

服务器端渲染 3DBIMGIS 轻量化融合引擎 二次开发指南



西安葛兰创智信息科技有限公司

www.glendale.top

目录

1.	服务端渲染 GIS/BIM/3D 融合引擎概述.....	3
1.1	服务器端渲染 GIS/BIM/3D 融合引擎.....	3
1.2	推荐学习路线	3
2.	二次开发：支持格式	4
3.	二次开发：前期准备	5
3.1	私有化搭建环境	6
3.2	服务端接口地址获取方式：	6
3.3	获取开发资源包	7
4.	二次开发：创建应用	8
4.1	上传 BIM 模型	8
4.1.1	上传模型	8
4.1.2	获取 BIM 模型轻量化结果	10
4.1.3	创建应用	12
4.2	GIS 场景开发	16
4.2.1	OSGB 数据开发	16
4.3	获取模型构件属性的方法	22
4.4	多数据场景集成开发	23
4.4.1	多模型集成开发	23
4.4.2	GIS+BIM 集成开发	23
5.	二次开发：一般建议.....	24
5.1	轻量化模型上传贴图方法	24
5.2	链接模型压缩层级介绍	25
5.3	OSGB 压缩层级介绍	25
5.4	MAX 源文件如何导出带贴图的 FBx 格式.....	26

1. 服务端渲染 GIS/BIM/3D 融合引擎概述

1.1 服务器端渲染 GIS/BIM/3D 融合引擎

葛兰岱尔服务器端渲染 3DBIMGIS 轻量化融合引擎具有自主知识产权，自主研发、技术领先、支持模型格式多，基于用户对 3D/BIM/GIS 渲染效果需求的不断提升，同时也为了更好地支持大体量模型的加载和渲染，充分发挥服务器或云端的硬件算力（CPU+GPU）优势，葛兰岱尔充分利用 UE4（Unreal Engine4）在 3D/BIM/GIS 渲染方面的基础优势，实现了基于服务器端渲染+WebRTC 推流技术架构的轻量化引擎产品，满足 3D/BIM/GIS 模型的服务器端渲染+Web 端应用的需求场景。支持 一键安装、本地私有化部署、保证数据安全，用户无需了解或学习 UE4，简单易用，服务器并发支持能力强、Web 端响应速度快，支持模型格式多，是工程建筑、制造业、石油化工领域的专业轻量化 3D 渲染引擎；纯中间件产品、API 及模型属性数据全开放、二次开发（不限开发语言）方便灵活

1.2 推荐学习路线

模块开发	类型	是否支持二次开发	学习路径
BIM 模型开发	三维模型	√	点我跳转
GIS 场景开发	场景模块	√	点我跳转

➤ 技术支持群

QQ-1 群：649763516

QQ-2 群：466682725

➤ 开发者宝典

- [JavaScriptAPI](#)：包含全部开放的 JSAPI 说明，帮助你快速搭建属于自己的 BIM 应用；
- [ServiceAPI](#)：包含服务端全部开放的服务端 API 接口说明，包含数据的上传、状态查询、属性同步至业务库等重要接口，实现超低代码量进行二次开发；
- [DB 表说明](#)：模型属性*.db 库文件表结构说明；
- [其他宝典](#)：其他技术类文档&开发使用过程中的高频问题汇总；

2. 二次开发：支持格式

目前支持的格式:

三维格式				
免费版引擎支持的格式				
.rvt（Revit）	.rfa（Revit）	.step	.fbx	.ifc
.stp				
收费版引擎支持的格式或软件				
Rhino	PTC Creo	.dgn(Bentley)	.CEL（Bentley）	.SKP
.STL	Tekla	.MAX	.OBJ	.DAE
.3DS	.rvn(PDMS)	Simens NX	.3DXML	
.CATPART/.CATPRODUCT(CATIA)		.nwd/.nwc(Naviswork)		
.sldprt/.sldasm(Solidworks)				
二维格式				
.dwg	.dxf			
倾斜摄影数据				
.osgb				
地形数据				
.tif				
点云数据				
.las	.ply	.pnts		
白模数据				
.shp				

不同格式支持的功能:

