**1.如何发布ArcgisJavascript API应用**

0.准备工作：

       (1).有web应用:

       (2).有jsapi开发包

1.将jsapi目录拷贝到自己的Web服务器目录下

2.在页面中添加引用(2个css,1个js)

<link rel=*"stylesheet"* type=*"text/css"* href=*"jsapi/js/dojo/dijit/themes/claro/claro.css"*>

<link rel=*"stylesheet"* href=*"jsapi/js/esri/css/esri.css"*>

<script src=*"jsapi/init.js"*></script>

3.在页面中添加加载地图的脚本

dojo.ready(**function**(){

   mapObj=**new** esri.Map("map");

**var** districtLayer = **new**esri.layers.ArcGISDynamicMapServiceLayer(“http://localhost:6080/arcgis/rest/services/TF/baseMap/MapServer”, { id: "districLayer"});

   mapObj.addLayer(districtLayer);

});

3.发布Web应用

web应用发布的地址是:

  localost/webLab/layerOper（web服务器发布的根路径）

4.写出访问jsapi/js/dojo/dojo/dojo.js的url

<http://localhost/webLab/layerOper/jsapi/js/dojo/dojo/dojo.js>（发布后的dojo.js路径）

在浏览器中检查

确认地址正确后,截取url一段

localhost/webLab/layerOper/jsapi/js/dojo/dojo

5.将Url片段替换init.js中的baseUrl:

在init.js中搜索 **baseUrl:**

baseUrl:(location.protocol === 'file:' ? 'http:' : location.protocol) + '//' + "[HOSTNAME\_AND\_PATH\_TO\_JSAPI]js/dojo/dojo"

将localhost/webLab/layerOper/jsapi/js/dojo/dojo替换掉

 [HOSTNAME\_AND\_PATH\_TO\_JSAPI]js/dojo/dojo

保存init.js并退出。

 5.将Url片段替换dojo.js的baseUrl:

对dojo/dojo/dojo.js施加同样的操作。

6.发布成功后,可以浏览地图了

# 1. 如何部署本地sdk

 什么是sdk?

sdk=tutorials+API Referenc+Samples

如何部署sdk?

(1)下载sdk,解压得到目录arcgis\_js\_v37\_sdk

(2)将arcgis\_js\_v37\_sdk\ arcgis\_js\_api\sdk目录拷贝到tomcat Webapp目录下

(3)startUp tomcat, 访问localhost/sdk

接下来存在的问题：

拔掉网线，会发现reference页面，无法向下滚动. 解决方式：

(1)鼠标左下向下拉动

# 发布GIS应用常见问题

## Init.js 和 dojo.js编码

  默认使用的编码是 ANSI

  不要改变编码,否则 发布后 加载init.js出问题

## <script src=”Init.js”></script>提示ieDefineFailed

原因：页面中引用了purl.js和xdate.js，冲突

## 服务端自带的服务加载脚本

使用方法：

-1将两个css指向本地服务的css

2-将init.js指向本地服务的init.js

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<meta http-equiv="X-UA-Compatible" content="IE=7" />

<title>ArcGIS JavaScript API: TF/newmap</title>

<style type="text/css"> @import "http://serverapi.arcgisonline.com/jsapi/arcgis/2.6/js/dojo/dijit/themes/tundra/tundra.css";

@import "/arcgis/rest/static/jsapi.css";

</style> <script type="text/javascript"> var djConfig = { parseOnLoad: true };

</script> <script type="text/javascript" src="http://serverapi.arcgisonline.com/jsapi/arcgis/?v=2.6"> </script> <script type="text/javascript"> dojo.require("dijit.layout.BorderContainer");

dojo.require("dijit.layout.ContentPane");

dojo.require("esri.map");

var map;

function init() {

    map = new esri.Map("map");

          var layer = new esri.layers.ArcGISDynamicMapServiceLayer("http://10.254.53.85:6080/arcgis/rest/services/TF/newmap/MapServer");

          map.addLayer(layer);

    var resizeTimer;

     dojo.connect(map, 'onLoad', function(theMap) {

      dojo.connect(dijit.byId('map'), 'resize', function() {

       clearTimeout(resizeTimer);

       resizeTimer = setTimeout(function() {

       map.resize();

          map.reposition();

         }, 500);

       });

     });

    }

dojo.addOnLoad(init);

</script> </head>

<body class="tundra">

 <div dojotype="dijit.layout.BorderContainer" design="headline" gutters="true" style="width: 100%; height: 100%; margin: 0;">

 <div dojotype="dijit.layout.ContentPane" region="top" id="navtable">

<div style="float:left;" id="breadcrumbs">ArcGIS JavaScript API: TF/newmap</div>

<div style="float:right;" id="help">

Built using the <a href="http://help.arcgis.com/en/webapi/javascript/arcgis/">ArcGIS JavaScript API</a>

</div> </div> <div id="map" dojotype="dijit.layout.ContentPane" region="center"> </div> </div> </body>

</html>

# 2. 图层操作（layers）

Prerequirement:

必须添加的引用：

  <link rel="stylesheet" type="text/css" href="../gis/jsapi/3.4/js/dojo/dijit/themes/claro/claro.css"/><!--GISCSS-->

        <link rel="stylesheet" href="../gis/jsapi/3.4/js/esri/css/esri.css"/>

   <script type="text/javascript" src="../gis/jsapi/3.4/init.js" charset="utf-8"></script><!--GISAPI-->

        <script type="text/javascript">    var dojoConfig = { parseOnLoad: true };</script>

关键:4句:

dojo.require(“esri.map”);

var map=new esri.map(“map”);

var layer=new esri.layer.DynamicLayer(“REST Service URL”)

map.addlayer(layer)

详细：参见1.

## 图层对象

矢量图：ArcGISDynamicMapServiceLayer

影像图：ArcGISImageServiceLayer

GraphicsLayer

 ….

 加载矢量图(mxd)

关键:esri.layers.ArcGISDynamicMapServiceLayer

     map.addLayer

**var** districtLayer = **new** esri.layers.ArcGISDynamicMapServiceLayer(gSettings.serviceUrl.districtMapUrl, { id: "baseMapLayer" });

mapAppObj.map.addLayer(districtLayer);

**//url:**

[**http://10.254.53.85:6080/arcgis/rest/services/TF/baseMap/MapServer**](http://10.254.53.85:6080/arcgis/rest/services/TF/baseMap/MapServer)

## 加载影像图层

keyWord:esri.layers.ArcGISImageServiceLayer

**var** imageLayer = **new** esri.layers.ArcGISImageServiceLayer(gSettings.serviceUrl.imageServiceUrl, { id: "baseMapLayer" });

                   mapAppObj.map.addLayer(imageLayer);

备注：影像图层有两种方式

   （1）在arccatLog中，右键点击，share as imageService.如果这样，使用ArcGISImageServiceLayer加载影像

   （2）在arcMap中，share as Service.如果这样，使用ArcGISDynamicMapService加载。

## 加载GraphicsLayer

//创建graphicsLayer

**var**graphicLines=**new** esri.layers.GraphicsLayer();

var graphicLines=new esri.layers.GraphicsLayer({id:”graFun”});

//添加到地图

map.addLayer(graphicLines)

//查询line图层,获取feature

**var** queryTask = **new** esri.tasks.QueryTask(getQueryUrl(1));

**var** query = **new** esri.tasks.Query();

         query.outSpatialReference = gSettings.spatialReference;

         query.returnGeometry = **true**;

         query.outFields = ["\*"];

         query.where = "1=1";

**var** tempAr = [queryTask.execute(query)];

**var** deferList = **new** dojo.DeferredList(tempAr);

deferList.then(**function** (results) {

**if** (results[0][0] == **false** || results[0][1].features.length == 0) {

                console.log("获取线路图层失败,请检查GIS服务");

**return**;

             }

console.log("graphicLines get,count:"+results[0][1].features.length);

**var** lineFeatures = results[0][1].features;

  //从feature生成graphic

**for** (**var** i = 0; i <= lineFeatures.length - 1; i++) {

**var** lineName = lineFeatures[i].attributes.KEY\_NAME;

**var** lineWidth = 4;

var lineSymbol = **new** esri.symbol.SimpleLineSymbol().setColor(**new** dojo.Color([255,255,0])).setWidth(lineWidth);

 //生成lineGraphic并放入graphicsLayer

\_this.graphicLines.add(**new** esri.Graphic(lineFeatures[i].geometry, lineSymbol, lineFeatures[i].attributes, **null**));

}

## GraphicsLayer添Graphic(点、线、面)

前提:**从feature创建Graphic.**

**如果无feature**,请参考从坐标生成feature.

步骤:为feature设置symbol

关键句: var gra=new esri.Graphic(feature,symbol,attributes,infoTemplate);

       graphicsLayer.add(gra);

备注:-1.graphic的feature是传址引用。

-2.如果需要从坐标生成feature,参考feature操作。

#### 设置点symbol,生成 graphic

var graphicLayer=new esri.layers.GraphicsLayer();

//方法1

feature.setSymbol(**new** esri.symbol.PictureMarkerSymbol(“start.png”, 21, 21));

feature.setInfoTemplate("起点信息", "${NAME}");

graphicLayer.add(feature);

//方法2

var pointGra=new esri.Graphic(feature,symbol,attributes,infoTemplate);

graphicLayer.add(pointGra);

备注:创建graphic,feature是必须的.

必须有symbol才能显示.

attributes，可选。

infoTemplate,点击graphic弹出的信息窗体,可选.

#### 设置线symbol,生成 graphic

var graphicLayer=new esri.layers.GraphicsLayer();

//方法1

var lineSymbol = **new** esri.symbol.SimpleLineSymbol().setColor(**new** dojo.Color([255,255,0])).setWidth(4);

var lineGra=new esri.Graphic(linefeature,lineSymbol,null,null);

graphicLayer.add(lineGra);

//方法2

var lineSymbol = **new** esri.symbol.SimpleLineSymbol().setColor(**new** dojo.Color([255,255,0])).setWidth(4);

linefeature.setSymbol(lineSymbol);

graphicLayer.add(linefeature);

#### 设置面symbol,生成graphic

**var** ringPolygons=**new** esri.layers.GraphicsLayer();

//方法1：

polygonFeature.setSymbol(**new** esri.symbol.SimpleFillSymbol(esri.symbol.SimpleFillSymbol.STYLE\_SOLID, **new**esri.symbol.SimpleLineSymbol(esri.symbol.SimpleLineSymbol.STYLE\_DASHDOT, **new** dojo.Color([255, 0, 0]), 2), **new**dojo.Color(colors.Yellow)));

graphicsLayer.add(polygonFeature);

方法2：

var graPolygon=new esri.Graphic(polygonFeature,symbol,null,null)

graphicsLayer.add(graPolygon);

## 加载自定义切片图层

参考 reference, Layers->Creating a custom tiled layer type

举例：

## 删除图层

关键：map.removeLayer(map.getLayer(id));

**var** targetLayer=mapAppObj.map.getLayer("baseMapLayer");

mapAppObj.map.removeLayer(targetLayer);

方法2：

var graFun=new esri.layers.GraphicsLayer();

map.addLayer(graFun);

 map.removeLayer(graFun)

## 图层显/隐

关键句：map.getLayer(id).show();

        map.getLayer(id).hide();

 mapAppObj.map.getLayer("baseMapLayer").show()

mapAppObj.map.getLayer("baseMapLayer").hide()

## 设置图层可见scale

说明：设置图层可见比例尺。

参数：

minScale-最小比例尺-再远就看不见；minScale设置为0，则在无穷远也显示

  maxScale-最大比例尺-再近就看不见；maxScale设置为0，在无穷近也显示

关键句：

       Map.getLayer(“id”).setScaleRange(minScale,maxScale);

举例：

   Map.getLayer(“baseMapLayer”).setScaleRange(1155600,0)

   Map.getLayer(“baseMapLayer”).setScaleRange(800,0)

备注：

tileMapService中控制maxLevel和minLevel,

map加载 dynamicMapService, 那么dynamicMap的scale也会被限制.

## 设置图层透明度

试用图层:

 imageLayer,dynamicMapLayer,graphicsLayer,etc…

方法1:创建图层时设置

  var stationLayer = new esri.layers.ArcGISDynamicMapServiceLayer("http://10.254.53.85:6080/arcgis/rest/services/TF/newmap/MapServer", { "opacity": 0.5 });

map.addLayer(stationLayer);

方法2:

layer.setOpacity(0.5);

**应用**:

通过JqueryUI 滚动条控制透明度;

   $(".testScroll").slider(

         {

            max:10,//最右侧值

            value:10,//初始值

            orientation:"horizonal",//朝向

            slide: **function**(event, ui) {//滑动回调函数

**var** value=ui.value;

**var** type=event.target.getAttribute("data");

**var** percent=(value/$(event.target).slider("option","max")).toFixed(1);

**var** targetLayer=mapObj.getLayer(type);

**if**(targetLayer==**null**){**return**;}

            targetLayer.setOpacity(percent);

**return**;

         }

         });

## 图层顺序

#### dynamicLayer,imageLayer,tileLayer之间切换

引用:jquery-ui.js

前端:

<ul id="layerController" class="operUI">

<li class="layerItem" data="imageLayer">

   <div class="layerItemFirstRow"><div class="layerCb operUI layerOn" title="显隐图层" data="imageLayer"></div><div class="layerText" title="拖拽,改变图层顺序">影像图</div></div>

   <div class="testScroll" data="imageLayer" title="改变影像图透明度"></div>

</li>

<li class="layerItem" data="districtLayer">

<div class="layerCb operUI layerOn" data="districtLayer" title="显隐图层"></div><div class="layerText" title="拖拽,改变图层顺序">行政图</div>

      <div class="testScroll" data="districtLayer" title="改变行政图透明度"></div>

</li>

<li class="layerItem" data="titleLayer">

   <div class="layerCb operUI layerOn" data="tileLayer" title="显隐图层"></div><div class="layerText" title="拖拽,改变图层顺序">百度图</div>

   <div class="testScroll" data="titleLayer" title="改变百度图透明度"></div>

</li>

</ul>

js:

    //拖拽,改变图层顺序

   //拖拽,改变图层顺序

   $("#layerController").sortable({

      stop:**function**(event,ui){

         console.log("layerController.stop is triggered");

**var** layerSortArr=[];

**var** itemCount=$(".layerItem").length;

         $(".layerItem").each(**function**(i,item){

**var** tempObj={layer:mapObj.getLayer(item.getAttribute("data")),index:itemCount-1-i};

            layerSortArr.push(tempObj);

         });

         resortMapLayers(layerSortArr);//重排序图层

      }

   });

//重排序图层

//obj,Array[{layer:,index},{layer:,index}];

**function** resortMapLayers(obj){

   console.log("resortMapLayers is triggered");

**var** tempStr="[";

**for**(**var** i=0;i<=obj.length-1;i++){

**if**(obj[i].layer!=**null**){

         tempStr+="{id:"+obj[i].layer.id+",index:"+obj[i].index+"},";

         mapObj.reorderLayer(obj[i].layer,obj[i].index);//index越大,则优先级越高

      }

   }

   tempStr+="]";

   console.log(tempStr);

