

中交兴路物流 aPaaS 产品白皮书

模板编号	中交兴路车联网—智运
文档版本号	华为云版 V1

北京中交兴路车联网科技有限公司

2022 年 11 月 16 日

目录

1. 概述	3
2. 适用对象	3
3. 产品功能服务详细介绍	3
3.1.1. 运输节点服务	3
3.1.2. 必经地/禁行地管理	6
3.1.3. 运输行程服务	8
3.1.4. 运输中转服务	15
3.1.5. 运输规划服务	17
3.1.6. 晚点预警服务	21
3.1.7. 目的地-实时通知服务	23
3.1.8. 起始地-实时通知服务	28
3.1.9. 行驶报警服务	33
3.1.10. 货物失联提醒服务	38
3.1.11. 区域内停车提醒服务	40
3.1.12. 偏航预警服务	44
3.1.13. 货车城市溯源服务	50
3.1.14. 中途停车服务	51
3.1.15. 行驶里程服务	52
3.1.16. 车辆完整信息验证服务	54
3.1.17. 智能电子围栏 v1.0	58
3.1.18. 历史业务质量分析服务	63
4. 技术服务支持	65
4.1. 术语和定义	65
4.2. 产品服务技术联调流程	66
4.2.1. 第一步：获取账密	66
4.2.2. 第二步：编写服务	66
4.2.3. 第三步：联调测试	66
4.3. 调用要求	66
4.3.1. 服务调用方式	66
4.3.2. 用户认证和服务安全	67
5. 常见问题	68
附录 A：状态码说明	69
附录 B：Java 示例代码	71

1. 概述

中交兴路携手 3000+生态合作伙伴，依托“人、车、货、企”海量物流大数据，共创物流生态数智化升级；结合自身在北斗时空应用、物流 AI、物联网等领域的专精优势打造物流数字化首个 AaaS 开放平台，率先拓展了物流运输可视化、数字化厂区物流协同、一站式货物跟踪、时空网格运输节点为代表的多个物流主场景，为物流行业的降本增效，低碳运行赋能。

2. 适用对象

适用于客户企业的运营，产品，研发等部门。

适用于实体物流行业的大宗货运，快递快运，大件运输，车队等；

适用于物流平台类企业，如网络货运，物流平台等；

适用于科技信息企业如软件和信息技术服务业，科技信息类等；

适用于批发零售，环保资源等；

适用于银行，金融，保险等企业；

适用于钢铁行业，农业林业等行业运输企业。

3. 产品功能服务详细介绍

3.1.1. 运输节点服务

3.1.1.1. 产品典型场景

物流场景下，在运输前承运方需要协调或调度车辆、制定运输计划，运输中需要关注货物运输进程，了解运输在途情况实现过程管理，传统人工的方式费事费力。

运输节点服务实现运输全程帮助调度人员掌握运输实时动态，提供基于数据的运单运输节点分析，提供及时准确的运单运输情况

通过对实时计算引擎批量分析结合地图导航算法，提供及时准确的运距以及预计到达时间。

3.1.1.2. 产品功能说明

基于实时计算引擎，批量分析目的地相同的运输行为的剩余时间及距离。根据输入的车牌号，车牌颜色，始发地坐标，目的地坐标或者始发地行政区划编码，目的地行政区划编码，返回计算后的预计到达时间，已行驶距离，剩余运距等信息。

● API 服务说明

通过 API 的方式对接，根据传入参数，系统自动计算后，按返回参数，通过 API 的方式回传给客户。批量最大不超过 100 个。

接口整体技术服务支持，[跳转](#)。

取值方式：行政区划编码为传入值，没有传入目的地和起始地坐标时，取行政区划默认坐标作为起点或者终点。（默认坐标是地图默认的该地区行政机关所在地坐标）

业务规则：

- 1) 预计到达时间：从运输工具当前所在地到目的地的预计时间。
- 2) 剩余运距：运输工具当前所在地到目的地的剩余距离。
- 3) 已行驶距离：从始发地到当前位置的行驶距离。

3.1.1.3. API 对接说明

3.1.1.3.1. 运输节点服务

- 对接地址

请求地址格式 </save/apis/transTimeManageV3>

- 输入参数列表

参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识（API 访问凭证），对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
vnos	车牌号_车牌颜色，多个车辆以英文逗号分隔（最多 100 台车，颜色：1 蓝色、2 黄色、3 黄绿）	京 A78601_1, 京 A78602_2, 京 A78603_3	是
startLonlat	始发地坐标 (WGS84 坐标系)	116. 31795, 30. 4252	否，始发地坐标和目的地坐标为一组，始发地行政区划编码和目的地行政区划编码为一组，都输入的时候以始发地坐标、目的地坐标为准。
endLonlat	目的地坐标 (WGS84 坐标系)	110. 9946817, 25. 9089816	
startAreaCode	始发地行政区划编码（区县编码）	431102	
endAreaCode	目的地行政区划编码（区县编码）	450324	

- 输出参数列表

参数	说明
result	查询结果
status	返回状态码：详见附录 A

接口返回数据解码后样例：

```
{
  "result": {
    "firstVcl": {
      "adr": "河南省洛阳市涧西区入口, 西北方向, 132.0 米",
      "city": "洛阳市",
      "country": "涧西区"
    }
  }
}
```

```

    "drc": "0", "estimateArriveTime": "1661538106278", "lat": "20788210", "lon": "67406467", "mil": "91410.7", "offlineState": true, "offlineTime": "29", "province": "河南省", "remainDistance": "1181.6", "runDistance": "0.0", "spd": "0.0", "status": "1001", "utc": "1661477671000", "vno": "豫 CG5239"}, {"others": [{"adr": "河北省唐山市滦南县井离庄村, 东南方向, 530.1 米", "city": "唐山市", "country": "滦南县", "drc": "0", "lat": "23616055", "lon": "71213647", "mil": "396632.1", "province": "河北省", "spd": "0.0", "status": "1001", "utc": "1661479441000", "vno": "冀 B2229U"}, {"adr": "江西省南昌市南昌县富山一路东口 18 号中铁物流, 东南方向, 83.0 米", "city": "南昌市", "country": "南昌县", "drc": "0", "lat": "17131345", "lon": "69540358", "mil": "44108.7", "province": "江西省", "spd": "0.0", "status": "1001", "utc": "1645969947000", "vno": "赣 AB0091"}, {"adr": "", "city": "", "country": "", "drc": "", "lat": "", "lon": "", "mil": "", "province": "", "spd": "", "status": "1006", "utc": "", "vno": "陕 YH0009"}, {"adr": "", "city": "", "country": "", "drc": "", "lat": "", "lon": "", "mil": "", "province": "", "spd": "", "status": "1002", "utc": "", "vno": "皖 K3D8329"}]}, {"status": "1001"}

```

注：多车返回时，添加了 1002 状态码的优化。原有功能是多车中有一个 1002 状态，全部以 1002 状态返回；优化后，仅有车牌号错误的车辆以 1002 状态返回，其他各车正常检索数据，正常返回。

数据体详细字段说明：

输出参数名称	输出参数字段	示例数据	备注
首车查询结果	firstVcl		JSON 对象（详见下字段说明）
车牌号	vno	京 AB0001	
状态码	status	1001	业务查询状态：详见附录 A
车辆最后定位经度	lon	19356858	格式：1/600000.0(WGS84 坐标系)
车辆最后定位纬度	lat	71847300	格式：1/600000.0(WGS84 坐标系)
车辆地理位置名称	adr	江苏省镇江市扬中市	
车辆定位时间	utc	1635497914000	时间戳
速度	spd	39.1	单位 km/h
方向	drc	231	正北，大于 0 且小于 90：东北，等于 90：正东，大于 90 且小于 180：东南，等于 180：正南，大于 180 且小于 270：西南，* 等于 270：正西，大于 270 且小于等于 359：西北，其他：未知
省	province	江苏省	
市	city	镇江市	
县	country	扬中市	
里程	mil	0.0	单位：km
离线状态	offlineState	true	true:离线， false:在线

离线时长	offlineTime	5422	单位：分钟
车辆已行驶里程	runDistance	724.0	单位：km
剩余运距	remainDistance	54.1	单位：km
预计到达时间	estimateArriveTime	1626067123000	
其余车辆列表	others		JSON 对象列表（详见下字段说明）
车牌号	vno	京 AB0001	
状态码	status	1001	业务查询状态：详见附录 A
车辆最后定位经度	lon	15545388	格式：1/600000.0(WGS84 坐标系)
车辆最后定位纬度	lat	66596808	格式：1/600000.0(WGS84 坐标系)
车辆地理位置名称	adr	广西壮族自治区桂林市全州县	
车辆定位时间	utc	1635238617000	时间戳
速度	spd	68.0	单位 km/h
方向	drc	217	正北，大于 0 且小于 90：东北，等于 90：正东，大于 90 且小于 180：东南，等于 180：正南，大于 180 且小于 270：西南，* 等于 270：正西，大于 270 且小于等于 359：西北，其他：未知
省	province	广西壮族自治区	
市	city	桂林市	
县	country	全州县	
里程	mil	4003.0	单位：千米

3.1.2. 必经地/禁行地管理

3.1.2.1. 产品典型场景

监控物流运输场景下串货频发地的进出情况，通过必经地/禁行地服务能复盘高货值物品运输过程中在必经地的停车情况。

根据车辆行驶过程进行精准判断，快速且直观的查看到运输过程中是否进出过串货频发地。

3.1.2.2. 产品功能说明

支持自定义运输必经地/禁行地，基于实时计算引擎计算分析异常预警，规避运输风险。根据输入的车牌号，车牌颜色，回溯行程的时间，禁行地/必经地围栏列表（圆形围栏中心点位置和半径），返回计算后的围栏，进出围栏类型和进出围栏时间。

● API 服务说明

通过 API 的方式对接，根据传入参数，系统自动计算后，按返回参数，通过 API 的方式回传给客户。

接口整体技术服务支持，[跳转](#)。

业务规则：

- 1) 围栏坐标：提供圆形围栏中心点坐标，不支持多边形围栏。
- 2) 围栏半径：围栏半径范围最小 1 公里，最大 5 公里。
- 3) 回溯行程时间：给出回溯最长历史行程，目前支持 72 小时。