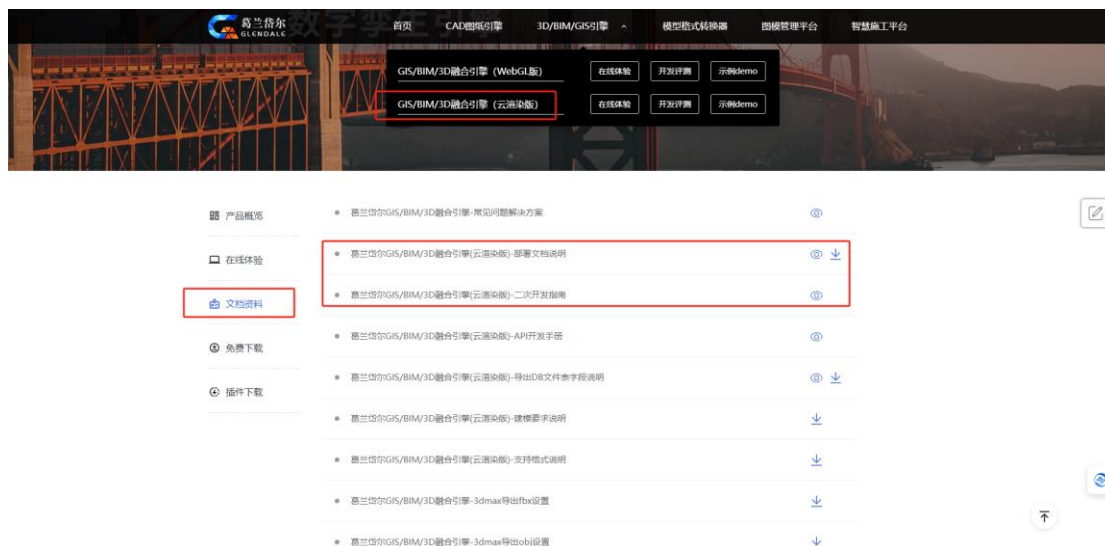
模型格式	属性解析	结构树解析	贴图
.RVT	√	√	√

.RFA	√	√	√
.CATPRODUCT	√	√	×
.CATPART	√	√	×
.SKP	√	√	√
.MAX	√	√	√
.DGN	√	√	√
.NWD	√	√	×
.NWC	√	√	×
Tekla	√	√	×
.IFC	√	√	×
.3DXML	×	√	×
.FBX	√	√	√
.OBJ	√	√	√
.DAE	√	√	√
.3DS	√	√	√
.RVM	√	√	×
.STP	×	√	×
.STEP	×	√	×
.STL	×	√	×

3. 二次开发：前期准备

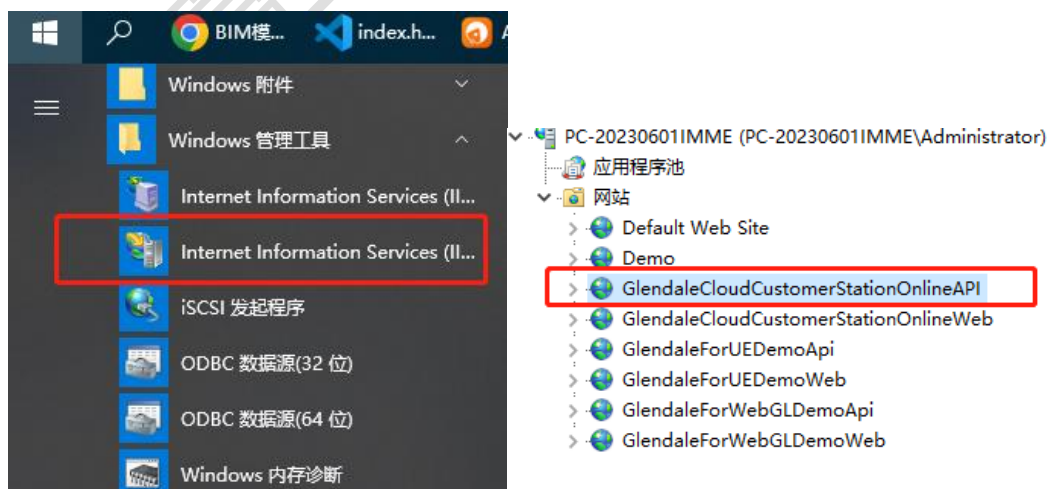
3.1 私有化搭建环境

此步骤是基于引擎进行二次开发的关键一步，需要客户本地搭建轻量化环境，具体部署工作请参考[请参考](#)



