|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| 二维地图 | | | | | | |
| 操作接口帮助文档 | | | | | | |
|  | | | | | | |
| **Shenjie** | | | | | | |
|  | | | | | | |
| **文件名称** | | 二维地图操作接口API\_V2.0 | | |  |  |
| **版本历史** | | | | | | |
| **版本号** | **时间** | | **作者** | **说明** | | |
| V2.0 | 2015/2/5 | | 申杰 | 重新整理文档 | | |
|  |  | |  |  | | |

|  |
| --- |
|  |

简 介

二维地图提供一系列和地图进行交互的JS操作类接口，主要包括位置操作类接口、图层操作类接口、图形绘制类接口、地图缩放接口、地图坐标操作类接口、标绘操作类接口、地图权限操作类接口、地图监听类接口、地图信息窗操作类接口、工具操作类接口、Geoserver操作类接口及其他操作类接口等。通过使用这些接口，可以更加方便地对地图进行操作。

目 录

[简 介 2](#_Toc411440023)

[1. 位置操作类接口 8](#_Toc411440024)

[panDown 8](#_Toc411440025)

[地图向下平移，系统将地图向下平移1/4地图高度 8](#_Toc411440026)

[panUp 8](#_Toc411440027)

[地图向上平移，系统将地图向上平移1/4地图高度 8](#_Toc411440028)

[panRight 8](#_Toc411440029)

[地图向右平移，系统将地图向右平移1/4地图宽度 8](#_Toc411440030)

[panLeft 8](#_Toc411440031)

[地图向左平移，系统将地图向左平移1/4地图宽度 8](#_Toc411440032)

[centerAt(mapX,mapY) 8](#_Toc411440033)

[地图居中显示 8](#_Toc411440034)

[2. 图层操作类接口 8](#_Toc411440035)

[refreshLayerByID(layerId) 8](#_Toc411440036)

[根据图层ID刷新图层 8](#_Toc411440037)

[refreshLayerByName(layerName) 8](#_Toc411440038)

[根据图层名刷新图层 8](#_Toc411440039)

[removeAllLayers 9](#_Toc411440040)

[移除所有图层 9](#_Toc411440041)

[layerVisibleSwitch(layerID,layerVisible) 9](#_Toc411440042)

[设置图层是否可见 9](#_Toc411440043)

[getLayerVisible(layerID,callback) 9](#_Toc411440044)

[根据图层ID查看该图层的可见性 9](#_Toc411440045)

[clearGraphicsLayer(layerID) 9](#_Toc411440046)

[根据图层ID清空图形 9](#_Toc411440047)

[setPointLayerRenderer(layerID,markerSymbolObject) 9](#_Toc411440048)

[根据图层ID及标记符号对象渲染点层 9](#_Toc411440049)

[setPointLayerPictureRenderer(layerID,pictureSymbolObject) 9](#_Toc411440050)

[根据图层ID及图片符号对象渲染点层 9](#_Toc411440051)

[setLineLayerRenderer(layerID,lineSymbolObject) 9](#_Toc411440052)

[根据图层ID及标记符号对象渲染线层 9](#_Toc411440053)

[setPolygonLayerRenderer(layerID,fillSymbolObject) 10](#_Toc411440054)

[根据图层ID及标记符号对象渲染面层 10](#_Toc411440055)

[3. 图形绘制类接口 10](#_Toc411440056)

[flashPoints(pointString,delay,repeatCount,markSymbolObject) 10](#_Toc411440057)

[绘制点并进行闪烁 10](#_Toc411440058)

[flashPolyline(polylineString,delay,repeatCount,lineSymbolObject) 10](#_Toc411440059)

[绘制线并进行闪烁 10](#_Toc411440060)

[flashPolygon(polygonString,delay,repeatCount,fillSymbolObject) 10](#_Toc411440061)

[绘制面并进行闪烁 10](#_Toc411440062)

[flashGraphic(graphicID,geometryObject,symbolObject,delay,repeatCount,tipMode,attribute,tipClass,tipWidth,tipHeight,tipUrl,autoZoom) 10](#_Toc411440063)

[闪烁标记，闪烁后支持鼠标交互行为 10](#_Toc411440064)

[addPointGraphics(pointString,markerSymbolObject) 11](#_Toc411440065)

[在地图上添加点图形 11](#_Toc411440066)

[addPictureGraphics(pointString,pictureSymbolObject) 11](#_Toc411440067)

[在地图上添加点图片图形 11](#_Toc411440068)

[addPointInfoWindow(id,x,y,markerSymbolObject,mode,title,simpleHtml,width,height,xoff,yoff,layerID) 11](#_Toc411440069)

