《插件开发规范 V1.0》

日期：2022 年 11 月 6 日

目录

1. 概述

本文主要说明插件开发的结构、方式和开发要求，针对读者为系统负责人、插件开发者以及项目管理者。

2. 插件结构

每个插件为一个独立的文件夹（如：Demo_Menu），放到指定目录（plugins）下，插件包含 JSON 配置文件(如：Demo_Menu.json)，JSON 配置文件名称必须和插件文件夹同名，格式为 JSON，其他资源可以自行分类存放，结构如下图：

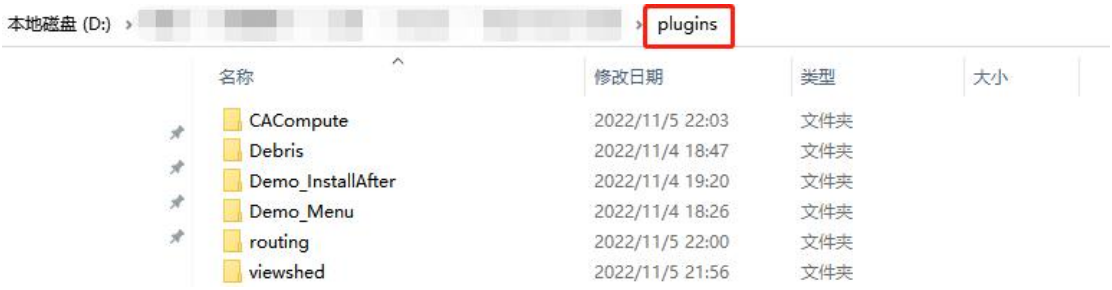

图 1 插件目录

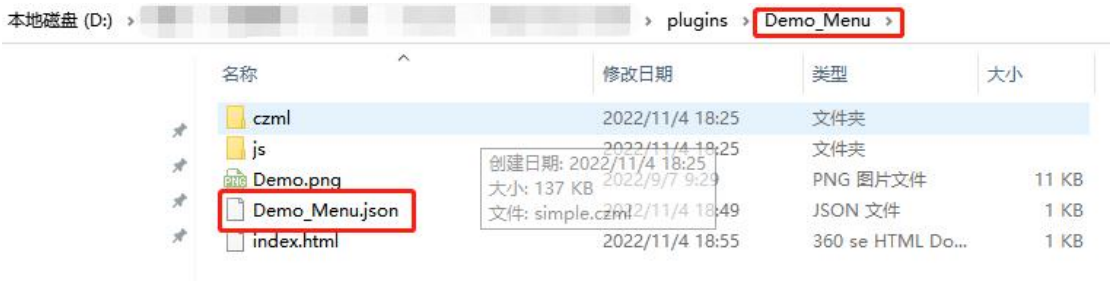

图 2 插件结构

3. 插件规范

3.1. JSON 配置文件

配置文件主要是为了生成菜单或安装插件后运行插件业务使用。

3.1.1. 配置文件字段

序号	字段名称	是否必需	字段说明	值
1	id	是	插件唯一编号	String:Plugins_{插件文件夹名字}
2	centralBody	否	菜单支持的中心天体，默认支持所有中心天体	中心天体名称：Earth、Moon 等
3	main	是	插件内容页面路径	文件名称：*.html，内容为：HTML 格式
4	menu	否	菜单配置	JSON,参考 menu 字段说明表
5	installAfter	否	插件安装后触发的方法	String，必需在插件对象里定义此值对应的方法

3.1.2. main 内容说明

在内容页面里必须注册插件对象（参考：3.2 插件对象），可以定义相关界面内容和引入需要的 js（提醒：1.最好以闭包的形式进行封装，如果要定义为全局类型，设置给 window 对象；2.由于 js 动态引入，在定义 class 对象后需要将 class 对象赋值给 window 对象，例如：class Demo{};window.Demo=Demo）、css 相关资源。例如：

index.html

```
<!-- 引入相关js文件 -->
<script src="/plugins/Demo_Menu/js/Func.js"></script>
<script src="/plugins/Demo_Menu/js/Demo.js"></script>

<!-- 引入相关css文件 -->
<link rel="stylesheet" href="/plugins/Debris/css/element.css">

<!-- 定义界面内容 -->
<div id="plugins_Demo_Menu" style="display:none;" class="demo">
    <button  onclick="Plugins.values['Plugins_Demo_Menu'].addEntity();">addEntity</button>
    <button  onclick="Plugins.values['Plugins_Demo_Menu'].addCzml();">addCzml</button>
    <button  onclick="Plugins.values['Plugins_Demo_Menu'].addPrimitives();">addPrimitives</button>
    <button  onclick="Plugins.values['Plugins_Demo_Menu'].clear();">clear</button>
</div>
```

