

标识: MarsGIS

版本号: V1.6

# MarsGIS for Leaflet 地图平台软件

---

## 开发使用手册

合肥火星科技有限公司

2019 年 9 月 18 日

## 目录

1 前言.....	4
1.1 编写目的.....	4
1.2 项目资料.....	4
2 开发中心服务网站.....	6
2.1 站点部署.....	6
3 底层框架类库.....	7
3.1 类库源码.....	7
3.2 类明细.....	8
4 基础项目.....	9
4.1 代码说明.....	9
4.2 CONFIG.JSON 配置文件说明.....	9
4.2.1 可配置内容.....	9
4.2.2 使用方式.....	9
4.2.3 多种使用模式.....	10
4.3 WIDGET 模块化开发说明.....	10
4.3.1 Widget 配置说明.....	11
4.3.2 如何新增一个 widget 模块.....	11
4.3.3 widget.js 文件介绍.....	12
4.3.4 view.html 界面介绍.....	12
4.3.5 如何外部使用 widget.....	14
4.3.6 widget 之间的通信交互示例.....	15
4.4 快速开始源码.....	16
5 配置项手册.....	23
5.1 MAP 地图配置.....	23
5.1.1 地图控件配置（controls）.....	23
5.1.2 地图图层配置（basemaps 和 operationallayers）.....	24

---

5.2 WIDGET 模块配置.....	33
5.2.1 单个模块配置.....	33
5.2.2 弹窗参考配置.....	34

# 1 前言

## 1.1 编写目的

本手册主要根据“MarsGIS for Leaflet 地图平台软件”建设方案和项目功能进行编写，通过对系统框架的功能及操作的详细说明，指导技术人员能够按照文档正确使用各项功能。

## 1.2 项目资料

交付资料是根据合同购买范围而定，各项交付资料如下表：

序号	软件名称	交付文件名称
1	开发中心服务网站	<a href="#">marsv.rar</a>
2	基础项目 源码	<a href="#">web 基础项目.rar</a>
3	综合态势展示系统源码	<a href="#">web 综合态势展示系统.rar</a>

**项目脚手架：**提供 vue 技术栈下的 mars2d-vue-cli.rar 和传统技术下的 mars2d-web-cli.rar2 个脚手架。

**类库源码：**如果合同有购买 leaflet-mars 类库源码，在上面所有包中 leaflet-mars 目录下包含 leaflet-src.js 源码文件。

**技术咨询服务：**按合同规定服务期限内提供技术咨询服务，有关“平台”的使用及二次开发等相关问题均在第一时间回复解答，最迟不超过 24 小时。

注：后续文档操作说明中所涉及数据均为实验室测试数据。

## 2 开发中心服务网站

### 2.1 站点部署

在目标服务器或电脑中安装好任意 http 服务器，确保可正常提供 Web 信息浏览服务即可。如：IIS、Apache、Tomcat、Nginx、Nodejs 等均可。

将 [marsv.rar](#) 压缩包文件解压，使用 http 服务器发布为站点，发布方式请参考各 http 服务器使用帮助或自行百度。例如使用 Tomcaat 发布的目录结构和最终效果如下图：



后续文档中介绍默认按 <http://localhost:8888/marsv/> 进行介绍说明，实际 url 请以您实际部署为准。

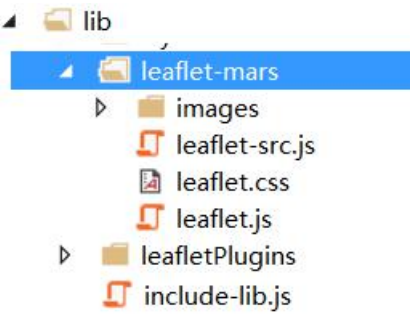
该站点和包括下面内容：

功能名称	说明
基础知识	提供 web 前端、GIS、可视化相关基础知识资料
Leaflet 基础知识	翻译了 Leaflet 官方教程、API 文档、插件介绍等资料，全中文网页文档便于学习和查阅。
MarsGIS 框架	提供平台框架开发使用手册、API 文档、Example 示例（含源码）等
其他	附赠提供其他一些可视化相关资料

3 底层框架类库

3.1 类库源码

框架核心功能编写在 leaflet-mars 类库中，该类库是我们开发并打包编译生产的一个 js 和一个 css 文件，引用这 2 个文件即可使用框架功能(类似 leaflet 的使用,已包含最新版本 leaflet)，这样提高效率、方便使用。



当前项目引用类库使用 include-lib.js 统一配置,默认引用的是压缩版类库,如果需要调试或修改类库,请修改 include-lib.js 中 leaflet.js 为 leaflet-src.js

```
include-lib.js 119 ],
120 'leaflet-mars': [
121     libpath + "/leaflet-mars/leaflet.css",
122     libpath + "/leaflet-mars/leaflet-src.js",
123 ],
```

普通 html 页面中使用时直接按下图引用 leaflet-src.js 文件即可

```
<!--地图lib-->
<link href="../../lib/leaflet-mars/leaflet.css" rel="stylesheet" />
<script src="../../lib/leaflet-mars/leaflet.js"></script>
```

## 3.2 类明细

类库由 3 部分组成，最新版本 Leaflet、一些常用通用 leaflet 插件（draw、MarkerCluster、contextmenu、SimpleGraticule、proj4 等）、火星科技编写扩展的其他类和方法（命名为 L.mars 类下），类库具体包括：

类别	包名或类名	说明
Leaflet 库	支持 Leaflet API 所有类及方法	来源 leaflet 官方, <a href="#">详情</a>
	在 Leaflet 基础类中扩展了一些方法	火星科技开发
Leaflet 插件	Leaflet.draw 绘制点、线、面、矩形、圆。	来源 GitHub, <a href="#">详情</a>
	leaflet-contextmenu 右键菜单	来源 GitHub, <a href="#">详情</a>
	Leaflet.markercluster 标注点聚类图层	来源 GitHub, <a href="#">详情</a>
	L.SimpleGraticule 经纬网图层	来源 GitHub, <a href="#">详情</a>
	proj4leaflet 坐标系	来源 GitHub, <a href="#">详情</a>
	L.tileLayer.baidu 互联网百度地图图层	火星科技开发, <a href="#">详情</a>
	L.tileLayer.tdt 互联网天地图地图图层	火星科技开发
	L.tileLayer.wmts OGC 标准 wmts 地图图层	火星科技开发
	L.tileLayer.arcGISTile 离线 arcgis 格式瓦片图层	火星科技开发
	L.control.toolbar 地图控件（默认区域、定位、全屏）	火星科技开发
平台 类	L.mars 平台地图一系列支撑类	火星科技开发
	L.widget 平台 widget 模块化功能支撑类	火星科技开发

更多细节请参看 API 介绍，API 文档在

开发中心服务网站中：<http://localhost:8888/marsv/?url=leaflet-docs/api.html>

项目源码目录下：</lib/leaflet-mars/doc/index.html>

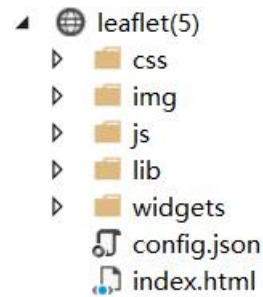


## 4 基础项目

[web 基础项目.rar](#) 基础项目主要提供一个基础项目模版，包含常用基础地图功能，可在该基础项目上快速开发搭建新项目。

### 4.1 代码说明

基础项目是我们需要开发和学习的主要代码，新项目在此基础上开始编码即可，通过代码编辑器打开该项目，其中 `config.json` 为项目图层和模块等相关配置信息，`widgets` 目录为模块功能目录，一般正常开发项目只用修改配置文件和开发新 `widget` 即可。



