function () {

//定位闪烁

bridge.Invoke({

"ActionName": "goToPosition",

"Parameters": {

positon: {

x: 0,

y: 0,

z: 10

}

}

});

}

});

//鼠标操作地图回发消息处理

bridge.addEventListener(function (arg) {

switch(arg.action)

{

case "mapclick":

console.log(JSON.stringify(arg.data));

//地图点选消息

break;

case "ResetMap":

//地图重置完成消息

break;

case "changeTheme":

//地图主题切换消息

break;

case "Clear":

//地图清空消息

break;

}

// eval(arg.action).apply(window, [arg.data]);

}, this);

</script>

</head>

<body>

<iframe id="map" class="mapiframe" frameborder="no" scrolling="no" allowtransparency="true">

</iframe>

</body>

地图控件API

##### 地图基本功能函数（说明：命令为字符串，参数为json对象）

地图响应: bridge.Invoke({

"ActionName": 命令,

"Parameters":参数

})

###### 1．重置

|  |  |
| --- | --- |
| 命令 | ResetMap |
| 参数 | 无 |

###### 2．清空地图

|  |  |  |
| --- | --- | --- |
| 命令 | Clear | 清除 |
| 参数 | {  "position": false,  "search": false  } | position 如果为true 只清除定位点  search 如果为true 只清除地图绘制图形  默认无参数或全部为false 时地图只保留底图 |

###### 3．放大

|  |  |
| --- | --- |
| 命令 | ZoomIn |
| 参数 | 无 |

###### 4．缩小

|  |  |
| --- | --- |
| 命令 | ZoomOut |
| 参数 | 无 |

###### 5．全景

|  |  |
| --- | --- |
| 命令 | FullExtent |
| 参数 | 无 |

###### 6．遥感

|  |  |
| --- | --- |
| **命令** | ChangeMapTheme |
| 参数 | {mapthemeid:'yaogantheme'}  返回默认地图参数为  {mapthemeid:'basetheme  '} |

###### 7．测距

|  |  |  |
| --- | --- | --- |
| 命令 | Measure | 触发/关闭地图功能 |
| 参数 | { type: 'length', isuse: true } | Isuse 为true开启测距，false关闭测距，并清除地图上的测距内容 |

###### 8．测面积

|  |  |  |
| --- | --- | --- |
| 命令 | Measure | 触发/关闭地图功能 |
| 参数 | { type: 'area', isuse: true} | Isuse 为true开启测面积，false关闭测面积，并清除地图上的测面积内容 |

###### 9．剖切面(3D)

|  |  |  |
| --- | --- | --- |
| 命令 | cuttingPlane | 触发/关闭地图功能 |
| 参数 | 无 |  |

###### 10．搜索(已关闭,仅提供地址检索服务)

|  |  |  |
| --- | --- | --- |
| 命令 | SearchUI | 触发地图上搜索面板 |
| 参数 | { visible:true } | true为打开，false关闭 |

###### 11．图层显示

[【图层名字参考】](https://docs.qq.com/sheet/DZktqYkVHSXdDWklU)

|  |  |  |
| --- | --- | --- |
| 命令 | LayerVisible | 图层开关 |
| 参数 | {name: "上水管线", visible:true,  isfilter: true,  legendVisible:true,  popupEnabled:true,  where:"所在路名='真南路'",  "popupUrl": "http://192.168.123.103/map/test.html?width=300&height=200"  } | name: 图层名字  true为打开，false关闭  Isfilter 为true where条件有效,  legendVisible:是否显示图例,  popupEnabled:是否弹出默认信息框，默认为true，显示  popupUrl:弹出框 |

###### 12.鹰眼(测试版)

|  |  |  |
| --- | --- | --- |
| 命令 | OverviewMap | 地图鹰眼 |
| 参数 | {  "layers": [ "街道边界" ],  "visible": true  } | layers: 图层名字  visible：true为打开，false关闭 |

###### 13.自定义菜单(3D)

|  |  |  |
| --- | --- | --- |
| 命令 | userMenu | 地图右侧菜单栏 |
| 参数 | {  "url": "../MapUI/mapmeun/mapmeun.html?code=city",  "visible": true,  "position":"bottom-right"  "cssText": "width:156px;height: 500px"  } | url: 菜单地址  visible：true为打开，  position:参数为:bottom-left | bottom-right | bottom-trailing | top-leading | top-left | top-right  cssText:菜单样式，默认不需要设置 |

###### 14.自定义图例(3D)

注:当地图重置或清空方法后该方法会自动释放.

|  |  |  |
| --- | --- | --- |
| 命令 | userLegend | 地图左侧菜单栏 |
| 参数 | {  "url": "../MapUI/mapmeun/mapmeun.html?code=city",  "visible": true,  "cssText": "width:156px;height: 500px"  } | url: 图例地址  visible：true为打开，  cssText:菜单样式，默认不用设置 |

###### 15.自定义地图上层UI边距(3D)

|  |  |  |
| --- | --- | --- |
| 命令 | padding | 地图上图例和菜单边距控制 |
| 参数 | {  "left": 300,  "top": 0,  "right": 0,  "bottom": 0  } | "left": 300,  "top": 0,  "right": 0,  "bottom": 0 |