[绘制点并添加信息窗 11](#_Toc411440070)

[addPolylineGraphics(polylineString,lineSymbolObject) 12](#_Toc411440071)

[在地图上添加线图形 12](#_Toc411440072)

[addPolygonGraphics(polygonString,fillSymbolObject) 12](#_Toc411440073)

[在地图上添加面图形 12](#_Toc411440074)

[addPolygonGraphicInfoWindow(polygonString,fillSymbolObject,mode,title,simpleHtml,width,height) 12](#_Toc411440075)

[在地图上绘制面并添加信息窗 12](#_Toc411440076)

[addGraphic(id,geometryObject,tipMode,attribute,symbolObject,tipClass,tipWidth,tipHeight,tipUrl,autoZoom,layerID) 12](#_Toc411440077)

[绘制图形统一接口 12](#_Toc411440078)

[addGraphics(ids,geometryObjects,tipModes,attributes,symbolObjects,tipClasses,tipWidths,tipHeights,tipUrls,autoZoom,layerID) 13](#_Toc411440079)

[专题绘制 13](#_Toc411440080)

[addCircle(center,radius,callback,centerMarkerSymbol,symbolObject,autoZoom,layerID) 13](#_Toc411440081)

[在地图上添加圆图形标记 13](#_Toc411440082)

[editPolyline(graphicID,polylineJson) 13](#_Toc411440083)

[在地图上画一个可编辑的线（已作废，改用editGraphic） 13](#_Toc411440084)

[editPolygon(graphicID,polygonJson) 14](#_Toc411440085)

[在地图上画一个可编辑的面（已作废，改用editGraphic） 14](#_Toc411440086)

[editGraphic(graphicID,geoJson) 14](#_Toc411440087)

[编辑图形 14](#_Toc411440088)

[drawGraphic(graphicID,drawType,callback) 14](#_Toc411440089)

[鼠标绘制图形 14](#_Toc411440090)

[drawExtent(callback) 14](#_Toc411440091)

[利用鼠标拖拽显示一个矩形框 14](#_Toc411440092)

[animatePolylineInTime(polylineString,lineSymbolObject,duration) 14](#_Toc411440093)

[在地图上按时间画线 14](#_Toc411440094)

[animatePolylineInSpeed (polylineString,lineSymbolObject,speed,ratio) 14](#_Toc411440095)

[在地图上按速度画线 14](#_Toc411440096)

[animateArrowLineInTime(polylineString,arrowlineSymbolObject,duration,layerID) 15](#_Toc411440097)

[在地图上按时间动画箭头线 15](#_Toc411440098)

[animateArrowLineInSpeed(polylineString,arrowlineSymbolObject,speed,ratio) 15](#_Toc411440099)

[在地图上按时间动画箭头线（图层为AnimationLayer） 15](#_Toc411440100)

[drawGraphicByErisGeoJson(graphicID,drawType,geoJson) 15](#_Toc411440101)

[根据ArcGIS Geo Json绘图（不推荐使用） 15](#_Toc411440102)

[getGraphic(graphicID,callback) 15](#_Toc411440103)

[根据图形ID获取图形 15](#_Toc411440104)

[getEditGraphic(graphicID,callback)----原getEidtGraphic 15](#_Toc411440105)

[根据图形ID获取编辑图形 15](#_Toc411440106)

[clearAllGraphics 15](#_Toc411440107)

[清除所有图形 15](#_Toc411440108)

[removeGraphics(ids,layerID) 16](#_Toc411440109)

[清除指定图层上的标记，用逗号隔开；或用\*号表示删除图层上的所有标记 16](#_Toc411440110)

[drawRoute(graphicID,callback) 16](#_Toc411440111)

[绘制路由 16](#_Toc411440112)

[editRoute(graphicID,routeObject,callback) 16](#_Toc411440113)

[编辑路由 16](#_Toc411440114)

[getRoute(graphicID, callback) 16](#_Toc411440115)

[获取路由坐标 16](#_Toc411440116)

[locateRoute(graphicID,routeObject,callback) 16](#_Toc411440117)

[定位路由 16](#_Toc411440118)

[getRouteCN(graphicID, callback) 16](#_Toc411440119)

[获取路由坐标，解决中文乱码问题 16](#_Toc411440120)

[4. 地图缩放类接口 17](#_Toc411440121)

[setLevel(level) 17](#_Toc411440122)

[设置地图级别 17](#_Toc411440123)

[levelUp 17](#_Toc411440124)

[提示当前地图级别为更高一级 17](#_Toc411440125)

[levelDown 17](#_Toc411440126)

[降低当前地图级别为更低一级 17](#_Toc411440127)

[zoomIn 17](#_Toc411440128)