第三方类库及 `leaflet-mars` 类库存放在 `lib` 目录下，第三方 `lib` 用到了 `jquery`、`bootstrap`、`font-awesome` 字体、`layer` 弹窗、`echarts` 图表等。

基础项目中最重要的是 2 个点是，[config.json 配置信息](#)和 [widget 模块化开发](#)。

### 4.2 Config.json 配置文件说明

框架对地图的初始化进行了封装，通过传入 `json` 配置信息按配置内容对地图进行初始化，因为大部分项目中，地图初始化都是类似的，不同的是一些相关参数和图层不同，我们使用时不用关注内部实现，只用按项目需求修改或使用不同的配置信息即可。

项目配置信息通过 `json` 格式的配置文件配置加载，该配置文件可以是静态 `json` 文件，也可以放在服务端动态根据用户权限动态返回后的数据。

#### 4.2.1 可配置内容

在 `config.json` 文件中，包括 `map` 和 `widget` 共 2 个子节点，其中 `map` 为地图信息配置，`widget` 为模块信息，具体各参数详情参看本文档“[5 配置项手册](#)”章节。

#### 4.2.2 使用方式

用户使用时可以通过 `L.mars.createMap` 方法传入 `url` 参数(`json` 文件路径)或 `data` 参数(服务端返回的 `json` 格式数据)进行地图初始化。

更多细节请参看 API 介绍：

开发中心服务网站中：<http://localhost:8888/marsv/?url=leaflet-docs/api.html>

项目源码目录下：</lib/leaflet-mars/doc/index.html>

如有开发中心服务网站，请参看配置示例页面及其源码：



## 4.2.3 多种使用模式

在同一个项目中不改变代码，可以使用不同配置方式来适应不同场景，例如：

### (1) 不同用户使用不同的配置

在后台中根据登录用户的权限，返回不同参数的 json 配置和图层给前端框架。

[http://localhost:8888/marsv/leaflet-example/editor.html#14\\_for\\_config\\_data](http://localhost:8888/marsv/leaflet-example/editor.html#14_for_config_data)

### (2) 不用修改代码快速切换项目中的地图

我们的项目经常会被 N 次复用，修改的也可能只是地图的初始化中心点，也有可能是将天地图改为百度地图，这样我们可以在不修改任何代码情况下，通过配置多个 json 文件，通过 Url 传参调用不同配置，快速切换地图，例如通过下面方法分别加载天地图、高德和百度底图：

[http://localhost:8888/marsv/leaflet-example/editor.html?config=wgs84.json#14\\_for\\_config\\_url](http://localhost:8888/marsv/leaflet-example/editor.html?config=wgs84.json#14_for_config_url)

[http://localhost:8888/marsv/leaflet-example/editor.html?config=gcj.json#14\\_for\\_config\\_url](http://localhost:8888/marsv/leaflet-example/editor.html?config=gcj.json#14_for_config_url)

[http://localhost:8888/marsv/leaflet-example/editor.html?config=baidu.json#14\\_for\\_config\\_url](http://localhost:8888/marsv/leaflet-example/editor.html?config=baidu.json#14_for_config_url)

## 4.3 Widget 模块化开发说明

框架除了基础的地图初始化外，还有 widget 模块化功能，设计思想主要是借鉴了 gis 行业的 arcgis flexviewer 和 jsviewer，也借鉴了前端行业的 React 和 Vue，

其实最佳方式是整个项目用 React 或 Vue 开发最佳,而为什么用 widget 的方法设计框架的模块化,主要考虑到当前大部分公司项目还是传统方式开发的,而且 widget 迁移到 vue 或 react 也很简单。

widget 方式加载各模块业务功能,框架可按配置自动释放激活 widget 功能,适用于在一张地图上做多业务复杂功能时比较实用。将每个业务模块设计为独立的 widget 模块,互相都是独立的,是按需加载和初始化,默认是不加载的,单击激活某个模块后才会加载其相关 js 和 html 等资源。

widget 基类为 **L.widget.BaseWidget** 类, widget 的统一管理由 **L.widget** 类进行统一处理。

4.3.1 Widget 配置说明

在 config.json 文件中,除了 map 根节点,我们还有个 widget 根节点,这是节点记录了 widget 模块的基础信息。具体配置项目参看“[5.2widget 模块配置](#)”。

插件默认是不会加载的,首次单击后才会加载相关 js 和初始化插件功能。为了方便使用 widget 和不同项目使用不同配置,我们使用一个 json 文件来记录这些配置,配置文件可以单独编写,也可以放在地图配置的 config.json 文件中。页面通过 **L.widget.init**(map, **jsondata.widget**)来传入配置信息进行初始化;

4.3.2 如何新增一个 widget 模块

请参考 widgets\\_example 命名的 3 个示例的新建一个 widget 模块,建议复制后改下目录名称即可。每个模块必须存在 **widget.js** 入口文件,该文件内部定义了一个继承了



BaseWidget 的类,将 js 文件路径按下节说明定义至 widget.json 文件中。

默认每个插件是**无界面或弹窗**的,如果有弹窗或界面请定义 options.view 属性即可。 config.json 配置示例: {**"name": "模板"**, **"uri": "\_example/widget.js"**},

文件	角色	说明	主要参与开发人员
widget.js	一个 js 模块类	编写和 GIS 地图相关的代码	GIS 工程师
view.html	弹窗或界面	编写非地图相关的前端 UI 及代码	非 GIS 工程师

4.3.3 widget.js 文件介绍

widget.js 文件内部定义了一个 js 的类，继承自 BaseWidget，具有下面接口属性和方法。

分类	配置项	说明
options 配置	resources	需要预先加载到主页面的 js、css 资源文件
	view	存在弹窗或界面时配置，有 type,url 属性，详见后节说明
类公共 属性	config	widget.json 中对应 widget 定义的配置信息
	path	当前 widget 所在目录的相对于网页的路径
	map	当前 leaflet 的 map 地图对象
类接口 方法	create	插件初始化触发，仅执行 1 次
	winCreateOK	options.view 定义的界面每次初始化后调用
	activate	打开激活插件 触发
	disable	关闭释放插件 触发

4.3.4 view.html 界面介绍

在 widget.js 中配置 options.view 参数来配置各种界面效果，如弹窗、div 层、按钮等，配置中必须存在 type 和 url 属性，type 标识类型，url 为 html 地址。目前 view 有下面 3 种模式供使用：

【1】如果是 iframe 弹窗时配置（子页面）

当配置 type 是 window 或空时，框架会自动创建一个 iframe 模式的弹窗，请参阅 widgets\\_example\view.html，弹窗中加载配置的 url 页面，该页面是一个独立的 html 页面，可以任意编写相关代码，该页面必须定义一个 initWidgetView 方法，框架会自动调用进行传入当前 widget 对象，用于页面中调用 widget 中的方法属性。

此模式弹窗优势是 widget.js 只写与主页面地图交互的业务内容，view.html 弹窗是一个独立的 html 页面，比较自由，简单粗暴、无任何限制；不用担心 id 命名冲突；可以每个页面用不同的 ui 和第三方插件不用考虑冲突问题；任何水平的开发人员均容易快速开发。

配置示例：{ type: "window", url: "view.html", windowOptions: { width:

```
250, height: 300 } }
```

windowOptions详细配置参看“[5.2.2弹窗参考配置](#)”



## 【2】如果是 div 元素弹窗时配置（同页面）

当配置 type 是 divwindow 时，框架会自动创建一个 div 层到地图主页面中，模拟一个弹窗，此模式只需在 view.html 定义 html 结构即可，js 全部写在 widget.js 中，因为他们是在同一个页面中，可直接互相访问，这种模式弊端是易引起模块间 id 命名冲突，在 css 和 html 中命名时需注意。

