

GEOVIS iCenter
空天大数据共享服务平台
V1.1.2
服务接口文档

中科星图股份有限公司

2020 年 1 月

目 录

| | |
|--------------------------------|----|
| 一、总体概述 | 1 |
| (一) Web 地图服务规范 | 1 |
| (二) 数据规范 | 3 |
| 1. GML-地理标记语言 | 3 |
| 2. GeoJson 数据规范 | 5 |
| (三) 坐标系统 | 8 |
| 二、datamanager 数据管理服务 | 8 |
| (一) 本地文件管理 | 8 |
| 1. 文件搜索 | 8 |
| 2. 文件夹搜索 | 10 |
| 3. 删除文件(夹) | 11 |
| 4. 获取文件详情 | 12 |
| 5. 获取文件路径 | 14 |
| 6. 获取文件目录 | 15 |
| 7. 查询图片 | 16 |
| (二) 服务器文件管理 | 17 |
| 1. 导入文件 | 17 |
| 2. 导入进度 | 18 |
| 3. 文件预览 | 18 |
| (三) 其他 | 19 |
| 1. 高级查询接口 | 19 |
| 2. SHP 上传 | 23 |
| 三、tiltphoto 倾斜摄影服务 | 24 |
| (一) 获取倾斜摄影列表 | 24 |
| (二) 通过文件名称下载文件 | 26 |
| 四、place 地名服务 | 26 |
| (一) 按地名标识查询 | 26 |
| (二) 按名称查询地名 | 27 |
| (三) 按位置查询地名 | 29 |
| (四) 查询地理编码 | 30 |
| (五) 获取地名数目总量 | 32 |
| (六) 获取批量地名数据 | 32 |
| (七) 获取世界行政区划 | 34 |
| (八) 获取省市县名称的树状结构 | 34 |
| (九) 获取省市县名称以及范围 | 35 |
| (十) 按全称查询地名 | 36 |
| 五、streetview 街景服务 | 37 |
| (一) 获取当前位置最近的街景点位编码 | 37 |
| (二) 由街景点位编码获取街景瓦片 | 38 |
| (三) 获取街景当前位置图像的属性信息 | 39 |
| (四) 获取向前/后/左/右操作的下一跳街景位置 | 41 |
| 六、GeoServerPlus 服务 | 42 |

| | |
|------------------------------------|----|
| （一）矢量数据服务 | 42 |
| 1. WMS 基础服务 | 42 |
| 2. WMS GetCapabilities 接口 | 43 |
| 3. WMTS 基础服务 | 48 |
| 4. WMTS GetCapabilities 接口 | 50 |
| 5. WFS 基础服务接口 | 55 |
| 6. WFS GetCapabilities 接口 | 56 |
| 7. TMS-按名称查询 TMS 服务元信息 | 57 |
| 8. TMS-获取所有 TMS 服务元信息 | 59 |
| 9. TMS- 获取指定图层、层、行、列的资源 | 61 |
| （二）影像数据服务 | 61 |
| 1. WMS 基础服务 | 61 |
| 2. WMS GetCapabilities 接口 | 62 |
| 3. TMS 基础服务 | 63 |
| 4. WMTS 基础服务 | 63 |
| 5. WMTS GetCapabilities 接口 | 64 |
| 七、GeoIot 实时数据服务 | 65 |
| （一）获取聚合数据 | 65 |
| （二）单目标查询 | 67 |
| （三）多目标查询 | 69 |
| （四）历史轨迹查询 | 70 |
| 八、geodata 服务 | 72 |
| （一）按数管编码查询图层信息 | 74 |
| （二）查询所有图层 | 77 |
| （三）按图层编码查询图层 | 81 |
| （四）删除图层 | 84 |
| （五）查询指定名称的图层 | 85 |
| （六）更新图层样式 | 88 |
| 九、tileCache 服务 | 89 |
| （一）基础影像 | 89 |
| 1. TMS 获取指定图层、层、行、列的资源 | 89 |
| 2. TMS 查询所有服务元信息 | 90 |
| 3. TMS 按名称查询服务元信息 | 90 |
| 4. WMS GetCapabilities 服务接口 | 91 |
| 5. WMTS GetCapabilities 服务接口 | 91 |
| 6. WMS 基础服务 | 91 |
| 7. WMTS 基础服务 | 93 |
| 8. 时序影像获取时间轴 | 93 |
| 9. 时序影像基础服务 | 94 |
| （二）图层管理接口 | 95 |
| 1. 按关键字查询图层信息 | 95 |
| 2. 查询指定编码的图层 | 97 |
| 3. 基础影像图层局部更新 | 99 |
| 4. 基础影像图层更新回滚 | 99 |

| | |
|------------------------------|-----|
| 5. 删除指定编码的图层 | 100 |
| （三）服务地址 | 101 |
| 1. 基础地形服务（待完善） | 101 |
| 十、 ServiceManager 系统服务 | 101 |
| （一）用户登录 | 101 |
| 1. 验证 token 的有效性 | 101 |
| 2. 用户登陆接口 | 102 |
| （二）lisence 服务（待完善） | 102 |

一、总体概述

（一）Web 地图服务规范

WMS 规范

Web Map Service（网络地图服务），简称 WMS，由开放地理信息联盟（Open GeoSpatial Consortium，OGC）制定。该规范定义了 Web 客户端从网络地图服务器获取地图的接口标准。一个 WMS 可以动态地生成具有地理参考数据的地图，这些地图通常用 GIF JPEG 或 PNG 等图像格式，或者 SVG、KML、VML 和 WebCGM 等矢量图形格式来表现。使用者通过指定的参数获取相应的地图图片。

WMS 实现规范由三个基础性操作协议(GetCapabilities, GetMap 和 GetFeatureInfo)组成，这些协议共同构成了利用 WMS 创建和叠加显示不同来源的远程异构地图服务的基础。

| 操作 | 实现要求 | 描述 |
|-----------------|------|--|
| GetCapabilities | 强制实现 | 获取 WMS 的能力文档（即元数据文档），里面包含服务的所有信息 |
| GetMap | 强制实现 | 获取地图图片。该操作根据客户端发出的请求参数在服务端进行检索，服务器端返回一个地图图像，其地理空间参数和大小参数是已经明确定义的，返回的地图图像可以是 GIF JPEG PNG 或 SVG 格式。 |
| GetFeatureInfo | 选择实现 | 该操作根据用户所请求的 X Y 坐标或感兴趣的图层，返回地图上某些特殊要素的信息，信息以 HTML，GML 或 ASCII 的格式表示。 |

WMTS 规范

Web Map Tile Service（网络地图瓦片服务），简称 WMTS，由开放地理信息联盟（Open GeoSpatial Consortium，OGC）制定，是和 WMS 并列的重要 OGC 规范之一。WMTS 不同于 WMS,它最重要的特征是采用缓存技术能够缓解 WebGIS 服务器端数据处理的压力，提高交互响应速度，大幅改善在线地图应用客户端的用户体验。WMTS 是 OGC 主推的缓存技术规范，是目前各种缓存技术相互兼容的一种方法。WMTS 服务支持 RESTful 访问，其接口包括 GetCapabilities、GetTile 和 GetFeatureInfo3 个操作，这些操作允许用户访问切片地图。

| 操作 | 实现要求 | 描述 |
|-----------------|------|--------------------------------|
| GetCapabilities | 强制实现 | 获取 WMTS 的能力文档（即元数据文档），里面包含服务的所 |

| 操作 | 实现要求 | 描述 |
|----------------|------|---|
| | | 有信息 |
| GetTile | 强制实现 | 获取地图瓦片。该操作根据客户端发出的请求参数在服务端进行检索，服务器端返回地图瓦片图像 |
| GetFeatureInfo | 选择实现 | 通过在 WMTS 图层上指定一定的条件，返回指定的地图瓦片内容对应的要素信息 |

WFS 规范

Web Feature Service（网络要素服务），简称 WFS，由开放地理信息联盟（Open GeoSpatial Consortium，OGC）制定。该规范主要对 OpenGIS 简单要素的数据编辑操作进行规范，从而使服务器端和客户端能够在要素层面进行“通讯”。其返回结果的 XML 格式的 WFS 服务元数据文档，通过该文档用户能够了解：WFS 服务器支持的所有操作列表，GetFeature 操作返回的数据格式，可用的坐标参照系统列表，操作异常信息的列表，WFS 服务提供商的相关信息，WFS 服务器的可用要素类列表等。

WFS 通过 GML（Geography Markup Language，地理标记语言）传递地理空间数据，它支持在基于 HTTP 协议对地理要素进行插入（INSERT）、更新（UPDATE）、删除（DELETE）和发现（DISCOVERY）等操作，并且在这些操作的过程中保证了地理数据变化的一致性。

WFS 服务接口规范定义了 GetCapabilities、DescribeFeatureType、GetFeature、Transaction、GetGmlObject 和 LockFeature 一共 6 种操作。其中，前三个 GetCapabilities、DescribeFeatureType 和 GetFeature 为必须实现的操作，也即只要实现了这三个操作的服务均可称为 WFS 服务。WFS 的操作见下表所示。

| 操作 | 实现要求 | 描述 |
|---------------------|------|---|
| GetCapabilities | 强制实现 | GetCapabilities 请求用于查询 WFS 服务的能力信息，包括支持的操作、支持的格式、空间坐标、包含的资源等。它主要的目的是使客户端在使用 GetFeature 请求前可以对 WFS 服务有一个基本的了解，从而可以设置正确的参数。 |
| DescribeFeatureType | 强制实现 | 用于生成一个 Schema 描述，该 Schema 描述了 WFS 服务提供的要素类型（Feature Type），以及要素类型的结构信息。该 Schema 还定义了 WFS 服务所期望的要素实例在输入时如何编码以及输出时如何生成要素实例。 |

| 操作 | 实现要求 | 描述 |
|--------------|------|--|
| GetFeature | 强制实现 | GetFeature 用于向 WFS 的客户端程序提供查询特定地理信息的能力，通过 GetFeature 操作可以由指定的属性条件 空间条件或者两者叠加的条件进行空间查询。 |
| Transaction | 选择实现 | 允许 Transaction 操作，使客户端可对服务器端所提供的地图要素类执行插入，更新，删除等命令 |
| GetGmlObject | 选择实现 | 通过 XLink 获取 GML 对象 |
| LockFeature | 选择实现 | 在事务过程中锁定要素 |

TMS 规范

TMS 与 WMTS 都是针对瓦片地图服务的推出的规范，两者的相似性大，主要差异如下：

- 由不同组织推进制定：WMTS 是 OGC 的标准（开放地理信息联盟，Open GeoSpatial Consortium，其主要目标是制定地理标准），TMS 是 OSGEO 的标准（OSGeo，开源空间信息基金会，是一个全球性非营利性组织，目标是支持全球性的合作,建立和推广高品质的空间信息开源软件。）
- 协议差异：TMS 是纯 RESTful 的；而 WMTS 可以有三种：KVP SOAP RESTful。
- 瓦片的组织方式：TMS 瓦片是正方形 WMTS 瓦片是矩形；在纵轴方向上面相反；WMTS 中对应的不同比例尺瓦片可以尺寸不同。

（二）数据规范

1. GML-地理标记语言

GML（OpenGIS® Geography Markup Language Encoding Standard）是一种基于 XML 的地理要素描述语言标准，用以在不同的软件或系统间交换空间数据，比如后面会介绍的 WFS 标准就使用 GML 作为输入和输出格式。GML 同时也是 ISO 标准。当前版本是 3.2.1，GML 3.0 版是对 GML 2.0 版的扩充，并且向后兼容。Schema 集合的组织具有了模块化特点，即用户能够有选择地使用所需部分，减化和缩小了执行的尺寸，提供了面向 WEB 应用 基于对象的地理数据描述语言。此外，3.0 版增加了对复杂的几何实体 拓扑 空间参照系统 元数据 时间特征和动态数据等的支持，使其更加适合描述现实世界问题，如基于位置服务的行程安排和高速公路设计等，GML 3.0 版新增的主要特性包括：

GML 标准其实就是通过 XML Schema (XSD) 来定义了 GML 文档的结构，这些定义都可以访问在线为：<http://schemas.opengis.net/gml/> 得到。目前 3.2.1 版本的 GML 中包含 7 个顶级 XSD (其中一个废弃类型，为了向前兼容)，这些 XSD 下还有其它子 XSD，它们组合成为如下图结构。



GML 封装了地理信息及其属性 **GML** 基于地理信息抽象模型，即空间实体特征及封装。地理特征（**Feature**）包括一系列的属性和相应的几何信息，一般来说属性由名字、类型和值组成，几何信息由基本元素如点、线、面、曲线、多边形等组成。目前 **GML** 主要局限在二维应用，正扩展到三维空间以及特征间的拓扑关系。**GML** 允许相当复杂的特征，如特征间的嵌套；例如飞机场由出租汽车道、飞机跑道等组成。

2) GML 示例

以下是一个封装建筑物的 **GML** 例子：

```
<Feature fid="142" featureType="school" Description="A middle school">
  <Polygon name="extent" srsName="epsg:27354">
    <LineString name="extent" srsName="epsg:27354">
      <CDATA>
        491888.999999459,5458045.99963358 491904.999999458,5458044.99963358
        491908.999999462,5458064.99963358 491924.999999461,5458064.99963358
        491925.999999462,5458079.99963359 491977.999999466,5458120.9996336
        491953.999999466,5458017.99963357 </CDATA>
      </LineString>
    </Polygon>
  </Feature>
```

2. GeoJson 数据规范

(1) GeoJson 规范概述

GeoJSON 是一种对各种地理数据结构进行编码的格式。**GeoJSON** 对象可以表示几何、特征或者特征集合。**GeoJSON** 支持下面几何类型：点、线、面、多点、多线、多面和几何集合。**GeoJSON** 里的特征包含一个几何对象和其他属性，特征集合表示一系列特征。

一个完整的 **GeoJSON** 数据结构总是一个（**JSON** 术语里的）对象。在 **GeoJSON** 里，对象由名/值对--也称作成员的集合组成。对每个成员来说，名字总是字符串。成员的值要么是字符串、数字、对象、数组，要么是下面文本常量中的一个：**"true"**、**"false"**和**"null"**。数组是由值是上面所说的元素组成。

GeoJSON 特征集合如下：

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
```

```

"geometry": {"type": "Point", "coordinates": [102.0, 0.5]},
"properties": {"prop0": "value0"}
},
{ "type": "Feature",
  "geometry": {
    "type": "LineString",
    "coordinates": [
      [102.0, 0.0], [103.0, 1.0], [104.0, 0.0], [105.0, 1.0]
    ],
    "properties": {
      "prop0": "value0",
      "prop1": 0.0
    }
  },
  { "type": "Feature",
    "geometry": {
      "type": "Polygon",
      "coordinates": [
        [100.0, 0.0], [101.0, 0.0], [101.0, 1.0],
        [100.0, 1.0], [100.0, 0.0] ]
      },
      "properties": {
        "prop0": "value0",
        "prop1": {"this": "that"}
      }
    }
  ]
}

```

(2) GeoJSON 对象

GeoJSON 总是由一个单独的对象组成。这个对象（指的是下面的 GeoJSON 对象）表示几何、特征或者特征集合。

GeoJSON 对象可能有任何数目成员（名/值对）。

- GeoJSON 对象必须由一个名字为"type"的成员。这个成员的值是由 GeoJSON 对象的类型所确定的字符串。

- **type** 成员的值必须是下面之一: "Point", "MultiPoint", "LineString", "MultiLineString", "Polygon", "MultiPolygon", "GeometryCollection", "Feature", 或者 "FeatureCollection"。
- GeoJSON 对象可能有一个可选的"crs"成员, 它的值必须是一个坐标参考系统的对象(见坐标参考系统对象)。
- GeoJSON 对象可能有一个"bbox"成员, 它的值必须是边界框数组

几何对象: 几何对象是一种 GeoJSON 对象, 这时 **type** 成员的值是下面字符串之一: "Point", "MultiPoint", "LineString", "MultiLineString", "Polygon", "MultiPolygon", 或者 "GeometryCollection"。除了 "GeometryCollection" 外的其他任何类型的 GeoJSON 几何对象必须由一个名字为 "coordinates" 的成员。**coordinates** 成员的值总是数组。这个数组里的元素的结构由几何类型来确定。

位置: 位置是基本的几何结构。几何对象的 "coordinates" 成员由一个位置(这儿是几何点)、位置数组(线或者几何多点), 位置数组的数组(面、多线)或者位置的多维数组(多面)组成。

位置由数字数组表示。必须至少两个元素, 可以有更多元素。元素的顺序必须遵从 **x,y,z** 顺序(投影坐标参考系统中坐标的东向、北向、高度或者地理坐标参考系统中的坐标长度、纬度、高度)。

GeoJSON 几何对象包括:

点: "Point", "coordinates" 成员必须是一个单独的位置。

多点: "MultiPoint", "coordinates" 成员必须是位置数组。

线: "LineString", "coordinates" 成员必须是两个或者多个位置的数组。

线性环是具有 4 个或者更多位置的封闭的线。第一个和最后一个位置是相等的(它们表示相同的点)。虽然线性环没有鲜明地作为 GeoJSON 几何类型, 不过在面几何类型定义里有提到它。

多线: "MultiLineString" 来说, "coordinates" 成员必须是一个线坐标数组的数组。

面: "Polygon", "coordinates" 成员必须是一个线性环坐标数组的数组。对拥有多个环的面来说, 第一个环必须是外部环, 其他的必须是内部环或者孔。

多面: "MultiPolygon", "coordinates" 成员必须是面坐标数组的数组。

几何集合: 类型为 "GeometryCollection" 的 GeoJSON 对象是一个集合对象, 它表示几何对象的集合。几何集合必须有一个名字为 "geometries" 的成员。与 "geometries" 相对应的值是一个数组。这个数组中的每个元素都是一个 GeoJSON 几何对象。

（三）坐标系统

目前 Web 地图数据发布支持三种坐标系统，分别为：

EPSG:4326，代表 wgs84 坐标系；（地理坐标系统）

EPSG:4490，代表 cscs2000 地理坐标系；（地理坐标系统）

EPSG:3857，Web Mercator 投影坐标系，也就是现在我们常用的 Web 地图的坐标系，并且给定官方命名“WGS 84 / Pseudo-Mercator”。经常用于在 Google 地图，OpenStreetMap 等中呈现地图的投影坐标系，国内的互联网地图百度，高德，腾讯也是这一坐标系，在做地图发布时在这个坐标系基础上做了加密或偏移处理。

二、datamanager 数据管理服务

（一）本地文件管理

1. 文件搜索

简要描述：获取指定父目录下的含有某文字的文件信息

请求 URL：/datamanager/api/v1/folders

请求说明：需要请求的 header 中包含 Authorization 信息

请求方式：GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|-------------|----|--------|--------------|
| page | 是 | int | 当前页，默认是 1 |
| pageSize | 是 | int | 每页条目数，默认是 50 |
| parentId | 否 | string | 父目录 id |
| name | 否 | long | 搜索关键字 |
| orderColumn | 否 | string | 排序列名 |
| orderType | 否 | string | 排序方式 |

请求示例：<http://192.168.4.17:8310/datamanager/api/v1/folders?currentPage=1&pageSize=50&orderColumn=createTime&orderType=desc>

返回示例