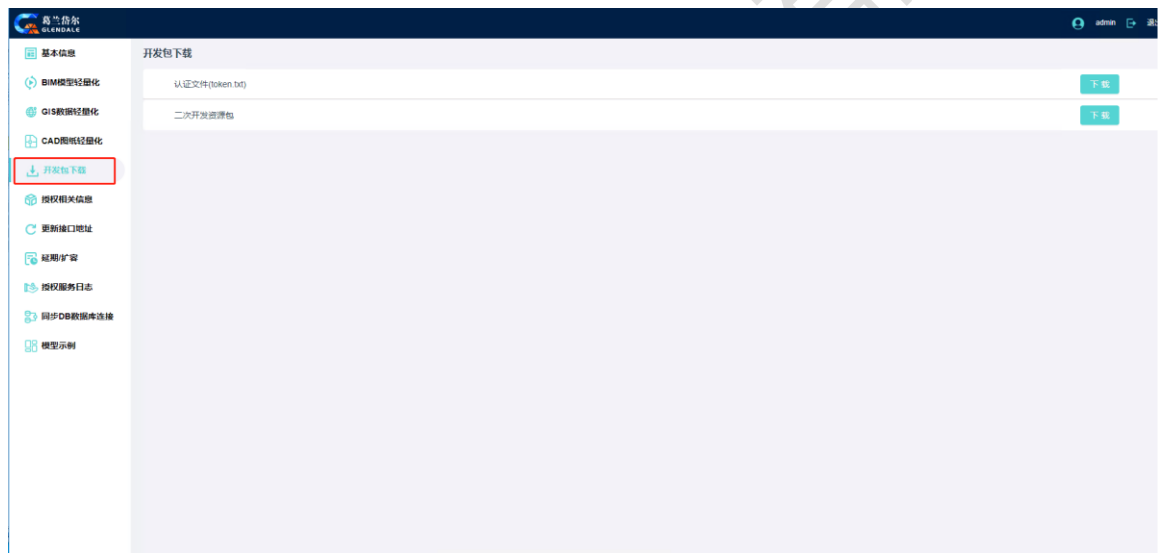
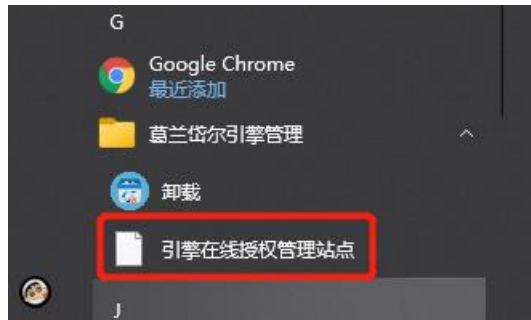
3.2 服务端接口地址获取方式:

打开本地 IIS 管理器，在左侧目录树中找到 GlendaleCloudCustomerStationOnlineAPI 站点，该站点的 web 访问地址为所需的服务端接口地址。



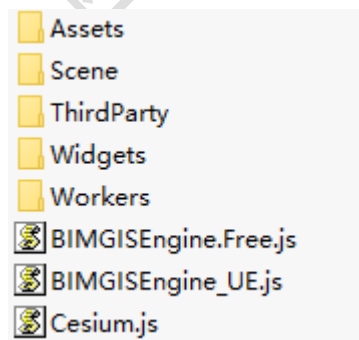
3.3 获取开发资源包

完成私有化搭建后,在“开始菜单”中找到并登入“引擎授权管理站点”(账号: admin, 密码: glendale.top), 点击左侧“开发包下载”, 下载 Token 及二次开发资源包.



认证文件 (**token.txt**): 前端引擎授权文件

二次开发资源包 (**Cesium 包**): 二次开发时所需引入的资源包



Js 版本及使用说明：

引擎类别	js 名称	版本名称	备注
WebGL 引擎	BIMGISEngine.Developer	开发版	本产品不用引入
	BIMGISEngine.Free	免费版	
	BIMGISEngine.Reviewer	评测版	
	BIMGISEngine_edit	正式商用版	
	BIMGISEngine.GMcharge	图模正式版	
服务端渲染引擎	BIMGISEngine_UE	UE 引擎	

Ps: 用户需根据自己申请或者购买的引擎授权 License 在二次开发时引入相应的 js 文件。

4. 二次开发：创建应用

4.1 上传 BIM 模型

开发流程：

上传 BIM 模型 → 获取 BIM 模型轻量化结果 → 创建应用

4.1.1 上传模型

引擎提供上传服务端 API 接口，需用户编写请求代码，传入对应请求参数进行上传。

也可使用其他开源接口测试工具，实现模型上传（以下代码粘贴即用，其参数调整[可参考](#)）

注意：我们也提供了[大文件切块上传接口](#)供您使用，有些接口测试工具并不支持此种方式，还需客户在使用时注意下。

代码样例：

PSOT 请求 服务端接口访问地址/api/app/model/upload-file

参数：

类型	名称	说明
Header	Content-Type	application/octet-stream
Header	Token	token.txt 中内容 如何获取?
Query	input	<pre>{ "Name": "name", "InitiatingUser": "admin", "UniqueCode": "", "Priority": "202", "ModelUploadUrl": "", "OtherInfo": "", "ConfigJson": { "style": 1, "zGrid": 0, "viewStyle": 0, "drawing": 0, "accuracy": 5, "parametric": 0, "familyName": "", "writetype": 0, "locationType": 0, "vertexNormal": 0, "isExportLines": 0, "unitRatio": 0.001, "type": 2, "offsetX": 0, "offsetY": 0, "offsetZ": 0, "isInstance": 0, "maxCountInstance": 100, "isLod": 0, "srs": "", "srsOrigin": [], "longitude": 1.9003144895714261, "latitude": 0.5969026041820608, "transHeight": 0, "edgeOn": 0, "level": 1, "xCount": 1, "yCount": 1, } }</pre>

		<pre>"draco": 0, "flipY": 0, "linePercent": 1, "minDistance": 0, "compatible": 0, "faceNumLimit": 1000000, "textureLimit": 10240, "textureSizePercent": 50, "textureQuality": 30, "combineTexture": 0, "blockRender": 1, "useGisCS": 1, "engineType": 2, "materialType": 0 } }</pre>
Body	file	上传需要轻量化的模型文件（文件流）