###### 16.天气效果(3D)

|  |  |  |
| --- | --- | --- |
| 命令 | ShowData | 模拟天气 |
| 参数 | {  "name": "car\_layer",  "type": "weather",  "renderer": {  "type": "simple",  "label": "强雷阵雨",  "symbol": {  "type": "simple-marker"  }  }  } | renderer.label目前支持天气效果:小雨,中雨  大雨,强雷阵雨,雨夹雪,  雪,小雪,大雪,中雪 |

##### 定位

###### 获取特定点相机位置(3D)

|  |  |  |
| --- | --- | --- |
| 命令 | getCamera | 打印当前中心 |
| 全部参数 | 无 |  |
| 控制打印数据 | {  center: {  x:0,  y:0,  z:0  }  heading: 0  scale: 430375.90447438153  tilt: 0.49999999999913347  } | 可使用实例获取  bridge.**getCamera**().**then**(function(c){  **console**.**log**(c);                          }) |

###### 2.点定位

|  |  |  |
| --- | --- | --- |
| 命令 | goToPosition | 定位到地图上某点 |
| 全部参数 | {  mode:"replace",  positon: {x: -18357, y: 59449, z: 10},  heading:0,  tilt:45,  hasImg:true,  marker: {  size: 16,  colorType: 0,  color:"#ff0000"  },  isfilter:false,  radius:1000， | mode:模式 replace/delete  positon 点坐标 可选(当不传入 只改变相机角度)  heading航向角度(0~360)  tilt 偏航角度 (0~90)  hasimg 是否标点，默认标点  marker.size 闪烁点大小  marker.colorType 闪烁点颜色 0,1,2,3,4,  marker.color 当设置color后colorType 失效.  isfilter:过滤图形  radius 显示周边时使用的半径。默认1000米  zoom：缩放级别，默认为8 |
| 示例1参数 | {  positon: {x: -18357, y: 59449, z: 10},  hasImg:false  } | 无标点定位 |
| 示例2参数 | {  positon: {x: -18357, y: 59449, z: 10},  hasImg:true  } | 有标点定位 |
| 示例3参数（测试） | {  positon: {x: -18357, y: 59449, z: 10},  hasImg:true，  isfilter:true  } | 有标点，并显示周边1000半径半径，针对定义撒点只会在该范围内显示 |

###### 3.区域定位

|  |  |  |
| --- | --- | --- |
| 命令 | ShowData | 根据动态图层数据定位 |
| 全部参数 | {  "name": "car\_layer",  "type": "layer",  "isLocate": true, "highLight": {  "color": "blue"  },  "data": {  "layers": {  "name": "区县边界",  "where": "区代码='06'"  }  },  "renderer": {  "type": "simple",  "symbol": {  "type": "line-3d",  "symbolLayers": [  {  "type": "line",  "size": 5,  "material": {  "color": "red"  }  }  ]  }  }  } | name 用户定义的名称唯一  type: 固定为layer  isLocate 是否定位到区域范围  highLight 高亮属性  highLight.color 颜色  data 数据源  data.layers 服务图层数据源  data.layers.name 图层名  data.layers.where 筛选条件  renderer 根据图层属性分点线面 定义对应的渲染符号 |

区县图层定位示例

{

"ActionName": "ShowData",

"Parameters": {

"name": "car\_layer",

"type": "layer",

"isLocate": true,

"highLight": {

"color": "blue"

},

"data": {

"layers": {

"name": "区县边界",

"where": "区代码='06'"

}

},

"labels": [

{

"fields": [

"#.名称",

"#.序号"

],

"color": [

255,

255,

0,

0.5

],

"size": 12

}

],

"renderer": {

"type": "simple",

"symbol": {

"type": "line-3d",

"symbolLayers": [

{

"type": "line",

"size": 5,

"material": {

"color": "red"

}

}

]

}

}

}

}

###### 4.清除动画点(已移至清空地图方法)

|  |  |  |
| --- | --- | --- |
| 命令 | ClearPostion | 清除动画闪烁点 |
| 全部参数 | 无 |  |
|  |  |  |

##### 草图绘制

###### 绘制点

|  |  |  |
| --- | --- | --- |
| 命令 | ShowData | 地图上标点 |
| 全部参数 | {    "name": "car\_layer",  "type":"point",  ”mode":"replace",  "popupEnabled": true,  "legendVisible": true,  "isLocate": true,  "isFiltered":true,  "isMouseHoverInfo": true,  "popupUrl": "http://10.81.71.48/shscimp/bullet?width=900&height=500&CName=vlp&dockposition=top-left",  "data": {  "url": "../Assets/data/point.json",  "parsedata": "function(d){return d.features}",  "parsegeometry": "function(item){return item.geometry}"  },  "labels": [  {  "fields": [  "#.名称",  "#.地址"  ],  "color": [  255,  255,  0,  0.5  ],  "size": 30  }  ],  "renderer": {  "type": "simple",  "label": "测试车辆",  "visualVariables": [],  "symbol": {  "type": "simple-marker",  "size": 6,  "color": "black",  "outline": {  "width": 0.5,  "color": "white"  }  }  }  } | name: 唯一名称  type:"point" 默认为点  mode: 操作3种模式默认replace：  "add": 增加  ”replace”：如果name已  存在则替换  "delete":删除  popupEnabled 是否弹框  legendVisible 是否显示图例  isLocate：地图是否缩放到最大外接矩形  isFiltered:参数图形过滤,  isMouseHoverInfo true开启鼠标移入移出显示信息  popupUrl 地图点选后弹框嵌入具体参考附录  data 数据解析(参考附录3)  url: 数据api  parsedata 返回空间数据数组  parsegeometry 返回空间数据  labels 标注  fields 标注字段  color 颜色  size 大小  *labelPlacement标注位置*  renderer 地图呈显  参考arcgis renderer |