```
{
  "total": 2,
  "pageSize": 50,
  "page": 1,
  "items": [
    {
```

```

        "id": 144362,
        "createTime": "2019-09-26 20:37:41",
        "updateTime": "2019-09-26T12:37:41.045+0000",
        "fileName": "123",
        "dataType": null,
        "fileSize": 0,
        "md5Num": null,
        "extName": null,
        "serviceName": null,
        "operationTypeEn": null,
        "analysisStatus": null,
        "pId": -1,
        "pName": "全部文件",
        "dir": true,
        "display": true,
        "empty": true
    },
    {
        "id": 11,
        "createTime": "2019-09-24 16:37:05",
        "updateTime": "2019-09-24T08:37:10.981+0000",
        "fileName": "taiwan.tif",
        "dataType": "image",
        "fileSize": 72120693,
        "md5Num": "278b4c3234fa4296318ec6961c80d059",
        "extName": "tif",
        "serviceName": "成果影像",
        "operationTypeEn": "parsefile,publishfile",
        "analysisStatus": "2",
        "pId": -1,
        "pName": "全部文件",
        "dir": false,
        "display": true,
        "empty": true
    }
]
}

```

返回参数说明

| 参数名 | 类型 | 说明 |
|------------|--------|------------|
| total | int | 总个数 |
| pageSize | int | 每页的大小 |
| page | int | 当前页码 |
| items | | 返回的列表 |
| id | int | 文件或者目录的 id |
| createTime | Date | 创建时间 |
| updateTime | Date | 更新时间 |
| fileName | string | 文件名称 |

| 参数名 | 类型 | 说明 |
|-----------------|---------|--|
| dataType | string | 数据类型，目录为 null，文件是 image, vector, originalimage 等 |
| fileSize | int | 文件大小，目录为 null |
| md5Num | string | 文件的 md5，目录为 null |
| extName | string | 文件后缀名，可以是 tif, zip 等，目录为 null。 |
| serviceName | string | 文件是矢量，地形，成果影像等，目录为 null。 |
| operationTypeEn | string | 支持的操作方式，包括 parsefile, publishfile 等，目录为 null。 |
| analysisStatus | string | 解析的状态：1：解析中 2：解析成功 3：解析失败，目录为 null。 |
| pId | string | 父目录 id |
| pName | string | 父目录名称 |
| dir | boolean | true：文件夹，false：文件 |
| display | boolean | true：显示，false：不显示，用来多文件展示主文件 |
| empty | boolean | true：没有下级文件，false：有下级文件 |

2. 文件夹搜索

简要描述：获取某目录下的所有的子文件夹

请求 URL：/datamanager/api/v1/folders/{parentId}/folders

请求说明：需要请求的 header 中包含 Authorization 信息

请求方式：GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|----------|----|-----|--------|
| parentId | 是 | int | 目录的 id |

请求示例

http://192.168.4.17:8310/datamanager/api/v1/folders/-1/folders

返回示例

```
[
  {
    "id": 144362,
    "createTime": "2019-09-26 20:37:41",
    "updateTime": "2019-09-26T12:37:41.045+0000",
    "fileName": "123",
    "dataType": null,
```

```

    "fileSize": 0,
    "md5Num": null,
    "extName": null,
    "serviceName": null,
    "operationTypeEn": null,
    "analysisStatus": null,
    "pId": -1,
    "pName": "全部文件",
    "dir": true,
    "display": true,
    "empty": true
  }
]

```

返回参数说明

| 参数名 | 类型 | 说明 |
|-----------------|---------|--|
| id | int | 文件或者目录的 id |
| createTime | Date | 创建时间 |
| updateTime | Date | 更新时间 |
| fileName | string | 文件名称 |
| dataType | string | 数据类型，目录为 null，文件是 image, vector, originalimage 等 |
| fileSize | int | 文件大小，目录为 null |
| md5Num | string | 文件的 md5，目录为 null |
| extName | string | 文件后缀名，可以是 tif, zip 等，目录为 null。 |
| serviceName | string | 文件是矢量，地形，成果影像等，目录为 null。 |
| operationTypeEn | string | 支持的操作方式，包括 parsefile, publishfile 等，目录为 null。 |
| analysisStatus | string | 解析的状态：1：解析中 2：解析成功 3：解析失败，目录为 null。 |
| pId | string | 父目录 id |
| pName | string | 父目录名称 |
| dir | boolean | true：文件夹，false：文件 |
| display | boolean | true：显示，false：不显示，用来多文件展示主文件 |
| empty | boolean | true：没有下级文件，false：有下级文件 |

3. 删除文件(夹)

简要描述：删除文件(夹)。

请求 URL：/datamanager/api/v1/folders

请求说明：需要请求的 header 中包含 Authorization 信息

请求方式：DELETE

参数：

| 参数名 | 必选 | 类型 | 说明 |
|-----|----|------|-------------------------|
| ids | 是 | list | 目录 id, {"ids": [0,1,2]} |

请求示例

<http://192.168.4.38:8319/datamanager/api/v1/folders>

返回示例：

删除成功：返回 null；

删除失败：返回占用文件列表；

返回参数示例：

["World111-PNG-4326", "World123433-PNG-4326"]

4. 获取文件详情

简要描述：获取文件详情

请求 URL：/datamanager/api/v1/files/FileDetails

请求说明：需要请求的 header 中包含 Authorization 信息

请求方式：POST

参数：

| 参数名 | 必选 | 类型 | 说明 |
|-----|----|------|-------|
| id | 是 | long | 文件 id |

请求示例

<http://192.168.4.17:8310/datamanager/api/v1/files/FileDetails?id=144363>

返回示例

```
{
  "children": [],
  "analysisBasicDatas": [
    {
      "id": 2,
      "createTime": "2019-09-26 20:38:29",
      "updateTime": "2019-09-26T12:38:29.717+0000",
    }
  ]
}
```



```

      "geo": "POLYGON ((108.517364418026 20.233551754187, 111.13403108678601
20.233551754187, 111.13403108678601 18.063273974673002, 108.517364418026
18.063273974673002, 108.517364418026 20.233551754187))",
      "typeEnName": "image",
      "fileName": "海南 dem1",
      "dataEntityId": 144363,
      "info": {
        "geo": "POLYGON ((108.517364418026 20.233551754187,
111.13403108678601 20.233551754187, 111.13403108678601 18.063273974673002,
108.517364418026 18.063273974673002, 108.517364418026 20.233551754187))",
        "format": "tif",
        "gridSet": "EPSG:4326",
        "dataType": "成果影像",
        "fileName": "海南 dem1",
        "bandCount": 1,
        "upLeftLat": 20.233551754187,
        "upLeftLon": 108.517364418026,
        "lowLeftLat": 18.063273974673002,
        "lowLeftLon": 108.517364418026,
        "resolution": 30.83333335800005,
        "upRightLat": 20.233551754187,
        "upRightLon": 111.13403108678601,
        "lowRightLat": 18.063273974673002,
        "lowRightLon": 111.13403108678601,
        "rasterXSize": 9420,
        "rasterYSize": 7813,
        "dataEntityId": 144363,
        "iconFileName": "largeThumb.png",
        "iconRelativePath": "/parseData/2019/9/26/144363/海南
dem1/largeThumb.png",
        "thumbnailFileName": "smallThumb.png",
        "thumbnailRelativePath": "/parseData/2019/9/26/144363/海南
dem1/smallThumb.png"
      }
    },
    "filePath": "/home/iCenter/data/service-data/datamanager/dir/0/123/海南
dem1.tif",
    "dataEntityId": 144363,
    "status": "2"
  }
}

```

返回参数说明

| 参数名 | 类型 | 说明 |
|--------------------|----------|---------|
| children | List | 附属文件列表 |
| analysisBasicDatas | | 解析信息 |
| id | long | 解析信息 id |
| createTime | Date | 创建时间 |
| updateTime | Date | 更新时间 |
| geo | Geometry | 地理信息 |

| 参数名 | 类型 | 说明 |
|-----------------------|----------|--------------------------|
| typeEnName | string | 文件类型的英文名 |
| fileName | string | 文件名称 |
| dataEntityId | long | 原始文件 id |
| info | | 地理解析信息 |
| geo | Geometry | 地理信息 |
| format | string | 类型后缀名 |
| bandCount | int | 等级数 |
| upLeftLat | double | 左上纬度 |
| upLeftLon | double | 左上经度 |
| lowLeftLat | double | 左下纬度 |
| lowLeftLon | double | 左下经度 |
| resolution | double | 分辨率 |
| upRightLat | double | 右上纬度 |
| upRightLon | double | 右上经度 |
| lowRightLat | double | 右下纬度 |
| lowRightLon | double | 右下经度 |
| rasterXSize | int | X 光栅 |
| rasterYSize | int | Y 光栅 |
| dataEntityId | long | 原始文件 id |
| iconFileName | string | 缩略图文件名称 |
| iconRelativePath | string | 缩略图文件相对路径 |
| thumbnailFileName | string | 拇指图文件名称 |
| thumbnailRelativePath | string | 拇指图文件相对路径 |
| filePath | string | 原始文件路径 |
| dataEntityId | long | 原始文件 id |
| status | int | 解析状态：1：解析中 2：解析成功 3：解析失败 |

5. 获取文件路径

简要描述：获取文件路径

请求 URL: /datamanager/api/v1/files/{dataEntityId}/pathRoad

请求说明: 需要请求的 header 中包含 Authorization 信息

请求方式: GET

请求参数:

| 参数名 | 必选 | 类型 | 说明 |
|--------------|----|-----|-------|
| dataEntityId | 是 | int | 文件 id |

请求示例

<http://192.168.4.17:8310/datamanager/api/v1/files/144363/pathRoad>

返回结果说明:

```
/home/iCenter/data/service-data/datamanager/dir/0/123/海南 dem1.tif
```

返回参数说明:

文件在 docker 环境中的路径

6. 获取文件目录

简要描述: 获取文件目录信息

请求 URL: /datamanager/api/v1/files/{dataEntityId}/folders

请求说明: 需要请求的 header 中包含 Authorization 信息

请求方式: GET

请求参数:

| 参数名 | 必选 | 类型 | 说明 |
|--------------|----|-----|-------|
| dataEntityId | 是 | int | 文件 id |

请求示例

<http://192.168.4.17:8310/datamanager/api/v1/files/144363/folders>

返回示例

```
{
  "id": 12,
  "createTime": "2019-09-27 09:22:42",
  "updateTime": "2019-09-27T01:22:43.061+0000",
  "fileName": "World.tif",
  "dataType": "image",
  "fileSize": 25167244,
  "md5Num": "d9fb58fc86597f1bebef5a1e87aff0aa",
  "extName": ".tif",
  "serviceName": "成果影像",
  "operationTypeEn": "parsefile,publishfile",
  "analysisStatus": "2",
  "pId": -1,
  "pName": "全部文件",
  "display": true,
  "dir": false,
```

```
"empty": true
}
```

返回参数说明

| 参数名 | 类型 | 说明 |
|-----------------|---------|--|
| id | long | 文件 id |
| createTime | Date | 创建时间 |
| updateTime | Date | 更新时间 |
| fileName | string | 文件名称 |
| dataType | string | 数据类型，目录为 null，可以是 image,vector,originalimage 等 |
| fileSize | long | 文件大小，目录为 null |
| md5Num | string | 文件的 md5，目录为 null |
| extName | string | 文件后缀名，可以是 tif, zip 等，目录为 null。 |
| serviceName | string | 文件是矢量，地形，成果影像等，目录为 null。 |
| operationTypeEn | string | 支持的操作方式，包括 parsefile,publishfile 等，目录为 null。 |
| analysisStatus | string | 解析的状态：1：解析中 2：解析成功 3：解析失败，目录为 null。 |
| pId | string | 父目录 id |
| pName | string | 父目录名称 |
| display | boolean | true：显示，false：不显示，用来多文件展示主文件 |
| dir | boolean | true：文件夹，false：文件 |
| empty | boolean | true：没有下级文件，false：有下级文件 |

7. 查询图片

简要描述：查询图片

请求 URL：/datamanager/api/v1/iconImage

请求方式：GET

请求参数：

| 参数名 | 类型 | 说明 |
|------|--------|------|
| path | string | 文件路径 |

请求示例：

<http://192.168.4.17:8310/datamanager/api/v1/iconImage?path=/parseData/2019/9/27/7/World/largeThumb.png>

返回示例

图片文件。

返回参数说明

无

(二) 服务器文件管理

1. 导入文件

简要描述：从服务器导入文件

请求 URL：/datamanager/api/v1/server/fileImport

请求说明：需要请求的 header 中包含 Authorization 信息

请求方式：POST

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|----------|----|--------|----------------|
| filePath | 是 | string | 文件路径 |
| destPath | 是 | string | 目标路径 |
| fileType | 是 | string | 文件类型 |
| parentId | 是 | long | 文件的父目录 Id |
| status | 是 | string | 导入模式：1 剪切；2 复制 |

请求示例：

<http://192.168.4.85:8310/datamanager/api/v1/server/fileImport>

requestbody 写入下面的 json：

```
{
  "filePath": "/home/iCenter/data/import-data/OriginalImage/JL101A_PMS_20161113080850_000015636_202_0011_001_L1_MSS.tar.gz",
  "destPath": "/home/iCenter/data/service-data/datamanager/dir/0/yuanshi",
  "fileType": "originalImage",
  "parentId": 3,
  "status": "2"
}
```

返回示例：

```
{
  "id": 16,
  "resultInfo": "导入成功"
}
```

返回参数说明

| 参数名 | 类型 | 说明 |
|------------|--------|-----------------|
| id | int | 导入文件后在文件管理中的 id |
| resultInfo | string | 导入结果：导入成功；导入失败 |

2. 导入进度

简要描述：从服务器导入的进度

请求 URL：/datamanager/api/v1/server/fileImportProgress

请求说明：需要请求的 header 中包含 Authorization 信息

请求方式：GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|------------|----|--------|------|
| sourcePath | 是 | string | 源地址 |
| targetPath | 是 | string | 目的地址 |
| status | 是 | string | 状态 |

http://192.168.4.85:8310/datamanager/api/v1/server/fileImportProgress?sourcePath=/home/iCenter/data/import-data/OriginalImage/JL101A_PMS_20161113080850_000015636_202_0011_001_L1_MSS.tar.gz&targetPath=/home/iCenter/data/service-data/datamanager/realfile&status=2

返回示例

```
{"progress": "13.871952747467638"}
```

返回参数说明

| 参数名 | 类型 | 说明 |
|----------|--------|--------------|
| progress | double | 进度：0.0–100.0 |

3. 文件预览

简要描述：通过指定文件 id 进行预览, 只能对图片和文档进行预览

请求 URL：/datamanager/api/v1/filePreview/{dataEntityId}?gateway=gateway

请求说明：需要请求的 header 中包含 Authorization 信息

请求方式：GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|--------------|----|---------|------------|
| dataEntityId | 是 | integer | 网盘对应的文件 id |
| gateway | 是 | string | 是否是网关访问 |

请求示例

```
http://192.168.4.85:8310/datamanager/api/v1/filePreview/9?gateway=
```

返回示例

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

<head>
  <base id="base" href="/datamanager">
  <meta charset="utf-8">
  <title>Excel 预览</title>
  <style>
    html,
    body {
      width: 100%;
      height: 100%;
      margin: 0;
      padding: 0;
    }
  </style>
</head>

<body>
  <script>
    window.location.href =
    encodeURI("/datamanager/preview/71100f4337d1e2ec7c0001eb66536178_瓦片格式.html");
  </script>
</body>
```

（三）其他

1. 高级查询接口

简要描述：高级查询接口

请求 URL：/datamanager/api/v1/advancedSearch

请求说明：需要请求的 header 中包含 Authorization 信息

请求方式：POST

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|--------------|----|---------|------|
| cloudPercent | 否 | integer | 云量 |
| currentPage | 是 | integer | 当前页 |
| dataType | 是 | string | 数据类型 |

| 参数名 | 必选 | 类型 | 说明 |
|---------------|----|---------|--------|
| geo | 否 | string | 地理空间类型 |
| keyWordName | 否 | string | 关键词名 |
| level | 否 | string | 产品等级 |
| maxResolution | 否 | double | 最高分辨率 |
| pageSize | 否 | integer | 页面容量 |
| satelliteId | 否 | string | 卫星型号 |
| sensorId | 否 | string | 传感器类型 |
| timeIntervals | 是 | 数组 | 时间间隔 |
| start | 是 | date | 起始时间 |
| end | 是 | date | 结束时间 |
| propertyName | 是 | string | 属性名称 |

请求示例

```
{
  "geo": "POLYGON((-180 -90, -180 90, 180 90, 180 -90, -180 -90))",
  "minResolution": 0,
  "dataType": "image",
  "pageSize": 13,
  "currentPage": 1,
  "timeIntervals": [
    {
      "propertyName": "startTime",
      "start": 1569427200000,
      "end": 1572364799000
    }
  ]
}
```