3.1.3. menu 字段

序号	字段名称	是否必需	字段说明	值
1	parent	否	菜单的上级菜单	Menu_FENXI（分析菜单）、Menu_GONGJVJI（工具集菜单）
2	icon	是	菜单图标路径	String
3	name	是	菜单名称	String
4	width	否	菜单宽度，根据名称长度确定，如果不设置，名称超出部分不显示	Int
5	click	是	插件点击后触发的方法	String，必需在插件对象的 menu 属性里定义此值对应的方法

3.2. 插件管理器

用于插件的注册、获取插件对象等方法，全局名称为：Plugins。

3.2.1. 属性

序号	名称	类型	参数说明
1	values	Map<key:String,value:Object>	plugin 插件对象容器,key:插件对象的 id,value:插件对象

3.2.2. 方法

序号	方法名称	参数说明
1	add(plugin:Object)	plugin 插件对象

3.3. 插件对象

3.3.1. 插件对象属性

序号	字段名称	是否必需	字段说明	值
1	id	是	插件唯一编号,需要和配置文件里的 id 保持一致	String:Plugins_{插件文件夹名字}
4	menu	根据配置文件确定	用于菜单点的点击事件	JSON
5	installAfter	根据配置文件确定	插件安装后触发的方法	String，必需在插件对象里定义此值对应的方法

3.3.2. 插件定义和注册方法

```
var Demo = {
    //插件唯一编号,需要和配置文件里的 id 保持一致
    id: "Plugins_Demo_Menu",
    menu: {
        //点击菜单后触发的事件
        click: function(element) {
          //以下添加点击菜单后的业务逻辑
        }
    },
    //插件安装后触发的方法
    installAfter: function() {
      //以下添加安装后的业务逻辑
    },
};
//把插件对象添加到插件管理器
Plugins.add(Demo);
```

4. 全局对象和方法说明

4.1. solarSystem

负责管理 Viewer 对象（多窗口）、可以在 Viewer 里添加 Entity 对象、DataSource 对象、Primitive 对象和相关 PrimitiveCollection 对象等。

4.1.1. 属性

序号	名称	类型	参数说明
1	baseViewer	Viewer	基础视图对象，其他视图可视化内容和此视图同步显示。

4.1.2. 方法

序号	方法名称	参数说明
1	addEntity(entity:Entity|Object):Entity	添加 Entity 对象
2	removeEntity(entity:Entity):boolean	移除 Entity 对象
3	addDataSource(ds:DataSource |Promise<DataSource>):Promise<DataSource>	添加 DataSource 对象
4	removeDataSource(ds:DataSource):boolean	删除 DataSource 对象
5	addPrimitive(p:Primitive |PrimitiveCollection):Object	添加 Primitive 对象或集合
6	removePrimitive(p:Primitive |PrimitiveCollection):boolean	删除 Primitive 对象或集合

5. 插件配置示例

以下为基本示例，如果要查看完整示例，请参考 plugins/Demo_Menu、plugins/Demo_InstallAfter。

5.1. 第一步（新建插件）

创建文件夹：Demo_Menu，并在文件夹里创建配置文件 Demo_Menu.json。

5.2. 第二步（插件配置）

打开 Demo_Menu.json，编写配置内容：

```
{
    "id": "Plugins_Demo_Menu",
    "centralBody": "Earth",
    "main":"index.html",
    "menu":
    {
        "parent": "Menu_FENXI",
        "icon": "plugins/Demo_Menu/Demo.png",
        "name": "示例",
        "width": 120,
        "click": "click1"
    },
    "installAfter": "installAfter1"
}
```

