

中国移动通信企业标准

QB-XX-XXX-XXXX

位置业务

个人GIS系统接口规范

Interface protocol of GIS for

Location based Service

版本号：0.1.0

XXXX-XX-XX实施

XXXX-XX-XX发布

中国移动通信有限公司 发布

**前 言**

本标准为中国移动建设GIS平台提供技术依据。

本标准由中移有限技〔2012〕 号印发。

本标准由中国移动通信集团技术部提出并归口。

本标准起草单位：中国移动通信集团辽宁有限公司。

本标准主要起草人：

目 录

[1. 范围 1](#_Toc354494509)

[2. 规范性引用文件 1](#_Toc354494510)

[3. 术语、定义和缩略语 1](#_Toc354494511)

[3.1. 术语和定义 1](#_Toc354494512)

[3.2. 缩略语 2](#_Toc354494513)

[4. GIS系统结构 2](#_Toc354494514)

[4.1. GIS系统在位置服务中的位置 2](#_Toc354494515)

[4.2. 网元功能描述 3](#_Toc354494516)

[4.3. 接口描述 3](#_Toc354494517)

[5. GIS核心服务接口说明 4](#_Toc354494518)

[5.1. POI查询接口 4](#_Toc354494519)

[5.1.1. 关键字查询 4](#_Toc354494520)

[5.1.2. 周边查询 7](#_Toc354494521)

[5.1.3. PGUID查询 21](#_Toc354494522)

[5.1.4. 拉框查询 23](#_Toc354494523)

[5.1.5. 缓冲区查询 30](#_Toc354494524)

[5.1.6. 关键字提示接口服务请求 34](#_Toc354494525)

[5.1.7. 沿线搜索 35](#_Toc354494526)

[5.2. 导航服务接口 37](#_Toc354494527)

[5.2.1. 路径导航服务请求接口 37](#_Toc354494528)

[5.2.2. 导航途经点信息请求接口 42](#_Toc354494529)

[5.3. 公交查询接口 45](#_Toc354494530)

[5.3.1. 公交路径导航请求接口 45](#_Toc354494531)

[5.3.2. 公交线路服务请求接口 49](#_Toc354494532)

[5.4. 地址匹配接口 61](#_Toc354494533)

[5.4.1. 接口功能 61](#_Toc354494534)

[5.4.2. 接口参数说明 61](#_Toc354494535)

[5.4.3. 接口URL示例 61](#_Toc354494536)

[5.4.4. 返回结果说明 61](#_Toc354494537)

[5.4.5. 其他需要说明的部分 63](#_Toc354494538)

[5.5. 位置描述接口 63](#_Toc354494539)

[5.5.1. 单点位置描述 63](#_Toc354494540)

[5.5.2. 多点位置描述 67](#_Toc354494541)

[5.5.3. 简化版位置描述 68](#_Toc354494542)

[5.6. 道路查询接口 69](#_Toc354494543)

[5.6.1. 根据道路名称查询道路信息 69](#_Toc354494544)

[5.6.2. 查询两条道路的交叉口位置 71](#_Toc354494545)

[5.6.3. 查询一条道路上的所有交叉路口位置 74](#_Toc354494546)

[5.7. 空间计算接口 76](#_Toc354494547)

[5.7.1. 位置判断接口 76](#_Toc354494548)

[5.7.2. 距离计算接口 78](#_Toc354494549)

[5.7.3. 计算路径缓冲区范围 80](#_Toc354494550)

[5.7.4. 面积计算接口 83](#_Toc354494551)

[5.8. 坐标处理接口 84](#_Toc354494552)

[5.8.1. 经纬度与屏幕像素坐标转换 84](#_Toc354494553)

[5.9. WMAP取图接口 86](#_Toc354494554)

[5.9.1. 接口功能介绍 86](#_Toc354494555)

[5.9.2. 接口参数说明 86](#_Toc354494556)

[5.9.3. 接口URL示例 93](#_Toc354494557)

[5.9.4. 返回结果说明 93](#_Toc354494558)

范围

本规范规定了中国移动开通 LBS业务的位置业务平台和 GIS平台之间，以及 SP与位置业务平台之间涉及 GIS访问功能的接口协议。包括消息的类型和消息的元素的定义。 开通 LBS业务、且用到 GIS访问的中国移动各省公司、LBS平台提供商、应用服务提供商（SP）以及GIS平台提供商在向中国移动提供 LBS业务时，应遵循本规范。本规范供中国移动通信有限公司内部和厂商共同使用，可作为总公司和省公司 LBS功能开发的技术依据。适用于GSM/GPRS/EDGE/TD-SCDMA网络环境。

规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本标准，然而，鼓励根据本标准达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本标准。

表 2-1 规范性引用文件

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **标准编号** | **标准名称** | **发布单位** |
| [1] |  | Hypertext Transfer Protocol–HTTP/1.1 | RFC 2616, June 1999 |
| [2] |  | Extensible Markup Language  (XML) 1.0 | W3C Recommendation:  REC-xml-20001006 |

术语、定义和缩略语

“必须”、“推荐”/“建议”、和“可选”等词语在本标准中的使用需遵循以下指导。

- “必选” / “必须” 项是指业务、产品和设备所必须提供的功能或性能要求；对应于RFC21MUST，REQUIRED，SHALL。

- “推荐”/“建议”/“应”项是指在标准中未作强制要求，若业务、产品和设备提供的能或性能要求被认为更佳；对应于RFC2119 RECOMMENDED，SHOULD。

- “可选”/“可”项指参考性要求，是业务、产品和设备在目前阶段可不提供的功能或能要求；对应于RFC2119 MAY，OPTIONAL。

- 必不能，不能，不得：表示绝对的禁止；对应于RFC2119 MUST NOT，SHALL NOT。

- 不推荐， 不建议： 表示若业务、 产品和设备按照所述内容制作， 被认为略次； 对应于RFC21SHOULD NOT，NOT RECOMMENDED。

规范中除了明确指明为 “推荐”/“建议”、“可选”外，均为必须要求。

* 1. 术语和定义

表3-1 术语/定义

|  |  |
| --- | --- |
| **术语/定义** | **解释** |
| 智能纠错 | 针对用户输入的错词，生活搜索返回该搜索词对应的最可能正确词列表，如用户输入“文党”，智能纠错系统要能够给出对应的纠错词”文档”。 |
| 搜索提示 | 通过互联网进行搜索时，能够对用户的输入做出动态的提示，减轻用户输入负担。 |
| 搜索推荐 | 平台通过挖掘所有用户日志，建立递进关系和并列关系推荐模型，对用户的查询进行相关推荐。 |
| 导航参数 | 请求返回导航数据的类型标示，用来是否提供道路名称、坐标数据、图片数据、视频数据、文本数据、声音数据等信息。一个bit位对应一种类型数据类型。 |

* 1. 缩略语

表3-2 缩略语

|  |  |  |
| --- | --- | --- |
| 缩略语 | 英文全称 | 中文含义 |
| DTD | Document Type Definition | XML文档定义 |
| LSP | Location Service Platform | 位置服务平台 |
| GMT | Greenwich Mean Time | 格林威治时间 |
| HTTP | Hypertext Transfer Protocol | 超文本传输协议 |
| HTTPS | HTTP Secure | 超文本传输协议安全 |
| GIS | Geographic Information System | 地理信息系统 |
| SSL | Secure Socket Layer | 加密套接字协议层 |
| URI | Uniform Resource Identifier | 统一资源标识符 |
| URL | Uniform Resource Locator | 全球资源定位器 |
| UTM | Universal Transverse Mercator | 通用墨卡托投影 |
| XML | Extensible Markup Language | 扩展标记语言 |
| LBS | Location Based Service | 基于位置的服务 |

GIS系统结构

* 1. GIS系统在位置服务中的位置



图-4-1GIS系统在位置服务中的位置示意图

本文将对WAP、短彩信及其他应用调用GIS WEB服务版接口进行说明（上图标注红色部分）。

* 1. 网元功能描述

**客户端**

* PC浏览器应用：通过调用GIS的JavaScript版地图API开发的Web应用，运行于PC浏览器。
* WAP、短彩信及其他应用：通过调用GIS的Web服务版地图API开发的WAP、短彩信及其他应用，客户端请求先发送至业务平台，由业务平台将请求转换成标准接口协议发送至GIS系统。

**业务层**

* 地图业务平台：为客户端提供位置应用及服务的业务系统。如手机地图HMP平台，车载导航NIP平台，面向用户提供手机地图、车载导航等应用服务，其负责业务页面的呈现，与MSP（手机业务管理服务器）交互处理请求、用户请求结果下发，业务数据的统计等。

**资源层**

* GIS系统：开放GIS能力并响应应用的请求，提供与之需求相对应的GIS服务功能，如地图服务、地理编码服务、逆地理编码服务、路径搜索服务等。
  1. 接口描述

I\_1接口：JavaScript版API调用后台GIS服务的接口，接口类型为REST，详见《位置业务GIS系统JavaScript服务版API开发手册》。

快速入门

以下全文中的请求url中 “http://xxIP:xx/xxxx?” 应该替换为实际的GIS服务的正式地址：  [http://223.100.246.24：XX/XX](%20http://223.100.246.24：XX/XX)?”.

http://223.100.246.24:8096

基于WEB服务版地图API，开发的WAP、短彩信及其他应用，下面为请求地址示例：

|  |
| --- |
| 请求URL：  请求地址示例(以调用关键字查询接口为例):  http://223.100.246.24:8080/mapapi?ability=apiserver&abilityuri=sisserver&config=BESN&resType=xml&cityCode=010&searchName=%BE%C6%B5%EA&number=10&batch=1&srctype=USERPOI1000%2BPOI&a\_k=\*\*\*\*\* |

参数说明：

a\_k:位置基地为业务分配的key值

GIS核心服务接口说明

* 1. POI查询接口
     1. 关键字查询
        1. 接口功能介绍

用户输入关键字，系统根据输入的关键字进行POI查询，然后返回查询结果。

* + - 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| config | config=BESN | 关键字查询服务 | 必填参数 |
| cityCode | cityCode=城市编码 | 城市编码:支持编码和名称形式，如010或者北京，需要进行urlencode编码，编码格式参照enc参数指定，  全国为total | 必填参数 |
| ability | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必添参数 |
| searchName | searchName=查询名称 | 为查询名称:默认匹配名称,地址,类型,需要进行urlencode编码，编码格式参照enc参数指定 | 必填参数 |
| searchType | searchType=查询类型 | 查询类型:POI分类类型,需要进行urlencode编码，编码格式参照enc参数指定 | 可选参数 |
| number | number=返回的POI结果个数 | 每页记录数 | 必填参数 |
| batch | batch=POI结果显示的页数 | 分页号 | 必填参数 |
| resType | resType=xml/json | 返回格式 | 可选参数，支持xml和json格式默认xml |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| srctype | srctype=数据源类型 | 数据源类型:  基础库 POI  企业地标 DIBIAO  公交站台 BUS  通用编辑器 USERPOI  上述数据源可以混合查询，例如公交数据+基础库数据查询  srctype=BUS:1000%2bPOI,代表前1000条数据为公交数据，后面为基础库数据 | 可选参数，默认 ENPOI:3%2bPOI  代表前三条返回地标数据，后面返回基础库数据。 |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

* + - 1. 接口URL示例

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&config=BESN&resType=xml&cityCode=010&searchName=%E9%85%92%E5%BA%97&number=10&batch=1&enc=utf-8&a\_k=[用户KEY]

**检索基础库数据：**

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&enc=utf-8&config=BESN&resType=xml&cityCode=010&searchName=%e9%85%92%e5%ba%97&number=10&batch=1&a\_k=[用户KEY]

**检索公交数据：**

[http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&enc=utf-8&config=BESN&resType=xml&cityCode=010&searchName=%e9%85%92%e5%ba%97&number=10&batch=1&srctype=BUS&a\_k](http://search1.mapabc.com/sisserver?&config=BESN&resType=xml&cityCode=010&searchName=%BE%C6%B5%EA&number=10&batch=1&srctype=BUS&a_k)=[用户KEY]

**检索编辑器数据：**

[http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&enc=utf-8&config=BESN&resType=xml&cityCode=010&searchName=%e9%85%92%e5%ba%97&number=10&batch=1&srctype=USERPOI&a\_k](http://search1.mapabc.com/sisserver?&config=BESN&resType=xml&cityCode=010&searchName=%BE%C6%B5%EA&number=10&batch=1&srctype=USERPOI&a_k)=[用户KEY]

**检索基础库+编辑器数据,其中前1000条编辑器数据优先显示**

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver[&enc=utf-8&config=BESN&resType=xml&cityCode=010&searchName=%e9%85%92%e5%ba%97&number=10&batch=1&srctype=USERPOI：1000%2BPOI&a\_k](http://search1.mapabc.com/sisserver?&config=BESN&resType=xml&cityCode=010&searchName=%BE%C6%B5%EA&number=10&batch=1&srctype=USERPOI：1000%2BPOI&a_k)=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - 1. 返回结果说明

<?xml version="1.0" encoding="GBK"?>

<searchresult servername="服务器名" versionname=”版本号”>

<count>此接口返回的结果集数量（默认最大结果集为400，可通过key权限修改，需要联系管理员）</count>

<cache>此数据是否从缓存读取</cache>

<total>引擎返回的结果集总数量</total>

<record>此页返回的结果集数量（最大结果集为20）</record>

<searchtime>查询时间</searchtime>

<bounds>bounds值，当前返回的POI点统一个矩形框内，bounds为矩形框的左上右下坐标对</bounds>

<list type="**list**">

<poi>

<address>地址</address>

<name>名称</name>

<type>数据类型</type>

<timestamp>时间戳</timestamp>

<match>匹配度</match>

<url >url</url>

<citycode>城市代码</citycode>

<xml >扩展描述信息</xml>

<x>x坐标</x>

<y>y坐标</y>

<code>区域代码</code>

<distance>距离查询点距离</distance>

<drivedistance>导航距离查询点距离</drivedistance>

<pguid>索引id</pguid>

<srctype>数据源类型</srctype>

<gridcode>gridcode代码</gridcode>

<tel>联系电话</tel>

<buscode>公交代码</buscode>

<imageid>图片id</imageid>

<linkid>关联id</linked>

<newtype>数据类型编码</newtype>

</poi>

</list>

<pinyin>

<list/>

<pinyin>

</searchresult>

* + - 1. 其他需要说明的部分

**公交类型POI扩展字段说明：**

包含多个STATION元素。station为站点信息，包含字段如下表：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| NAME |  | 站点名称 |  |
| XY\_COORDS |  | 站点坐标，以;分隔。如：116.332433;39.745734 |  |
| SPELL |  | 站点名称拼音 |  |
| Code |  | 站点所在城市行政区划码 |  |
| STATION\_NUM |  | 站序 |  |
|  |  |  |  |
|  |  |  |  |

**示例：**

<?xml version="1.0" encoding="UTF-8"?>

<CONTENT>

<STATION>

<DATA NAME="NAME"><![CDATA[清城]]></DATA>

<DATA NAME="XY\_COORDS"><![CDATA[116.327246;39.737597]]></DATA>

<DATA NAME="SPELL"><![CDATA["qing cheng"]]></DATA>

<DATA NAME="Code"><![CDATA[110100]]></DATA>

<DATA NAME="STATION\_NUM"><![CDATA[1]]></DATA>

</STATION>

<STATION>

<DATA NAME="NAME"><![CDATA[清源西里]]></DATA>

<DATA NAME="XY\_COORDS"><![CDATA[116.332433;39.745734]]></DATA>

<DATA NAME="SPELL"><![CDATA["qing yuan xi li"]]></DATA>

<DATA NAME="Code"><![CDATA[110100]]></DATA>

<DATA NAME="STATION\_NUM"><![CDATA[2]]></DATA>

</STATION>

<STATION>

<DATA NAME="NAME"><![CDATA[清源西里南站]]></DATA>

<DATA NAME="XY\_COORDS"><![CDATA[116.339311;39.74475]]></DATA>

<DATA NAME="SPELL"><![CDATA["qing yuan xi li nan zhan"]]></DATA>

<DATA NAME="Code"><![CDATA[110100]]></DATA><DATA NAME="STATION\_NUM">

<![CDATA[3]]></DATA>

</STATION>

</CONTENT>

* + 1. 周边查询
       1. 根据中心点的坐标（x，y）查询周边POI
          1. 接口功能介绍

在基础库库和地标库中，根据中心点的坐标（x，y）, 查询周边POI。

* + - * 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| ability | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必添参数 |
| config | config=BELSBXY | 服务参数名称 | 必填参数 |
| cityCode | cityCode=城市编码 | 城市编码:支持编码和名称形式，如010或者北京，需要进行urlencode编码，编码格式参照enc参数指定，  全国为total | 必填参数 |
| searchName | searchName=查询名称 | 为查询名称:默认匹配名称,地址,类型,需要进行urlencode编码，编码格式参照enc参数指定 | 必填参数 |
| searchType | searchType=查询类型 | 查询类型:POI分类类型,需要进行urlencode编码，编码格式参照enc参数指定 | 可选参数 |
| cenX | cenX=中心点x | 中心点x坐标 | 必填参数 |
| cenY | cenX=中心点y | 中心点y坐标 | 必填参数 |
| range | 周边查询范围 | 周边查询时，圆型区域的 | 必填参数，默认3000 |
| number | number=返回的POI结果个数 | 每页记录数 | 必填参数 |
| batch | batch=POI结果显示的页数 | 分页号 | 必填参数 |
| resType | resType=xml/json | 返回格式 | 可选参数，支持xml和json格式默认xml |
| srctype | srctype=数据源类型 | 数据源类型:  基础库 POI  企业地标 DIBIAO  公交站台 BUS  通用编辑器 USERPOI  上述数据源可以混合查询，例如公交数据+基础库数据查询  srctype=BUS:1000%2bPOI,代表前1000条数据为公交数据，后面为基础库数据 | 可选参数，默认 ENPOI:3%2bPOI  代表前三条返回地标数据，后面返回基础库数据。 |
| naviflag | naviflag=0/1 | 导航距离排序标识 | 可选参数  默认为0，不按照导航距离排序，如果为1将按照导航距离排序，但性能比直线距离排序差 |
| sr | sr=排序参数 | sr=排序参数 | 编辑器周边查询中查询中，sr=1代表按照距离排序 |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

* + - * 1. 接口URL示例

[http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&enc=utf-8&config=BELSBXY&resType=xml&cityCode=010&cenX=116.4&cenY=39.9&searchName=](http://sisv3.test.mapabc.com/sisserver?&config=BELSBXY&resType=xml&cityCode=010&cenX=116.4&cenY=39.9&searchName=)大学&number=10&batch=1&range=3000&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - * 1. 返回结果说明

<?xml version="1.0" encoding="GBK"?>

<searchresult servername="服务器名" versionname=”版本号”>

<count>此接口返回的结果集数量（默认最大结果集为400，可通过key权限修改，需要联系管理员）</count>

<cache>此数据是否从缓存读取</cache>

<total>引擎返回的结果集总数量</total>

<record>此页返回的结果集数量（最大结果集为20）</record>

<searchtime>查询时间</searchtime>

<bounds>bounds值，当前返回的POI点统一个矩形框内，bounds为矩形框的左上右下坐标对</bounds>

<list type="**list**">

<poi>

<address>地址 </address>

<name>名称</name>

<type>数据类型</type>

<timestamp>时间戳</timestamp>

<match>匹配度</match>

<url >url</url>

<citycode>城市代码</citycode>

<xml >扩展描述信息</xml>

<x>x坐标</x>

<y>y坐标</y>

<code>区域代码</code>

<distance>距离查询点距离</distance>

<drivedistance>导航距离查询点距离</drivedistance>

<pguid>索引id</pguid>

<srctype>数据源类型</srctype>

<gridcode>gridcode代码</gridcode>

<tel>联系电话</tel>

<buscode>公交代码</buscode>

<imageid>图片id</imageid>

<linkid>关联id</linked>

<newtype>数据类型编码</newtype>

</poi>

</list>

<cenpoi>

<x>中心点x坐标 </x>

<y>中心点y坐标</y>

<citycode/>

<drivedistance>导航距离查询点距离</drivedistance>

<xml/>

</cenpoi>

</searchresult>

* + - * 1. 其他需要说明的部分
      1. 根据中心点关键字查询周边POI
         1. 接口功能介绍

给定一个点的xy坐标，以此点为中心点进行周边查询，返回查询结果

* + - * 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| config | config=BELSBN | 服务参数名称 | 必填参数 |
| ability | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必添参数 |
| cityCode | cityCode=城市编码 | 城市编码:支持编码和名称形式，如010或者北京，需要进行urlencode编码，编码格式参照enc参数指定，全国为total | 必填参数 |
| searchName | searchName=查询名称 | 为查询名称:默认匹配名称,地址,类型,需要进行urlencode编码，编码格式参照enc参数指定 |  |
| searchType | searchType=查询类型 | 查询类型:POI分类类型,需要进行urlencode编码，编码格式参照enc参数指定 | 必填参数 |
| cenName | cenName=中心点名称 | 中心点名称 |  |
| range | 周边查询范围 | 周边查询时，圆型区域的 | 必填参数 |
| number | number=返回的POI结果个数 | 每页记录数 | 必填参数，默认3000 |
| batch | batch=POI结果显示的页数 | 分页号 | 必填参数 |
| resType | resType=xml/json | 返回格式 | 可选参数，支持xml和json格式默认xml |
| srctype | srctype=数据源类型 | 数据源类型:  基础库 POI  企业地标 DIBIAO  公交站台 BUS  通用编辑器 USERPOI  上述数据源可以混合查询，例如公交数据+基础库数据查询  srctype=BUS:1000%2bPOI,代表前1000条数据为公交数据，后面为基础库数据 | 可选参数，默认 ENPOI:3%2bPOI  代表前三条返回地标数据，后面返回基础库数据。 |
| naviflag | naviflag=0/1 | 导航距离排序标识 | 可选参数  默认为0，不按照导航距离排序，如果为1将按照导航距离排序，但性能比直线距离排序差 |
| sr | sr=排序参数 | sr=排序参数 | 编辑器周边查询中查询中，sr=1代表按照距离排序 |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