接口返回：

```
{
  "code": 1,
  "codeMsg": "上传成功!",
  "datas": {
    "lightweightName": "202112161733280636"
  },
  "dataCount": 0
}
```

4.1.2 获取 BIM 模型轻量化结果

➤ 轻量化后模型默认存放路径：

. \GLCloudEngineServerOnline\CustomerStationAPI\Tools\output\model

➤ 通过服务端 API 接口获取轻量化后模型访问地址方式：

代码样例:

POST 请求: 服务端接口访问地址/api/app/model/query-model-info

参数:

类型	名称	说明
Header	Token	token.txt 中内容 如何获取?
Query	LightweightName	模型轻量化名称

PS: **modelAccessAddress** 的值为轻量化后模型访问地址

返回结果:

```
{
  "code": 1,
  "codeMsg": "查询成功",
  "datas": [
    {
      "name": "",
      "lightweightName": "202112041418225685",
      "lastStatus": 0,
      "status": 100,
      "statusDescription": "转码成功",
      "isMain": false,
      "taskCode": "96914eb5-a33d-40c1-95d9-390817353965",
      "transcodingInfo": "",
      "initiatingUser": "",
      "uniqueCode": "",
      "fileType": "",
      "softwareType": "",
      "dataVersion": ""
    }
  ]
}
```

```
"fileSize": 0,
"creationTime": "2021-12-04T14:18:23.840158",
"executionTime": null,
"completionTime": "2021-12-04T14:19:47.456946",
"priority": 2,
"modelDownloadUrl": "",
"modelUploadUrl": "",
"md5": "DAA85BFF36338D7417BBDD1D0E99B930",
"otherInfo": "",
"callbackInterfaceURL": "",
"modelAccessAddress": "http://192.168.31.109:18086/Tools/output/model/202203111407228582/root.glt",
"dbSyncStatus": 2,
"plugin": null,
"pluginVersion": null,
"toolsVersion": null,
"id": 105
}
],
"dataCount": 0
}
```

4.1.3 创建应用

1) 首先在磁盘中创建 Demo_01 目录，其次将下载的[二次开发包](#)解压到该目录，其次再创建一个 index.html 文件，最后将以下代码复制进去。

名称	修改日期	类型	大小
 BIMGISEngine.js	2022-2-24 下午 5:01	JavaScript 文件	21 KB
 index.html	2022-3-11 下午 4:26	Chrome HTML Docu...	9 KB

```
<!DOCTYPE html>
<html lang="en">

  <style>
    /* 第三步：容器样式，根据自身业务调整，这里只做参考 */
    html,
    body,
    #gl_UEngineContainer {
      width: 100%;
      height: 100%;
      margin: 0;
      padding: 0;
      overflow: hidden;
      position: relative;
    }
  </style>
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>云渲染引擎</title>
    <script src="BIMGISEngine.js"></script><!-- 第一步：先引入相关 css
及 js 文件 -->
  </head>

  <body>
    <div id="gl_UEngineContainer"></div><!-- 第二步：创建容器 -->
    <script>
      //配置参数
      // 第四步： 初始化 Api
```

```

var filepath

='http://192.168.31.109:18086/Tools/output/model/202203111407228582/root.glt'//模型加载地址填写至此

var api= new SAPI({
  serverIP: `127.0.0.1`, //服务 ip 地址
  port: 8000, //HTTP 端口
  container: `gl_UEngineContainer`, //[必须]容器 id
  secretKey: '', // token.txt 文件内容填写至此
  openEarth: false, //[可选]开启场景
  bgColor: "rgba(135 ,206 ,250,1)", //[可选]bim 模式场景背景色
  sceneTime: [8, 2022, 2, 14, 13, 00], //[可选]场景时间
}, () => {
  console.log("初始化成功"); //初始化成功后在进行其他操作，如加载模型等

  api.Model.add(filepath
    ,//模型加载地址

    "175",
    () => {
      console.log("模型开始加载");
    }, () => {
      console.log("模型加载成功");
    }, {
      FlyTo: true,
    }
  )
});
</script>
</body>
</html>