[地图放大，提升当前地图级别为更高一级 17](#_Toc411440129)

[zoomOut 17](#_Toc411440130)

[地图缩小，降低当前地图级别为更低一级 17](#_Toc411440131)

[zoomAt(level,x,y) 17](#_Toc411440132)

[地图以指定点为中心缩放到指定级别 17](#_Toc411440133)

[zoomGeometrys(geometryJsons) 17](#_Toc411440134)

[根据几何图形数组缩放地图 17](#_Toc411440135)

[getMapLevel(callback) 17](#_Toc411440136)

[获取当前地图级别 17](#_Toc411440137)

[getMapMaxLevel 18](#_Toc411440138)

[获取地图最大级别 18](#_Toc411440139)

[zoomExtent( xmin,ymin,xmax,ymax,callback) 18](#_Toc411440140)

[将地图定位到指定范围 18](#_Toc411440141)

[5. 地图坐标操作类接口 18](#_Toc411440142)

[getXY(callback) 18](#_Toc411440143)

[获取地图上所选点的坐标 18](#_Toc411440144)

[getMapCenter 18](#_Toc411440145)

[获取地图中心点坐标 18](#_Toc411440146)

[getMapExtent 18](#_Toc411440147)

[方法描述：获取地图范围坐标 18](#_Toc411440148)

[getCircle(center,radius,pointSize) 18](#_Toc411440149)

[根据圆中心点坐标及半径得到圆的坐标串 18](#_Toc411440150)

[getCenterLocation(strGeo) 18](#_Toc411440151)

[获取几何中心点 18](#_Toc411440152)

[6. 标绘操作类接口 19](#_Toc411440153)

[openAndClosePlotToolbar 19](#_Toc411440154)

[打开和关闭标绘工具条 19](#_Toc411440155)

[loadPlotToolbar(graphicID,graphicinfo,name,callback) 19](#_Toc411440156)

[向地图中加载图形，对应标绘工具条的加载按钮 19](#_Toc411440157)

[7. 地图权限操作类接口 19](#_Toc411440158)

[setVerifiedLayer(layerIds) 19](#_Toc411440159)

[设置用户权限图层 19](#_Toc411440160)

[8. 地图监听类接口 19](#_Toc411440161)

[addMapLoadedEventListener (callback) 19](#_Toc411440162)

[添加地图加载事件 19](#_Toc411440163)

[removeMapLoadedEventListener 19](#_Toc411440164)

[移除地图加载事件监听 19](#_Toc411440165)

[addZoomEventListener(callback) 19](#_Toc411440166)

[添加地图缩放事件监听 19](#_Toc411440167)

[removeZoomEventListener 20](#_Toc411440168)

[移除地图缩放事件监听 20](#_Toc411440169)

[addExtentEventListener(callback) 20](#_Toc411440170)

[添加地图范围事件监听 20](#_Toc411440171)

[removeExtentEventListener 20](#_Toc411440172)

[移除地图范围事件监听 20](#_Toc411440173)

[9. 地图信息窗操作类接口 20](#_Toc411440174)

[showInfoWindow(x,y,title,simpleHtml,width,height) 20](#_Toc411440175)

[地图上显示信息窗 20](#_Toc411440176)

[10. 工具操作类接口 20](#_Toc411440177)

[mapDownload(xmin,ymin,xmax,ymax) 20](#_Toc411440178)

[根据传入的地图范围保存地图 20](#_Toc411440179)

[printMap(title,subtitle) 20](#_Toc411440180)

[交互式打印地图 20](#_Toc411440181)

[getUTMCoordinate(longitude,latitude) 21](#_Toc411440182)

[根据经纬度坐标获取投影坐标（UTM投影法） 21](#_Toc411440183)

[createMenuList(id,x,y,attribute,flag,type) 21](#_Toc411440184)

[创建菜单 21](#_Toc411440185)

[removeMenuList(id) 21](#_Toc411440186)

[根据菜单ID移除菜单 21](#_Toc411440187)

[11. Geoserver操作类接口 21](#_Toc411440188)

[getGeometryAttributeByPolygon (url,tppeName,outputFields,polygonStr,callbackFunc) 21](#_Toc411440189)

[访问Geoserver服务，进行面查询 21](#_Toc411440190)

[getGeometryAttributeById (url,tppeName,id,outputFields,callbackFunc) 21](#_Toc411440191)

[方法描述：访问Geoserver服务，根据ID查询对象 21](#_Toc411440192)

[12. 其他操作类接口 22](#_Toc411440193)

[startGeoNetworkAnalysis(x,y,tolerance,elementType,callback) 22](#_Toc411440194)