* + - * 1. 接口URL示例

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&srctype=POI&config=BELSBN&searchName=%E9%85%92%E5%BA%97&cityCode=010&cenName=%E5%A4%A7%E6%81%92%E7%A7%91%E6%8A%80%E5%A4%A7%E5%8E%A6&searchType=%E4%BD%8F%E5%AE%BF&number=10&batch=1&resType=xml&enc=utf-8&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - * 1. 返回结果说明

<?xml version="1.0" encoding="GBK"?>

<searchresult servername="服务器名" versionname=”版本号”>

<count>此接口返回的结果集数量（默认最大结果集为400，可通过key权限修改，需要联系管理员）</count>

<cache>此数据是否从缓存读取</cache>

<total>引擎返回的结果集总数量</total>

<record>此页返回的结果集数量（最大结果集为20）</record>

<searchtime>查询时间</searchtime>

<bounds>bounds值，当前返回的POI点统一个矩形框内，bounds为矩形框的左上右下坐标对</bounds>

<cenpoi>

<address>地址</address>

<name>POI名称</name>

<type>POI类型</type>

<timestamp>数据时间戳</timestamp>

<match>匹配度</match>

<url>url连接</url>

<distance>周边查询时，距离中心点距离</distance>

<x>x坐标</x>

<y>y坐标</y>

<citycode>城市号</citycode>

<pguid>索引ID，可以根据此ID查询POI</pguid>

<srctype>数据源</srctype>

<gridcode>5916728612</gridcode>

<tel>电话</tel>

<imageid>图片ID</imageid>

<xml>扩展查询字段，编辑器或者公交数据中，可能存在该字段，扩展字段的含义和具体行业有关，该部分内容由应用自己去解析</xml>

<drivedistance>周边查询按照导航距离排序时，距中心点的导航距离</drivedistance>

<code>区域代码</code>

<buscode>公交代码</buscode>

<imageid > 图片id</imageid>

<linkid >关联id </linked>

<newtype>数据类型编码</newtype>

</cenpoi>

<list type="**list**">

<poi>

<address>地址 </address>

<name>名称</name>

<type>数据类型</type>

<timestamp>时间戳</timestamp>

<match>匹配度</match>

<url >url</url>

<citycode>城市代码</citycode>

<xml >扩展描述信息</xml>

<x>x坐标</x>

<y>y坐标</y>

<code>区域代码</code>

<distance>距离查询点距离</distance>

<drivedistance>导航距离查询点距离</drivedistance>

<pguid>索引id</pguid>

<srctype>数据源类型</srctype>

<gridcode>gridcode代码</gridcode>

<tel>联系电话</tel>

<buscode>公交代码</buscode>

<imageid>图片id</imageid>

<linkid>关联id</linked>

<newtype>数据类型编码</newtype>

</poi>

</list>

</searchresult>

* + - * 1. 其他需要说明的部分
      1. 根据中心点gridcode查询周边POI
         1. 接口功能介绍

给定一点的gridcode，以此点为中心点进行周边查询，返回查询结果。

* + - * 1. 接口参数说明

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数名称** | | **参数格式** | | **参数含义** | **备注** |
| ability | | ability=apiserver | | 服务类型 | 必添参数 |
| abilityuri | | abilityuri=sisserver | | 请求类型 | 必添参数 |
| config | | config=BELSBGC | | 服务参数名称 | 必填参数 |
| cenGridCode | | cenGridCode=中心点GridCode | | 中心点GridCode，不输入查询将无结果。 | 必填参数 |
| cityCode | | cityCode=城市编码 | | 城市编码:支持编码和名称形式，如010或者北京，需要进行urlencode编码，编码格式参照enc参数指定，全国为total | 必填参数 |
| searchName | | searchName=查询名称 | | 为查询名称:默认匹配名称,地址,类型,需要进行urlencode编码，编码格式参照enc参数指定 |  |
| searchType | | searchType=查询类型 | | 查询类型:POI分类类型,需要进行urlencode编码，编码格式参照enc参数指定 | 必填参数 |
| cenName | | cenName=中心点名称 | | 中心点x坐标 |  |
| range | | 周边查询范围 | | 周边查询时，圆型区域的 | 必填参数 |
| number | | number=返回的POI结果个数 | | 每页记录数 | 必填参数，默认3000 |
| batch | | batch=POI结果显示的页数 | | 分页号 | 必填参数 |
| resType | | resType=xml/json | | 返回格式 | 可选参数，支持xml和json格式默认xml |
| srctype | | srctype=数据源类型 | | 数据源类型:  基础库 POI  企业地标 DIBIAO  公交站台 BUS  通用编辑器 USERPOI  上述数据源可以混合查询，例如公交数据+基础库数据查询  srctype=BUS:1000%2bPOI,代表前1000条数据为公交数据，后面为基础库数据 | 可选参数，默认 ENPOI:3%2bPOI  代表前三条返回地标数据，后面返回基础库数据。 |
| naviflag | | naviflag=0/1 | | 导航距离排序标识 | 可选参数  默认为0，不按照导航距离排序，如果为1将按照导航距离排序，但性能比直线距离排序差 |
| enc | enc=utf-8 | | 编码设定，仅支持utf-8 | | 必填参数 |
| sr | | sr=排序参数 | | sr=排序参数 | 编辑器周边查询中查询中，sr=1代表按照距离排序 |
| a\_k | | a\_k=api key | | 用户申请的key | 必填参数 |

* + - * 1. 接口URL示例

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&enc=utf-8&config=BELSBGC&cityCode=010&cenGridCode=5916727421&searchName=%e9%85%92%e5%ba%97&number=10&batch=1&range=3000&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - * 1. 返回结果说明

<?xml version="1.0" encoding="GBK"?>

<searchresult servername="服务器名" versionname=”版本号”>

<count>此接口返回的结果集数量（默认最大结果集为400，可通过key权限修改，需要联系管理员）</count>

<cache>此数据是否从缓存读取</cache>

<total>引擎返回的结果集总数量</total>

<record>此页返回的结果集数量（最大结果集为20）</record>

<searchtime>查询时间</searchtime>

<bounds>bounds值，当前返回的POI点统一个矩形框内，bounds为矩形框的左上右下坐标对</bounds>

<list type="**list**">

<poi>

<address>地址 </address>

<name>名称</name>

<type>数据类型</type>

<timestamp>时间戳</timestamp>

<match>匹配度</match>

<url >url</url>

<citycode>城市代码</citycode>

<xml >扩展描述信息</xml>

<x>x坐标</x>

<y>y坐标</y>

<code>区域代码</code>

<distance>距离查询点距离</distance>

<drivedistance>导航距离查询点距离</drivedistance>

<pguid>索引id</pguid>

<srctype>数据源类型</srctype>

<gridcode>gridcode代码</gridcode>

<tel>联系电话</tel>

<buscode>公交代码</buscode>

<imageid>图片id</imageid>

<linkid>关联id</linked>

<newtype>数据类型编码</newtype>

</poi>

</list>

</searchresult>

* + - * 1. 其他需要说明的部分
      1. 根据两条线的交叉点坐标查询周边POI
         1. 接口功能介绍

根据两条线的交叉点坐标查询周边POI。

* + - * 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| ability | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必添参数 |
| config | config=SPAS | 服务参数名称 | 必填参数 |
| spatialXml | spatialXml=查询条件 | spatialXml参数为一个标准的xml格式 | 必填参数 |
| resType | resType=xml | 返回格式 |  |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

**未编码前的spatialXml：**

<?xml version="1.0" encoding="GB2312"?>

<spatial\_request method="localSearchByLineIntersection">

<searchName>查询关键字名称</searchName>

<searchType>查询类型</searchType>

<pageNum>每页记录数</pageNum>

<batch>页码</batch>

<range>查询范围</range>

<sortRule>排序规则</sortRule>

<spatial\_geos>

<spatial\_geo type="Line">

<coor>线坐标串</coor>

<buffer>多边形缓冲区范围，单位米-</buffer>

</spatial\_geo>

<spatial\_geo type="Line">

<coor>线坐标串</coor>

<buffer>多边形缓冲区范围，单位米-</buffer>

</spatial\_geo>

</spatial\_geos>

</spatial\_request>

* + - * 1. 接口URL示例

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&config=SPAS&enc=utf-8&spatialXml=%3C%3fxml+version%3d%221.0%22+encoding%3d%22GB2312%22%3f%3E%0a%3Cspatial\_request+method%3d%22localSearchByLineIntersection%22%3E%0a+%3CsearchName%3E%E5%8C%BB%E9%99%A2%3C%2fsearchName%3E%0a+%3CsearchType%3E%E5%8C%BB%E9%99%A2%3C%2fsearchType%3E%0a+%3CpageNum%3E10%3C%2fpageNum%3E%0a+%3CpageSum%3E1%3C%2fpageSum%3E%0a+%3Cbatch%3E1%3C%2fbatch%3E%0a+%3Crange%3E500%3C%2frange%3E%0a+%3CsortRule%3E1%3C%2fsortRule%3E%0a+%3Cspatial\_geos%3E%0a++%3Cspatial\_geo+type%3d%22Line%22%3E%0a+++%3Ccoor%3E116.2982023,39.9238786,116.5955507,39.9369959,116.5955507,39.9369959,116.2982023,39.9238786%3C%2fcoor%3E%0a+++%3Cbuffer%3E0%3C%2fbuffer%3E%0a++%3C%2fspatial\_geo%3E%0a++%3Cspatial\_geo+type%3d%22Line%22%3E%0a+++%3Ccoor%3E116.2982023,39.9238786,116.5955507,39.8320577,116.4250962,39.8328293,116.430778,40.0056686%3C%2fcoor%3E%0a+++%3Cbuffer%3E0%3C%2fbuffer%3E%0a++%3C%2fspatial\_geo%3E%0a+%3C%2fspatial\_geos%3E%0a%3C%2fspatial\_request%3E&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - * 1. 返回结果说明

<?xml version="1.0" encoding="GB2312"?>

<spatial\_response type="类型 " servername="服务器名 " versionname=”版本号” >

<searchresult>

<searchtime>查询时间</searchtime>

<count>查询的结果集数</count>

<list type="list">

<poi>

<match>匹配度</match>

<type>数据类型 </type>

<id>数据id</id>

<citycode>城市代码</citycode>

<extid>扩展id</extid>

<pguid>索引id</pguid>

<timestamp>时间戳</timestamp>

<name>名称</name>

<srctype>数据源类型</srctype>

<gridcode>gridcode</gridcode>

<address>地址</address>

<tel>电话</tel>

<icon>图标</icon>

<url>url</url>

<buscode>公交代码</buscode>

<distance>距离查询点距离 </distance>

<direction>距离查询点的方位</direction>

<xml>扩展描述信息</xml>

<newtype>数据类型编码</newtype>

</poi>

</list>

</searchresult>

</spatial\_response>

* + - * 1. 其他需要说明的部分
      1. 根据两条道路交叉点查询
         1. 接口功能介绍

给定两条道路，以这两条路的交叉点为中心点进行周边查询，返回查询结果

* + - * 1. 接口参数说明

**1、通过道路ID查询两条道路的交叉路口**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| ability | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必添参数 |
| config | config=SPAS | 服务参数名称 | 必填参数 |
| spatialXml | spatialXml=查询条件 | spatialXml参数为一个标准的xml格式 | 必填参数 |
| resType | resType=xml | 返回格式 |  |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

**未编码前的spatialXml：**

<?xml version="1.0" encoding="GB2312"?>

<spatial\_request method="searchRoadIntersection">

<cityCode>城市代码</cityCode>

<id1>道路id</id1>

<id2>道路id</id2>

<sortRule>排序规则</sortRule>

</spatial\_request>

**2、通过道路名称查询两条道路的交叉路口**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| ability | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必添参数 |
| config | config=SPAS | 服务参数名称 | 必填参数 |
| spatialXml | spatialXml=查询条件 | spatialXml参数为一个标准的xml格式 | 必填参数 |
| resType | resType=xml | 返回格式 |  |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

**未编码前的spatialXml：**

<?xml version="1.0" encoding="GB2312"?>

<spatial\_request method=" searchRoadIntersectorByName">

<cityCode>城市代码</cityCode>

< roadName1>道路名称</roadName1>

< roadName2>道路名称</roadName2>

<roadLevel>道路等级< /roadLevel>

</spatial\_request>

* + - * 1. 接口URL示例

**1、通过道路ID查询两条道路的交叉路口**

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&enc=utf-8&config=SPAS&spatialXml=%3C%3Fxml+version%3D%221.0%22+encoding%3D%22GB2312%22%3F%3E%0D%0A%3Cspatial\_request+method%3D%22searchRoadIntersection%22%3E%0D%0A+%3CcityCode%3E010%3C%2FcityCode%3E%0D%0A+%3Cid1%3E010J50F0010192159%3C%2Fid1%3E%0D%0A+%3Cid2%3E010J50F0010192563%3C%2Fid2%3E%0D%0A+%3CsortRule%3E0%3C%2FsortRule%3E%0D%0A%3C%2Fspatial\_request%3E&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

**2、通过道路名称查询两条道路的交叉路口**

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&enc=utf-8&config=SPAS&spatialXml=%3C%3Fxml+version%3D%221.0%22+encoding%3D%22GB2312%22%3F%3E%0A%3Cspatial\_request+method%3D%22searchRoadIntersectorByName%22%3E%0A+%3CcityCode%3E010%3C%2FcityCode%3E%0A+%3CroadName1%3E%E9%BC%93%E6%A5%BC%E5%A4%96%E5%A4%A7%E8%A1%97%3C%2FroadName1%3E%0A+%3CroadName2%3E%E5%AE%89%E5%BE%B7%E9%87%8C%E5%8C%97%E8%A1%97%3C%2FroadName2%3E%0A+%3CroadLevel+%2F%3E%0A%3C%2Fspatial\_request%3E&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - * 1. 返回结果说明

**1、通过道路ID查询两条道路的交叉路口**

<?xml version="1.0" encoding="GB2312" ?>

<spatial\_response type="**searchresult**">

<searchresult>

  <searchtime>查询时间</searchtime>

  <count>查询结果集数</count>

<list type="**list**">

<poi>

  <match >匹配度</match>

  <type>数据类型</type>

  <id >poi信息id</id>

  <citycode >城市代码</citycode>

  <extid >扩展id</extid>

  <pguid >索引id</pguid>

  <timestamp >时间戳</timestamp>

  <name >名称</name>

  <srctype >数据源类型</srctype>

  <gridcode >gridcode</gridcode>

  <address >地址</address>

  <tel >电话</tel>

  <icon >图标</icon>

  <url >url</url>

  <buscode>公交代码</buscode>

  <distance>距离查询点距离</distance>

  <direction >距离查询点的方位</direction>

  <xml>扩展信息</xml>

  <en\_type>英文类型</en\_type>

  <en\_name>英文名称</en\_name>

  <en\_address>英文地址</en\_address>

  <x>x坐标 </x>

  <y>y坐标</y>

<newtype>数据类型编码</newtype>

  </poi>

  </list>

 </searchresult>

</spatial\_response>

**2、通过道路名称查询两条道路的交叉路口**

<?xml version="1.0" encoding="GB2312" ?>

<spatial\_response type="**searchresult**">

<searchresult>

  <searchtime>查询时间</searchtime>

  <count>查询结果集数</count>

<list type="**list**">

. <poi>

  <match >匹配度</match>

  <type>数据类型</type>

  <id >poi信息id</id>

  <citycode >城市代码</citycode>

  <extid >扩展id</extid>

  <pguid >索引id</pguid>

  <timestamp >时间戳</timestamp>

  <name >名称</name>

  <srctype >数据源类型</srctype>

  <gridcode >gridcode</gridcode>

  <address >地址</address>

  <tel >电话</tel>

  <icon >图标</icon>

  <url >url</url>

  <buscode>公交代码</buscode>

  <distance>距离查询点距离</distance>

  <direction >距离查询点的方位</direction>

  <xml>扩展信息</xml>

  <en\_type>英文类型</en\_type>

  <en\_name>英文名称</en\_name>

  <en\_address>英文地址</en\_address>

  <x>x坐标 </x>

  <y>y坐标</y>

<newtype>数据类型编码</newtype>

  </poi>

</list>

<bounds>bounds值</bounds>

</searchresult>

</spatial\_response>

* + - * 1. 其他需要说明的部分
    1. PGUID查询
       1. 接口功能介绍

给定全局索引库的ID，根据ID进行POI查询，返回查询结果。

* + - 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| config | config=BESI | 服务参数名称 | 必填参数 |
| ability | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必添参数 |
| cityCode | cityCode=城市代码 | 为限定城市区域查询 | 必填参数 |
| resType | resType=xml | 返回格式 |  |
| pguid | pguid=全局索引ID | 为必填项关键字，不输入将查询无结果。 | 必填参数 |
| srcType | srcType=数据源类型 | 数据源类型，支持混排接口查询，缺省为POI查询。 |  |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

* + - 1. 接口URL示例

[http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&enc=utf-8&config=BESI&resType=xml&cityCode=010&pguid=B000A7YYW9&a\_k=[用户KEY]](http://search1.mapabc.com/sisserver?&config=BESI&resType=xml&cityCode=010&pguid=B000A7YYW9&a_k=%5b用户key%5d)

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - 1. 返回结果说明

<?xml version="1.0" encoding="GBK"?>

<searchresult servername="服务器名" versionname=”版本号”>

<count>此接口返回的结果集数量（默认最大结果集为400，可通过key权限修改，需要联系管理员）</count>

<cache>此数据是否从缓存读取</cache>

<total>引擎返回的结果集总数量</total>

<record>此页返回的结果集数量（最大结果集为20）</record>

<searchtime>查询时间</searchtime>

<bounds>bounds值，当前返回的POI点统一个矩形框内，bounds为矩形框的左上右下坐标对</bounds>

<list type="**list**">

<poi>

<address>地址 </address>

<name>名称</name>

<type>数据类型</type>

<timestamp>时间戳</timestamp>

<match>匹配度</match>

<url >url</url>

<citycode>城市代码</citycode>

<xml >扩展描述信息</xml>

<x>x坐标</x>

<y>y坐标</y>

<code>区域代码</code>

<distance>距离查询点距离</distance>

<drivedistance>导航距离查询点距离</drivedistance>

<pguid>索引id</pguid>

<srctype>数据源类型</srctype>

<gridcode>gridcode代码</gridcode>

<tel>联系电话</tel>

<buscode>公交代码</buscode>

<imageid>图片id</imageid>

<linkid>关联id</linked>

<newtype>数据类型编码</newtype>

</poi>

</list>

<cenpoi>

<x>中心点x坐标 </x>

<y>中心点y坐标</y>

<drivedistance>导航距离查询点距离</drivedistance>

</cenpoi>

</searchresult>

* + - 1. 其他需要说明的部分
    1. 拉框查询
       1. 在基础库和地标库中查询
          1. 接口功能介绍

数据源为基础库和地标库，在一个图形区域内查询关键字.支持Circle（圆）、Ellipse（椭圆）、Rectangle（矩形）、Polygon（多边形）

* + - * 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| ability | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必添参数 |
| config | config=SPAS | 服务参数名称 | 必填参数 |
| spatialXml | spatialXml=查询条件 | spatialXml参数为一个标准的xml格式 | 必填参数 |
| resType | resType=xml | 返回格式 |  |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