```

2) 配置授权及模型加载地址说明

- i. 页面中配置如下图控制台中的服务 ip 地址及 Http 端口

服务控制台 - 葛兰岱尔

常规选项

☒ 启动后自动后台运行
 ☒ 退出时自动关闭所有进程

信令服务器

流端口: 20001~20005

http端口: 1001~1005

注: 端口之间可用逗号分隔,范围段端口用~表示。
例:8081,100~1000,可进行组合配置

服务选项

本机IP地址或域名: 192.168.1.65

matchmakerPort: 20000

httpPort: 8000

云服务

☒ 启用coturn服务
 ☒ 自动断开长时间没有交互操作的用户连接

时长(秒): 30

启动

停止

[系统信息 2022-12-06 13:37:21] checkFortService

[系统信息 2022-12-06 13:38:21] checkFortService

[系统信息 2022-12-06 13:39:21] checkFortService

[系统信息 2022-12-06 13:40:21] checkFortService

[系统信息 2022-12-06 13:41:21] checkFortService

ii. 将[下载的 token](#)及返回的(modelAccessAddress 字段值) 填写至 index 页面对应的字段值中

```

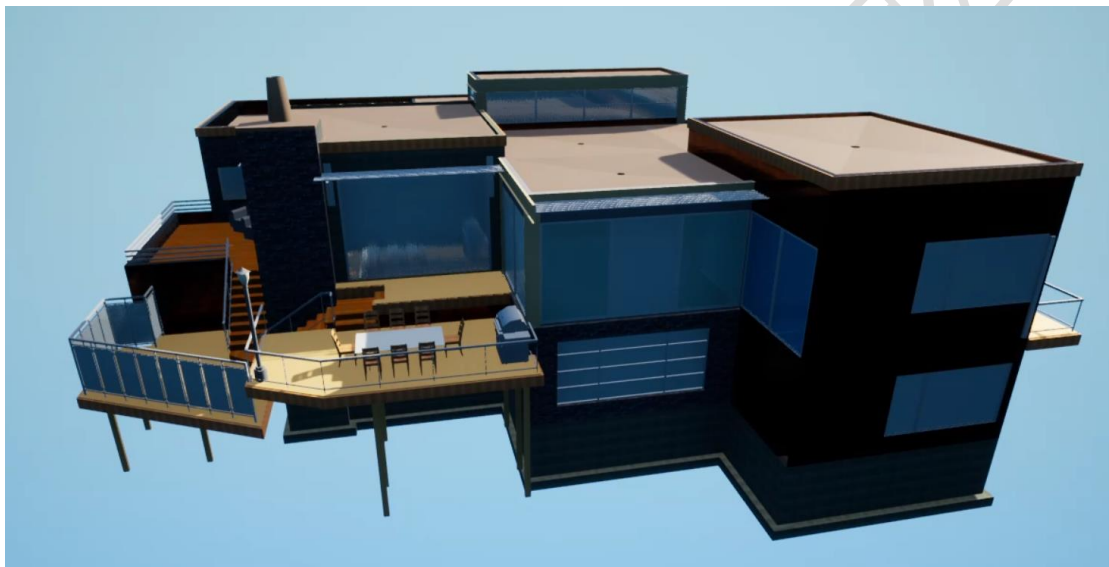
<script>
    //配置参数
    // 第四步: 初始化 Api
    接口返回的 modelAccessAddress 值的加载地址填写至此
    var filepath =
    'http://localhost:8090/Tools/output/model/1149552/root.glt'
    var api= new SAPI({
        serverIP: `127.0.0.1`, //服务 ip 地址
        port: 8000, //HTTP 端口
        container: `gl_UEngineContainer`, //[必须]容器 id
        secretKey: '', //下载的 token.txt 文件内容填写至此
        openEarth: false, //[可选]开启场景
        bgColor: "rgba(135 ,206 ,250,1)", //[可选]bim 模式场景背景色
        sceneTime: [8, 2022, 2, 14, 13, 00], //[可选]场景时间
    }, () => {
        console.log("初始化成功"); //初始化成功后在进行其他操作, 如加载模型等
        api.Model.add(filepath, //模型加载地址
            "175",
            () => {
                console.log("模型开始加载");
            }
        );
    });
  
```

服务器端渲染 3DBIMGIS 轻量化融合引擎 | WebGL 轻量化 CAD 图纸引擎 | 智慧施工管理 (+GIS/BIM) 平台 | 设计成果 (+BIM/CAD) 在线交付平台

官网 www.glendale.top | 热线 029-81163590 | 邮箱 publish@glendale.top

```
    }, () => {  
      console.log("模型加载成功");  
    }, {  
      FlyTo: true,  
    }  
  )  
});  
</script>
```

3) 用代码编辑器 (Hbuilder, vscode) 或 IIS, Nginx 等发布此 demo。



4.2 GIS 场景开发

开发流程:

上传数据→获取数据轻量化结果→创建应用

4.2.1 OSGB 数据开发

1 上传 osgb 数据

引擎提供上传服务端 API 接口，需用户编写请求代码，传入对应请求参数进行上传。

也可使用其他开源接口测试工具，实现模型上传（以下代码粘贴即用，其参数调整[可参考](#)）

注意：我们也提供了[大文件切块上传接口](#)供您使用，有些接口测试工具并不支持此种方式，还需客户在使用时注意下。

代码样例：

PSOT 请求 服务端接口访问地址/api/app/gismodel/OsgbUploadFile

参数：

类型	名称	说明
Header	Content-Type	application/octet-stream
Header	Token	token.txt 中内容 如何获取?
Query	input	<pre>{ "Name": "name", "LightweightName": "", "InitiatingUser": "admin", "UniqueCode": "customize", "Priority": "202", "ModelUploadUrl": "", "OtherInfo": "" }</pre>
Body	file	上传需要轻量化的 OSGB 数据文件（文件流）