}

**常见问题**

添加图层的步骤？

1.添加li标签到ul中；

 2.将li下 所有data改成图层的id;

#### dynamicLayer内部layers切换

背景:dynamicMapService 包含两个图层station,line

   MapServer/0,station

   MapServer/1,line

显示效果:站在线上。

目标:jsAPI,让站在线下。

mapObj.getLayer("themeLayer").setDynamicLayerInfos

获取原始DynamicLayerInfos

var firstLayerInfo=mapObj.getLayer("themeLayer").createDynamicLayerInfosFromLayerInfos();

设置DynamicLayerInfos

setDynamicLayerInfos(dynamicLayerInfos, doNotRefresh)

//code snippets

var firstLayerInfo=mapObj.getLayer("themeLayer").createDynamicLayerInfosFromLayerInfos();

var newOrder=[firstLayerInfo[2],firstLayerInfo[1],firstLayerInfo[0]];

mapObj.getLayer("themeLayer").setDynamicLayerInfos(newOrder,false);

 变更DynamicLayerInfo.id

背景:.id决定在上还是在下,0在最上,越小越下

## 图层事件

#### graphicsLayer-onMouseOver

描述：mouseOver graphic时,显示属性信息

总结:map.infoWindow.show(map.toScreenPoint(evt.mapPoint));

**if**(equipMouseOverHandler==**null**){

        equipMouseOverHandler=dojo.connect(equipmentGraphics,"onMouseOver",**function**(evt){

            console.log("equipMouseOver triggered");

            evt.graphic.setSymbol(highlightSymbol);

            mapAppObj.map.infoWindow.setTitle("设备类型:"+evt.graphic.attributes.type);

            mapAppObj.map.infoWindow.setContent("设备名称:"+evt.graphic.attributes.NAME);

            //x方向偏小

            mapAppObj.map.infoWindow.show(mapAppObj.map.toScreen(evt.mapPoint));

         });

         console.log("equipMouseOver binded");

      }

#### graphicsLayer点击,显示属性信息框

说明:graphicsLayer={graphic};

esri.Graphic对象

       graphic.setInfoTemplate(infoTemplate).点击时会则弹出属性窗口。

      无需绑定graphic.onclick.

      var infoTemplate = new esri.InfoTemplate("Attributes", "<tr>State Name: <td>${STATE\_NAME}

  </tr></td><br><tr>Population:<td>${Pop2001}</tr></td>");

graphic.setInfoTemplate(infoTemplate);

## 其它细节

#### map.graphics 和GraphicsLayer

**map.graphics:**

map.graphcis对象：map对象默认的一层graphicsLayer.

//map.graphics=GraphicsLayer of a map;

//map.graphics.graphics  = all the graphics of map.graphics

**graphicsLayer:**

esri.layers.GraphicsLayer

={graA,graB,….};

备注:graEle的geometry type可以是任意的。即,允许同时存放lineGra,pointGra,

sectionGra

# 3.要素操作(geometry)

## geometry的类型

 String,type.

The type of geometry.

**Known values:** point | multipoint | polyline | polygon | extent

## point操作：

#### 坐标生成point

**var** point=**new** esri.geometry.Point(x,y,{wkid:102113});

 var point= new esri.geometry.Point(-118.15, 33.80, newesri.SpatialReference({ wkid: 4326 }));

#### 复制point feature

**this**.getCopyOfTransfers=**function**(){

**var** results=[];

**for**(**var** i=0;i<=pointFeatures.length-1;i++){

**var** geoJson= pointFeatures [i].geometry.toJson();

**var** attrJson=JSON.stringify(pointFeatures [i].attributes);

**var** newAttr=JSON.parse(attrJson);

            newAttr.odType="w";

**var** newTransfer=**new** esri.Graphic(**new** esri.geometry.Point(geoJson),**null**,newAttr,**null**);

            results.push(newTransfer);

         }

**return** results;

   };

## Point和MultiPoint

通过geometry引用第一个点

  Point：geometry.point

  Multipoint: geometry.points[0]

通过geometry获取第一个点的坐标

geometry.point.x, geometry.point.y

geometry.points[0][0], geometry.points[0][1]

## polyline操作

#### 坐标生成polyline

**var** newLine=**new** esri.geometry.Polyline(lineJSON);

#### 复制polyline

**this**.getCopyOfLines=**function**(){

**var** results=[];

**for**(**var** i=0;i<=lineFeatures.length-1;i++){

**var** geoJson=lineFeatures[i].geometry.toJson();

**var** attrJson=JSON.stringify(lineFeatures[i].attributes);

**var** newAttr=JSON.parse(attrJson);

**var** newLine=**new** esri.geometry.Polyline(geoJson);

**var** tempGra=**new** esri.Graphic(newLine,**null**,newAttr,**null**);

         results.push(tempGra);

      }

**return** results;

   };

#### polyline抗锯齿

方案1：发布service时,使用antiAliasing

方案2：(1)随意发布

       (2)query.where=”1=1”;queryTask(url=”line.url”);

          获取到lineFeatures,使用GraphicsLayer绘制线图层

效果对比:

antiAliasing

 graphicsLayer重新绘制

结论:

使用GraphicsLayer绘制线图层效果比 antiAliasing 好。

 代价是，写queryTask,setSymbol…..

## polygon操作

#### 坐标生成polygon

/\*描述：给定圆心,半径,生成缓冲圆

     \* pt-点,radius 米

\*/

**var** createCircleGeometry=**function**(pt,radius){

**var** polygon = **new** esri.geometry.Polygon();

**var** points = [];

**for** (**var** i = 0; i <= 360; i += 10) {

**var** radian = i \* (Math.PI / 180.0);

**var** x = pt.x + radius \* Math.cos(radian);//r \*[-1.1]

**var** y = pt.y + radius \* Math.sin(radian);

            points.push(**new** esri.geometry.Point(x, y));

        }

        polygon.addRing(points);

        polygon.spatialReference = mapAppObj.map.spatialReference;

**return** polygon;

    };

#### 复制polygon

var geoJson=polygon.geometry.toJson();

**var** polygon = **new** esri.geometry.Polygon(geoJson);

#### geometry.Polygon contains

Sentence:

polygonGeometry.contains(pointGeometry)

Description: check whether a point is inside or outside polygon

Code Snippets

     var feature = dojo.byId("imgSelectDistrict").district;//feature.geometry.type="polygon"

                    for (var i = graLayerSelectO.graphics.length - 1; i >= 0; i--) {

                        if (feature.geometry.contains(graLayerSelectO.graphics[i].geometry)) {

                            graLayerSelectO.remove(graLayerSelectO.graphics[i]);

                        }

                    }

## extent操作

#### 坐标生成extent

function createExtentGeometry(point,delta) {

    var xmin = point.x - delta;

    var xmax = point.x + delta;

    var ymin = point.y - delta;

    var ymax = point.y + delta;

    var extent = new esri.geometry.Extent(xmin, ymin, xmax, ymax, new esri.SpatialReference({ wkid: 4326 }));

    return extent;

}

#### 复制extent

var geoJson=extent.geometry.toJson();

var newExtent=new esri.geometry.Extent(geoJson);

#### extent扩大缩小

map.setExtent(map.extent.expand(0.5));

map.setExtent(map.extent.expand(2));

## 给定半径,生成圆

function createCircleGeometry(pt, radius) {

    var polygon = new esri.geometry.Polygon();

    var points = [];

    for (var i = 0; i <= 360; i += 10) {

        var radian = i \* (Math.PI / 180.0);

        var x = pt.x + radius \* Math.cos(radian);//r \*[-1.1]

        var y = pt.y + radius \* Math.sin(radian);

        points.push(new esri.geometry.Point(x, y));

    }

    polygon.addRing(points);

    polygon.spatialReference = map.spatialReference;

    return polygon;

}

# 4 Graphic系列

## 构造函数

       var graphic=new esri.Graphic(geometry,symbol,attributes,infotemplate);

Note: Graphic and feature are equivalent

[new esri.Graphic(geometry, symbol, attributes, infoTemplate)](file:///F:\DevNoteTut\ArcGIS%20Server%2010.1%20SP1_%C3%A8%C2%AF%C2%B4%C3%A6%C2%98%C2%8E_%C3%A6%C2%95%C2%99%C3%A7%C2%A8%C2%8B\JSAPI%C3%A5%C2%BC%C2%80%C3%A5%C2%8F%C2%91%C3%A7%C2%AC%C2%94%C3%A8%C2%AE%C2%B0\JavascriptAPI_Reference\v33_Reference\arcgis\graphic.html#GraphicConst1)

Note:

1.graphic is used to store queried feature,

       var graphic=new esri.Graphic(feature,null,null,null);

       then graphic stored the feature Object. for further deal;

2.both **feature**and **symbol** are required for showing a graphic on map.

## click事件

举例：点击graphic,然后从graphicsLayer中删除

dojo.connect(map.graphics, "onClick", function(e){

       var graphic=evt.graphic;

       map.graphics.remove(graphic);

});

## Graphic.attriutes

#### 复制

var attrJson=JSON.stringify(graphic.attributes);

var newAttrObj=JSON.parse(attrJSON);

#### setAttributes

方法1：

var JsonObj={

       Name:”西直门”,

       id:”01”

}

graphic.setAttributes(JsonObj);

 方法2:

graphic.atrributes.Name=”西直门”;

graphic.attributes.id=”01”;

#### 设置

graphic.atrributes.Name=”西直门”;

graphic.attributes.id=”01”;

## onMouseOver/onMouseOut事件

举例：鼠标悬浮在pointGraphic上,graphic图片放大。

              鼠标移开，图片变小。

  dojo.connect(map.graphics, "onMouseOver", function(e){

              var graphic=e.graphic;

              var bigSymbol=new esri.symbol.PictureMarkerSymbol(imgUrl,25,20)

              graphic.setSymbol(bigSymbol);

});

  dojo.connect(map.graphics, "onMouseOut", function(e){

              var graphic=e.graphic;

              var smallSymbol=new esri.symbol.PictureMarkerSymbol(imgUrl,20,20)

              graphic.setSymbol(smallSymbol);

});

## Graphic部件 infoTemplate

说明：graphic的弹出窗体 通过 infoTemplate控制。

       graphic.setInfoTemplate(infoTemplate);

#### 构造函数:

var  infoTemplate = new esri.InfoTemplate("Attributes", "<tr>State Name: <td>${STATE\_NAME}

  </tr></td><br><tr>Population:<td>${Pop2001}</tr></td>");

#### 举例,infoTemplate插入链接

  var infoTemplate = new esri.InfoTemplate("${title}", "<tr>途经换乘:<a href='http://www.baidu.com'>http://www.baidu.com</a><td>${routeName}</tr></td><br><tr>点到点OD总量:<td>${ODTotal}</tr></td><br><tr>本方案OD量:<td>${ODValue}</tr></td><br><tr>占比:<td>${ODPartition}</tr></td>");

                    middlefeature.setInfoTemplate(infoTemplate);

 显示所有属性的infoTemplate

var template = new esri.InfoTemplate("${name}", "${\*}");

#### 带按钮,input录入框的popWindow[动态Popwindow]

路子：

function createPopUpWindow(){

       var operPane=document.createElement(“div”);

       operPane.innerHTML=htmlString;

       operPane.onclick=function(e){

       var operId=e.target.id;

       if(operId==””){

}

else if(operId==””){

}

…

return operPane;

}

 var infoContent =getODWPopup(tempGra);

 mapAppObj.map.infoWindow.setTitle("站点信息");

mapAppObj.map.infoWindow.setContent(infoContent);

mapAppObj.map.infoWindow.resize(300, 260);

mapAppObj.map.infoWindow.show(tempGra.geometry, mapAppObj.map.getInfoWindowAnchor(tempGra.geometry));

## Graphic 部件 Symbol

说明：graphic必须设置symbol，才能显示

symbol种类

#### 线symbol

 构造函数：esri.symbolSimpleLineSymbol(线型,颜色,宽度)

举例：

var sls = new esri.symbol.SimpleLineSymbol(esri.symbol.SimpleLineSymbol.STYLE\_DASH,

  new dojo.Color([255,0,0]), 3);

#### 填充 simpleFillSymbol

var sfs = new esri.symbol.SimpleFillSymbol(esri.symbol.SimpleFillSymbol.STYLE\_SOLID,

  new esri.symbol.SimpleLineSymbol(esri.symbol.SimpleLineSymbol.STYLE\_DASHDOT,

  new dojo.Color([255,0,0]), 2),new dojo.Color([255,255,0,0.25]));

#### 点symbol

参见 simpleFillSymbol

#### 图片 symbol

new esri.symbol.PictureMarkerSymbol('http://www.esri.com/graphics/aexicon.jpg', 51, 51);

#### Symbol-分类渲染(ClassBreaksRender)

Desciption：render by a the value range  of specific field

Steps:

  à1 set field for ClassBreaksRenderer

  à2 addBreaks for each value Range, set color ,size for them

Code Snippet:

var observationRenderer = new esri.renderer.ClassBreaksRenderer(new esri.symbol.SimpleMarkerSymbol(), "CheckOut"); //(fieldName)

                            //(symbol for each value range)

                            observationRenderer.addBreak(700, 100000000, new esri.symbol.SimpleMarkerSymbol(esri.symbol.SimpleMarkerSymbol.STYLE\_CIRCLE, dojox.gfx.px2pt(30), new esri.symbol.SimpleLineSymbol().setStyle(esri.symbol.SimpleLineSymbol.STYLE\_SOLID).setColor(new dojo.Color([100, 100, 100])), new dojo.Color([255, 0,0,255])));

                            observationRenderer.addBreak(600, 700, new esri.symbol.SimpleMarkerSymbol(esri.symbol.SimpleMarkerSymbol.STYLE\_CIRCLE, dojox.gfx.px2pt(25), new esri.symbol.SimpleLineSymbol().setStyle(esri.symbol.SimpleLineSymbol.STYLE\_SOLID).setColor(new dojo.Color([100, 100, 100])), new dojo.Color([255, 0, 0, 255])));

                            observationRenderer.addBreak(500, 600, new esri.symbol.SimpleMarkerSymbol(esri.symbol.SimpleMarkerSymbol.STYLE\_CIRCLE, dojox.gfx.px2pt(20), new esri.symbol.SimpleLineSymbol().setStyle(esri.symbol.SimpleLineSymbol.STYLE\_SOLID).setColor(new dojo.Color([100, 100, 100])), new dojo.Color([255, 0, 0, 200])));

                            observationRenderer.addBreak(400, 500, new esri.symbol.SimpleMarkerSymbol(esri.symbol.SimpleMarkerSymbol.STYLE\_CIRCLE, dojox.gfx.px2pt(15), new esri.symbol.SimpleLineSymbol().setStyle(esri.symbol.SimpleLineSymbol.STYLE\_SOLID).setColor(new dojo.Color([100, 100, 100])), new dojo.Color([255, 0, 0, 150])));