1．构造Circle（圆），未编码前的spatialXml

<?xml version="1.0" encoding="GB2312"?>

<spatial\_request method="searchPoiInGeoObject">

<!-- 关键字-->

<searchName>学校</searchName>

<!-- 类型-->

<searchType/>

<!-- 每页记录数-->

<pageNum>10</pageNum>

<!-- 批号-->

<batch>1</batch>

<!-- 数据源类型，缺省为POI类型查询，此参数为扩展参数-- >

<srcType>POI</srcType>

<!--以分号分隔的多组查询条件之间的与或关系, 默认为true，即与关系。此参数为扩展参数-- >

<custom>address:学院路;tel:66882222</custom>

<!--自定义查询参数，默认为true，与custom联合出现 。此参数为扩展参数-- >

<custom\_and>true</custom\_and>

<spatial\_geos>

<!-- type类型有Circle（圆）、Ellipse（椭圆）、Rectangle（矩形），对应的bounds坐标取值分别为：左下和右上坐标坐标集合-->

<spatial\_geo type="Circle">

<bounds>

116.3084014;39.9640344;116.4599165;39.8583246

</bounds>

<!--多边形缓冲区范围，单位米-->

<buffer>0</buffer>

</spatial\_geo>

</spatial\_geos>

<!--自定义排序规则。针对周边查询和包含通用编辑器POI数据来源的查询可在查询时通过此参数进行排序自定义。

赋值含义

0 默认混合排序方式

1 周边查询只按距离排序

2 通用编辑器POI按名称字符串排序

3 通用编辑器POI按字段长度排序

4 通用编辑器POI按饱和度排序

-- >

<sortRule>0</sortRule>

</spatial\_request>

2．构造Polygon（多边形），未编码前的spatialXml

<?xml version="1.0" encoding="GB2312"?>

<spatial\_request method="searchPoiInGeoObject">

<!-- 关键字-->

<searchName>学校</searchName>

<!-- 类型-->

<searchType/>

<!-- 每页记录数-->

<pageNum>10</pageNum>

<!-- 批号-->

<batch>1</batch>

<spatial\_geos>

<spatial\_geo type="Polygon">

<coor>

LQOHKOTQONNKIIH,LQOHKTTRXNHKMMH,LQOHKTPUOPJKIIL,

LQOHKRQUUPPKIMH

</coor>

<!--多边形缓冲区范围，单位米-->

<buffer>0</buffer>

</spatial\_geo>

</spatial\_geos>

<!--自定义排序规则。针对周边查询和包含通用编辑器POI数据来源的查询可在查询时通过此参数进行排序自定义。

赋值含义

0 默认混合排序方式

1 周边查询只按距离排序

2 通用编辑器POI按名称字符串排序

3 通用编辑器POI按字段长度排序

4 通用编辑器POI按饱和度排序

-- >

<sortRule>0</sortRule>

</spatial\_request>

* + - * 1. 接口URL示例

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&resType=xml&config=SPAS&enc=utf-8&spatialXml=%3c%3fxml+version%3d%221.0%22+encoding%3d%22GB2312%22%3f%3e%0a%3cspatial\_request+method%3d%22searchPoiInGeoObject%22%3e%0a+%3csearchName%3e%e5%ad%a6%e6%a0%a1%3c%2fsearchName%3e%0a+%3csearchType+%2f%3e%0a+%3cpageNum%3e10%3c%2fpageNum%3e%0a+%3cpageSum%3e1%3c%2fpageSum%3e%0a+%3cbatch%3e1%3c%2fbatch%3e%0a+%3cspatial\_geos%3e%0a++%3cspatial\_geo+type%3d%22Circle%22%3e%0a+++%3cbounds%3eIONMLHXSMIRNMHI%3bKWFXOLPRQLFIHM%3bIONMMMYXNNSJELM%3bKWFWNPSQQNFILE%3b%3c%2fbounds%3e%0a+++%3cbuffer%3e0%3c%2fbuffer%3e%0a++%3c%2fspatial\_geo%3e%0a+%3c%2fspatial\_geos%3e%0a+%3csortRule%3e10%3c%2fsortRule%3e%0a%3c%2fspatial\_request%3e&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - * 1. 返回结果说明

<?xml version="1.0" encoding="GBK"?>

<searchresult servername="服务器名" versionname=”版本号”>

<count>此接口返回的结果集数量（默认最大结果集为400，可通过key权限修改，需要联系管理员）</count>

<cache>此数据是否从缓存读取</cache>

<total>引擎返回的结果集总数量</total>

<record>此页返回的结果集数量（最大结果集为20）</record>

<searchtime>查询时间</searchtime>

<bounds>bounds值，当前返回的POI点统一个矩形框内，bounds为矩形框的左上右下坐标对</bounds>

<list type="**list**">

<poi>

<address>地址 </address>

<name>名称</name>

<id>poi数据id</id>

<type>数据类型</type>

<timestamp>时间戳</timestamp>

<match>匹配度</match>

<url >url</url>

<citycode>城市代码</citycode>

<xml >扩展描述信息</xml>

<x>x坐标</x>

<y>y坐标</y>

<code>区域代码</code>

<distance>距离查询点距离</distance>

<drivedistance>导航距离查询点距离</drivedistance>

<pguid>索引id</pguid>

<srctype>数据源类型</srctype>

<gridcode>gridcode代码</gridcode>

<tel>联系电话</tel>

<buscode>公交代码</buscode>

<icon>图标</icon>

<direction>距离查询点的方位</direction>

<extid>外部扩展id</extid>

<en\_type>英文类型</en\_type>

<en\_name>英文名称</en\_name>

<en\_address>英文地址</en\_address>

<newtype>数据类型编码</newtype>

</poi>

</list>

</searchresult>

* + - * 1. 其他需要说明的部分
      1. 在编辑器库中查询
         1. 接口功能介绍

数据源为编辑器库，在一个图形区域内查询关键字.支持Circle（圆）、Ellipse（椭圆）、Rectangle（矩形）、Polygon（多边形）

* + - * 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| ability | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必添参数 |
| config | config=USPAS | 服务参数名称 | 必填参数 |
| spatialXml | spatialXml=查询条件 | spatialXml参数为一个标准的xml格式 | 必填参数 |
| resType | resType=xml | 返回格式 |  |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

1．构造Circle（圆），未编码前的spatialXml

<?xml version="1.0" encoding="GB2312"?>

<spatial\_request method="searchPoiInGeoObject">

<!-- 关键字-->

<searchName>学校</searchName>

<!-- 类型-->

<searchType/>

<!-- 每页记录数-->

<pageNum>10</pageNum>

<!-- 批号-->

<batch>1</batch>

<!-- 数据源类型，缺省为POI类型查询，此参数为扩展参数-- >

<srcType>POI</srcType>

<!--以分号分隔的多组查询条件之间的与或关系, 默认为true，即与关系。此参数为扩展参数-- >

<custom>address:学院路;tel:66882222</custom>

<!--自定义查询参数，默认为true，与custom联合出现 。此参数为扩展参数-- >

<custom\_and>true</custom\_and>

<spatial\_geos>

<!-- type类型有Circle（圆）、Ellipse（椭圆）、Rectangle（矩形），对应的bounds坐标取值分别为：左下和右上坐标坐标集合-->

<spatial\_geo type="Circle">

<bounds>

IONMLHXSMIRNMHI;KWFXOLPRQLFIHM;IONMMMYXNNSJELM;K WFWNPSQQNFILE

</bounds>

<!--多边形缓冲区范围，单位米-->

<buffer>0</buffer>

</spatial\_geo>

</spatial\_geos>

<!--自定义排序规则。针对周边查询和包含通用编辑器POI数据来源的查询可在查询时通过此参数进行排序自定义。

赋值含义

0 默认混合排序方式

1 周边查询只按距离排序

2 通用编辑器POI按名称字符串排序

3 通用编辑器POI按字段长度排序

4 通用编辑器POI按饱和度排序

-- >

<sortRule>0</sortRule>

</spatial\_request>

2．构造Polygon（多边形），未编码前的spatialXm

<?xml version="1.0" encoding="GB2312"?>

<spatial\_request method="searchPoiInGeoObject">

<!-- 关键字-->

<searchName>学校</searchName>

<!-- 类型-->

<searchType/>

<!-- 每页记录数-->

<pageNum>10</pageNum>

<!-- 批号-->

<batch>1</batch>

<spatial\_geos>

<spatial\_geo type="Polygon">

<coor>

LQOHKOTQONNKIIH,LQOHKTTRXNHKMMH,LQOHKTPUOPJKIIL,

LQOHKRQUUPPKIMH

</coor>

<!--多边形缓冲区范围，单位米-->

<buffer>0</buffer>

</spatial\_geo>

</spatial\_geos>

<!--自定义排序规则。针对周边查询和包含通用编辑器POI数据来源的查询可在查询时通过此参数进行排序自定义。

赋值含义

0 默认混合排序方式

1 周边查询只按距离排序

2 通用编辑器POI按名称字符串排序

3 通用编辑器POI按字段长度排序

4 通用编辑器POI按饱和度排序

-- >

<sortRule>0</sortRule>

</spatial\_request>

* + - * 1. 接口URL示例

[http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&enc=utf-8&config=USPAS&resType=xml&spatialXml=%3C%3Fxml+version%3D%221.0%22+encoding%3D%22GB2312%22%3F%3E%0D%0A%3Cspatial\_request+method%3D%22searchPoiInGeoObject%22%3E%0D%0A+%3CsearchName%3E%E6%9C%9D%E9%98%B3%3C%2FsearchName%3E%0D%0A+%3CsearchType+%2F%3E%0D%0A+%3CpageNum%3E10%3C%2FpageNum%3E%0D%0A+%3Cbatch%3E1%3C%2Fbatch%3E%0D%0A+%3Cspatial\_geos%3E%0D%0A++%3Cspatial\_geo+type%3D%22Rectangle%22%3E%0D%0A+++%3Cbounds%3EOLUIRNSNHJG%3BQTMTYQWNLJK%3BJJPMKLUMMOH%3BLRHWOJRIMGL%3C%2Fbounds%3E%0D%0A+++%3Cbuffer%3E0%3C%2Fbuffer%3E%0D%0A++%3C%2Fspatial\_geo%3E%0D%0A+%3C%2Fspatial\_geos%3E%0D%0A%3C%2Fspatial\_request%3E&a\_k=](%20http://221.180.145.25:9083/omp/gatesrv?omp_appid=000702000000000096&omp_restype=autonavi_Gis&extparam=&ability=apiserver&abilityuri=sisserver&enc=utf-8&config=USPAS&resType=xml&enc=utf-8&spatialXml=%3C%3Fxml+version%3D%221.0%22+encoding%3D%22GB2312%22%3F%3E%0D%0A%3Cspatial_request+method%3D%22searchPoiInGeoObject%22%3E%0D%0A+%3CsearchName%3E%E6%9C%9D%E9%98%B3%3C%2FsearchName%3E%0D%0A+%3CsearchType+%2F%3E%0D%0A+%3CpageNum%3E10%3C%2FpageNum%3E%0D%0A+%3Cbatch%3E1%3C%2Fbatch%3E%0D%0A+%3Cspatial_geos%3E%0D%0A++%3Cspatial_geo+type%3D%22Rectangle%22%3E%0D%0A+++%3Cbounds%3EOLUIRNSNHJG%3BQTMTYQWNLJK%3BJJPMKLUMMOH%3BLRHWOJRIMGL%3C%2Fbounds%3E%0D%0A+++%3Cbuffer%3E0%3C%2Fbuffer%3E%0D%0A++%3C%2Fspatial_geo%3E%0D%0A+%3C%2Fspatial_geos%3E%0D%0A%3C%2Fspatial_request%3E&enc=utf-8&a_k=)[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - * 1. 返回结果说明

<?xml version="1.0" encoding="GBK"?>

<searchresult servername="服务器名" versionname=”版本号”>

<count>此接口返回的结果集数量（默认最大结果集为400，可通过key权限修改，需要联系管理员）</count>

<cache>此数据是否从缓存读取</cache>

<total>引擎返回的结果集总数量</total>

<record>此页返回的结果集数量（最大结果集为20）</record>

<searchtime>查询时间</searchtime>

<bounds>bounds值，当前返回的POI点统一个矩形框内，bounds为矩形框的左上右下坐标对</bounds>

<list type="**list**">

<poi>

<address>地址 </address>

<name>名称</name>

<id>poi数据id</id>

<type>数据类型</type>

<timestamp>时间戳</timestamp>

<match>匹配度</match>

<url >url</url>

<citycode>城市代码</citycode>

<xml >扩展描述信息</xml>

<x>x坐标</x>

<y>y坐标</y>

<code>区域代码</code>

<distance>距离查询点距离</distance>

<drivedistance>导航距离查询点距离</drivedistance>

<pguid>索引id</pguid>

<srctype>数据源类型</srctype>

<gridcode>gridcode代码</gridcode>

<tel>联系电话</tel>

<buscode>公交代码</buscode>

<icon>图标</icon>

<direction>距离查询点的方位</direction>

<extid>外部扩展id</extid>

<en\_type>英文类型</en\_type>

<en\_name>英文名称</en\_name>

<en\_address>英文地址</en\_address>

<newtype>数据类型编码</newtype>

</poi>

</list>

</searchresult>

* + - * 1. 其他需要说明部分
    1. 缓冲区查询
       1. 接口功能介绍

道路沿线缓冲区查询。

* + - 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| ability | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必添参数 |
| config | config=SPAS | 服务参数名称 | 必填参数 |
| spatialXml | spatialXml=查询条件 | spatialXml参数为一个标准的xml格式 | 必填参数 |
| resType | resType=xml | 返回格式 |  |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

**1、道路沿线缓冲区查询**

**未编码前的spatialXml：**

<?xml version="1.0" encoding="GB2312"?>

<spatial\_request method="searchPoiByRoadId">

<!--城市号-->

<cityCode>010</cityCode>

<!-- 道路id-->

<id>1105</id>

<!-- 关键字名称-->

<searchName>医院</searchName>

<!-- 关键字类型-->

<searchType>医院</searchType>

<!-- 道路沿线缓冲范围-->

<range>300</range>

<!--自定义排序规则。针对周边查询和包含通用编辑器POI数据来源的查询可在查询时通过此参数进行排序自定义。

赋值含义

0 默认混合排序方式

1 周边查询只按距离排序

2 通用编辑器POI按名称字符串排序

3 通用编辑器POI按字段长度排序

4 通用编辑器POI按饱和度排序

-- >

<sortRule>1</sortRule>

</spatial\_request>

* + - 1. 接口URL示例

**1、道路沿线缓冲区查询**

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&enc=utf-8&config=SPAS&spatialXml=%3C%3fxml+version%3d%221.0%22+encoding%3d%22GB2312%22%3f%3E%0a%3Cspatial\_request+method%3d%22searchPoiByRoadId%22%3E%0a+%3CcityCode%3E010%3C%2fcityCode%3E%0a+%3Cid%3E001010730%3C%2fid%3E%0a+%3CsearchName%3E%E5%8C%BB%E9%99%A2%3C%2fsearchName%3E%0a+%3CsearchType%3E%E5%8C%BB%E9%99%A2%3C%2fsearchType%3E%0a+%3Crange%3E1000%3C%2frange%3E%0a+%3CsortRule%3E1%3C%2fsortRule%3E%0a%3C%2fspatial\_request%3E&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - 1. 返回结果说明

**1、道路沿线缓冲区查询**

<?xml version="1.0" encoding="GBK"?>

<searchresult servername="服务器名" versionname=”版本号”>

<count>此接口返回的结果集数量（默认最大结果集为400，可通过key权限修改，需要联系管理员）</count>

<cache>此数据是否从缓存读取</cache>

<total>引擎返回的结果集总数量</total>

<record>此页返回的结果集数量（最大结果集为20）</record>

<searchtime>查询时间</searchtime>

<bounds>bounds值，当前返回的POI点统一个矩形框内，bounds为矩形框的左上右下坐标对</bounds>

<list type="**list**">

<poi>

<address>地址 </address>

<name>名称</name>

<id>poi数据id</id>

<type>数据类型</type>

<timestamp>时间戳</timestamp>

<match>匹配度</match>

<url >url</url>

<citycode>城市代码</citycode>

<xml >扩展描述信息</xml>

<x>x坐标</x>

<y>y坐标</y>

<code>区域代码</code>

<distance>距离查询点距离</distance>

<drivedistance>导航距离查询点距离</drivedistance>

<pguid>索引id</pguid>

<srctype>数据源类型</srctype>

<gridcode>gridcode代码</gridcode>

<tel>联系电话</tel>

<buscode>公交代码</buscode>

<icon>图标</icon>

<direction>距离查询点的方位</direction>

<extid>外部扩展id</extid>

<en\_type>英文类型</en\_type>

<en\_name>英文名称</en\_name>

<en\_address>英文地址</en\_address>

<newtype>数据类型编码</newtype>

</poi>

</list>

</searchresult>

* + - 1. 其他需要说明的部分
    1. 关键字提示接口服务请求
       1. 接口功能介绍

用户进行搜索时，能够对用户的输入做出动态的提示，减轻用户输入负担

* + - 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| ability | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必添参数 |
| config | config=TIP | 服务参数名称 | 必填参数 |
| searchName | searchName=查询名称 |  | 必填参数 |
| cityCode | cityCode=城市代码 |  | 必填参数 |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

* + - 1. 接口URL示例

[http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&enc=utf-8&config=TIP&cityCode=010&searchName=](%20http://221.180.145.25:9083/omp/gatesrv?omp_appid=000702000000000096&omp_restype=autonavi_Gis&extparam=&ability=apiserver&abilityuri=sisserver&enc=utf-8&config=TIP&cityCode=010&searchName=)西单&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - 1. 返回结果说明

西单,西单商场,西单北大街,西单停车场,西单路口南,西单大悦城,西单市场,西单王府井,西单广场,西单商场宏图三胞

* + - 1. 其他需要说明的部分
    1. 自驾沿线搜索
       1. 接口功能介绍

沿自驾线路搜索周边兴趣点。

* + - 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数格式 | 参数含义 | 备注 |
| sid | sid=1100 | 服务编码 | 必填参数 |
| keyword | keyword=关键字 | 搜索关键字 | 必填参数 |
| asid | asid=自驾请求的rid | 沿线搜索ID | 必填参数 |
| xys | x1=起点x坐标  y1=起点y坐标  x2=终点x坐标  y2=终点y坐标 | 坐标对集合，支持13个途经点，多个途经点使用";"隔开，坐标点内部使用","隔开，格式如下：x1,y1;x2,y2;… | 必填参数 |
| ability | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必添参数 |
| rid |  | 请求ID | 可选参数 |
| number | number=每页记录数 | 每页记录数，默认为每页显示10条记录 | 可选参数 |
| batch |  | 当前显示页，默认为第一页 | 可选参数 |
| resType | resType=xml/json | 返回结果的格式，支持XML与JSON，不区分大小写，默认为XML格式 | 可选参数 |
| routeType | routeType=路径计算规则 | 路径计算规则，默认值为0  routeType=0速度优先（时间,拐点最少）  routeType=1费用优先（尽量避开收费路段）  routeType=2距离优先  routeType=5不走快速路（快速路包含高速路）  routeType=6国道优先  routeType=7省道优先  routeType=9多策略（同时使用速度优先、费用优先、距离优先三个策略计算路径） | 可选参数 |
| range | range=查询范围 | 周边查询时，圆型区域的半径，单位：米，默认值为3000 | 可选参数 |
| avoidName | avoidName=避让名称 | 避让名称，如一条道路的名称 | 可选参数 |
| avoidRegion | avoidRegion=区域避让 | 区域避让，支持32个避让区域，每个区域最多可有16个顶点，避让区域之间使用":"区分，避让区域由坐标对组成，坐标对之间使用";"分隔，如果是四边形则有四个坐标点，如果是五边形则有五个坐标点 |  |
| encode | encode=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| key | Key=[用户key] | 用户key | 必填参数 |

* + - 1. 接口URL示例

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=gss/simple&encode=utf-8&number=20&batch=1&range=100&resType=xml&sid=1100&keyword=%E8%B6%85%E5%B8%82&routeType=0&avoidName=%E5%AD%A6%E9%99%A2%E8%B7%AF&avoidRegion=&pathNum=1&xys=123.45471,41.723335;123.436457,41.81823&rid=982358&key=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - 1. 返回结果说明

<searchresult version="v2.0.0">

<status>E0</status>

<time>0.283</time> <!--查询时间，单位：毫秒-->

<resultList type="list"> <!—根据路线总数返回的结果-->

<result><!—第一条线路返回的结果-->

<count>69</count>

<total>69</total> <!—第一条路径返回的记录集总数-->

<record>2</record><!--此页返回的记录数-->

<bounds>123.436605,41.755449;123.43695,41.784928</bounds>

<poi type="list">

<poi><!—POI返回信息-->

<name>皇朝万豪酒店</name><!--POI的名称-->

<type>住宿服务;宾馆酒店;五星级宾馆</type><!--数据分类类别-->

<tel>024-23883456</tel><!--联系电话-->

<address>青年大街388号</address><!—地址-->

<x>123.436950</x><!--经度-->

<y>41.755449</y>!—纬度-->

<srctype>basepoi</srctype><!--数据源-->

</poi>

<poi>...</poi>

</poi>

</result>

<result>...</result><!—第二条线路返回的结果-->

<result>...</result><!—第三条线路返回的结果-->

</resultList>

</searchresult>

* + - 1. 其他需要说明的部分
  1. 导航服务接口
     1. 路径导航服务请求接口
        1. 接口功能介绍

驾车路径导航，用户可以选择路线搜索条件。

* + - 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| ability | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | abilityuri=route/simple | 请求类型 | 必添参数 |
| sid | sid=8000 | 服务编码 | 必填参数 |
| rid | rid=0 | 请求ID | 必填参数 |
| routeType |  | 路径计算规则，默认值为0  routeType=0速度优先（时间,拐点最少）  routeType=1费用优先（尽量避开收费路段）  routeType=2距离优先  routeType=5不走快速路（快速路包含高速路）  routeType=6国道优先  routeType=7省道优先  routeType=9多策略（同时使用速度优先、费用优先、距离优先三个策略计算路径） | 可选参数 |
| xys |  | 坐标对集合，支持13个途经点，多个途经点使用";"隔开，坐标点内部使用","隔开，格式如下：  x1,y1;x2,y2;… | 必填参数 |
| avoidRegion |  | 区域避让，支持32个避让区域，每个区域最多可有16个顶点  避让区域之间使用":"区分，避让区域由坐标对组成，坐标对之间使用";"分隔，如果是四边形则有四个坐标点，如果是五边形则有五个坐标点 | 可选参数 |
| avoidName |  | 避让名称  如一条道路的名称 | 可选参数 |
| resType | resType=xml/json | 返回结果的格式，支持XML与JSON，不区分大小写，默认为XML格式 | 可选参数 |
| pathNum | pathNum=1/2/3 | 返回方案结果数量,最多三条,默认1条 | 可选参数 |
| signInfo | signInfo=1/0 | 返回自驾途中路牌信息 | 可选参数 |
| landmark | landmark=n | 地标参数，需要返回地标的时候使用，n为前台要求返回的地标个数，如果引擎计算得到的沿路地标个数大于n，则返回n个地标给前台，如果小于n，则返回实际计算得到的地标。最多返回10个地标信息。 | 可选参数 |
| group | group=0/1 | 路径聚合功能: group=返回线路是否带有聚合线路，默认为group=0, 返回的线路规划方案中各路段不进行合并；  group=1, 返回的线路规划方案中，按照规则将城市内路段和城市间路段进行合并。 | 可选参数 |
| per | per=n | 抽希函数，在数据做抽希的时候应用，per值可以为空，默认值为50，如果小于50取实际坐标点数，如果大于50取50个点 | 可选参数 |
| encode | encode=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| key |  | 用户key | 必填参数 |