配置示例：

```
{ type: "divwindow", url: "view.html", windowOptions: { width: 250, height: 300 } }
```

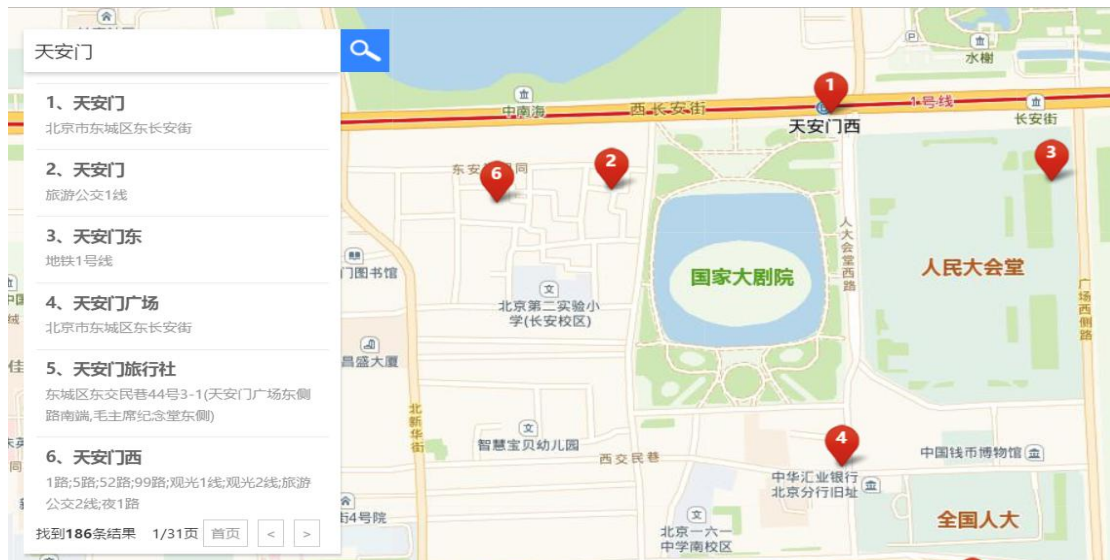




### 【3】如果是 html 元素时（同页面）

当配置 type 是 append 时，框架会自动加载 url 中的 html，append 添加至主页面中 id 为 parent 的节点下，默认不配置时是 body。

配置示例：{type: "append", url: 'querybar.html', parent: 'body'} 其中 parent 为主页面的 dom 元素（如 body）或元素 ID。



#### 4.3.5 如何外部使用 widget

在外部调用 widget 功能，都是通过 `L.widget` 类来管理的。在需要外部使用的地方通过 js 代码加载，如：`L.widget.activate('bookmark')`；其中 `bookmark` 为 `widgets.json` 中 `widgets` 配置的 id 或者 uri 值。如果你要在激活此插件时不释放其他已打开的插件，第 2 个参数传入 `true`，例如：`L.widget.activate('bookmark', true)`；

另外也可以直接传入 config 未配置的 widget，进行使用激活，这样可以将 widgets 分散配置或设置在其他任意地方进行多样化兼容使用。以下示例各种调用方式，效果都一样：

```
L.widget.activate('bookmark'); //widget 中配置有 id 时
L.widget.activate('widgets/bookmark/widget.js'); //直接使用 uri
L.widget.activate({"name": "书签", "uri": "widgets/bookmark/widget.js"}); //config
中未配置任何信息，直接传参调用
```

更多方法请参看 API:

三

BaseWidget 模块基础类

L.widget.BaseWidget 类是平台所有widget功能模块的基础类，开发widget时需要继承该类。该类不是直接new实例化使用的，是新建一个类继承该类后使用。

使用示例

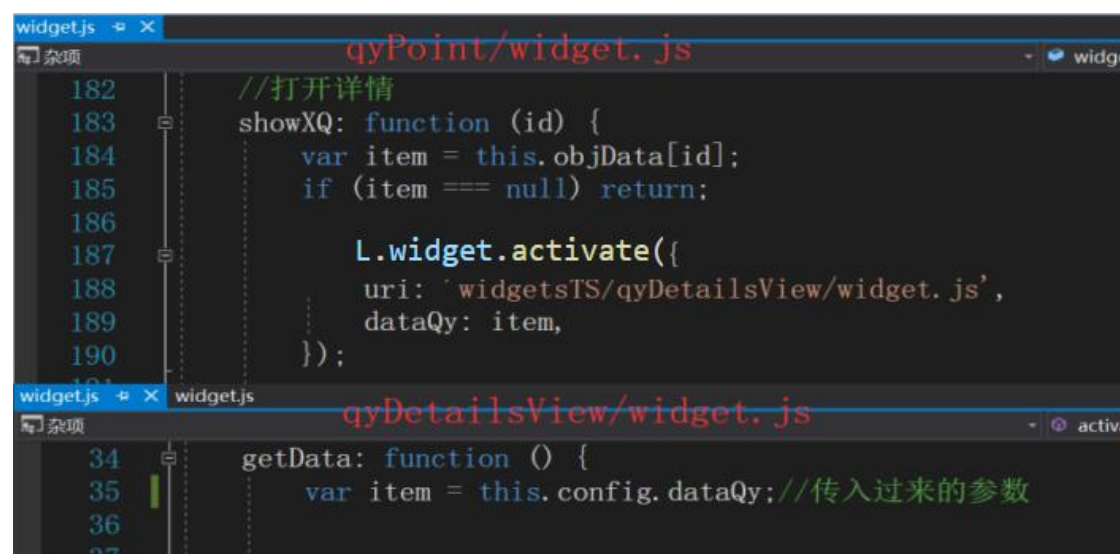
```
var MyWidget = L.widget.BaseWidget.extend({
  map: null, //框架会自动对map赋值
  options: {
    resources: ['style.css'],
    view: {
      type: "window",
      url: "view.html",
      windowOptions: {width: 250, height: 300}
    },
  },
});
```

开发中心服务网站中: <http://localhost:8888/marsv/?url=leaflet-docs/api.html>

项目源码目录下: </lib/leaflet-mars/doc/index.html>

### 4.3.6 widget 之间的通信交互示例

(1) qyPoint 模块中单击地图上的点后，激活弹出详情窗口 qyDetailsView 模块:



(2) roamFly 模块中，动态去更新已打开的 roamChars 模块，更新显示数据。



(2) A、B 两个都是 **iframe 弹窗模式**的 widget 模块，在 A 模块中的 view.html 中需要调用 B 模块的 view.html，其调用流程是下图过程，这里面分 2 个层次，widget.js 都在 index.html 主页面，view.html（含 view.js）是 iframe 子页面：