[启动GeometryNetworkAnalysis（目前供停气分析使用） 22](#_Toc411440195)

[startGeoNetworkAnalysisWidthMap(callback) 22](#_Toc411440196)

[从地图上取点启动GeoNetwork 22](#_Toc411440197)

[loadPlaceName(placeName) 22](#_Toc411440198)

[设置地名信息对话框中地名信息 22](#_Toc411440199)

[loadSchemeName(schemeName) 22](#_Toc411440200)

[加载方案名称 22](#_Toc411440201)

[loadBookmark(bookmark) 22](#_Toc411440202)

[加载地图书签 22](#_Toc411440203)

[loadBookmarkList 22](#_Toc411440204)

[加载地图书签列表 22](#_Toc411440205)

[13. 附录 23](#_Toc411440206)

[13.1. Geometry Object 23](#_Toc411440207)

[13.2. Symbol Object 24](#_Toc411440208)

[13.3. Feature Object 27](#_Toc411440209)

**二维地图接口说明**

# 位置操作类接口

## panDown

地图向下平移，系统将地图向下平移1/4地图高度

## panUp

地图向上平移，系统将地图向上平移1/4地图高度

## panRight

地图向右平移，系统将地图向右平移1/4地图宽度

## panLeft

地图向左平移，系统将地图向左平移1/4地图宽度

## centerAt(mapX,mapY)

地图居中显示

参数描述：mapX 地图中心点X坐标

mapY 地图中心点Y坐标

向下平移，调用接口panUp()，向上平移，调用接口panDown()------mapviewer底层问题。

# 图层操作类接口

## refreshLayerByID(layerId)

根据图层ID刷新图层

参数描述：layerId 图层ID

## refreshLayerByName(layerName)

根据图层名刷新图层

参数描述：layerName 图层名

## removeAllLayers

移除所有图层

## layerVisibleSwitch(layerID,layerVisible)

设置图层是否可见

参数描述：layerID 图层ID

layerVisible 是否可见（true:可见|false:不可见）

## getLayerVisible(layerID,callback)

根据图层ID查看该图层的可见性

参数描述：layerID 图层ID

callback 回调函数

## clearGraphicsLayer(layerID)

根据图层ID清空图形

参数描述：layerID 图层ID

## setPointLayerRenderer(layerID,markerSymbolObject)

根据图层ID及标记符号对象渲染点层

参数描述：layerID 图层ID

markerSymbolObject 标记符号对象

## setPointLayerPictureRenderer(layerID,pictureSymbolObject)

根据图层ID及图片符号对象渲染点层

参数描述：layerID 图层ID

pictureSymbolObject 图片标记符号对象

## setLineLayerRenderer(layerID,lineSymbolObject)

根据图层ID及标记符号对象渲染线层

参数描述：layerID 图层ID

lineSymbolObject 线标记符号对象

## setPolygonLayerRenderer(layerID,fillSymbolObject)

根据图层ID及标记符号对象渲染面层

参数描述：layerID 图层ID

fillSymbolObject 面填充标记符号对象

# 图形绘制类接口

## flashPoints(pointString,delay,repeatCount,markSymbolObject)

绘制点并进行闪烁

参数描述：pointString 点字符串

delay 闪烁间隔（毫秒）

repeatCount 闪烁次数

markSymbolObject 标记符号对象

## flashPolyline(polylineString,delay,repeatCount,lineSymbolObject)

绘制线并进行闪烁

参数描述：polylineString 线字符串

delay 闪烁间隔（毫秒间隔）

repeatCount 闪烁次数

lineSymbolObject 线标记符号对象

## flashPolygon(polygonString,delay,repeatCount,fillSymbolObject)

绘制面并进行闪烁

参数描述：polygonString 面字符串

delay 闪烁间隔（毫秒）

repeatCount 闪烁次数

fillSymbolObject 面填充标记符号对象

## flashGraphic(graphicID,geometryObject,symbolObject,delay,repeatCount,tipMode,attribute,tipClass,tipWidth,tipHeight,tipUrl,autoZoom)