3.1.2.3. API 对接说明

3.1.2.3.1. 必经地/禁行地管理服务

- 对接地址

请求地址格式 </save/apis/groundPasses>

- 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识 (API 访问凭证)， 对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
vno	车牌号	陕 YH0009	是
vco	车牌颜色 (1 蓝色、2 黄色、3 黄绿色)	2	是
timeNearby	时间范围，单位：小时，指返回车辆最近时间范围内的最后定位信息	20 (注：单位为小时，结合需要实际使用情况动态设置)	是
areaArray	围栏列表 (数组) 字符串		最多三个动态围栏
serialNum	围栏序号	1	否，围栏取值 1, 2, 3 (依次传入，正整数)
centerPoint	围栏中心点 (WGS84 坐标系)	116.287149, 39.858427	否，经纬度坐标，格式：lon, lat
radius	围栏半径	3000	否，范围 [1000, 5000]，单位：米

接口传参数数据解析后样例：

```
{
  "vco": "2",
  "areaArray": [
    {
      "serialNum": 1,
      "centerPoint": "119.464817, 35.45284",
      "radius": 1852
    },
    {
      "serialNum": 2,
      "centerPoint": "119.503613, 35.510011",
      "radius": 1027
    },
    {
      "serialNum": 3,
      "centerPoint": "119.524899, 35.532925",
      "radius": 1041
    }
  ],
  "vno": "XXX",
  "sign": "XXX",
  "timeNearby": "24",
  "cid": "XXX",
  "token": "XXX"
}
```

- 输出参数列表

参数	说明
result	查询结果
status	返回状态码：详见附录 A

接口返回数据解码后样例：

```
{
  "result": {
    "country": "渝北区",
    "city": "重庆市",
    "utc": "1628485611000",
    "spd": "0.0",
    "lon": "64007815",
    "adr": "重庆市渝北区铁家湾, 东北方向, 229.0 米",
    "mil": "100600.8",
    "drc": "270",
    "province": "重庆市",
    "areaArray": [
      {
        "serialNum": "1",
        "inOutTime": "1628486013000",
        "inOutType": "1"
      }
    ],
    "isInside": "0",
    "lat": "17855836"
  },
  "status": 1001
}
```

数据体详细字段说明

输出参数名称	输出参数字段	示例数据	备注
是否在围栏内部	isInside	0	1-是 0-否
纬度	lat	17855836	格式：1/600000.0(WGS84 坐标系)
经度	lon	64007815	格式：1/600000.0(WGS84 坐标系)
车辆地理位置名称	adr	重庆市渝北区铁家湾, 东北方向, 229.0 米	
车辆定位时间	utc	1628485611000	时间戳
速度	spd	62.0	单位：千米/小时
方向	drc	215	正北，大于 0 且小于 90：东北，等于 90：正东，大于 90 且小于 180：东南，等于 180：正南，大于 180 且小于 270：西南，* 等于 270：正西，大于 270 且小于等于 359：西北，其他：未知
省	province	重庆市	
市	city	重庆市	
区县	country	渝北区	
上报里程	mil	123.0	单位：千米
若“围栏列表”入参为空，则以下信息返回为空			
围栏列表	areaArray		
围栏编号	serialNum	1	对应入参围栏序号
进出围栏时间	inOutTime	1628486013000	时间戳
进出围栏类型	inOutType	1	1-进围栏 2-出围栏

3.1.3. 运输行程服务

3.1.3.1. 产品典型场景

物流运单完成时，对于运单真实性、途经区域情况往往无有效手段追溯，增加了企业后期复盘的难度。

通过运输行程服务可知晓货物运输过程中的途径城市追溯、停车点追溯、订单线路追溯，

并可在费用结算时校验业务真实性。

3.1.3.2. 产品功能说明

基于实时计算引擎分析运输全程停车点、途径城市信息，支撑调度人员管理。可追溯近 6 个月行程，单次最长可追溯 72 小时行程。能够精确反馈停车及途径城市的情况。

输入车牌号，车牌颜色，车辆出发时间和到达时间，始发地，目的地，始发地坐标，目的地坐标，停车时长。返回行程节点，途径城市情况，中途停留点情况。

● API 服务说明

通过 API 的方式对接，根据传入参数，系统自动计算后，按返回参数，通过 API 的方式回传给客户。

接口整体技术服务支持，[跳转](#)。

取值方式：行政区划编码为传入值，没有传入目的地和起始地坐标时，取行政区划默认坐标作为起点或者终点。（默认坐标是地图默认的该地区行政机关所在地坐标）

业务规则：

- 1) 时间段：最长时间差是 72 小时，最长历史时间是 6 个月以内。
- 2) 支持蓝色车牌，黄色车牌和黄绿色车牌。
- 3) 停车时长若是不填写，默认 10 分钟。填写，最小 10 分钟。

3.1.3.3. API 对接说明

3.1.3.3.1. 运输行程服务

● 对接地址

请求地址格式 </save/apis/routerPath>

● 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识（API 访问凭证），对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
vclN	车牌号	陕 YH0009	是
vco	车牌颜色 (1 蓝色、2 黄色、3 黄绿色)	2	是
qryBtm	开始时间	2021-09-05 15:56:46	是，近 6 个月自然月
qryEtm	结束时间	2021-09-06 15:56:46	是，与开始时间相差 72 小时之内
startLonlat	始发地坐标 (WGS84 坐标系)	116.31795, 30.4252	否，始发地坐标和目的地坐标为一组，始发地行政区划

endLonlat	目的地坐标 (WGS84 坐标系)	110.9946817, 25.9089816	编码和目的地行政区划编码为一组, 都输入的时候以始发地坐标、目的地坐标为准;
startAreaCode	始发地行政区划编码	431102	
endAreaCode	目的地行政区划编码	450324	
parkMins	停车时长 (分钟)	10	否, 不填默认 10min

● 输出参数列表

参数	说明
result	查询结果
status	返回状态码: 详见附录 A

接口返回数据解码后样例:

```
{
  "result": {
    "cityArray": [
      {
        "cityCode": "500106",
        "cityName": "沙坪坝区",
        "vInOutTime": "1636473625000",
        "vInOutType": "1"
      },
      {
        "cityCode": "500106",
        "cityName": "沙坪坝区",
        "vInOutTime": "1636473927000",
        "vInOutType": "2"
      },
      {
        "cityCode": "500120",
        "cityName": "璧山区",
        "vInOutTime": "1636473927000",
        "vInOutType": "1"
      },
      {
        "cityCode": "500120",
        "cityName": "璧山区",
        "vInOutTime": "1636474558000",
        "vInOutType": "2"
      },
      {
        "cityCode": "500151",
        "cityName": "铜梁区",
        "vInOutTime": "1636474558000",
        "vInOutType": "1"
      },
      {
        "cityCode": "500151",
        "cityName": "铜梁区",
        "vInOutTime": "1636475189000",
        "vInOutType": "2"
      },
      {
        "cityCode": "500111",
        "cityName": "大足区",
        "vInOutTime": "1636475189000",
        "vInOutType": "1"
      },
      {
        "cityCode": "500111",
        "cityName": "大足区",
        "vInOutTime": "1636476779000",
        "vInOutType": "2"
      },
      {
        "cityCode": "512021",
        "cityName": "安岳县",
        "vInOutTime": "1636476779000",
        "vInOutType": "1"
      }
    ],
    "mileage": "89.0",
    "parkArray": [
      {
        "parkAdr": "安徽省宿州市萧县王挖勺庄, 东北方向, 355.7 米",
        "parkBte": "1646119642000",
        "parkEte": "1646127297000",
        "parkLat": "20416010",
        "parkLon": "70006099",
        "parkMins": "127.0"
      },
      {
        "parkAdr": "安徽省宿州市萧县孟楼村, 东北方向, 560.8 米",
        "parkBte": "1646180061000",
        "parkEte": "1646181193000",
        "parkLat": "20487928",
        "parkLon": "70172121",
        "parkMins": "18.0"
      },
      {
        "parkAdr": "安徽省宿州市萧县萧县鑫安机动车检测中心, 西北方向, 96.7 米",
        "parkBte": "1646182783000",
        "parkEte": "1646183421000",
        "parkLat": "20524680",
        "parkLon": "70136188",
        "parkMins": "10.0"
      }
    ],
    "parkSize": "0",
    "trackArray": [
      {
        "agl": "268",
        "gtm": "20211110/000025",
        "hgt": "302",
        "lat": "17783673",
        "lon": "63774741",
        "mlg": "24284.1",
        "spd": "0.0"
      },
      {
        "agl": "268",
        "gtm": "20211110/000055",
        "hgt": "302",
        "lat": "17783673",
        "lon": "63774741",
        "mlg": "24284.1",
        "spd": "0.0"
      },
      {
        "agl": "268",
        "gtm": "20211110/000125",
        "hgt": "302",
        "lat": "17783673",
        "lon": "63774741",
        "mlg": "24284.1",
        "spd": "0.0"
      },
      {
        "agl": "268",
        "gtm": "20211110/000155",
        "hgt": "302",
        "lat": "17783673",
        "lon": "63774741",
        "mlg": "24284.1",
        "spd": "0.0"
      },
      {
        "agl": "266",
        "gtm": "20211110/000229",
        "hgt": "302",
        "lat": "17784535",
        "lon": "63755820",
        "mlg": "24287.2",
        "spd": "88.0"
      },
      {
        "agl": "265",
        "gtm": "20211110/000259",
        "hgt": "299",
        "lat": "17784262",
        "lon": "63750864",
        "mlg": "24288.0",
        "spd": "100.0"
      },
      {
        "agl": "272",
        "gtm": "20211110/000328",
        "hgt": "305",
        "lat": "17784139",
        "lon": "63745801",
        "mlg": "24288.9",
        "spd": "91.0"
      },
      {
        "agl": "273",
        "gtm": "20211110/000358",
        "hgt": "321",
        "lat": "17784444",
        "lon": "63741394",
        "mlg": "24289.6",
        "spd": "84.0"
      },
      {
        "agl": "263",
        "gtm": "20211110/000428",
        "hgt": "314",
        "lat": "17784315",
        "lon": "63736743",
        "mlg": "24290.4",
        "spd": "98.0"
      },
      {
        "agl": "271",
        "gtm": "20211110/000458",
        "hgt": "303",
        "lat": "17783902",
        "lon": "63731550",
        "mlg": "24291.3",
        "spd": "98.0"
      },
      {
        "agl": "281",
        "gtm": "20211110/000527",
        "hgt": "307",
        "lat": "17784394",
        "lon": "63726900",
        "mlg": "24292.1",
        "spd": "88.0"
      },
      {
        "agl": "281",
        "gtm": "20211110/000557",
        "hgt": "320",
        "lat": "17785176",
        "lon": "637226"
      }
    ]
  }
}
```