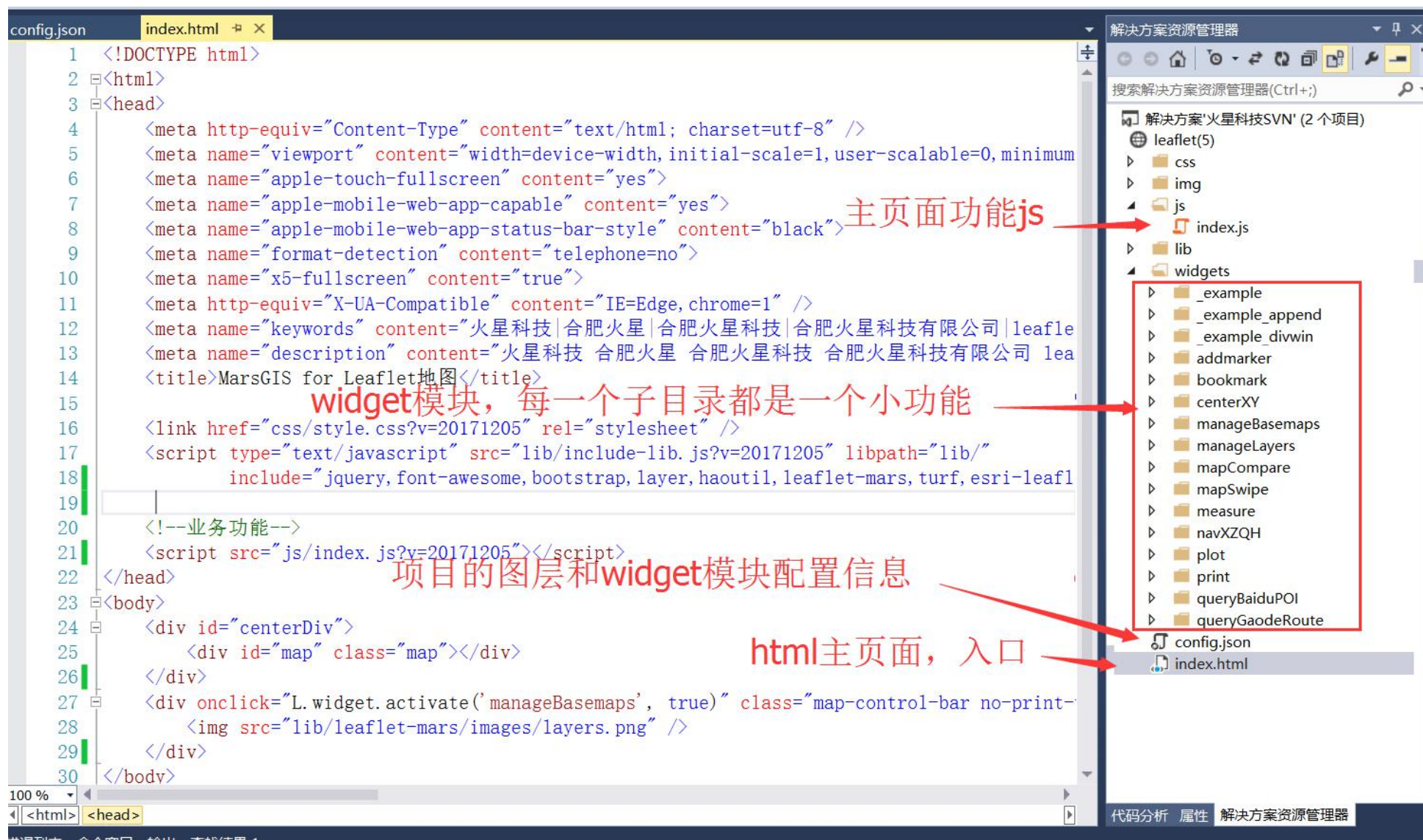


## 4.4 快速开始源码

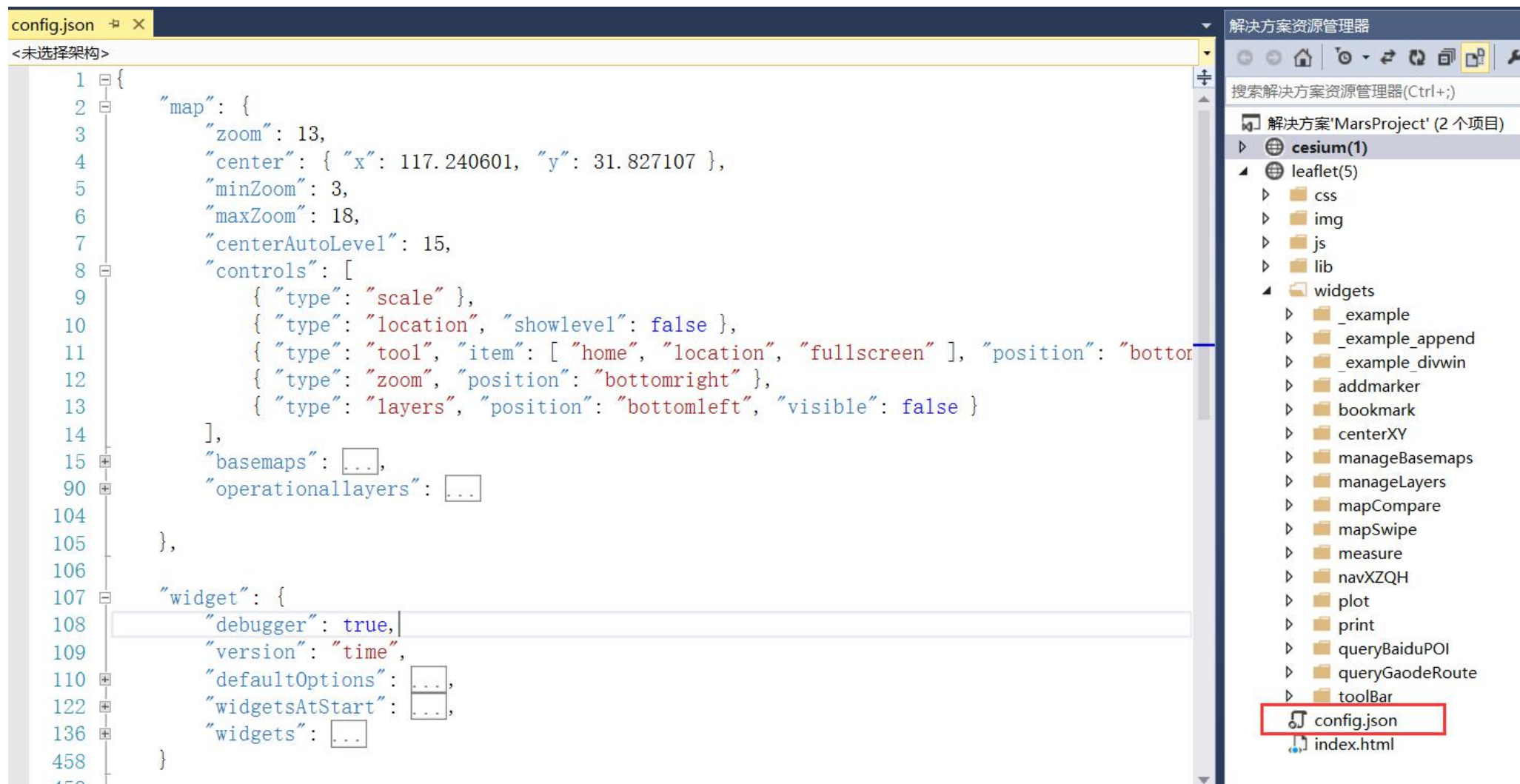
为了让你快速开始读懂源码，下面截图解说，请参考下面步骤进行对照阅读。



打开项目，查看各目录。正常一个项目的开发只用修改 config.json 配置和开发新 widget 即可



查阅本文档“4.2 Config.json 配置文件说明”了解 config.json 文件，参阅“5 配置项手册”了解各参数配置。



打开 widget 测试功能，方便测试，发布的正式版本记得改回 false

The screenshot displays the MarsGIS for Leaflet development environment. The main editor shows the `config.json` file with the following configuration:

```
1 {  
2   "map": ...,  
56  
57   "widget": {  
58     "debugger": true,  
59     "version": "time",  
60     "defaultOptions": {  
61       "windowOptions": {  
62         "position": "rt",  
63         "maxmin": false,  
64         "resize": true  
65       },  
66       "autoDisable": true,  
67       "disableOhter": true  
68     },  
69     "widgetsAtStart": [  
70       {  
71         "name": "百度POI查询",  
72         "uri": "widgets/queryBaiduPOI/widget.js"  
73       },  
74       {  
75         "name": "路线规划",  
76         "uri": "widgets/queryGaodeRoute/widget.js"  
77       }  
78     ],  
79     "widgets": [  
80  
81   ]  
}
```