闪烁标记，闪烁后支持鼠标交互行为

必填参数：graphicID 图形ID

geometryObject 图形几何JSON对象

可选参数：symbolObject 图形样式JSON对象

delay 闪烁间隔（毫秒）

repeatCount 闪烁次数

tipMode tip显示模式（none|open|hover|click，默认为none）

attribute 属性JSON对象，可为空

tipClass tip类型（IFrame|none，默认为none）

tipWidth tip框宽度，默认为200

tipHeight tip框高度，默认为200

tipUrl 当tipClass为IFrame时，IFrame地址

autoZoom 是否自动缩放，默认为true

## addPointGraphics(pointString,markerSymbolObject)

在地图上添加点图形

参数描述：pointString 点字符串

markerSymbolObject 标记符号对象

## addPictureGraphics(pointString,pictureSymbolObject)

在地图上添加点图片图形

参数描述：pointString 点字符串

pictureSymbolObject 图片标记符号对象

## addPointInfoWindow(id,x,y,markerSymbolObject,mode,title,simpleHtml,width,height,xoff,yoff,layerID)

绘制点并添加信息窗

必填参数：id 图形ID

x 图形x坐标

y 图形y坐标

markerSymbolObject 标记符号对象

mode 信息窗弹出模式 （open|click|hover|doubleclick）

title 信息窗标题

simpleHtml信息窗的内容（仅支持简单 HTML，即文本、图片、链接等）

可选参数：width 信息窗宽度

height 信息窗高度

xoff

yoff

layerID 图层ID，默认为HightlightLayer

## addPolylineGraphics(polylineString,lineSymbolObject)

在地图上添加线图形

参数描述：polyllineString 线字符串

lineSymbolObject 线标记符号对象

## addPolygonGraphics(polygonString,fillSymbolObject)

在地图上添加面图形

参数描述：polygonString 面字符串

fillSymbolObject 面填充标记符号对象

## addPolygonGraphicInfoWindow(polygonString,fillSymbolObject,mode,title,simpleHtml,width,height)

在地图上绘制面并添加信息窗

参数描述：polygonString 面字符串

fillSymbolObject 面填充标记符号对象

mode 信息窗弹出模式（open|click|hover|doubleclick）

title 信息窗标题

simpleHtml信息窗的内容（仅支持简单 HTML，即文本、图片、链接等）

width 信息窗宽度

height 信息窗高度

## addGraphic(id,geometryObject,tipMode,attribute,symbolObject,tipClass,tipWidth,tipHeight,tipUrl,autoZoom,layerID)

绘制图形统一接口

必填参数：

id 图形ID

geometryObject 图形几何JSON对象

可选参数：

tipMode tip显示模式（none|open|hover|click，默认为none）

attribute 属性JSON对象，默认为空

symbolObject 图形样式JSON对象，默认为空

tipClass tip类型（IFrame|simpleHtml|none，默认为none）

tipWidth tip框宽度，默认为200

tipHeight tip框高度，默认为200

tipUrl 当tipClass为IFrame时，IFrame地址

autoZoom 是否自动缩放，默认为false（true|false）

layerID 在指定图层上添加标记

当tipClass为simpleHtml时，attribute中应包含simpleHtml

## addGraphics(ids,geometryObjects,tipModes,attributes,symbolObjects,tipClasses,tipWidths,tipHeights,tipUrls,autoZoom,layerID)

专题绘制

必填参数：

ids 图形ID数组

geometryObjects 图形几何JSON对象数组

可选参数：

tipModes tip显示模式数组（none|open|hover|click，默认为none）

attributes 属性JSON对象数组，默认为空

symbolObjects 图形样式JSON对象数组，默认为空

tipClasses tip类型数组（IFrame|simpleHtml|none，默认为none）

tipWidths tip框宽度数组，默认为200

tipHeights tip框高度数组，默认为200

tipUrls 当tipClass为IFrame时，IFrame地址

autoZooms 是否自动缩放数组，默认为false（true|false）

layerID 在指定图层上添加标记，默认为HightlightLayer

## addCircle(center,radius,callback,centerMarkerSymbol,symbolObject,autoZoom,layerID)

在地图上添加圆图形标记

必填参数：center 中心点坐标对象

radius 半径（米）

可选参数：

callback 回调函数，返回polygonJSON字符串，默认为空

centerMarkerSymbol 圆点标记样式符号对象，默认为空

symbolObject 图形样式符号对象，默认为空

autoZoom 是否自动缩放，默认为false（true|false）

layerID 在指定图层上添加图形标记

## editPolyline(graphicID,polylineJson)

在地图上画一个可编辑的线（已作废，改用editGraphic）

参数描述：graphicID 图形ID

polyginJson 线坐标串对象

## editPolygon(graphicID,polygonJson)

在地图上画一个可编辑的面（已作废，改用editGraphic）

参数描述：graphicID 图形ID

polygonJson 面坐标串对象

## editGraphic(graphicID,geoJson)

编辑图形

参数描述：graphicID 图形ID

geoJSON Polyline或Polygon Json对象

## drawGraphic(graphicID,drawType,callback)

鼠标绘制图形

必填参数：graphicID 图形ID

drawType 可绘制图形类型（freehandpolyline|freehandpolygon|polygon|polyline|extent|mappoint|circle）