返回示例

```
{
  "total": 3,
  "pageSize": 13,
  "page": 1,
  "items": [
    {
      "id": 8,
      "level": null,
      "upLeftLat": 22.236328125,
      "upLeftLon": 113.48876953125,
      "lowLeftLat": 22.08251953125,
      "lowLeftLon": 113.48876953125,
      "upRightLat": 22.236328125,
      "upRightLon": 113.62060546875,
      "lowRightLat": 22.08251953125,
      "lowRightLon": 113.62060546875,
      "region": null,
      "sensorId": null,
      "format": "tif",
      "resolution": 9.527206420898438,
      "satelliteId": null,
      "dataType": "image",
      "fileName": "2",
      "bandCount": 3,
      "rasterXSize": 1536,
      "rasterYSize": 1792,
    }
  ]
}
```



```

        "gridSet": "EPSG:4326",
        "description": null,
        "gatherTime": null,
        "cloudPercent": null,
        "dataEntityId": 15,
        "iconRelativePath": "/parseData/2019/9/27/15/2/largeThumb.png",
        "thumbnailRelativePath": "/parseData/2019/9/27/15/2/smallThumb.png",
        "downloadids": "15",
        "createTime": "2019-09-27 10:35:50",
        "geo": "POLYGON ((113.48876953125 22.236328125, 113.62060546875
22.236328125, 113.62060546875 22.08251953125, 113.48876953125 22.08251953125,
113.48876953125 22.236328125)))"
    },
    {
        "id": 5,
        "level": null,
        "upLeftLat": 90.0,
        "upLeftLon": -180.0,
        "lowLeftLat": -90.0,
        "lowLeftLon": -180.0,
        "upRightLat": 90.0,
        "upRightLon": 180.0,
        "lowRightLat": -90.0,
        "lowRightLon": 180.0,
        "region": null,
        "sensorId": null,
        "format": "tif",
        "resolution": 9755.859375,
        "satelliteId": null,
        "dataType": "image",
        "fileName": "World",
        "bandCount": 3,
        "rasterXSize": 4096,
        "rasterYSize": 2048,
        "gridSet": "EPSG:4326",
        "description": null,
        "gatherTime": null,
        "cloudPercent": null,
        "dataEntityId": 12,
        "iconRelativePath": "/parseData/2019/9/27/7/World/largeThumb.png",
        "thumbnailRelativePath": "/parseData/2019/9/27/7/World/smallThumb.png",
        "downloadids": "12",
        "createTime": "2019-09-27 09:22:43",
        "geo": "POLYGON ((-180 90, 180 90, 180 -90, -180 -90, -180 90)))"
    },
    {
        "id": 3,
        "level": null,
        "upLeftLat": 90.0,
        "upLeftLon": -180.0,
        "lowLeftLat": -90.0,
        "lowLeftLon": -180.0,
        "upRightLat": 90.0,
        "upRightLon": 180.0,
        "lowRightLat": -90.0,
        "lowRightLon": 180.0,

```

```

        "region": null,
        "sensorId": null,
        "format": "tif",
        "resolution": 9755.859375,
        "satelliteId": null,
        "dataType": "image",
        "fileName": "World",
        "bandCount": 3,
        "rasterXSize": 4096,
        "rasterYSize": 2048,
        "gridSet": "EPSG:4326",
        "description": null,
        "gatherTime": null,
        "cloudPercent": null,
        "dataEntityId": 7,
        "iconRelativePath": "/parseData/2019/9/27/7/World/largeThumb.png",
        "thumbnailRelativePath": "/parseData/2019/9/27/7/World/smallThumb.png",
        "downloadids": "7",
        "createTime": "2019-09-27 09:17:18",
        "geo": "POLYGON ((-180 90, 180 90, 180 -90, -180 -90, -180 90))"
    }
}
}

```

返回参数说明:

| 参数名 | 类型 | 说明 |
|-------------|--------|------------|
| total | int | 总个数 |
| pageSize | int | 每页的大小 |
| page | int | 当前页码 |
| items | | 返回的列表 |
| id | int | 文件或者目录的 id |
| level | int | 层级 |
| createTime | Date | 创建时间 |
| upLeftLat | double | 左上维度 |
| upLeftLon | double | 左上经度 |
| lowLeftLat | double | 左下维度 |
| lowLeftLon | double | 左上经度 |
| upRightLat | double | 右上维度 |
| upRightLon | double | 右上经度 |
| lowRightLat | double | 右下维度 |
| lowRightLon | double | 右下经度 |

| 参数名 | 类型 | 说明 |
|-----------------------|----------|----------|
| region | string | 区域 |
| sensorId | int | 传感器 id |
| format | string | 文件格式 |
| resolution | double | 分辨率 |
| sattlliteId | int | 卫星 id |
| dataType | string | 数据类型 |
| flieName | string | 文件名称 |
| bandCount | int | 等级数 |
| rasterXSize | int | X 光栅 |
| rasterYSize | int | Y 光栅 |
| gridSet | string | grid 级 |
| description | string | 描述 |
| gatherTime | Date | 采集时间 |
| cloudPercent | string | 云量 |
| dataEntityId | long | 文件 id |
| iconRelativePath | string | 缩略图相对路径 |
| thumbnailRelativePath | string | 拇指图相对路径 |
| downloadids | string | 下载文件的 id |
| createTime | Date | 创建时间 |
| geo | geometry | 地理信息 |

2. SHP 上传

简要描述：SHP 上传

请求 URL：/datamanager/api/v1/shpUpload

请求说明：需要请求的 header 中包含 Authorization 信息

请求方式：POST

请求参数：

Content-Disposition: form-data; name="file";

filename="第三个全球河流.zip"

Content-Type: application/octet-stream

返回示例

```
MULTIPOLYGON (((113.59269714355469 6.943723201751709, 113.59770202636719
6.943723201751709, 113.59770202636719 6.938203811645508, 113.59269714355469
6.938203811645508, 113.59269714355469 6.943723201751709)), ((113.24459838867188
6.339902877807617, 113.25080108642578 6.339902877807617, 113.25080108642578
6.334391117095947, 113.24459838867188 6.334391117095947, 113.24459838867188
6.339902877807617)), ((113.22969818115234 6.323439121246338, 113.23639678955078
6.323439121246338, 113.23639678955078 6.318641185760498, 113.22969818115234
6.318641185760498, 113.22969818115234 6.323439121246338)))
```

返回参数说明:

返回 GeoJson 数据格式。

三、tiltphoto 倾斜摄影服务

(一) 获取倾斜摄影列表

简要描述: 获取倾斜摄影列表

请求 URL: /tiltphoto/api/v1/layers

请求方式: GET

参数:

| 参数名 | 必选 | 类型 | 说明 |
|-----------|----|--------|-------------|
| page | 是 | int | 当前页, 从 1 开始 |
| pageSize | 是 | int | 每页的条目数 |
| layerName | 是 | string | 搜索关键字 |
| order | 否 | string | 排序列名 |
| orderType | 否 | string | 排序类型 |

请求示例

http://192.168.4.38:8319/tiltphoto/api/v1/layers?layerName=8d393ce8f3c194e975305d6876f7e8c8&order=createTime&orderType=desc&page=1&pageSize=50

返回示例

```
{
  "total": 35,
  "data": [
    {
      "id": 921040,
      "layerName": "8d393ce8f3c194e975305d6876f7e8c8",
```

```

"description": null,
"thumb": null,
"format": "B3DM",
"longitude": 0,
"latitude": 0,
"altitude": 0,
"createTime": "2018-11-29 19:04:50",
"fileSize": 0,
"url": "http://localhost:8230/api/v1/layers/dataset/8d393ce8f3c194e975305d6876f7e8c8",
"dmId": null,
"userId": null
} .....
]
}

```

返回参数说明

| 参数名 | 类型 | 说明 |
|-------------|--------|-----------|
| data | array | 列表数据 |
| total | int | 列表中总的条目数 |
| id | Long | 主键 |
| layerName | String | 图层名称 |
| description | String | 图层名称 |
| thumb | String | 缩略图 |
| format | String | 数据格式 |
| longitude | double | 中心点经度 |
| latitude | double | 中心点纬度 |
| altitude | double | 中心点高度 |
| createTime | Date | 创建时间 |
| fileSize | float | 文件大小 |
| url | String | 文件存储路径 |
| dmId | String | dm 库中的 id |
| userId | String | 用户 id |

（二）通过文件名称下载文件

简要描述：通过文件名称下载倾斜摄影存储的文件格式。

请求 URL：/tiltphoto/api/v1/layers/dataset/{dataLayerName}

请求方式：GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|---------------|----|--------|----|
| dataLayerName | 是 | String | 名称 |

请求示例

http://192.168.4.38:8319/tiltphoto/api/v1/layers/dataset/8d393ce8f3c194e975305d6876f7e8c8/lab_a_003_002_L18_000.json

返回示例

```
lab_a_003_002_L18_000.json 这个文件
```

返回参数说明

| 参数名 | 类型 | 说明 |
|------|----|--------|
| byte | 文件 | 下载一个文件 |

四、place 地名服务

（一）按地名标识查询

简要描述：按地名标识查询接口

请求 URL：/placeservice/api/v1/query/dmbs

请求方式：GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|------|----|--------|----|
| dmbs | 是 | string | 标识 |

请求示例

http://192.168.4.38:8319/placeservice/api/v1/query/dmbs=6fc18a70-7888-4337-95c4-4e1628860535

返回示例

```
[{
  "DMBS": "c1309542-b29b-490c-a247-f275eef47aab",
  "DMCJBM": 4,
  "DMMC": "北京市",
```

```

"WZ": "POINT(116.410323 40.18498)",
"MS": "sheng",
"DLBM": 500000230000052,
"DMCCD": 3,
"DMJP": "beijingshi",
"DMQP": "",
"DMBM": null,
"BDY": null,
"YWMC": null,
"DMDZ": null,
"DMGC": 0,
"LV": 5
}]

```

返回参数说明

| 参数名 | 类型 | 说明 |
|--------|--------|-------|
| DMBS | String | 标识 |
| DMCJBM | String | 层级编码 |
| DMMC | String | 名称 |
| WZ | String | 位置 |
| MS | String | 描述 |
| DLBM | String | 地理编码 |
| DMCCD | String | DMCCD |
| DMJP | String | 简拼 |
| DMQP | String | 全拼 |
| DMBM | String | 编码 |
| BDY | String | DBY |
| YWMC | String | YWMC |
| DMDZ | String | 地址 |
| DMGC | String | DMGC |
| LV | int | 层级 |

(二) 按名称查询地名

简要描述：按地名查询接口

请求 URL：/placeservice/api/v1/query/placeName={name}

请求方式：GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|-----------|----|--------|----|
| placeName | 是 | string | 地名 |

请求示例

http://192.168.4.38:8319/placeservice/api/v1/query/placeName=北京

返回示例

```
[{
  "DMBS": "c1309542-b29b-490c-a247-f275eef47aab",
  "DMCJBM": 4,
  "DMMC": "北京市",
  "WZ": "POINT(116.410323 40.18498)",
  "MS": "sheng",
  "DLBM": 5000002300000052,
  "DMMCCD": 3,
  "DMJP": "beijingshi",
  "DMQP": "",
  "DMBM": null,
  "BDY": null,
  "YWMC": null,
  "DMDZ": null,
  "DMGC": 0,
  "LV": 5
}, ... ]
```

返回参数说明

| 参数名 | 类型 | 说明 |
|--------|--------|------|
| DMBS | String | 标识 |
| DMCJBM | String | 层级编码 |
| DMMC | String | 名称 |
| WZ | String | 位置 |

| 参数名 | 类型 | 说明 |
|-------|--------|-------|
| MS | String | 描述 |
| DLBM | String | 地理编码 |
| DMCCD | String | DMCCD |
| DMJP | String | 简拼 |
| DMQP | String | 全拼 |
| DMBM | String | 编码 |
| BDY | String | DBY |
| YWMC | String | YWMC |
| DMDZ | String | 地址 |
| DMGC | String | DMGC |
| LV | int | 层级 |

（三）按位置查询地名

简要描述：查询位置接口

请求 URL：/placeservice/api/v1/query/location

请求方式：GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|-----|----|--------|----|
| lon | 是 | String | 经度 |
| lat | 是 | String | 纬度 |

请求示例

```
{
lon: "116.46",
lat: "39.92"
}
```

返回示例

```
[{
"DMBS": "c1309542-b29b-490c-a247-f275eef47aab",
"DMCJBM": 4,
"DMMC": "北京市",
"WZ": "POINT(116.410323 40.18498)",
}
```

```

"MS": "sheng",
"DLBM": 500000230000052,
"DMCCD": 3,
"DMJP": "beijingshi",
"DMQP": "",
"DMBM": null,
"BDY": null,
"YWMC": null,
"DMDZ": null,
"DMGC": 0,
"LV": 5
}]

```

返回参数说明

| 参数名 | 类型 | 说明 |
|--------|--------|-------|
| DMBS | String | 标识 |
| DMCJBM | String | 层级编码 |
| DMMC | String | 名称 |
| WZ | String | 位置 |
| MS | String | 描述 |
| DLBM | String | 地理编码 |
| DMCCD | String | DMCCD |
| DMJP | String | 简拼 |
| DMQP | String | 全拼 |
| DMBM | String | 编码 |
| BDY | String | DBY |
| YWMC | String | YWMC |
| DMDZ | String | 地址 |
| DMGC | String | DMGC |
| LV | int | 层级 |

（四）查询地理编码

简要描述：查询地理编码接口

请求 URL: /placeservice/api/v1/query/dlbm={code}

请求方式: GET

参数:

| 参数名 | 必选 | 类型 | 说明 |
|------|----|--------|----|
| code | 是 | string | 编码 |

请求示例

http://192.168.4.38:8319/placeservice/api/v1/query/dlbm=1500214740054811

返回示例

```
[
{
"DMBS": "291c0b1d-d6c1-4abe-b8bc-2593a69405ea",
"DMCJBM": 0,
"DMMC": "金川大酒店",
"WZ": "POINT(121.08586 27.963524)",
"MS": "餐饮",
"DLBM": 1500214740054811,
"DMCCD": 5,
"DMJP": "jcdjd",
"DMQP": "",
"DMBM": null,
"BDY": null,
"YWMC": null,
"DMDZ": null,
"DMGC": 0,
"LV": 15
}...]
```

返回参数说明

| 参数名 | 类型 | 说明 |
|--------|--------|------|
| DMBS | String | 标识 |
| DMCJBM | String | 层级编码 |
| DMMC | String | 名称 |
| WZ | String | 位置 |

| 参数名 | 类型 | 说明 |
|-------|--------|-------|
| MS | String | 描述 |
| DLBM | String | 地理编码 |
| DMCCD | String | DMCCD |
| DMJP | String | 简拼 |
| DMQP | String | 全拼 |
| DMBM | String | 编码 |
| BDY | String | DBY |
| YWMC | String | YWMC |
| DMDZ | String | 地址 |
| DMGC | String | DMGC |
| LV | int | 层级 |

（五）获取地名数目总量

简要描述：获取数目总量接口

请求 URL：/placeservice/api/v1/query/count

请求方式：GET

请求示例：

http://192.168.4.38:8319/placeservice/api/v1/query/count

参数：

无

返回示例

16778770

返回参数说明

| 参数名 | 类型 | 说明 |
|-----|----|------|
| 无 | 无 | 数目总量 |

（六）获取批量地名数据

简要描述：获取批量地名数据接口

请求 URL：/placeservice/api/v1/query/batch

请求方式：GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|--------|----|-----|------|
| limit | 否 | int | 条数 |
| offset | 否 | int | 起始偏移 |

返回示例

```
[ {
  "DMBS": "8f713a58-bef4-4659-9359-81b727e595c0",
  "DMCJBM": 0,
  "DMMC": "宿州市",
  "WZ": "POINT(116.95854 33.647715)",
  "MS": "地级市",
  "DLBM": 800001750000422,
  "DMMCCD": 3,
  "DMJP": "",
  "DMQP": "",
  "DMBM": null,
  "BDY": null,
  "YWMC": null,
  "DMDZ": null,
  "DMGC": 0,
  "LV": 8
}]
```

返回参数说明

| 参数名 | 类型 | 说明 |
|--------|--------|--------|
| DMBS | String | 标识 |
| DMCJBM | String | 层级编码 |
| DMMC | String | 名称 |
| WZ | String | 位置 |
| MS | String | 描述 |
| DLBM | String | 地理编码 |
| DMMCCD | String | DMMCCD |
| DMJP | String | 简拼 |
| DMQP | String | 全拼 |

| 参数名 | 类型 | 说明 |
|------|--------|------|
| DMBM | String | 编码 |
| BDY | String | DBY |
| YWMC | String | YWMC |
| DMDZ | String | 地址 |
| DMGC | String | DMGC |

（七）获取世界行政区划

简要描述：获取世界行政区划

请求 URL：/placeservice/api/v1/vectors/district/world

请求方式：GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|----------------|----|---------|--------------|
| dataName | 是 | String | 名称 |
| returnGeometry | 是 | boolean | 是否是 Geometry |

请求示例

```
{
  "dataName": "津巴布韦",
  "returnGeometry": true
}
```

返回示例

```
MULTIPOLYGON(((31.287890625 -22.40205078125, 31.197265625 -22.344921875, 31.0734375 -22.3078125, 30.91611328125 -22.29072265625, 30.71162109375 -22.2978515625, 30.46015625 -22.32900390625, 30.1904296875 ....
```

返回参数说明

| 参数名 | 类型 | 说明 |
|--------------|--------|-------|
| MULTIPOLYGON | String | 多边形区域 |

（八）获取省市县名称的树状结构

简要描述：获取省市县名称的树状结构

请求 URL：/placeservice/api/v1/district/district

请求方式：GET

请求示例

<http://192.168.4.38:8319/placeservice/api/v1/vectors/district/treelist>

返回示例

```
[{"code":110000,"children":[{"code":110100,"children":[{"code":110106,"name":"丰台区"}, {"code":110116,"name":"怀柔区"}, {"code":110114,"name":"昌平区"}, {"code":110117,"name":"平谷区"}, {"code":110105,"name":"朝阳区"}, {"code":110112,"name":"通州区"}, {"code":110108,"name":"海淀区"}, {"code":110101,"name":"东城区"}, {"code":110109,"name":"门头沟区"}], ... }
```

返回参数说明

| 参数名 | 类型 | 说明 |
|----------|---------|------|
| code | Integer | 地区编码 |
| children | String | 下级 |
| name | String | 名称 |

(九) 获取省市县名称以及范围

简要描述：获取省市县名称以及范围

请求 URL：/placeservice/api/v1/vectors/district

请求方式：GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|----------------|----|---------|-------------------------------------|
| code | 是 | String | 地区编码 |
| returnGeometry | 是 | boolean | 是否需要返回地区的多边形几何信息，true 代表是，false 代表否 |

请求示例

<http://192.168.4.38:8319/placeservice/api/v1/vectors/district?code=110000&returnGeometry=false>

返回示例

```
[{"code":110100,"name":"市辖区"}, {"code":110200,"name":"县"}]
```

返回参数说明

| 参数名 | 类型 | 说明 |
|-----|----|----|
|-----|----|----|

| 参数名 | 类型 | 说明 |
|------|--------|------|
| code | String | 地区编码 |
| name | String | 地区名称 |

（十）按全称查询地名

简要描述：按全称查询地名

请求 URL：/placeservice/api/v1/query/fullPlaceName={name}

请求方式：GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|---------------|----|--------|------|
| fullPlaceName | 是 | string | 全称地名 |