Annotations in the image include:

- A red arrow pointing to the `"debugger": true` line with the text: 修改为true, 打开widget测试工具栏
- A red arrow pointing to the `config.json` file in the file explorer with the text: 页面效果

The right sidebar shows the file explorer with the following structure:

- 解决方案资源管理器
  - cesium(1)
    - leaflet(5)
      - css
      - img
      - js
        - index.js
      - lib
      - widgets
        - \_example
        - \_example\_append
        - \_example\_divwin
        - addmarker
        - bookmark
        - centerXY
        - manageBasemaps
        - manageLayers
        - mapCompare
        - mapSwipe
        - measure
        - navXZQH
        - plot
        - print
        - queryBaiduPOI
        - queryGaodeRoute

The bottom status bar shows: 代码分析 属性 解决方案资源管理器



查看“POI 兴趣点查询”（queryBaiduPOI）的源码、配置信息、运行效果。

The image displays a development environment with three main components:

- config.json (Left Panel):** Shows the configuration for the "widget". A red box highlights the entry for "百度POI查询" (Baidu POI Query) with the URI "widgets/queryBaiduPOI/widget.js". A red arrow points from the text "config.json中的配置信息" to this box.
- File Explorer (Right Panel):** Shows the project structure. A red box highlights the "queryBaiduPOI" folder, which contains "images", "config.json", "querybar.css", "querybar.html", and "widget.js". A red arrow points from the text "对应的源码" to this box.
- Preview (Bottom Center):** Shows a screenshot of the web application interface. It features a search bar labeled "搜索 地点" (Search Location) and four category buttons: "美食" (Food), "酒店" (Hotel), "景点" (Scenic Spot), and "小区" (Residential Area). Below these are search results for "华地公馆" (Huadi Gongguan) and a "删除历史" (Delete History) button. A red arrow points from the text "页面展示效果" to this preview.

## 查看三个模版 widget 熟悉 widget 机制

The screenshot displays the MarsGIS for Leaflet development interface. The main editor shows the `index.html` file with a configuration object for a widget. The configuration includes a name, type, children array, and window options (width, height, position, maxmin, openAtStart). A red arrow points from the text "模板widget" (Template widget) to the `widgets` directory in the file explorer. Another red arrow points from the text "单击相应菜单，激活测试widget" (Click the corresponding menu to activate the test widget) to the "iframe弹窗模板" (iframe popup template) menu item in the map interface.

```
81 {  
82   "name": "模板",  
83   "type": "group",  
84   "children": [  
85     {  
86       "name": "iframe弹窗模板",  
87       "uri": "widgets/_example/widget.js",  
88       "windowOptions": {  
89         "width": 300,  
90         "height": 400,  
91         "position": {  
92           "top": 10,  
93           "right": 10  
94         },  
95         "maxmin": true,  
96         "require": true  
97       },  
98       "openAtStart": false  
99     }  
100   ]  
101 }
```

模板widget

单击相应菜单，激活测试widget

iframe弹窗模板

div弹窗模板

append模板

widget测试

漫游 模板 地图工具

解决方案资源管理器

搜索解决方案资源管理器(Ctrl+;)

解决方案'火星科技SVN' (2个项目)

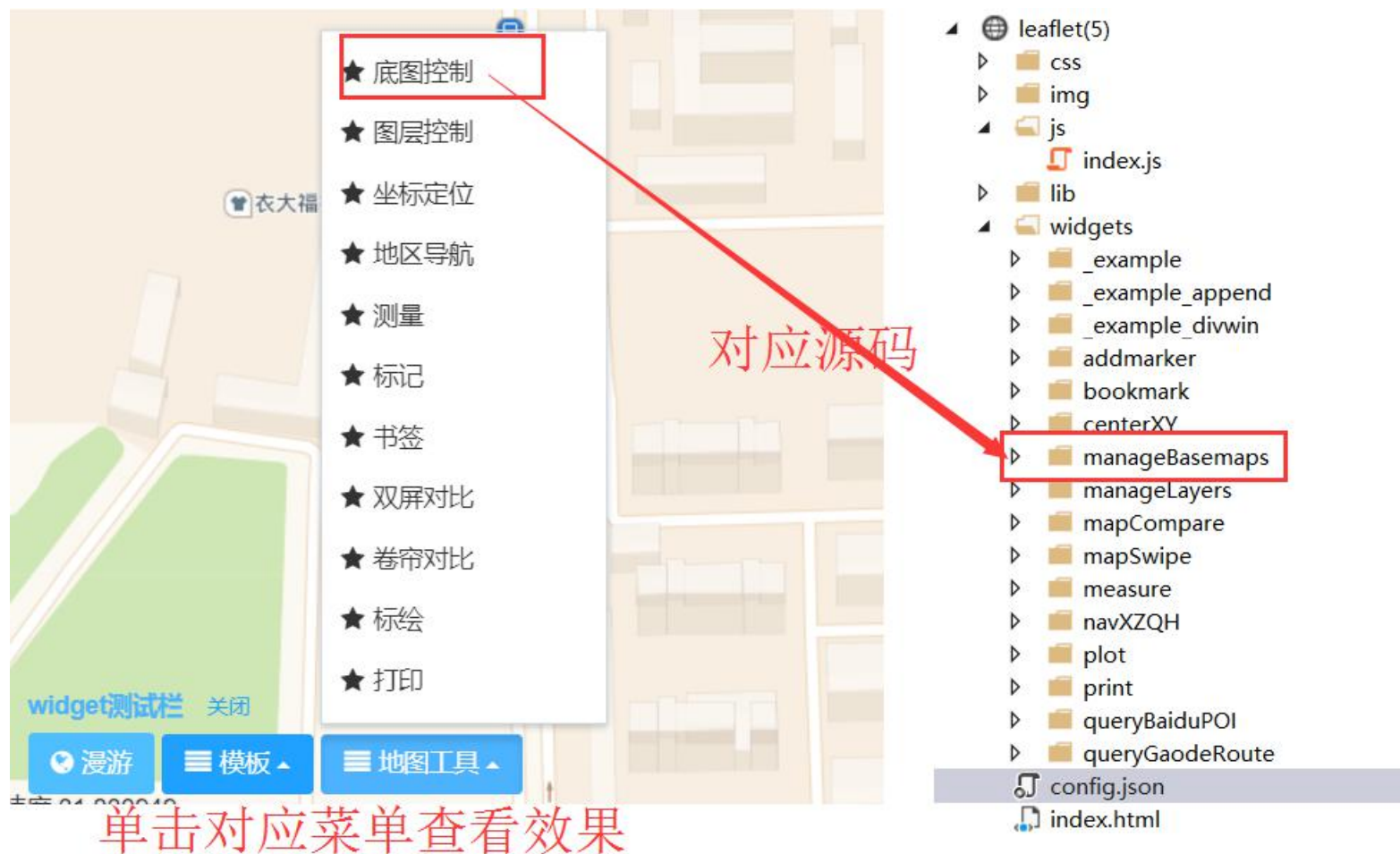
- cesium(1)
- leaflet(5)
  - css
  - img
  - js
    - index.js
  - lib
  - widgets
    - \_example
    - \_example\_append
    - \_example\_divwin
    - addmarker
    - bookmark
    - centerXY
    - manageBasemaps
    - manageLayers
    - mapCompare
    - mapSwipe
    - measure
    - navXZQH
    - plot
    - print
    - queryBaiduPOI
    - queryGaodeRoute