可选参数：callback 回调函数，返回图形JSON字符串

## drawExtent(callback)

利用鼠标拖拽显示一个矩形框

参数描述：callback 回调函数，返回矩形框坐标串

## animatePolylineInTime(polylineString,lineSymbolObject,duration)

在地图上按时间画线

参数描述：polylineString 线字符串

lineSymbolObject 线符号对象

duration 持续时间（毫秒）

## animatePolylineInSpeed (polylineString,lineSymbolObject,speed,ratio)

在地图上按速度画线

参数描述：polylineString 线字符串

lineSymbolObject 线符号对象

speed 时速（每小时移动多少地图长度）

ratio 真实动画比，实际1小时相当于多少毫秒动画时间

## animateArrowLineInTime(polylineString,arrowlineSymbolObject,duration,layerID)

在地图上按时间动画箭头线

参数描述：polylineString 线字符串

arrowlineSymbolObject 箭头线符号对象

duration 持续时间（毫秒）

layerID 在指定图层上添加图形标记，默认为“HightlightLayer”

## animateArrowLineInSpeed(polylineString,arrowlineSymbolObject,speed,ratio)

在地图上按时间动画箭头线（图层为AnimationLayer）

参数描述：polylineString 线字符串

arrowlineSymbolObject 箭头线符号对象

speed 时速（每小时移动多少地图长度）

ratio 真实动画比，实际1小时相当于多少毫秒动画时间

## drawGraphicByErisGeoJson(graphicID,drawType,geoJson)

根据ArcGIS Geo Json绘图（不推荐使用）

参数描述：graphicID 图形ID

drawType 图形类型（Point/Polyline/Polygon/Multipoint/Envelope）

geoJson ArcGIS Json Geometry Objects

## getGraphic(graphicID,callback)

根据图形ID获取图形

参数描述：graphicID 图形ID

callback 回调函数，返回图形JSON字符串

## getEditGraphic(graphicID,callback)----原getEidtGraphic

根据图形ID获取编辑图形

参数描述：graphicID 图形ID

callback 回调函数，返回图形JSON字符串

## clearAllGraphics

清除所有图形

## removeGraphics(ids,layerID)

清除指定图层上的标记，用逗号隔开；或用\*号表示删除图层上的所有标记

必填参数：ids 图形ID集合字符串

可选参数：layerID 图层ID，默认为HightlightLayer

## drawRoute(graphicID,callback)

绘制路由

必填参数：graphicID 图形ID

可选参数：callback 回调函数，返回路由图形JSON字符串

## editRoute(graphicID,routeObject,callback)

编辑路由

必填参数：graphicID 图形ID

routeObject 路由JSON对象

可选参数：callback 回调函数，返回路由图形JSON字符串

## getRoute(graphicID, callback)

获取路由坐标

参数描述：graphicID 图形ID

callback 回调函数，返回路由图形JSON字符串

## locateRoute(graphicID,routeObject,callback)

定位路由

必填参数：graphicID 图形ID

routeObject 路由JSON对象

可选参数：callback 回调函数，返回空值

## getRouteCN(graphicID, callback)

获取路由坐标，解决中文乱码问题

参数描述：graphicID 图形ID

callback 回调函数，返回路由图形JSON字符串

# 地图缩放类接口

## setLevel(level)

设置地图级别

参数描述：level 地图级别

## levelUp

提示当前地图级别为更高一级

## levelDown

降低当前地图级别为更低一级

## zoomIn

地图放大，提升当前地图级别为更高一级

## zoomOut

地图缩小，降低当前地图级别为更低一级

## zoomAt(level,x,y)

地图以指定点为中心缩放到指定级别

参数描述：level 地图级别

x 中心点x坐标

y 中心点y坐标

## zoomGeometrys(geometryJsons)

根据几何图形数组缩放地图

参数描述：geometryJsons Json几何图形数组

## getMapLevel(callback)

获取当前地图级别

参数描述：callback 回调函数，返回当前地图级别

## getMapMaxLevel

获取地图最大级别

## zoomExtent( xmin,ymin,xmax,ymax,callback)

将地图定位到指定范围

必填参数：xmin 地图最小x

ymin 地图最小y

xmax 地图最大x

ymax 地图最大y

可选参数：callback 回调函数，无返回值

# 地图坐标操作类接口

## getXY(callback)

获取地图上所选点的坐标

参数描述：callback 回调函数，返回坐标字符串

## getMapCenter

获取地图中心点坐标

## getMapExtent

方法描述：获取地图范围坐标

## getCircle(center,radius,pointSize)

根据圆中心点坐标及半径得到圆的坐标串

参数描述：center 中心点坐标 JSON对象