###### 绘制线

|  |  |  |
| --- | --- | --- |
| 命令 | ShowData | 地图上标点 |
| 全部参数 | {    "name": "car\_layer",  "type":"polyline",  ”mode":"replace",  "popupEnabled": true,  "legendVisible": true,  "islocate": true,  "isMouseHoverInfo": true,  "popupUrl": "http://10.81.71.48/shscimp/bullet?width=900&height=500&CName=vlp&dockposition=top-left",  "data": {  "url": "../Assets/data/point.json",  "parsedata": "function(d){return d.features}",  "parsegeometry": "function(item){return item.geometry}"  },  "labels": [  {  "fields": [  "#.名称",  "#.地址"  ],  "color": [  255,  255,  0,  0.5  ],  "size": 30  }  ],  "renderer": {  "type": "simple",  "label": "测试车辆",  "visualVariables": [],  "symbol": {  "type": "line-3d",  "symbolLayers": [  {  "type": "line",  "size": 2,  "material": {  "color": "black"  },  "cap": "round",  "join": "round"  }  ]  }  }  } | name: 唯一名称  type:线类型"polyline"  mode: 操作3种模式默认replace：  "add": 增加  ”replace”：如果name已  存在则替换  "delete":删除  popupEnabled 是否弹框  legendVisible 是否显示图例  islocate：地图是否缩放到最大外接矩形  isMouseHoverInfo true开启鼠标移入移出显示信息  popupUrl 地图点选后弹框嵌入具体参考附录  data 数据解析(参考附录)  url: 数据api  parsedata 返回空间数据数组  parsegeometry 返回空间数据  labels 标注  fields 标注字段  color 颜色  size 大小  renderer 地图呈显  参考arcgis renderer |

###### 绘制面

|  |  |  |
| --- | --- | --- |
| 命令 | ShowData | 地图上标点 |
| 全部参数 | {    "name": "car\_layer",  "type":"polygon",  ”mode":"replace",  "popupEnabled": true,  "legendVisible": true,  "islocate": true,  "isMouseHoverInfo": true,  "popupUrl": "http://10.81.71.48/shscimp/bullet?width=900&height=500&CName=vlp&dockposition=top-left",  "data": {  "url": "../Assets/data/point.json",  "parsedata": "function(d){return d.features}",  "parsegeometry": "function(item){return item.geometry}"  },  "labels": [  {  "fields": [  "#.名称",  "#.地址"  ],  "color": [  255,  255,  0,  0.5  ],  "size": 30  }  ],  "renderer": {  "type": "simple",  "label": "测试车辆",  "visualVariables": [],  "symbol": {  "type": "polygon-3d",  "symbolLayers": [  {  "type": "fill",  "material": {  "color": "red"  }  }  ]  }  }  } | name: 唯一名称  type:面类型"polygon"  mode: 操作3种模式默认replace：  "add": 增加  ”replace”：如果name已  存在则替换  "delete":删除  popupEnabled 是否弹框  legendVisible 是否显示图例  islocate：地图是否缩放到最大外接矩形  isMouseHoverInfo true开启鼠标移入移出显示信息  popupUrl 地图点选后弹框嵌入具体参考附录  data 数据解析(参考附录3)  url: 数据api  parsedata 返回空间数据数组  parsegeometry 返回空间数据  labels 标注  fields 标注字段  color 颜色  size 大小  renderer 地图呈显  参考arcgis renderer |

###### 基于图层服务的绘制

|  |  |  |
| --- | --- | --- |
| 命令 | ShowData | 地图上标点 |
| 全部参数 | {  "ActionName": "ShowData",  "Parameters": {  "name": "car\_layer",  "type": "layer",  "legendVisible": true,  "popupEnabled": true,  "isLocate": true,  "isMouseHoverInfo": true,  "popupUrl": "http://10.81.71.48/shscimp/bullet?width=900&height=500&CName=vlp&dockposition=top-left",  "data": {  "layers": [  {  "name": "区县边界",  "where": "区代码='06'"  }  ]  }  }  } | name: 唯一名称  type:固定参数"layer"  mode: 操作3种模式默认replace：  "add": 增加  ”replace”：如果name已  存在则替换  "delete":删除  popupEnabled 是否弹框  legendVisible 是否显示图例  islocate：地图是否缩放到最大外接矩形  isMouseHoverInfo true开启鼠标移入移出显示信息  popupUrl 地图点选后弹框嵌入具体参考附录  data 数据项  layers: 图层集  name 对应后台服务名称  where 筛选条件  多条件 "where": "区代码='06' or 区代码='01'"  labels 标注  renderer 地图呈显  参考arcgis renderer |