10", "mlg": "24292.9", "spd": "84.0"}, {"agl": "269", "gtm": "20211110/000627", "hgt": "334", "lat": "17785620", "lon": "63718242", "mlg": "24293.6", "spd": "87.0"}, {"agl": "267", "gtm": "20211110/000657", "hgt": "350", "lat": "17785404", "lon": "63713676", "mlg": "24294.4", "spd": "95.0"}, {"agl": "275", "gtm": "20211110/000726", "hgt": "371", "lat": "17785563", "lon": "63708979", "mlg": "24295.2", "spd": "89.0"}, {"agl": "262", "gtm": "20211110/000756", "hgt": "384", "lat": "17785605", "lon": "63704406", "mlg": "24296.0", "spd": "91.0"}, {"agl": "254", "gtm": "20211110/000826", "hgt": "364", "lat": "17784519", "lon": "63699636", "mlg": "24296.8", "spd": "96.0"}, {"agl": "259", "gtm": "20211110/000856", "hgt": "371", "lat": "17783506", "lon": "63695029", "mlg": "24297.6", "spd": "92.0"}, {"agl": "261", "gtm": "20211110/000926", "hgt": "388", "lat": "17782884", "lon": "63690504", "mlg": "24298.4", "spd": "86.0"}, {"agl": "249", "gtm": "20211110/000956", "hgt": "391", "lat": "17781919", "lon": "63685938", "mlg": "24299.2", "spd": "96.0"}, {"agl": "252", "gtm": "20211110/001026", "hgt": "377", "lat": "17780481", "lon": "63681040", "mlg": "24300.0", "spd": "99.0"}, {"agl": "266", "gtm": "20211110/001055", "hgt": "364", "lat": "17779767", "lon": "63677274", "mlg": "24300.7", "spd": "98.0"}, {"agl": "0", "gtm": "20211110/001125", "hgt": "0", "lat": "17779767", "lon": "63677274", "mlg": "24300.7", "spd": "0.0"}, {"agl": "0", "gtm": "20211110/001155", "hgt": "364", "lat": "17779767", "lon": "63677274", "mlg": "24300.7", "spd": "0.0"}, {"agl": "0", "gtm": "20211110/001225", "hgt": "364", "lat": "17779767", "lon": "63677274", "mlg": "24300.7", "spd": "0.0"}, {"agl": "0", "gtm": "20211110/001255", "hgt": "364", "lat": "17779767", "lon": "63677274", "mlg": "24300.7", "spd": "0.0"}, {"agl": "0", "gtm": "20211110/001325", "hgt": "364", "lat": "17779767", "lon": "63677274", "mlg": "24300.7", "spd": "0.0"}, {"agl": "0", "gtm": "20211110/001355", "hgt": "364", "lat": "17779767", "lon": "63677274", "mlg": "24300.7", "spd": "0.0"}, {"agl": "270", "gtm": "20211110/001430", "hgt": "279", "lat": "17779734", "lon": "63641488", "mlg": "24306.5", "spd": "94.0"}, {"agl": "267", "gtm": "20211110/001558", "hgt": "304", "lat": "17780206", "lon": "63627415", "mlg": "24309.0", "spd": "90.0"}, {"agl": "274", "gtm": "20211110/001629", "hgt": "317", "lat": "17780274", "lon": "63622743", "mlg": "24309.7", "spd": "89.0"}, {"agl": "282", "gtm": "20211110/001659", "hgt": "309", "lat": "17780932", "lon": "63618093", "mlg": "24310.5", "spd": "104.0"}, {"agl": "282", "gtm": "20211110/001729", "hgt": "304", "lat": "17781904", "lon": "63612963", "mlg": "24311.4", "spd": "99.0"}, {"agl": "274", "gtm": "20211110/001759", "hgt": "319", "lat": "17782614", "lon": "63608175", "mlg": "24312.3", "spd": "91.0"}, {"agl": "275", "gtm": "20211110/001829", "hgt": "328", "lat": "17782584", "lon": "63603481", "mlg": "24313.1", "spd": "91.0"}, {"agl": "296", "gtm": "20211110/001859", "hgt": "335", "lat": "17783742", "lon": "63598887", "mlg": "24313.9", "spd": "93.0"}, {"agl": "296", "gtm": "20211110/001929", "hgt": "346", "lat": "17785656", "lon": "63594565", "mlg": "24314.7", "spd": "94.0"}, {"agl": "294", "gtm": "20211110/001959", "hgt": "348", "lat": "17786428", "lon": "63592611", "mlg": "24315.1", "spd": "0.0"}, {"agl": "294", "gtm": "20211110/002029", "hgt": "348", "lat": "17786428", "lon": "63592611", "mlg": "24315.1", "spd": "0.0"}, {"agl": "294", "gtm": "20211110/002129", "hgt": "348", "lat": "17786428", "lon": "63592611", "mlg": "24315.1", "spd": "0.0"}, {"agl": "276", "gtm": "20211110/002200", "hgt": "343", "lat": "17791782", "lon": "63572038", "mlg": "24318.6", "spd": "89.0"}, {"agl": "271", "gtm": "20211110/002230", "hgt": "317", "lat": "17791998", "lon": "63566982", "mlg": "24319.4", "spd": "98.0"}, {"agl": "270", "gtm": "20211110/002259", "hgt": "297", "lat": "17792095", "lon": "63562150", "mlg": "24320.2", "spd": "88.0"}, {"agl": "277", "gtm": "20211110/002329", "hgt": "297", "lat": "17792328", "lon": "63557610", "mlg": "24321.0", "spd": "92.0"}, {"agl": "279", "gtm": "20211110/002358", "hgt": "297", "lat": "17792328", "lon": "63557610", "mlg": "24321.0", "spd": "92.0"}]

02429", "hgt": "303", "lat": "17793589", "lon": "63548488", "mlg": "24322.6", "spd": "88.0"}, {"agl": "270", "gtm": "20211110/002459", "hgt": "318", "lat": "17793901", "lon": "63543919", "mlg": "24323.3", "spd": "90.0"}, {"agl": "269", "gtm": "20211110/002529", "hgt": "331", "lat": "17793939", "lon": "63539512", "mlg": "24324.1", "spd": "82.0"}, {"agl": "275", "gtm": "20211110/002559", "hgt": "324", "lat": "17794068", "lon": "63535245", "mlg": "24324.8", "spd": "87.0"}, {"agl": "282", "gtm": "20211110/002629", "hgt": "328", "lat": "17794812", "lon": "63530647", "mlg": "24325.6", "spd": "94.0"}, {"agl": "281", "gtm": "20211110/002659", "hgt": "332", "lat": "17795742", "lon": "63525919", "mlg": "24326.4", "spd": "92.0"}, {"agl": "267", "gtm": "20211110/002729", "hgt": "332", "lat": "17796096", "lon": "63521196", "mlg": "24327.2", "spd": "93.0"}, {"agl": "264", "gtm": "20211110/002759", "hgt": "316", "lat": "17795631", "lon": "63516501", "mlg": "24328.0", "spd": "91.0"}, {"agl": "276", "gtm": "20211110/002829", "hgt": "325", "lat": "17795697", "lon": "63511965", "mlg": "24328.8", "spd": "86.0"}, {"agl": "276", "gtm": "20211110/002859", "hgt": "345", "lat": "17796159", "lon": "63507567", "mlg": "24329.6", "spd": "85.0"}, {"agl": "274", "gtm": "20211110/002959", "hgt": "356", "lat": "17796348", "lon": "63504550", "mlg": "24330.1", "spd": "0.0"}, {"agl": "274", "gtm": "20211110/003029", "hgt": "358", "lat": "17796348", "lon": "63504550", "mlg": "24330.1", "spd": "0.0"}, {"agl": "285", "gtm": "20211110/003100", "hgt": "378", "lat": "17796819", "lon": "63491595", "mlg": "24332.2", "spd": "95.0"}, {"agl": "260", "gtm": "20211110/003159", "hgt": "393", "lat": "17797978", "lon": "63481936", "mlg": "24333.9", "spd": "94.0"}, {"agl": "258", "gtm": "20211110/003229", "hgt": "392", "lat": "17797203", "lon": "63477411", "mlg": "24334.7", "spd": "89.0"}, {"agl": "270", "gtm": "20211110/003259", "hgt": "392", "lat": "17796708", "lon": "63472935", "mlg": "24335.5", "spd": "84.0"}, {"agl": "276", "gtm": "20211110/003329", "hgt": "387", "lat": "17797060", "lon": "63468501", "mlg": "24336.2", "spd": "89.0"}, {"agl": "261", "gtm": "20211110/003359", "hgt": "379", "lat": "17797065", "lon": "63463821", "mlg": "24337.0", "spd": "88.0"}, {"agl": "249", "gtm": "20211110/003429", "hgt": "381", "lat": "17796021", "lon": "63459568", "mlg": "24337.8", "spd": "84.0"}, {"agl": "254", "gtm": "20211110/003459", "hgt": "390", "lat": "17794713", "lon": "63455466", "mlg": "24338.5", "spd": "85.0"}, {"agl": "265", "gtm": "20211110/003529", "hgt": "389", "lat": "17794102", "lon": "63451122", "mlg": "24339.2", "spd": "86.0"}, {"agl": "266", "gtm": "20211110/003559", "hgt": "391", "lat": "17793831", "lon": "63446709", "mlg": "24340.0", "spd": "89.0"}, {"agl": "259", "gtm": "20211110/003629", "hgt": "390", "lat": "17793405", "lon": "63442179", "mlg": "24340.8", "spd": "88.0"}, {"agl": "252", "gtm": "20211110/003659", "hgt": "392", "lat": "17792380", "lon": "63437745", "mlg": "24341.6", "spd": "91.0"}, {"agl": "261", "gtm": "20211110/003729", "hgt": "399", "lat": "17791455", "lon": "63433297", "mlg": "24342.3", "spd": "92.0"}, {"agl": "275", "gtm": "20211110/003759", "hgt": "398", "lat": "17791446", "lon": "63428478", "mlg": "24343.1", "spd": "93.0"}, {"agl": "274", "gtm": "20211110/003829", "hgt": "393", "lat": "17791845", "lon": "63423564", "mlg": "24344.0", "spd": "96.0"}, {"agl": "270", "gtm": "20211110/003859", "hgt": "386", "lat": "17792169", "lon": "63418617", "mlg": "24344.8", "spd": "93.0"}, {"agl": "263", "gtm": "20211110/003929", "hgt": "396", "lat": "17791881", "lon": "63414027", "mlg": "24345.6", "spd": "85.0"}, {"agl": "267", "gtm": "20211110/003959", "hgt": "408", "lat": "17791455", "lon": "63409534", "mlg": "24346.4", "spd": "89.0"}, {"agl": "287", "gtm": "20211110/004059", "hgt": "403", "lat": "17793228", "lon": "63400498", "mlg": "24348.0", "spd": "94.0"}, {"agl": "288", "gtm": "20211110/004259", "hgt": "406", "lat": "17797356", "lon": "63383190", "mlg": "24351.0", "spd": "80.0"}, {"agl": "303", "gtm": "20211110/004359", "hgt": "387", "lat": "17800188", "lon": "63378488", "mlg": "24352.6", "spd": "88.0"}]