                            observationRenderer.addBreak(300, 400, new esri.symbol.SimpleMarkerSymbol(esri.symbol.SimpleMarkerSymbol.STYLE\_CIRCLE, dojox.gfx.px2pt(10), new esri.symbol.SimpleLineSymbol().setStyle(esri.symbol.SimpleLineSymbol.STYLE\_SOLID).setColor(new dojo.Color([100, 100, 100])), new dojo.Color([255, 0, 0, 100])));

                            observationRenderer.addBreak(0, 300, new esri.symbol.SimpleMarkerSymbol(esri.symbol.SimpleMarkerSymbol.STYLE\_CIRCLE, dojox.gfx.px2pt(5), new esri.symbol.SimpleLineSymbol().setStyle(esri.symbol.SimpleLineSymbol.STYLE\_SOLID).setColor(new dojo.Color([100, 100, 100])), new dojo.Color([255, 0, 0, 50])));

                            graLayerSelect.setRenderer(observationRenderer); //(render)

详细：参考Reference

## Graphic 编辑控件

[file:///F:/DevelopSoftware/ArcGIS%20Server%2010.1%20SP1\_%E8%AF%B4%E6%98%8E\_%E6%95%99%E7%A8%8B/JSAPI%E5%BC%80%E5%8F%91%E7%AC%94%E8%AE%B0/JavascriptAPI\_Reference/v33\_Samples/arcgis/toolbar\_edit.html](file:///F:\DevelopSoftware\ArcGIS%20Server%2010.1%20SP1_%C3%A8%C2%AF%C2%B4%C3%A6%C2%98%C2%8E_%C3%A6%C2%95%C2%99%C3%A7%C2%A8%C2%8B\JSAPI%C3%A5%C2%BC%C2%80%C3%A5%C2%8F%C2%91%C3%A7%C2%AC%C2%94%C3%A8%C2%AE%C2%B0\JavascriptAPI_Reference\v33_Samples\arcgis\toolbar_edit.html)

# 5 feature系列

 feature=geometry+attributes+infoTemplate+symbol

feature is **equivalent** to graphic.

 geometry存矢量信息

 attributes存属性信息

 infoTemplate 弹出窗体

 symbol  feature的符号

## feature设置属性和读取属性

à1设置属性

   à(1) setAttributes(Json)

  feature.setAttributes(fieldValuePair);

       à(2)

## feature.setInfoTemplate

//弹出窗体的 标题和内容

  middlefeature.setInfoTemplate(new esri.InfoTemplate(this.title, "<tr>途经换乘:<td>${routeName}</tr></td><br><tr>点到点OD总

量:<td>${ODTotal}</tr></td><br><tr>本方案OD量:<td>${ODValue}</tr></td><br><tr>占比:<td>${ODPartition}</tr></td>"));

## feature添加属性

  à1stMethod

   var attrObj = feature.attributes;

    attrObj.routeID = i.toString();

    attrObj.routeName = routesArray[i - 2].routeName;

    ….

    feature.setAttributes(attrObj);

 à2ndMethond

this.feature.attributes.ODTYPE = "o";

## show features on Map

Summary:

   Feature.setAttribute(json)

   Feature.setSymbol()

   Feature.setInfoTemplate(“${name}”,”${\*}”);

 Example

  dojo.connect(geocoder, "onFindResults", function(response) {

          console.log("find results: ", response);

          var l = map.getLayer("results");

          l.clear();

          map.infoWindow.hide();

          dojo.forEach(response.results, function(r) {

            r.feature.attributes.name = r.name;

            r.feature.setSymbol(symbol);

            r.feature.setInfoTemplate(template);

            l.add(r.feature);

          });

## 拷贝feature

Feature=geometry+attributes

Geometry拷贝：var geoJson=geometry.toJson();

                        Var newgeo =new esri.geometry(geoJson);

Attributes.拷贝：

                            Var attrStr=JSON.stringify(feature.attributes);

                            Var newAttr=JSON.parse(attrStr);

返回feature:

             Var gra=new esri.Graphic(geometry,null,attributes,null)

**var** pointJSON=stationFeatures[i].geometry.toJson();

**var** newPt=**new** esri.geometry.Point(pointJSON);

**var** attrJSON=JSON.stringify(stationFeatures[i].attributes);

**var** newAttr=JSON.parse(attrJSON);

            newAttr.routeId=routeId;

**var** tempGra=**new** esri.Graphic(newPt,**null**,newAttr,**null**);

**if**(newPt!=stationFeatures[i].geometry){

                console.log("new point created");

            }

**else**{

                console.log("fail to create newPoint");

            }

**return** tempGra;

# 6. Map操作

## map组件

说明:new esri.Map(div,{options}); map组件指options

   mapObj=**new** esri.Map("map",{

      nav:**true**,//8个pan 箭头

      slider:**false**,//左上的缩放 +/-;

      logo:**false**,//右下的esri logo

      showAttribution:**false**,//右下的gisNeu (logo左侧)

      extent://地图的extent

      });

控制ZoomSlider的方法

(1)map.showZoomSlider();

   (2)map.hideZoomSlider();

控制nav的方法

  (1)map.showPanArrows();

  (2)map.hidePanArrows();

执行nav8个方向移动的方法

  .panUpperLeft()

  .panUp();

  .panUpperRight()

  .panLeft()

  .panRight()

  .panLowerLeft();

  .panDown()

  .panLowerRight();

## dijit Map组件

#### scalebar

背景:scalebar控制单位

var scalebar=new esri.dijit.Scalebar({map:map,scalebarUnit:”english/metric”});

## map尺寸改变

背景：客户要求地图大小有两种模式

--模式1：窗体模式,450\*450

--模式2：全屏模式:document.documentElement.clientHeight\*clientWidth

**路子**

**在mapDiv外层放置mapPanel,**

**mapDiv的尺寸设置 100%，100%；**

**通过改变mapPanel直接调整mapDiv的尺寸。不必直接操作mapDiv**

**html布局**

 <div id=”mapPanel” class=”part”>

   <div id=”mapDiv” style=”height:100%;width:100%”>

<div class=”operUI partStatus” id=”mapSizeBtn”></div>

</div>

</div>

**css:**

.part:{

       height:450px;

   width:450px;

}

**js:**

   $(".operUI").on("click",**function**(e){

**var** targetId=e.target.id;

**if**(targetId=="mapSizeBtn"){//change map size

         if($(“#mapPanel”).hasClass(“part”)){

   $(“#mapPanel”).removeClass(“part”);

   $(“#mapPanel”).css(“width”,document.documentElement.clientWidth+”px”);

  $(“#mapPanel”).css(“height”,document.documentElement.clientHeight+”px”);

}

else{

$(“#mapPanel”).addClass(“part”);

}

}

**return** **false**;

   });

## map 导航 （pan/zoom/preview/fullExtent）

#### 1.     使用esri.tools.navigation实现

#### 2.使用代码实现

$(".toolNavigateItem").on("click",**function**(){//导航项

**if**(**this**.id!="pan"&&**this**.id!="zoomin"&&**this**.id!="zoomout"&&**this**.id!="fullextent"){

**return** **false**;

        }

**if**(**this**.id=="pan"){

         mapAppObj.drawHandler.disconnectPrevious();

         removeSelectTypeDown();

        }

**else** **if**(**this**.id=="zoomin"){//放

**var** extent=mapAppObj.map.extent;

           mapAppObj.map.setExtent(extent.expand(0.5));

        }

**else** **if**(**this**.id=="zoomout"){//缩

**var** extent=mapAppObj.map.extent;

           mapAppObj.map.setExtent(extent.expand(2));

        }

**else**{//全

           mapAppObj.map.setExtent(**new** esri.geometry.Extent({ "xmin": gSettings.originExtent.xMin, "ymin": gSettings.originExtent.yMin, "xmax": gSettings.originExtent.xMax, "ymax": gSettings.originExtent.yMax, "spatialReference":gSettings.spatialReference}));

        }

     });

## map坐标和Screen坐标

#### mapPoint转screenPoint

var sPt=map.toScreen(mapPoint)

#### screenPoint转mapPoint

var mPt=map.toMap(screenPoint);

## map.setExtent

 关键:

       var extent=new esri.geometry.extent(xmin,ymin,xmax,ymax,srObj);

    map.setExtent(extent);

举例:

mapAppObj.map.setExtent(**new** esri.geometry.Extent({ "xmin": gSettings.originExtent.xMin, "ymin": gSettings.originExtent.yMin, "xmax": gSettings.originExtent.xMax, "ymax": gSettings.originExtent.yMax, "spatialReference":gSettings.spatialReference}));

        }

     });

举例2：

map.setExtent(map.extent.expand(0.5));

map.setExtent(map.extent.expand(2));

## map.infoWindow

description: map.infoWindow is show infoWindow on map.

#### 尺寸

   map.infoWindow.resize(800,600);

   map.infoWindow.show(pt, map.getInfoWindowAnchor(pt));

 显示

方法1：

var pt = new esri.geometry.Point(middlefeature.geometry.paths[0][tempLoc], map.spatialReference);

map.infoWindow.show(pt, map.getInfoWindowAnchor(pt));//手动控制

方法2:

   mapAppObj.map.infoWindow.show(mapAppObj.map.toScreen(evt.mapPoint));

#### setFeatures/clearFeatures

 setFeatures([graphics]);

description:which graphics attributes are showed in infoWindow.(one infoWindow will show all features, pager show one feature at once)

 clearFeatures();

map.infoWindow doesn’t show any features

## map.getLayer(id)

// get layer by id

var l = map.getLayer("results");

 map.addLayer

备注：id用于在 getLayer(id)获取图层

//add layer to map

 map.addLayer(new esri.layers.GraphicsLayer({

          id: "results"

 }));

## map.removeLayer

map.removeLayer(map.getLayer("layer0"));

## map 取中心坐标

**var** centerPt=**new** esri.geometry.Point((mapAppObj.map.extent.xmin+mapAppObj.map.extent.xmax)/2,(mapAppObj.map.extent.ymin+mapAppObj.map.extent.ymax)/2,mapSettings.spatialReference);

## map设置中心 和 extent

Description:set the Extent of Map

**对于dynamicMapLayer**

dojo.require("esri.geometry");

  var initialExtent = new esri.geometry.Extent({ "xmin": 116.0391859574468, "ymin": 39.69010477397335, "xmax": 116.8337859574469, "ymax": 40.19729626333512, "spatialReference": { "wkid": 4326} });

   map.setExtent(mapExtent);

description: set Map Center

map.centerAt(feature.geometry.points[0]);

**对于tileMapLayer**

mapAppObj.map.setZoom(11);//设置级别

                     var centerPt=new esri.geometry.Point(mapSettings.centerPt.x-30000,mapSettings.centerPt.y+6000,mapSettings.spatialReference);

                     mapAppObj.map.centerAt(centerPt);

## map事件 onload

Fires when the first or base layer has been successfully added to the map.

Code snippet:

map = new esri.Map("map");

    var baselayer = new esri.layers.ArcGISDynamicMapServiceLayer(baseMapUrl); //districtMap

    map.addLayer(baselayer);

    dojo.connect(map, "onLoad", function (results) {

});

## 查看map上所有layers

Properties are following:

* **Map Layers** - The code loops through each layer in the map and reports the layer ID, visibility, and opacity.
* **Spatial Reference** - Reports the well-known ID (WKID) of the map's spatial reference.
* **Scales** - Reports information about each scale level in the portlandParcels tiled map service layer. The function uses the ArcGISTiledMapServiceLayer.tileInfo property, which can show you how many scale levels are in the map cache, what scales they are, and what the resolution is at each scale.
* **Map width and height** - Reports the map width and height in pixels.

Code snippets:

----------------------------------

Layers:

          var layerInfo = [];

          dojo.forEach(map.layerIds,function(id){

            var layer = map.getLayer(id);

            layerInfo.push("id: " + layer.id + " visible: " + layer.visible + " opacity: " + layer.opacity + "<br />");

          });

--------------------------

Scales:  basemap.tileInfo

basemap = new esri.layers.ArcGISTiledMapServiceLayer("http://sampleserver1.arcgisonline.com/ArcGIS/rest/services/TaxParcel/AssessorsBasemap/MapServer");

dojo.forEach(basemap.tileInfo.lods,function(lod){

            var level = lod.level;

            var scale = lod.scale;

            resolution = lod.resolution;

            scales.push( "level: " + level + " scale: " + scale + " resolution: " + resolution + "<br />");

});

------------------------

Width,height,and spatialReference:

  map.spatialReference.wkid

  map.width

map.height

## 鼠标移动 显示坐标

Code snippets:

  dojo.connect(map, "onMouseMove", showCoordinates);

   function showCoordinates(evt) {

        //get mapPoint from event

        //The map is in web mercator - modify the map point to display the results in geographic

        var mp = esri.geometry.webMercatorToGeographic(evt.mapPoint);

        //display mouse coordinates

        dojo.byId("info").innerHTML = mp.x.toFixed(3) + ", " + mp.y.toFixed(3);

      }

## 鼠标事件 down/up/drag

支持：mouseDown,mouseUp

不支持：mouseMove,mouseDrag

//surpport in 3.4

      dojo.connect(map,"onMouseDown",**function**(evt){

         console.log("mouseDown triggered");

      });

      //sur int 3.4

      dojo.connect(map,"onMouseUp",**function**(evt){

         console.log("mouseUp triggered");

      });

      //not sur in 3.4

      dojo.connect(map,"onMouseMove",**function**(evt){

         console.log("mouseMove triggered");

      });

      dojo.connect(map,"onMouseDragStart",**function**(evt){

          console.log("dragStart trigered");

         startPt=evt;

          console.log("drawStart triggered,start point:"+evt.mapPoint.x+","+evt.mapPoint.y+".screenPoint:"+evt.clientX+","+evt.clientY);

       });

      dojo.connect(map,"onMouseDrag",**function**(evt){

         console.log("mouseDrag triggered");

      });

      dojo.connect(map,"onMouseDragEnd",**function**(evt){

         console.log("mouseDragEnd triggered");

      });

## map. setMapCursor 设置地图指针

map.setMapCursor("help") //

map.setMapCursor("default")//

map.setMapCursor("pointer")//

map.setMapCursor("wait")//

map.setMapCursor("progress")//

map.setMapCursor("cell")// 粗十字

map.setMapCursor("crosshair")// 细十字

map.setMapCursor("text")//文本I

map.setMapCursor("vertical-text")//放倒的I

//自定义图标

map.setMapCursor("url(../src/assets/images/cursors/4WAY01.CUR),auto")

//圆选图标

map.setMapCursor("url(../src/assets/images/cursors/ellipse-pro.cur),auto")

//矩形图标

map.setMapCursor("url(../src/assets/images/cursors/move-pro.cur),auto")

# 7.绘制要素操作

## 基础类

**画完军规**

军规：activateDrawTool,必须启动deactivate使鼠标恢复正常

//鼠标画矩形

drawTool.activate(esri.toolbars.Draw.POINT);

//鼠标终止矩形

drawTool.deactivate

 点

drawTool.activate(esri.toolbars.Draw.POINT);

## 线

drawTool.activate(esri.toolbars.Draw.POLYLINE);

## 矩形

drawTool.activate(esri.toolbars.Draw.EXTENT);