* + - 1. 接口URL示例

**请求示例:**

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=route/simple&encode=utf-8&resType=xml&routeType=0&avoidName=%E5%AD%A6%E9%99%A2%E8%B7%AF&avoidRegion=&pathNum=1&sid=8000&xys=123.45471,41.723335;123.436457,41.81823&landmark=7&group=1&rid=823096&key=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - 1. 返回结果说明

<searchresult version="v2.0.0">

<status>E0</status>

<time>0.106</time> <!--搜索时间，单位：毫秒-->

<asrid>966646</asrid><!—沿路搜索id标志-->

<pathList type="list"><!—返回方案列表-->

<path><!—第一条方案-->

<count>24</count><!—当前方案路段数量-->

<bounds>123.41,41.72;123.46,41.87</bounds><!—导航起始范围-->

<coors>

123.4548,41.72338;123.4562,41.72128;123.4564,41.72133;123.4515,41.72917;123.4499,41.73306;123.4483,41.73383;123.4499,41.73597</coors> <!--区域范围坐标串-->

<list type="list"> <!—当前方案路段导航信息列表-->

<segment> <!--路段导航信息-->

<roadName>新隆街</roadName><!—道路名称-->

<direction>东南</direction><!—行驶方向-->

<roadLength>264米</roadLength><!—行驶距离-->

<action>左转调头</action><!—辅助动作-->

<accessorialInfo/>进入主路<accessorialInfo/><!—动作-->

<driveTime>1分钟</driveTime><!—行驶时间-->

<grade>次要道路</grade><!--道路等级,参见下表-->

<form>辅路</form><!—道路性质描述-->

<signInfo>南京街/沈阳站</signInfo><!—路牌信息提示-->

<cameraInfo>监控摄像头,0,123.456215,41.721325;</cameraInfo><!—途径摄像头信息-->

<textInfo>沿新隆街向东南行驶264米左转调头</textInfo><!—文字描述-->

<landmarkList type="list"><!—返回地标列表-->

<landmark><!—地标信息-->

<name>百银大厦</name><!—地标名称-->

<tel/><!—地标电话-->

<address>友好街18</address><!—地标街道位置-->

<x>123.4411786111</x><!—地标经度-->

<y>41.8117016667</y><!—地标纬度-->

<type>商务住宅;楼宇;商务写字楼</type><!—地标类型-->

<direction>South</direction><!—地标相对拐点方向-->

<distance>168.683</distance><!—地标相对拐点距离-->

</landmark><!—地标信息-->

</landmarkList> <!—返回地标列表-->

<coor>

123.45484,41.72338;123.45503,41.723095;123.45537,41.72253;123.45571,41.72196;123.455894,41.72187;123.45624,41.72128

</coor><!—行驶路段坐标-->

</segment>

<segment><!—路段描述-->

<roadName>浑河大街,青年大街</roadName>

<direction/>

<roadLength>7.837千米</roadLength>

<action>向右前方行驶</action>

<accessorialInfo>进入右转专用道</accessorialInfo>

<driveTime>10分钟</driveTime>

<grade>主要道路</grade>

<form>主路</form>

<signInfo/>

<cameraInfo/>

<textInfo>沿浑河大街/青年大街行驶7.837千米向右前方行驶进入右转专用道</textInfo>

<coor>

123.45699,41.73815;123.456795,41.738216;123.45667,41.73826;123.45655,41.73833;123.456474,41.73838;123.456345,41.73849;

</coor>

<detailList type="list"><!—详细路段列表-->

<detailSegment><!—路段细节描述-->

<roadName>浑河大街</roadName>

<direction/>

<roadLength>2.361千米</roadLength>

<action>直行</action>

<accessorialInfo/>

<driveTime>3分钟</driveTime>

<grade>主要道路</grade>

<form>主路</form>

<signInfo/>

<cameraInfo>

测速摄像头,60,123.45022,41.74501;监控摄像头,0,123.44302,41.750687;

</cameraInfo>

<textInfo>沿浑河大街行驶2.361184千米</textInfo>

<coor>

123.45699,41.73815;123.456795,41.738216;123.45667,41.73826;;

</coor>

</detailSegment>

<detailSegment>

<roadName>青年大街</roadName>

<direction/>

<roadLength>5.476千米</roadLength>

<action>向右前方行驶</action>

<accessorialInfo>进入右转专用道</accessorialInfo>

<driveTime>7分钟</driveTime>

<grade>主要道路</grade>

<form>主路</form>

<signInfo/>

<cameraInfo>

测速摄像头,60,123.45022,41.74501;监控摄像头,0,123.44302,41.750687; </cameraInfo>

<textInfo>沿青年大街行驶5.476千米向右前方行驶进入右转专用道</textInfo>

<coor>

123.43889,41.754177;123.438614,41.75454;123.438614,41.75454;

</coor>

</detailSegment>

</detailList>

</segment>

</list>

</path>

<path>...</path>

<path>...</path>

</pathList>

</searchresult>

* + - 1. 其他需要说明的部分

**子节点描述：**

新增cameraInfo节点:

用四元组标识,格式为 摄像头类型,限速大小,x坐标,y坐标;摄像头2类型,,,;

新增 signInfo节点:(路牌提示)

-------------------------------------------------------------------------------

form节点：

|  |
| --- |
| Form |
| 左右转专用道 |
| 主路 |
| 复杂节点内部道路 |
| 高架 |
| 环岛 |
| 辅助道路 |
| 匝道 |
| 辅路 |
| 匝道 |
| 出口 |
| 入口 |
| 右转专用道 |
| 右转专用道 |
| 左转专用道 |
| 左转专用道 |
| 普通道路 |
| 左右转专用道 |

grade节点：

|  |
| --- |
| Grade |
| 高速公路 |
| 国道 |
| 省道 |
| 县道 |
| 乡公路 |
| 县乡村内部道路 |
| 主要大街、城市快速路 |
| 主要道路 |
| 次要道路 |
| 普通道路 |
| 非导航道路 |

action 节点：

|  |
| --- |
| Action |
| 左转 |
| 右转 |
| 偏左转 |
| 偏右转 |
| 左后转 |
| 右后转 |
| 左转调头 |
| 直行 |
| 靠左 |
| 靠右 |
| 进入环岛 |
| 减速行驶 |

accessoriaInfo节点:

|  |
| --- |
| accessorialInfo |
| 进入主路 |
| 进入辅路 |
| 进入高速 |
| 进入匝道 |
| 进入隧道 |
| 进入中间岔道 |
| 进入右岔路 |
| 进入左岔路 |
| 进入轮渡 |
| 驶离轮渡 |
| 进入右转专用道 |
| 进入左转专用道 |
| 到达出口 |
| 到达服务区 |
| 到达收费站 |
| 到达途径地 |
| 到达目的地 |
| 走第1个出口 |
| 走第2个出口 |
| 走第3个出口 |
| 走第4个出口 |
| 走第5个出口 |
| 走第6个出口 |
| 走第7个出口 |
| 到达途经点，走第1出口 |
| 到达途经点，走第2出口 |
| 到达途经点，走第3出口 |
| 到达途经点，走第4出口 |
| 到达途经点，走第5出口 |
| 到达途经点，走第6出口 |
| 到达途经点，走第7出口 |
| 到达途经点或目的地 |
| 复杂路口右1 |
| 复杂路口右2 |
| 复杂路口右3 |
| 复杂路口右4 |
| 复杂路口右5 |
| 复杂路口左1 |
| 复杂路口左2 |
| 复杂路口左3 |
| 复杂路口左4 |
| 复杂路口左5 |

* + 1. 导航途经点信息请求接口
       1. 接口功能介绍

提示自驾线路周边的出口、收费站、加油站、服务区、终点停车场信息。

* + - 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| ability | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必添参数 |
| sid | sid=1101 | 服务编码 | 必填参数 |
| keyword | keyword=关键字 | 搜索关键字 | 必填参数 |
| asid | asid=自驾请求的rid | 沿线搜索ID | 必填参数 |
| xys | x1=起点x坐标  y1=起点y坐标  x2=终点x坐标  y2=终点y坐标 | 坐标对集合，支持13个途经点，多个途经点使用";"隔开，坐标点内部使用","隔开，格式如下：x1,y1;x2,y2;… | 必填参数 |
| rid |  | 请求ID | 可选参数 |
| number | number=每页记录数 | 每页记录数，默认为每页显示10条记录 | 可选参数 |
| batch |  | 当前显示页，默认为第一页 | 可选参数 |
| resType | resType=xml/json | 返回结果的格式，支持XML与JSON，不区分大小写，默认为XML格式 | 可选参数 |
| routeType | routeType=路径计算规则 | 路径计算规则，默认值为0  routeType=0速度优先（时间,拐点最少）  routeType=1费用优先（尽量避开收费路段）  routeType=2距离优先  routeType=5不走快速路（快速路包含高速路）  routeType=6国道优先  routeType=7省道优先  routeType=9多策略（同时使用速度优先、费用优先、距离优先三个策略计算路径） | 可选参数 |
| avoidName | avoidName=避让名称 | 避让名称，如一条道路的名称 | 可选参数 |
| avoidRegion | avoidRegion=区域避让 | 区域避让，支持32个避让区域，每个区域最多可有16个顶点，避让区域之间使用":"区分，避让区域由坐标对组成，坐标对之间使用";"分隔，如果是四边形则有四个坐标点，如果是五边形则有五个坐标点 |  |
| pathNum | pathNum=路径返回条数 | 路径返回条数，默认为3，pathNum=3，返回3条路径；  pathNum=2，返回2条路径；  pathNum=1，返回1条路径。 | 可选参数 |
| encode | encode=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| key |  | 用户key | 必填参数 |

* + - 1. 接口URL示例

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=gss/simple&encode=utf-8&number=100&batch=1&range=100&resType=xml&sid=1101&keyword=%E9%81%93%E8%B7%AF%E4%BF%A1%E6%81%AF&asid=123 &routeType=0&avoidName=%E5%AD%A6%E9%99%A2%E8%B7%AF&avoidRegion=&pathNum=1&xys=123.45471,41.723335;125.324099,43.909906&rid=53184&key=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - 1. 返回结果说明

<searchresult version="v2.0.0">

<status>E1</status>

<time>0.651</time>

<resultList type="list">

<result>

<count>0</count>

<total>0</total> <!-- 出口，收费站总数 -->

<record>0</record> <!-- 出口，收费站实际返回数 -->

<bounds/>

<poi type="list"/> <!-- 出口，收费站信息列表 -->

<poi>

<name>蒲河收费站</name>

<type>道路附属设施;收费站;收费站</type>

...

</poi>

<poi>

<name>梨树出口</name>

<type>地名地址信息;交通地名;高速路出口</type>

...

</poi>

…

</poi>

<gascount>4</gascount>

<gastotal>4</gastotal> <!-- 加油站、服务区总数 -->

<gasrecord>4</gasrecord><!-- 加油站、服务区实际返回数 -->

<gasstation type="list"><!-- 加油站、服务区列表 -->

<poi>

<name>长城加油站</name>

<type>汽车服务;加油站;加油站</type>

…

</poi>

<poi>

<name>昌图服务区</name>

<type>道路附属设施;服务区;高速服务区</type>

…

</poi>

…

</gasstation>

<parkrecord>100</parkrecord> <!--终点停车场总数-->

<actualrecord>10</actualrecord> <!--终点停车场实际返回数-->

<Carpark type="list"> <!--终点停车场列表-->

<poi>

<name>停车场</name>

<type>交通设施服务;停车场;公共停车场</type>

…

</poi>

…

</Carpark></result>

<result>...</result><!—驾车方案2-->

<result>...</result>

</resultList>

</searchresult>

* + - 1. 其他需要说明的部分
  1. 公交查询接口
     1. 公交路径导航请求接口
        1. 接口功能

根据起点终点查询公交路径导航。

* + - 1. 接口参数说明

用户向服务器发起http请求的接口参数说明见表 2‑11。

表 2‑11 新公交路径导航接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |

|  |  |  |  |
| --- | --- | --- | --- |
| sid | sid=8001 | 服务编码 | 必填参数 |
| ability | ability=apiserver | 服务类型 | 必填参数 |
| abilityuri | abilityuri=bus/simple | 请求类型 | 必填参数 |
| encode | encode=utf-8 | 请求串编码 | 必填参数 |
| xys | x1=起点x坐标  y1=起点y坐标  x2=终点x坐标  y2=终点y坐标 | 途经点坐标集合,用分号”;”分隔 | 必填参数 |
| rid |  | 请求ID | 可选参数 |
| resType | resType=xml/json | 返回结果的格式，支持XML与JSON，不区分大小写，默认为XML格式 | 可选参数 |
| type | type =路径计算规则 | 路径计算规则：  type =0，最快捷模式  type =1，最经济模式  type =2，最少换乘模式  type =3，最少步行模式  type =4，最舒适模式  type=10，地铁优先模式  type=11，只乘公交模式 | 必选参数 |
| night | night=0，1 | night=0，没有夜班车  night=1，有夜班车 | 可选参数 |
| distance | distance=X,Y,Z  距离限制参数 | X，Y，Z为距离限制参数，其中X为换乘最大步行距离，Y为起点最大步行距离，Z为终点最大步行距离。X、Y、Z中间必须有“，”隔开，若没有，默认为只查X。 | 可选参数 |
| lineName | lineName=ABCD，X,Y,Z,W  A为避开线路  B为必选线路  C为出发必乘  D为到达必乘 | A,B,C,D必须为0或1,1表示该选项成立，0则表示不成立。ABCD后面为所选方式所要的线路名称，有几项成立则有几个线路名，比如1000,85代表的意思是避开85路 | 可选参数 |
| via | via=0,1途经点标志位 | via=0，默认为0，表示不带途经点，返回格式按照原格式返回。  via=1，表示带途经点，返回格式按照新格式返回 | 可选参数 |
| number | number=返回规划导航的个数，默认10 | number=1，指定本次线路规划方案条数最小值；  number=10，指定本次线路规划方案条数最大值，如果实际规划出的方案小于此值，则按实际方案数返回。 | 可选参数 |
| group | group=返回线路是否带有聚合线路，默认为0 | group=0, 返回结果不包含聚合线路；  group=1, 返回的线路规划方案中，如果有和主线路起终点相同的备选线路，将返回备选线路方案。 | 可选参数 |
| isGateWayTip | isGateWayTip=判断是否返回地铁输入口提示，默认0 | isGateWayTip=1，返回线路方案中，如果有地铁输入口提示，则返回地铁输入口提示；  isGateWayTip=0,返回线路规划方案中不包含地铁输入口提示。 | 可选参数 |
| isWalk | isWalk=返回结果是否包含步行线路，默认0 | isWalk =1，返回线路方案中，如果有步行线路，则返回步行线路；  isWalk =0，返回线路规划方案中不包含步行线路。 | 可选参数 |
| landmark | landmark =返回结果是否包含地标，以及返回地标个数，默认0 | landmark =0，返回线路方案中不包含地标；  landmark >0，返回线路规划方案中包含地标，且每个拐点返回的地标个数为landmark。支持最多返回10个地标。 | 可选参数 |
| landmarkRange | landmarkRange=地标查询范围 | 地标查询时，圆型区域的半径，单位：米，默认值为500米 | 可选参数 |
| key |  | 用户key | 必填参数 |

* + - 1. 接口URL示例

**示例一：**

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=bus/simple&encode=utf-8&resType=xml&per=150&number=10&night=0&maxtrans=4&type=0&sid=8001&ditance=1500,1500&lineName=0000,85&xys=123.454712,41.723336;123.403797,41.832525&city=沈阳&rid=589375 &key=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

**示例二：group=1（含备选线路）**

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=bus/simple&encode=utf-8&resType=xml&per=150&number=10&night=0&maxtrans=4&type=0&sid=8001&ditance=1500,1500&lineName=0000,85&xys=123.4295486111,41.8523566667;123.4885155556,41.8074586111&city=沈阳&rid=589375&group=1&key=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

**示例三：isGateWayTip=1 （含地铁出入口提示）**

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=bus/simple&encode=utf-8&resType=xml&per=150&number=10&night=0&maxtrans=4&type=0&sid=8001&ditance=1500,1500&lineName=0000,85&xys=116.3159566667,39.9798866667;116.3161325,39.9596555556&city=北京&rid=589375 & isGateWayTip=1 &key=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

**示例四：isWalk=1 （含步行指引）**

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=bus/simple&encode=utf-8&resType=xml&per=150&number=10&night=0&maxtrans=4&type=0&sid=8001&ditance=1500,1500&lineName=0000,85&xys=123.454712,41.723336;123.403797,41.832525&city=沈阳&rid=589375& isWalk=1 &key=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

**示例五：landmark=10 & landmarkRange=500（含地标）**

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=bus/simple&encode=utf-8&resType=xml&per=150&number=10&night=0&maxtrans=4&type=0&sid=8001&ditance=1500,1500&lineName=0000,85&xys=123.454712,41.723336;123.403797,41.832525&city=沈阳&rid=589375 & landmark=10& landmarkRange=500&key=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

**示例六：via=1（带途经点请求示例）**

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=bus/simple&encode=utf-8&resType=xml&type=0&sid=8001&xys=123.454712,41.723336;123.403797,41.832525;123.4654712,41.723336&city=%E6%B2%88%E9%98%B3&rid=1&via=1&key= [用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - 1. 返回结果说明

**示例一结果展示：**

<searchresult servername=“服务器名” versionname=”版本号”>

<searchtime>查询时间</searchtime>

<count>10</count>

<list type="list"><！--换乘方案列表-->

<bus><！--换乘方案1-->

<expense>0</expense><！--换乘方案价格-->

<segmentList><！--换乘段列表，每段代表一次换乘-->

<segment>

<startName>世纪大厦</startName><!—起点名称-->

<endName>中医药大学</endName> <!—终点名称-->

<busName>地铁2号线(全运路--蒲田路)</busName><!—公交名称-->

<startTime>暂时没有提供</startTime><!—首发车时间-->

<endTime>暂时没有没提供</endTime><!—末班车时间-->

<passDepotName>营盘街 奥体中心 五里河 市图书馆 工业展览馆 青年大街 市府广场 金融中心 沈阳北站 岐山路</passDepotName><!—路径站点-->

<driverLength>13724</driverLength><!—行驶距离-->

<footLength>575</footLength><!—步行距离-->

<passDepotCount>10</passDepotCount><!—途径站点数-->

<coordinateList>

123.460769,41.720825,123.46067,41.72098916666667, 123.459671,41.722815833333335...

</coordinateList><！--路线坐标-->

<passDepotCoordinate>

123.455978,41.731185833333335,123.451538,41.74377, 123.441024,41.75215083333333...

</passDepotCoordinate><！--途径站点坐标，和passDepotCount对应，passDepotCount=1，此段为1对坐标-->

<footCoordinateList/><！--步行站点坐标，暂时未提供-->

</segment>

<segment><--下面为换乘段-->

<startName>辽宁中医</startName><!—起点名称-->

<endName>百鸟公园</endName><!—终点名称-->

<busName>205路(富山花园--沈阳站北)</busName><!—公交名称-->

<startTime>0530</startTime><!—首发车时间-->

<endTime>1930</endTime><!—末班车时间-->

<passDepotName>市金融学校 辽宁大学崇山校区

</passDepotName><!—公交站名称-->

<driverLength>2075</driverLength><!—行驶距离-->

<footLength>94</footLength><!—步行距离-->

<passDepotCount>2</passDepotCount><!—途径站数-->

<coordinateList>

123.428153,41.83247333333333,123.427627,41.83245083333333,123.425765...

</coordinateList><！--路线坐标-->

<passDepotCoordinate>

123.418617,41.83202333333333,123.411254, 41.831675833333335

</passDepotCoordinate><！--途径站点坐标，和passDepotCount对应，passDepotCount=1，此段为1对坐标-->

<footCoordinateList/><！--步行坐标，暂时未提供-->

</segment>

</segmentList>

<footEndLength>141</footEndLength><！--步行距离-->

<footEndCoordinateList/>

<bounds>123.403175;41.720825;123.460769;

41.83247333333333;</bounds>

</bus>

<bus>...</bus>< --公交方案2-->

<bus>...</bus>< --公交方案3-->

...

</list>

</searchresult>

**示例二结果展示：**

<searchresult servername=“服务器名” versionname=”版本号”>

<status>E0</status>

<time>0.809</time>

<count>10</count>

<list type="list">

<bus>

<expense>5</expense>

<segmentList>

<segment>

<startName>金山小区</startName>

<endName>北陵公园</endName>

<busName>242路(金山小区--保工街南十四路)</busName>

<startTime>05:20</startTime>

<endTime>21:00</endTime>

<passDepotName>北陵东门 省军区北门 北陵公园</passDepotName>

<driverLength>2011</driverLength>

<footLength>182</footLength>

<passDepotCount>3</passDepotCount>

<coordinateList>

123.437889,41.850800;123.438760,41.847646;123.438775,41.847588;123.439221,41.845921;...