###### 5.地图显示点(弃用)

|  |  |  |
| --- | --- | --- |
| 命令 | doShowPointData（弃用） | 地图上标点 |
| 全部参数 | {  dataSource: [],  postUrl: "<http://www.china-gis.com:8080/api/c/f/p>",  keyfield: "ID",  popupUrl: "http://10.231.26.17:8018/fangxun/shuizha/shuizha.html?width=1193&height=843&CName=ID",  updateinfo: {  updateurl: "<http://www.china-gis.com:8080/api/c/f/p>",  updatefields: ["status", "value"],  interval: 60  },  options: {  layerid: "test1",  geofield: {  XFIELD: "COORDX",  YFIELD: "COORDY",  ZFIELD: "COORDZ"  },  textoption: [{  where: "1=1",  minscale: 4000,  maxscale: 500,  textformat: ["#.NAME", "'潮位:'+#.OUTWATER"];,  placement: "center-right",  textstyle: {  color: [255, 255, 0, 0.5],  size: 30,  halo: {  color: [164, 207, 251],  size: 4  }  }  }],  layerRender: {},  }  } | dataSource 点数据源为数组  postUrl数据请求的API  keyfield 数据源主键  popupUrl 信息面板框地址  updateinfo 指定更新参数  updateurl需要更新请求数据的API为空则调用postUrl  updatefields指定需要更新的字段  Interval 刷新时间间隔秒  options 参数项  数据源中坐标字段的指定:  XFIELD x坐标字段名称  YFIELD y坐标字段名称  ZFIELD z坐标字段名称（无可以不指定）  layerid 标记该次标点  width面板宽度  Height面板高度  CName 页面需要接收的参数，根据页面设定，ID为API数据中对应的字段名  textoption 标点的时候显示属性  textformat 多行属性显示  textstyle属性字体颜色，大小  halo 外光圈颜色，大小  layerRender 见地图样式渲染 |

###### 6.移除标点层(弃用)

|  |  |  |
| --- | --- | --- |
| 命令 | doRemoveShowData（弃用） | 移除地图上标点 |
| 参数 | [layerid] | 移除上面标点的layerid: "test1",  参数为["test1"] |

##### 四、绘线(弃用)

|  |  |  |
| --- | --- | --- |
| 命令 | ShowPath（弃用） |  |
| 全部参数 | {  url:"",  xfield: "X",  yfield: "Y",  multiple: true,  visible:true,  isZoom: true,  isPlay:false,  size:’3px’,  color:[255,0,0]  } or  {  multiple: true,  visible:true,  isPlay:false,  data: [{ paths: [[0,0],[1000,5000]],id:"1"}],  size:’3px’,  color:[255,0,0]  } | 方法一  Url API获取线数据  x坐标字段  y坐标字段  前一次绘制的线清除  visible:为false时清空  缩放到线范围  sPlay:不进行播放  size线宽度，  color线颜色  方法二  前一次绘制的线清除  数据源格式为数组传入  visible:为false时清空  isPlay:不进行播放  size线宽度  color线颜色 |

##### 五 色块

|  |  |  |
| --- | --- | --- |
| 命令 | ShowData |  |
| 全部参数 | {  "name": "car\_layer",  "type": "layer",  "isLocate": true,  "data": {  "url": "../Data/area.json",  "layers": {  "name": "区县边界",  "where": "区县代码='06'"  },  "join": "区县名称=name"  },  "labels": [  {  "fields": [  "#.区县名称",  "#.from"  ],  "color": [  255,  255,  0,  0.5  ],  "size": 12  }  ],  "renderer": {  "type": "unique-value",  "field": "from",  "uniqueValueInfos": [  {  "value": "null",  "symbol": {  "type": "simple-fill",  "color": [  128,  216,  33,  0.5  ],  "style": "solid",  "outline": {  "color": [  15,  185,  208,  0.5  ],  "width": 1  }  }  },  {  "value": "2",  "symbol": {  "type": "simple-fill",  "color": [  128,  33,  255,  0.5  ],  "style": "solid",  "outline": {  "color": [  15,  0,  208,  0.5  ],  "width": 1  }  }  },  {  "value": "3",  "symbol": {  "type": "simple-fill",  "color": [  0,  216,  255,  0.5  ],  "style": "solid",  "outline": {  "color": [  15,  185,  208,  0.5  ],  "width": 1  }  }  }  ]  }  } |  |