[illegible]

```
90.0"}, {"agl": "306", "gtm": "20211110/005959", "hgt": "378", "lat": "17886360", "lon": "63285909", "mlg": "24377.5", "spd": "84.0"}]], "status": 1001}
```

数据体详细字段说明:

输出参数名称	输出参数字段	示例数据	备注
路程总里程	mileage	100.8	单位: 千米
停车次数	parkSize	1	
停车列表	parkArray		
停车地址	parkAdr	重庆市渝北区铁家湾, 东北方向, 229.0 米	
停车开始时间	parkBte	1628477210000	时间戳
停车结束时间	parkEte	1628485611000	时间戳
纬度	parkLat	17855836	格式: 1/600000.0 (WGS84 坐标系)
经度	parkLon	64007815	格式: 1/600000.0 (WGS84 坐标系)
停车时长	parkMins	30	单位: 分钟
轨迹列表	trackArray		
上报里程	mlg	123.0	单位: 千米
正北方向夹角	agl	270	正北, 大于 0 且小于 90: 东北, 等于 90: 正东, 大于 90 且小于 180: 东南, 等于 180: 正南, 大于 180 且小于 270: 西南, * 等于 270: 正西, 大于 270 且小于等于 359: 西北, 其他: 未知
GPS 时间	gtm	20210809/104650	
海拔	hgt	326	单位: 米
纬度	lat	17855836	格式: 1/600000.0 (WGS84 坐标系)
经度	lon	64007815	格式: 1/600000.0 (WGS84 坐标系)
GPS 速度	spd	54.3	单位: 千米/小时
以下信息若始发地、目的地或始发地行政区划、目的地行政区划未传入, 则不显示			
途经城市列表	cityArray		当始发地坐标、目的地坐标或是始发地行政区划、目的地行政区划不传入时, 途经城市列表不展示
城市名称	cityName	1	对应入参围栏序号

城市编码	cityCode	500106	
进出城市时间	vInOutTime	1628486013000	时间戳
进出城市类型	vInOutType	1	1-进 2-出

3.1.4. 运输中转服务

3.1.4.1. 产品典型场景

物流运输中货物中转场景，出现中途换车的情况，往往无法对全链路的车辆线路进行全盘展示，对货物交接的时间、地点无法精准采集。运输中转服务可线上化管理的对此线路闭环追溯管理，提高企业复杂运输线路可追溯、可分析能力。

通过实时计算引擎，给出精准且完整的行程路线，对行程情况及停车情况进行智能分析，最终形成完整路线。

3.1.4.2. 产品功能说明

基于实时计算引擎分析多车倒货停车点，分析运输过程中的中转行为。本接口提供根据用户传入的车辆车牌信息和出发/到达时间查询车辆轨迹和车辆交汇，输出车辆轨迹、停靠点和交汇点。支持多车最大不超过 5 个。

输入车辆列表：车牌号，车牌颜色，车辆出发时间和到达时间。返回车牌号，车牌颜色，行程节点，停车情况，交汇停靠情况。

● API 服务说明

通过 API 的方式对接，根据传入参数，系统自动计算后，按返回参数，通过 API 的方式回传给客户。

接口整体技术服务支持，[跳转](#)。

业务规则：

1) 时间段：最长时间差是 72 小时，最长历史时间是 6 个月以内。

2) 支持蓝色车牌，黄色车牌和黄绿色车牌。

3) 车辆列表：最多 5 车，不用输入停车和倒货地点，系统自动计算多车交汇点和停车点以及自动识别倒货点。

3.1.4.3. API 对接说明

3.1.4.3.1. 运输中转服务

● 对接地址

请求地址格式：</save/apis/goodsTransferTrack>

● 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识（API 访问凭证），对应账密中的	xxx	是

	client_id		
srt	私钥	xxx	是
vList	车辆列表（对象数组）		最多输入 5 辆车
sort	排序号，车辆前后运输顺序	1	是
vcIN	车牌号	陕 YH0009	是
vco	车牌颜色 (1 蓝色、2 黄色、3 黄绿色)	2	是
qryBtm	出发时间：格式：yyyy-MM-dd HH:mm:ss 或 yyyy-MM-dd（系统自动转换为 yyyy-MM-dd 00:00:00）	2021-07-28 18:09:30	是，近 6 个月自然月
qryEtm	到达时间：格式：yyyy-MM-dd HH:mm:ss 或 yyyy-MM-dd（系统自动转换为 yyyy-MM-dd 23:59:59）	2021-07-28 19:20:30	是，与开始时间相差 72 小时之内，首车和尾车间隔时间不得大于 10 天

接口传参数数据解析后样例：

```
{
  "vList": [
    {
      "sort": 1,
      "vcIN": "冀 LW0366",
      "vco": 2,
      "qryBtm": "2021-07-28 18:09:30",
      "qryEtm": "2021-07-28 19:20:30"
    },
    {
      "sort": 2,
      "vcIN": "冀 LW0377",
      "vco": 2,
      "qryBtm": "2021-07-28 18:40:30",
      "qryEtm": "2021-07-28 20:50:30"
    },
    {
      "sort": 3,
      "vcIN": "冀 LW0388",
      "vco": 2,
      "qryBtm": "2021-07-28 20:35:30",
      "qryEtm": "2021-07-28 22:09:30"
    }
  ],
  "sign": "XXX",
  "token": "XXX",
  "cid": "XXX"
}
```

● 输出参数列表

数据以 json 格式返回，其中主要包含以下几类信息

（注：信息返回格式为转码格式，需解码后才可显示以下数据样例格式）

参数	说明
result	查询结果
status	返回状态码：详见附录 A

接口返回数据解码后样例：

```
{
  "routes": [
    {
      "agl": "213",
      "gtm": "20210414/162636",
      "hgt": "500",
      "lat": "18015379",
      "lon": "69454255",
      "mlg": "1913.0",
      "spd": "75.0",
      "vcIN": "冀 LW0366",
      "vco": "2"
    },
    {
      "agl": "213",
      "gtm": "20210414/162736",
      "hgt": "500",
      "lat": "18015379",
      "lon": "69454255",
      "mlg": "1913.0",
      "spd": "4.9",
      "vcIN": "冀 LW0366",
      "vco": "2"
    },
    {
      "agl": "213",
      "gtm": "20210414/162836",
      "hgt": "500",
      "lat": "18015379",
      "lon": "69454255",
      "mlg": "1913.0",
      "spd": "74.9",
      "vcIN": "冀 LW0377",
      "vco": "2"
    },
    {
      "agl": "213",
      "gtm": "20210414/162836",
      "hgt": "500",
      "lat": "18015379",
      "lon": "69454255",
      "mlg": "1913.0",
      "spd": "4.9",
      "vcIN": "冀 LW0377",
      "vco": "2"
    },
    {
      "agl": "213",
      "gtm": "20210414/164436",
      "hgt": "500",
      "lat": "18015379",
      "lon": "69454255",
      "mlg": "1913.0",
      "spd": "4.9",
      "vcIN": "冀 LW0377",
      "vco": "2"
    },
    {
      "agl": "213",
      "gtm": "20210414/164436",
      "hgt": "500",
      "lat": "18015379",
      "lon": "69454255",
      "mlg": "1913.0",
      "spd": "74.9",
      "vcIN": "冀 LW0377",
      "vco": "2"
    }
  ],
  "transferPoints": [
    {
      "parkAdr": "湖北省黄冈市武穴市",
      "parkBte": "161838856000",
      "parkEte": "1618389816000",
      "parkLat": "23949063",
      "parkLon": "69784863",
      "parkMins": "16",
      "parkSize": "1",
      "vcIN": "冀 LW0366",
      "vco": "2"
    }
  ]
}
```

数据体详细字段说明：

输出参数名称	输出参数字段	示例数据	备注
轨迹列表	routes		
车牌号	vc1N	冀 LW0366	
车牌颜色	vco	2	1 代表蓝色、2 代表黄色、3 代表黄绿
纬度	lat	18015379	格式：1/600000.0 (WGS84 坐标系)
经度	lon	69454255	格式：1/600000.0 (WGS84 坐标系)
GPS 时间	gtm	20210414/162636	格式：yyyyMMdd/HHmmss
GPS 速度	spd	75.0	单位：千米/小时
里程	mlg	1913.0	单位：千米（精确一位小数）
海拔	hgt	500	单位：米
正北方向夹角	agl	213	
交汇/停靠点列表	transferPoints		
交汇点标记	crossFlag	1	0 代表该点为停靠点，1 代表该点为交汇点
交汇前车标记	crossFormerFlag	1	0 代表该点为交汇后车，1 代表该点为交汇前车
车牌号	vc1N	冀 LW0366	
车牌颜色	vco	2	1 代表蓝色、2 代表黄色、3 代表黄绿
停车时长	parkMins	16.0	单位：分钟
停车开始时间	parkBte	1618388856000	格式：时间戳
停车结束时间	parkEte	1618389816000	格式：时间戳
停车位置经度	parkLon	69784863	格式：1/600000.0 (WGS84 坐标系)
停车位置纬度	parkLat	23949063	格式：1/600000.0 (WGS84 坐标系)
停车位置描述	parkAdr	湖北省…，西方向，24.0 米	

3.1.5. 运输规划服务

3.1.5.1. 产品典型场景

物流运输中在途的运单，平台开发者往往需要展示出发地至目的地的全程路径，该情形往往需要集成多个功能，增加了开发复杂度。

运输规划服务可回溯订单的出发地至当前的已行驶路径，同时导航当前到目的地的规划线路，实现功能的二合一。

实现运输全程可视化，辅助物流内控管理。实时精准回溯货物运输线路，结合加老司机线路规划实用导航，能够准确避开难行线路。

3.1.5.2. 产品功能说明

基于实时计算引擎分析运输过程中任一时点起始的最佳路线规划。根据输入的车牌号，车牌颜色，起始时间，预计到达时间，始发地坐标，目的地坐标或者始发地行政区划编码，目的地行政区划编码，返回计算后的已行驶运输线路情况，和从车辆当前到目的地的线路规划。