</coordinateList>

<passDepotCoordinate>

123.438775,41.847588;123.436249,41.842548;123.426817,41.839194

</passDepotCoordinate>

<AlterSegment type="list">< --备选线路列表-->

<AlterLine>< --第一条备选线路-->

<busName>227路(金山小区--煤炭设计院)</busName>< --名称-->

<startTime>05:00</startTime>< --线路起始时间-->

<endTime>19:40</endTime>< --线路末班车时间-->

<passDepotName>北陵东门 省军区北门</passDepotName>< --途径站点名称-->

<driverLength>2150</driverLength>< --行驶长度-->

<footLength>859</footLength>< --步行长度-->

<passDepotCount>2</passDepotCount>< --途经站点数-->

<coordinateList>< --线路坐标点-->

123.437674,41.851545,123.439205,41.845953,123.438898,41.845695,123.438018,41.844486,...

</coordinateList>

<passDepotCoordinate>123.438018,41.844486,123.434104,41.841041</passDepotCoordinate>< --途径站点坐标-->

</AlterLine>

</AlterSegment>

</segment>

<segment>...</segment>

<segment>...</segment>

</segmentList>

<footEndLength>341</footEndLength>

<bounds>123.424915;41.792204;123.488329;41.8508

</bounds>

</bus>

<bus>...</bus>

...

</list>

</searchresult>

**示例三结果展示：**

<searchresult servername=“服务器名” versionname=”版本号”>

<status>E0</status>

<time>0.244</time>

<count>10</count>

<list>

<bus>

<expense>2</expense>

<segmentList type="list">

<segment>

<inport>< --地铁入口提示-->

<name>D(西南口)</name>< --地铁入口名称-->

<coord>116.315964,39.983415</coord>< --地铁入口坐标-->

</inport>

<outport>< --地铁出口提示-->

<name>A(西北口)</name>< --地铁出口名称-->

<coord>116.322800,39.958297</coord>< --地铁出口坐标-->

</outport>

<startName>中关村</startName>

<endName>魏公村</endName>

<busName>地铁4号线(安河桥北--公益西桥)</busName>

<startTime>05:00</startTime>

<endTime>22:45</endTime>

<passDepotName>海淀黄庄 人民大学 魏公村</passDepotName>

<driverLength>3020</driverLength>

<footLength>422</footLength>

<passDepotCount>3</passDepotCount>

<coordinateList>

116.316488,39.983910;116.316656,39.979614;116.316842,39.978625;116.316981,39.977918;...

</coordinateList>

<passDepotCoordinate>

116.317529,39.976001;116.321223,39.966916;116.323074,39.958096

</passDepotCoordinate>

</segment>

</segmentList>

<footEndLength>778</footEndLength>

<bounds>116.316488;39.958096;116.323074;39.98391</bounds>

</bus>

<bus>...</bus>

...

</list>

</searchresult>

**示例四结果展示：**

<searchresult servername=“服务器名” versionname=”版本号”>

<status>E0</status>

<time>0.254</time>

<count>10</count>

<list>

<bus>

<expense>3</expense>

<segmentList>

<segment>

<startName>世纪大厦</startName>

<endName>中医药大学</endName>

<busName>地铁2号线(全运路--蒲田路)</busName>

<startTime>05:30</startTime>

<endTime>21:00</endTime>

<passDepotName>

营盘街 奥体中心 五里河 市图书馆 工业展览馆 青年大街 市府广场 金融中心 沈阳北站 岐山路 中医药大学

</passDepotName>

<driverLength>13310</driverLength>

<footLength>914</footLength>

<passDepotCount>11</passDepotCount>

<coordinateList>

123.460772,41.720827;123.460672,41.720988;123.459670,41.722817;123.458225,41.725210;...

</coordinateList>

<passDepotCoordinate>

123.455982,41.731186;123.451536,41.743771;123.441026,41.752150;123.434828,41.760548;...

</passDepotCoordinate>

<walk type="list">< --步行指引-->

<dir>东南</dir>< --方向-->

<info>< --第一段步行信息-->

<main>向右前方</main>< --主动做-->

<assist>无辅助导航动作</assist>< --辅动作-->

<road>新隆街</road>< --道路名称-->

<dis>190</dis>< --步行距离-->

<coord>< -- 步行线路坐标-->

123.454878,41.723390,123.455055,41.723099,123.455139,41.722946,123.455734,41.721966,123.455894,41.721870

</coord>

</info>

<info>...</info>< --第二段步行信息-->

...

</walk>

</segment>

<segment>...</segment>

</segmentList>

<footEndLength>150</footEndLength>

<bounds>123.403476;41.720827;123.460772;41.832667</bounds>

<walkEnd type="list">< --最后一段到终点的步行指引-->

<dir>东</dir>< --方向-->

<info>

<main>左转</main>< --主动做-->

<assist>无辅助导航动作</assist>< --辅动作-->

<road>崇山中路</road>< --道路名称-->

<dis>27</dis>< --步行距离-->

<coord>123.403472,41.831324,123.403801,41.831337</coord>< --步行坐标-->

</info>

<info>...</info>

...

</walkEnd>

</bus>

<bus>...</bus>

...

</list>

</searchresult>

**示例五结果展示：**

<searchresult servername=“服务器名” versionname=”版本号”>

<status>E0</status>

<time>0.427</time>

<count>10</count>

<list>

<bus>

<expense>3</expense>

<segmentList>

<segment>

<startName>世纪大厦</startName>

<endName>中医药大学</endName>

<busName>地铁2号线(全运路--蒲田路)</busName>

<startTime>05:30</startTime>

<endTime>21:00</endTime>

<passDepotName>

营盘街 奥体中心 五里河 市图书馆 工业展览馆 青年大街 市府广场 金融中心 沈阳北站 岐山路 中医药大学

</passDepotName>

<driverLength>13310</driverLength>

<footLength>914</footLength>

<passDepotCount>11</passDepotCount>

<coordinateList>

123.460772,41.720827;123.460672,41.720988;123.459670,41.722817;123.458225,41.725210;...

</coordinateList>

<passDepotCoordinate>

123.455982,41.731186;123.451536,41.743771;123.441026,41.752150;123.434828,41.760548;...

</passDepotCoordinate>

<landmarkStartList type="list">< --起点地标列表-->

<landmark>< --第一个地标信息-->

<name>二十一世纪大厦</name>< --地标名称-->

<tel>024-23745002</tel>< --电话-->

<address>世纪路1号</address>< --地址-->

<x>123.460605</x>< --X坐标-->

<y>41.7184816667</y>< --y坐标-->

<type>商务住宅;楼宇;商务写字楼</type>< --类型-->

<direction>South</direction>< --方向-->

<distance>261.163</distance>< --距离-->

</landmark>

<landmark>...</landmark>< --第二个起点地标-->

<landmark>...</landmark>< --第三个起点地标-->

</landmarkStartList>

<landmarkEndList type="list">< --终点地标列表-->

<landmark>

<name>御金名苑</name>< --名称-->

<tel></tel>< --电话-->

<address></address>< --地址-->

<x>123.42732</x>< --x坐标-->

<y>41.8316736111</y>< --y坐标-->

<type>商务住宅;住宅区;住宅小区</type>< --类型-->

<direction>West</direction>< --方向-->

<distance>185.069</distance>< --距离-->

</landmark>

<landmark>...</landmark>< --第二个终点地标-->

<landmark>...</landmark>< --第三个终点地标-->

</landmarkEndList>

</segment>

<segment>...</segment>

</segmentList>

<footEndLength>150</footEndLength>

<bounds>123.403476;41.720827;123.460772;41.832667</bounds>

</bus>

<bus>...</bus>

...

</list>

</searchresult>

**示例六结果展示：**

<searchresult servername="服务器名" ver="版本号">

<status>E0</status>

<time>0.147</time>

<stations>3</stations><!-- 请求点的个数（包括起点、终点、途经点） -->

<routeList type="list"><!-- 换乘方案描述列表 -->

<route><!-- 各部分的换乘方案描述 -->

<count>6</count>

<busList type="list">

<bus>

<expense>3</expense>

<segmentList type="list">

<segment>

<startName>世纪大厦</startName>

<endName>中医药大学</endName>

<busName>地铁2号线(全运路--蒲田路)</busName>

...

</bus>

<bus>...</bus>

<bus>...</bus>

<bus>...</bus>

<bus>...</bus>

<bus>...</bus>

</busList>

</route>

<route>...</route>

...

</routeList>

</searchresult>

* + - 1. 其他需要说明的部分
    1. 公交线路服务请求接口
       1. 根据道路ID查询公交路线
          1. 接口功能

根据公交线路id查询公交线路数据

* + - * 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| config | config=BusLine | 服务参数名称 | 必填参数 |
| cityCode | cityCode=城市代码 | 为限定城市区域查询 | 必填参数 |
| resType | resType=xml | 返回格式 |  |
| ability | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必添参数 |
| ids | ids=道路id集合 | 道路id集合 ，如果有多个id中间用","分割。 | 必填参数 |
| number | number=10 | 每页记录数 | 可选默认10条 |
| batch | batch=1 | 页号 | 可选默认为1 |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

* + - * 1. 接口URL示例

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&highLight=false&config=BusLine&ver=2.0&ids=110100011436&cityCode=%E5%8C%97%E4%BA%AC&enc=utf-8&resType=xml&resData=0&flag=1&number=10&batch=1&ctx=1547165&a\_nocache=515601701164&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - * 1. 返回结果说明

<?xml version="1.0" encoding="GBK"?>

<searchresult servername="服务器名" versionname=”版本号”>

<count>此接口返回的结果集数量（默认最大结果集为400，可通过key权限修改，需要联系管理员）</count>

<cache>此数据是否从缓存读取</cache>

<total>引擎返回的结果集总数量</total>

<record>此页返回的结果集数量（最大结果集为20，可通过key权限修改，需要联系管理员）</record>

<searchtime>查询时间</searchtime>

<list >

<bus>

<!--线路名称，包含线路编号和文字名称、类型、起止点-- >

<name>**300路内环(草桥--草桥)**</name>

<!--公交线路的类型，比如：普通公交、地铁等-- >

<type>**1**</type>

<!--线路的长度（单位：公里）-->

<length>**50.02910**</length>

<!--线路的备注信息-- >

<description>**以下为实际运行站点，照片0012为价格表**</description>

<!--线路状态，0：停止，1：普通-- >

<status>**1**</status>

<!--线路唯一id标识-- >

<line\_id>**110100011436**</line\_id>

<!--线路的名称的关键部分-- >

<key\_name>**300路内环**</key\_name>

<!--始发站名-- >

<front\_name>**草桥**</front\_name>

<!--始发站点拼音名称-- >

<front\_spell>**"cao qiao"**</front\_spell>

<!--终点站名称-- >

<terminal\_name>**草桥**</terminal\_name>

<!--终点站点拼音名称-- >

<terminal\_spell>**"cao qiao"**</terminal\_spell>

<!--首班车发车时间，格式初定为”HHMM”,时间为24小时制(0000--2359) -- >

<start\_time>**0530**</start\_time>

<!--末班车发车时间，格式初定为”HHMM”,时间为24小时制(0000--2359) -- >

<end\_time>**2200**</end\_time>

<!--服务时间周期描述，例如：夏季线路、周末线路、夜班车等-- >

<service\_period />

<!--发车间隔1 -- >

<time\_interval1>**05302200**</time\_interval1>

<!--发车间隔1适用的时间段，起始时间（HHMM）空格截至时间（HHMM）-- >

<interval1 />

<!--发车间隔2 -- >

<time\_interval2 />

<!--发车间隔2适用的时间段，起始时间（HHMM）空格截至时间（HHMM）-- >

<interval2 />

<!--发车间隔3 -- >

<time\_interval3 />

<!--发车间隔3适用的时间段，起始时间（HHMM）空格截至时间（HHMM）-- >

<interval3 />

<!-- 发车间隔4 -- >

<time\_interval4 />

<!--发车间隔4适用的时间段，起始时间（HHMM）空格截至时间（HHMM）-- >

<interval4 />

<!--发车间隔5 -- >

<time\_interval5 />

<!--发车间隔5适用的时间段，起始时间（HHMM）空格截至时间（HHMM） -- >

<interval5 />

<!--发车间隔6-- >

<time\_interval6 />

<!--发车间隔6适用的时间段，起始时间（HHMM）空格截至时间（HHMM）-- >

<interval6 />

<!--发车间隔7-- >

<time\_interval7 />

<!--发车间隔7适用的时间段，起始时间（HHMM）空格截至时间（HHMM） -- >

<interval7 />

<!--发车间隔8 -- >

<time\_interval8 />

<!--发车间隔8适用的时间段，起始时间（HHMM）空格截至时间（HHMM）-- >

<interval8 />

<!--用文字来描述无特定发车规则（即发车无规律）的发车信息-- >

<time\_desc />

<!--公交所属运营公司-- >

<company>**北京公交集团有限公司第二客运分公司**</company>

<!--基本价格-- >

<basic\_price>**1**</basic\_price>

<!--总价格（当为单一票价的时候，总价格=基本价格）-- >

<total\_price>**5**</total\_price>

<!--是否可以使用月票，0：无，1：有-- >

<commutation\_ticket>**0**</commutation\_ticket>

<!--是否自动（无人）售票，0：否，1：是-- >

<auto>**0**</auto>

<!--是否使用电子售票/交通卡，0：否，1：是-- >

<ic\_card>**1**</ic\_card>

<!--公共交通线是否路经高速道路，0:否，1：是-- >

<express\_way>**0**</express\_way>

<!--是否环线，没有起点，没有终点，周而复始，比如地铁环线。0：否，1：是-- >

<loop>**1**</loop>

<!--是否双层，0：否，1：是-- >

<double\_deck>**0**</double\_deck>

<!--采集线路时，采到的线路对应名称-- >

<gpsfile\_id />

<!--采集线路时，采集到的照片的编号-- >

<photo\_id />

<!--采集线路时，采集到的照片所放置的文件夹的名称-- >

<photo\_folder />

<!--采集线路时，所填写的纸质表格的编号-- >

<paper\_table\_id />

<!--线路的平均行驶速度（目前是以采集到的速度为准）,单位为 km/h -- >

<speed />

<!--线路坐标-- >

<xys />

<!--数据来源。1：自行采集，0：其他-- >

<data\_source>**1**</data\_source>

<!--车站xml描述-- >

<stationdes />

<!--是否有空调，0：无，1：有-- >

<air>**0**</air>

</bus>

</list>

</searchresult>

* + - * 1. 其他需要说明的部分

**公交站点扩展XML描述：**

站台（二次聚类）：一个二次聚类站台包含一个或多个一次聚类站台，一个一次聚类站台包含一个或多个单一站点

包含字段如下表：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| BUSINFO\_ALIAS |  | 站台别名 |  |
| BUSINFO\_LINE\_KEYS |  | 经过此站台的公交线路名称关键字,由一个或多个一次聚类站台经过的线路关键字组合而成，一次聚类站台有多条线路经过时以;分隔，二次聚类有多个一次聚类站台构成时以|分隔。如：376路;491路内环;919路支3空调|376路;492路外环;919路支3空调 |  |
| BUSINFO\_LINEIDS |  | 经过此站台的公交线路id，同BUSINFO\_LINE\_KEYS |  |
| XS |  | 此站台包含的一次聚类站台X坐标，以|分隔 |  |
| YS |  | 此站台包含的一次聚类站台Y坐标,以|分隔 |  |
| BUSINFO\_LINE\_NAMES |  | 经过此站台的公交线路名称，同BUSINFO\_LINE\_KEYS |  |
| BUSINFO\_ANGLES |  | 此站台包含的一次聚类站台的角度,以|分隔 |  |

**示例：**

<?xml version="1.0" encoding="UTF-8"?>

<CONTENT>

<DATA NAME="BUSINFO\_ALIAS"><![CDATA[]]></DATA>

<DATA NAME="BUSINFO\_LINE\_KEYS"><![CDATA[376路;491路内环;919路支3空调|376路;492路外环;919路支3空调]]></DATA>

<DATA NAME="BUSINFO\_LINEIDS"><![CDATA[110100011632;110100013001;110100012302|1101 00011633;110100013000;110100012303]]></DATA>

<DATA NAME="XS"><![CDATA[116.250760|116.250836]]></DATA>

<DATA NAME="YS"><![CDATA[40.211440|40.211040]]></DATA>

<DATA NAME="BUSINFO\_LINE\_NAMES"><![CDATA[376路(昌平东关--昌平南口北站);491路内环(朝凤庵村--朝凤庵村);919路支3空调(昌平东关--德胜门)|376路(昌平南口北站--昌平东关);492路外环(朝凤庵村--朝凤庵村);919路支3空调(德胜门--昌平东关)]]></DATA>

<DATA NAME="BUSINFO\_ANGLES"><![CDATA[271.012505|90.471537]]></DATA>

</CONTENT>

* + - 1. 根据公交名称查询公交路线
         1. 接口功能

根据公交线路名查询公交线路

* + - * 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| config | config=BusLine | 服务参数名称 | 必填参数 |
| cityCode | cityCode=城市代码 | 为限定城市区域查询 | 必填参数 |
| resType | resType=xml | 返回格式 |  |
| ability | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | abilityuri= sisserver | 请求类型 | 必添参数 |
| busName | busName=公交名 | 公交名，如busName=300，将查询300路公交车的途径站点信息。 | 必填参数 |
| resData | resData =0/1 | 返回数据样式 |  |
| number | number=10 | 每页记录数 | 可选默认10条 |
| batch | batch=1 | 页号 | 可选默认为1 |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

* + - * 1. 接口URL示例

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&highLight=false&config=BusLine&ver=2.0&busName=300&cityCode=%E5%8C%97%E4%BA%AC&enc=utf-8&resType=xml&resData=1&flag=1&number=10&batch=1&ctx=1231033&a\_nocache=315300191174&a\_k=[用户KEY]

* + - * 1. 返回结果说明

<?xml version="1.0" encoding="GBK"?>

<searchresult servername="服务器名" versionname=”版本号”>

<count>此接口返回的结果集数量（默认最大结果集为400，可通过key权限修改，需要联系管理员）</count>

<cache>此数据是否从缓存读取</cache>

<total>引擎返回的结果集总数量</total>

<record>此页返回的结果集数量（最大结果集为20，可通过key权限修改，需要联系管理员）</record>

<searchtime>查询时间</searchtime>

<list>

<bus>

<!--线路名称，包含线路编号和文字名称、类型、起止点-- >

<name>**300路内环(草桥--草桥)**</name>

<!--公交线路的类型，比如：普通公交、地铁等-- >

<type>**1**</type>

<!--线路的长度（单位：公里）-->

<length>**50.02910**</length>

<!--线路的备注信息-- >

<description>**以下为实际运行站点，照片0012为价格表**</description>

<!--线路状态，0：停止，1：普通-- >

<status>**1**</status>

<!--线路唯一id标识-- >

<line\_id>**110100011436**</line\_id>

<!--线路的名称的关键部分-- >

<key\_name>**300路内环**</key\_name>

<!--始发站名-- >

<front\_name>**草桥**</front\_name>

<!--始发站点拼音名称-- >

<front\_spell>**"cao qiao"**</front\_spell>

<!--终点站名称-- >

<terminal\_name>**草桥**</terminal\_name>

<!--终点站点拼音名称-- >

<terminal\_spell>**"cao qiao"**</terminal\_spell>

<!--首班车发车时间，格式初定为”HHMM”,时间为24小时制(0000--2359) -- >

<start\_time>**0530**</start\_time>

<!--末班车发车时间，格式初定为”HHMM”,时间为24小时制(0000--2359) -- >

<end\_time>**2200**</end\_time>

<!--服务时间周期描述，例如：夏季线路、周末线路、夜班车等-- >

<service\_period />

<!--发车间隔1 -- >

<time\_interval1>**05302200**</time\_interval1>

<!--发车间隔1适用的时间段，起始时间（HHMM）空格截至时间（HHMM）-- >

<interval1 />

<!--发车间隔2 -- >

<time\_interval2 />

<!--发车间隔2适用的时间段，起始时间（HHMM）空格截至时间（HHMM）-- >

<interval2 />

<!--发车间隔3 -- >

<time\_interval3 />

<!--发车间隔3适用的时间段，起始时间（HHMM）空格截至时间（HHMM）-- >

<interval3 />

<!-- 发车间隔4 -- >

<time\_interval4 />

<!--发车间隔4适用的时间段，起始时间（HHMM）空格截至时间（HHMM）-- >

<interval4 />

<!--发车间隔5 -- >

<time\_interval5 />

<!--发车间隔5适用的时间段，起始时间（HHMM）空格截至时间（HHMM） -- >

<interval5 />

<!--发车间隔6-- >

<time\_interval6 />

<!--发车间隔6适用的时间段，起始时间（HHMM）空格截至时间（HHMM）-- >

<interval6 />

<!--发车间隔7-- >

<time\_interval7 />

<!--发车间隔7适用的时间段，起始时间（HHMM）空格截至时间（HHMM） -- >

<interval7 />

<!--发车间隔8 -- >

<time\_interval8 />

<!--发车间隔8适用的时间段，起始时间（HHMM）空格截至时间（HHMM）-- >

<interval8 />

<!--用文字来描述无特定发车规则（即发车无规律）的发车信息-- >

<time\_desc />

<!--公交所属运营公司-- >

<company>**北京公交集团有限公司第二客运分公司**</company>

<!--基本价格-- >

<basic\_price>**1**</basic\_price>

<!--总价格（当为单一票价的时候，总价格=基本价格）-- >

<total\_price>**5**</total\_price>

<!--是否可以使用月票，0：无，1：有-- >

<commutation\_ticket>**0**</commutation\_ticket>

<!--是否自动（无人）售票，0：否，1：是-- >

<auto>**0**</auto>

<!--是否使用电子售票/交通卡，0：否，1：是-- >

<ic\_card>**1**</ic\_card>

<!--公共交通线是否路经高速道路，0:否，1：是-- >

<express\_way>**0**</express\_way>

<!--是否环线，没有起点，没有终点，周而复始，比如地铁环线。0：否，1：是-- >

<loop>**1**</loop>

<!--是否双层，0：否，1：是-- >

<double\_deck>**0**</double\_deck>

<!--采集线路时，采到的线路对应的名称-- >

<gpsfile\_id />

<!--采集线路时，采集到的照片的编号-- >

<photo\_id />

<!--采集线路时，采集到的照片所放置的文件夹的名称-- >

<photo\_folder />

<!--采集线路时，所填写的纸质表格的编号-- >

<paper\_table\_id />

<!--线路的平均行驶速度（目前是以采集到的速度为准）,单位为 km/h -- >

<speed />

<!--线路坐标-- >

<xys />

<!--数据来源。1：自行采集，0：其他-- >

<data\_source>**1**</data\_source>

<!--车站xml描述-- >

<stationdes />

<!--是否有空调，0：无，1：有-- >

<air>**0**</air>

</bus>

</list>

</searchresult>

* + - * 1. 其他需要说明部分

公交站点扩展XML描述请见3.2.1.5。

* + - 1. 根据公交车站名称查询公交路线
         1. 接口功能

根据途经站点名查询经过改点的公交线路

* + - * 1. 接口参数说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数格式** | | **参数含义** | **备注** |
| config | config=BusLine | | 服务参数名称 | 必填参数 |
| cityCode | cityCode=城市代码 | | 为限定城市区域查询 | 必填参数 |
| resType | resType=xml | | 返回格式 |  |
| ability | | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | | abilityuri=sisserver | 请求类型 | 必添参数 |
| stationName | stationName=站点名 | | 站点名，如stationName=中关村南，将查询途径中关村南站的公交车信息。 | 必填参数 |
| resData | resData =0/1 | | 返回数据样式 |  |
| number | number=10 | | 每页记录数 | 可选默认10条 |
| batch | batch=1 | | 页号 | 可选默认为1 |
| enc | enc=utf-8 | | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=api key | | 用户申请的key | 必填参数 |