|  |  |  |
| --- | --- | --- |
| 命令1 | doQueryLocate |  |
| 全部参数 | {  layername: "区县边界",  popupUrl: "http://192.168.123.103/citygis/map716/WidgetPages/WidgetGIS.html?width=300&height=200",  where: "1=1",  options: {  layerid: "streetlayer2",  minscale: 0,//default is 0; 0 mean no set  maxscale: 0,// default is 0; 0 mean no set  islocate: false,//default is true,true is to locate queryresults  isshow: true,//default is true,true is to display queryresults in map  textoption: {  textfield: "SHOWTEXT",  textstyle: {  color: [255, 255, 0, 0.5],  size: 30  },  isalone: true //是否单独配置相应的标注图层 默认为false  },  appendinfo: [  { AREACODE: "06", 产值: 100, SHOWTEXT: "产值:100(万元)" },  { AREACODE: "30", 产值: 200, SHOWTEXT: "产值:200(万元)" },  { AREACODE: "12", 产值: 300,SHOWTEXT: "产值:300(万元)" },  { AREACODE: "13", 产值: 400, SHOWTEXT: "产值:400(万元)" }],  relatefield: { infield: "区县代码", outfield: "AREACODE" },  layerRender: {  type: "class-breaks",  field: "产值",  defaultSymbol: {  symbol: {  type: "simple-fill",  color: "orange",  outline: { // autocasts as new SimpleLineSymbol()  color: [100, 100, 100, 0.5],  width: "0.5px"  }  }  },  classBreakInfos: [  {  minValue: 1,  maxValue: 150,  symbol: {  type: "simple-fill",  color: "red",  outline: { // autocasts as new SimpleLineSymbol()  color: [128, 128, 128, 0.5],  width: "0.5px"  }  }  }, {  minValue: 150,  maxValue: 250,  symbol: {  type: "simple-fill",  color: "blue",  outline: { // autocasts as new SimpleLineSymbol()  color: [64, 196, 255, 0.5],  width: "0.5px"  }  }  }, {  minValue: 250,  maxValue: 350,  symbol: {  type: "simple-fill",  color: "green",  outline: { // autocasts as new SimpleLineSymbol()  color: [128, 216, 255, 0.5],  width: "0.5px"  }  }  },  {  minValue: 350,  maxValue: 450,  symbol: {  type: "simple-fill",  color: "orange",  outline: { // autocasts as new SimpleLineSymbol()  color: [128, 216, 255, 0.5],  width: "0.5px"  }  }  }]  }  }  } |  |

##### 六、热力(3D平面)

|  |  |  |
| --- | --- | --- |
| 命令 | doShowHeat |  |
| 全部参数 | datasource :[{x:0,y:0,count:10}，{x:1000,y:0,count:100}，{x:0,y:1000,count:50}],  postUrl:"", visible:true,  weightFied:"count" ,  gradient: {  '.90': '#4ef1b2',  '.95': '#7cd346',  '.97': '#f3f12c',  '.99': '#fd2f02'  },  radius:35,  geofield:{  xfield:"X",  yfield:"Y",  zfield:"Z"  }    } | datasource 点数据源为数组  postUrl数据请求的API  visible 是否显示  weightFied数据源中权重字段名 |

##### 七、迁徙图(3D)

##### 八、三维

###### 白模(当参数visible为true则显示反之则隐藏)

{

"ActionName": "themeLayer",

"Parameters": {

name: "建筑白模",

visible: true //为true则显示反之则隐藏

renderer: {//可选节点

"type": "simple",

"symbol": {

"type": "mesh-3d",

"symbolLayers": [

{

"type": "fill",

"material": {

"color": [

255,

255,

255,

0

],

"colorMixMode": "replace"

}

}

]

}

}

}

}

###### 精模

{

"ActionName": "themeLayer",

"Parameters": {

name: "建筑精模",

visible: true //为true则显示反之则隐藏

}

}

###### 3.特殊应用可参考场景应用功能

##### 九、轨迹(3D)

|  |  |  |
| --- | --- | --- |
| 命令 | ShowData | 地图上展示轨迹 |
| 全部参数 | {  "name": "car\_layer1",  "type": "track",  "mode":"replace",  "popupEnabled":true,  "legendVisible":true  "islocate":true,  "data": {  "url": "../Data/area.json",  "isComputingRote": true,  "parsedata": "function(d){return d}",  "parsegeometry": "function(item){return {x:Number(item.x),y:Number(item.y)}}"  },  "legendVisible": true,  "popupEnabled": true,  "renderer": {  "type": "unique-value",  "uniqueValueInfos": [  {  "value": "line",  "symbol": {  "type": "line-3d",  "symbolLayers": [  {  "type": "line",  "size": 2,  "material": {  "color": "black"  },  "cap": "round",  "join": "round"  }  ]  }  },  {  "value": "start",  "symbol": {  "type": "simple-marker",  "size": 20,  "color": "red",  "outline": {  "width": 0.5,  "color": "white"  }  }  },  {  "value": "end",  "symbol": {  "type": "simple-marker",  "size": 20,  "color": "orange",  "outline": {  "width": 0.5,  "color": "white"  }  }  }  ]  }  } | name: 唯一名称  type:"track"  mode: 操作3种模式默认replace：  "add": 增加  ”replace”：如果name已  存在则替换  "delete":删除  popupEnabled 是否弹框  legendVisible 是否显示图例  islocate：地图是否缩放到最大外接矩形  data 数据解析  url: 数据api  "isComputingRote": 是否拟合道路,默认为false,  parsedata 返回空间数据数组  parsegeometry 返回空间数据  renderer 地图呈显  固定格式,只能修改symbol节点 |

|  |  |  |
| --- | --- | --- |
| 命令 | ShowPath(弃用) | 默认自动播放 |
| 全部参数 | {  speed: 50,  color: [255, 0, 0], linecolor: [0, 255, 0],  visible:true,  markimg: "http://192.168.123.103/citygis/map7131/Assets/image/RealTimeLocation/Track/online.png",  multiple: true,  url: "http://192.168.29.100/sequences/vehiclePosition?pageSize=10000&vlp=%E6%B2%AAAKA300",  xfield: "x",  yfield: "y"  } | speed 播放速度  color: 底线颜色  linecolor 已播放线颜色  visible:为false时清空  markimg 播放图标  multiple 是否单轨迹  url数据请求的API  xfield 数据源中的x坐标字段  yfield 数据源中的x坐标字段 |