接口返回：

```
{
  "code": 1,
  "codeMsg": "上传成功!",
  "datas": {
    "lightweightName": "202201261451175181"
  },
  "dataCount": 0
}
```

2 获取 osgb 数据轻量化结果信息

➤ 轻量化后模型默认存放路径：

. \GLCloudEngineServerOnline\CustomerStationAPI\Tools\output\model

➤ 通过服务端 API 接口获取轻量化后模型访问地址方式：

代码样例：

POST 请求：服务端接口访问地址/api/app/gismodel/QueryModelInfo

参数：

类型	名称	说明
Header	Token	token.txt 中内容 如何获取?
Query	LightweightName	模型轻量化名称
Query	FileType	OSGB

PS: **modelAccessAddress** 的值为轻量化后 osgb 数据访问地址

返回结果：

```
{
  "code": 1,
  "codeMsg": "查询成功",
  "datas": [
    {
      "name": "test",
      "lightweightName": "202204091617268149",
      "transcodingInfo": "",
      "initiatingUser": "admin",
      "uniqueCode": null,
      "fileType": "OSGB",
      "creationTime": "2022-04-09T16:18:42.38165",
      "executionTime": null,
      "completionTime": "2022-04-09T16:20:05.002348",
      "priority": 202,
      "modelDownloadUrl": "",
      "modelUploadUrl": ""
    }
  ]
}
```

```
"otherInfo": "",
"callbackInterfaceURL": null,
"sourceFileName": "202204091617268149.zip",
"lastStatus": null,
"status": 100,
"statusDescription": "轻量化成功",
"modelAccessAddress": "http://localhost:8086/Tools/output/model/202204091617268149/root.glt",
"configJson": null,
"id": 3
},
"dataCount": 0
}
```

3 创建应用

- 1) 首先在磁盘中创建 Demo_01 目录，其次将下载的[二次开发包](#)解压到该目录，然后再创建一个 index.html 文件，最后将如下示例代码复制进去。

名称	修改日期	类型	大小
 BIMGISEngine.js	2022-2-24 下午 5:01	JavaScript 文件	21 KB
 index.html	2022-3-11 下午 4:26	Chrome HTML Docu...	9 KB

```
<!DOCTYPE html>
<html lang="en">

  <style>
    /* 第三步：容器样式，根据自身业务调整，这里只做参考 */
    html,
    body,
    #gl_UEngineContainer {
      width: 100%;
      height: 100%;
```

```

margin: 0;
padding: 0;
overflow: hidden;
position: relative;
}
</style>
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>云渲染引擎</title>
  <script src="BIMGISEngine.js"></script><!-- 第一步：先引入相关 css
及 js 文件 -->
</head>

<body>
  <div id="gl_UEngineContainer"></div><!-- 第二步：创建容器 -->
  <script>
    //配置参数
    // 第四步： 初始化 Api
    var filepath = '
http://192.168.31.109:18086/Tools/output/model/202203111407228582/root.
glt'//模型加载地址填写至此

    var api= new SAPI({
      serverIP: `127.0.0.1`, //服务 ip 地址
      port: 8000, //HTTP 端口
      container: `gl_UEngineContainer`, //[必须]容器 id
      secretKey: '', // token.txt 文件内容填写至此
      openEarth: false, //[可选]开启场景
      bgColor: "rgba(135 ,206 ,250,1)", //[可选]bim 模式场景背景色
      sceneTime: [8, 2022, 2, 14, 13, 00], //[可选]场景时间
    }, () => {
      console.log("初始化成功"); //初始化成功后在进行其他操作，如加载模
型等

      api.Model.add(filepath
, //模型加载地址

        "175",
        () => {
          console.log("模型开始加载");
        }, () => {

```

```

        console.log("模型加载成功");
    }, {
        FlyTo: true,
    }
)
});
</script>
</body>
</html>

```

2) 配置授权及模型加载地址说明

i. 页面中配置如下图控制台中的服务 ip 地址及 Http 端口

服务控制台 - 葛兰岱尔

常规选项

☒ 启动后自动后台运行

☒ 退出时自动关闭所有进程

信令服务器

流端口: 20001~20005

http端口: 1001~1005

注: 端口之间可用逗号分隔,范围段端口用~表示。
例:8081,100~1000,可进行组合配置

云服务

☒ 启用coturn服务

☒ 自动断开长时间没有交互操作的用户连接

时长(秒): 30

服务选项

本机IP地址或域名: 192.168.1.65

matchmakerPort: 20000

httpPort: 8000

启动 停止

【系统信息 2022-12-06 13:37:21】 checkPortService
 【系统信息 2022-12-06 13:38:21】 checkPortService
 【系统信息 2022-12-06 13:39:21】 checkPortService
 【系统信息 2022-12-06 13:40:21】 checkPortService
 【系统信息 2022-12-06 13:41:21】 checkPortService

ii. 将[下载的 token](#)及返回的(modelAccessAddress 字段值) 填写至 index 页面对应的字段值中。

```

<script>
    //配置参数
    // 第四步: 初始化 Api
    接口返回的 modelAccessAddress 值的加载地址填写至此
var filepath =
'http://localhost:8090/Tools/output/model/1149552/root.glt'
    var api= new SAPI({
        serverIP: `127.0.0.1`, //服务 ip 地址