config.json

index.html

代码分析 属性 解决方案资源管理器

查看 config.json 并单击测试栏查看各 widget 效果，浏览对应的 widget 源码，学习每一个 widget，熟悉框架开发。



## 5 配置项手册

该章主要详细说明 `config.json` 文件的各项可配置信息，方便查阅。

### 5.1 Map 地图配置

Json 配置文件下面有 `map` 根节点，`map` 中可以配置下面参数进行初始化地图。

json 节点	说明	示例配置
<code>zoom</code>	地图默认显示级别，可选值 1-18	9
<code>center</code>	地图默认显示中心点，其中 <code>x</code> 为经度， <code>y</code> 为纬度	{ "x": 123.220102, "y": 33.834912 }
<code>extent</code>	地图默认显示矩形区域，与 <code>center</code> 作用相同	{ "xmin": 113, "ymin": 39, "xmax": 117, "ymax": 32 },
其它	所有 <code>L.map</code> 的 <code>Options</code> 参数都可以	如 <code>minZoom</code> ， <code>maxZoom</code> 等
<code>centerAutoLevel</code>	定位到点标注时，当地图级别小于此值时自动放大至该级别	11
<code>controls</code>	数组， <code>leaflet</code> 地图原生功能控件	
<code>crs</code>	地图坐标系，可选值： <code>gcj</code> ， <code>baidu</code> ， <code>EPSG3857</code> ， <code>EPSG4326</code> ， <code>EPSG4490</code>	<code>EPSG3857</code>
<code>basemaps</code>	数组，基础底图图层	这 2 个配置方式一致
<code>operationallayers</code>	数组，可叠加的图层	

#### 5.1.1 地图控件配置（controls）

Json 配置文件的 `map` 中下的 `controls` 参数，可以配置地图原生的一些控件和框架扩展的控件。所有控件的通用属性：

json 节点	说明
<code>type</code>	控件类型，必须存在
<code>position</code>	控件显示所在位置
<code>visible</code>	是否加载显示，默认为 <code>true</code>

可配置项目：

名称	配置项
比例尺	{ "type": "scale" }
放大缩小	{ "type": "zoom", "position": "bottomright" }
图层控制	{ "type": "layers", "position": "topright" }
鼠标经纬度提示	{ "type": "location", "showlevel": true }
自定义控件集合	{ "type": "tool", "item": [ "home", "location", "fullscreen" ], "position": "bottomright" }

### 5.1.2 地图图层配置 (basemaps 和 operationallayers)

Json 配置文件的 map 中下的 basemaps 参数和 operationallayers 参数，分别可以配置地图的基础图层和叠加图层，2 者配置参数都一样。所有图层使用 type 参数进行类别区分。

图层通用属性：

通用属性	说明
name	图层名称，必须存在
type	图层类型，必须存在
visible	图层默认加载后是否显示

#### 5.1.2.1 图层分组组合

可以将其他任意 type 类型图层放在 layers 参数数组内形成一个组合，方便一起显示隐藏，layers 参数为空时为虚节点，只是为了方便图层控制。

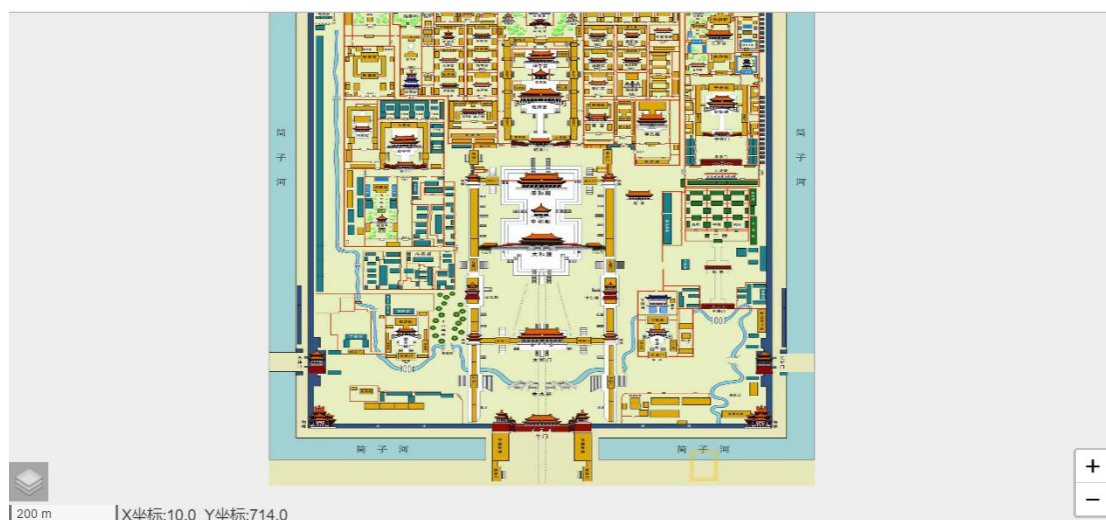
示例：{ "name": "卫星地图", "type": "group", "layers": [  
     { "name": "底图", "type": "arcgis\_tile", "url":  
     "http://services.arcgisonline.com/arcgis/rest/services/World\_Imagery/MapServer"  
     },  
     { "name": "注记", "type": "tile", "url":  
     "http://t{s}.tianditu.cn/DataServer?T=cia\_w&X={x}&Y={y}&L={z}", "subdomains":  
     "01234567"  
   ] },



### 5.1.2.2 单张图片作为底图

在无地图的图片模式下使用，其它参数支持 url 和 bounds 参数，注意此图层的"crs"参数必须为"Simple"

示例： { "name": "图片 1", "type": "image", "url": "img/uqmmmap.png", "bounds":  
[ [ -26.5, -25 ], [ 1021.5, 1023 ] ] }



### 5.1.2.3 具有 xyz 格式地址的瓦片图层

当瓦片地图 url 地址具有 xyz 规则的标准图层时使用，其它参数支持 L.tileLayer 所有 options。

示例： { "name": "OSM 地图", "type": "tile", "url":  
"http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png", "subdomains": "abc" }

### 5.1.2.4 OGC 标准 wms 服务图层

OGC 标准 WMS 瓦片图层， L.tileLayer.wms 所有 options

示例： { "name": "电子地图", "type": "wms", "url":  
"http://127.0.0.1:8888/geowebcache/service/wms", "layers": "china\_mercator\_vec" }

### 5.1.2.5 OGC 标准 wmts 服务图层

OGC 标准 WMTS 瓦片图层，可以定义 wmts 的所有参数，如 layer、format、style、tilematrixSet 等，并且支持 tilematrixBefore 或 matrixIds 来自定义 tilematrix 参数。

示例: { "name": "电子地图", type: "wmts", url: "http://srv1.zjditu.cn/ZJEMAPANNO\_2D/wmts", layer: "ZJEMAPANNO", style: "default", tilematrixSet: "TileMatrixSet0", format: "image/png", }

### 5.1.2.6 ArcGIS 标准的切片数据

将 arcgis 的切片数据直接通过 http 服务发布后使用, 其它参数支持 L.tileLayer 所有 options, 额外参数 isUpper 标识图片名称是否为大写, imgType 标识图片格式, 默认为.jpg

示例: { "name": "电子地图", "type": "arcgis\_cache", "url": "http://127.0.0.1:8888/mapcache/china\_mercator\_vec" }

Arcgis 标准切片数据:



### 5.1.2.7 ArcGIS Server 标准 MapServer 瓦片服务

加载 arcgis server 发布的标准瓦片服务图层, 其它参数支持 L.esri.tiledMapLayer 所有 options

示例: { "name": "全球地形", "type": "arcgis\_tile", "url": "http://server.arcgisonline.com/arcgis/rest/services/World\_Topo\_Map/MapServer" }