##### 十、标绘动画点(3D)

|  |  |  |
| --- | --- | --- |
| 命令 | FlashPoints | 地图上标闪烁点 |
| 全部参数 | {  geofield:{  XFIELD:"x",  YFIELD:"y",  ZFIELD:"z"  },  data:[]||{url:"<http://www.china-gis.com:8080/api/c/f/p>"}    keyfield:"ID", popupUrl:"http://10.231.26.17:8018/fangxun/shuizha/shuizha.html?width=1193&height=843&CName=ID"  "size": 16,  "visible": true,  "color": "#00fff" | 数据源中坐标字段的指定  XFIELD x坐标字段名称  YFIELD y坐标字段名称  ZFIELD z坐标字段名称（无可以不指定）  data 数据源  data可传数组数据或者  传入api url  Keyfield 数据源主键  popupUrl 信息面板框地址  width面板宽度  Height面板高度  CName 页面需要接收的参数，根据页面设定，ID为API数据中对应的字段名  图标大小  是否显示  颜色 |

##### 十一、监督员轨迹(3D)

|  |  |  |
| --- | --- | --- |
| 命令 | CommTracks | 地图上标闪烁点 |
| 全部参数 | params={  "action": "showTrack",  trackSymbol:{}, //轨迹线样式  startPointSymbol:{},//起始点样式  endPointSymbol:{},//结束点样式  playSymbol:{  type: "picture-marker",  url: "Keeper/keeperBig.png",  width: 150,  height: 150},//播放时样式  geoFields: {  data: [  {  "COORDX": "5189.324",  "COORDY": "268.982"  },  {  "COORDX": "6967.16",  "COORDY": "6568.582"  }]  },  speed:播放速度  width:5 线宽  color :"green"  playcolor："red"  } | 数据源中坐标字段的指定  atcion: "showTrack"展示轨迹  "playTrack" 播放轨迹  "stopTrack"停止播放  "clearTrack"清除轨迹  起始和终止点样式不指定显示  speed:播放速度  geoFields.data：监督员数据，  COORDX x坐标字段名称  COORDY y坐标字段名称  width:线宽，不指定默认为5  color:初始化颜色  playcolor: 播放颜色 |

##### 十二 聚合(3D)

|  |  |  |
| --- | --- | --- |
| 命令 | ShowData | 地图上根据区域聚合 |
| 全部参数 | {  "name": "car\_layer",  "type": "cluster",  "isLocate": true,  "data": {  "url": "http://172.16.21.40/node/server/preControl/get\_TaskInfo",  "parsedata": "function(d){return d.data.task}",  "parsegeometry": "function(item){return {x:Number(item.X),y:Number(item.Y)}}",  "layers": [  {  "name": "街道乡镇",  "where": "区代码='12'",  "join": "街道名称=STREETNAME",  "subTypeFlareProperty": "TASKID",  "singleFlareTooltipProperty": "TASKID",  "renderer": {  "uniqueValueInfos": [  {  "value": "top-5",  "symbol": {  "type": "simple-marker",  "size": "22",  "color": [  255,  204,  102,  0.8  ],  "outline": {  "color": [  221,  159,  34,  0.8  ]  }  }  },  {  "value": "midder-5",  "symbol": {  "type": "simple-marker",  "size": "22",  "color": [  102,  204,  255,  0.4  ],  "outline": {  "color": [  82,  163,  204,  0.8  ]  }  }  },  {  "value": "later-5",  "symbol": {  "type": "simple-marker",  "size": "22",  "color": [  51,  204,  51,  0.4  ],  "outline": {  "color": [  41,  163,  41,  0.8  ]  }  }  }  ]  },  "cacul": "GCI",  "minScale": 1000,  "maxScale": 104001  }  ]  },  "popupTemplate": {  "title": "{门牌地址}",  "content": [  {  "type": "fields",  "fieldInfos": [  {  "fieldName": "所属区县",  "label": "区县名",  "visible": true  },  {  "fieldName": "街道名称",  "label": "街道名",  "visible": true  },  {  "fieldName": "路",  "label": "路名",  "visible": true  },  {  "fieldName": "号",  "label": "号名",  "visible": true  }  ]  }  ]  },  "renderer": {  "type": "unique-value",  "field": "STATUSNAME",  "uniqueValueInfos": [  {  "value": "已结案",  "label": "已结案",  "symbol": {  "type": "simple-marker",  "size": 20,  "color": "green",  "outline": {  "width": 0.5,  "color": "white"  }  }  },  {  "value": "待申请核查",  "label": "待申请核查",  "symbol": {  "type": "simple-marker",  "size": 20,  "color": "red",  "outline": {  "width": 0.5,  "color": "white"  }  }  },  {  "value": "已下发核查",  "label": "已下发核查",  "symbol": {  "type": "simple-marker",  "size": 20,  "color": "orange",  "outline": {  "width": 0.5,  "color": "white"  }  }  }  ]  }  } | name: 唯一名称  type:"cluster"  mode: 操作3种模式默认replace：  "add": 增加  ”replace”：如果name已  存在则替换  "delete":删除  popupEnabled 是否弹框  legendVisible 是否显示图例  islocate：地图是否缩放到最大外接矩形  data 数据解析  url: 数据api  parsedata 返回空间数据数组  parsegeometry 返回空间数据  data.layers 集合依赖  [  name:图层名  where:筛选条件  join 图层字段与url字段关联关系  subTypeFlareProperty 子类聚合属性  singleFlareTooltipProperty:  不集合后显示属性  renderer 固定格式  只可修改symbol  "cacul": "GCI",渲染规则  "minScale": 最小比率  "maxScale":最大比率  ]  renderer: 不聚合后显示符号 |