```

```

port: 8000, //HTTP 端口
container: `gl_UEngineContainer`, //[必须]容器 id
secretKey: '', //下载的 token.txt 文件内容填写至此
openEarth: false, //[可选]开启场景
bgColor: "rgba(135,206,250,1)", //[可选]bim 模式场景背景色
sceneTime: [8, 2022, 2, 14, 13, 00], //[可选]场景时间
}, () => {
  console.log("初始化成功"); //初始化成功后在进行其他操作，如加载模
  api.Model.add(filepath, //模型加载地址
    "175",
    () => {
      console.log("模型开始加载");
    }, () => {
      console.log("模型加载成功");
    }, {
      FlyTo: true,
    }
  )
});
</script>

```

1) 用代码编辑器(Hbuilder, vscode) 或 IIS, Nginx 等发布此 demo。



4.3 获取模型构件属性的方法

我们还提供了以下两种方式助力客户获取模型构件属性：

- 提供了获取[模型构件属性的接口](#)，前端调用即可获取构件相关属性。
- 提供了[属性同步接口](#)，将轻量化后 db 文件中的表结构及数据同步至客户数据库中，助力数据对接。

4.4 多数据场景集成开发

4.4.1 多模型集成开发

模型上传，获取轻量化结果和加载请参考“BIM 模型开发”模块

注：

- 1) 多个模型多次调用 [api.Model.add\(\)](#) 接口即可。
- 2) 每个模型配置文件中所配置的坐标位置必须一致

影响参数：

```
"ConfigJson":{  
  "srs": "",  
  "srsOrigin": [],  
  "longitude": 1.9003144895714261,  
  "latitude": 0.59690260418206076,  
}
```

4.4.2 GIS+BIM 集成开发

- 模型上传、获取轻量化结果、数据加载请参考“BIM 模型开发”和“GIS 场景开发”模块

BIM 模型影响参数：

```
"ConfigJson":{  
  //.....省略其他参数.....  
  "type": 2, //2 为自定义计算偏移值  
  "srs": "", //自定义 EPSG 代号  
  "srsOrigin": [], //自定义项目基点坐标  
  "longitude": 1.9003144895714261, //自定义弧度制经度值  
  "latitude": 0.59690260418206076, //自定义弧度制纬度值  
}
```

- 场景初始化时打开 GIS 场景

```
var defaults = {  
  container: "cesiumContainer", [必须]容器 id
```

```

openearth: false,[可选]开启场景
imageryprovider: "tianditu_image",[可选]gis 底图
openterrain: true,[可选]开启地形
};
var api = new SAPI(defaults);/api 对象后面调用接口要全程使用，控制好作用域

```

- 轻量化配置坐标信息

- a) 可以提供相关 EPSG 代号及项目基点坐标

上传 BIM 模型时将提供的坐标信息配置到 configJson 中

例：上传接口 input 参数： "ConfigJson":{

```
//.....省略其他参数.....
```

```
"srs": "EPSG:2437",//自定义 EPSG 代号
```

```
"srsOrigin": [563453.236,3563157.00354],//自定义项目基点坐标
```

```
}
```

- b) 无精确坐标,只能提供 GIS 场景中的某个坐标点

首先在上传 BIM 模型时将提供的坐标信息配置到 configJson 中

例：上传接口 input 参数： "ConfigJson":{

```
//.....省略其他参数.....
```

```
"longitude": 1.9003144895714261,//自定义弧度制经度值
```

```
"latitude": 0.59690260418206076, //自定义弧度制纬度值
```

```
}
```

然后调用引擎提供的偏移和旋转接口，每调一次接口都会返回模型当前最新的矩阵值，当调整好之后，把最后一次这个矩阵值记录下来(或业务存库)，在后边加载这个模型的时候，将该偏移值配置到加载模型接口的扩展参数中的 matrix 字段值里，最后重新加载模型就会按照你最终调整的那个位置去加载。