### 5.1.2.8 ArcGIS Server 标准 MapServer 矢量服务

加载 arcgis server 发布的标准服务图层, 其它参数支持 L.esri.dynamicMapLayer 所有 options, 额外支持 popup 或 columns 配置信息窗格式

示例: { "name": "大坝", "type": "arcgis\_dynamic", "url": "http://127.0.0.1:8888/arcgis/rest/services/MGDT/MGH/MapServer", "layers": [ 2 ], "popup": "<p>名称: {DAM\_NAME}<br>地区: {COUNTRY}<br>备注: {COMMENTS}</p>" },

### 5.1.2.9 ArcGIS Server 标准 FeatureServer 服务

加载 arcgis server 发布的标准 featureLayer 服务图层，其它参数支持 L.esri.featureLayer 所有 options，额外支持 popup 或 columns 配置信息窗格式，symbol 配置显示样式。

示例：{ "name": "摄像机", "type": "arcgis\_feature", "url": "http://127.0.0.1:8888/kcgis/services/CameraMap/featureserver/1", "popup": "<p>名称: {name}<br>仰角: {elevationAngle}<br>高度: {height}<br>型号: {model}</p>" }

➤ symbol 参数说明：当是点数据时 symbol 可支持 L.marker 所支持 options，iconField 和 iconOptions (L.icon)来配置显示的图标；当是线面数据时，symbol 支持 styleField 和 styleOptions 来配置显示的线面 style，见下图示例

(1) 点图层：同一个图片图标时

```
"symbol": {
  "draggable": true,
  "iconOptions": {
    "iconUrl": "img/marker/fx/fx1.png",
    "iconSize": [ 30, 30 ],
    "iconAnchor": [ 15, 15 ],
    "popupAnchor": [ 0, -15 ]
  }
},
```

(2) 点图层：根据数据使用不同图片图标时

```
"symbol": {
  "iconOptions": {
    "iconUrl": "img/marker/fx/wx4.png",
    "iconSize": [ 30, 30 ],
    "iconAnchor": [ 15, 15 ],
    "popupAnchor": [ 0, -15 ]
  },
  "iconField": "type",
  "iconFieldOptions": {
    "1": { "iconUrl": "img/marker/fx/wx1.png" },
    "2": { "iconUrl": "img/marker/fx/wx2.png" },
    "3": { "iconUrl": "img/marker/fx/wx3.png" },
    "4": { "iconUrl": "img/marker/fx/wx4.png" }
  }
},
```

(3) 点图层：根据数据使用不同 font 字体图标(如 font-awesome)时

```
"symbol": {
  "iconOptions": {
    "iconFont": "fa fa-flag",
    "color": "#FFEDA0",
    "iconSize": [ 30, 30 ]
  },
  "iconField": "type",
  "iconFieldOptions": {
    "1": { "color": "#FC4E2A" },
    "2": { "color": "#E31A1C" },
    "3": { "color": "#BD0026" },
    "4": { "color": "#800026" }
  }
},
```

(3) 线面图层：使用同一种样式时

```
"symbol": {
  "styleOptions": {
    "color": "white", "opacity": 1,
    "weight": 2,
    "fillColor": "#0000ff", "fillOpacity": 0.7
  }
},
```

(4) 线面图层：根据数据使用不同样式时

```
"symbol": {
  "styleOptions": {
    "color": "white", "opacity": 1, "weight": 2,
    "fillColor": "#FED976", "fillOpacity": 0.7
  },
  "styleField": "县代码",
  "styleFieldOptions": {
    "130702": { "fillColor": "#FD8D3C" },
    "130703": { "fillColor": "#FC4E2A" },
    "130725": { "fillColor": "#E31A1C" },
    "130732": { "fillColor": "#BD0026" }
  }
},
```



5.1.2.10 OSM 在线地图

在框架内已经内置 OSM 开源在线地图，使用配置如下: { "name": "OSM 地图", "type": "www\_osm" }



5.1.2.11 天地图在线地图

在框架内已经内置天地图在线地图，使用 layer 参数标识图层类型，crs 参数标识坐标系，默认为墨卡托坐标系，可以配置 4326 标识 wgs84 坐标系、4490 标识大地 2000 坐标系。

示例: { "name": "天地图", "type": "www\_tdt", "crs": "4326", "layer": "vec" }

layer 参数	说明
vec_d	电子底图单图层
vec_z	电子注记单图层
vec	电子地图（含底图和注记）
img_d	卫星底图单图层
img_z	卫星注记单图层
img	卫星地图（含底图和注记）
ter_d	地形底图单图层
ter_z	地形注记单图层
ter	地形地图（含底图和注记）

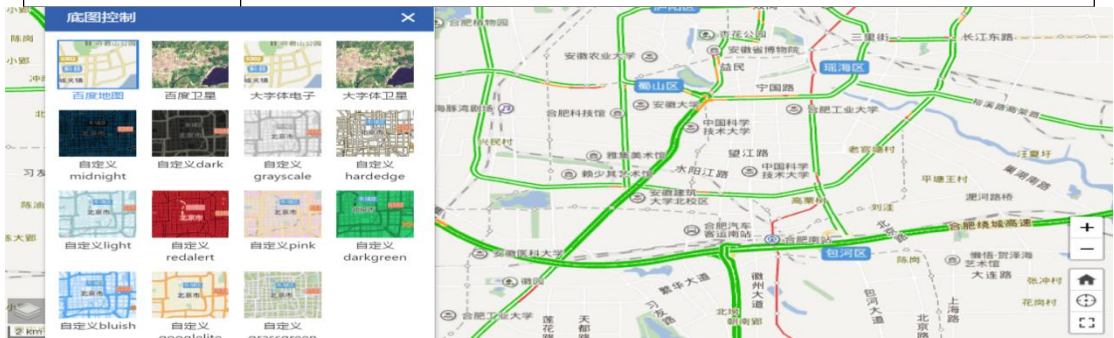


5.1.2.12 百度在线地图

在框架内已经内置百度在线地图，使用 layer 参数标识图层类型，bigfont 标识是否大字体地图，地图必须配置为 baidu 坐标系。

示例：{ "name": "百度", "type": "www\_baidu", "crs": "baidu", "layer": "vec" }

layer 参数	说明
vec	电子地图（含底图和注记）
img_d	卫星底图单图层
img_z	卫星注记单图层
img	卫星地图（含底图和注记）
time	百度实时路况信息
custom	各种自定义样式, 可选值: dark, midnight, grayscale, hardedge, light, redalert, googlelite, grassgreen, pink, darkgreen, bluish



5.1.2.13 高德在线地图

在框架内已经内置高德在线地图，使用 layer 参数标识图层类型，bigfont 标识是否大字体地图，crs 参数配置为 gcj，标识是国测局偏移坐标系，方便内部数据纠偏。

示例：{ "name": "高德", "type": "www\_gaode", "crs": "gcj", "layer": "vec" }

layer 参数	说明
vec	电子地图（含底图和注记）
img_d	卫星底图单图层
img_z	卫星注记单图层
img	卫星地图（含底图和注记）
time	高德实时路况信息



5.1.2.14 谷歌国内在线地图

在框架内已经内置国内谷歌在线地图，使用 layer 参数标识图层类型，crs 参数配置为 gcj，标识是国测局偏移坐标系，方便内部数据纠偏。

示例：{ "name": "谷歌", "type": "www\_google", "crs": "gcj", "layer": "img" }

layer 参数	说明
vec	电子地图（含底图和注记）
img_d	卫星底图单图层
img_z	卫星注记单图层
img	卫星地图（含底图和注记）
ter	地形地图（含底图和注记）