* + - * 1. 接口URL示例

[http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&highLight=false&config=BusLine&ver=2.0&stationName=%e4%b8%ad%e5%85%b3%e6%9d%91&cityCode=%E5%8C%97%E4%BA%AC&enc=utf-8&resType=xml&a\_k=[用户KEY]](http://221.180.145.25:9083/omp/gatesrv?omp_appid=000702000000000096&omp_restype=autonavi_Gis&extparam=&ability=apiserver&abilityuri=sisserver&highLight=false&config=BusLine&ver=2.0&stationName=%e4%b8%ad%e5%85%b3%e6%9d%91&cityCode=%E5%8C%97%E4%BA%AC&enc=utf-8&resType=xml&a_k=348d0073e7548f443b63b221ccc8b93c4607946b2659fc519890c924f82b4475ddd71b058178d02b)

* + - * 1. 返回结果说明

<?xml version="1.0" encoding="GBK"?>

<searchresult servername="服务器名" versionname=”版本号”>

<count>此接口返回的结果集数量（默认最大结果集为400，可通过key权限修改，需要联系管理员）</count>

<cache>此数据是否从缓存读取</cache>

<total>引擎返回的结果集总数量</total>

<record>此页返回的结果集数量（最大结果集为20，可通过key权限修改，需要联系管理员）</record>

<searchtime>查询时间</searchtime>

<list type="**list**">

<bus>

<!--线路名称，包含线路编号和文字名称、类型、起止点-- >

<name>**300路内环(草桥--草桥)**</name>

<!--公交线路的类型，比如：普通公交、地铁等-- >

<type>**1**</type>

<!--线路的长度（单位：公里）-->

<length>**50.02910**</length>

<!--线路的备注信息-- >

<description>**以下为实际运行站点，照片0012为价格表**</description>

<!--线路状态，0：停止，1：普通-- >

<status>**1**</status>

<!--线路唯一id标识-- >

<line\_id>**110100011436**</line\_id>

<!--线路的名称的关键部分-- >

<key\_name>**300路内环**</key\_name>

<!--始发站名-- >

<front\_name>**草桥**</front\_name>

<!--始发站点拼音名称-- >

<front\_spell>**"cao qiao"**</front\_spell>

<!--终点站名称-- >

<terminal\_name>**草桥**</terminal\_name>

<!--终点站点拼音名称-- >

<terminal\_spell>**"cao qiao"**</terminal\_spell>

<!--首班车发车时间，格式初定为”HHMM”,时间为24小时制(0000--2359) -- >

<start\_time>**0530**</start\_time>

<!--末班车发车时间，格式初定为”HHMM”,时间为24小时制(0000--2359) -- >

<end\_time>**2200**</end\_time>

<!--服务时间周期描述，例如：夏季线路、周末线路、夜班车等-- >

<service\_period />

<!--发车间隔1 -- >

<time\_interval1>**05302200**</time\_interval1>

<!--发车间隔1适用的时间段，起始时间（HHMM）空格截至时间（HHMM）-- >

<interval1 />

<!--发车间隔2 -- >

<time\_interval2 />

<!--发车间隔2适用的时间段，起始时间（HHMM）空格截至时间（HHMM）-- >

<interval2 />

<!--发车间隔3 -- >

<time\_interval3 />

<!--发车间隔3适用的时间段，起始时间（HHMM）空格截至时间（HHMM）-- >

<interval3 />

<!-- 发车间隔4 -- >

<time\_interval4 />

<!--发车间隔4适用的时间段，起始时间（HHMM）空格截至时间（HHMM）-- >

<interval4 />

<!--发车间隔5 -- >

<time\_interval5 />

<!--发车间隔5适用的时间段，起始时间（HHMM）空格截至时间（HHMM） -- >

<interval5 />

<!--发车间隔6-- >

<time\_interval6 />

<!--发车间隔6适用的时间段，起始时间（HHMM）空格截至时间（HHMM）-- >

<interval6 />

<!--发车间隔7-- >

<time\_interval7 />

<!--发车间隔7适用的时间段，起始时间（HHMM）空格截至时间（HHMM） -- >

<interval7 />

<!--发车间隔8 -- >

<time\_interval8 />

<!--发车间隔8适用的时间段，起始时间（HHMM）空格截至时间（HHMM）-- >

<interval8 />

<!--用文字来描述无特定发车规则（即发车无规律）的发车信息-- >

<time\_desc />

<!--公交所属运营公司-- >

<company>**北京公交集团有限公司第二客运分公司**</company>

<!--基本价格-- >

<basic\_price>**1**</basic\_price>

<!--总价格（当为单一票价的时候，总价格=基本价格）-- >

<total\_price>**5**</total\_price>

<!--是否可以使用月票，0：无，1：有-- >

<commutation\_ticket>**0**</commutation\_ticket>

<!--是否自动（无人）售票，0：否，1：是-- >

<auto>**0**</auto>

<!--是否使用电子售票/交通卡，0：否，1：是-- >

<ic\_card>**1**</ic\_card>

<!--公共交通线是否路经高速道路，0:否，1：是-- >

<express\_way>**0**</express\_way>

<!--是否环线，没有起点，没有终点，周而复始，比如地铁环线。0：否，1：是-- >

<loop>**1**</loop>

<!--是否双层，0：否，1：是-- >

<double\_deck>**0**</double\_deck>

<!--采集线路时，采到的线路对应的名称-- >

<gpsfile\_id />

<!--采集线路时，采集到的照片的编号-- >

<photo\_id />

<!--采集线路时，采集到的照片所放置的文件夹的名称-- >

<photo\_folder />

<!--采集线路时，所填写的纸质表格的编号-- >

<paper\_table\_id />

<!--线路的平均行驶速度（目前是以采集到的速度为准）,单位为 km/h -- >

<speed />

<!--线路坐标-- >

<xys />

<!--数据来源。1：自行采集，0：其他-- >

<data\_source>**1**</data\_source>

<!--车站xml描述-- >

<stationdes />

<!--是否有空调，0：无，1：有-- >

<air>**0**</air>

</bus>

</list>

</searchresult>

* + - * 1. 其他需要说明部分

公交站点扩展XML描述请见3.2.1.5。

* 1. 地址匹配接口
     1. 接口功能

根据地址信息，查询该地址所对应的点坐标等。

* + 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| config | config=GOC | Geocoding服务请求参数 | 必填参数 |
| ability | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必添参数 |
| address | address=地址 | 需要匹配的中文地址，例如：北京市海淀区苏州街 | 必填参数 |
| ver | ver=版本号 | ver=1.0/2.0返回不同的结果集 |  |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | key=api key | 用户申请的key | 必填参数 |

* + 1. 接口URL示例

[http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&config=GOC&address=%E5%8C%97%E4%BA%AC%E5%B8%82%E6%B5%B7%E6%B7%80%E5%8C%BA%E8%8B%8F%E5%B7%9E%E8%A1%97&resType=xml&enc=utf-8&a\_k=[用户KEY]](http://221.180.144.57:17083/sisserver?highLight=false&config=GOC&address=%E5%8C%97%E4%BA%AC%E5%B8%82%E6%B5%B7%E6%B7%80%E5%8C%BA%E8%8B%8F%E5%B7%9E%E8%A1%97&resType=xml&enc=utf-8&a_k=348d0073e7548f443b63b221ccc8b93c4607946b2659fc519890c924f82b4475ddd71b058178d02b&ctx=1520991&a_nocache=115909031134)

**注：此URL仅为示例使用，不是最终服务提供URL**

* + 1. 返回结果说明

<searchresult servername="**服务器名**" versionname="**版本号**">

<!-- 匹配结果数量 -->

<count>1</count>

<!--拼音纠错-->

<spellcorrect/>

<list type="list">

<poi>

<!—地址，该属性可能会出现-->

<address>苏州街</address>

<!—中文名称，该属性一定会出现-->

<name>北京市海淀区苏州街</name>

<!-- 匹配级别，标识坐标可信程度，该属性一定会出现-->

<level>GL\_NUMBER</level>

<!--参与本结果计算的POI的覆盖范围，（本地址的影响范围,单位米），该属性一定会出现-->

<range>342</range>

<!—匹配经度，该属性一定会出现-->

<x>116.305404</x>

<!—匹配纬度，该属性一定会出现-->

<y>39.982689</y>

<!--市级行政区划，直辖市、直辖县无赋值，该属性可能会出现-->

<city/>

<!—省级行政区划，该属性可能会出现-->

<province>北京市</province>

<!—区县级行政区划，该属性可能会出现-->

<district>海淀区</district>

<!—英文名称，该属性一定会出现-->

<ename>Beijing City Haidian District Suzhou Street No. 3</ename>

<!--权重公式计算结果，该属性可能会出现-->

<score>0.625000</score>

<!—英文省级行政区划，该属性可能会出现-->

<eprovince>Beijing City</eprovince>

<!--英文市级行政区划，直辖市、直辖县无赋值-->

<ecity/>

<inum/>

<prox/>

<!—英文区县级行政区划，该属性可能会出现-->

<edistrict>Haidian District</edistrict>

<!—英文地址，该属性可能会出现-->

<eaddress>Suzhou Street No. 3</eaddress>

<!--最近一条道路的道路点集，该属性可能会出现-->

<roadpts>116.305404,39.982689…</roadpts>

<!--本地址下所有相关POI，该属性可能会出现-->

<subpois>

<subpoi>

<!-- POI名称-- >

<name>大河庄苑</name>

<!—POI英文名称 -- >

<ename>Dahe Zhuangyuan</ename>

<!-- POI经度-- >

<x>116.305404</x>

<!-- POI纬度-- >

<y>39.982689</y>

</subpoi>

…

</subpois>

</poi>

…

</list>

</searchresult>

* + 1. 其他需要说明的部分

结果级别共分为10级，有几级是预留的，已用的按级别从大粒度到小粒度分别是：

* GL\_COUNTRY # 国家，目前只有中国
* GL\_PROVINCE # 省
* GL\_CITY # 市地，包括直辖市
* GL\_DISTRICT # 区县
* GL\_SUBDISTRICT # 区域地名，包括村镇、商圈等
* GL\_BUS # 公交站点
* GL\_ROAD # 不带点集的道路
* GL\_LINE # 带点集的道路
* GL\_NUMBER # 数字，包括门牌、楼层、房间号等
* GL\_POI # 兴趣点
* GL\_OTHER # 其他类型
* GL\_SIZE # 保留级别，不会返回
* GL\_INVALID # 保留级别，不会返回
* GL\_INTER # 插值的结果
* GL\_NEARBY # 附近的门牌
* GL\_SUB # 包括方位、小区（例如东一区）、多个数字表示的楼层等
  1. 位置描述接口
     1. 单点位置描述
        1. 接口功能介绍

根据xy坐标返回此点的空间位置描述信息，包括该点所在省市区，周边的POI和道路。

* + - 1. 接口参数说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数格式** | | **参数含义** | **备注** |
| config | config=SPAS | | 服务参数名称 | 必填参数 |
| ability | | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | | abilityuri=sisserver | 请求类型 | 必添参数 |
| enc | enc=utf-8 | | 编码规则 | 必填参数 |
| spatialXml | spatialXml=查询条件 | | spatialXml参数为一个标准的xml格式 | 必填参数 |
| resType | resType=xml | | 返回格式 |  |
| enc | enc=utf-8 | | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | key=api key | | 用户申请的key | 必填参数 |

**未编码前的spatialXml：**

<?xml version="1.0" encoding="GB2312"?>

<spatial\_request method="searchPoint">

<!--x坐标-->

<x>116.334</x>

<!-- y坐标-->

<y>39.9872</y>

<!-- 返回周边的POI数量-->

<poiNumber>10</poiNumber>

<!-- 限定周边热点POI和道路的距离范围-->

<range>300</range>

<!--返回数据的模式，0为返回地标性POI，1为返回全部POI，

如果xy在繁华地区设置为0效果较好，如果xy在偏远地区设置1更好-->

<pattern>0</pattern>

<!-- 道路等级，详见5.1.5节-->

<roadLevel>0</ roadLevel >

</spatial\_request>

* + - 1. 接口URL示例

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&config=SPAS&enc=utf-8&spatialXml=%3C%3Fxml+version%3D%221.0%22+encoding%3D%22GB2312%22%3F%3E%0D%0A%3Cspatial\_request+method%3D%22searchPoint%22%3E%0D%0A+%3Cx%3E116.334%3C%2Fx%3E%0D%0A+%3Cy%3E39.9872%3C%2Fy%3E%0D%0A+%3CpoiNumber%3E10%3C%2FpoiNumber%3E%0D%0A+%3Crange%3E300%3C%2Frange%3E%0D%0A+%3Cpattern%3E0%3C%2Fpattern%3E%0D%0A+%3CroadLevel%3E0%3C%2FroadLevel%3E%0D%0A%3C%2Fspatial\_request%3E&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - 1. 返回结果说明

<?xml version="1.0" encoding="utf-8" ?>

<spatial\_response type="**SpatialBean**" servername="**服务器名**" versionname="**版本号**">

<SpatialBean ver="**1.0**">

<Province ver="**1.0**">

<name>省名</name>

<code>省编号</code>

</Province>

<City ver="**1.0**">

<name>市名</name>

<code>区域名称</code>

<telnum>区号</telnum>

</City>

<District ver="**1.0**">

<name>地区名称 </name>

<code>地区编码</code>

<x>x坐标</x>

<y>y坐标</y>

<bounds>bounds值</bounds>

</District>

<roadList type="**list**">

<Road ver="**1.0**">

<id>道路id</id>

<name>道路名称</name>

<direction>相对xy方向</direction>

<distance>相对距离</distance>

<width>宽度</width>

<ename>英文名称</ename>

<level>等级</level>

</Road>

</roadList>

<poiList type="**list**">

<poi>

<address>地址</address>

<name>名称</name>

<id>poi数据id</id>

<type>数据类型</type>

<timestamp>时间戳</timestamp>

<match>匹配度</match>

<url >url</url>

<citycode>城市代码</citycode>

<xml >扩展描述信息</xml>

<x>x坐标</x>

<y>y坐标</y>

<code>区域代码</code>

<distance>距离查询点距离</distance>

<drivedistance>导航距离查询点距离</drivedistance>

<pguid>索引id</pguid>

<srctype>数据源类型</srctype>

<gridcode>gridcode代码</gridcode>

<tel>联系电话</tel>

<buscode>公交代码</buscode>

<icon>图标</icon>

<direction>距离查询点的方位</direction>

<extid>外部扩展id</extid>

<en\_type>英文类型</en\_type>

<en\_name>英文名称</en\_name>

<en\_address>英文地址</en\_address>

</poi>

</poiList>

<crossPoiList type="**list**">

<cross>

<name>交叉路名</name>

<x>x坐标</x>

<y>y坐标</y>

</cross>

</crossPoiList>

</SpatialBean>

</spatial\_response>

* + - 1. 其他需要说明的部分

**1、地标性POI包括以下类型点：**

大厦、写字楼、电视台、公园、风景点、博物馆、展览馆、体育场馆、图书馆、宾馆、酒店、医院、大学、影剧院、中学、小学、银行、政府机关、科研院所、公安机关、邮政局 、电信营业厅、幼儿园。

**2、道路等级说明**

roadLevel - 道路等级,此参数目前为0－3，和国家标准的道路等级划分的对应关系如下:  
roadLevel＝0,代表的标准道路等级0-11范围的道路

roadLevel＝1,代表的标准道路等级2-11范围的道路  
roadLevel＝2,代表的标准道路等级7-11范围的道路  
roadLevel＝3,代表的标准道路等级9-11范围的道路

国家标准的道路等级划分共12级,参考如下：0 图廓线 未命名道路  
1 非导航道路  
2 普通道路  
3 次要道路  
4 县乡村内部道路  
5 乡公路  
6 县道  
7 主要道路  
8 主要大街、城市快速道  
9 省道  
10 国道  
11 高速公路

* + 1. 多点位置描述
       1. 接口功能介绍

根据多个点xy坐标返回这些点的空间位置描述信息，包括该点所在省市区，周边的POI和道路。

* + - 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| config | config=SPAS | 服务参数名称 | 必填参数 |
| ability | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必添参数 |
| enc | enc=utf-8 | 编码规则 | 必填参数 |
| spatialXml | spatialXml=查询条件 | spatialXml参数为一个标准的xml格式 | 必填参数 |
| resType | resType=xml | 返回格式 |  |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

**未编码前的spatialXml：**

<?xml version="1.0" encoding="GB2312"?>

<spatial\_request method="searchPoints">

<!-- x坐标集合-->

<xs>116.319358,116.219358,</xs>

<!--y坐标集合-->

<ys>39.917411,39.817411</ys>

<!-- 道路等级，详见3.1.5->

<roadLevel>0</roadLevel>

</spatial\_request>

* + - 1. 接口URL示例

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&config=SPAS&enc=utf-8&spatialXml=%3C%3Fxml+version%3D%221.0%22+encoding%3D%22GB2312%22%3F%3E%0D%0A%3Cspatial\_request+method%3D%22searchPoints%22%3E%0D%0A+%3Cxs%3E116.3%2C116.2%2C116.4%3C%2Fxs%3E%0D%0A+%3Cys%3E39.1%2C39.2%2C39.3%3C%2Fys%3E%0D%0A+%3CpoiNumber%3E10%3C%2FpoiNumber%3E%0D%0A+%3Crange%3E10%3C%2Frange%3E%0D%0A+%3Cpattern%3E0%3C%2Fpattern%3E%0D%0A+%3CroadLevel%3E0%3C%2FroadLevel%3E%0D%0A%3C%2Fspatial\_request%3E&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - 1. 返回结果说明

<?xml version="1.0" encoding="utf-8" ?>

<spatial\_response type="**SpatialBean**" servername="**服务器名**" versionname="**版本号**">

<SpatialBean ver="**1.0**">

<Province ver="**1.0**">

<name>省名</name>

<code>省编号</code>

</Province>

<City ver="**1.0**">

<name>市名</name>

<code>区域名称</code>

<telnum>区号</telnum>

</City>

<District ver="**1.0**">

<name>地区名称 </name>

<code>地区编码</code>

<x>x坐标</x>

<y>y坐标</y>

<bounds>bounds值</bounds>

</District>

<roadList type="**list**">

<Road ver="**1.0**">

<id>道路id</id>

<name>道路名称</name>

<direction>相对xy方向</direction>

<distance>相对距离</distance>

<width>宽度</width>

<ename>英文名称</ename>

<level>等级</level>

</Road>

</roadList>

</SpatialBean>

</spatial\_response>

* + - 1. 其他需要说明的部分
    1. 简化版位置描述
       1. 接口功能介绍

根据xy坐标返回此点的空间位置描述信息，包括该点所在省市区，周边的POI和道路。

* + - 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| ability | ability=apiserver | 服务类型 | 必添参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必添参数 |
| config | config=POSDES | 服务参数名称 | 必填参数 |
| x1 | x1=x点坐标 |  | 必填参数 |
| y1 | y1=y点坐标 |  | 必填参数 |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