## 圆

drawTool.activate(esri.toolbars.Draw.CIRCLE);

# 8.查询

## 几何查询

引用:dojo.DefferedList

举例：

       var queryTask=new esri.tasks.QueryTask(url);

       var query = new esri.tasks.Query();

       query.outSpatialReference=map.spatialReference;

       query.geometry=geom;

        if (layerID == "0") {//点 查询

            query.spatialRelationship = esri.tasks.Query.SPATIAL\_REL\_CONTAINS;

         }

        else {//框选、圆选相交

            query.spatialRelationship = esri.tasks.Query.SPATIAL\_REL\_INTERSECTS;

        }

       query.outFields=”\*”;

        query.returnGeometry = true;

       var tempAr=new Array();

        tempAr.push(queryTask.execute(query));

var deferList=new Dojo.DeferredList(tempAr);

       deferList.then(function(results){

});

备注: Query.SPATIAL\_REL\_CONTAINS参见reference;

## 属性查询

引用:dojo.DefferedList

    var queryTask=new esri.tasks.QueryTask(url);

       var query = new esri.tasks.Query();

       query.outSpatialReference=map.spatialReference;

              query.Where=”1=1”

query.outFields = ["\*"];  //查询所有字段

        query.returnGeometry = true;

       var tempAr=new Array();

        tempAr.push(queryTask.execute(query));

var deferList=new Dojo.DeferredList(tempAr);

       deferList.then(function(results){

 });

备注：

query.where, 可先在arcgis/manager中写好。再copy到此处

**等于/or/and**

à字段等于

àor

àand

## deferList一次提交多个query

  var whereClause1 = “…”;

  var whereClause2 =”..”;

   var queryTask = new esri.tasks.QueryTask(themeMapUrl + "/2"); //查询section

   var query= new esri.tasks.Query();

    query1.where = whereClause1;

    query1.outFields = ["\*"];

    query1.outSpatialReference = map.spatialReference;

    query1.returnGeometry = true;

       var tempAr=new Array();

       tempAr.push(queryTask.execute(query1));

    query1.where = whereClause2;

       tempAr.push(queryTask.execute(query1));

    var deferList = new dojo.DeferredList(tempAr);

    deferList.then(function (results) {

      …

    });    //回调函数

# 9常用功能

## 距离面积量算

备注:

以下document mode **能够触发** mouseDrag事件()

<meta http-equiv="X-UA-Compatible" content="IE=9"/>

<meta http-equiv="X-UA-Compatible" content="IE=8"/>

<meta http-equiv="X-UA-Compatible" content="IE=7"/>

以下document mode 不能触发 mouseDrag事件()

<meta http-equiv="X-UA-Compatible" content="IE=10"/>

//计算,显示圆的半径 (圆选)

**function** shapeCalculator(){

**this**.downPt=**null**;//drag down pt

**this**.onMapDragStart=**null**;

**this**.onMapDragging=**null**;

**this**.onMapDragEnd=**null**;

**this**.measureDiv=**null**;//result div

**this**.targetShape=-1;//-1 null;0 rectangle;1 circle

**var** \_this=**this**;

   //初始化div,绑定

**this**.init=**function**(div){

       console.log("shapeCalculator.init() called");

**if**(div==**null**){console.log("unexpected para,function init");}

**this**.measureDiv=div;

    };

   //东西向偏移

**var** getDeltaX=**function**(ptB){

**if**(\_this.downPt==**null**||\_this.downPt.mapPoint==**null**){

         console.log("unexpected para,function getDeltaX()");

**return** "";

       }

**return** ptB.x-\_this.downPt.mapPoint.x;

    };

    //南北向偏移

**var** getDeltaY=**function**(ptB){

**if**(\_this.downPt==**null**||\_this.downPt.mapPoint==**null**){

         console.log("unexpected para,function getDeltaY()");

**return** "";

       }

**return** ptB.y-\_this.downPt.mapPoint.y;

    };

   //两点距(米)

**var** getDistance=**function**(ptB){

**if**(\_this.downPt==**null**||\_this.downPt.mapPoint==**null**){

         console.log("unexpected para,function getDistance()");

**return** "";

     }

**var** ptA=\_this.downPt.mapPoint;

**return** Math.sqrt(Math.pow(ptA.x-ptB.x,2)+Math.pow(ptA.y-ptB.y,2));

    };

    //A圆心,B圆周点.return 面积(平方米)

**var** getArea=**function**(ptB){

**var** r=getDistance(ptB);

**if**(r==**null**){

          console.log("unexpected para,function getArea()");

**return** "";

       }

**return** Math.PI\*r\*r;

    };

    //长度格式化

**var** formatLength=**function**(len){

**if**(len<1000){

**return** (Math.ceil(len)+"米");

      }

**else** **if**(len>=1000&&len<1000000){

         len=(Math.ceil(len)/1000).toFixed(2);

**return** (len+"千米;");

      }

**else**{//超过1000千米

         len=(Math.ceil(len)/1000).toFixed(2);

**return** (len+"千米;");

      }

    };

    //面积格式化

**var** formatArea=**function**(tempArea){

       //面积

**if**(tempArea<1000000){

**return** Math.ceil(tempArea)+"平方米";

      }

      //1平方公里-1万平方公里

**else** **if**(tempArea>=1000000&&tempArea<10000000000){

**return** ((Math.ceil(tempArea))/1000000).toFixed(2)+"平方公里";

      }

      //1万平方公里

**else**{

**return** ((Math.ceil(tempArea))/10000000000).toFixed(2)+"万平方公里";

      }

    };

    //show radius

**this**.showRadius=**function**(){

**if**(**this**.measureDiv==**null**){

          console.log("no measureDiv,can't show radius");

**return**;

       }

       console.log("showRadius() called");

       $(**this**.measureDiv).removeClass("dNone");

    };

    //clear radiusText,hide

**this**.clearRadius=**function**(){

       console.log("shapeCalculator.clearRadius() called");

       $(**this**.measureDiv).html("");

       $(**this**.measureDiv).addClass("dNone");

    };

    //绑定mapMouseDrag事件

**this**.bindDrags=**function**(type){

**if**(type!=0&&type!=1){

          console.log("unexpected para,function bindDrags()");

**return**;

       }

**if**(**this**.measureDiv==**null**){

          console.log("bindDrags(),failed. cause:measureDiv is null");

**return**;

       }

**this**.targetShape=type;//rectangle or circle to be measured

       console.log("radiusObj.targetShape is:"+type);

**if**(**this**.onMapDragStart==**null**){

          console.log("dragStart binded");

**this**.onMapDragStart=dojo.connect(mapAppObj.map,"onMouseDragStart",**function**(evt){

             console.log("dragStart triggered");

             mapAppObj.regionObj.hide();//隐藏区域

             mapAppObj.ringObj.hide();

             \_this.downPt=evt;

             \_this.showRadius();

          });

       }

**if**(**this**.onMapDragging==**null**){

       console.log("dragging binded");

**this**.onMapDragging=dojo.connect(mapAppObj.map,"onMouseDrag",**function**(evt){

         //console.log("dragging triggered");

**if**(\_this.targetShape!=0&&\_this.targetShape!=1){

            console.log("unexpected para,function onMapDragging");

**return**;

         }

**var** tempRad;

**var** tempArea;

**var** tempText="";

**if**(\_this.targetShape==1){//圆形

         tempRad=getDistance(evt.mapPoint);

         tempArea=getArea(evt.mapPoint);

         tempText+="半径:"+formatLength(tempRad);

         tempText+="  面积:"+formatArea(tempArea);

      }

**else**{//矩形

**var** tempWE=getDeltaX(evt.mapPoint);

**var** tempNS=getDeltaY(evt.mapPoint);

**var** weFlag=tempWE>0?"向东:":"向西:";

**var** nsFlag=tempNS>0?"  向北:":"  向南:";

      tempText+=weFlag+formatLength(Math.abs(tempWE))+nsFlag+formatLength(Math.abs(tempNS));

         tempText+="  面积:"+formatArea(Math.abs(tempWE\*tempNS));

      }

         $(\_this.measureDiv).html(tempText);

         $(\_this.measureDiv).css("left",evt.clientX+15);

         $(\_this.measureDiv).css("top",evt.clientY-15);

      });

    }

**if**(**this**.onMapDragEnd==**null**){

      console.log("dragEnd binded");

**this**.onMapDragEnd=dojo.connect(mapAppObj.map,"onMouseDragEnd",**function**(evt){

         console.log("dragEnd triggered");

         \_this.clearRadius();

      });

   }

    };//end bindDrags

    //解除drags Events

**this**.unbindDrags=**function**(){

      dojo.disconnect(**this**.onMapDragStart);

      dojo.disconnect(**this**.onMapDragging);

      dojo.disconnect(**this**.onMapDragEnd);

**this**.onMapDragStart=**null**;

**this**.onMapDragging=**null**;

**this**.onMapDragEnd=**null**;

**this**.targetShape=-1;

      console.log("drag events unbinded");

   };

}

## 自动补全搜索

路子:

       1.将关键字字典存到dictionary对象中

       2.input每次keyPress时，从dictionary中查出similar对象

       3.update tipsDiv 的innerHTML,显示tipsDiv

       4.点击tipsDiv或者 点击document某个部分时,隐藏tipsDiv

界面设计:

              <input> <searchButton>

              <tipsDiv>

举例：

//搜索工具, input-button-tips

//搜索工具, input-button-tips

**function** searchTool(){

**var** input=**null**;//关键字

**var** button=**null**;//搜索

**var** tips=**null**;//下拉选项

**var** \_this=**this**;

   //初始化

**this**.init=**function**(inputDiv,butnDiv,tipsDiv){

      console.log("searchTool.init is called");

**if**(inputDiv==**null**||inputDiv.tagName.toLowerCase()!="input"){

         console.log("invalid para,function searchToo.init()");

**return**;

      }

**if**(butnDiv==**null**||butnDiv.tagName.toLowerCase()!="input"){

         console.log("invalid para,function searchToo.init()");

**return**;

      }

**if**(butnDiv==**null**||tipsDiv.tagName.toLowerCase()!="div"){

         console.log("invalid para,function searchToo.init()");

**return**;

      }

      input=inputDiv;

      button=butnDiv;

      tips=tipsDiv;

      console.log("searchTool widgets validated");

      //文本框获取焦点,显示tips

      $(input).on("focus",**function**(){

         console.log("input focus is triggered");

         \_this.showTips(**true**);

      });

       mapAppObj.bindMapClick();//点击地图关闭searchTool

       //文本框失去焦点

      //本地运行正常,但放在门户上,无法触发input的onblur事件.

      $(input).on("blur",**function**(e){

         //onblur,e always be input itsself

         //when tips.children.click caused inputBlur,

         //if not delay, showTips(false) precede children().click,

         //then unable to selected tip;

         /\*

         console.log("input blur triggered,hideTips delayed 200ms");

         setTimeout(function(){

            \_this.showTips(false);

         },200);

         \*/

      });

      //关键字变更

      $(input).on("keyup",**function**(){

         console.log("input keyup is triggered");

         \_this.updateTips($(**this**).val());//更新备选项

         \_this.showTips(**true**);

      });

      //搜索按钮

      $(button).on("click",**function**(){

         console.log("search click is triggered");

**var** key=\_this.getKey();

         console.log("关键字是:"+key);

         \_this.showTips(**false**);

**var** stationFea=mapAppObj.stationIDNameObj.createStationFeature(key);

**if**(stationFea!=**null**){

            locateAndAddStation(stationFea);

**return**;

         }

         console.log("key not match any station");

**var** lineFea=mapAppObj.stationIDNameObj.createLineFeature(key);

**if**(lineFea!=**null**){

            locateAndAddLine(lineFea);

**return**;

         }

         alert("未找到要素:"+key);

      });

      //点击tips以外的区域,tips关闭

   };//end 初始化

   //定位,添加站为备选

   //参数:stationFea,graphic

**var** locateAndAddStation=**function**(stationFea){

      mapAppObj.map.centerAt(stationFea.geometry);

      formalizeFSAttributes([stationFea],"cp");

**var** flag=mapAppObj.candidatesDLSS.checkCandidateSelected(stationFea);

      console.log("searched feauture in already in candidate?:"+flag);

**if**(!flag){//if not added,add

         mapAppObj.candidatesDLSS.addCandidates([stationFea]);

      }

**return** **false**;

   };

   //定位,添加线为备选

**var** locateAndAddLine=**function**(lineFea){

      console.log("key found match in lines");

      //centerAt

**var** centerPt=mapAppObj.geometryUtil.getCenterPointOfLine(lineFea);

      mapAppObj.map.centerAt(centerPt);

      formalizeFSAttributes([lineFea],"cl");

**var** flag=mapAppObj.candidatesDLSS.checkCandidateSelected(lineFea);

      console.log("searched feauture in already in candidate?:"+flag);

**if**(!flag){//if not added,add

         mapAppObj.candidatesDLSS.addCandidates([lineFea]);

      }

**return**;

   };

   //获取输入的关键字

**this**.getKey=**function**(){

**return** $.trim($(input).val());

   };

   //显隐备选项

**this**.showTips=**function**(flag){

      console.log("showTips("+flag+") is called");

**if**(flag){

         //显示

**if**($(tips).hasClass("dNone")){

**if**($(tips).children().length==0){

               console.log("no children in tips,tips won't be showed");

**return**;

            }

**var** left=$(input).css("left");

**var** top=$(input).css("top")+$(input).css("height");

            console.log("tips location:"+left+","+top);

            $(tips).removeClass("dNone");

         }

**else**{**return**;}

      }

**else**{$(tips).addClass("dNone");}

   };

   //清空提示项,解除点击事件

**this**.clearTips=**function**(){

      console.log("clearTips is called");

**var** length=$(tips).children().length;

      console.log("tips.childeren.length is:"+length);

      $(tips).children().off("click");

      $(tips).empty();

   };

   //查询,更新tips

**this**.updateTips=**function**(key){

      console.log("updateTips is called");

**this**.clearTips();

      //get stationTips,lineTips

**var** stationTips=mapAppObj.stationIDNameObj.getSimilarStationNames(key);

**var** lineTips=mapAppObj.stationIDNameObj.getSimilarLineNames(key);

      //into dvTips

      console.log("tips get,stations:"+$.toJSON(stationTips)+",lines:"+$.toJSON(lineTips));

**var** tipsHtml="";

**for**(**var** i=0;i<=stationTips.length-1;i++){

         tipsHtml+="<p class='similarItem' objType=0>"+stationTips[i]+"</p>";

      }

**for**(**var** i=0;i<=lineTips.length-1;i++){

         tipsHtml+="<p class='similarItem' objType=1>"+lineTips[i]+"</p>";

      }

      $(tips).html(tipsHtml);

      //on click

      $(".similarItem").on("click",**function**(){

         console.log("tipItem "+**this**.innerHTML+" is clicked");

         $(input).val(**this**.innerHTML);

         \_this.showTips(**false**);

**var** type=**this**.getAttribute("objType");

         console.log("objType is:"+type);