5.3. 第三部（插件对象定义（闭包定义））

新建文件 Demo.js，编写内容如下：

```
(function() {

    var Demo = {
        //插件唯一编号,需要和 json 里的 id 保持一致
        id: "Plugins_Demo_Menu",
        menu: {
            //点击菜单后触发的事件
            click1: function(element) {
                const openNewLayerIndex = layer.open({
                    type: 1,
                    title: "示例_点击菜单后显示",
                    shadeClose: true,
                    shade: false,
                    area: '340px', // 宽高
                    offset: ['140px', ($(window).width() - 450) + 'px'],
                    success: function(layero, index) {
                    },
                    content: $('#plugins_Demo_Menu'),
                    btn: [],
                    end: function() {
                        //关闭窗口时删除所有添加的对象
                        Demo.clear();
                    }
```

```
            });
        }
    },
    //插件安装后触发的方法
    installAfter: function() {
     $('#plugins_Demo_Menu').show();
    },
    addEntity: function() {
        if (this.entity) {
          //删除 Entity 对象
            solarSystem.removeEntity(this.entity);
            this.entity = undefined;
        }
        this.entity = solarSystem.addEntity({
            position: Cesium.Cartesian3.fromDegrees(115.59777, 30.03883),
            point: {
                pixelSize: 5,
                color: Cesium.Color.RED,
                outlineColor: Cesium.Color.WHITE,
                outlineWidth: 2
            },
            label: {
                text: "测试",
                font: "24px Helvetica",
                fillColor: Cesium.Color.SKYBLUE,
                outlineColor: Cesium.Color.BLACK,
                outlineWidth: 2,
                style: Cesium.LabelStyle.FILL_AND_OUTLINE,
                pixelOffset: new Cesium.Cartesian2(0, -30)
            }
        });
    },
    addCzml: function() {
        if (this.dataSource) {
            solarSystem.removeDataSource(this.dataSource, true);
            this.dataSource = undefined;
        }
        var dataSourcePromise =
solarSystem.createCzmlDataSource("plugins/Demo_Menu/czml/simple.czml");
        var that = this;
        solarSystem.addDataSource(dataSourcePromise).then(function(ds) {
            that.dataSource = ds;
        });
    },
```

```
addPrimitives: function() {
    if (this.IntervalHanlder) {
        clearInterval(this.IntervalHanlder);
    }
    this.IntervalHanlder = setInterval(Demo.addPrimitives_, 50);
},
addPrimitives_: function() {
    var numPoints = 100000;
    if (Demo.pointPrimitives) {
        Demo.pointPrimitives.removeAll();
    } else {
        Demo.pointPrimitives = solarSystem.addPrimitive(
            new Cesium.PointPrimitiveCollection()
        );
    }
    var base = solarSystem.baseViewer.scene.globe.ellipsoid.radii.x;
    var color = Cesium.Color.LIGHTSKYBLUE;
    var outlineColor = Cesium.Color.BLACK;

    for (var j = 0; j < numPoints; ++j) {
        var position = new Cesium.Cartesian3(
            16000000 * Math.random() - 8000000,
            -((4000000 * j) / numPoints + base),
            2000000 * Math.random() - 1000000
        );

        Demo.pointPrimitives.add({
            pixelSize: 5,
            color: color,
            outlineColor: outlineColor,
            outlineWidth: 0,
            position: position,
        });
    }
},
clear: function() {
    if (this.IntervalHanlder) {
        clearInterval(this.IntervalHanlder);
        this.IntervalHanlder = undefined;
    }
    if (this.entity) {
        solarSystem.removeEntity(this.entity);
        this.entity = undefined;
    }
```

```
            if (this.dataSource) {
                solarSystem.removeDataSource(this.dataSource, true);
                this.dataSource = undefined;
            }
            if (this.pointPrimitives) {
                this.pointPrimitives.removeAll();
            }
            if (this.labelPrimitives) {
                this.labelPrimitives.removeAll();
                solarSystem.removePrimitive(this.labelPrimitives);
                this.labelPrimitives = undefined;
            }
        }
    };
    //把插件对象添加到插件管理器
    Plugins.add(Demo);
})();
```

5.4. 第四步（编写 main 界面）

根据配置文件里的 main 属性设置，新建文件 index.html（编码为：UTF-8），编写内容如下：

```
<!-- 引入相关 js 文件 -->
<script src="/plugins/Demo_Menu/js/Demo.js"></script>

<!-- 引入相关 css 文件 -->
<link rel="stylesheet" href="/plugins/Demo_Menu/css/demo.css">

<!-- 定义界面内容 -->
<div id="plugins_Demo_Menu" style="display:none;" class="demo">
    <button
onclick="Plugins.values['Plugins_Demo_Menu'].addEntity();">addEntity</b
utton>
    <button
onclick="Plugins.values['Plugins_Demo_Menu'].addCzml();">addCzml</butto
n>
    <button
onclick="Plugins.values['Plugins_Demo_Menu'].addPrimitives();">addPrimi
tives</button>
    <button
onclick="Plugins.values['Plugins_Demo_Menu'].clear();">clear</button>
```

```
</div>
```

5.5.第五步（发布插件）

将以上编写的文件和相关资源放（菜单图标、第三方库等）到 Demo_Menu 文件夹下后，把 Demo_Menu 拷贝至 plugins 下，即完成发布，最终目录如下：