接口整体技术服务支持，[跳转](#)。

● API 服务说明

通过 API 的方式对接，根据传入参数，系统自动计算后，按返回参数，通过 API 的方式回传给客户。

取值方式：行政区划编码为传入值，没有传入目的地和起始地坐标时，取行政区划默认坐标作为起点或者终点。（默认坐标是地图默认的该地区行政机关所在地坐标）

业务规则：

1) 预计到达时间：从运输工具当前所在地到目的地的预计时间。

2) 从车辆当前位置到目的地线路规划：是指我司结合老司机线路和导航技术，进行大数据计算后，给出的最优线路规划。

3.1.5.3. API 对接说明

3.1.5.3.1. 运输规划服务

● 对接地址

请求地址格式：</save/apis/visualRoute>

● 输入参数列表

参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识（API 访问凭证）， 对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
vc1N	车牌号	陕 YH0008	是
vco	车牌颜色(1 蓝色、2 黄色、3 黄绿色)	2	是
qryBtm	开始时间	2021-09-05 15:56:46	是, 近 6 个月自然月
qryEtm	结束时间	2021-09-06 15:56:46	是, 与开始时间相差 72 小时之内
startLonlat	始发地坐标(WGS84 坐标系)	116. 31795, 30. 4252	否, 始发地坐标和目的地坐标为一组, 始发地行政区
endLonlat	目的地坐标(WGS84 坐标系)	110. 9946817, 25. 9089816	

startAreaCode	始发地行政区划编码（区县编码）	431102	划编码和目的地的行政区划编码为一组，都输入的时候以始发地坐标、目的地坐标为准
endAreaCode	目的地行政区划编码（区县编码）	450324	

● 输出参数列表

参数	说明
result	查询结果
status	返回状态码：详见附录 A

接口返回数据解码后样例：

```
{
  "result": {
    "adr": "",
    "city": "",
    "country": "",
    "drc": "",
    "estimateArriveTime": "",
    "lat": "",
    "lon": "",
    "mil": "",
    "offlineState": false,
    "offlineTime": "",
    "province": "",
    "remainDistance": "",
    "runDistance": "23.0",
    "runRoute": [
      {
        "agl": "0",
        "gtm": "20210906/154324",
        "hgt": "500",
        "lat": "24027137",
        "lon": "69767413",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154325",
        "hgt": "500",
        "lat": "24027137",
        "lon": "69767413",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154326",
        "hgt": "500",
        "lat": "24027137",
        "lon": "69767413",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154336",
        "hgt": "500",
        "lat": "24068106",
        "lon": "69867654",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154337",
        "hgt": "500",
        "lat": "24068106",
        "lon": "69867654",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154341",
        "hgt": "500",
        "lat": "24068106",
        "lon": "69867654",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154342",
        "hgt": "500",
        "lat": "24068106",
        "lon": "69867654",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154343",
        "hgt": "500",
        "lat": "24068106",
        "lon": "69867654",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154344",
        "hgt": "500",
        "lat": "24068106",
        "lon": "69867654",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154345",
        "hgt": "500",
        "lat": "24068106",
        "lon": "69867654",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154346",
        "hgt": "500",
        "lat": "24068106",
        "lon": "69867654",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154347",
        "hgt": "500",
        "lat": "24068106",
        "lon": "69867654",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154348",
        "hgt": "500",
        "lat": "24068106",
        "lon": "69867654",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154349",
        "hgt": "500",
        "lat": "24068106",
        "lon": "69867654",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154350",
        "hgt": "500",
        "lat": "24068106",
        "lon": "69867654",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154351",
        "hgt": "500",
        "lat": "24068106",
        "lon": "69867654",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154402",
        "hgt": "500",
        "lat": "24027137",
        "lon": "69767413",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154403",
        "hgt": "500",
        "lat": "24027137",
        "lon": "69767413",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154404",
        "hgt": "500",
        "lat": "24027137",
        "lon": "69767413",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154405",
        "hgt": "500",
        "lat": "24027137",
        "lon": "69767413",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154406",
        "hgt": "500",
        "lat": "24027137",
        "lon": "69767413",
        "mil": "1002.0",
        "spd": "60.0"
      },
      {
        "agl": "0",
        "gtm": "20210906/154407",
        "hgt": "500",
        "lat": "24027137",
        "lon": "69767413",
        "mil": "1002.0",
        "spd": "60.0"
      }
    ]
  }
}
```

```
02.0", "spd": "60.0"}, {"agl": "0", "gtm": "20210906/154408", "hgt": "500", "lat": "24027137",
"lon": "69767413", "mil": "1002.0", "spd": "60.0"}, {"agl": "0", "gtm": "20210906/154409", "h
gt": "500", "lat": "24027137", "lon": "69767413", "mil": "1002.0", "spd": "60.0"}, {"agl": "0",
"gtm": "20210906/154451", "hgt": "500", "lat": "24027137", "lon": "69767413", "mil": "1002.0
", "spd": "60.0"}, {"agl": "0", "gtm": "20210906/154521", "hgt": "500", "lat": "24027137", "lo
n": "69767413", "mil": "1003.0", "spd": "60.0"}, {"agl": "0", "gtm": "20210906/154551", "hgt":
"500", "lat": "24027137", "lon": "69767413", "mil": "1004.0", "spd": "60.0"}, {"agl": "0", "gt
m": "20210906/154621", "hgt": "500", "lat": "24027137", "lon": "69767413", "mil": "1005.0", "
spd": "60.0"}, {"agl": "0", "gtm": "20210906/154651", "hgt": "500", "lat": "24027137", "lon":
"69767413", "mil": "1006.0", "spd": "60.0"}, {"agl": "0", "gtm": "20210906/154721", "hgt": "5
00", "lat": "24027137", "lon": "69767413", "mil": "1007.0", "spd": "60.0"}, {"agl": "0", "gtm":
"20210906/154751", "hgt": "500", "lat": "24027137", "lon": "69767413", "mil": "1008.0", "spd
": "60.0"}, {"agl": "0", "gtm": "20210906/154821", "hgt": "500", "lat": "24027137", "lon": "69
767413", "mil": "1009.0", "spd": "60.0"}, {"agl": "0", "gtm": "20210906/154851", "hgt": "500",
"lat": "24027137", "lon": "69767413", "mil": "1010.0", "spd": "60.0"}, {"agl": "0", "gtm": "20
210906/154921", "hgt": "500", "lat": "24027137", "lon": "69767413", "mil": "1011.0", "spd": "
60.0"}, {"agl": "0", "gtm": "20210906/154951", "hgt": "500", "lat": "24027137", "lon": "69767
413", "mil": "1012.0", "spd": "60.0"}, {"agl": "0", "gtm": "20210906/155021", "hgt": "500", "l
at": "24027137", "lon": "69767413", "mil": "1013.0", "spd": "60.0"}, {"agl": "0", "gtm": "2021
0906/155051", "hgt": "500", "lat": "24027137", "lon": "69767413", "mil": "1014.0", "spd": "60.
0"}, {"agl": "0", "gtm": "20210906/155121", "hgt": "500", "lat": "24027137", "lon": "69767413
", "mil": "1015.0", "spd": "60.0"}, {"agl": "0", "gtm": "20210906/155151", "hgt": "500", "lat":
"24027137", "lon": "69767413", "mil": "1016.0", "spd": "60.0"}, {"agl": "0", "gtm": "20210906
/155221", "hgt": "500", "lat": "24027137", "lon": "69767413", "mil": "1017.0", "spd": "60.0"},
{"agl": "0", "gtm": "20210906/155251", "hgt": "500", "lat": "24027137", "lon": "69767413", "m
il": "1018.0", "spd": "60.0"}, {"agl": "0", "gtm": "20210906/155321", "hgt": "500", "lat": "24
068106", "lon": "69867654", "mil": "1019.0", "spd": "60.0"}, {"agl": "0", "gtm": "20210906/15
5351", "hgt": "500", "lat": "24068106", "lon": "69867654", "mil": "1020.0", "spd": "60.0"}, {"
agl": "0", "gtm": "20210906/155421", "hgt": "500", "lat": "24068106", "lon": "69867654", "mil
": "1021.0", "spd": "60.0"}, {"agl": "0", "gtm": "20210906/155451", "hgt": "500", "lat": "2406
8106", "lon": "69867654", "mil": "1022.0", "spd": "60.0"}, {"agl": "0", "gtm": "20210906/1555
21", "hgt": "500", "lat": "24068106", "lon": "69867654", "mil": "1023.0", "spd": "60.0"}, {"ag
l": "0", "gtm": "20210906/155551", "hgt": "500", "lat": "24068106", "lon": "69867654", "mil":
"1024.0", "spd": "60.0"}, {"agl": "0", "gtm": "20210906/155621", "hgt": "500", "lat": "240681
06", "lon": "69867654", "mil": "1025.0", "spd": "60.0"}], "spd": "", "utc": "", "status": "1001
"}
```

数据体详细字段说明：

输出参数名称	输出参数字段	示例数据	备注
车辆最后定位 经度	lon	23972521	格式：1/600000.0(WGS84 坐标系)
车辆最后定位 纬度	lat	69825114	格式：1/600000.0(WGS84 坐标系)
车辆地理位置 名称	adr	北京市西城区塔院胡 同 12-3 龙翔刀削面川	

		湘菜, 东南方向, 27.0 米	
车辆定位时间	utc	1628487201000	时间戳
速度	spd	65.4	单位: 千米/小时
方向	drc	254	正北, 大于 0 且小于 90: 东北, 等于 90: 正东, 大于 90 且小于 180: 东南, 等于 180: 正南, 大于 180 且小于 270: 西南, * 等于 270: 正西, 大于 270 且小于等于 359: 西北, 其他: 未知
省	province	北京市	
市	city	北京市	
县	country	西城区	
上报里程	mil	0.0	单位: 千米
已行驶轨迹列表	runRoute		
经度	lon	23921481	
纬度	lat	69865608	
GPS 时间	gtm	20210809/082050	yyyymmdd/hhmmss
GPS 速度	spd	46.2	单位: 千米/小时
上报里程	mil	0	单位: 千米
海拔	hgt	62	单位: 米
正北方向夹角	agl	222	
当始发地坐标、目的地坐标或始发地行政区划、目的地行政区为空时, 不展示预估轨迹; 当结束时间早于当前时间时, 不展示预估轨迹			
预估轨迹列表	estimateRoute		
经度	lon	23921481	格式: 1/600000.0(WGS84 坐标系)
纬度	lat	69865608	格式: 1/600000.0(WGS84 坐标系)