**if**(type==0){//station

         //(1)locate； the feature

**var** tempStation=mapAppObj.stationIDNameObj.createStationFeature(**this**.innerHTML);

**if**(tempStation==**null**){

**return** **false**;

         }

         locateAndAddStation(tempStation);

**return** **false**;

      }//end if station

**else**{//line

**var** tempLine=mapAppObj.stationIDNameObj.createLineFeature(**this**.innerHTML);

**if**(tempLine==**null**){

**return** **false**;

         }

         locateAndAddLine(tempLine);

**return** **false**;

      }//end line

      });

   };

}

## 底图切换

界面：div影像,div行政区,divTieMap

 效果描述:

   3个图层:影像图,行政区划图,tileMap

   点击UI,移除当前底图,切换到目标底图

**function** baseMapMgr(){

   //底图类型  -1-无,0-高德,1-影像,2-行政

**this**.baseMapType=-1;

   //create autoNavi tile map calss;

   //MapABCClass can be used as constructor ONLY After this function executed

**this**.createMapABCClass=**function**(){

      //relied Modules:esri.layers.agstiled,esri.geometry,

      dojo.declare("MapABCLayer", esri.layers.TiledMapServiceLayer, {

          constructor: **function** () {

             console.log("MapABCLayer constructor is called");

**this**.spatialReference = **new** esri.SpatialReference({ wkid: 102113 });

**this**.initialExtent = (**this**.fullExtent = **new** esri.geometry.Extent(-20037508.342787, -20037508.342787, 20037508.342787, 20037508.342787, **this**.spatialReference));

**this**.id = "baseMapLayer";

**this**.tileInfo = **new** esri.layers.TileInfo({

                  "rows": 256,

                  "cols": 256,

                  "compressionQuality": 0,

                  "origin": {

                      "x": -20037508.342787,

                      "y": 20037508.342787

                  },

                  "spatialReference": {

                      "wkid": 102113

                  },

                  "lods": [

      { "level": 0, "resolution": 156543.033928, "scale": 591657527.591555 },

      { "level": 1, "resolution": 78271.5169639999, "scale": 295828763.795777 },

      { "level": 2, "resolution": 39135.7584820001, "scale": 147914381.897889 },

      { "level": 3, "resolution": 19567.8792409999, "scale": 73957190.948944 },

      { "level": 4, "resolution": 9783.93962049996, "scale": 36978595.474472 },

      { "level": 5, "resolution": 4891.96981024998, "scale": 18489297.737236 },

      { "level": 6, "resolution": 2445.98490512499, "scale": 9244648.868618 },

      { "level": 7, "resolution": 1222.99245256249, "scale": 4622324.434309 },

      { "level": 8, "resolution": 611.49622628138, "scale": 2311162.217155 },

      { "level": 9, "resolution": 305.748113140558, "scale": 1155581.108577 },

      { "level": 10, "resolution": 152.874056570411, "scale": 577790.554289 },

      { "level": 11, "resolution": 76.4370282850732, "scale": 288895.277144 },

      { "level": 12, "resolution": 38.2185141425366, "scale": 144447.638572 },

      { "level": 13, "resolution": 19.1092570712683, "scale": 72223.819286 },

      { "level": 14, "resolution": 9.55462853563415, "scale": 36111.909643 },

      { "level": 15, "resolution": 4.77731426794937, "scale": 18055.954822 },

      { "level": 16, "resolution": 2.38865713397468, "scale": 9027.977411 },

      { "level": 17, "resolution": 1.19432856685505, "scale": 4513.988705 },/\*equipents show,when scale inside 4513\*/

      { "level": 18, "resolution": 0.597164283559817, "scale": 2256.994353 },

      { "level": 19, "resolution": 0.298582141647617, "scale": 1128.497176 },

      { "level": 20, "resolution": 0.15, "scale": 564 },/\*the following scale is for showing equipment\*/

      { "level": 21, "resolution":0.075, "scale":282 },

      { "level": 22, "resolution": 0.037, "scale": 141},

      { "level": 23, "resolution": 0.018, "scale": 70 },

      { "level": 24, "resolution": 0.09, "scale": 35 },

      { "level": 25, "resolution": 0.04, "scale": 17},

      { "level": 26, "resolution": 0.02, "scale": 8},

      { "level": 27, "resolution": 0.01, "scale": 4},

      { "level": 28, "resolution": 0.005, "scale": 2},

      { "level": 29, "resolution": 0.002, "scale": 1},

      { "level": 30, "resolution": 0.001, "scale": 0},

      { "level": 31, "resolution": 0.0001, "scale": 0}

      ]

              });

**this**.loaded = **true**;

**this**.onLoad(**this**);

          },

          //返回level下 row,col处切片的url

          getTileUrl: **function** (level, row, col) {

              //云端地址：http://emap0.mapabc.com/mapabc/maptile?v=w2.61&x=845&y=387&z=10

              //内网地址：http://10.254.3.56:8082/maptile-service/maptile?v=w2.61&x=845&y=387&z=10

**return** gSettings.serviceUrl.autoNaviTileMap+"&z=" + level + "&x=" + col + "&y=" + row;

          }

      });

   };

   /\*描述：设置地图

    \* para:-1-无,0-高德,1-影像,2-行政

    \* \*/

**this**.setBaseMap=**function**(para){

         console.log("baseMap.setBaseMap has been triggered");

**if**(para!=-1&&para!=0&&para!=1&&para!=2){

            console.log("unexpected para,function setBaseMap");

**return**;

         }

**if**(para==**this**.baseMapType){//如果相同

**return**;

         }

**else**{//不同

**if**(**this**.baseMapType!=-1){//clear previous

**var** targetLayer=mapAppObj.map.getLayer("baseMapLayer");

                mapAppObj.map.removeLayer(targetLayer);

**this**.baseMapType=-1;

            }

**if**(para==-1){**return**;}

**else** **if**(para==0){

**var** mapABCLayer = **new** MapABCLayer();

                  mapAppObj.map.addLayer(mapABCLayer);

**this**.baseMapType=0;

            }

**else** **if**(para==1){

**var** imageLayer = **new** esri.layers.ArcGISImageServiceLayer(gSettings.serviceUrl.imageServiceUrl, { id: "baseMapLayer" });

                   mapAppObj.map.addLayer(imageLayer);

**this**.baseMapType=1;

            }

**else**{

**var** districtLayer = **new** esri.layers.ArcGISDynamicMapServiceLayer(gSettings.serviceUrl.districtMapUrl, { id: "baseMapLayer" });

                  mapAppObj.map.addLayer(districtLayer);

**this**.baseMapType=2;

            }

            mapAppObj.themeMaps.reloadEquipmentLayer();

         }//end 不同

   };//end setBaseMap

}

## 图层/要素闪烁

  //闪烁效果

              \_this.selectedW.graphicLayer.hide();

              setTimeout(**function**(){

                console.log("W delayedShow has been called");

                \_this.selectedW.graphicLayer.show();

              },200);

 10常见错误

## 缓冲区查询  dt.join is not a function

原因：dojo传递参数的时候出错。

解决方式：这是仍然使用GeometryService.buffer的方式。（可以使用生成圆来取代）

function getCircleFromPoint(centerPointGraphic, radius)

{

    var geometryService = new esri.tasks.GeometryService(\_thisMap.GeometryServiceUrl);

    var point = new esri.geometry.Point(centerPointGraphic.geometry.x,

                                        centerPointGraphic.geometry.y,

                                        new esri.SpatialReference({ wkid: 4326 })

                                        );

    var params = new esri.tasks.BufferParameters();

    params.geometries = [point];

    params.distances = [new Number(radius)];

    params.unit = esri.tasks.GeometryService.UNIT\_STATUTE\_MILE;

    params.bufferSpatialReference = new esri.SpatialReference({ wkid: 4326 });

    params.outSpatialReference = map.spatialReference;

    var obj = this;

    dojo.connect(geometryService, "onBufferComplete", obj, "geometryBufferCompleteCallback");

    geometryService.buffer(params);

}

## buffer,queryTask, 服务端Proxy脚本配置和客户端调用

备注：框选的圆大时，query.geometry是个含40~50个点的polygon,长度超过2000，在IE浏览器中，需要设置esri.config.proxyUrl

ArcGIS API for JavaScript默认方式HTTP请求是GET方式, 而不是Post方式。

当HTTP为长URL请求时，GET将无法使用，而必须采用Post的方式。Esri默认的http请求方式是GET。对于这种应用，如：以缓冲后的几何为参数的QueryTask，此情况下就需要使用POST的Proxy来处理这种情况。

出错提示：

RequestError: Unable to load /proxy?http://10.254.53.85:6080/arcgis/rest/services/TF/station/MapServer/0/query status: 12030

 ----------------------------------------------------------

解决步骤：添加proxy.ashx->

添加proxy.ashx (IIS)：

1,将proxy.ashx和proxy.config拷贝到网站目录下（IIS中，要将网站转为应用程序）

2.打开proxy.config,添加serverItem标签，url指向GISServer, 设置matchAll=false

    <serverItem url="http://localhost:6080/arcgis/rest/services/MyMapService/MapServer"

                matchAll="false" />

访问Url：

<http://localhost/LocalJsoutput/proxy.ashx>

测试方式

<http://localhost/LocalJsoutput/proxy.ashx?ping>

Buffer,queryTask页面添加代码：

esri.config.defaults.io.proxyUrl=

esri.config.defaults.io.alwaysUseProxy=false;

## proxy拒绝访问

        esri.config.defaults.io.proxyUrl = gProxyUrl;(http://10.254.53.75/jsoutput/proxy.ashx)

        esri.config.defaults.io.alwaysUseProxy = false;

 var gProxyUrl = "http://localhost/jsoutput/proxy.ashx";//防止跨域

 备注：即使服务器是 10.254.53.75,在客户端设置10.254.53.75,仍然是跨域

# 11常用布局

## Map全屏

说明：html,body,#map{width:100%,height:100%}

如果map外层套有其它标签，请把它们的width,height设置成100%;

 <style type="text/css">

\*{margin:0px; padding:0px;}

**html,body,**#map{width:100%;height:100%;}

</style>

<script type="text/javascript">

**var** mapObj=**null**;

dojo.ready(**function**(){

   mapObj=**new** esri.Map("map");

**var** districtLayer = **new** esri.layers.ArcGISDynamicMapServiceLayer(gSettings.serviceUrl.districtMapUrl, { id: "baseMapLayer" });

   mapObj.addLayer(districtLayer);

});

</script>

</head>

<body>

<div id="map"></div>

</body>

# 12路径分析

## RouteParameters

 routeParams = new esri.tasks.RouteParameters();

      routeParams.stops = new esri.tasks.FeatureSet();//push stop graphic into ithis

      routeParams.returnRoutes = true;

      routeParams.returnDirections = true;//sometime may error

      routeParams.returnStops = true;//stops you pushed into

      routeParams.directionsLengthUnits = esri.Units.KILOMETERS;

      routeParams.outSpatialReference = map.spatialReference;

## RouteParametres 错误和解决方式

[Failed to convert the input propertyset into a recordset to load into the NAClass "Stops".  Input property [FID] could not be mapped to a valid field type or was not specified in the first PropertySet which is used to define the input fields.]

 //Reason, attributes of stop graphic has extra attributes can’t be found in published service;

 //solution: delete unnecessary attributes

var pathByGraphic = new esri.Graphic(noDuplicatedStations[i].geometry, null, {"NAME":noDuplicatedStations[i].attributes.NAME}, null);

## RouteTask, solveResluts

## 在线例子列表:

 Find a route

 Find directions

驾车路线，每一段

 Find a rounte around Barriers

Find Nearest facility

Find Area arriving with 1 minute

 Dynamic show tranvle area  with time changing

## Route  结构

最关键的部分是 polyline

## RouteTask 和RouteParameter

**Tips**:

à(1)routeTaskUrl point to  NAServer

  (2)routeParameters.features must accept **Graphic** from Point(not multipoint);

**Sentences**:

var routeTask = new esri.tasks.RouteTask(networkService);

        var routeParas = new esri.tasks.RouteParameters();

        routeParas.outSpatialReference = map.spatialReference;//outspatialReference of rp

        routeParas.stops = new esri.tasks.FeatureSet();//stops of rp, each stop is a esri.Grahic

       var ptStart = new esri.geometry.Point(startFeatrue.geometry.points[0], map.spatialReference);

       var ptEnd = new esri.geometry.Point(endFeature.geometry.points[0], map.spatialReference);

       var tempGra1=new esri.Graphic(ptStart,symbol,null,null);

       var tempGra2=new esri.Graphic(ptEnd,symbol,null,null);

       routeParas.stops.features.push(tempGra1);//routeParameter.features must be type Graphic

       routeParas.stops.features.push(tempGra2);

----------------------------

      var routeSymbol = new esri.symbol.SimpleLineSymbol().setColor(new dojo.Color([0, 0, 255, 0.5])).setWidth(5);

        routeTask.solve(routeParas, function (solveResults) {

            for (var j = 0; j <= solveResults.routeResults.length - 1; j++) {

                var oneRouteGra = solveResults.routeResults[0].route;

                oneRouteGra.setInfoTemplate(new esri.InfoTemplate("奥林匹克公园-->天通苑", "<tr>途径换乘<td>奥林匹克公园-->霍营-->立水桥-->天通苑</tr></td><br><tr>OD量：<td>4503/tr></td><br><tr>OD占比:<td>83.1%</tr></td><br>"));

                oneRouteGra.setSymbol(routeSymbol);

                map.graphics.add(oneRouteGra);

            }

        }, function (error) {

            alert(error);

        });

**Example**:

var startGra = new esri.Graphic(ptStart, new esri.symbol.PictureMarkerSymbol("src/assets/images/layer.png", 24, 24), null, null);

var endGra = new esri.Graphic(ptEnd, new esri.symbol.PictureMarkerSymbol("src/assets/images/station\_small.png", 24, 24), null, null);

 var routeTask = new esri.tasks.RouteTask(networkService);//routeTask and routePara

var routeParas = new esri.tasks.RouteParameters();

routeParas.outSpatialReference = map.spatialReference;

routeParas.returnRoutes = true;

routeParas.stops = new esri.tasks.FeatureSet();

routeParas.stops.features.push(startGra);

routeParas.stops.features.push(endGra);

routeTask.solve(routeParas, function (solveResults) {//geRoute

    if (solveResults.routeResults.length == 0) { alert("failed to find route"); return; }

    var oneRouteResult = solveResults.routeResults[0];

    var routeSymbol = new esri.symbol.SimpleLineSymbol().setColor(new dojo.Color([255, 0, 0])).setWidth(5);

    var routeGraphic = new esri.Graphic(oneRouteResult.route.geometry, routeSymbol, null, null);

    graLayerRoute.clear(); //(clear previous)

    map.infoWindow.hide();

                                                    //(infoWindow location)

    var tempLoc = parseInt(oneRouteResult.route.geometry.paths[0].length / 2); //(中间段的 中间点)

    var pt = new esri.geometry.Point(oneRouteResult.route.geometry.paths[0][tempLoc], map.spatialReference);

    map.infoWindow.setTitle(endFeature.attributes.NAME + "-->" + destName);

    map.infoWindow.setContent("<tr>途经换乘:<td></tr></td><br><tr>点到点OD总量:<td></tr></td><br><tr>本方案OD量:<td></tr></td><br><tr>占比:<td></tr></td>");

    map.infoWindow.show(pt, map.getInfoWindowAnchor(pt));

    graLayerRoute.add(routeGraphic); //route gra

    graLayerRoute.add(startGra);

    graLayerRoute.add(endGra);

    map.addLayer(graLayerRoute);

}); //end routeTask callback

## Get Route Length

solveResults[i][1].routeResults[0].route.attributes.Total\_Length

## Route sort by routeLength

**//sort Function**

<script type="text/javascript">

    var allRoutes = [{ "routeResults": [{ "route": { "attributes": { "Total\_Length": 4, "NAME": "Route1"}}}] },

    { "routeResults": [{ "route": { "attributes": { "Total\_Length": 3, "NAME": "Route2"}}}] },

    { "routeResults": [{ "route": { "attributes": { "Total\_Length": 2, "NAME": "Route3"}}}] },

    { "routeResults": [{ "route": { "attributes": { "Total\_Length": 1, "NAME": "Route4"}}}] }

    ];

     function sortFunction(a, b) {

        return a.routeResults[0].route.attributes.Total\_Length - b.routeResults[0].route.attributes.Total\_Length;

    }

        allRoutes.sort(sortFunction);

</script>

**//get the top 3**

  var noDupplicatedRoutes = clearSameLengthRoute(solveResults);

                 var topThree = function () {

                    var sortedRoutes = noDupplicatedRoutes.sort(sortRouteFunction);

                    return sortedRoutes.slice(0,3);//only top 3 left

                } (); //get the 3 shortest routes

## Route 地铁 前3换乘路径算法

1.传起点，终点，找出最短路径

   (1)如果起点，终点的lineID相同,显示最短路径。

       结束。

   (2)如果起点，终点的lineID不相同.