请求示例

http://192.168.4.38:8319/placeservice/api/v1/query/fullPlaceName=北京

返回示例

```
[{
  "DMBS": "c1309542-b29b-490c-a247-f275eef47aab",
  "DMCJBM": 4,
  "DMMC": "北京市",
  "WZ": "POINT(116.410323 40.18498)",
  "MS": "sheng",
  "DLBM": 5000002300000052,
  "DMMCCD": 3,
  "DMJP": "beijingshi",
  "DMQP": "",
  "DMBM": null,
  "BDY": null,
  "YWMC": null,
  "DMDZ": null,
  "DMGC": 0,
  "LV": 5
}, ... ]
```

返回参数说明

| 参数名 | 类型 | 说明 |
|-----|----|----|
|-----|----|----|

| 参数名 | 类型 | 说明 |
|--------|--------|-------|
| DMBS | String | 标识 |
| DMCJBM | String | 层级编码 |
| DMMC | String | 名称 |
| WZ | String | 位置 |
| MS | String | 描述 |
| DLBM | String | 地理编码 |
| DMCCD | String | DMCCD |
| DMJP | String | 简拼 |
| DMQP | String | 全拼 |
| DMBM | String | 编码 |
| BDY | String | DBY |
| YWMC | String | YWMC |
| DMDZ | String | 地址 |
| DMGC | String | DMGC |
| LV | int | 层级 |

五、streetview 街景服务

（一）获取当前位置最近的街景点位编码

简要描述：指定经纬度寻找最近的街景点位编码（panold）

请求 URL：streetview/api/v1/{layer}/findNear.json?lng={lng}&lat={lat}

请求方式：GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|-------|----|--------|--------|
| layer | 是 | string | 街景发布名称 |
| lng | 是 | string | 经度 |
| lat | 是 | string | 纬度 |

请求示例

http://192.168.4.38:8319/streetview/api/v1/test/findNear.json?lng=113.549996&lat=22.11904

返回示例

```
{
  "panoId": "Zy5AMY8yssJE91YEmjggyA",
  "lng": 113.549996,
  "lat": 22.11904,
  "status": "ok"
}
```

返回参数说明

| 参数名 | 类型 | 说明 |
|--------|--------|------------------|
| panoId | string | id |
| lng | string | 经度 |
| lat | string | 纬度 |
| status | string | 返回状态，正确返回时值为“ok” |

（二）由街景点位编码获取街景瓦片

简要描述：通过街景点位编码（panoid）和经纬度以及行列号 x，y 获取对应的街景瓦片

请求 URL：streetview/api/v1/{layer}/tile?panoid={panoid}&x={x}&y={y}&lon={lon}&lat={lat}

请求方式：GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|--------|----|--------|--------|
| layer | 是 | string | 街景发布名称 |
| panoid | 是 | string | id |
| x | 是 | string | 行号 |
| y | 是 | string | 列号 |
| lon | 是 | string | 经纬 |
| lat | 是 | string | 维度 |

请求示例

`http://192.168.4.38:8319/streetview/api/v1/test/tile? panoid=Zy5AMY8yssJE91YEmjgyyA&x=1&y=1&lon=113.549996&lat=22.11904`

返回参数说明

返回图片

(三) 获取街景当前位置图像的属性信息

简要描述：通过查询到的 **panoid** 和经纬度，查找离指定 **pano** 点位最近的 **pano** 点位的 **panoid** 以及经纬度信息

请求 URL： `streetview/api/v1/{layer}/meta.json?panoid={panoid}&lon={log}&lat={lat}`

请求方式： GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|-------|----|--------|--------|
| layer | 是 | string | 街景发布名称 |
| log | 是 | string | 经度 |
| lat | 是 | string | 纬度 |

请求示例

`http://192.168.4.38:8319/streetview/api/v1/test/meta.json?panoid=Zy5AMY8yssJE91YEmjgyyA&lng=113.549996&lat=22.11904`

返回示例

```
{
  "Data": {
    "image_width": "13312",
    "image_height": "6656",
    "tile_width": "512",
    "tile_height": "512",
    "image_date": "2016-02",
    "imagery_type": 1,
    "copyright": "© 2018 Google"
  },
  "Projection": {
    "projection_type": "spherical",
    "pano_yaw_deg": "343.72",
```

```

"tilt_yaw_deg": "-74.84",
"tilt_pitch_deg": "1.93"
},
"Location": {
"panoId": "KH-fJheBNF2XuU6ks0FHjA",
"zoomLevels": "5",
"lat": "35.624010",
"lng": "139.676049",
"original_lat": "35.624021",
"original_lng": "139.676063",
"elevation_wgs84_m": "74.139831",
"description": "",
"region": "目黒区, 东京都",
"country": "日本",
"best_view_direction_deg": "10.3357",
"elevation_egm96_m": "37.560196"
},
"Links": [{
"yawDeg": "70.31",
"panoId": "ImZuj2Sx00nViiZSeeX0pg",
"road_argb": "0x80fdf872",
"description": ""
}, {
"yawDeg": "264.33",
"panoId": "SdIqwNShTPGNC4o4byH3Dg",
"road_argb": "0x80fdf872",
"description": ""
}, {
"yawDeg": "163.74",
"panoId": "QSEWFtOWj3arEV4uRhfdA",
"road_argb": "0x80fdf872",
"description": ""
}, {

```

```

"yawDeg": "343.57",
"panoId": "he7pwTBWhS0yFQJXfa_1WA",
"road_argb": "0x80fdf872",
"description": ""
}],
"model": {
"depth_map": "...",
"pano_map": "...
}
}

```

（四）获取向前/后/左/右操作的下一跳街景位置

简要描述：通过页面传来的相应位置的 **panoid** 以及经纬度查找离指定 **pano** 点位最近的 **pano** 点位的 **panoid** 以及经纬度信息。

请求 URL：streetview/api/v1/{layer}/findNearNext.json?id={panoid}&lng={lng}&lat={lat}

请求方式：GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|-------|----|--------|--------|
| layer | 是 | string | 街景发布名称 |
| id | 是 | string | id |
| lng | 是 | string | 经度 |
| lat | 是 | string | 纬度 |

请求示例

http://192.168.4.38:8319/streetview/api/v1/test/ findNearNext.json?lng=113.549996&lat=22.11904

返回示例

```

{
"panoId": "Gz_ZcW1zVf-blDbsqRaj4g",
"lng": 113.549955,
"lat": 22.119136,
"status": "ok"
}

```

返回参数说明

| 参数名 | 类型 | 说明 |
|--------|--------|------------|
| panoId | string | id |
| lng | string | 经度 |
| lat | string | 维度 |
| status | string | 状态，成功为“ok” |

六、GeoServerPlus 服务

(一) 矢量数据服务

1. WMS 基础服务

简要描述：WMS 基础服务，又称为“WMS-GetMap 服务”，获取地图图片。该操作根据客户端发出的请求参数在服务端进行检索，服务器端返回一个地图图像，其地理空间参数和大小参数是已经明确定义的，返回的地图图像可以是 GIF JPEG PNG 或 SVG 格式。

请求 URL： /geoserverplus/icenter/{LAYER}/wms?VERSION={VERSION}&REQUEST={REQUEST}&FORMAT={FORMAT}&TRANSPARENT={TRANSPARENT}&LAYERS={LAYERS}&TILED={TILED}&WIDTH={WIDTH}&HEIGHT={HEIGHT}&CRS={CRS}&STYLES={STYLES}&BBOX={BBOX}

请求方式： GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|---------|----|---------|--|
| LAYER | 是 | string | 基础影像名称 |
| VERSION | 是 | string | 支持的 WMS 规范的版本号，1.3.0 |
| REQUEST | 是 | string | 请求类型，默认值为 GetMap |
| FORMAT | 是 | string | 切片存储类型，包括 jpeg png 等 |
| TILED | 否 | boolean | 是否为标准瓦片，值为 true 时，WIDTH 和 Height 为 256 像素的标准瓦片；值为 false 时，WIDTH 和 Height 为任意大小瓦片 |
| WIDTH | 是 | integer | 瓦片宽度，单位：像素 |
| HEIGHT | 是 | integer | 瓦片高度，单位：像素 |
| CRS | 是 | string | 切片策略，包括瓦片坐标系等信息 |

| 参数名 | 必选 | 类型 | 说明 |
|-------------|----|------------------|------|
| TRANSPARENT | 是 | boolean | 透明度 |
| STYLES | 否 | string | 样式 |
| BBOX | 是 | application/json | 空间范围 |

请求示例：

http://192.168.4.96:9091/geoserverplus/icenter/%E6%B0%B4%E7%B3%BB0322-PNG-4326/wms?VERSION=1.3.0&REQUEST=GetMap&FORMAT=image%2Fpng&TRANSPARENT=true&LAYERS=icenter%3A%E6%B0%B4%E7%B3%BB0322-PNG-4326&TILED=true&WIDTH=256&HEIGHT=256&CRS=EPSG%3A4326&STYLES=&BBOX=26.0595703125%2C119.28955078125%2C26.08154296875%2C119.3115234375

返回示例

合法请求，服务器将根据请求参数（空间坐标参考系 图层 范围 图片大小等）返回一幅地图。

非法请求，服务器将根据请求的异常格式返回一个异常。

2. WMS GetCapabilities 接口

简要描述：矢量数据 WMS 基础服务 GetCapabilities 接口，主要用于获取矢量数据 WMS 服务的元信息

请求 URL：/geoserverplus/gwc/service/wms?layer={layer}&request={request}

请求方式：GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|---------|----|--------|---------------------------|
| layer | 否 | string | 图层名 |
| request | 是 | string | 请求类型，默认值为 GetCapabilities |

请求示例：

http://192.168.46.45:8310/geoserverplus/gwc/service/wms?request=GetCapabilities&layer=icenter:WorldCities10241447-GEOJSON-4326

返回示例

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE WMT_MS_Capabilities SYSTEM "http://schemas.opengis.net/wms/1.1.1/capabilities_1_1_1.dtd" >
<WMT_MS_Capabilities version="1.1.1">
```

```
<Service>
<Name>OGC:WMS</Name>
<Title>Web Map Service - GeoWebCache</Title>
<OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple" xlink:href="http://192.168.4.98:8319/geoserverplus/gwc/service/wms?SERVICE=WMS&" />
</Service>
<Capability>
<Request>
<GetCapabilities>
<Format>application/vnd.ogc.wms_xml</Format>
<DCPType>
<HTTP>
<Get>
<OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple" xlink:href="http://192.168.4.98:8319/geoserverplus/gwc/service/wms?SERVICE=WMS&" />
</Get>
</HTTP>
</DCPType>
</GetCapabilities>
<GetMap>
<Format>image/png</Format>
<Format>application/json;type=utfgrid</Format>
<Format>image/jpeg</Format>
<Format>image/terrain</Format>
<Format>application/json;type=geojson</Format>
<Format>application/json;type=topojson</Format>
<Format>application/x-protobuf;type=mapbox-vector</Format>
<DCPType>
<HTTP>
<Get>
<OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
```



```

e" xlink:href="http://192.168.4.98:8319/geoserverplus/gwc/service/wms?SERVIC
E=WMS&";"/>
</Get>
</HTTP>
</DCPType>
</GetMap>
<GetFeatureInfo>
<Format>text/html</Format>
<Format>application/json</Format>
<Format>application/vnd.ogc.gml/3.1.1</Format>
<Format>text/xml</Format>
<Format>application/vnd.ogc.gml</Format>
<Format>text/plain</Format>
<DCPType>
<HTTP>
<Get>
<OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simpl
e" xlink:href="http://192.168.4.98:8319/geoserverplus/gwc/service/wms?SERVIC
E=WMS&";"/>
</Get>
</HTTP>
</DCPType>
</GetFeatureInfo>
<DescribeLayer>
<Format>application/vnd.ogc.wms_xml</Format>
<DCPType>
<HTTP>
<Get>
<OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simpl
e" xlink:href="http://192.168.4.98:8319/geoserverplus/gwc/service/wms?SERVIC
E=WMS&";"/>
</Get>
</HTTP>

```

```

</DCPType>
</DescribeLayer>
<GetLegendGraphic>
<Format>image/png</Format>
<Format>image/jpeg</Format>
<Format>image/gif</Format>
<DCPType>
<HTTP>
<Get>
<OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
xlink:href="http://192.168.4.98:8319/geoserverplus/gwc/service/wms?SERVICE=WMS&" />
</Get>
</HTTP>
</DCPType>
</GetLegendGraphic>
</Request>
<Exception>
<Format>application/vnd.ogc.se_xml</Format>
</Exception>
<Layer>
<Title>GeoWebCache WMS</Title>
<Abstract>Note that not all GeoWebCache instances provide a full WMS service.
</Abstract>
<LatLonBoundingBox minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"/>
<Layer>
<Name>habse22222222-JPEG-4326</Name>
<Title>habse22222222-JPEG-4326</Title>
<SRS>EPSG:4326</SRS>
<LatLonBoundingBox minx="71.982421875" miny="9.931625697923742" maxx="144.0527089989704"
maxy="58.0078125"/>
<BoundingBox SRS="EPSG:4326" minx="71.982421875" miny="9.931625697923742" maxx="144.0527089989704"
maxy="58.0078125"/>

```

```

</Layer>
<Layer>
<Name>澳门 15 米时序影像-PNG-4326</Name>
<Title>澳门 15 米时序影像-PNG-4326</Title>
<SRS>EPSG:3857</SRS>
<SRS>EPSG:4326</SRS>
<LatLonBoundingBox minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"/>
<BoundingBox SRS="EPSG:3857" minx="-2.003750834E7" miny="-2.003750834E7" maxx="2.003750834E7" maxy="2.003750834E7"/>
<BoundingBox SRS="EPSG:4326" minx="-180.0" miny="-90.0" maxx="180.0" maxy="90.0"/>
</Layer>
</Layer>
</Capability>
</WMT_MS_Capabilities>

```

返回参数说明

返回 xml 文件，主要包含 **Service** 和 **Capability** 两个组成部分。

- **Service**

WMS Capabilities XML 中的元素中的内容表示通用服务元数据（General service metadata），该元素提供了对服务器通用元数据的一个整体描述。该数据包括 **Name** **Title** 以及 **Online Resource URL** 元素，另外包括一些可选的内容：**Abstract** **KeywordList** **ContactInformation** **Fees** 和 **AccessConstraints** 元素等。

| 元素名称 | 描述 |
|--------------------|---|
| Name | WMS 服务的名称 |
| Title | WMS 服务标题 |
| Abstract | WMS 服务摘要，提供有关服务器信息的更多描述 |
| KeywordList | 关键字，服务器作为一个整体，能够有助于目录搜索 |
| OnlineResource | 在线资源，提供服务提供者的网站参考，其他 OnlineResource 元素是每个操作的 URL 前缀 |
| ContactInformation | 联系信息，包括了服务提供者名称 地址 电话 传真 邮箱等信息 |
| Fees | 费用 |
| AccessConstraints | 访问限制，如果服务器没有应用访问限制，该元素可以忽略，如 |

| 元素名称 | 描述 |
|------|---|
| | 果使用了该元素，“none”（不区分大小写）表示没有访问限制，比如“none”，如果使用该元素，该元素的内容没有准确的语法定义，客户端可显示用户信息和行为内容 |

- **Capability**

元素中的内容表达了服务器实例所支持的请求（GetCapabilities GetMap 或 GetFeatureInfo），每一种操作提供的输出格式，以及每一个操作的 URL 前缀。XML DTD 包括了分布式平台（Distributed Computing Platforms, DCP）的占位符，以及其它的 HTTP 请求方法。但目前只定义了 HTTP GET 作为基本的 WMS 操作。可忽略的服务提供者特定元素可以包含在 元素中。WMS 中还会包含 元素以及 HTTP POST 请求的 URL。Layer 和 Style 是 WMS Capabilities XML 文档中最重要的部分。每个可用的地图都通过元素列举在 Capabilities XML 文档中。一个父图层可以包含任意数目的子图层。地图服务器必须在每个提供的地图中至少包含一个 元素。元素的嵌套子元素，表达了图层的属性信息。

3. WMTS 基础服务

简要描述：WMTS 基础服务，GetTile 服务接口，获取地图瓦片，该操作根据客户端发出的请求参数在服务端进行检索，服务器端返回地图瓦片图像。

请求 URL： /geoserverplus/gwc/service/wmts?Request={Request}&layer={layer}&Format={Format}&TileCol={TileCol}&TileRow={TileRow}&tilematrixset={tilematrixset}&TileMatrix={TileMatrix}

请求方式： GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|---------------|----|--------|-------------------------|
| layer | 是 | string | 矢量影像名称 |
| Request | 是 | string | 请求操作，默认值为“GetTile” |
| Format | 是 | string | 瓦片地图的输出格式，包括 png jpeg 等 |
| tilematrixset | 是 | string | 瓦片矩阵数据集，其值在服务的元数据文档中指定 |
| tilematrix | 是 | string | 瓦片矩阵，最后一位是层级 |
| TileCol | 是 | string | 值为大于 0 的整数，表示瓦片矩阵的列号 |
| TileRow | 是 | string | 值为大于 0 的整数，表示瓦片矩阵的行号 |

请求示例 1：

返回一张图片

<http://192.168.4.15:8310/geoserverplus/gwc/service/wmts?layer=icenter%3Agoogle2-PNG-3857&tilematrixset=EPSG%3A3857&Request=GetTile&Format=image%2Fpng&TileMatrix=EPSG:3857:2&TileCol=3&TileRow=2>

返回值:

返回一张 png 或 jpg 格式图片

请求示例 2:

返回 geojson 矢量瓦片

<http://192.168.4.15:8310/geoserverplus/gwc/service/wmts?Request=GetTile&layer=icenter:city10211336-GEOJSON-4326&Format=application%2Fjson%3Btype%3Dgeojson&TileCol=12&TileRow=2&tilematrixset=EPSG:4326&TileMatrix=EPSG:4326:3>

返回值: 参考 geojson 数据格式

```
{
  "type": "FeatureCollection",
  "totalFeatures": "unknown",
  "features": [
    {
      "type": "Feature",
      "id": "乡镇村道-json-GEOJSON-4326.0",
      "geometry": {
        "type": "MultiLineString",
        "coordinates": [
          [
            [
              119.267546,
              26.040293
            ],
            [
              119.266365,
              26.041504
            ]
          ], ...
        ]
      }
    }
  ]
}
```