* + - 1. 接口URL示例

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&config=POSDES&x1=117.3&y1=38.3&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - 1. 返回结果说明

河北省沧州市黄骅市;沧黄高速出口-205国道交叉路口;杨常庄村,于常庄村附近

* + - 1. 其他需要说明的部分
  1. 道路查询接口
     1. 根据道路名称查询道路信息
        1. 接口功能介绍

根据道路名称查询道路的位置信息。

* + - 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| config | config=SPAS | 服务参数名称 | 必填参数 |
| ability | ability=apiserver | 服务类型 | 必填参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必填参数 |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| rexy | rexy=true/false | 是否返回坐标,缺省不返回坐标 |  |
| spatialXml | spatialXml=查询条件 | spatialXml参数为一个标准的xml格式 | 必填参数 |
| resType | resType=xml | 返回格式 |  |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

**未编码前的spatialXml：**

<?xml version="1.0" encoding="GB2312"?>

<spatial\_request method="searchRoadName">

<!-- 城市号-->

<cityCode>010</cityCode>

<!-- 道路名-->

<roadName>北三环</roadName>

<!—每页记录数-->

<number>10</ number >

<!—分页号-->

< batch >1</ batch >

</spatial\_request>

* + - 1. 接口URL示例

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&rexy=true&config=SPAS&enc=utf-8&resType=xml&spatialXml=%3C%3fxml+version%3d%221.0%22+encoding%3d%22GB2312%22%3f%3E%3Cspatial\_request+method%3d%22searchRoadName%22%3E%3CcityCode%3E010%3C%2fcityCode%3E%3CroadName%3E%E5%8C%97%E4%B8%89%E7%8E%AF%3C%2froadName%3E%3CroadLevel%3E4%3C%2froadLevel%3E%3C%2fspatial\_request%3E&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - 1. 返回结果说明

<?xml version="1.0" encoding="gbk"?>

<spatial\_response type="**SpatialBean**" servername="**服务器名**" versionname="**版本号**">

<roadList type="list">

<count>

<total>

<record>

<cache>

<buffer

<Road ver="1.0">

<!--道路ID-->

<id>001011355</id>

<!--道路名称-->

<name>安德里北街</name>

<!--道路方向-->

<direction/>

<!-- 距离-- >

<distance>0.000000</distance>

<!--道路宽度-->

<width>8</width>

<!--英文名称-->

<ename>Andeli North Street</ename>

<!--道路等级-->

<level>45000</level>

<spatial\_geo type="Line">

<coor>

116.4076924,39.957157,116.406821,39.957159

</coor>

<!-- 范围-- >

<spatial\_geo type="Polygons">

<coors>

116.4076924,39.96165863,116.406821,39.957157

</coors>

</spatial\_geo>

</spatial\_geo>

</Road>

</roadList>

</spatial\_response>

* + - 1. 其他需要说明的部分
    1. 查询两条道路的交叉口位置
       1. 接口功能介绍

根据两条道路的交叉路口，查询其位置。

* + - 1. 接口参数说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** | |
| config | config=SPAS | 服务参数名称 | 必填参数 | |
| ability | ability=apiserver | 服务类型 | | 必填参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | | 必填参数 |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | | 必填参数 |
| spatialXml | spatialXml=查询条件 | spatialXml参数为一个标准的xml格式 | 必填参数 | |
| resType | resType=xml | 返回格式 |  | |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 | |

**未编码前的spatialXml：**

<?xml version="1.0" encoding="GB2312"?>

<spatial\_request method="searchRoadIntersectorByName">

<!-- 城市号-->

<cityCode>010</cityCode>

<!-- 道路名1-->

<roadName1>北四环</roadName1>

<!-- 道路名2-->

<roadName2>学院路</roadName2>

<!-- 道路等级-->

<roadLevel />

</spatial\_request>

* + - 1. 接口URL示例

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&config=SPAS&enc=utf-8&spatialXml=<?xml+version="1.0"+encoding="GB2312"?><spatial\_request+method="searchRoadIntersectorByName">+<cityCode>010</cityCode>+<roadName1>北四环</roadName1>+<roadName2>学院路</roadName2>+<roadLevel+/></spatial\_request>&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - 1. 返回结果说明

<?xml version="1.0" encoding="GBK"?>

<searchresult servername="服务器名" versionname=”版本号”>

<count>此接口返回的结果集数量（默认最大结果集为400，可通过key权限修改，需要联系管理员）</count>

<cache>此数据是否从缓存读取</cache>

<total>引擎返回的结果集总数量</total>

<record>此页返回的结果集数量（最大结果集为20）</record>

<searchtime>查询时间</searchtime>

<bounds>bounds值，当前返回的POI点统一个矩形框内，bounds为矩形框的左上右下坐标对</bounds>

<list type="**list**">

<poi>

<address>地址 </address>

<name>名称</name>

<id>poi数据id</id>

<type>数据类型</type>

<timestamp>时间戳</timestamp>

<match>匹配度</match>

<url >url</url>

<citycode>城市代码</citycode>

<xml >扩展描述信息</xml>

<x>x坐标</x>

<y>y坐标</y>

<code>区域代码</code>

<distance>距离查询点距离</distance>

<drivedistance>导航距离查询点距离</drivedistance>

<pguid>索引id</pguid>

<srctype>数据源类型</srctype>

<gridcode>gridcode代码</gridcode>

<tel>联系电话</tel>

<buscode>公交代码</buscode>

<icon>图标</icon>

<direction>距离查询点的方位</direction>

<extid>外部扩展id</extid>

<en\_type>英文类型</en\_type>

<en\_name>英文名称</en\_name>

<en\_address>英文地址</en\_address>

</poi>

</list>

</searchresult>

* + - 1. 其他需要说明的部分

**关于道路等级的说明：**

roadLevel - 道路等级,此参数目前为0－3，和国家标准的道路等级划分的对应关系如下:  
roadLevel＝0,代表的标准道路等级0-11范围的道路  
roadLevel＝1,代表的标准道路等级2-11范围的道路  
roadLevel＝2,代表的标准道路等级7-11范围的道路  
roadLevel＝3,代表的标准道路等级9-11范围的道路

**国家标准的道路等级划分共12级,参考如下：**0 图廓线 未命名道路  
1 非导航道路  
2 普通道路  
3 次要道路  
4 县乡村内部道路  
5 乡公路  
6 县道  
7 主要道路  
8 主要大街、城市快速道  
9 省道  
10 国道  
11 高速公路

* + 1. 查询一条道路上的所有交叉路口位置
       1. 接口功能介绍

查询一条道路上所有交叉路口的位置信息。

* + - 1. 接口参数说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** | |
| config | config=SPAS | 服务参数名称 | 必填参数 | |
| ability | ability=apiserver | 服务类型 | | 必填参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | | 必填参数 |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | | 必填参数 |
| spatialXml | spatialXml=查询条件 | spatialXml参数为一个标准的xml格式 | 必填参数 | |
| resType | resType=xml | 返回格式 |  | |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 | |

**未编码前的spatialXml：**

<?xml version="1.0" encoding="GB2312"?>

<spatial\_request method="searchAllIntersectionByRoadName">

<!-- 城市号-->

<cityCode>010</cityCode>

<!-- 道路名-->

<roadkey>知春路</roadkey>

<!—名称类别-->

<keyType>name</keyType>

</spatial\_request>

* + - 1. 接口URL示例

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&config=SPAS&enc=utf-8&resType=xml&spatialXml=%3C%3fxml+version%3d%221.0%22+encoding%3d%22GB2312%22%3f%3E%0A%3Cspatial\_request+method%3d%22searchAllIntersectionByRoadName%22%3E%0A+%3CcityCode%3E010%3C%2fcityCode%3E%0A+%3Croadkey%3E%E7%9F%A5%E6%98%A5%E8%B7%AF%3C%2froadkey%3E%0A+%3CkeyType%3Ename%3C%2fkeyType%3E%0A%3C%2fspatial\_request%3E&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - 1. 返回结果说明

<?xml version="1.0" encoding="GBK"?>

<searchresult servername="服务器名" versionname=”版本号”>

<count>此接口返回的结果集数量（默认最大结果集为400，可通过key权限修改，需要联系管理员）</count>

<cache>此数据是否从缓存读取</cache>

<total>引擎返回的结果集总数量</total>

<record>此页返回的结果集数量（最大结果集为20）</record>

<searchtime>查询时间</searchtime>

<bounds>bounds值，当前返回的POI点统一个矩形框内，bounds为矩形框的左上右下坐标对</bounds>

<list type="**list**">

<poi>

<address>地址 </address>

<name>名称</name>

<id>poi数据id</id>

<type>数据类型</type>

<timestamp>时间戳</timestamp>

<match>匹配度</match>

<url >url</url>

<citycode>城市代码</citycode>

<xml >扩展描述信息</xml>

<x>x坐标</x>

<y>y坐标</y>

<code>区域代码</code>

<distance>距离查询点距离</distance>

<drivedistance>导航距离查询点距离</drivedistance>

<pguid>索引id</pguid>

<srctype>数据源类型</srctype>

<gridcode>gridcode代码</gridcode>

<tel>联系电话</tel>

<buscode>公交代码</buscode>

<icon>图标</icon>

<direction>距离查询点的方位</direction>

<extid>外部扩展id</extid>

<en\_type>英文类型</en\_type>

<en\_name>英文名称</en\_name>

<en\_address>英文地址</en\_address>

</poi>

</list>

</searchresult>

* + - 1. 其他需要说明的部分
  1. 空间计算接口
     1. 位置判断接口
        1. 接口功能介绍

判断点与几何区域的相对位置（在几何区域外，在几何图形上或在几何区域内）,目前支持的几何区域包括Polygon多边形,Round圆形,Rectangle矩形,Ellipse椭圆。

* + - 1. `接口参数说明

**1、点与几何区域的位置**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| config | config=SPAS | 服务参数名称 | 必填参数 |
| ability | ability=apiserver | 服务类型 | 必填参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必填参数 |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| spatialXml | spatialXml=查询条件 | spatialXml参数为一个标准的xml格式 | 必填参数 |
| resType | resType=xml | 返回格式 |  |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

**未编码前的spatialXml：**

1，构造Rectangle（矩形）的示例:

<?xml version="1.0" encoding="GB2312"?>

<spatial\_request method="surveyByXY">

<!-- x坐标集合-->

<xs>117.1882194, 117.2044723</xs>

<!--y坐标集合-->

<ys> 36.76434461, 36.80461832 </ys>

<spatial\_geos>

<!--type类型有Circle（构造圆）、Ellipse（构造椭圆）、Rectangle（构造矩形）、对应的bounds坐标取值分别为：左下和右上坐标、坐标集合-->

<spatial\_geo type="Rectangle">

<bounds>

116.982422,36.667317,117.380676, 36.827974</bounds>

</spatial\_geo>

</spatial\_geos>

</spatial\_request>

2，构造Polygon（多边形）的示例：

<?xml version="1.0" encoding="GB2312"?>

<spatial\_request method="surveyByXY">

<!-- x坐标集合-->

<xs> 117.1882194, 117.2044723</xs>

<!-- y坐标集合-->

<ys> 36.76434461, 36.80461832 </ys>

<spatial\_geos>

<spatial\_geo type="Polygon">

<coor>

116.982422,36.667317,117.380676, 36.827974

</coor >

</spatial\_geo>

</spatial\_geos>

</spatial\_request>

* + - 1. 接口URL示例

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&config=SPAS&enc=utf-8&spatialXml=%3C%3Fxml+version%3D%221.0%22+encoding%3D%22GB2312%22%3F%3E%0D%0A%3Cspatial\_request+method%3D%22surveyByXY%22%3E%0D%0A+%3Cxs%3E117.1882194%2C117.2044723%3C%2Fxs%3E%0D%0A+%3Cys%3E36.76434461%2C36.80461832%3C%2Fys%3E%0D%0A+%3Cspatial\_geos%3E%0D%0A++%3Cspatial\_geo+type%3D%22Rectangle%22%3E%0D%0A+++%3Cbounds%3E116.982422%3B36.667317%3B117.380676%3B36.827974%3C%2Fbounds%3E%0D%0A+++%3Cbuffer%3E0%3C%2Fbuffer%3E%0D%0A++%3C%2Fspatial\_geo%3E%0D%0A+%3C%2Fspatial\_geos%3E%0D%0A%3C%2Fspatial\_request%3E&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - 1. 返回结果说明

<?xml version="1.0" encoding="GB2312" ?>

<spatial\_response type="**Result**" servername="**search1-2**">

<!-- 点在区域内-->

  <Result>INTERIOR;INTERIOR;</Result>

  </spatial\_response>

* + - 1. 其他需要说明的部分

返回结果<Result>节点对应三种情况：

INTERIOR **|** EXTERIOR |BOUNDARY（区域内|区域外|区域边界上）

* + 1. 距离计算接口
       1. 接口功能介绍

计算两点或多点之间的距离。

* + - 1. 接口参数说明

**1、单点距离计算**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| config | config=CD | 服务参数名称 | 必填参数 |
| x1 | x1=x1坐标 |  | 必填参数 |
| y1 | y1=y1坐标 |  | 必填参数 |
| x2 | x2=x2坐标 |  | 必填参数 |
| y2 | y2=y2坐标 |  | 必填参数 |
| resType | resType=XML | 返回的数据格式 |  |
| ability | ability=apiserver | 服务类型 | 必填参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必填参数 |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

**2多点距离计算**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| config | config=CDX | 服务参数名称 | 必填参数 |
| x1 | X1=x1坐标 | x1,y1,x2,y2不为空，xs,ys为空。此查询为计算两点之间的距离。 | 必填参数 |
| y1 | Y1=y1坐标 | x1,y1,x2,y2不为空，xs,ys为空。此查询为计算两点之间的距离。 | 必填参数 |
| x2 | X2=x2坐标 | x1,y1,xs,ys不为空，x2,y2为空。此查询为计算xy到最近线的距离。 | 必填参数 |
| y2 | Y2=y2坐标 | x1,y1,xs,ys不为空，x2,y2为空。此查询为计算xy到最近线的距离。 | 必填参数 |
| xs | Xs=x坐标集合 | xs,ys不为空，x1,y1,x2,y2为空。此查询为计算线的长度。 | 必填参数 |
| ys | Ys=y坐标集合 | xs,ys不为空，x1,y1,x2,y2为空。此查询为计算线的长度。 | 必填参数 |
| resType | resType=XML | 返回的数据格式 | 必填参数 |
| ability | ability=apiserver | 服务类型 | 必填参数 |
| abilityuri | abilityuri= sisserver | 请求类型 | 必填参数 |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

* + - 1. 接口URL示例

**1、单点距离计算**

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&resType=XML&enc=utf-8&config=CD&x1=116.281&y1=39.8881&x2=116.3744111&y2=39.9120711&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

**2、多点距离计算**

共有三种请求形式：

1. **计算两点之间的距离**

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&resType=XML&enc=utf-8&config=CDX&x1=116.381&y1=39.6881&x2=116.3744111&y2=39.9120711&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

1. **计算（x，y）到线的最近距离**

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&resType=XML&enc=utf-8&config=CDX&x1=116.381&y1=39.6881&xs=116.281,116.3744111&ys=39.8881,39.9120711&routeType=0&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

1. **计算线的长度**

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&resType=XML&enc=utf-8&config=CDX&xs=116.281,116.3744111&ys=39.8881,39.9120711&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - 1. 返回结果说明

**1、单点距离计算**

<?xml version="1.0" encoding="gbk" ?>

<Distance>**8411**</Distance>

**2、多点距离计算**

1. **计算两点之间的距离**

<?xml version="1.0" encoding="gbk" ?>

<distance>

<length>**24938**</length>

</distance>

1. **计算（x，y）到最近线的距离**

<?xml version="1.0" encoding="gbk" ?>

<distance>

<length>**23850**</length>

</distance>

1. **计算线的长度**

<?xml version="1.0" encoding="gbk" ?>

<distance>

<length>**8411**</length>

</distance>

* + - 1. 其他需要说明的部分
    1. 计算路径缓冲区范围
       1. 接口功能介绍

计算缓冲区范围。

* + - 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| config | config=LB | 服务参数名称 | 必填参数 |
| ability | ability=apiserver | 服务类型 | 必填参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必填参数 |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| spatialXml | spatialXml=查询条件 | spatialXml参数为一个标准的xml格式 | 必填参数 |
| resType | resType=xml | 返回格式 |  |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

**未编码前的spatialXml：**

<?xml version="1.0" encoding="GB2312"?>   
<spatial\_request method="countLineBuffer"> <!--method:请求方法-->   
    <range>500</range> <!--缓冲范围-->   
    <spatial\_geos type="list">   
       <spatial\_geo type="Line"><!--线模型，可以为多条线-->   
           <coor>116.32569,39.96682,116.12568,39.9671,116.32568,39.8671</coor> <!--线坐标串集合-->   
           <buffer>0</buffer>   
       </spatial\_geo>   
        ...   
     </spatial\_geos>   
</spatial\_request>

* + - 1. 接口URL示例

http://xxIP:xx/xxxx? ability=apiserver&abilityuri=sisserver&config=LB&enc=utf-8&spatialXml=%3Cspatial\_request+method%3D%22countLineBuffer%22%3E%3Crange%3E500%3C%2Frange%3E%3Cspatial\_geos+type%3D%22list%22%3E%3Cspatial\_geo+type%3D%22Line%22%3E%3Ccoor%3E116.32569%2C39.96682%2C116.12568%2C39.9671%2C116.32568%2C39.8671%3C%2Fcoor%3E%3Cbuffer%3E0%3C%2Fbuffer%3E%3C%2Fspatial\_geo%3E%3C%2Fspatial\_geos%3E%3C%2Fspatial\_request%3E%0A&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - 1. 返回结果说明

根据一条线路，计算出线路周围一定范围的区域

<?xml version="1.0" encoding="GB2312" ?>

<spatial\_response type="searchresult" servername="172.17.33.89">

<searchresult>

  <searchtime>0</searchtime>

  <count>2</count>

<spatial\_geos type="list">

<spatial\_geo type="Polygon">

 <coor> 116.14479430723907,39.9625735968698,116.32769230019574,39.87112460039147,116.32843879499809,39.87065468853028,116.32907927085581,39.87004817198236,116.32958911464088,39.869328358838516,116.32994873334256,39.86852291111419,116.33014430701583,39.86766278171372,116.33016831987408,39.8667810249276,116.33001984911664,39.86591152617463,116.32970460039148,39.86508769980426,116.32923468853028,39.864341205001914,116.32862817198237,39.86370072914419,116.32790835883853,39.86319088535912,116.3271029111142,39.86283126665744,116.32624278171372,39.862635692984185,116.3253610249276,39.86261168012594,116.32449152617464,39.86276015088337,116.32366769980428,39.86307539960853,116.12366769980427,39.963075399608535,116.12293889779971,39.963531650365184,116.12231018804356,39.964118200582384,116.12180452807931,39.96481363218604,116.12144038224692,39.96559255126242,116.12123104745119,39.966426515325246,116.12118416762105,39.96728507190248,116.12130145458892,39.96813687051833,116.12157862558257,39.96895080746618,116.12200555961205,39.96969716157066,116.1225666670408,39.97034867946618,116.12324145884655,39.970881570762494,116.12400529478457,39.971276376758645,116.12483028313433,39.97151868098384,116.12568629917492,39.9715996356196,116.32569629917491,39.971319635619594,116.32657401350001,39.97123194747482,116.32741775568206,39.97097471066283,116.3281951011822,39.97055781065006,116.32887617704877,39.96999726867019,116.3294348099182,39.96931462603716,116.32984953184257,39.96853611632453,116.33010440529094,39.96769165722395,116.33018963561959,39.96681370082509,116.33010194747482,39.965935986499986,116.32984471066283,39.96509224431793,116.32942781065005,39.9643148988178,116.32886726867018,39.963633822951216,116.32818462603716,39.963075190081796,116.32740611632452,39.96266046815742,116.32656165722395,39.962405594709054,116.32568370082508,39.9623203643804,116.14479430723907,39.9625735968698

</coor>

  <buffer>0</buffer>

  </spatial\_geo>

<spatial\_geo type="Polygon">

 <coor> 116.14479430723907,39.9625735968698,116.32769230019574,39.87112460039147,116.32843879499809,39.87065468853028,116.32907927085581,39.87004817198236,116.32958911464088,39.869328358838516,116.32994873334256,39.86852291111419,116.33014430701583,39.86766278171372,116.33016831987408,39.8667810249276,116.33001984911664,39.86591152617463,116.32970460039148,39.86508769980426,116.32923468853028,39.864341205001914,116.32862817198237,39.86370072914419,116.32790835883853,39.86319088535912,116.3271029111142,39.86283126665744,116.32624278171372,39.862635692984185,116.3253610249276,39.86261168012594,116.32449152617464,39.86276015088337,116.32366769980428,39.86307539960853,116.12366769980427,39.963075399608535,116.12293889779971,39.963531650365184,116.12231018804356,39.964118200582384,116.12180452807931,39.96481363218604,116.12144038224692,39.96559255126242,116.12123104745119,39.966426515325246,116.12118416762105,39.96728507190248,116.12130145458892,39.96813687051833,116.12157862558257,39.96895080746618,116.12200555961205,39.96969716157066,116.1225666670408,39.97034867946618,116.12324145884655,39.970881570762494,116.12400529478457,39.971276376758645,116.12483028313433,39.97151868098384,116.12568629917492,39.9715996356196,116.32569629917491,39.971319635619594,116.32657401350001,39.97123194747482,116.32741775568206,39.97097471066283,116.3281951011822,39.97055781065006,116.32887617704877,39.96999726867019,116.3294348099182,39.96931462603716,116.32984953184257,39.96853611632453,116.33010440529094,39.96769165722395,116.33018963561959,39.96681370082509,116.33010194747482,39.965935986499986,116.32984471066283,39.96509224431793,116.32942781065005,39.9643148988178,116.32886726867018,39.963633822951216,116.32818462603716,39.963075190081796,116.32740611632452,39.96266046815742,116.32656165722395,39.962405594709054,116.32568370082508,39.9623203643804,116.14479430723907,39.9625735968698