      -->1.得到最短路径

           奥林匹克公园->大屯路东->天通苑

     //方法1（换乘站个数比方法2多）

      (起点lineID:15号线,8号线

         终点lineID:5号线

         -->找出起点线相交的线

            (15号线)5号线,13号线,14号线

               (8号线)13号线,15号线,10号线,2号线,6号线

          )

     //方法2

      -->使用矩形范围，求出矩形范围内的换乘站

           望京西,霍营,立水桥,奥林匹克公园,大屯路东

         -->删除在最短路径上的换乘站

            删除： 奥林匹克公园,大屯路东,立水桥。

               余下：望京西,霍营

         -->求经过这两个换乘站的路径.

             (1)奥->霍->天

                (2)奥->望->大->天

         -->删除自相交的路线

            删除：奥->望->大->天

               余下：奥->霍->天

/\*路径分析 (多条路径)\*/

function queryRoutes(startName, endName) {

    var queryTask = new esri.tasks.QueryTask(analysisMapUrl + "/0");

    var query = new esri.tasks.Query();

    query.outSpatialReference = map.spatialReference;

    query.returnGeometry = true;

    query.outFields = ["\*"];

    query.where = "NAME='" + startName + "'";

    var tepmlist = new Array();

    tepmlist.push(queryTask.execute(query));

    query.where = "NAME='" + endName + "'";

    tepmlist.push(queryTask.execute(query));

    var deferList = new dojo.DeferredList(tepmlist);

    deferList.then(function (results) {//get start and end

        if (results[0][1].features.length == 0) { alert("未找到起点站'" + startName + "'.如果该站已经开通,请更新数据"); return; }

        if (results[1][1].features.length == 0) { alert("未找到终点站'" + endName + "'.如果该站已经开通,请更新数据"); return; }

        var tempGra1 = new esri.Graphic(results[0][1].features[0].geometry, new esri.symbol.PictureMarkerSymbol("src\assets\images\startStation.png", 22, 22), results[0][1].features[0].attributes, new esri.InfoTemplate("起点信息：", "${NAME}"));

        var tempGra2 = new esri.Graphic(results[1][1].features[0].geometry, new esri.symbol.PictureMarkerSymbol("src\assets\images\end.png", 22, 22), results[1][1].features[0].attributes, new esri.InfoTemplate("终点信息：", "${NAME}"));

        //get shortestRoute

        var routeTask = new esri.tasks.RouteTask(networkService);

        var routeParas = new esri.tasks.RouteParameters();

        routeParas.outSpatialReference = map.spatialReference;

        routeParas.returnStops = true;

        routeParas.returnRoutes = true;

        routeParas.stops = new esri.tasks.FeatureSet();

        routeParas.stops.features.push(tempGra1);

        routeParas.stops.features.push(tempGra2);

        var tempAr = [routeTask.solve(routeParas)];

        var deferList = new dojo.DeferredList(tempAr);

        deferList.then(function (results) {//defer for shortest route

            if (results[0][0] == false) { alert("未能执行查询,请检查network service"); return; }

            if (results[0][1].length == 0) { alert("两站间没有可达路径"); return; }

            var routeSymbol = new esri.symbol.SimpleLineSymbol().setColor(new dojo.Color([255, 0, 0])).setWidth(5);

            var shortestRoute = results[0][1].routeResults[0].route;

            //start end at one line, show one only

            if (checkStationSameLine(tempGra1, tempGra2)) {

                //alert("在同一条线上");

                shortestRoute.setInfoTemplate(new esri.InfoTemplate(tempGra1.attributes.NAME + "-->" + tempGra2.attributes.NAME, "<tr>乘车路线<td>" + tempGra1.attributes.NAME + "-->" + tempGra2.attributes.NAME + "</tr></td><br><tr>OD量：<td>4503</tr></td><br><tr>OD占比:<td>83.1%</tr></td><br>"));

                shortestRoute.setSymbol(routeSymbol);

                var routeLength = results[0][1].routeResults[0].route.attributes.Total\_Length;

                var li = document.createElement("li"); //创建li

                li.className = "marTB5";

                li.routeGraphic = shortestRoute; //li添加属性,features,起点,终点

                li.routeID = 1;

                li.startStation = tempGra1.attributes.NAME;

                li.endStation = tempGra2.attributes.NAME;

                li.startPoint = tempGra1.geometry;

                li.endPoint = tempGra2.geometry;

                li.title = tempGra1.attributes.NAME + "-->" + tempGra2.attributes.NAME;

                li.routeName = tempGra1.attributes.NAME + "-->" + tempGra2.attributes.NAME + "(距离:" + Math.round(routeLength) + "米)";

                li.ODTotal = 5638;

                li.ODValue = 4533;

                li.ODPartition = (li.ODValue / li.ODTotal \* 100).toString().substr(0, 5) + "%";

                li.appendChild(document.createTextNode("1. " + li.title));

                li.onmouseover = function () {//悬浮 显示

                    this.style.cursor = 'pointer';

                    graLayerRoute.clear(); //清空

                    map.infoWindow.hide();

                    graLayerRoute.add(this.routeGraphic); //线路

                    //起点，终点

                    var infoTemplate = new esri.InfoTemplate("起点信息", "站点名:" + this.startStation);

                    var startGra = new esri.Graphic(this.startPoint, new esri.symbol.PictureMarkerSymbol('src/assets/images/layer40.png', 22, 22), null, infoTemplate);

                    infoTemplate = new esri.InfoTemplate("终点信息", "站点名:" + this.startStation);

                    var endGra = esri.Graphic(this.endPoint, new esri.symbol.PictureMarkerSymbol('src/assets/images/end.png', 22, 22), null, infoTemplate);

                    graLayerRoute.add(startGra);

                    graLayerRoute.add(endGra);

                    map.addLayer(graLayerRoute);

                }

                dojo.byId("ulRoutes").appendChild(li);

            }

            else {//start,end not at same Line

                //alert("不在同一条线上");

                var xmin = (tempGra1.geometry.x >= tempGra2.geometry.x) ? tempGra2.geometry.x : tempGra1.geometry.x;

                var ymin = (tempGra1.geometry.y >= tempGra2.geometry.y) ? tempGra2.geometry.y : tempGra1.geometry.y;

                var xmax = (tempGra1.geometry.x <= tempGra2.geometry.x) ? tempGra2.geometry.x : tempGra1.geometry.x;

                var ymax = (tempGra1.geometry.y <= tempGra2.geometry.y) ? tempGra2.geometry.y : tempGra1.geometry.y;

                var transferExtent = new esri.geometry.Extent(xmin - 0.04, ymin - 0.03, xmax + 0.04, ymax + 0.03);

                var queryTask = new esri.tasks.QueryTask(analysisMapUrl + "/0");

                var query = new esri.tasks.Query();

                query.outSpatialReference = map.spatialReference;

                query.outFields = ["\*"];

                query.where = "isTransfer=1";

                query.geometry = transferExtent;

                query.returnGeometry = true;

                var deferList = new dojo.DeferredList([queryTask.execute(query)]);

                esri.config.defaults.io.proxyUrl = gProxyUrl;

                esri.config.defaults.io.alwaysUseProxy = false;

                deferList.then(function (transferStations) {//transferStations

                    var routeSovleArray = [];

                    var noDuplicatedStations = clearSameNameFeatures(transferStations[0][1]); //clearSameName

                    var hasTransferStation = false;

                    for (var i = noDuplicatedStations.length - 1; i >= 0; i--) {

                        console.log("NO" + i.toString() + " transfer Name:" + noDuplicatedStations[i].attributes.NAME);

                        if (noDuplicatedStations[i].attributes.NAME == tempGra1.attributes.NAME || noDuplicatedStations[i].attributes.NAME == tempGra2.attributes.NAME) {

                            console.log("removed transfer: " + "NO " + i.toString() + ":" + noDuplicatedStations[i].attributes.NAME);

                            noDuplicatedStations.splice(i, 1); //remove station, if it is start or end

                        }

                    }

                    for (var i = noDuplicatedStations.length - 1; i >= 0; i--) {

                        console.log("NO" + i.toString() + " transfer Name:" + noDuplicatedStations[i].attributes.NAME);

                    }

                    if (noDuplicatedStations.length != 0) {// if has transfer within extent

                        hasTransferStation = true;

                        for (var i = 0; i <= noDuplicatedStations.length - 1; i++) {

                            var routeTask = new esri.tasks.RouteTask(networkService);

                            var routeParas = new esri.tasks.RouteParameters();

                            routeParas.returnStops = true;

                            routeParas.returnRoutes = true;

                            routeParas.outSpatialReference = map.spatialReference;

                            routeParas.stops = new esri.tasks.FeatureSet();

                            routeParas.stops.features.push(tempGra1);

                            var pathByGraphic = new esri.Graphic(noDuplicatedStations[i].geometry, null, { "NAME": noDuplicatedStations[i].attributes.NAME }, null);

                            routeParas.stops.features.push(pathByGraphic);

                            routeParas.stops.features.push(tempGra2);

                            routeSovleArray.push(routeTask.solve(routeParas));

                        }

                    }

                    else {// don't have transfer

                        var routeTask = new esri.tasks.RouteTask(networkService);

                        var routeParas = new esri.tasks.RouteParameters();

                        routeParas.returnStops = true;

                        routeParas.returnRoutes = true;

                        routeParas.outSpatialReference = map.spatialReference;

                        routeParas.stops = new esri.tasks.FeatureSet();

                        routeParas.stops.features.push(tempGra1);

                        routeParas.stops.features.push(tempGra2);

                        routeSovleArray.push(routeTask.solve(routeParas));

                    }

                    var derferListRoute = new dojo.DeferredList(routeSovleArray);

                    derferListRoute.then(function (solveResults) {

                        var routeSymbol = new esri.symbol.SimpleLineSymbol().setColor(new dojo.Color([255, 0, 0])).setWidth(5);

                        for (var j = 0; j <= solveResults.length - 1; j++) { //get transferName

                            if (solveResults[j][1].routeResults[0].stops.length > 2) {

                                solveResults[j][1].routeResults[0].routeName = solveResults[j][1].routeResults[0].stops[0].attributes.Name + "-->" + solveResults[j][1].routeResults[0].stops[1].attributes.Name + "-->" + solveResults[j][1].routeResults[0].stops[2].attributes.Name;

                            }

                            else { solveResults[j][1].routeResults[0].routeName = solveResults[j][1].routeResults[0].stops[0].attributes.Name + "-->" + solveResults[j][1].routeResults[0].stops[1].attributes.Name; }

                        }

                        var filtereddRoutes = filterRoutes(solveResults); //clear same length Routes

                        for (var j = 0; j <= filtereddRoutes.length - 1; j++) {

                            var oneRouteGra = filtereddRoutes[j][1].routeResults[0].route;

                            var transferName = filtereddRoutes[j][1].routeResults[0].stops[1].attributes.Name;

                            oneRouteGra.setInfoTemplate(new esri.InfoTemplate(tempGra1.attributes.NAME + "-->" + transferName + "-->" + tempGra2.attributes.NAME, "<tr>途径换乘<td>奥林匹克公园-->霍营-->立水桥-->天通苑</tr></td><br><tr>OD量：<td>4503</tr></td><br><tr>OD占比:<td>83.1%</tr></td><br>"));

                            oneRouteGra.setSymbol(routeSymbol);

                            var routeLength = filtereddRoutes[j][1].routeResults[0].route.attributes.Total\_Length;

                            var li = document.createElement("li"); //创建li

                            li.className = "marTB5";

                            li.routeGraphic = oneRouteGra; //li添加属性,features,起点,终点

                            li.routeID = (j + 1).toString();

                            li.startStation = tempGra1.attributes.NAME;

                            li.endStation = tempGra2.attributes.NAME;

                            li.startPoint = tempGra1.geometry;

                            li.endPoint = tempGra2.geometry;

                            li.hasTransferStation = hasTransferStation;

                            if (hasTransferStation) {

                                li.title = tempGra1.attributes.NAME + "-->" + transferName + "-->" + tempGra2.attributes.NAME;

                                li.routeName = tempGra1.attributes.NAME + "-->" + transferName + "-->" + tempGra2.attributes.NAME + "(距离:" + Math.round(routeLength) + "米)";

                            }

                            else {

                                li.title = tempGra1.attributes.NAME + "-->" + tempGra2.attributes.NAME;

                                li.routeName = tempGra1.attributes.NAME + "-->" + tempGra2.attributes.NAME + "(距离:" + Math.round(routeLength) + "米)";

                            }

                            li.ODTotal = 5638;

                            li.ODValue = 4533;

                            li.ODPartition = (li.ODValue / li.ODTotal \* 100).toString().substr(0, 5) + "%";

                            li.appendChild(document.createTextNode((j + 1).toString() + "  ." + li.title));

                            li.onmouseover = function () {//悬浮 显示

                                this.style.cursor = 'pointer';

                                graLayerRoute.clear(); //清空

                                map.infoWindow.hide();

                                var middlefeature = this.routeGraphic; //infoWindow位置

                                var tempLoc = parseInt(middlefeature.geometry.paths[0].length / 2); //(中间段的 中间点)

                                var pt = new esri.geometry.Point(middlefeature.geometry.paths[0][tempLoc], map.spatialReference);