```

},
"geometry_name": "geom",
"properties": {
"name": "cd1"
}
},
{
...
}
]
}

```

4. WMTS GetCapabilities 接口

简要描述：WMTS 规范的 GetCapabilities 接口，主要用于获取矢量数据 WMTS 服务的元信息。

请求 URL： /geoserverplus/gwc/service/wmts?layer={layer}&request=GetCapabilities

请求方式： GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|---------|----|--------|-------------------------|
| layer | 否 | string | 图层名 |
| request | 是 | string | 请求类型，值为 GetCapabilities |

请求示例：

http://192.168.46.45:8310/geoserverplus/gwc/service/wmts?request=GetCapabilities&layer=icenter:ChinaProvinces-Vector-PNG-4326

返回示例

```

<Capabilities xmlns="http://www.opengis.net/wmts/1.0" xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:gml="http://www.opengis.net/gml" xsi:schemaLocation="http://www.opengis.net/wmts/1.0 http://schema

```

```
s.opengis.net/wmts/1.0/wmtsGetCapabilities_response.xsd" version="1.0.0">
<ows:ServiceIdentification>
<ows:Title>GeoServer Web Map Tile Service</ows:Title>
<ows:Abstract>A compliant implementation of WMTS service.</ows:Abstract>
<ows:Keywords/>
<ows:ServiceType>OGC WMTS</ows:ServiceType>
<ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
<ows:Fees>NONE</ows:Fees>
<ows:AccessConstraints>NONE</ows:AccessConstraints>
</ows:ServiceIdentification>
<ows:ServiceProvider>
<ows:ProviderName>http://geoserver.org</ows:ProviderName>
<ows:ServiceContact>
<ows:ContactInfo>
<ows:Address/>
</ows:ContactInfo>
</ows:ServiceContact>
</ows:ServiceProvider>
<ows:OperationsMetadata>
<ows:Operation name="GetCapabilities">
<ows:DCP>
<ows:HTTP>
<ows:Get xlink:href="http://192.168.4.98:8319/geoserverplus/gwc/service/wmts?">
<ows:Constraint name="GetEncoding">
<ows:AllowedValues>
<ows:Value>KVP</ows:Value>
</ows:AllowedValues>
</ows:Constraint>
</ows:Get>
</ows:HTTP>
</ows:DCP>
</ows:Operation>
```

```
<ows:Operation name="GetTile">
  <ows:DCP>
    <ows:HTTP>
      <ows:Get xlink:href="http://192.168.4.98:8319/geoserverplus/gwc/service/wmts?">
        <ows:Constraint name="GetEncoding">
          <ows:AllowedValues>
            <ows:Value>KVP</ows:Value>
          </ows:AllowedValues>
        </ows:Constraint>
      </ows:Get>
    </ows:HTTP>
  </ows:DCP>
</ows:Operation>

<ows:Operation name="GetFeatureInfo">
  <ows:DCP>
    <ows:HTTP>
      <ows:Get xlink:href="http://192.168.4.98:8319/geoserverplus/gwc/service/wmts?">
        <ows:Constraint name="GetEncoding">
          <ows:AllowedValues>
            <ows:Value>KVP</ows:Value>
          </ows:AllowedValues>
        </ows:Constraint>
      </ows:Get>
    </ows:HTTP>
  </ows:DCP>
</ows:Operation>
</ows:OperationsMetadata>
<Contents>
  <TileMatrixSet>
    <TileMatrix>
      <ows:Identifier>GlobalCRS84Pixel:2</ows:Identifier>
```



```

<ScaleDenominator>1.9878480498798856E8</ScaleDenominator>
<TopLeftCorner>90.0 -180.0</TopLeftCorner>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>3</MatrixWidth>
<MatrixHeight>2</MatrixHeight>
</TileMatrix>
</TileMatrixSet>
</Contents>
<ServiceMetadataURL xlink:href="http://192.168.4.98:8319/geoserverplus/gwc/
service/wmts?SERVICE=wmts&REQUEST=getcapabilities&VERSION=1.0.0"/>
<ServiceMetadataURL xlink:href="http://192.168.4.98:8319/geoserverplus/gwc/
rest/wmts/WMTSCapabilities.xml"/>
</Capabilities>

```

返回参数说明

在调用 WMTS 服务的时候，通过 GetCapabilities 操作来请求 ServiceMetadata 资源，WMTS 将返回一个包含服务元数据文档的响应文件。WMTS Capabilities XML 主要包含 ServiceIdentification、ServiceProvider、OperationsMetadata、Contents 和 Themes 几个组成部分。另外，每个服务元数据文档还必须包含唯一的版本号，以及获取该服务元数据的 URI 参数 ServiceMetadataURL，另外也可以包含服务元数据文档版本参数 UpdateSequence。

- ServiceIdentification

ServiceIdentification 是关于具体服务的元数据，是对服务的一个整体描述。它包含了 ServiceType、ServiceTypeVersion、Version、Title 等元素，还包括一些可选元素：Abstract、Keywords、Fees 和 AccessConstraints。

| 元素名称 | 是否必需 | 描述 |
|--------------------|------|------------------------------|
| ServiceType | 是 | 服务类型，由于是 WMTS 服务，其值为 WMTS |
| ServiceTypeVersion | 是 | 服务版本。WMTS 目前只有一个版本，其值为 1.0.0 |
| Title | 否 | 服务标题，可以是一个或多个 |
| Abstract | 否 | 服务摘要，提供有关服务器信息的简短描述 |
| Keywords | 否 | 关键词，服务器中常用的词或短语 |
| Fees | 否 | 费用。如果服务器没有使用费用，该元素可以忽略 |
| AccessConstraints | 否 | 访问限制。如果服务器没有应用访问限制，则可以 |

| 元素名称 | 是否必需 | 描述 |
|------|------|-----|
| | | 忽略。 |

- **ServiceProvider**

ServiceProvider（可选）是关于服务提供商的信息，它包含了 **ProviderName** **ProviderSite** 和 **ServiceContact** 元素。

| 元素名称 | 是否必需 | 描述 |
|----------------|------|---------------------|
| ProviderName | 是 | 提供商名称。是对服务提供商的唯一标识。 |
| ProviderSite | 否 | 提供商地址。 |
| ServiceContact | 是 | 服务提供商的联系信息。 |

- **OperationsMetadata**

OperationsMetadata 描述了当前服务中支持的所有操作以及操作请求的 URI。**OperationsMetadata** 元素必须包含一个或多个 **Operation** 子节点，每个 **Operation** 子节点对应一个操作。**OperationsMetadata** 的元素说明见下表

| 元素名称 | 是否必需 | 描述 |
|----------------------|------|---|
| Operation | 是 | 操作。用来标识该服务支持的操作，每个 Operation 节点对应一个操作，以 name 属性进行标识，如 |
| Parameter | 否 | 参数。用于指定操作中一个或一些参数的有效域，以 name 属性标识参数名， Value 标签中给出参数的有效值，这些参数可能是操作中的输入输出参数 |
| Constraint | 否 | 限制。对应用于此服务器的非参属性的有效域的限制，以 name 属性标识非参属性名， Value 标签中给出限定值 |
| ExtendedCapabilities | 否 | 扩展能力。关于服务器和软件附加能力的描述 |

- **Contents**

Contents 是对服务中可用图层的整体描述，它包括 **Layer**、**OtherSource** 和 **TileMatrixSet** 三个子元素，下面重点介绍 **TileMatrixSet** 元素。

| 元素名称 | 是否必需 | 描述 |
|-------------|------|------------------------|
| Identifier | 是 | 标识符。即名称。 |
| Title | 否 | 标题。 |
| Abstract | 否 | 摘要。对瓦片矩阵集的简短描述。 |
| Keywords | 否 | 关键词。数据集中的常用词或短语的无序列表。 |
| BoundingBox | 否 | 边界框。在支持的坐标系下包围瓦片矩阵集的最小 |

| 元素名称 | 是否必需 | 描述 |
|-------------------|------|-----------------------------|
| | | 矩形。 |
| SupportedCRS | 是 | 支持的坐标系 |
| WellKnownScaleSet | 否 | 通用比例尺集 |
| TileMatrix | 是 | 瓦片矩阵。每个瓦片矩阵集中可以包含一个或多个瓦片矩阵。 |

5. WFS 基础服务接口

简要描述：DescribeFeatureType 服务接口，用于生成一个 Schema 描述，该 Schema 描述了 WFS 服务提供的要素类型（Feature Type），以及要素类型的结构信息。该 Schema 还定义了 WFS 服务所期望的要素实例在输入时如何编码以及输出时如何生成要素实例。

请求 URL：/geoserverplus/icenter/ows?version={version}&SERVICE=WFS&REQUEST=DescribeFeatureType&TYPENAME={TYPENAME}

请求方式：GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|----------|----|--------|--|
| SERVICE | 是 | string | 服务类型 |
| REQUEST | 是 | string | 请求的操作名称，值为“DescribeFeatureType” |
| version | 否 | string | 版本号 |
| TYPENAME | 否 | string | 字符类型，值为要素类型的列表，多个值之间用“，”隔开，默认解析包括的全部要素类型 |

请求示例

http://192.168.4.96:9091/geoserverplus/icenter/ows?version=1.0.0&SERVICE=WFS&REQUEST=DescribeFeatureType&TYPENAME=icenter:shengjiewww-PNG-4326

返回示例

```
<xsd:schema xmlns:gml="http://www.opengis.net/gml" xmlns:icenter="http://icenter" xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" targetNamespace="http://icenter">
<xsd:import namespace="http://www.opengis.net/gml" schemaLocation="http://192.168.4.96:8240/schemas/gml/2.1.2/feature.xsd"/>
<xsd:complexType name="shengjiewww-PNG-4326Type">
<xsd:complexContent>
```

```

<xsd:extension base="gml:AbstractFeatureType">
<xsd:sequence>
<xsd:element maxOccurs="1" minOccurs="0" name="type" nillable="true" type="xsd:string"/>
<xsd:element maxOccurs="1" minOccurs="0" name="subtype" nillable="true" type="xsd:string"/>
<xsd:element maxOccurs="1" minOccurs="0" name="name" nillable="true" type="xsd:string"/>
<xsd:element maxOccurs="1" minOccurs="0" name="geom" nillable="true" type="gml:GeometryPropertyType"/>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:element name="shengjiewww-PNG-4326" substitutionGroup="gml:_Feature" type="icenter:shengjiewww-PNG-4326Type"/>
</xsd:schema>

```

返回参数说明

WFS 2.0.0 服务在响应 DescribeFeatureType 操作时，支持两种输出格式（outputFormat）。这两种格式在 Capabilities XML 文档中有明确声明，分别为“application/gml+xml;version=3.2”和“text/xml;subtype=gml/3.2”。

WFS 2.0.0 支持多个名字空间（Namespace），而一个 XML Schema 文档只能描述单一名字空间的元素，所以一个 WFS 2.0.0 服务不能通过一个 XML Schema 文档描述多个名字空间。为解决这一问题，WFS 2.0.0 可以生成一个 XML Schema 文档，这个 XML Schema 文档是一个“包” Schema，它能够引入（import）不同名字空间（XML 名字空间的属性为 xmlns）。

6. WFS GetCapabilities 接口

简要描述：矢量数据的 WFS 基础服务 GetCapabilities 接口，获取矢量数据 WFS 服务的元信息。

请求 URL： /geoserverplus/icenter/ows?service=WFS&request=GetCapabilities

请求方式： GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|-----|----|----|----|
|-----|----|----|----|

| 参数名 | 必选 | 类型 | 说明 |
|---------|----|--------|--------------------------|
| service | 是 | string | 服务类型，服务类型标识值为“WFS” |
| request | 是 | string | 请求的操作值为“GetCapabilities” |
| version | 否 | string | 值为请求的 WFS 的版本号 |

请求示例：

http://192.168.4.18:8310/geoserverplus/icenter/ows?service=WFS&request=GetCapabilities&version=1.0.0

返回示例

```
<WFS_Capabilities xmlns="http://www.opengis.net/wfs" xmlns:icenter="http://icenter" xmlns:ogc="http://www.opengis.net/ogc" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0.0" xsi:schemaLocation="http://www.opengis.net/wfs http://a70e5439d9fc:8240/schemas/wfs/1.0.0/WFS-capabilities.xsd">
  <Service>
    <Name>My GeoServer WFS</Name>
    <Title>My GeoServer WFS</Title>
    <Abstract>
      This is a description of your Web Feature Server. The GeoServer is a full transactional Web Feature Server, you may wish to limit GeoServer to a Basic service level to prevent modificaiton of your geographic data.
    </Abstract>
    <Keywords>WFS, WMS, GEOSERVER</Keywords>
    <OnlineResource>http://a70e5439d9fc:8240/icenter/wfs</OnlineResource>
    <Fees>NONE</Fees>
    <AccessConstraints>NONE</AccessConstraints>
  </Service>
  ...
  </ogc:Arithmetic_Operators>
  </ogc:Scalar_Capabilities>
  </ogc:Filter_Capabilities>
</WFS_Capabilities>
```

返回参数说明

参考 GML 规范

7. TMS-按名称查询 TMS 服务元信息

简要描述：获取指定名称的 TMS 服务（包括矢量数据 影像数据）元信息

请求 URL： /service/tms/{version}/{layerName}@{gridSet}@{mimetype}

请求方式： GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|-----------|----|--------|--|
| version | 是 | string | 版本号 |
| layername | 是 | string | 矢量名称 |
| gridSet | 是 | string | 切片策略 |
| mimetype | 是 | string | 切片存储类型，包括 png jpeg, geojson, topojson pbf (pbf 是经过压缩的瓦片) |

请求示例：

http://192.168.4.98:8319/geoserverplus/gwc/service/tms/1.0.0/icenter:%E6%B0%B4%E7%B3%BB13-PNG-4326@EPSG:4326@png

返回示例

```
<TileMap version="1.0.0" tilemapservice="http://192.168.4.98:8319/geoserverplus/gwc/service/tms/1.0.0">
<Title>shengtianffpng4326</Title>
<Abstract/>
<SRS>EPSG:4326</SRS>
<BoundingBox minx="0.0" miny="-90.0" maxx="180.0" maxy="90.0"/>
<Origin x="0.0" y="-90.0"/>
<TileFormat width="256" height="256" mime-type="image/png" extension="png" />
<TileSets profile="local">
<TileSet href="http://192.168.4.98:8319/geoserverplus/gwc/service/tms/1.0.0/icenter%3Ashengtianff-PNG-4326@EPSG%3A4326@png/0" units-per-pixel="0.703125" order="0"/>
<TileSet href="http://192.168.4.98:8319/geoserverplus/gwc/service/tms/1.0.0/icenter%3Ashengtianff-PNG-4326@EPSG%3A4326@png/1" units-per-pixel="0.3515625" order="1"/>
<TileSet href="http://192.168.4.98:8319/geoserverplus/gwc/service/tms/1.0.0/icenter%3Ashengtianff-PNG-4326@EPSG%3A4326@png/2" units-per-pixel="0.17578125" order="2"/>
</TileSets>
</TileMap>
```

```
<TileSet href="http://192.168.4.98:8319/geoserverplus/gwc/service/tms/1.0.0/icenter%3Ashengtianff-PNG-4326@EPSG%3A4326@png/3" units-per-pixel="0.087890625" order="3"/>
</TileSets>
</TileMap>
```

返回参数说明

| 参数名 | 类型 | 说明 |
|-------------|--------|---|
| TileMap | string | version, 支持 tms 规范的版本; tilemapservice tms 服务资源的 url 地址 |
| Title | string | TMS 服务的标题 |
| Abstract | string | TMS 服务的摘要 |
| SRS | string | 坐标系 |
| BoundingBox | string | 包括以下属性: minx x 轴最小值, miny y 轴最小值, maxx x 轴最大值, maxy y 轴最大值 |
| Origin | string | x 坐标原点 x 值, y 坐标原点 y 值 |
| TileFormat | string | 包括以下属性: width 瓦片宽度, height 瓦片高度, mime-type 瓦片类型, extension 瓦片扩展 |
| TileSets | string | profile 描述信息, 下一级节点由若干个 TileSet 组成, TileSet 数据结构见下表 |

TileSet 数据表描述

| 参数名 | 类型 | 说明 |
|-----------------|--------|------------|
| href | string | 资源的 url 位置 |
| units-per-pixel | string | 每个像素点的尺寸 |
| order | string | 资源显示时排列的顺序 |

8. TMS-获取所有 TMS 服务元信息

简要描述: -获取服务中心所有 TMS 服务元信息 (包括矢量 影像)

请求 URL: /geoserverplus/gwc/service/tms/1.0.0

请求方式: GET

参数:

无

请求示例:

<http://192.168.4.98:8319/geoserverplus/gwc/service/tms/1.0.0>

返回示例