3.1.6. 晚点预警服务

3.1.6.1. 产品典型场景

物流运输中在途的运单, 收货方想实时看到货物何时到达, 以便合理的排产。发货方也关注货物何时到达, 关注货物安全。运输公司关注货物何时到达, 以便合理安排下一步运输。

晚点预警服务结合导航技术, 可实时计算当前所在地至目的地预计到达时间, 剩余运距, 可满足三方的需求。

3.1.6.2. 产品功能说明

基于导航和大数据算法, 提供到货预警服务。本服务根据客户提供的车牌号和线路终点信息, 结合车辆当前位置, 通过导航技术与历史运输线路大数据分析, 实时计算运抵目的地

预计到达时间，帮助用户获取预判结果（准时/延迟）。

● API 服务说明

通过 API 的方式对接，根据传入参数，系统自动计算后，按返回参数，通过 API 的方式回传给客户。

接口整体技术服务支持，[跳转](#)。

业务规则：

- 1) 系统通过导航加老司机线路进行判断。
- 2) 起点即运输工具目前所在地，终点，客户传入。
- 3) 起点和终点之间距离必须大于 50KM。

3.1.6.3. API 对接说明

3.1.6.3.1. 晚点预警服务

● 对接地址

请求地址格式 [/save/apis/vArrivePredictV3](#)

● 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识（API 访问凭证），对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
vcIN	车牌号	京 xxx	是
vco	车牌颜色，1：蓝色；2：黄色	2	是
endLon	目的地经度	118.346985	是
endLat	目的地纬度	38.856548	是

● 输出参数列表

接口返回数据解码后样例

```
{"result":{"adr":"安徽省安庆市怀宁县长琳塑业，向西方向，148米", "wholeDis":"225", "lat":"18451089", "lon":"70094469", "residueDis":"73.0", "preTime":"2017-08-20 11:00:00", "state":"1"}, "status":1001}
```

参数	说明
result	查询结果
status	返回状态码：详见附录 A

数据体详细字段说明

输出参数名称	输出参数字段	示例数据	备注
当前位置经度	lon	23986841	字符串类型单位：1/600000.0，WGS84 坐标系
当前位置纬度	lat	69807550	字符串类型单位：1/600000.0，WGS84 坐标系

当前位置描述信息	adr	安徽省安庆市怀宁县长琳塑业，向西方向，148 米	字符串类型
剩余运距	residueDis	304.4	从车辆当前位置至目的地的剩余里程信息，单位：公里，字符串类型
预计到达时间	preTime	2018-09-20 11:00:00	字符串类型，格式如：yyyy-MM-dd HH:mm:ss
查询状态	state	1	1. 成功 2. 运距过短，无法预判（全程运距需大于 50km） 3. 位置信息异常，无法预判 4. 起点与终点无法计算线路距离，无法预判

3.1.7. 目的地-实时通知服务

3.1.7.1. 产品典型场景

在规范化的厂区和园区，装卸货往往需要提前安排装卸设备和装卸人员，对于车辆的准时到达和出发有较强的时效要求。如预先知道车辆的驶达情况，可更便于现场提前部署安排。

目的地-实时消息服务通过订阅车辆的方式，实时计算货物到达情况，通过信息推送的方式完成关键节点提醒，优化周转率。

通过大数据算法和实时计算引擎，对数据进行比较精确和精准的计算，科技含量高，计算精准。通过实时计算引擎，实时高效的提供车辆入场信息。

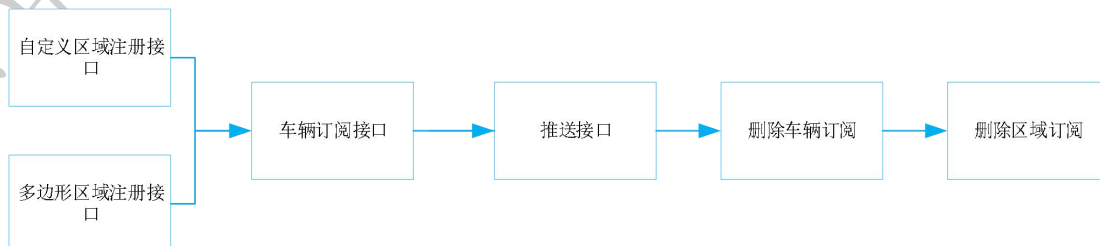
3.1.7.2. 产品功能说明

基于实时进出区域算法，对货物进出区域进行实时提醒。本服务根据客户提供的围栏和绑定的车辆，进行围栏和车辆的监控。

本服务提供方式：由 API 接口提供。

本服务包含 6 个接口：自定义区域注册接口，多边形区域注册接口，车辆订阅接口，车辆驶入事件接口，删除车辆订阅接口，删除自定义区域注册接口。

多接口按如下顺序调用



● API 服务说明

通过 API 的方式对接，根据传入参数，系统自动计算后，按返回参数，通过 API 的方式回传给客户。

接口整体技术服务支持，[跳转](#)。

- 业务规则：

- 订阅围栏

- 1) 有多边形（多边形区域注册接口）和圆形（自定义区域注册接口）
- 2) 传入的坐标为 84 坐标系，小数点后 7 位，经度与纬度以英文逗号隔开。
- 3) 多边形围栏，第一个坐标点与最后一个坐标点必须一样。目前仅支持 1200 个字符，即 57 个坐标点围成的围栏。
- 4) 订阅类型为指定车辆。企业仅允许指定车辆。政府才能使用全国车辆。
- 5) 事件类型：指监控进区域还是出区域。这里默认进区域。
- 6) 围栏生命周期：注册后 10 分钟生效；直至围栏删除。
- 7) 围栏删除的情况：①客户自己删除。②删除订阅车辆时，若该围栏下没有绑定的车辆，围栏会立即删除。③用户账号有效期，过期时，围栏会自动删除。
- 8) 提醒：注册车辆和围栏后，使用完成请及时删除，长期不删除，车辆进出围栏会一直推送消息，扣费较多（按推送扣费）

- 订阅车辆

- 1) 为区域 ID，绑定车辆。
- 2) 系统根据绑定的车辆进行监控。
- 3) 生命周期，注册后 10 分钟生效。

- 推送信息

- 1) 你方需为我方开放平台提供车辆事件的通知回调接口，接口需遵循如下规范：
采用 https 方式对外提供接口服务，提供一个 POST 请求接口，x-www-form-urlencoded 方式接收入参列表的所有参数，然后返回字符串"success"即可。
- 2) 推送接口不支持 IP 调用，必须使用域名调用。

- 删除车辆订阅接口

- 1) 一次性最多请求 100 台车，进行删除订阅，但必须是绑定同一个围栏的车辆。

3.1.7.3. API 对接说明

3.1.7.3.1. 自定义区域注册接口

- 对接地址

请求地址格式 [/save/apis/areaReg](#)

- 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识（API 访问凭证），对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
userflag	用户标识(cid)	zjxl	是
areaname	自定义区域名称（名称不能超 25 个字符，不能包含特殊字符，比如	北仑港	是

	下划线)		
lonlat	自定义区域坐标(84 坐标系, 小数点后小于等于 7 位), 格式: 经度, 纬度二者以英文逗号分隔	121.84431, 29.89889	是
radius	半径(米), 大于 0	30000	是
type	订阅类型: 1: 指定车辆	1	是
actiontype	事件类型 1: 进区域通知	1	是

● 输出参数列表

接口返回数据解码后样例

```
{"result":{"areaid":"48649262BA33928BE050A8C0366804C9","state":"1"},"status":1001}
```

数据以 json 格式返回, 其中主要包含以下几类信息:

参数	说明
result	详见数据说明
status	状态: 详见附录 A;

数据说明

result	说明
areaid	自定义区域 ID (当注册成功时返回自定义区域 ID, 失败时为空)
state	1: 注册成功, 2: 注册失败

3.1.7.3.2. 多边形区域注册接口

● 对接地址

请求地址格式 </save/apis/areaPolygonReg>

● 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识 (API 访问凭证), 对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
areaname	自定义区域名称 (名称不能超 25 个字符, 不能包含特殊字符, 比如下划线)	北仑港	是
lonlats	自定义区域坐标(小数点后小于等于 7 位), 格式: 经度, 纬度二者以英文逗号分隔, 多个坐标用 分割。	113.796387, 41.19519 118.410645, 41.228249 117.773438, 40.245992 116.015625, 40.212441 114.938965, 39.520992 113.796387, 41.19519	是
type	订阅类型: 1: 指定车辆	1	是
actiontype	事件类型 1: 进区域通知	1	是
coordType	坐标类型	坐标类型 1. 谷歌、2. 百度、3. 高德	是

- 输出参数列表

接口返回数据解码后样例

```
{"result":{"areaid":"48649262BA33928BE050A8C0366804C9","state":"1"},"status":1001}
```

数据以 json 格式返回，其中主要包含以下几类信息：

参数	说明
result	详见数据说明
status	状态：详见附录 A；

数据说明

result	说明
areaid	自定义区域 ID（当注册成功时返回自定义区域 ID，失败时为空）
state	1：注册成功，2：注册失败，3：重复注册

3.1.7.3.3. 车辆订阅接口

- 对接地址

请求地址格式 [/save/apis/vnoReg](#)

- 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识（API 访问凭证），对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
userflag	用户标识(cid)	zjxl	是
vnos	车牌号_车牌颜色，英文逗号分隔 车牌颜色：1 蓝色，2 黄色	京 A12345_1, 京 B12345_2	是（最大支持 50 个）
areaid	自定义区域 ID	48649262BA33928BE050A8C0366804C9	是

- 输出参数列表

接口返回数据解码后样例

```
{"result":"1","status":1001}
```

数据以 json 格式返回，其中主要包含以下几类信息

参数	说明
result	1：成功，2：自定义区域 ID 不存在
status	状态：详见附录 A；

3.1.7.3.4. 车辆驶入事件通知接口

- 对接地址

提供一个 POST 请求接口，x-www-form-urlencoded 方式接收入参列表的所有参数

<https://www.XXX.cn/areamsg>

- 输入参数列表

拼接参数	参数说明	参数值	是否必选
------	------	-----	------

vno	车牌号	京 xxx	是
plateColor	车牌颜色 1 蓝色, 2 黄色	2	是
areaId	区域 ID	48649262BA33928BE050A8C0 366804C9	是
areaName	区域名称	北京市	是
type	事件类型 1: 进区域	1	是
utc	时间: UTC 格式精确到毫秒	1484978812831	是

● 接口返回数据样例

"success"