                                map.infoWindow.setTitle(this.title);

                                if (this.hasTransferStation) {

                                    map.infoWindow.setContent("<tr>途经换乘:<td>" + this.routeName + "</tr></td><br><tr>点到点OD总量:<td>" + this.ODTotal + "</tr></td><br><tr>本方案OD量:<td>" + this.ODValue + "</tr></td><br><tr>占比:<td>" + this.ODPartition + "</tr></td>");

                                }

                                else {

                                    map.infoWindow.setContent("<tr>乘车路线:<td>" + this.routeName + "</tr></td><br><tr>点到点OD总量:<td>" + this.ODTotal + "</tr></td><br><tr>本方案OD量:<td>" + this.ODValue + "</tr></td><br><tr>占比:<td>" + this.ODPartition + "</tr></td>");

                                }

                                map.infoWindow.resize(250, 200);

                                map.infoWindow.show(pt, map.getInfoWindowAnchor(pt));

                                graLayerRoute.add(this.routeGraphic); //线路

                                //起点，终点

                                var infoTemplate = new esri.InfoTemplate("起点信息", "站点名:" + this.startStation);

                                var startGra = new esri.Graphic(this.startPoint, new esri.symbol.PictureMarkerSymbol('src/assets/images/layer40.png', 22, 22), null, infoTemplate);

                                infoTemplate = new esri.InfoTemplate("终点信息", "站点名:" + this.startStation);

                                var endGra = esri.Graphic(this.endPoint, new esri.symbol.PictureMarkerSymbol('src/assets/images/end.png', 22, 22), null, infoTemplate);

                                graLayerRoute.add(startGra);

                                graLayerRoute.add(endGra);

                                map.addLayer(graLayerRoute);

                            }

                            dojo.byId("ulRoutes").appendChild(li);

                        } //for end

                    }); //end deferList.solveRoutes

                }); //end query transferStations

            }

        }); //end defer for shortest route

    });         //end defer start,end

}

# 13地图打印

## 代码打印

preRequirement: Printing tools must be started at ServerSide

<http://localhost:6080/arcgis/rest/services/Utilities/PrintingTools/GPServer/Export%20Web%20Map%20Task>

代码：

function startPrintTask() {

    //var url = 'http://10.254.53.85:6080/arcgis/rest/services/TF/station/MapServer/export';

    var printTask = new esri.tasks.PrintTask(mapPrintService);

    esri.config.defaults.io.proxyUrl = gProxyUrl;

    esri.config.defaults.io.alwaysUseProxy = false;

    //get print templates from the export web map task

    var printInfo = esri.request({

        "url": mapPrintService,

        "content": { "f": "json" }

    });

    printInfo.then(function (resp) {

        var layoutTemplate, templateNames, mapOnlyIndex, templates;

        layoutTemplate = dojo.filter(resp.parameters, function (param, idx) {

            return param.name === "Layout\_Template";

        });

        if (layoutTemplate.length == 0) {

            console.log("print service parameters name for templates must be \"Layout\_Template\"");

            return;

        }

        templateNames = layoutTemplate[0].choiceList;

        // remove the MAP\_ONLY template then add it to the end of the list of templates

        mapOnlyIndex = dojo.indexOf(templateNames, "MAP\_ONLY");

        if (mapOnlyIndex > -1) {

            var mapOnly = templateNames.splice(mapOnlyIndex, mapOnlyIndex + 1)[0];

            templateNames.push(mapOnly);

        }

        // create a print template for each choice

        templates = dojo.map(templateNames, function (ch) {

            var plate = new esri.tasks.PrintTemplate();

            plate.layout = plate.label = ch;

            plate.format = "PDF";

            plate.layoutOptions = {

                "authorText": "北京轨道交通指挥中心",

                "copyrightText": "<北京轨道交通指挥中心>",

                "legendLayers": [],

                "titleText": "北京地铁路网图",

                "scalebarUnit": "Kilometers"

            };

            return plate;

        });

        // create the print dijit

        printer = new esri.dijit.Print({

            "map": map,

            //"templates": templates,

            url: mapPrintService

        }, dojo.byId("print\_button"));

        printer.startup();

    },function () { }); //(handlerEnd, errohandler start)

}

## 用控件打印

Html code:

<div id="print\_button"></div>

PrintCode:

var printer

function startPrintEasyTask() {

       printer = new esri.dijit.Print({

            "map": map,

            //"templates": templates,

            url: mapPrintService

        }, dojo.byId("print\_button"));

        printer.startup();

}

# 14 jsAPI各版本新特性

## 3.7特性

## 支持 on(“mouse-move”,function(){});

map1.on("extent-change", onExtentChange);

     map1.on("pan",onPan);

     map1.on("mouse-move", drawCoordinates1);

     map1.on("mouse-drag", drawCoordinates1);

举例:

  map.on("mouse-move",**function**(evt){

            console.log("mouse coord is:(x,y)=("+evt.mapPoint.x+","+evt.mapPoint.y+")");

         });

# 15 浏览器对事件支持

|  |  |  |
| --- | --- | --- |
| 事件 | 支持版本 | 不支持 |
| Mouse-move | IE7,IE8,IE9,IE10 |  |
| Mouse-pan | IE7,IE8,IE9,IE10 |  |
| Mouse-drag  Mouse-drag-start  Mouse-drag-end | IE7,IE8,IE9 | IE10 |
| Pan-out-of-box | IE10 | IE9(IE9仅支持inside box pan) |
|  |  |  |

# 95HTML5 特性

## Canvas to show Raster Layer

## Cross-Origin-Resource Sharing

Requirement: both Server side and Browser support CORS

                     For server side, web servers must be pre-configured to support CORS.

           For browser , FF20 suppor CORS, while IE9 doesn’t by default

Boost: no proxy page needed any more

 Details:

       Cross-Origin Resource Sharing(CORS) allows web applications to bypass a browser's same origin policy. When both the web server and browser support CORS, a proxy is not required to do cross-domain requests. This both simplifies application development and potentially provides a performance boost. Development is simplified because it is no longer necessary to maintain a proxy page on your server. The performance boost comes from accessing a resource directly rather than sending a request back to your server, which then requests the specified URL and returns the result.

## Access Local file by drag/drop

Resource:

file:///F:/DevelopSoftware/ArcGIS%20Server%2010.1%20SP1\_%E8%AF%B4%E6%98%8E\_%E6%95%99%E7%A8%8B/JSAPI%E5%BC%80%E5%8F%91%E7%AC%94%E8%AE%B0/JavascriptAPI\_Reference/v33\_Samples/arcgis/exp\_dragdrop.html

## Local Storage

  原文地址：http://www.cnblogs.com/xiaowei0705/archive/2011/04/19/2021372.html

 //判断是否支持：

if(window.localStorage){  
 alert('This browser supports localStorage');  
}else{  
 alert('This browser does NOT support localStorage');  
}

//读写：

localStorage.a = 3;//设置a为"3"  
localStorage["a"] = "sfsf";//设置a为"sfsf"，覆盖上面的值  
localStorage.setItem("b","isaac");//设置b为"isaac"  
var a1 = localStorage["a"];//获取a的值  
var a2 = localStorage.a;//获取a的值  
var b = localStorage.getItem("b");//获取b的值  
localStorage.removeItem("c");//清除c的值

# 96POST & AJAX

## 解析Request的InputStream，QueryString

## Post- InputStream

  /\*descrition: convert, req data stream to string

     \* when xhrPost, context.Request.InputStream will have data

     \*/

       public string getRequestInputStream(HttpContext context) {

        string result = "";

        System.IO.Stream stream = context.Request.InputStream;

        int dataLength=Convert.ToInt32(stream.Length);

        byte[] bufferBytes=new byte[dataLength];

        stream.Read(bufferBytes, 0, dataLength);

        //Decode format shall be same as charset in page

        result= HttpUtility.UrlDecode(bufferBytes, System.Text.Encoding.UTF8);

        return result;

    }

## Get-QueryString

"OStations:苹果园,国贸,passStations:西直门,建国门,DStations:五道口,芍药居,Date:2013-04-01,Time:9:00-19:00"

## 服务端发httpRequest请求GET

          string newUrl = context.Request.QueryString["url"].ToString()+"?type=";

          System.Net.WebRequest newRequest = System.Net.WebRequest.Create(newUrl);

          System.Net.WebResponse response = newRequest.GetResponse();

          System.IO.Stream responseStream = response.GetResponseStream();

          System.IO.StreamReader readStream = new System.IO.StreamReader(responseStream);

          string results = readStream.ReadToEnd();

          context.Response.Write(results);

## 服务端发httpRequest请求POST

  public void ProcessRequest (HttpContext context) {

        System.IO.Stream dataStream = context.Request.InputStream;

        string url = context.Request.QueryString["href"] ;

        string result = SendPostWebRequest(url, "POST", dataStream);

        context.Response.ContentType = "text/plain";

        context.Response.Write(result);

    }

/// <summary>

    /// 转发post请求

    /// </summary>

    /// <param name="url">newUrl</param>

    /// <param name="method">"POST"</param>

    /// <param name="dataStream">data stream</param>

    /// <returns>resultString</returns>

    public string SendPostWebRequest(string url,string method,System.IO.Stream dataStream) {

        //data transported on internet are stream.

        //1.stream ->stream.read(byte[],0,stream.length)  stream into bytes[]

        //2.streamReader=new streamReader(stream), streamReader.ReadToEnd()   stream to string

        //3.stream.write(bytes[],0,bytes.length)   bytes write into stream

        System.Net.WebRequest request = System.Net.WebRequest.Create(url);

        request.Method = method;

        request.ContentType = "application/x-www-form-urlencoded";

        byte[] dataBytes = new byte[dataStream.Length];

        dataStream.Read(dataBytes, 0,dataBytes.Length);//stream into bytes[]

        System.IO.Stream postStream = request.GetRequestStream();

        postStream.Write(dataBytes, 0, dataBytes.Length);//bytes[] into stream

        postStream.Close();

        System.Net.WebResponse response=request.GetResponse();

        System.IO.Stream responseStream = response.GetResponseStream();

        System.IO.StreamReader sr = new System.IO.StreamReader(responseStream);//stream to string

        string resultString = sr.ReadToEnd();

        return resultString;

    }

## AJAX发送GET和POST

## AJAX军规

在url后面添加?time=一个不会重复的时间, 否则将受缓存影响

    var url = "http://localhost/ODWeb/getODHandler.ashx?type=&time=" + new Date().toLocaleString();

        xmlHttp.open("POST", url, true);

        xmlHttp.send(postData);

## AJAX GET

function sendReq() {

        var xmlHttp = GetXmlHttpObject();

        xmlHttp.onreadystatechange = function () {

            if (this.readyState == 4) {

                alert(this.responseText); //call back snippets, use responseText for data

            }

        };

        var url="photos.json";

        xmlHttp.open("GET", url, true);

        xmlHttp.send(null);

    }

    function GetXmlHttpObject() {

        var xmlHttp = null;

        try {

            // Firefox, Opera 8.0+, Safari

            xmlHttp = new XMLHttpRequest();

        }

        catch (e) {

            // Internet Explorer

            try {

                xmlHttp = new ActiveXObject("Msxml2.XMLHTTP");

            }

            catch (e) {

                xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");

            }

        }

        return xmlHttp;

    }

## AJAX POST

function sendReq() {

        var xmlHttp = GetXmlHttpObject();

        xmlHttp.onreadystatechange = function () {

            if (this.readyState == 4) {

                alert(this.responseText); //call back snippets, use responseText for data

            }

        };

        var url = "../getODHandler.ashx";

        xmlHttp.open("POST", url, true);

        xmlHttp.send(postData);

    }

    function GetXmlHttpObject() {

        var xmlHttp = null;

        try {

            // Firefox, Opera 8.0+, Safari

            xmlHttp = new XMLHttpRequest();

        }

        catch (e) {

            // Internet Explorer

            try {

                xmlHttp = new ActiveXObject("Msxml2.XMLHTTP");

            }

            catch (e) {

                xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");

            }

        }

        return xmlHttp;

    }

# 97 Type of Services in GIS Server

## Dynamic Map Service

   MapDocument, share as Service

## NetworkAnalysis  Service

 Tool:ArcMap ->share as

 Data requirements:  Network Dataset->New Route.

 Operation: Enable NAService in publish process

 Publish: Publish the Route is enough.

## Geocoder Service

Introductions:

http://localhost:6080/arcgis/help/en/#/Geocode\_services/01540000035t000000/

Steps:

  à1.create address locator in catlog

  à2.publish address locator as geocoder service

  à3.use javascript API to call service

Example:

**Step1 Create a station Locator:**

   à(1)open toolbox, search ”address Locator”,

   à(2) address Locator Style: General single field

        Reference data, station.shp

        Key Field:NAME

   à(3) select a place for locator, 3 files will be created:

              stationLocator.loc,

              stationLocator.loc.xml

              stationLocator.lox

   à(4) add the stationLocator into arcMap, Use the Find button

 In toolbar, select tab “Locator”, input a name ”xizhimen”, you’ll find a point shrink on map. By now, you’ve created a locator.

**Step2 publish address locator to geocode service**

After published to Server, visit this page, stationLocator/GeocodeServer, click the

Find address candidate, then input a address for search, if location returned, works well.