```
<TileMapService version="1.0.0" services="http://icenter.geovis.online/geoserverplus/gwc/">
<Title>Tile Map Service</Title>
<Abstract>A Tile Map Service served by GeoWebCache</Abstract>
<TileMaps>
<TileMap title="country2png4326" srs="EPSG:3857" profile="local" href="http://icenter.geovis.online/geoserverplus/gwc/service/tms/1.0.0/icenter%3Acountry2-PNG-4326@EPSG%3A3857@utfgrid"/>
<TileMap title="country2png4326" srs="EPSG:3857" profile="local" href="http://icenter.geovis.online/geoserverplus/gwc/service/tms/1.0.0/icenter%3Acountry2-PNG-4326@EPSG%3A3857@png"/>
<TileMap title="country2png4326" srs="EPSG:3857" profile="local" href="http://icenter.geovis.online/geoserverplus/gwc/service/tms/1.0.0/icenter%3Acountry2-PNG-4326@EPSG%3A3857@jpeg"/>
<TileMap title="country2png4326" srs="EPSG:3857" profile="local" href="http://icenter.geovis.online/geoserverplus/gwc/service/tms/1.0.0/icenter%3Acountry2-PNG-4326@EPSG%3A3857@geojson"/>
<TileMap title="Macao_1984.tif" srs="EPSG:4326" profile="local" href="http://icenter.geovis.online/geoserverplus/gwc/service/tms/1.0.0/icenter%3AMacao_198411-PNG-4326@EPSG%3A4326@png"/>
<TileMap title="东方明珠.tif" srs="EPSG:4326" profile="local" href="http://icenter.geovis.online/geoserverplus/gwc/service/tms/1.0.0/icenter%3A%E4%B8%9C%E6%96%B9%E6%98%8E%E7%8F%A0-PNG-4326@EPSG%3A4326@png"/>
</TileMaps>
</TileMapService>
```

返回参数说明

| 参数名 | 类型 | 说明 |
|----------------|--------|---|
| TileMapService | string | TMS 服务英文全称, TileMapService, 描述服务版本 version, 服务域名 services |
| Title | string | TMS 服务的标题 |
| Abstract | string | TMS 服务的摘要 |

| 参数名 | 类型 | 说明 |
|----------|--------|---|
| TileMaps | string | TMS 服务的根节点名称，里面包含多个图层（TileMap），每个图层的描述见下表 |

TileMap 图层信息描述

| 参数名 | 类型 | 说明 |
|---------|--------|----------------|
| title | string | 图层主题 |
| srs | string | TMS 服务的标题 |
| profile | string | local，表示本地 |
| href | string | 图层代表的资源 url 位置 |

9. TMS- 获取指定图层、层、行、列的资源

简要描述：TMS- 获取指定图层（包括矢量数据 影像数据）、层、行、列的固定大小（256*256 像素）的图像瓦片

请求 URL： /geoserverplus/gwc/service/tms/1.0.0/{layer}@{type}/{z}/{x}/{y}

请求方式： GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|-------|----|--------|--|
| LAYER | 是 | string | 图层名称 |
| type | 是 | string | 瓦片资源格式， jpeg |
| z | 是 | string | 层， tms 包括 0-23 层，第 0 层包括两个瓦片，每增加一层瓦片数量增加四倍 |
| x | 是 | string | 行号 |
| y | 是 | string | 列号 |

请求示例： <http://icenter.geovis.online/geoserverplus/gwc/service/tms/1.0.0/icenter%3A省道-PNG-4326@EPSG%3A4326@jpeg/21/0/0>

返回示例

合法请求，服务器将根据请求参数（空间坐标参考系 图层名 层 行 列信息）返回一张 256*256 大小的图像瓦片。

非法请求，服务器将根据请求的异常格式返回一个异常。

（二）影像数据服务

1. WMS 基础服务

简要描述：WMS 基础服务，又称为“WMS-GetMap 服务”，主要用来获取影像数据的图片信息，该操作根据客户端发出的请求参数在服务端进行检索，服务器端返回一个影像图片，其地理空间参数和大小参数是已经明确定义的，返回图片可以是 GIF JPEG PNG 或 SVG 格式。

请求 URL： /geoserverplus/icenter/{LAYER}/wms?REQUEST=GetMap&BBOX=90,0,180,90&WIDTH=256&HEIGHT=256&LAYERS=Image_Global_Base_JPEG_G1C5-JPEG-4326&STYLES=&FORMAT=image/jpeg&TRANSPARENT=TRUE&tiled=true&SRS=EPSG%3A432

请求方式： GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|----------------|----|------------------|-----------------------------|
| REQUEST | 是 | string | 请求类型，值为“GetMap” |
| SRS | 是 | string | 坐标参照系统 |
| layers | 是 | string | 值为一个或多个地图图层列表，多个图层之间用“,” 隔开 |
| FORMAT | 是 | string | 切片存储类型，格式如下：image/jpeg |
| TILED | 否 | boolean | 是否切片 |
| WIDTH | 是 | integer | 竖向像素值 |
| HEIGHT | 是 | integer | 横向像素值 |
| MAP_RESOLUTION | 否 | integer | 地图分辨率 |
| FORMAT_OPTIONS | 否 | integer | 输出打印格式单位 |
| TRANSPARENT | 是 | boolean | 是否透明 |
| STYLES | 否 | string | 矢量样式 |
| BBOX | 是 | application/json | 范围 |

请求示例

http://192.168.4.96:9091/geoserverplus/icenter/Image_Global_Base_JPEG_G1C5-JPEG-4326/wms?SERVICE=WMS&REQUEST=GetMap&BBOX=90,0,180,90&WIDTH=256&HEIGHT=256&LAYERS=Image_Global_Base_JPEG_G1C5-JPEG-4326&STYLES=&FORMAT=image/jpeg&TRANSPARENT=TRUE&tiled=true&SRS=EPSG%3A432

返回示例

返回一个 png 或 jpg 图片

2. WMS GetCapabilities 接口

简要描述：影像数据的 WMS 服务的 GetCapabilities 接口，主要用于获取影像数据 WMS 服务的元信息

请求 URL： /geoserverplus/gwc/service/wms?layer={layer}&request={request}

请求方式： GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|---------|----|--------|---------------------------|
| layer | 否 | string | 图层名称 |
| request | 是 | string | 请求类型，默认值为 GetCapabilities |

请求示例：

http://192.168.4.98:8319/geoserverplus/gwc/service/wms?layer=icenter:test0722-PNG-4326&request=GetCapabilities

返回示例

与矢量数据服务-WMS GetCapabilities 接口返回示例和返回参数说明相同。

3. TMS 基础服务

简要描述： TMS 基础服务

请求 URL： /geoserverplus/gwc/service/tms/1.0.0/{layerName}@{gridSet}@{mimetype}

请求方式： GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|-----------|----|--------|------------------------------|
| layerName | 是 | string | 图层名称 |
| gridSet | 是 | string | 切片策略，EPSG:4326 代表坐标系统为 WGS84 |
| mimetype | 是 | string | 切片存储类型，包括 png jpeg 等 |

请求示例：

http://192.168.4.98:8319/geoserverplus/gwc/service/tms/1.0.0/icenter:test0722-PNG-4326@EPSG:4326@png

返回示例

返回参数说明

4. WMTS 基础服务

简要描述：影像数据的 WMTS 基础服务，即 GetTile 服务接口，主要用于获取指定范围、指定特征的影像数据的瓦片图像

请求 URL： /geoserverplus/gwc/service/wmts?layer={layer}&tilematrixset={tilematrixset}&Request={Request}&Format={Format}&TileMatrix={TileMatrix}&TileCol={TileCol}&TileRow={TileRow}

请求方式： GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|---------------|----|---------|-------------------|
| layer | 是 | string | 影像名称 |
| tilematrixset | 是 | string | 切片策略 |
| Request | 是 | string | 请求类型，默认值为 GetTile |
| Format | 是 | string | 切片存储类型 |
| TileMatrix | 是 | string | 切片结构 |
| TileCol | 是 | integer | 切片的列号 |
| TileRow | 是 | integer | 切片的行号 |

请求 URL：

http://192.168.4.98:8319/geoserverplus/gwc/service/wmts?layer=icenter%3Atest0722-PNG-4326&tilematrixset=EPSG%3A4326&Request=GetTile&Format=image%2Fpng&TileMatrix=EPSG%3A4326%3A2&TileCol=4&TileRow=0

返回值： 返回一张 png 或 jpg 格式图片

5. WMTS GetCapabilities 接口

简要描述：影像数据的 WMTS 服务的 GetCapabilities 接口，主要用于

请求 URL： /geoserverplus/gwc/service/wmts?layer={layer}&request={request}

请求方式： GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|---------|----|--------|---------------------------|
| layer | 否 | string | 影像名称 |
| request | 是 | string | 请求类型，默认值为 GetCapabilities |

请求示例：

http://192.168.4.98:8319/geoserverplus/gwc/service/wmts?layer=icenter:test0722-PNG-4326&request=GetCapabilities

返回示例

与矢量数据服务-WMTS GetCapabilities 接口返回示例和返回参数说明相同。

七、Geolot 实时数据服务

（一）获取聚合数据

简要描述：获取聚合数据

请求 URL： geoiot/api/v1/data/cluster?sourceId=1&minLon=114.74079895019531&maxLon=116.40934753417969&minLat=39.57028198242188&maxLat=40.59475708007813&level=8

请求方式：GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|----------|----|--------|---------------------|
| sourceId | 是 | string | 数据源 Id |
| level | 是 | int | 聚合层级 |
| duration | 否 | long | 数据期限（单位毫秒，若为空则不限时间） |
| minLon | 是 | double | 区域范围——最小经度 |
| maxLon | 是 | double | 区域范围——最大经度 |
| minLat | 是 | double | 区域范围——最小纬度 |
| maxLat | 是 | double | 区域范围——最大纬度 |

请求示例

http://192.168.4.38:8319/geoiot/api/v1/data/cluster? sourceId=1&minLon=114.74079895019531&maxLon=116.40934753417969&minLat=39.57028198242188&maxLat=40.59475708007813&level=11

返回示例

```
{
  "grids": [{
    "geoHash": -1781173652625031168,
    "count": 385,
    "boundingBox": {
      "maxLon": 116.015625,
      "minLon": 115.3125,
      "minLat": 40.078125,
```

```
"maxLat": 39.375
}, {
"geoHash": -1781032915136675840,
"count": 32408,
"boundingBox": {
"maxLon": 116.71875,
"minLon": 116.015625,
"minLat": 40.078125,
"maxLat": 39.375
}, {
"geoHash": -1780892177648320512,
"count": 304,
"boundingBox": {
"maxLon": 116.015625,
"minLon": 115.3125,
"minLat": 40.78125,
"maxLat": 40.078125
}, {
"geoHash": -1780751440159965184,
"count": 1537,
"boundingBox": {
"maxLon": 116.71875,
"minLon": 116.015625,
"minLat": 40.78125,
"maxLat": 40.078125
}
}],
"maxCount": 32408
}
```

返回参数说明

| 参数名 | 类型 | 说明 |
|-------------|--------|------------------|
| boundingBox | object | 网格的 bbox |
| count | long | 网格的内目标的数量 |
| geoHash | long | 网格的 geohash 整型表达 |
| maxCount | long | 本次返回最大聚合数量 |

（二）单目标查询

简要描述：单目标最新数据查询，通过目标编码 **targetId** 精确查询或 **targetName** 模糊匹配。

请求 URL：geiot/api/v1/data/target?sourceId=1&targetName=1233

请求方式：GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|------------|----|--------|----------------------------------|
| sourceId | 是 | string | 数据源 Id |
| targetId | 否 | string | 目标 Id |
| targetName | 否 | string | 目标名称（targetId targetName 不能同时为空） |

请求示例

- **targetId 精确查询** <http://192.168.4.38:8319/geiot/api/v1/data/target?sourceId=1&targetName=1233>
- **targetName 模糊匹配** <http://192.168.4.38:8319/geiot/api/v1/data/target?sourceId=1&targetId=13301130340>

返回示例

```
[
{
"sourceId": "1",
"targetType": "5",
"targetId": "13301130340",
"targetName": "京 BM1233",
"targetTime": 1566195882045,
"lon": 116.268,
"lat": 39.8934,
"extend": null,
"alt": 0,
```

```
"heading": 0
},
{
"sourceId": "1",
"targetType": "5",
"targetId": "8613436736845",
"targetName": "京 BP1233",
"targetTime": 1566195855376,
"lon": 116.395,
"lat": 39.989,
"extend": null,
"alt": 0,
"heading": 0
},
{
"sourceId": "1",
"targetType": "5",
"targetId": "8613488636234",
"targetName": "京 BQ1233",
"targetTime": 1566195835788,
"lon": 116.301,
"lat": 39.9699,
"extend": null,
"alt": 0,
"heading": 0
}
]
```

返回参数说明

| 参数名 | 类型 | 说明 |
|------------|--------|--------|
| sourceId | string | 数据源 Id |
| targetId | string | 目标 Id |
| targetName | string | 目标名称 |
| targetTime | long | 目标时间 |

| 参数名 | 类型 | 说明 |
|------------|--------|----------------|
| targetType | string | 目标类型 |
| alt | double | 高程 |
| extend | string | 扩展内容 |
| heading | double | 目标朝向方位角（0~360） |
| lat | double | 坐标经度 |
| lon | double | 坐标纬度 |

（三）多目标查询

简要描述：根据 bBox 查询区域内的目标

请求 URL：geoiot/api/v1/data/targets?sourceId=1&minLon=116.3&maxLon=116.35

7&minLat=39.9&maxLat=40.0

请求方式：Get

参数：

| 参数名 | 必选 | 类型 | 说明 |
|----------|----|--------|-------------------------|
| sourceId | 是 | string | 数据源 Id |
| duration | 否 | long | 数据期限（单位毫秒，若为空则以服务器配置为准） |
| minLon | 是 | double | 区域范围——最小经度 |
| maxLon | 是 | double | 区域范围——最大经度 |
| minLat | 是 | double | 区域范围——最小纬度 |
| maxLat | 是 | double | 区域范围——最大纬度 |

请求示例

http://192.168.4.38:8319/geoiot/api/v1/data/targets?sourceId=1&minLon=116.3&maxLon=116.357&minLat=39.9&maxLat=40.0

返回示例

```
[
{
"sourceId": "1",
"targetType": "5",
"targetId": "658152495355",
"targetName": "京 BR3046",
"targetTime": 1566151675808,
```

```

"lon": 116.302,
"lat": 39.9615,
"alt": 0,
"heading": 0
}, ...
]

```

返回参数说明

| 参数名 | 类型 | 说明 |
|------------|--------|-----------------|
| sourceId | string | 数据源 Id |
| targetId | string | 目标 Id |
| targetName | string | 目标名称 |
| targetTime | long | 目标时间 |
| targetType | string | 目标类型 |
| alt | double | 高程 |
| heading | double | 目标朝向方位角 (0~360) |
| lat | double | 坐标经度 |
| lon | double | 坐标纬度 |

备注

更多返回错误代码请看首页的错误代码描述

(四) 历史轨迹查询

简要描述：单目标历史轨迹查询

请求 URL：geoiot/api/v1/data/track

请求方式：GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|-----------|----|--------|--------------|
| sourceId | 是 | string | 数据源 Id |
| targetId | 是 | string | 目标 Id |
| startTime | 是 | long | 轨迹开始时间（单位毫秒） |
| endTime | 是 | long | 轨迹结束时间（单位毫秒） |
| minLon | 否 | double | 区域范围——最小经度 |
| maxLon | 否 | double | 区域范围——最大经度 |

| 参数名 | 必选 | 类型 | 说明 |
|--------|----|--------|------------|
| minLat | 否 | double | 区域范围——最小纬度 |
| maxLat | 否 | double | 区域范围——最大纬度 |

备注：若 minLon maxLon minLat maxLat 均不为空，则进行空间范围过滤，否则不进行空间过滤

请求示例

- 不做空间过滤 <http://192.168.4.38:8319/geoiot/api/v1/data/track?sourceId=1&targetId=658058585054&startTime=1566195502019&endTime=1566195782519>
- 空间过滤 <http://192.168.4.38:8319/geoiot/api/v1/data/track?sourceId=1&targetId=658058585054&startTime=1566195502019&endTime=1566195782519&minLon=116.30&maxLon=116.362&minLat=39.9120&maxLat=39.9127>

返回示例

```
{
  "sourceId": "1",
  "targetType": "5",
  "targetId": "658058585054",
  "targetName": "京 BQ9915",
  "extend": null,
  "minTime": 1566195546344,
  "maxTime": 1566195605697,
  "track": [
    {
      "targetTime": 1566195546344,
      "lon": 116.357,
      "lat": 39.913
    },
    {
      "targetTime": 1566195663032,
      "lon": 116.362,
      "lat": 39.9127
    }
  ]
}
```

```

"targetTime": 1566195752732,
"lon": 116.368,
"lat": 39.9124
},
{
"targetTime": 1566195782519,
"lon": 116.368,
"lat": 39.9087
},
{
"targetTime": 1566195605697,
"lon": 116.36,
"lat": 39.9127
}
]
}

```

返回参数说明

| 参数名 | 类型 | 说明 |
|------------|--------|--------|
| sourceId | string | 数据源 Id |
| targetId | string | 目标 Id |
| targetName | string | 目标名称 |
| targetType | string | 目标类型 |
| extend | string | 扩展内容 |
| lat | double | 坐标经度 |
| lon | double | 坐标纬度 |
| targetTime | long | 目标时间 |

（五）接收实时位置数据（http 方式）

简要描述：通过 http 方式，给实时数据服务发送实时位置数据

请求 URL：geoiot/api/v1/input/http

请求方式：POST

参数：

| 参数名 | 必选 | 类型 | 说明 |
|------------|----|--------|--------|
| sourceId | 是 | string | 数据源 Id |
| targetId | 是 | string | 目标 Id |
| targetName | 是 | string | 目标名称 |
| targetType | 是 | string | 目标类型 |
| extend | 否 | string | 扩展内容 |
| lat | 是 | double | 坐标经度 |
| lon | 是 | double | 坐标纬度 |
| targetTime | 是 | long | 目标时间 |
| alt | 是 | double | 高程 |
| heading | 是 | double | 方位角 |

请求示例

<http://192.168.4.38:8319/geoiot/api/v1/input/http>

```
{
  "alt": 0,
  "extend": "string",
  "heading": 0,
  "lat": 40.2,
  "lon": 116.1,
  "source_id": "1",
  "target_id": "1",
  "target_name": "小明的手机",
  "target_time": 1122,
  "target_type": "123"
}
```

返回示例

```
{"msg": "GvEvent 接收成功", "status": 200}
```

八、geodata 服务

（一）按数管编码查询图层信息

简要描述：根据数管编码查询影像、矢量数据发布的图层服务

请求 URL：/geodata/api/v1/{dataId}/layers

请求方式：GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|--------|----|--------|------|
| dataId | 是 | string | 数管编码 |

http://192.168.4.96:9091/geodata/api/v1/0/layers

返回示例

```
[
{
"dataId": 0,
"fileType": "string",
"geoDataId": 0,
"gridSet": "string",
"id": 0,
"layerName": "string",
"layerType": "IMAGELAYER",
"maxLevel": 0,
"maxX": 0,
"maxY": 0,
"mimeType": "JPEG",
"minLevel": 0,
"minX": 0,
"minY": 0,
"status": "string",
"styleName": "string",
"tileLayers": [
{
```

```
"id": 0,
"layermember": {
  "createTime": "2019-08-21T07:27:34.363Z",
  "dataEntityId": 0,
  "fileType": "string",
  "fullLayerName": "string",
  "gridSet": "EPSG:4490",
  "id": 0,
  "layerName": "string",
  "layerStatus": "string",
  "layerType": "IMAGELAYER",
  "maxLevel": 0,
  "maxX": 0,
  "maxY": 0,
  "mimeType": "JPEG",
  "minLevel": 0,
  "minX": 0,
  "minY": 0,
  "parentId": 0,
  "progress": 0,
  "seedStatus": "WAITING",
  "status": "string",
  "styleName": "string",
  "tableName": "string",
  "tileLayers": [
    null
  ],
  "transparentColor": "string",
  "userId": "string",
  "workspace": "string"
},
"sort": 0
}
```