</coor>

  <buffer>0</buffer>

  </spatial\_geo>

  </spatial\_geos>

  </searchresult>

  </spatial\_response>

* + - 1. 其他需要说明部分
    1. 面积计算接口
       1. 接口功能介绍

给定几个点的经纬度坐标，计算这几个点围成的面积

* + - 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| sid | sid=18000 | 服务参数码 | 必填参数 |
| lnglats | lnglats=坐标的经纬度 | 参与面积计算的坐标的经纬度，经纬度之间用” ,”逗号分割，每个经纬度之间用” ;”分号分割，经度在前，维度在后 | 必填参数 |
| resType | resType=xml或者json | 返回的数据格式 | 必选参数 |
| ability | ability=apiserver | 服务类型 | 必填参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必填参数 |
| encode | encode=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| key | key=api key | 用户申请的key | 必填参数 |

* + - 1. 接口URL实例

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=measure/simple&enc=utf-8&sid=18000&key=[用户KEY]&resType=xml&lnglats=116.40701293945312,39.91776787232845;116.40323638916016,39.899464899384725;116.41199111938477,39.899464899384725

* + - 1. 返回结果说明

给定几个点的经纬度坐标，返回计算出的面积值

<measure>

<lnglats>

116.40701293945312,39.91776787232845;116.40323638916016,39.899464899384725;116.41199111938477,39.899464899384725

</lnglats>

<result>

<value>761673.9296875</value>

<module>平方米</module>

</result>

</measure>

* + - 1. 其他需要说明的部分

Module有两种一种是平方公里，一种是平方米

* 1. 坐标处理接口
     1. 经纬度与屏幕像素坐标转换
        1. 接口功能

经纬度与屏幕像素坐标的相互转换

* + - 1. 接口参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 参数格式 | 参数含义 | 备注 |
| config | config=PLL | 服务参数名称 | 必填参数 |
| ability | ability=apiserver | 服务类型 | 必填参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必填参数 |
| xs | xs=精度坐标 | 精度坐标支持传入多个参数，中间用“,“分割。 |  |
| ys | ys=维度坐标 | 维度坐标支持传入多个参数，中间用“,”分割。经纬度坐标成对出现 | 当经纬度坐标转屏幕像素坐标时xs,ys不能为空，screenXs,screenYs为空。 |
| screenXs | screenXs=屏幕像素x坐标 | 屏幕像素x坐标支持传入多个参数，中间用“,”分割。 |  |
| screenYs | screenYs=屏幕像素y坐标 | 屏幕像素y坐标支持传入多个参数，中间用“,”分割。 | 当屏幕像素坐标转经纬度坐标时screenXs,screenYs不能为空，xs,ys为空。 |
| zoom | zoom=zoom 级别 | zoom级别 | 必填参数 |
| width | width=地图像素长度 | 地图像素长度 | 必填参数 |
| height | height=地图像素宽度 | 地图像素宽度 | 必填参数 |
| centerLng | centerLng=地图中心点精度坐标 | 地图中心点精度坐标 | 必填参数 |
| centerLat | centerLat=地图中心点纬度坐标 | 地图中心点纬度坐标 | 必填参数 |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=api key | 用户申请的key | 必填参数 |

* + - 1. 接口URL示例

1. 经纬度坐标转**屏幕**像素坐标

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&config=PLL&resType=xml&xs=116.3604213,116.4604213&ys=39.3204703,39.4204703&zoom=15&width=1000000&height=500000&centerLng=116.313031&centerLat=39.872196&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

1. 屏幕像素坐标转经纬度坐标

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&config=PLL&resType=xml&screenXs=501104.0,503435.0&screenYs=266685.0,263670.0&zoom=15&width=1000000&height=500000&centerLng=116.313031&centerLat=39.872196&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + - 1. 返回结果说明

1. 经纬度坐标转屏幕像素坐标

<?xml version="1.0" encoding="GBK" ?>

<searchresult servername="**localhost.localdomain**">

<list>

<screenPoint>

<screenX>**501104.0**</screenX>

<screenY>**266685.0**</screenY>

</screenPoint>

<screenPoint>

<screenX>**503435.0**</screenX>

<screenY>**263670.0**</screenY>

</screenPoint>

</list>

</searchresult>

1. 屏幕像素坐标转经纬度坐标

<?xml version="1.0" encoding="GBK" ?>

<searchresult servername="**localhost.localdomain**">

<list>

<lnglatPoint>

<lng>**116.36040687561035**</lng>

<lat>**39.32045443611561**</lat>

</lnglatPoint>

<lnglatPoint>

<lng>**116.46044254302979**</lng>

<lat>**39.4204805386993**</lat>

</lnglatPoint>

</list>

</searchresult>

* + - 1. 其他需要说明的部分
  1. WMAP取图接口
     1. 接口功能介绍
     2. 接口参数说明

**1、画点**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数名称** | **参数格式** | | **参数含义** | **备注** |
| config | config=WMAP | | 服务参数名称 | 必填参数 |
| ability | | ability=apiserver | 服务类型 | 必填参数 |
| abilityuri | | abilityuri=sisserver | 请求类型 | 必填参数 |
| xs | xs=x坐标串，以逗号分隔 | |  | 必填参数 |
| ys | ys =y坐标串，以逗号分隔 | |  | 必填参数 |
| names | names=名称坐标串，以逗号分隔 | |  | 必填参数 |
| content | content=内容展现形式 | | 包括URL/Base64/MAP/URLDATA,注意大小写,默认URL | 必填参数 |
| cenX | cenX =地图中心点x | |  |  |
| cenY | cenY=地图中心点 y | |  |  |
| width | width=地图宽度，默认240 | |  |  |
| height | height=地图高度，默认240 | |  |  |
| colors | colors=颜色，默认256 | |  |  |
| format | format=输出格式图片格,包括png/gif/jpg，,注意大小写，默认png | |  |  |
| maplevel | maplevel=地图等级，如果不传，默认根据图上的加入的poi和线，自适应大小 | | 地图级别0~17 |  |
| px | px=x轴偏移方向，东移为正，西移为负，单位1为1/4的图片宽度，例如px=1，等于地图向右移动1/4的长度 | |  |  |
| py | py= y轴偏移方向，北移为正，南移为负，单位1为1/4的图片高度，例如px=1，等于地图向上移动1/4 | |  |  |
| font | font=字体，字体风格，字号，字体颜色，字体背景，是否有背景色 | | **font=**宋体,1,20,0xff0cc0,0xffffe0,1  含义对应关系如下：  宋体---- 字体  1---字体风格  20---字号  0xff0cc0-----字体颜色  0xffffe0-----字体背景  1------是否有背景色，0无，1有 |  |
| icons | icons=定义的图片名 | | 用来表示需要加点的tip图片名，每个点之间的图片用逗号分隔和names参数和xs，ys参数顺序对应描述一个点,例如文件名为 15.gif ，则调用的时候 icons=15。用户自定义图片需上传至服务器。 |  |
| traffic | traffic=on/off | | 是否支持实时交通图层 |  |
| showLogo | showLogo=true/false | | 是否展现maplogo图标，默认为true，展示maplogo图标 |  |
| nobase | nobase=off/on | | 默认为off，如果为on，代表只返回实时交通图层，不显示基础图层 |  |
| site | site=图标位置 | | 范围1-9, 1为图标位于点的左上位置，2为正上，以此类推，9为右下，默认为5，即正中。 |  |
| language | language=cn/en | | 底图语言，默认为中文 |  |
| region | region=左下角，右上角坐标 | | 设定取图的区域范围，坐标为取图的左下角和右上角坐标，坐标用逗号隔开，设定该坐标后地图中心点参数无效。返回为包含该区域的最大级别图片。 |  |
| enc | enc=utf-8 | | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=apiKey | | 用户申请的key | 必填参数 |

**2、自定义画点**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| config | config=WMAP | 服务参数名称 | 必填参数 |
| ability | ability=apiserver | 服务类型 | 必填参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必填参数 |
| xs | xs=x坐标串，以逗号分隔 |  | 必填参数 |
| ys | ys =y坐标串，以逗号分隔 |  | 必填参数 |
| names | names=名称坐标串，以逗号分隔 |  | 必填参数 |
| cenX | cenX =地图中心点x |  |  |
| cenY | cenY=地图中心点 y |  |  |
| width | width=地图宽度，默认240 |  |  |
| height | height=地图高度，默认240 |  |  |
| colors | colors=颜色，默认256 |  |  |
| format | format=输出格式图片格,包括png/gif/jpg，,注意大小写，默认png | **font=**宋体,1,20,0xff0cc0,0xffffe0,1  含义对应关系如下：  宋体---- 字体  1---字体风格  20---字号  0xff0cc0-----字体颜色  0xffffe0-----字体背景  1------是否有背景色，0无，1有 |  |
| maplevel | maplevel=地图等级，如果不传，默认根据图上的加入的poi和线，自适应大小 | 地图级别0~17 |  |
| px | px=x轴偏移方向，东移为正，西移为负，单位1为1/4的图片宽度，例如px=1，等于地图向右移动1/4的长度 |  |  |
| py | py=y轴偏移方向，北移为正，南移为负，单位1为1/4的图片高度，例如px=1，等于地图向上移动1/4 |  |  |
| font | font=字体，字体风格，字号，字体颜色，字体背景，是否有背景色 | **font=**宋体,1,20,0xff0cc0,0xffffe0,1  含义对应关系如下：  宋体---- 字体  1---字体风格  20---字号  0xff0cc0-----字体颜色  0xffffe0-----字体背景  1------是否有背景色，0无，1有 |  |
| icons | icons=定义的图片名 | 用来表示需要加点的tip图片名，每个点之间的图片用逗号分隔和names参数和xs，ys参数顺序对应描述一个点,例如文件名为 15.gif ，则调用的时候 icons=15。用户自定义图片需上传至服务器。 |  |
| traffic | traffic=on/off | 是否支持实时交通图层 |  |
| showLogo | showLogo=true/false | 是否展现maplogo图标，默认为true，展示maplogo图标 |  |
| nobase | nobase=off/on | 默认为off，如果为on，代表只返回实时交通图层，不显示基础图层 |  |
| region | region=左下角，右上角坐标 | 设定取图的区域范围，坐标为取图的左下角和右上角坐标，坐标用逗号隔开，设定该坐标后地图中心点参数无效。返回为包含该区域的最大级别图片。 |  |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=apiKey | 用户申请的key | 必填参数 |

**3、多点画线**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| config | config=WMAP | 服务参数名称 | 必填参数 |
| ability | ability=apiserver | 服务类型 | 必填参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必填参数 |
| lines | lines=线的颜色值，线的宽度，拐点的x，y组合 | 1. 线的颜色值：默认为0X00000000（可以为空，传则必须为0x或者0X打头的，8位16进制的数字字符串 ）。在原有6位颜色基础上增加了2位透明度位0X12345678，其中12位用来设置线的透明度，FF为不透明00为全透明，345678位为RGB颜色位。（例0X80FF0000代表半透明的红色线条） 2. 线的宽度：默认为3（可以为空，传则必须为数字字符串） 3. 此线的拐点的x，y组合，必须配对，添加多组线则以分号分隔即可 |  |
| xs | xs=添加图标的x坐标串，以逗号分隔 |  |  |
| ys | ys =添加图标的y坐标串，以逗号分隔 |  |  |
| icons | icons=自定义图片名 | 用来表示需要加点的tip图片名，每个点之间的图片用逗号分隔和names参数和xs，ys参数顺序对应描述一个点,例如文件名为 15.gif ，则调用的时候 icons=15。用户自定义图片需上传至服务器。 |  |
| content | content=内容展现形式 | 包括URL/Base64/MAP/URLDATA,注意大小写,默认URL | 必填参数 |
| cenX | cenX =地图中心点x |  |  |
| cenY | cenY=地图中心点 y |  |  |
| width | width=地图宽度，默认240 |  |  |
| height | height=地图高度，默认240 |  |  |
| colors | colors=颜色，默认256 |  |  |
| format | format=输出格式图片格,包括png/gif/jpg，,注意大小写，默认png |  |  |
| maplevel | maplevel=地图等级，如果不传，默认根据图上的加入的poi和线，自适应大小 | 地图级别0~17 |  |
| px | px=x轴偏移方向，东移为正，西移为负，单位1为1/4的图片宽度，例如px=1，等于地图向右移动1/4的长度 |  |  |
| py | py=y轴偏移方向，北移为正，南移为负，单位1为1/4的图片高度，例如px=1，等于地图向上移动1/4 |  |  |
| traffic | traffic=on/off | 是否支持实时交通图层 |  |
| showLogo | showLogo=true/false | 是否展现maplogo图标，默认为true，展示maplogo图标 |  |
| nobase | nobase=off/on | 默认为off，如果为on，代表只返回实时交通图层，不显示基础图层 |  |
| midArrow | midArrow=3 | 在两点连线的中间加箭头，箭头位置为 midArrow数值点和前一点连线中间位置。 |  |
| endArrow | endArrow=4 | 在两点连线的末尾加箭头，箭头位置为 endArrow数值点位置。 |  |
| lineStyle | lineStyle=DASHED | 线类型，支持画线为实线或虚线，默认为实线，SOLID为实线，DASHED为虚线 |  |
| region | region=左下角，右上角坐标 | 设定取图的区域范围，坐标为取图的左下角和右上角坐标，坐标用逗号隔开，设定该坐标后地图中心点参数无效。返回为包含该区域的最大级别图片。 |  |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=apiKey | 用户申请的key | 必填参数 |

**4、多点画面**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数格式** | **参数含义** | **备注** |
| config | config=WMAP | 服务参数名称 | 必填参数 |
| ability | ability=apiserver | 服务类型 | 必填参数 |
| abilityuri | abilityuri=sisserver | 请求类型 | 必填参数 |
| polygons | polygon=画边框的颜色值，画边框的宽度，画填充色，画填充的透明度，此面的拐点x，y组合 | 1. 面边框的颜色值：默认为0X00000000（可以为空，传则必须为0x或者0X打头的，8位16进制的数字字符串 ）。可以传6位的RGB颜色值，也可以传8位带透明度的RGBA颜色值， 前2位为透明度位。0X12345678，其中12位用来设置线的透明度，FF为不透明00为全透明，345678位为RGB颜色位。（例0X80FF0000代表半透明的红色线条） 2. 面边框的宽度：默认为1（可以为空，传则必须为小于50的数字字符串，可以为0） 3. 面填充色：同面边框的颜色值，默认为0XFF0000 4. 面填充的透明度：默认为50（0为全透明，100为全不透明） 5. 此面的拐点的x，y组合（逗号分隔），必须配对   添加多组面则以分号分隔即可。 |  |
| content | content=内容展现形式 | 包括URL/Base64/MAP/URLDATA,注意大小写,默认URL | 必填参数 |
| cenX | cenX =地图中心点x |  |  |
| cenY | cenY=地图中心点 y |  |  |
| width | width=地图宽度，默认240 |  |  |
| height | height=地图高度，默认240 |  |  |
| colors | colors=颜色，默认256 |  |  |
| format | format=输出格式图片格,包括png/gif/jpg，,注意大小写，默认png |  |  |
| maplevel | maplevel=地图等级，如果不传，默认根据图上的加入的poi和线，自适应大小 | 地图级别0~17 |  |
| px | px=x轴偏移方向，东移为正，西移为负，单位1为1/4的图片宽度，例如px=1，等于地图向右移动1/4的长度 |  |  |
| py | py=y轴偏移方向，北移为正，南移为负，单位1为1/4的图片高度，例如px=1，等于地图向上移动1/4 |  |  |
| traffic | traffic=on/off | 是否支持实时交通图层 |  |
| showLogo | showLogo=true/false | 是否展现maplogo图标，默认为true，展示maplogo图标 |  |
| nobase | nobase=off/on | 默认为off，如果为on，代表只返回实时交通图层，不显示基础图层 |  |
| region | region=左下角，右上角坐标 | 设定取图的区域范围，坐标为取图的左下角和右上角坐标，坐标用逗号隔开，设定该坐标后地图中心点参数无效。返回  为包含该区域的最大级别图片。 |  |
| fillStyle | fillStyle=样式 | 设置填充样式，默认为color，可选样式：color，line，slash，grid，triangle |  |
| enc | enc=utf-8 | 编码设定，仅支持utf-8 | 必填参数 |
| a\_k | a\_k=apiKey | 用户申请的key | 必填参数 |

* + 1. 接口URL示例

**1、画点**

[http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&config=WMAP&xs=102.728904,102.795816&ys=25.048943,25.027231&names=a,b&content=map](http://search3.mapabc.com/sisserver?ability=apiserver&abilityuri=sisserver&omp_appid=000702000000000054&omp_restype=res13&config=WMAP&xs=102.728904,102.795816&ys=25.048943,25.027231&names=a,b&content=map)&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

**2、自定义画点**

[http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&config=WMAP&xs=102.728904,102.795816&ys=25.048943,25.027231&names=a,b&icons=105,105&content=map](http://search3.mapabc.com/sisserver?ability=apiserver&abilityuri=sisserver&omp_appid=000702000000000054&omp_restype=res13&config=WMAP&xs=102.728904,102.795816&ys=25.048943,25.027231&names=a,b&icons=105,105&content=map)&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

**3、多点画线**

[http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&config=WMAP&lines=0x80ff0000,3,123.460769,41.720825,123.454712,41.723336&content=map&xs=123.460769,123.454712&ys=41.720825,41.723336&icons=a\_0,a\_1&midArrow=2&a\_k=[用户KEY]](http://lngis.cmccnavi.com:11111/sisserver?ability=apiserver&abilityuri=sisserver&omp_appid=000702000000000054&omp_restype=res13&config=WMAP&lines=0x80ff0000,3,123.460769,41.720825,123.454712,41.723336&content=map&xs=123.460769,123.454712&ys=41.720825,41.723336&icons=a_0,a_1&midArrow=2&a_k=89f1f3f54e2400893d397b632fd8be26a315448f3a2dc677d8e85ac856541744e288d504882feb36)

**注：此URL仅为示例使用，不是最终服务提供URL**

**4、多点画多边形**

http://xxIP:xx/xxxx?ability=apiserver&abilityuri=sisserver&config=WMAP&polygons=0x000000,1,1,50,102.795816,25.027231,102.728904,25.048943,102.738219,25.0531675,102.730465,25.035184&content=map&fillStyle=line&a\_k=[用户KEY]

**注：此URL仅为示例使用，不是最终服务提供URL**

* + 1. 返回结果说明

**1、画点**



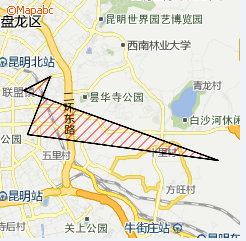
**2、自定义画点**



**3、多点画线**



**4、多点画面**



JSON数据格式说明

JSON格式的返回与结果与XML格式结果对应，JSON返回结果参考XML格式说明解析即可。

JSON格式与XML对应关系如下：

**XML格式说明：**

<?xml version="1.0" encoding="GBK"?>

<searchresult servername="服务器名" versionname=”版本号”>

<count>此接口返回的结果数量</count>

<cache>是否从缓存读取</cache>

<total>结果总数量</total>

<record>此页返回的结果数量</record>

<searchtime>查询时间</searchtime>

<bounds>bounds值</bounds>

<list type="list">

<poi>

  <address>地址</address>

  <name>名称</name>

  <type>数据类型</type>

  <timestamp>时间戳</timestamp>

  <match>匹配度</match>

  <url >url</url>

  <citycode>城市代码</citycode>

  <xml >扩展描述信息</xml>

  <x>x坐标</x>

  <y>y坐标</y>

  <code>区域代码</code>

  <drivedistance>导航距离查询点距离</drivedistance>

  <pguid>索引id</pguid>

  <srctype>数据源类型</srctype>

  <gridcode>gridcode代码</gridcode>

  <tel>联系电话</tel>

  <buscode>公交代码</buscode>

  <imageid>图片id</imageid>

  <linkid>关联id</linked>

  <newtype>数据类型编码</newtype>

</poi>

</list>

</searchresult>

**对应的json格式：**

{

"count":"此接口返回的结果数量",

"cache":"是否从缓存读取",

"total":"结果总数量",

"record":"此页返回的结果数量",

"searchtime":"查询时间",

"bounds":"bounds值",

"poilist":[

{

"address":"地址",

"name":"名称",

"type":"数据类型",

"timestamp":"时间戳",

"match":"匹配度",

"url":"url",

"citycode":"城市代码",

"xml":"扩展描述信息",

"x":"x坐标",

"y":"y坐标",

"code":"区域代码",

"drivedistance":"距离查询点导航距离",

"pguid":"索引id",

"srctype":"数据源类型",

"gridcode":"gridcode代码",

"tel":"联系电话",

"buscode":"公交代码",

"imageid":"图片id",

"linkid":"关联id",

"newtype":"数据类型编码",

}

],

}