Example:

http://localhost:6080/arcgis/rest/services/subwayWGS84/stationLocator/GeocodeServer/findAddressCandidates?SingleKey=%E4%B8%9C%E7%9B%B4%E9%97%A8&Single+Line+Input=&outFields=&outSR=&searchExtent=&f=html

## Image Service

What dataSource can be published as Image Service

à1Raster Dataset (栅格数据集)

à2Raster Layer(数据集中的某个图层)

à3Mosaic Dataset (镶嵌数据集)

à4Mosaic Layer(数据集中的某个图层)

备注，RasterCatalog 是 “栅格目录”

Steps:

 Right click the raster dataset in **Arccatlog**, share as image Service

Key:

image service  (arcMap help)

ResultURL:

[http://10.254.53.85:6080/arcgis/rest/services/2012bjsubwayrouteCopy/**ImageServer**](http://10.254.53.85:6080/arcgis/rest/services/2012bjsubwayrouteCopy/ImageServer)

**à**不拷贝数据到server目录下，如何发布服务

   设置影像数据为共享(有更好解决方式)

## 3 tools to publish services

Range: Server help

Key:How to publish service

<http://localhost:6080/arcgis/help/en/#/How_to_publish_a_service/0154000004n3000000/>

summary:

   service type:                                    tool

   àmap service, NA service                        arcMap

   àgeocoder service, geodatabase, imageService        catlog

   àgeoprocessing tool,Model                                  Results window

# 98 widget

## Measurement widget

## Geocoder And Locator

Workflow: shpà address Locatorà geocode service

Required service:

       stationLocator/GeocodeServer

    A geocodeserver contains geocode service

Required pre-data processing:

Toolbox create address locator

Catlog publish address locator as Geocode Service

## Geocoder  snippets

 dojo.require("esri.dijit.Geocoder"); //geocoder widget (the searchTextBox)

var geocoderParas = [{ //self-defined geocoder

                url: "http://localhost:6080/arcgis/rest/services/subwayWGS84/stationLocator/GeocodeServer",

                name: "stationLocator"// name of Locator  in geocodeService

            }];

            geocoder = new esri.dijit.Geocoder({// construct geocoder widget

                map: map,

                autoComplete: true,

                arcgisGeocoder: false, //don't use argis global geocoder

                geocoders: geocoderParas,

                outFields: ["\*"]

            }, "search");

            geocoder.startup();

            dojo.connect(geocoder, "onSelect", function (result) {//when one result is selected

                //result.feature is type graphic

                map.graphics.clear(); //clear previous graphics

                map.infoWindow.hide();

## Locator snippets

备注：自己实现效率高

dojo.require("esri.tasks.locator");

++++++++++++++++++++++++++++++++++++++++++

  locator = new esri.tasks.Locator("http://localhost:6080/arcgis/rest/services/subwayWGS84/stationLocator/GeocodeServer");

          locator.outSpatialReference = map.spatialReference;

++++++++++++++++++++++++++++++++++++

//paras for locator. The address attributes coud be found in server side //geocoder’s locator

          var optionsFrom = {

              //address:{"SingleLine":dojo.byId("fromTxf").value},

              //outFields:["Loc\_name"]

              address: { "SingleKey": dojo.byId("fromTxf").value },

              outFields: ["\*"]

          }

var fromAddress = locator.addressToLocations(optionsFrom);//find the location

  var dList = new dojo.DeferredList([fromAddress]);

          dList.then(function(result){

                            var fromStop = getCandidate(results[0][1]);

});

## Dijit.Menu

Dojo.connect(map,”onLoad”,function(){

//contextMenu

            ctxMenuForMap = new dijit.Menu({

                onOpen: function (box) {

                    currentLocation = getMapPointFromMenuPosition(box);

                }

            });

            ctxMenuForMap.addChild(new dijit.MenuItem({//add menuItem

                label: "设为起点",

                onClick: function (evt) {

                    if (graLayerCandidate.graphics.length == 0) { alert("未选中备选站"); return; }

                    var tempFeatureSet = { "features": [] };

                    for (var i = 0; i <= graLayerCandidate.graphics.length - 1; i++) {

                        var feature = graLayerCandidate.graphics[i];

                        tempFeatureSet.features.push(feature);

                    }

                    var formalizedFeatures = formalizeFSAttributes(tempFeatureSet, "o", "");

                    var ulObj = getUlByFeature(formalizedFeatures.features[0]);

                    var addedFeatures = addFeaturesToUlList(formalizedFeatures); //查询结果添加到列表,已经添加的不会再被添加

                    var featureSum = getLifeatureCount(ulObj);

                    updateFeaturesCount(ulObj, featureSum); //更新要素个数

                    for (var i = 0; i <= addedFeatures.length - 1; i++) {

                        var tempGra = new esri.Graphic(addedFeatures[i].geometry, new esri.symbol.PictureMarkerSymbol("src/assets/images/layer.png", 24, 24), addedFeatures[i].attributes, null);

                        graLayerSelectO.add(tempGra);

                    }

                    map.infoWindow.resize(200, 200);

                    map.infoWindow.hide();

                    graLayerCandidate.clear();

                    Navigation("pan");

                } //onclick end

            })); //menu set as start Points End

            ctxMenuForMap.addChild(new dijit.MenuItem({//add menuItem

                label: "设为终点",

                onClick: function (evt) {

                    if (graLayerCandidate.graphics.length == 0) { alert("未选中备选站"); return; }

                    var tempFeatureSet = { "features": [] };

                    for (var i = 0; i <= graLayerCandidate.graphics.length - 1; i++) {

                        var feature = graLayerCandidate.graphics[i];

                        tempFeatureSet.features.push(feature);

                    }

                    var formalizedFeatures = formalizeFSAttributes(tempFeatureSet, "d", "");

                    var ulObj = getUlByFeature(formalizedFeatures.features[0]);

                    var addedFeatures = addFeaturesToUlList(formalizedFeatures); //查询结果添加到列表,已经添加的不会再被添加

                    var featureSum = getLifeatureCount(ulObj);

                    updateFeaturesCount(ulObj, featureSum); //更新要素个数

                    for (var i = 0; i <= addedFeatures.length - 1; i++) {

                        var tempGra = new esri.Graphic(addedFeatures[i].geometry, new esri.symbol.PictureMarkerSymbol("src/assets/images/station\_small.png", 24, 24), addedFeatures[i].attributes, null);

                        graLayerSelectD.add(tempGra);

                    }

                    map.infoWindow.resize(200, 200);

                    map.infoWindow.hide();

                    graLayerCandidate.clear();

                    Navigation("pan");

                } //onclick end

            })); //menu set as end Points End

            ctxMenuForMap.addChild(new dijit.MenuItem({//add menuItem

                label: "清空备选站",

                onClick: function (evt) {

                    graLayerCandidate.clear();

                    Navigation("pan");

                }

            }));

            ctxMenuForMap.bindDomNode(map.container);

            ctxMenuForMap.startup();

            // Helper Methods

            function getMapPointFromMenuPosition(box) {

                var x = box.x, y = box.y;

                switch (box.corner) {

                    case "TR":

                        x += box.w;

                        break;

                    case "BL":

                        y += box.h;

                        break;

                    case "BR":

                        x += box.w;

                        y += box.h;

                        break;

                }

                var screenPoint = new esri.geometry.Point(x - map.position.x, y - map.position.y);

                return map.toMap(screenPoint);

            }

})//end map onLoad

#### 解除右键菜单显示

function selectDistrict() {

    //dojo.byId("map").style.cursor = "pointer";

    ctxMenuForMap.unBindDomNode(map.container);

}

#### MenuItem显隐

ctxMenuForMap.containerNode.childNodes;

ctxMenuForMap.containerNode.childNodes[0].disabled=true;//是否可选

ctxMenuForMap.containerNode.childNodes[0].style.display="block"//显隐

## dgrid.OnDemandGrid

备注：自己实现效率高

#### Download the js

#### require

dojo.require("dgrid.OnDemandGrid");//grid

    dojo.require("dgrid.Selection");

    dojo.require("dojo.store.Memory");

    dojo.require("dojo.\_base.declare");

    dojo.require("dojo.number");

    dojo.require("dojo.parser");

    dojo.require("dojo.domReady!");

#### create grid and fill in data

function createGrid(startStations, destinations) {

    if (startStations.length == 0 || destinations.length == 0) {

        alert("D or O hasn't been selected");

        return;

    }

     window.grid = new (dojo.declare([dgrid.OnDemandGrid, dgrid.Selection]))({

              bufferRows: Infinity, //show all records

              columns: {

                  "OriginName": "发站名",

                  "DestinationName": "到站名", //(columnName and columeTitle)

                  "CheckOutNum": "到站人数"

              }

          }, "grid");//create Grid

……………………………

var data = []; //set Data

   for (var i = 0; i <= validStartStations.length - 1; i++) {//(data for grid)

                  for (var j = 0; j <= validdestiNationFs.length - 1; j++) {

                      data.push({ "OriginName": validStartStations[i].attributes.NAME, "DestinationName": validdestiNationFs[j].attributes.NAME, "CheckOutNum": 1000 });

                  }

              }

              var memStore = new dojo.store.Memory({ data: data });

              window.grid.set("store", memStore);

              window.grid.refresh();

#### clearData

window.grid.set('store',new dojo.store.Memory({data:[]}));

#### singleSelect when clickRow

//添加 id属性，  grid.select(rowID)

//ID不能含中文字符

          grid.on(".dgrid-row:click", function (e) {

              clearRoutes();

              var rowID;

              var orinName;

              var destName;

              if (navigator.appName == "Netscape") {//FF

                  rowID = this.childNodes[0].childNodes[0].innerHTML;

                  orinName = this.childNodes[0].childNodes[1].innerHTML;

                  destName = this.childNodes[0].childNodes[2].innerHTML;

              }

              else { //IE

                  var tempData = this.innerText.split("\r\n\r\n");

                  rowID = dojo.trim(tempData[0]);

                  orinName = dojo.trim(tempData[1]);

                  destName = dojo.trim(tempData[2]);

              }

              grid.clearSelection();//clear selection

              grid.select(rowID);//select OneRow

          });

#### Get a Row

grid.row(gRowID[0])

移除row:

grid.row(gRowID[0]).remove()

#### Get selectedRow ID ,Clear Selected

Grid.selection

grid.clearSelection();

#### 设置列宽

**How to figure out**: showGrid(), 复制innerHTML,找到列 class规则

添加css限制列宽

/\*grid 列宽,其余部分在js/dgrid/css/dgrid\*/

.dgrid-column-id{ width:50px;}

.dgrid-column-OriginName{ width:80px;}

.dgrid-column-DestinationName{ width:80px;}

.dgrid-column-TransferName{ width:150px;}

.dgrid-column-Directions{ width:400px;}

.dgrid-column-Distance{ width:50px;}

.dgrid-column-StationCount{ width:50px;}

.dgrid-column-CheckOutNum{ width:50px;}

.dgrid-column-ODPartition{ width:50px;}

.dgrid-column-ODTotal{ width:50px;}

.dgrid-column-Price{ width:50px;}

.dgrid-cell {font-size:14px;}

#### Grid三样式

HowTodo:

开着console玩

## Dijit.TitlePane 控制展开/收缩

dijit.byId("dvSearchPart").set("open",false)

dijit.byId("dvSearchPart").set("open",true)

获取状态

dijit.byId("dvSearchPart").get("open")

dijit.byId("dvSearchPart").get("open")

备注：自己实现效率高

## BasemapGallery 地图切换控件(影像、栅格)

**function** createBasemapGallery(){

**var** basemaps=[];

    require(["esri/dijit/BasemapLayer","esri/dijit/Basemap","esri/dijit/BasemapGallery"],

**function**(BasemapLayer,Basemap,BasemapGallery){

**var** streetMap = **new** esri.dijit.Basemap({

                    layers: [**new** esri.dijit.BasemapLayer({

                       url: "http://server.arcgisonline.com/ArcGIS/rest/services/World\_Street\_Map/MapServer"

                    })],

                    id: "StreetMap",

                    title: "Street Map View",

                    thumbnailUrl:"img/lightColor.jpg"

                 });

                 basemaps.push(streetMap);

**var** satelliteMap = **new** esri.dijit.Basemap({

                     layers: [**new** esri.dijit.BasemapLayer({

                       url: "http://server.arcgisonline.com/ArcGIS/rest/services/World\_Imagery/MapServer"

                     })],

                    id: "Satellite",

                    title: "Satellite View",

                    thumbnailUrl:"img/sateImage.jpg"

                   });

                 basemaps.push(satelliteMap);

                 basemapGallery = **new** esri.dijit.BasemapGallery({

                   showArcGISBasemaps: **false**,

                   basemaps: basemaps,

                   map: map

                 },"dvBmGallery");

                 basemapGallery.startup();

             }//end para2

       );//end require

    }//end createBmGallery

2.切换地图，专题图一直在

Key: Basemap Gallery with operational layer

**BasemapGallery**

   dojo.require("esri.map");

        dojo.require("dijit.form.Button");

        dojo.require("dijit.layout.BorderContainer");

        dojo.require("dijit.layout.ContentPane");

        dojo.require("esri.dijit.BasemapGallery");

        dojo.require("esri.arcgis.utils");

        var map;

        function init() {

            map = new esri.Map("map");

            createBasemapGallery();

        }

        function createBasemapGallery() {

            //manually create basemaps to add to basemap gallery

            var basemaps = [];

            var waterTemplateLayer = new esri.dijit.BasemapLayer({

                // url: "http://sampleserver1.arcgisonline.com/ArcGIS/rest/services/WaterTemplate/LocalGovernmentInfrastructureBasemap/MapServer"

                url: "http://10.254.53.85:6080/arcgis/rest/services/TF/2012bjsubwayrouteCopy1/ImageServer"

            });

            var waterBasemap = new esri.dijit.Basemap({

                layers: [waterTemplateLayer],

                title: "影像图",

                thumbnailUrl: "src/thumbnails/hydroimage.png"

            });

            basemaps.push(waterBasemap);

            var publicSafetyLayer = new esri.dijit.BasemapLayer({

                // url: "http://sampleserver1.arcgisonline.com/ArcGIS/rest/services/PublicSafety/PublicSafetyBasemap/MapServer"

                url: "http://10.254.53.85:6080/arcgis/rest/services/TF/baseMap/MapServer"

            });

            var publicSafetyBasemap = new esri.dijit.Basemap({

                layers: [publicSafetyLayer],

                title: "行政图",

                thumbnailUrl: "src/thumbnails/district.png"

            });

            basemaps.push(publicSafetyBasemap);

            var basemapGallery = new esri.dijit.BasemapGallery({

                showArcGISBasemaps: false,

                basemaps: basemaps,

                map: map

            }, "basemapGallery");

            var selectionHandler = dojo.connect(basemapGallery, "onSelectionChange", function () {

                dojo.disconnect(selectionHandler);

                //add the esri population layer to the map

                var stationLayer = new esri.layers.ArcGISDynamicMapServiceLayer("http://10.254.53.85:6080/arcgis/rest/services/TF/newmap/MapServer");

                map.addLayer(stationLayer);

            });//选择完成之后，加载地铁站

            basemapGallery.startup();

            dojo.connect(basemapGallery, "onError", function (error) { console.log(error) });

        }

        dojo.ready(init);

备注：自己实现效率高

**HTML CSS**

<div class="switchBaseMapWidget" data-dojo-type="dijit.TitlePane" data-dojo-props="title:'切换',open:false">

             <div id="basemapGallery" class="dijitTitlePaneContentInner" role="region" aria-hidden="true" aria-labelledby="titlePane\_titleBarNode" data-dojo-attach-point="containerNode"></div>

           </div><!--switchBaseMapWidget-->

.switchBaseMapWidget{ position:absolute; z-index:2; left:205px;}/\*底图切换控件\*/

#basemapGallery{ width:120px;}/\*底图切换控件\*/

# 军规

## 界面，功能是产品，不是作业

   没做的东西，就没做。

   做好的东西，要给做完的样子

## 减少参数到1

  当函数不面向界面，

  减少参数：把多个参数，封装成对象的属性。传值对象。

## 只调用1次的函数 不要独立

  如果函数被调用次数只有1次，那就直接作为代码片段。

## 清空挂载在DOM元素上的对象、属性

dojo.byId("byAzimuthStart").removeAttribute(feature);

备注：方法，属性，对象在DOM中都是 Attribute

举例1：

   setAttribute(“disabled”,”disabled”); //控制是否可以点击

   removeAttibute(“disabled”);

举例2：

    dojo.byId("byAzimuthStart").removeAttribute(feature)//删除绑定在DOM上的要素

## 参数是出错的主要原因

## 路径分析 启示

à1不要做：

正则提取站数组

美化左侧面板

Graphic提示

 à2要做：

  数组的deferlist查询

à3如何做

   把语句集中在一个函数中  query,自己写

   Deferlist,自己写