# 空间分析(3D)

需配合数据检索,草图绘制使用

##### 1.缓冲区分析

###### 1.1 地图上动态绘制图形生成缓冲区

|  |  |  |
| --- | --- | --- |
| **命令** | DrawAddFilter | 说明 |
| 参数 | {  "type": "point",  "isfilter": true,  "distance": 200  } | type 可选"point","polyline","circle","polygon" |

###### 1.2 自定义面状数据的空间分析

|  |  |  |
| --- | --- | --- |
| **命令** | ShowData | 说明 |
| 参数 | {  "name": "kjfx",  "type": "polygon",  "filterMode": "all",  "legendVisible": false,  "popupEnabled": false,  "isLocate": true,  "data": {  "url": "../Data/filterarea.json",  "parsedata": "function(d){return d.features}",  "parsegeometry": "function(item){return {rings:item.geometry.rings}}"  },  "renderer": {  "type": "simple",  "visualVariables": [],  "symbol": {  "type": "polygon-3d",  "symbolLayers": [  {  "type": "fill",  "material": {  "color": [  0,  0,  0,  0  ]  },  "outline": {  "color": "red",  "size": "5px"  }  }  ]  }  }  } |  |

# 数据检索(3D)

###### 获取服务查询的服务信息

|  |  |
| --- | --- |
| **命令** | getQueryService |
| 参数 |  |

###### 2属性查询

1.基于iframe的使用

var bridge = new CityGis.Bridge(...);

bridge.QueryTask({name:"街道乡镇组",returnGeometry:true,where:"区代码='06'"})

.then(function(d){

console.log(JSON.stringify(d));

});

2.基于sokect的使用

options参数 使用getQueryService命令返回的结果

var task = new CityGis.Task(options);

task.QueryTask({name:"街道乡镇组",returnGeometry:true,where:"区代码='06'"})

.then(function(d){

console.log(JSON.stringify(d));

});

###### 3 空间查询

1.基于iframe的使用

arg.data 可以使用DrawGraphic返回的结果，也可以自己组织arcgis geometry格式

var bridge = new CityGis.Bridge(...);

bridge.addEventListener(function (arg) {

if(arg.action=="DrawGraphic")

{

bridge.QueryTask({name:"区县边界组",returnGeometry:true,geometry:arg.data})

.then(function(d){

console.log(JSON.stringify(d));

});

}

})

2.基于sokect的使用

options参数 使用getQueryService命令返回的结果

var task = new CityGis.Task(options);

task.QueryTask({name:"区县边界组",returnGeometry:true,geometry:arg.data})

.then(function(d){

console.log(JSON.stringify(d));

});

地图样式渲染(可参考arcgis Renderer)

##### 一、点样式

###### 1.单一渲染

{

type: "simple",

symbol: 点符号

}

###### 2.根据属性分值渲染(示例说明：当数据源中字段为“类型”的列有0，1，2三个值则根据这3个值渲染各自对应的符号)

{

type: "unique-value",

field: "类型",

defaultSymbol: 点符号,

uniqueValueInfos: [{

value: "0",

symbol: 点符号

}, {

value: "1",

symbol: 点符号

}, {

value: "2",

symbol: 点符号

}]

}

###### 3.根据属性分级渲染(示例说明：当数据源中字段为“产值”的列最小为1，最大为20 则根据1~10渲染一个符号，10~20的渲染成另一个符号)

{

type: "class-breaks",

field: "产值"，

defaultSymbol: 点符号,

classBreakInfos：[{

minValue: 1,

maxValue: 10,

symbol: 点符号

}, {

minValue: 10,

maxValue: 20,

symbol: 点符号

}]

}

##### 二、线样式

同点样式,symbol节点改为线符号

##### 三、面样式

同点样式,symbol节点改为面符号

地图符号(可参考arcgis symbol)

##### 一、点符号

###### 默认字符符号

{

type: "simple-marker",

style: "circle|cross|diamond|square|triangle|x",

color: "blue",

size: "8px", // pixels

outline: {

color: [ 255, 255, 0 ],

width: 3 // points

}

}

###### 2.图片符号

{

type: "picture-marker",

url:"https://static.arcgis.com/images/Symbols/Shapes/BlackStarLargeB.png",

width: "64px",

height: "64px"

};

##### 二、线符号

{

type: "line-3d",

symbolLayers: [{

type: "line",

size: 2, //线宽

material: { color: "black" }, //颜色

cap: "round",

join: "round"