3.1.7.3.5. 删除车辆订阅接口

● 对接地址

请求地址格式 [/save/apis/vnoDel](#)

● 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识 (API 访问凭证), 对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
userflag	用户标识(cid)	zjxl	是
vnos	车牌号_车牌颜色, 英文逗号分隔, 一次请求最多 100 个车牌号 车牌颜色: 1 蓝色, 2 黄色	京 A12345_1, 京 B12345_2	是
areaid	自定义区域 ID	48649262BA33928BE050A8C 0366804C9	是

● 输出参数列表

接口返回数据解码后样例

{"result": "1", "status": "1001"}

数据以 json 格式返回, 其中主要包含以下几类信息

参数	说明
result	1: 成功, 2: 自定义区域 ID 不存在
status	状态: 详见附录 A;

3.1.7.3.6. 删除自定义区域注册接口

● 对接地址

请求地址格式 [/save/apis/areaDel](#)

● 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是

cid	客户端标识（API 访问凭证）， 对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
userflag	用户标识(cid)	zjxl	是
areaid	自定义区域 ID	48649262BA33928BE050A 8C0366804C9	是

● 输出参数列表

接口返回数据解码后样例

```
{"result": "1", "status": 1001}
```

数据以 json 格式返回，其中主要包含以下几类信息

参数	说明
result	1: 成功, 2: 自定义区域 ID 不存在 3: 已删除
status	状态: 详见附录 A;

3.1.8. 起始地-实时通知服务

3.1.8.1. 产品典型场景

在规范化的厂区和园区，装卸货往往需要提前安排装卸设备和装卸人员，对于车辆的准时到达和出发有较强的时效要求。如预先知道车辆的驶达情况，可更便于现场提前部署安排。

起始地-实时消息服务通过订阅车辆的方式，实时计算货物到达情况，通过信息推送的方式完成关键节点提醒，优化周转率。

通过大数据算法和实时计算引擎，对数据进行比较精确和精准的计算，科技含量高，计算精准。通过实时计算引擎，实时高效的提供车辆入场信息。

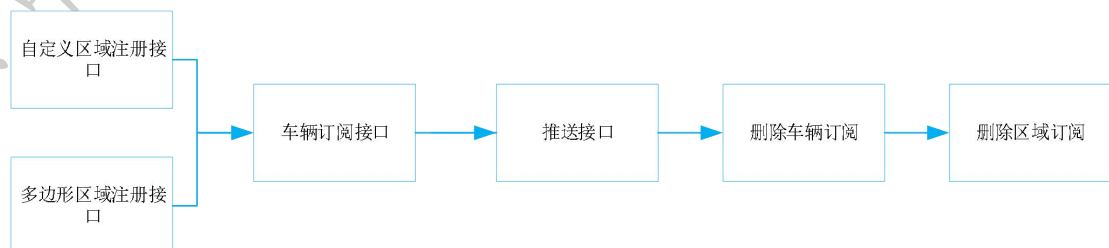
3.1.8.2. 产品功能说明

基于实时进出区域算法，对货物进出区域进行实时提醒。本服务根据客户提供的围栏和绑定的车辆，进行围栏和车辆的监控。

本服务提供方式：由 API 接口提供。

本服务包含 6 个接口：自定义区域注册接口，多边形区域注册接口，车辆订阅接口，车辆驶出事件接口，删除车辆订阅接口，删除自定义区域注册接口。

多接口按如下顺序调用



● API 服务说明

通过 API 的方式对接，根据传入参数，系统自动计算后，按返回参数，通过 API 的方式

回传给客户。

接口整体技术服务支持，[跳转](#)。

- 业务规则：

- 订阅围栏

- 1) 有多边形（多边形区域注册接口）和圆形（自定义区域注册接口）
- 2) 传入的坐标为 84 坐标系，小数点后 7 位，经度与纬度以英文逗号隔开。
- 3) 多边形围栏，第一个坐标点与最后一个坐标点必须一样。目前仅支持 1200 个字符，即 57 个坐标点围成的围栏。
- 4) 订阅类型为指定车辆。企业仅允许指定车辆。政府才能使用全国车辆。
- 5) 事件类型：指监控进区域还是出区域。
- 6) 围栏生命周期：注册后 10 分钟生效；直至围栏删除。
- 7) 围栏删除的情况：①客户自己删除。②删除订阅车辆时，若该围栏下没有绑定的车辆，围栏会立即删除。③用户账号有效期，过期时，围栏会自动删除。
- 8) 提醒：注册车辆和围栏后，使用完成请及时删除，长期不删除，车辆进出围栏会一直推送消息，扣费较多（按推送扣费）

- 订阅车辆

- 1) 为区域 ID，绑定车辆。
- 2) 系统根据绑定的车辆进行监控。
- 3) 生命周期，注册后 10 分钟生效。

- 推送信息

- 1) 你方需为我方开放平台提供车辆事件的通知回调接口，接口需遵循如下规范：
采用 https 方式对外提供接口服务，提供一个 POST 请求接口，x-www-form-urlencoded 方式接收入参列表的所有参数，然后返回字符串"success"即可。
- 2) 推送接口不支持 IP 调用，必须使用域名调用。

- 删除车辆订阅接口

- 1) 一次性最多请求 100 台车，进行删除订阅，但必须是绑定同一个围栏的车辆。

3.1.8.3. API 对接说明

3.1.8.3.1. 自定义区域注册接口

- 对接地址

请求地址格式 [/save/apis/areaReg](#)

- 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识（API 访问凭证），对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
userflag	用户标识(cid)	zjxl	是
areaname	自定义区域名称（名称不能超 25	北仑港	是

	个字符，不能包含特殊字符，比如下划线)		
lonlat	自定义区域坐标(84坐标系，小数点后小于等于7位)，格式：经度，纬度二者以英文逗号分隔	121.84431, 29.89889	是
radius	半径(米)，大于0	30000	是
type	订阅类型：1：指定车辆	1	是
actiontype	事件类型 2：出区域通知	1	是

● 输出参数列表

接口返回数据解码后样例

```
{"result":{"areaid":"48649262BA33928BE050A8C0366804C9","state":"1"},"status":1001}
```

数据以 json 格式返回，其中主要包含以下几类信息：

参数	说明
result	详见数据说明
status	状态：详见附录 A；

数据说明

result	说明
areaid	自定义区域 ID（当注册成功时返回自定义区域 ID，失败时为空）
state	1：注册成功，2：注册失败

3.1.8.3.2. 多边形区域注册接口

● 对接地址

请求地址格式 [/save/apis/areaPolygonReg](#)

● 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识（API 访问凭证），对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
areaname	自定义区域名称（名称不能超 25 个字符，不能包含特殊字符，比如下划线）	北仑港	是
lonlats	自定义区域坐标(小数点后小于等于7位)，格式：经度,纬度二者以英文逗号分隔,多个坐标用 分割。	113.796387, 41.19519 118.410645, 41.228249 117.773438, 40.245992 116.015625, 40.212441 114.938965, 39.520992 113.796387, 41.19519	是
type	订阅类型：1：指定车辆	1	是
actiontype	事件类型：2：出区域通知	1	是

coordType	坐标类型	坐标类型 1. 谷歌、2. 百度、3. 高德	是
-----------	------	------------------------	---

● 输出参数列表

接口返回数据解码后样例

```
{"result":{"areaid":"48649262BA33928BE050A8C0366804C9"},"state":"1"},"status":1001}
```

数据以 json 格式返回，其中主要包含以下几类信息：

参数	说明
result	详见数据说明
status	状态：详见附录 A；

数据说明

result	说明
areaid	自定义区域 ID（当注册成功时返回自定义区域 ID，失败时为空）
state	1：注册成功，2：注册失败，3：重复注册

3.1.8.3.3. 车辆订阅接口

● 对接地址

请求地址格式 [/save/apis/vnoReg](#)

● 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识（API 访问凭证），对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
userflag	用户标识(cid)	zjxl	是
vnos	车牌号_车牌颜色，英文逗号分隔 车牌颜色：1 蓝色，2 黄色	京 A12345_1, 京 B12345_2	是（最大支持 50 个）
areaid	自定义区域 ID	48649262BA33928BE050A8C0366804C9	是

● 输出参数列表

接口返回数据解码后样例

```
{"result":"1","status":1001}
```

数据以 json 格式返回，其中主要包含以下几类信息

参数	说明
result	1：成功，2：自定义区域 ID 不存在
status	状态：详见附录 A；

3.1.8.3.4. 车辆驶出事件通知接口

● 对接地址

提供一个 POST 请求接口，x-www-form-urlencoded 方式接收入参列表的所有参数

<https://www.XXX.cn/areamsg>

- 输入参数列表

拼接参数	参数说明	参数值	是否必选
vno	车牌号	京 xxx	是
plateColor	车牌颜色 1 蓝色, 2 黄色	2	是
areaId	区域 ID	48649262BA33928BE050A8C0 366804C9	是
areaName	区域名称	北京市	是
type	事件类型: 2: 出区域	1	是
utc	时间: UTC 格式精确到毫秒	1484978812831	是

- 接口返回数据样例

"success"

3.1.8.3.5. 删除车辆订阅接口

- 对接地址

请求地址格式 [/save/apis/vnoDel](#)

- 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识 (API 访问凭证), 对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
userflag	用户标识 (cid)	zjxl	是
vnos	车牌号_车牌颜色, 英文逗号分隔, 一次请求最多 100 个车牌号 车牌颜色: 1 蓝色, 2 黄色	京 A12345_1, 京 B12345_2	是
areaaid	自定义区域 ID	48649262BA33928BE050A8C 0366804C9	是

- 输出参数列表

接口返回数据解码后样例

{"result": "1", "status": "1001"}

数据以 json 格式返回, 其中主要包含以下几类信息

参数	说明
result	1: 成功, 2: 自定义区域 ID 不存在
status	状态: 详见附录 A;

3.1.8.3.6. 删除自定义区域注册接口

- 对接地址

请求地址格式 [/save/apis/areaDel](#)

- 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识（API 访问凭证）， 对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
userflag	用户标识(cid)	zjxl	是
areaid	自定义区域 ID	48649262BA33928BE050A 8C0366804C9	是

● 输出参数列表

接口返回数据解码后样例

```
{"result": "1", "status": 1001}
```

数据以 json 格式返回，其中主要包含以下几类信息

参数	说明
result	1: 成功, 2: 自定义区域 ID 不存在 3: 已删除
status	状态: 详见附录 A;