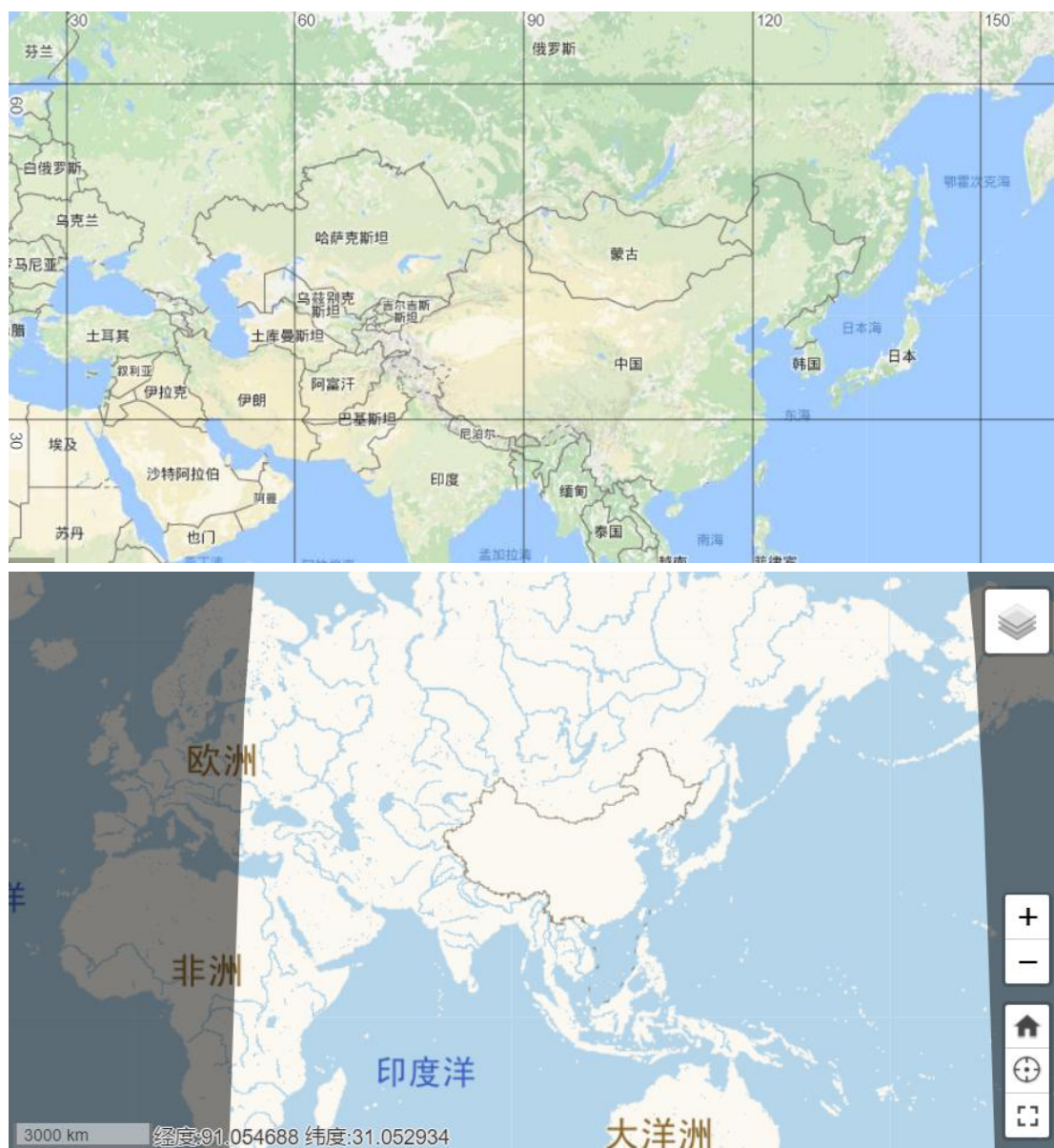


5.1.2.15 内置图层

在框架内有一些内置图层，该类图层只用配置 name 和 type 即可。

说明	配置
经纬网	{ "name": "经纬网", "type": "custom_jww" }
日照效果	{ "name": "日照效果", "type": "custom_lighting" }





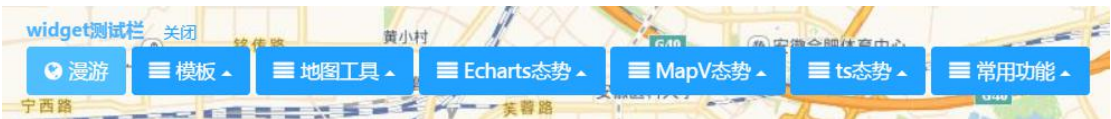


## 5.2 widget 模块配置

Json 配置文件下面有 widget 根节点，widget 中可以配置下面参数来记录 widget 模块的基础信息。

配置项	说明
debugger	是否显示插件测试栏，true 时会在地图下侧显示所有插件测试按钮，方便测试
version	为了清理浏览器缓存的版本值，开发时请设置为 <b>"time"</b> ，方便每次都不使用缓存，发布版本时每次指定不同值即可。
defaultOptions	所有插件的默认值
widgetsAtStart	数组，默认自启动并不可释放的插件，参数与 widgets 相同，其中 autoDisable 和 openAtStart 固定，设置无效
widgets	数组，所有插件。

debugger 打开后效果：



### 5.2.1 单个模块配置

**widget.widgetsAtStart** 和 **widget.widgets** 为数组，每个单项配置见下表，所有 widgets 共性的配置可放在默认值 defaultOptions 中配置。

配置项	说明	示例
name	<b>必须</b> ，插件中文名称，用于标识和弹窗标题。兼容用 label 命名定义	"书签"
uri	<b>必须</b> ，插件 JS 文件路径，路径是相对于 widgets 目录的路径	"widgets/bookmark/widget.js"
id	可选，定义该插件的唯一标识，用于外部调用传参 widget.activate('bookmark')；	"bookmark"
windowOptions	可选，存在弹窗的插件的弹窗相关参数配置，支持 layer 弹窗插件所有参数	见下面 5.2.2 表

openAtStart	可选, 进入地图后, 是否自动启动本插件, 默认 false	false
autoDisable	可选, 激活其他新插件时, 是否自动释放本插件, 默认 true	true
disableOther	可选, 激活本插件时, 是否释放其它已激活的插件, 默认 true	false
其他	可选, 定义的任意参数在 widget 内部方法中都可以通过 this.config 获取到	

### 5.2.2 弹窗参考配置

在有弹窗的情况下可以配置 windowOptions 参数, 也可以在 widget.js 中配置, 也可以在 json 文件中配置, json 文件配置优先, 会覆盖 widget.js 中配置, 目的是为了一个 widget 可以不同项目用不同配置。

配置项	说明
width	窗口宽度
height	窗口高度
position	窗口所在位置坐标, 可以配置字符串 t 顶部, b 底部, r 右边缘, l 左边缘, lt 左上角, lb 左下角, rt 右上角, rb 右下角; 也可以配置选择 { top:10, bottom:10, left:10, right:10 } 任意组合来定位;
maxmin	是否显示最大化按钮, 默认 false
resize	是否可以在弹层右下角拖动来拉伸尺寸。默认 true
shade	遮罩, 默认为 0 不显示, 可配置数字 0.3 透明度的黑色背景 ('#000'), 其他颜色, 可以 shade: [0.8, '#393D49']
shadeClose	当 shade 是存在的, 点击弹层外区域后是否关闭弹窗。
closeBtn	当为 0 时, 不显示关闭按钮, 配置 1 和 2 来展示两种风格的关闭按钮, 默认 1
其他	更多参数, 请参阅 <a href="http://www.layui.com/doc/modules/layer.html">http://www.layui.com/doc/modules/layer.html</a>