radius 半径（米）

pointSize 点的个数

返回值：圆的坐标串

## getCenterLocation(strGeo)

获取几何中心点

参数描述：strGeo 几何对象JSON字符串

返回值：中心点坐标字符串

# 标绘操作类接口

## openAndClosePlotToolbar

打开和关闭标绘工具条

## loadPlotToolbar(graphicID,graphicinfo,name,callback)

向地图中加载图形，对应标绘工具条的加载按钮

必填参数：graphicID 所要添加的图形ID

graphicInfo 所要添加的图形信息

可选参数：name 图形名称（结合PlotDisplayWidget.swf使用）

callback 回调函数

# 地图权限操作类接口

## setVerifiedLayer(layerIds)

设置用户权限图层

参数描述：layerIds 图层id集合，用逗号隔开

# 地图监听类接口

## addMapLoadedEventListener (callback)

添加地图加载事件

参数描述：callback 回调函数

## removeMapLoadedEventListener

移除地图加载事件监听

## addZoomEventListener(callback)

添加地图缩放事件监听

参数描述：callback 回调函数

## removeZoomEventListener

移除地图缩放事件监听

## addExtentEventListener(callback)

添加地图范围事件监听

参数描述：callback 回调函数

## removeExtentEventListener

移除地图范围事件监听

# 地图信息窗操作类接口

## showInfoWindow(x,y,title,simpleHtml,width,height)

地图上显示信息窗

参数描述：x 显示位置的x坐标

y 显示位置的y坐标

title 信息窗的标题

simpleHtml 信息窗的内容

width 信息窗的宽度，默认为300像素

height 信息窗的高度，默认为200像素

# 工具操作类接口

## mapDownload(xmin,ymin,xmax,ymax)

根据传入的地图范围保存地图

参数描述：xmin 最小x

ymin 最小y

xmax 最大x

ymax 最大y

## printMap(title,subtitle)

交互式打印地图

参数描述：title 正标题

subtitle 副标题

## getUTMCoordinate(longitude,latitude)

根据经纬度坐标获取投影坐标（UTM投影法）

参数描述：longitude 经度

latitude 纬度

返回值：坐标字符串（x:..,y:…）

## createMenuList(id,x,y,attribute,flag,type)

创建菜单

必填参数：id 菜单ID

x 菜单位置x坐标

y 菜单位置y坐标

attribute 菜单属性

可选参数：flag 是否添加收缩按钮

type 菜单类型（list|tree）

## removeMenuList(id)

根据菜单ID移除菜单

参数描述：id 菜单ID

# Geoserver操作类接口

## getGeometryAttributeByPolygon (url,tppeName,outputFields,polygonStr,callbackFunc)

访问Geoserver服务，进行面查询

参数描述：url geoserver服务地址：形如(http://服务器名：端口号//geoserver/wfs)

typeName图层名，形如（ChinaMap:Wellpos）

outputFields 输出字段，用逗号隔开的字符串，默认输出id+outputFields

polygonStr 查询面坐标串，形如（x1,y1 x2,y2 x3,y3 .....x1,y1）

callbackFunc 回调函数，返回JSON对象数组

## getGeometryAttributeById (url,tppeName,id,outputFields,callbackFunc)

方法描述：访问Geoserver服务，根据ID查询对象

参数描述：url geoserver服务地址：形如(http://服务器名：端口号//geoserver/wfs)

typeName图层名，形如（ChinaMap:Wellpos）

id 空间对象ID

outputFields 输出字段，用逗号隔开的字符串，默认输出id+outputFields

callbackFunc 回调函数，返回JSON对象数组

# 其他操作类接口

## startGeoNetworkAnalysis(x,y,tolerance,elementType,callback)

启动GeometryNetworkAnalysis（目前供停气分析使用）

必填参数：x x坐标

y y坐标

可选参数：tolerance 离线容差范围

elementType 分析类型（junction|edge|both）

callback 回调函数

## startGeoNetworkAnalysisWidthMap(callback)

从地图上取点启动GeoNetwork

参数描述：callback 回调函数

## loadPlaceName(placeName)

设置地名信息对话框中地名信息

参数描述：placeName 地名信息，JSON格式

## loadSchemeName(schemeName)

加载方案名称

参数描述：placeName 地名信息，JSON格式

## loadBookmark(bookmark)

加载地图书签

参数描述：bookmark 地名信息，JSON格式

## loadBookmarkList

加载地图书签列表

# 附录

## Geometry Object

### Point

{"x" : <x>, "y" : <y>, "z" : <z>, "m" : <m>, "spatialReference" : {<spatialReference>}}