注意：配置了 srs、srsOrigin 和 longitude、latitude。 srs 的优先级高

- 加载倾斜数据或 BIM 模型

加载方式请参考“BIM 模型开发”和“GIS 场景开发”模块

5. 二次开发： 一般建议

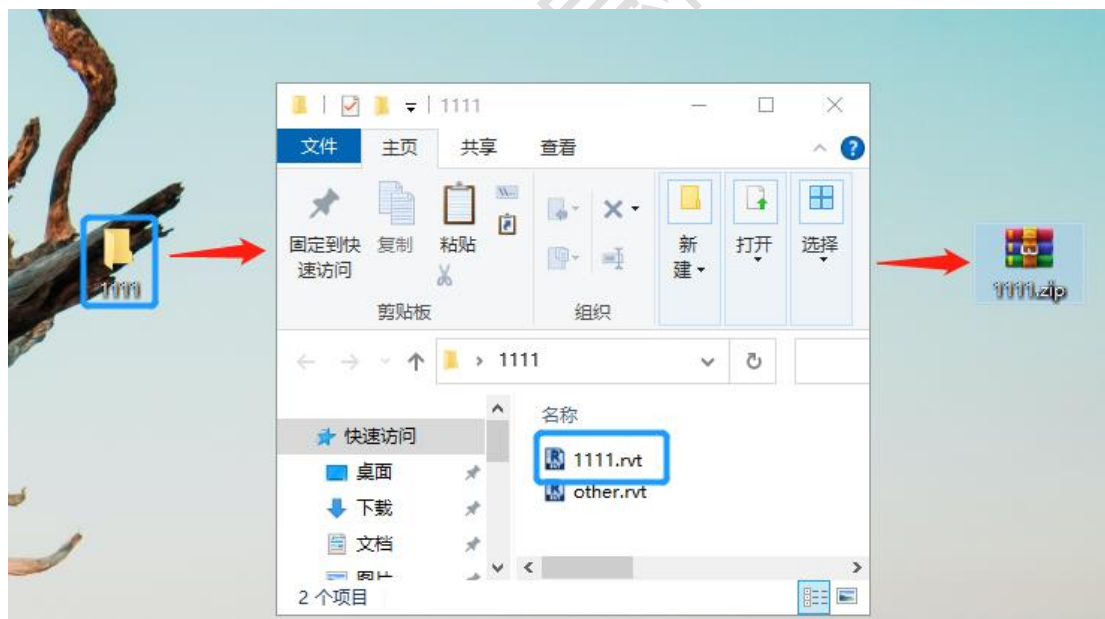
5.1 轻量化模型上传贴图方法

包名和主模型名称一致，名称为数字字母下划线组合，包内引入贴图及材质并在模型中更改绝对路径，再压缩成 zip 格式直接上传压缩包即可



5.2 链接模型压缩层级介绍

保持包名和主模型名称一致，名称为数字字母下划线组合，包内放入其它追加模型并在文件包内打开主模型连接好其它追加模型，再压缩成 zip 格式直接上传压缩包即可



5.3 OSGB 压缩层级介绍

[示例文件下载](#)

● osgb 标准文件的压缩方式

服务器端渲染 3DBIMGIS 轻量化融合引擎 | WebGL 轻量化 CAD 图纸引擎 | 智慧施工管理 (+GIS/BIM) 平台 | 设计成果 (+BIM/CAD) 在线交付平台

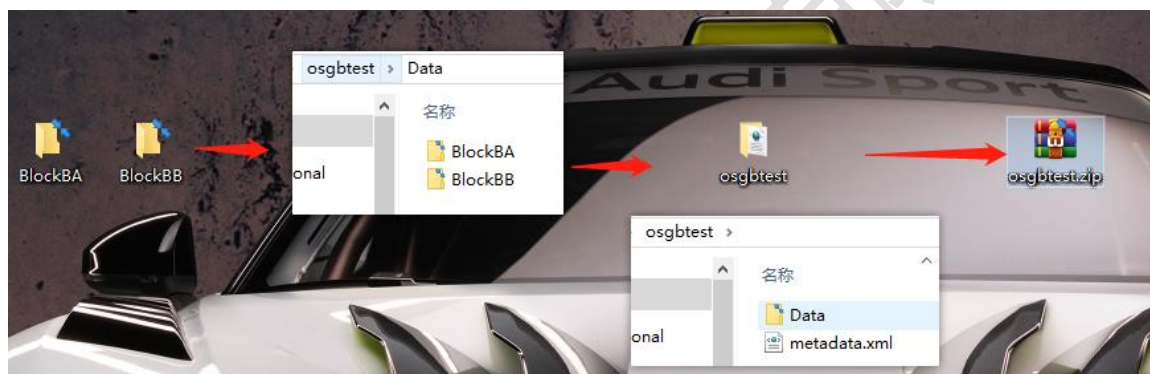
官网 www.glendale.top | 热线 029-81163590 | 邮箱 publish@glendale.top

Ps: 文件夹名为英文或英文数字的混排



- 其他数据压缩处理方式

部分用户的数据 为 BlockBA 等文件夹组成,压缩时先创建 Data 目录将这些文件夹放入 Data 目录,然后在创建一个包裹性文件夹,将 Data 和.xml 文件放入, 压缩为 zip 文件。



5.4 Max 源文件如何导出带贴图的 Fbx 格式

通过 3dMAX 软件将 Max 格式源文件打开后,并引入正确的贴图路径,再选择导出 Fbx 模型时注意勾选选项: **嵌入的媒体**再倒出即可