```

],
"tileStatus": "WAITING",
"workspace": "string"
}
]

```

返回参数说明

| 参数名 | 类型 | 说明 |
|------------------|---------|--------------|
| dataId | Long | 数管编码 |
| fileType | String | 文件类型 |
| geoDataId | Long | 图层唯一标识 |
| gridSet | String | 网格集，包括坐标系等信息 |
| id | Long | id 值 |
| layerName | String | 图层名 |
| layerType | String | 图层类型 |
| maxLevel | Integer | 最大层级 |
| maxX | Double | 最大 X 坐标 |
| maxY | Double | 最大 Y 坐标 |
| mimeType | String | 图片类型 |
| minLevel | Integer | 最小层级 |
| minX | Double | 最小 X 坐标 |
| minY | Double | 最小 Y 坐标 |
| parentId | String | parentId 值 |
| progress | String | 进度 |
| seedStatus | String | 图层切片状态 |
| status | String | 发布状态 |
| styleName | String | 样式名 |
| tableName | String | 表名 |
| tileLayers | List | tile 图层 |
| transparentColor | String | 透明色 |
| userId | String | 用户编码 |
| workspace | String | 工作空间 |

| 参数名 | 类型 | 说明 |
|------------|--------|---------|
| sort | String | 排序 |
| tileStatus | String | tile 状态 |

（二）查询所有图层

简要描述： 查询全部的图层信息（包括矢量数据 影像数据的图层信息）

请求 URL： /geodata/api/v1/layers

请求方式： GET

请求参数说明：

| 参数名 | 必选 | 类型 | 说明 |
|-----------|----|---------|--------------|
| page | 否 | integer | 页数 |
| isBase | 否 | boolean | 是否为基础影像 |
| layerName | 否 | string | 图层名称 |
| layerType | 否 | string | 图层的类型 |
| order | 否 | string | 通过某一字段进行默认排序 |
| pageSize | 否 | integer | 一页上显示的图层数量 |
| sortBy | 否 | string | 排序时用到的字段 |

请求示例：

http://192.168.4.96:9091/geodata/api/v1/layers/

返回示例

```
{
  "data": [
    {
      "id": 1004,
      "layerName": "China_shengjie-PNG-4326",
      "workspace": "icenter",
      "layerType": "VECTORLAYER",
      "minX": 73.4469604492188,
      "minY": 3.40847730636597,
      "maxX": 135.085830688477,
      "maxY": 53.5579261779785,
    }
  ]
}
```

```
"minLevel": 0,
"maxLevel": 5,
"mimeType": "PNG",
"fileType": "SHP"
},
{
  "id": 1006,
  "layerName": "world-PNG-4326",
  "workspace": "icenter",
  "layerType": "VECTORLAYER",
  "minX": -180,
  "minY": -90,
  "maxX": 180,
  "maxY": 83.6022033691406,
  "minLevel": 0,
  "maxLevel": 5,
  "mimeType": "PNG",
  "fileType": "SHP",
  "styleName": "generic"
},
{
  "id": 1007,
  "layerName": "LayerGroup-PNG-4326",
  "workspace": "icenter",
  "layerType": "LAYERGROUP",
  "minX": -180,
  "minY": -90,
  "maxX": 180,
  "maxY": 90,
  "minLevel": 0,
  "maxLevel": 13,
  "mimeType": "PNG",
  "tileLayers": [
```

```
{
  "id": 1008,
  "layerName": "LayerGroup11-PNG-4326",
  "workspace": "icenter",
  "layerType": "LAYERGROUP",
  "minX": -180,
  "minY": -90,
  "maxX": 180,
  "maxY": 90,
  "minLevel": 0,
  "maxLevel": 5,
  "mimeType": "PNG",
  "tileLayers": [
    {
      "id": 4,
      "sort": 1,
      "layermember": {
        "id": 1006,
        "layerName": "world-PNG-4326",
        "workspace": "icenter",
        "layerType": "VECTORLAYER",
        "seedStatus": null,
        "status": "发布成功",
        "userId": "0",
        "minX": -180,
        "minY": -90,
        "maxX": 180,
        "maxY": 83.6022033691406,
        "minLevel": 0,
        "maxLevel": 5,
        "mimeType": "PNG",
        "gridSet": null,
        "dataEntityId": null,

```

```

"progress": 0,
"createTime": null,
"transparentColor": null,
"tileLayers": [],
"styleName": "generic",
"fileType": "SHP",
"tableName": null,
"parentId": 1005,
"fullLayerName": "icenter:world-PNG-4326",
"layerStatus": null
}
}
]
}
],
"total": 23
}

```

返回参数说明

| 参数名 | 类型 | 说明 |
|-------------|---------|-------------|
| dataId | Long | dataId 值 |
| fileType | String | 文件类型 |
| geoDataId | Long | geoDataId 值 |
| gridSet | String | 网格 |
| id | Long | id 值 |
| layerName | String | 图层名 |
| layerType | String | 图层类型 |
| maxLevel | Integer | 最大层级 |
| maxX | Double | 最大 X 坐标 |
| maxY | Double | 最大 Y 坐标 |
| contentType | String | 图片类型 |
| minLevel | Integer | 最小层级 |
| minX | Double | 最小 Y 坐标 |

| 参数名 | 类型 | 说明 |
|------------------|--------|----------------|
| minY | Double | 最小 Y 坐标 |
| parentId | String | parentId 值 |
| progress | String | 进度 |
| seedStatus | String | seed 状态 |
| status | String | 发布状态 |
| styleName | String | 样式名 |
| tableName | String | 表名 |
| tileLayers | List | tile 图层 |
| transparentColor | String | 透明色 |
| userId | String | 用户 id |
| workspace | String | 工作空间 |
| sort | String | 排序 |
| tileStatus | String | tile 状态 |
| dataEntityId | Long | dataEntityId 值 |
| createTime | Date | 创建时间 |
| fullLayerName | String | 全图层名 |
| layerStatus | String | layer 状态 |

（三）按图层编码查询图层

简要描述：按图层编码查询图层

请求 URL：/geodata/api/v1/layers/{id}

请求方式：GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|-----|----|---------|------|
| id | 是 | integer | 图层编码 |

请求示例

http://192.168.4.96:9091/geodata/api/v1/layers/{id}

返回示例

```
{
  "dataId": 0,
```

```
"fileType": "string",
"geoDataId": 0,
"gridSet": "string",
"id": 0,
"layerName": "string",
"layerType": "IMAGELAYER",
"maxLevel": 0,
"maxX": 0,
"maxY": 0,
"mimeType": "JPEG",
"minLevel": 0,
"minX": 0,
"minY": 0,
"status": "string",
"styleName": "string",
"tileLayers": [
{
"id": 0,
"layermember": {
"createTime": "2019-08-21T07:24:33.563Z",
"dataEntityId": 0,
"fileType": "string",
"fullLayerName": "string",
"gridSet": "EPSG:4490",
"id": 0,
"layerName": "string",
"layerStatus": "string",
"layerType": "IMAGELAYER",
"maxLevel": 0,
"maxX": 0,
"maxY": 0,
"mimeType": "JPEG",
"minLevel": 0,
```

```

"minX": 0,
"minY": 0,
"parentId": 0,
"progress": 0,
"seedStatus": "WAITING",
"status": "string",
"styleName": "string",
"tableName": "string",
"tileLayers": [
null
],
"transparentColor": "string",
"userId": "string",
"workspace": "string"
},
"sort": 0
}
],
"tileStatus": "WAITING",
"workspace": "string"
}
],
"tileStatus": "WAITING",
"workspace": "string"
}

```

返回参数说明

| 参数名 | 类型 | 说明 |
|-----------|--------|-------------|
| dataId | Long | 数管编码 |
| fileType | String | 文件类型 |
| geoDataId | Long | geoDataId 值 |
| gridSet | String | 网格 |
| id | Long | 图层编码 |
| layerName | String | 图层名 |

| 参数名 | 类型 | 说明 |
|------------------|---------|----------------|
| layerType | String | 图层类型 |
| maxLevel | Integer | 最大层级 |
| maxX | Double | 最大 X 坐标 |
| maxY | Double | 最大 Y 坐标 |
| mimeType | String | 图片类型 |
| minLevel | Integer | 最小层级 |
| minX | Double | 最小 X 坐标 |
| minY | Double | 最小 Y 坐标 |
| parentId | String | parentId 值 |
| progress | String | 进度 |
| seedStatus | String | seed 状态 |
| status | String | 发布状态 |
| styleName | String | 样式名 |
| tableName | String | 表名 |
| tileLayers | List | 瓦片图层 |
| transparentColor | String | 透明色 |
| userId | String | 用户编码 |
| workspace | String | 工作空间 |
| sort | String | 排序 |
| tileStatus | String | 图层切片状态 |
| dataEntityId | Long | dataEntityId 值 |
| createTime | Date | 创建时间 |
| fullLayerName | String | 全图层名 |
| layerStatus | String | 图层状态 |

（四）删除图层

简要描述：删除指定 id 的图层

请求 URL：/geodata/api/v1/layers/{id}

请求方式：DELETE

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|-----|----|---------|------|
| id | 是 | integer | 图层编码 |

请求示例

http://192.168.4.96:9091/geodata/api/v1/layers/1

返回示例

删除成功，无返回值。

删除失败，返回值格式如下：

```
{
  "code": "02101233",
  "message": "删除失败，数据库连接失败！"
}
```

返回参数说明

| 参数名 | 类型 | 说明 |
|---------|--------|----------------|
| code | string | 错误编码，具体参考错误码描述 |
| message | string | 错误信息描述 |

（五）查询指定名称的图层

简要描述：查询指定名称的图层

请求 URL：/geodata/api/v1/layers/name/{layerName}

请求方式：GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|-----------|----|--------|------|
| layerName | 是 | string | 图层名称 |

请求示例：

http://192.168.4.96:9091/geodata/api/v1/layers/name/mylayer

返回示例

```
{
  "dataId": 0,
  "fileType": "string",
  "geoDataId": 0,
  "gridSet": "string",
  "id": 0,
  "layerName": "string",
}
```

```
"layerType": "IMAGELAYER",
"maxLevel": 0,
"maxX": 0,
"maxY": 0,
"mimeType": "JPEG",
"minLevel": 0,
"minX": 0,
"minY": 0,
"status": "string",
"styleName": "string",
"tileLayers": [
{
"id": 0,
"layermember": {
"createTime": "2019-08-21T07:26:20.072Z",
"dataEntityId": 0,
"fileType": "string",
"fullLayerName": "string",
"gridSet": "EPSG:4490",
"id": 0,
"layerName": "string",
"layerStatus": "string",
"layerType": "IMAGELAYER",
"maxLevel": 0,
"maxX": 0,
"maxY": 0,
"mimeType": "JPEG",
"minLevel": 0,
"minX": 0,
"minY": 0,
"parentId": 0,
"progress": 0,
"seedStatus": "WAITING",
```

```

"status": "string",
"styleName": "string",
"tableName": "string",
"tileLayers": [
null
],
"transparentColor": "string",
"userId": "string",
"workspace": "string"
},
"sort": 0
}
],
"tileStatus": "WAITING",
"workspace": "string"
}

```

返回参数说明

| 参数名 | 类型 | 说明 |
|-----------|---------|-------------|
| dataId | Long | dataId 值 |
| fileType | String | 文件类型 |
| geoDataId | Long | geoDataId 值 |
| gridSet | String | 网格 |
| id | Long | id 值 |
| layerName | String | 图层名 |
| layerType | String | 图层类型 |
| maxLevel | Integer | 最大层级 |
| maxX | Double | 最大 X 坐标 |
| maxY | Double | 最大 Y 坐标 |
| mimeType | String | 图片类型 |
| minLevel | Integer | 最小层级 |
| minX | Double | 最小 X 坐标 |
| minY | Double | 最小 Y 坐标 |

| 参数名 | 类型 | 说明 |
|------------------|--------|----------------|
| parentId | String | parentId 值 |
| progress | String | 进度 |
| seedStatus | String | seed 状态 |
| status | String | 发布状态 |
| styleName | String | 样式名 |
| tableName | String | 表名 |
| tileLayers | List | tile 图层 |
| transparentColor | String | 透明色 |
| userId | String | 用户 id |
| workspace | String | 工作空间 |
| sort | String | 排序 |
| tileStatus | String | tile 状态 |
| dataEntityId | Long | dataEntityId 值 |
| createTime | Date | 创建时间 |
| fullLayerName | String | 全图层名 |
| layerStatus | String | layer 状态 |

（六）更新图层样式

简要描述：用户更新图层样式

请求 URL：/geodata/api/v1/layers/styles

请求方式：PUT

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|-----------|----|--------|------|
| layerName | 是 | string | 图层名称 |
| styleName | 是 | string | 样式名称 |

请求示例

http://192.168.4.96:9091/geodata/api/v1/layers/styles

返回示例

```
{
  "code": "02100403",
```

```
"message": "图层样式更新失败"
}
```

返回参数说明

| 参数名 | 类型 | 说明 |
|---------|--------|----------------|
| code | string | 错误编码，具体参考错误码描述 |
| message | string | 错误信息描述 |

九、tileCache 服务

(一) 基础影像

1. TMS 获取指定图层、层、行、列的资源

查询 TMS 服务元信息。

请求 URL: /tilecache/service/tms/{version}/{layerName}@{gridSet}@{mimeType}/{z}/{x}/{y}

请求方式: GET

参数:

| 参数名 | 必选 | 类型 | 说明 |
|-----------|----|--------|-----------------------------|
| version | 是 | string | 版本号 |
| layerName | 是 | string | 基础影像名称 |
| gridSet | 是 | string | 切片策略 |
| mimeType | 是 | string | 切片存储类型 |
| z | 是 | string | 层，第 0 层包括两个瓦片，每增加一层瓦片数量增加四倍 |
| x | 是 | string | 行号 |
| y | 是 | string | 列号 |

请求示例:

http://192.168.4.39:8319/tilecache/service/tms/1.0.0/t082414550-PNG-4326@EPS
G:4326@png/1/0/0

返回示例

合法请求，服务器将根据请求参数（空间坐标参考系 图层名、层、行、列信息）返回一张 256*256 大小的图像瓦片。非法请求，服务器将根据请求的异常格式返回一个异常。

2. TMS 查询所有服务元信息

简要描述：查询所有 TMS 服务元信息。

请求 URL：/tilecache/service/tms/{version}

请求方式：GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|---------|----|--------|---------------|
| version | 是 | string | 版本号，默认为 1.0.0 |

请求示例：

http://192.168.4.39:8319/tilecache/service/tms/1.0.0

返回示例

返回值为 xml 文件;与 GeoServerPlus 服务-矢量数据服务-TMS-获取所有 TMS 服务元数据的返回示例和返回参数说明相同。

3. TMS 按名称查询服务元信息

简要描述：查询 TMS 服务元信息。

请求 URL：/tilecache/service/tms/{version}/{layerName}@{gridSet}@{mimeType}

请求方式：GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|-----------|----|--------|--------|
| version | 是 | string | 版本号 |
| layerName | 是 | string | 基础影像名称 |
| gridSet | 否 | string | 切片策略 |
| mimeType | 是 | string | 切片存储类型 |

请求示例：

http://192.168.4.39:8319/tilecache/service/tms/1.0.0/t082414550-PNG-4326@EPS

G:4326@png

返回示例

返回值为 xml 文件;与 GeoServerPlus 服务-矢量数据服务-TMS-按名称查询 TMS 服务元信息的返回示例和返回参数说明相同。

4. WMS GetCapabilities 服务接口

简要描述：基础影像 WMS GetCapabilities 服务接口，主要用于获取基础影像 WMS 服务的元信息。

请求 URL： /tilecache/service/wms?layer={layerName}&request={request}

请求方式： GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|-----------|----|--------|--------------------------|
| layerName | 是 | string | 基础影像名称 |
| request | 是 | string | 请求类型，默认值 GetCapabilities |

请求示例：

http://192.168.4.98:8319/tilecache/service/wms?layer=Chinamb-PNG-4326&request=GetCapabilities

返回示例

返回值为 xml 文件;与 GeoServerPlus 服务-矢量数据服务-WMS GetCapabilities 接口返回示例和返回参数说明相同。

5. WMTS GetCapabilities 服务接口

简要描述：WMTS 基础服务 GetCapabilities 服务接口，主要用来获取基础影像服务的 WMTS 服务的元信息。

请求 URL： /tilecache/service/wmts?layer={layerName}&request={request}

请求方式： GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|-----------|----|--------|---------------------------|
| layerName | 是 | string | 基础影像名称 |
| request | 是 | string | 请求类型，默认值为 GetCapabilities |

请求示例：

http://192.168.4.98:8319/tilecache/service/wmts?layer=Chinamb-PNG-4326&request=GetCapabilities

返回示例

返回对应的 xml 文件;与 GeoServerPlus 服务-WMTS GetCapabilities 接口返回示例和返回参数说明相同。

6. WMS 基础服务

简要描述：又称为“WMS-GetMap 服务”，基础影像的 WMS 基础服务可以动态地生成具有地理参考数据的地图，这些地图通常用 GIF JPEG 或 PNG 等图像格式。

请求 URL： /tilecache/service/wms?SERVICE={service}&VERSION={version}&REQUEST={request}&BBOX={bBox}&SRS={srs}&WIDTH={width}&HEIGHT={height}&LAYERS={layers}&STYLES={styles}&FORMAT={format}&DPI={dpi}&MAP_RESOLUTION={mapResolution}&FORMAT_OPTIONS={formatOption}&TRANSPARENT={transparent}

请求方式： GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|---------------|----|---------|------------------------------|
| service | 是 | string | 服务类型。值为 “wms” |
| request | 是 | string | 请求类型，值为 “GetMap” |
| version | 是 | string | 版本号 |
| srs | 是 | string | 坐标参照系统 |
| layers | 是 | string | 值为一个或多个地图图层列表，多个图层之间用 “,” 隔开 |
| format | 是 | string | 切片存储类型 |
| TILED | 否 | boolean | 是否切片 |
| width | 是 | integer | 竖向像素值 |
| height | 是 | integer | 横向像素值 |
| mapResolution | 否 | integer | 地图分辨率 |
| formatOption | 否 | integer | 输出打印格式单位 |
| transparent | 是 | boolean | 透明色 |
| styles | 否 | string | 矢量样式 |
| bBox | 是 | string | 范围 |
| dpi | 是 | string | 每英寸长度内的像素点数 |

请求示例

http://192.168.4.96:9091/tilecache/service/wms?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&BBOX=-180,-90,180,90&SRS=EPSG:4326&WIDTH=1522&HEIGHT=762&LAYERS=icenter:world-PNG-4326&STYLES=&FORMAT=image/png&DPI=144&MAP_RESOLUTION=144&FORMAT_OPTIONS=dpi:144&TRANSPARENT=TRUE

返回示例

返回一个 png 或 jpg 的图片

7. WMTS 基础服务

简要描述：基础影像的 WMTS 基础服务，主要负责获取指定范围 指定格式的切片图像

请求 URL： /tilecache/service/wmts?layer={layer}&tilematrixset={tilematrixset}&Request=GetTile&Format={format}&TileMatrix={tileMatrix}&TileCol={TileCol}&tileRow={tileCol}

请求方式： GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|---------------|----|---------|----------------------------------|
| layer | 是 | string | 基础影像名称 |
| tilematrixset | 是 | string | 切片策略 |
| request | 是 | string | 请求类型，默认值为 GetTile |
| format | 是 | string | 切片存储类型，包括 image/png，image/jpeg 等 |
| tileMatrix | 是 | string | 切片结构 |
| tileCol | 是 | integer | 切片的列号 |
| tileRow | 是 | integer | 切片的行号 |

请求示例：

http://192.168.4.98:8319/tilecache/service/wmts?layer=Chinamb-PNG-4326&tilematrixset=EPSG:4326&Request=GetTile&Format=image/png&TileMatrix=EPSG:4326:3&TileCol=10&TileRow=3

返回参数说明

返回对应的切片图像

8. 时序影像获取时间轴

简要描述：时序影像获取时间轴接口，主要负责获取已发布时序影像的时间范围。

请求 URL： /tilecache/service/time?lat={lat}&layer={layer}&lon={lon}&request={request}&zoo={zoo}

请求方式： GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|---------|----|--------|-------------------|
| layer | 是 | string | 时序影像名称 |
| request | 是 | string | 请求类型，默认值为 gettile |

| 参数名 | 必选 | 类型 | 说明 |
|-----|----|---------|-------------|
| lat | 是 | double | 切片中心的经度 |
| lon | 是 | double | 切片中心的纬度 |
| zoo | 是 | integer | 保留字段，填写整形即可 |

请求示例：

http://192.168.4.85:8310/tilecache/service/time?lat=22.159424&layer=Macao0918-4326888-PNG-4326&lon=113.5546875&request=gettime&zoo=10

返回参数说明

返回 JSON 格式时间轴信息，如：