}]

}

##### 面符号

平面 {

type: "simple-fill",

color: [ 51,51, 204, 0.9 ], //面颜色

style: "solid", //面样式

outline: {

color: "white", //边线颜色

width: 1 //边线宽度

}

}

拉伸面 {

type: "polygon-3d",

symbolLayers: [{

type: "extrude",

size: 10, //拉伸高度单位为米 已知高度太小可视范围大的情况存在破面情况

material: { color: "red" } //颜色

edges: { //边线

type: "solid",

color: [50, 50, 50, 0.5] 边线颜色

}

}]

}

场景应用功能

##### 一、玻璃幕墙动态分色功能

{

"ActionName": "themeLayer",

"Parameters": JSON.stringify({

"name": "玻璃幕墙分色",

"visible": true,

"popupUrl": "http://10.121.57.2:38082/partner/blmqWeb\_zj/#!/cw-InfoWindow?buildId=build\_id&width=400&height=400",

"update": {

"url": "http://192.168.29.100/scp-charts/districtblmqs?code=06&callId=201",

"layerkey": "build\_id",

"valuekey": "level1",

"joinfield": "buildId",

"joinvaluefield": "level1"

}

})

}

##### 二、基坑功能

{

"ActionName": "themeLayer",

"Parameters": JSON.stringify({

name: "基坑",

visible: true,

popupUrl: "弹出页面地址?width=300&height=300&id=BUILD\_ID"

})

}

##### 三、场景视频融合

{

"ActionName": "videosVR",

"Parameters": JSON.stringify([{

"id": "camera1",

"x": -4022.549,

"y": 5582.658,

"z": 18,

direction: 38, // 相机朝向(相对北方向)

pitchingAngle: 47, // 相机俯仰角（相对重力方向）

fovAngle: 40, // 相机视场角

imageWidth: 1800, // 图像宽度

imageHeight: 1000, // 图像高度

modelUrl: "../Assets/data/modelProgram1.txt",

videoSrc: 'http://hls01open.ys7.com/openlive/5c18b01fc3624a9788d187fbd2f98b91.m3u8'

}])

}

##### 三、优秀历史建筑（测试版）

{

"ActionName": "themeLayer",

"Parameters": JSON.stringify({

name: "优秀历史建筑",

visible: true,

popupUrl: "弹出页面地址?width=300&height=300&id=BUILD\_ID"

})

}

# 附录参考

###### 1. labelPlacement 可选参数

(点): above-center | above-left | above-right | below-center | below-left | below-right | center-center | center-left | center-right

(线): above-after | above-along | above-before | above-start | above-end | below-after | below-along | below-before | below-start | below-end | center-after | center-along | center-before | center-start | center-end

(面): always-horizontal

###### popupUrl 弹框url配置说明

例如:"popupUrl": "http://10.81.71.48/shscimp/bullet?width=900&height=500&CName=vlp&dockposition=top-left"

说明: width 设置页面宽度

height 设置页面高度

dockposition 设置页面停靠位置,不设置则停靠当前选中锚点.可选参数为

top-left,top-center,top-right,bottom-left,bottom-center,bottom-right

方法1: 当生成图形中数据集包含的列标头,在url中都可以设置其获取其值,比如在数据集的标题头有一列叫"code",则在url 上可拼接参数 mycode = code,当页面运行后code就会自动替换成当前选中的code数值,页面则可根据mycode取值.

例: "popupUrl": "http://10.81.71.48/shscimp/bullet?width=900&height=500&mycode=code"

方法2:

可在url后拼接参数 fjson=json 则可通过代码去监听整条记录的全部返回值

则在url页面初始化时加入下面代码则可获取当前记录的全部数据.

 window.**addEventListener**("message", function (data) {

**console**.**log**(data);

    });

页面中如果需要加入关闭按钮 调用方法

window.parent.**postMessage(**

 {

"ActionName": "userPopup",

"Parameters": {

"action": "close"

}

},地图url)

###### 自定义数据格式说明

注意: demo上json或url的数据仅供参考,非固定格式. data节点说明

方式一: "data": {

"url": "../Assets/data/point.json",

"parsedata": "function(data){return data.features}",

"parsegeometry": "function(row){return row.geometry}"

}

方式二: "data": {

"content": [

{

"x": 0,

"y": 0,

"id": 1

},

{

"x": 100,

"y": 100,

id": 2

}

],

"parsegeometry": "function(item){return {x:item.x,y:item.y}}"

}

当你的url请求返回的数据格式与demo的数据格式不一致时可设置解析函数.如上例:

要求 1.必须获取空间数据的数组

2.必须有空间数据,如点, 线,面

parsedata 函数中data为url请求后返回的全部对象,当返回的json数据包含的时对象而不是数组记录时则需要指定该解析函数让其返回数组对象

parsegeometry (参数row为parsedata解析返回数组中的每条记录)

点函数function(row){return {x:Number(row.x),y:Number(row.y)}}

线函数function(row){return {paths:row.paths}}

面函数function(row){return {rings:row.rings}}

线面数据结构说明

格式1: [[[0,0][100,0]]] (1条连续的线或一个封闭的面)

格式2: [[[0,0][100,0]][[200,300][500,300]]] (多条连续的线或多个封闭的面)