### MultiPoint

{

"hasM" : true | false,

"hasZ" : true | false,

"points" : [[ <x1>, <y1>, <z1>, <m1> ] , [ <x2>, <y2>, <z2>, <m2> ], ... ],

"spatialReference" : {<spatialReference>}

}

### Polyline

{

"hasZ" : true | false,

"hasM" : true | false,

"paths" : [[[<x11>, <y11>, <z11>, <m11>],...,[<x1N>, <y1N>, <z1N>, <m1N>]],

,...,[[<xk1>, <yk1>, <zk1>, <mk1>],...,[<xkM>, <ykM>, <zkM>, <mkM>]]],

"spatialReference" : {<spatialReference>}

}

### Polygon

{

"hasZ" : true | false,

"hasM" : true | false,

"rings" : [[[<x11>, <y11>, <z11>, <m11>],...,[<x1N>, <y1N>, <z1N>, <m1N>]],

,...,[[<xk1>, <yk1>, <zk1>, <mk1>],...,[<xkM>, <ykM>, <zkM>, <mkM>]]],

"spatialReference" : {<spatialReference>}

}

### Envelope

{

"xmin" : <xmin>, "ymin" : <ymin>,

"xmax" : <xmax>, "ymax" : <ymax>,

"zmin" : <zmin>, "zmax" : <zmax>,

"mmin" : <mmin>, "mmax" : <mmax>,

"spatialReference" : {<spatialReference>}

}

## Symbol Object

### Color

[ <red>, <green>, <blue>, <alpha> ]

### Simple Marker Symbol

{

"type" : "esriSMS",

"style" : "< esriSMSCircle | esriSMSCross | esriSMSDiamond | esriSMSSquare | esriSMSX | esriSMSTriangle >",

"color" : <color>,

"size" : <size>,

"angle" : <angle>,

"xoffset" : <xoffset>,

"yoffset" : <yoffset>,

"outline" : { //if outline has been specified

"color" : <color>,

"width" : <width>

}

}

### Simple Line Symbol

{

"type" : "esriSLS",

"style" : "< esriSLSDash | esriSLSDashDot | esriSLSDashDotDot | esriSLSDot | esriSLSNull | esriSLSSolid >",

"color" : <color>,

"width" : <width>

}

### Simple Fill Symbol

{

"type" : "esriSFS",

"style" : "< esriSFSBackwardDiagonal | esriSFSCross | esriSFSDiagonalCross | esriSFSForwardDiagonal | esriSFSHorizontal | esriSFSNull | esriSFSSolid | esriSFSVertical >",

"color" : <color>,

"outline" : <simpleLineSymbol> //if outline has been specified

}

### Picture Marker Symbol

{

"type" : "esriPMS",

"url" : "<imageUrl>", //Relative URL for static layers and full URL for dynamic layers. Access relative URL using http://<mapservice-url>/<layerId1>/images/<imageUrl11>

"imageData" : "<base64EncodedImageData>",

"contentType" : "<imageContentType>",

"width" : <width>,

"height" : <height>,

"angle" : <angle>,

"xoffset" : <xoffset>,

"yoffset" : <yoffset>

}

### Picture Fill Symbol

{

"type" : "esriPFS",

"url" : "<imageUrl>", //Relative URL for static layers and full URL for dynamic layers. Access relative URL using http://<mapservice-url>/<layerId1>/images/<imageUrl11>

"imageData" : "<base64EncodedImageData>",

"contentType" : "<imageContentType>",

"outline" : <simpleLineSymbol>, //if outline has been specified

"width" : <width>,

"height" : <height>,

"angle" : <angle>,

"xoffset" : <xoffset>,

"yoffset" : <yoffset>,

"xscale": <xscale>,

"yscale": <yscale>

}

### Text Symbol

{

"type" : "esriTS",

"color" : <color>,

"backgroundColor" : <color>,

"borderLineSize" : <size>,

"borderLineColor" : <color>,

"haloSize" : <size>,

"haloColor" : <color>,

"verticalAlignment" : "<baseline | top | middle | bottom>",

"horizontalAlignment" : "<left | right | center | justify>",

"rightToLeft" : <true | false>,

"angle" : <angle>,

"xoffset" : <xoffset>,

"yoffset" : <yoffset>,

"kerning" : <true | false>,

"font" : {

"family" : "<fontFamily>",

"size" : <fontSize>,

"style" : "<italic | normal | oblique>",

"weight" : "<bold | bolder | lighter | normal>",

"decoration" : "<line-through | underline | none>"

},

"text" : "<client-side graphic text>" //only applicable when specified as a client-side graphic.

}

## Feature Object

{

"geometry" : <geometry>,

"attributes" : {

"name1" : <value1>,

"name2" : <value2>,

}

}