```
{
  "times": ["1984-10-01", "1985-12-12"],
  "timepojo": [
    { "year": 1984, "month": 10, "day": 1, "index": 0 },
    { "year": 1985, "month": 12, "day": 12, "index": 1 }
  ],
  "tile": {
    "status": "HIT",
    "created": 0,
    "blob": null,
    "xyz": [1751121919, 1338114049, 909003],
    "parameters": {
      "zoo": "909003",
      "lat": "22.159424",
      "lon": "113.5546875",
      "request": "gettime",
      "layer": "Macao0918-4326888-PNG-4326",
      "gridSetId": null,
      "parametersId": null,
      "layerName": "Macao0918-4326888-PNG-4326",
      "type": "tile",
      "blobFormat": "image/jpeg",
      "blobSize": 0
    }
  }
}
```

9. 时序影像基础服务

简要描述：时序影像基础服务，主要负责获取指定范围 指定格式的切片图像

请求 URL：/tilecache/service/time?layer={layer}&request={request}&zoom=zoom&row={row}&col={col}&yy={yy}&dd={dd}&mm={mm}

请求方式：GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|---------|----|---------|-------------------|
| layer | 是 | string | 基础影像名称 |
| request | 是 | string | 请求类型，默认值为 gettile |
| zoom | 是 | integer | 切片的层级 |
| row | 是 | integer | 切片的行号 |
| col | 是 | integer | 切片的列号 |
| yy | 是 | integer | 切片所在年份，如“1984” |
| mm | 是 | integer | 切片所在月份，如“10” |
| Dd | 是 | integer | 切片所在日，如“1” |

请求示例：

<http://192.168.4.85:8310/tilecache/service/time?request=gettile&yy=1984&dd=1&mm=10&layer=Macao0918-4326888-PNG-4326&zoo=10&row=1670&col=385>

返回参数说明

返回对应的切片图像

（二）图层管理接口

1. 按关键字查询图层信息

简要描述：按关键字查询图层元信息，包括基础影像、时序影像、地形等数据发布的图层信息

请求 URL：查询所有图层信息：/tilecache/api/v1/layers

请求说明：需要请求的 header 中包含 Authorization 信息

查询指定图层名称的图层信息：/tilecache/api/v1/layers?layerName=澳门 15 米

查询指定图层类型（如：时序影像）的图层信息：/tilecache/api/v1/layers?layerType=TIME_LAYER

请求方式：GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|-----------|----|---------|--|
| page | 否 | integer | 页数，默认为“1” |
| isBase | 否 | boolean | 是否为基础影像 |
| layerName | 否 | string | 图层名称 |
| layerType | 否 | string | 图层类型，DEMLAYER 代表地形图层，TIME_LAYER 代表时序影像图层 |
| order | 否 | string | 排列顺序，默认为“ASC” |
| pageSize | 否 | integer | 每页显示的图层数量，默认为“20” |
| sortBy | 否 | string | 以此字段值为基准进行排序，默认为“id” |

请求示例

查询所有图层信息：<http://192.168.4.38:8319/tilecache/api/v1/layers>

返回示例

```
{
  "data": [
```

```
{
  "id": 65,
  "layerName": "澳门 15 米时序影像-08270947-PNG-4326",
  "layerType": "TIME_LAYER",
  "isBase": false,
  "tileStatus": "SUCCESS",
  "minX": 113.48877,
  "minY": 22.08252,
  "maxX": 113.620605,
  "maxY": 22.236328,
  "minLevel": 10,
  "maxLevel": 13,
  "mimeType": "PNG",
  "gridSet": "EPSG:4326",
  "encrypt": false
}
{
  "id": 93,
  "layerName": "taiwan-dem-08281001",
  "layerType": "DEMLAYER",
  "isBase": false,
  "tileStatus": "SUCCESS",
  "minX": 119.999583,
  "minY": 19.999584,
  "maxX": 125.000417,
  "maxY": 25.000417,
  "minLevel": 5,
  "maxLevel": 9,
  "mimeType": "PNG",
  "gridSet": "EPSG:4326",
  "encrypt": false
}
],
```

```
"total": 2
}
```

返回参数说明

| 参数名 | 类型 | 说明 |
|------------|---------|---|
| id | int | 图层编码 |
| layerName | String | 图层名称 |
| layerType | String | 图层类型, TIMELAYER 代表时序影像图层, DEMLAYER 代表高程图层 |
| isBase | boolean | 是否为基础影像 |
| tileStatus | String | 切片状态, SUCCESS 代表切片完成 |
| mimeType | String | 瓦片格式, 包含 png、jpeg 等 |
| minLevel | int | 最小层级 |
| maxLevel | int | 最大层级 |
| maxX | int | 最大纬度 |
| maxY | int | 最大经度 |
| minX | int | 最小纬度 |
| minY | int | 最小经度 |
| gridSet | String | 格网集, 定义坐标系等信息 |
| encrypt | boolean | 是否加密, true 代表加密; false 代表未加密 |

2. 查询指定编码的图层

简要描述: 根据图层编码查询图层信息, 包括基础影像、时序影像、地形等数据发布的图层信息

请求 URL: /tilecache/api/v1/layers/{id}

请求说明: 需要请求的 header 中包含 Authorization 信息

请求方式: GET

请求参数:

| 参数名 | 必选 | 类型 | 说明 |
|-----|----|---------|------|
| id | 是 | integer | 图层编码 |

请求示例:

http://192.168.4.38:8319/tilecache/api/v1/layers/93

返回示例

```
{
  "id": 93,
  "layerName": "taiwan-dem-08281001",
  "layerType": "DEMLAYER",
  "isBase": false,
  "tileStatus": "SUCCESS",
  "minX": 119.999583,
  "minY": 19.999584,
  "maxX": 125.000417,
  "maxY": 25.000417,
  "minLevel": 5,
  "maxLevel": 9,
  "mimeType": "PNG",
  "gridSet": "EPSG:4326",
  "encrypt": false
}
```

返回参数说明

| 参数名 | 类型 | 说明 |
|------------|---------|---|
| id | int | 图层编码 |
| layerName | String | 图层名称 |
| layerType | String | 图层类型，TIMELAYER 代表时序影像图层，DEMLAYER 代表高程图层 |
| isBase | boolean | 是否为基础影像 |
| tileStatus | String | 切片状态，SUCCESS 代表切片完成 |
| mimeType | String | 瓦片格式，包含 png、jpeg 等 |
| minLevel | int | 最小层级 |
| maxLevel | int | 最大层级 |
| maxX | int | 最大纬度 |
| maxY | int | 最大经度 |
| minX | int | 最小纬度 |
| minY | int | 最小经度 |

| 参数名 | 类型 | 说明 |
|---------|---------|----------------------------|
| gridSet | String | 格网集，定义坐标系等信息 |
| encrypt | boolean | 是否加密，true 代表加密；false 代表未加密 |

3. 基础影像图层局部更新

简要描述：基础影像图层局部更新，局部更新已发布的基础影像图层；

请求 URL：/tilecache/api/v1/layers/{id}/partUpdateTask

请求说明：需要请求的 header 中包含 Authorization 信息

请求方式：POST

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|------|----|---------|------------|
| id | 是 | integer | 需要更新的图层编码 |
| path | 是 | string | 要上传更新的数据地址 |

请求示例：

http://192.168.4.38:8319/tilecache/api/v1/layers/{id}/partUpdateTask

返回示例

更新成功无返回值，

更新失败，返回值如下：

```
{
  "code": "00401501",
  "message": "不支持对此图层进行局部更新"
}
```

返回参数说明

| 参数名 | 类型 | 说明 |
|---------|--------|----------------|
| code | string | 错误编码，具体参考错误码描述 |
| message | string | 错误信息描述 |

4. 基础影像图层更新回滚

简要描述：基础影像图层更新回滚，对局部更新过的基础影像图层做回滚

请求 URL：/tilecache/api/v1/layers/{id}/restore

请求说明：需要请求的 header 中包含 Authorization 信息

请求方式：POST

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|-----|----|---------|-------|
| id | 是 | integer | 图层 id |

请求 URL：

http://192.168.4.38:8310/tilecache/api/v1/layers/{id}/restore

返回示例

回滚成功无返回值，

回滚失败，返回值如下：

```
{
  "code": "00401504",
  "message": "局部更新回滚异常"
}
```

返回参数说明

| 参数名 | 类型 | 说明 |
|---------|--------|----------------|
| code | string | 错误编码，具体参考错误码描述 |
| message | string | 错误信息描述 |

5. 删除指定编码的图层

简要描述：根据用户输入的图层 id 对图层进行删除， 包括基础影像、时序影像、地形等数据发布的图层信息

请求 URL：/tilecache/api/vi/layer/{id}

请求说明：需要请求的 header 中包含 Authorization 信息

请求方式：DELETE

参数：

| 参数名 | 必选 | 类型 | 说明 |
|-----|----|---------|------|
| id | 是 | integer | 图层编码 |

请求示例：

http://192.168.4.38:8310/tilecache/api/vi/layer/93

更新成功无返回值，

更新失败，返回值如下：

```
{
  "code": "00401501",
  "message": "删除图层失败"
}
```


返回参数说明

| 参数名 | 类型 | 说明 |
|---------|--------|------|
| code | int | 错误码 |
| message | string | 错误说明 |

(三) 服务地址

1. 基础地形服务（待完善）

简要描述：基础地形服务

请求 URL：http://{ip}:{port}/service/terrain/{layerName}

请求方式：GET

请求参数：

| 参数名 | 必选 | 类型 | 说明 |
|-----------|----|--------|------|
| layerName | 是 | string | 地形名称 |

请求示例：

http://192.168.4.38:8310/service/terrain/{layerName}

返回示例：

十、ServiceManager 系统服务

(一) 用户登录

1. 验证 token 的有效性

简要描述：用户 token 验证接口

请求 URL：/user/jwt/verify

请求方式：GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|-------|----|--------|------------------|
| token | 是 | string | 登陆后获取用户 token 信息 |

请求示例

http://192.168.4.38:8310/user/jwt/verify

返回示例

无返回值，根据状态码判断

2. 用户登陆接口

简要描述：用户登陆接口，获取用户的基本信息和角色信息等。

请求 URL： /user/jwt/token

请求方式：POST

参数：

| 参数名 | 必选 | 类型 | 说明 |
|----------|----|--------|-------|
| username | 是 | string | 用户名 |
| password | 是 | string | 加密后密码 |

返回示例

注意事项：传输的密码为 RSA 加密之后的密文

```
{
  "token": "eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJhZG1pbGlzInVzZXJJZCI6IjAiLCJuYW11Ijoia566h55CG5ZGYIiwicm9sZXMiOiJbMSwgMiwgM10iLCJleHAiOjE1NjU3MDY2OTd9.YURRGiD1ZuodHLJkWNiQGy9QBF-h9R0e12faIC3WH4v0NkNZJHy_VLbi3lv8zaC7j3oMWD1tqN9RNN2XcxfMEWpj2RyfrjvUoZTxRKR90XRkyvQd_tA7ZGwGEqUhRTW1G1dUqDR5Jic7fjm2tluBJ5t48tmn_KG8fgrPU2wYDPk"
}
```

返回参数说明

| 参数名 | 类型 | 说明 |
|-------|--------|-------|
| token | string | token |

加密密码登陆使用说明

用户登陆涉及到两个接口：

- 1、GET /user/jwt/time
- 2、POST /user/jwt/token

第一个接口用于获取服务器的时间戳，调用之后，返回一个类似于“20191113160841”这样的字符串

第二个接口用于登陆，需要发送的参数为

```
{
  "password": "*****",
  "username": "*****"
```

```
}
```

username 为用户名，如 admin;

password 为加密之后的密码，密码加密规则如下：

通过第一个接口获取的时间戳+用户密码，得到字符串，然后将得到的字符串通过 rsa 公钥进行加密，然后将加密后的内容进行 base64 编码作为 password。

例如，获取到的时间戳为 20191113160841，用户的密码为 abcde，那么就需要将 20191113160841abcde 进行 rsa 加密。

rsa 公钥：

```
-----BEGIN PUBLIC KEY-----
```

```
MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBANbuVd2AW+xnDH82hNWFYjDWttKilGgt
S+tVxBQF4P0l7pkNccQNEfrklGsBvR1U4y jNhVeoLmO/Xd6WF0u0HsMCAwEAAQ==
```

```
-----END PUBLIC KEY-----
```

java 中的加密方式：

```
String encrypt(String passwd) throws InvalidKeySpecException, NoSuchAlgorithmException,
NoSuchPaddingException, InvalidKeyException, IllegalBlockSizeException, BadPaddingException,
UnsupportedEncodingException {
```

```
    String pubkeyStr = "MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBANbuVd2AW
+xnDH82hNWFYjDWttKilGgtS+tVxBQF4P0l7pkNccQNEfrklGsBvR1U4y jNhVeoLmO/Xd
6WF0u0HsMCAwEAAQ==";
```

```
    RSAPublicKey pubkey = (RSAPublicKey) KeyFactory.getInstance("
RSA").generatePublic(
```

```
        new X509EncodedKeySpec(Base64.getDecoder().decode(pubkeyStr)));
```

```
    Cipher cipher = Cipher.getInstance("RSA");
```

```
    cipher.init(Cipher.ENCRYPT_MODE, pubkey);
```

```
    String result = Base64.getEncoder().encodeToString(cipher.doFinal(
passwd.getBytes("UTF-8")));
```

```
    return result;
```

```
}
```

其中的参数 passwd，是添加过时间戳的密码，返回的结果是加密后的字符串。

3. 获取用户信息接口

简要描述：根据 token 获取用户的信息，如用户名、角色等

请求 URL： /user/api/v1/user/userinfo

请求方式：GET

参数：

| 参数名 | 必选 | 类型 | 说明 |
|---------------|----|--------|------------------------------|
| Authorization | 是 | string | 在请求头部添加的参数，参数的值是登陆时返回的 token |

返回示例

```
{
  "baseRoles": [
    {
      "id": 0,
      "roleDesc": "角色描述",
      "roleName": "角色名"
    }
  ],
  "id": 0,
  "sex": "用户性别",
  "username": "用户名"
}
```

返回参数说明

| 参数名 | 类型 | 说明 |
|--------------------|--------|---------------------|
| id | int | 用户 id |
| sex | string | 用户性别 |
| username | string | 用户名 |
| baseRoles | object | 用户所属的用户组，用户可以同时在多个组 |
| baseRoles.id | int | 用户角色 id |
| baseRoles.roleDesc | string | 用户角色描述 |
| baseRoles.roleName | string | 用户角色名称 |

十一、PlatformResource 平台资源配置服务

(一) 平台资源配置接口

1. 获取服务器目录

简要描述：获取平台服务导入、导出目录接口

请求 URL： /platformresource/api/v1/files/getSystemDirs

请求方式：POST

参数：

| 参数名 | 必选 | 类型 | 说明 |
|-----------|----|--------|----------------------------------|
| direction | 是 | string | 定向：1 代表 import 目录 2 代表 export 目录 |
| filePath | 是 | string | 父级目录 null 代表根目录 |

返回示例

```
[
  {
    "path": "/home/iCenter/data/export-data",
    "viewpath": "/export-data",
    "fileRealPath": "/data/export-data",
    "fileType": "Folder"
  },
  {
    "fileRealPath": "/mnt/mfs/GEOVIS/iCenter//data/import-data/taiwan.tif"
    "fileType": "File"
    "path": "/home/iCenter/data/import-data/taiwan.tif"
    "size": 72120693
    "state": "未入库"
    "viewpath": "/taiwan.tif"
  }
]
```

返回参数说明

| 参数名 | 类型 | 说明 |
|-----|----|----|
|-----|----|----|

| 参数名 | 类型 | 说明 |
|--------------|--------|---------------------------|
| path | string | 文件目录（docker 内虚拟目录） |
| viewpath | string | 虚拟目录（前端可视的目录） |
| fileRealPath | string | 文件真实目录（系统内真实目录） |
| fileType | string | 文件类型 Folder 为文件夹 File 为文件 |
| size | long | 文件大小 |
| state | string | 文件状态 入库，未入库 |

2. 获取物理机上路径

简要描述：平台获取物理机上实际存储路径。

请求 URL： /platformresource/api/v1/files/importFilePath

请求方式：GET

参数：无参数

返回示例

```
/mnt/mfs/GEOVIS/iCenter/data
```

返回参数说明

返回物理机存储数据位置的实际路径

3. 获取服务部署版本

简要描述：平台服务部署版本。

请求 URL： /platformresource/api/v1/files/serviceType

请求方式：GET

参数：无参数

返回示例

```
{
  "resultInfo": "LinuxSingleVersion"
}
```

返回参数说明

| 参数名 | 类型 | 说明 |
|------------|--------|------|
| resultInfo | string | 部署环境 |