3.1.9. 行驶报警服务

3.1.9.1. 产品典型场景

在运输过程中，司机疲劳驾驶、超速形式等极易引发安全事故，这种情况司机人身安全受到影响，同时作为承运方也会承担货物无法送达的违约责任。行驶报警预警服务通过预警提醒，让司机和服务接入方实时了解到车辆行驶的违规状态，也可及时根据路段情况进行预警。让接入方能通过预警信息进行下一步管控，降低事故发生概率，消弭运输风险。

通过实时计算引擎，向接入方推送超速、疲劳异常状态及车辆即将到达的区域出现恶劣天气或者危险路段信息。

通过多全国道路的危险路段，道路限速，车辆运行情况进行大数据的运算和分析，精准的计算出车辆当时情况，实时通知司机。

3.1.9.2. 产品功能说明

基于全国公里网络的限速和危险标识，对客户做实时危险路段，超速和疲劳提醒。

通过订阅实现对车辆的路过危险路段，恶劣天气等外在条件的监控。同时通过算法对疲劳驾驶，超速行驶等进行监控提醒。

本服务提供方式：由 API 接口提供。

接口整体技术服务支持，[跳转](#)。

报警生效时间为：订阅成功后十分钟生效。预警生效时间为：第二天 0 点生效。

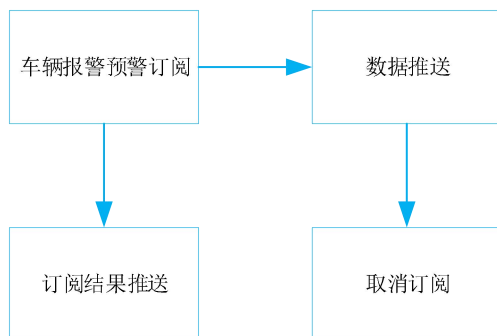
同一个车辆，同一天不能订阅两次（不管服务类型是否相同，都不要订阅两次，否则会重复推送，重复扣费。）

提醒：注册车辆和围栏后，使用完成请及时删除，长期不删除，车辆进出围栏会一直推送消息，扣费较多（按推送扣费）。

本服务包含 4 个接口：车辆报警预警订阅接口，订阅结果推送接口，车辆报警预警数据

推送接口，车辆报警预警取消订阅接口。

多接口按如下顺序调用



服务类型：两种报警类型（报警数据、预警数据），三种服务（数据接口、车机提醒、人工服务）组合后，有 5 种服务类型。客户只能选择一种。

报警数据类别

一级类型	二级类型	三级类型	备注
报警消息	超速	初级	超过道路限速值
		中级（速度）	超过道路限速值 20%
		中级（时长）	持续超速 10 分钟，即使达不到速度阈值也会报中级报警
		严重	超过道路限速值 40%
	疲劳	初级	连续驾驶时间超过 4 小时
		中级	连续驾驶时间超过 5 小时
		严重	连续驾驶时间超过 6 小时
	高速公路违规停车	初级	停车 5 分钟
		严重	停车 10 分钟

预警数据类别

一级类型	二级类型	三级类型	备注
预警消息	危险路段	事故易发路段	
		易滑路段	
		连续下坡路段	
	疲劳预警	高速服务区	
		高速停车区	
	恶劣天气	风雨	
		雨	
		大雨	
		雷雨	
		暴雨	
		雾	
		大雾	
		雪	
		大雪	
		暴雪	
		积雪	

		雨夹雪	
		冰雹	
		大风	
		台风	
		飓风	
		结冰	
		浮尘	
		沙尘暴	
		扬沙	
安全教育	每周	规则一	每周一，结合车辆运营情况进行总结性信息车机下发
服务消息	道路消息	道路事故	
		道路施工	
		其他	

3.1.9.3. API 对接说明

3.1.9.3.1. 行驶报警订阅接口

- 对接地址
请求地址格式 [/save/apis/rTimeWarnReg](#)
- 输入参数列表

拼接参数	参数说明	参数值	是否必选	备注
token	用户令牌	xxx	是	token
cid	客户端标识（API 访问凭证），对应账密中的 client_id	xxx	是	cid
srt	私钥	xxx	是	srt
vcIns	车牌号_车牌颜色	冀 MF1234_2, 冀 JQY045_2.....	是	二选一，最多 100 个，若两者都填写以 VIN 为准
vins	车架号			
comCode	企业编码		否	慧保企业编码，客户需要找慧保索取
owner	所属车队/人	宋慧乔	否	最多输入 30 个字
disCode	所属区/县编码		否	数字，最多 12 位
cityCode	所属市级编码		否	数字，最多 12 位
proCode	所属省级编码		否	数字，最多 12 位
qryBtm	订阅起始时间	2019-07-07 16:04:53	是	起始时间大于当前时间，
qryEtm	订阅结束时间	2019-07-08 16:04:53	是	结束时间大于起始时间
type	服务类型	A: 报警数据（数据接口），B: 预警数据（数据接口+车机提醒）	是	订阅后，预警数据第二天生效，报警

		C: 报警数据（数据接口+车机提醒），D: 报警数据+预警数据（两者均包含：数据接口+车机提醒+人工服务），E: 报警数据+预警数据（两者均包含：数据接口+车机提醒）		数据 10 分钟后生效
--	--	---	--	-------------

● 输出参数列表

```
{"status": 1001}
```

参数	说明
status	返回状态码：详见附录 A

3.1.9.3.2. 车辆订阅结果推送接口

● 对接地址

提供一个 POST 请求接口，x-www-form-urlencoded 方式接收入参列表的所有参数

<https://www.XXX.cn/areamsg>

● 输入参数列表

拼接参数	参数说明	参数值	是否必选
vno	车牌号	冀 MF1234	是
vco	车牌颜色	1、蓝色 2、黄色	是
vin	车架号		是
qryBtm	服务起期	2019-07-07 16:04:53	是
qryEtm	服务止期	2019-07-08 16:04:53	是
owner	所属车队/人	宋慧乔	否
disCode	所属区/县编码		否
cityCode	所属市级编码		否
proCode	所属省级编码		否
impStatus	订阅状态	1、成功 0、失败	是
impFailMsg	订阅失败原因	1、车辆信息列表为空 2、车辆数超过 300 3、系统异常	是
impTime	订阅成功时间	1484978812831	是

● 接口返回数据样例：

```
"success"
```

3.1.9.3.3. 车辆实时预警推送接口

● 对接地址

提供一个 POST 请求接口，x-www-form-urlencoded 方式接收入参列表的所有参数

<https://www.XXX.cn/areamsg>

● 输入参数列表

拼接参数	参数说明	参数值	是否必选
vno	车牌号	冀 MF1234	是
vco	车牌颜色	1、蓝色 2、黄色	是
oneType	一级类型	报警消息	是
twoType	二级类型	超速、疲劳、高速公路违规 停车、高速公路低速行驶	是
threeType	三级类型	初级、中级、严重	是
sendTime	下发时间	1484978812831	是
addr	位置描述	114. 503205, 37. 15352 (84 坐标系)	否
msg	提醒内容	国家货运平台提醒：您已超 速，请减速慢行。	是
dangerAreaLoc ation	危险标识牌经纬度	114. 503205, 37. 15352 (84 坐标系)	否
vin	车架号	LSSR6PEB7KR000810	否

● 接口返回数据样例：
"success"

3.1.9.3.4. 车辆报警预警取消订阅接口

● 对接地址

请求地址格式

<https://testopen.95155.com/save/apis/rTimeWarnCancel>

● 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识 (API 访问凭证)	xxx	是
srt	私钥	xxx	是
vc1N	车牌号 (单个车牌号)	冀 MF1234	是

● 输出参数列表

{"status": 1001}

(注：信息返回格式为转码格式，需解码后才可显示以下数据样例格式)

参数	说明
status	返回状态码：详见附录 A

3.1.10. 货物失联提醒服务

3.1.10.1. 产品典型场景

在运输过程中，货主，承运商，收货方都需要监控货物和运输工具的实时动态，但是若车机突然失联后，三方都无法监控。并且无法监控都是事后才知道。

我司提供货物失联提醒服务，当车机突然失联后，通过实时通知，告知客户，货物失联了。客户收到信息后，货主，承运商，收货方，三方都可以以电话通知或者其他方式咨询司机。

3.1.10.2. 产品功能说明

平台实时监控货物运行情况，当货物离线时长达到预设阈值（10 分钟）时，平台将车辆异常离线状态数据推送到用户数据接收接口。

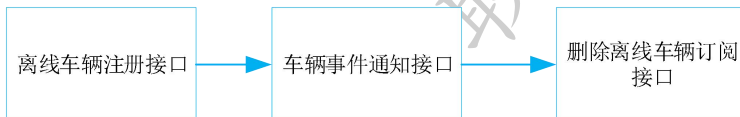
本服务提供方式：由 API 接口提供。

接口整体技术服务支持，[跳转](#)。

注册后 10 分钟生效，生效后 10 分钟（默认阈值），开始推送数据。

提醒：注册车辆和围栏后，使用完成请及时删除，长期不删除，车辆进出围栏会一直推送消息，扣费较多（按推送扣费）。

本服务包含 3 个接口：离线车辆注册接口，车辆事件通知接口，删除离线车辆订阅接口。多接口按如下顺序调用



3.1.10.3. API 对接说明

3.1.10.3.1. 离线车辆注册接口

- 对接地址
请求地址格式/save/apis/vnoOutReg

- 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识（API 访问凭证），对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
userflag	用户标识(cid)	zjx1	是
vnos	车牌号_车牌颜色，英文逗号分隔，一次请求最多 100 个车牌号 车牌颜色：1 蓝色，2 黄色	京 A12345_1, 京 B12345_2	是

- 输出参数列表

接口返回数据解码后样例：

```
{"result": "1", "status": 1001}
```

数据以 json 格式返回，其中主要包含以下几类信息

参数	说明
result	1: 成功
status	状态：详见附录 A；

3.1.10.3.2. 车辆离线事件通知接口

● 对接地址

提供一个 POST 请求接口，x-www-form-urlencoded 方式接收入参列表的所有参数

<https://www.XXX.cn/areamsg>

● 输入参数列表

拼接参数	参数说明	参数值	是否必选
userId	用户 ID	离线注册时使用的 API 账号	是
vno	车牌号	京 XXX	是
plateColor	车牌颜色：1 蓝色，2 黄色	2	是
type	事件类型：3: 车辆离线	3	是
utc	时间：UTC 格式精确到毫秒	1484978812831	是

● 接口返回数据解码后样例：

```
"success"
```

3.1.10.3.3. 删除离线车辆订阅接口

● 对接地址

请求地址格式/save/apis/vnoOutDel

● 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识（API 访问凭证），对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
userflag	用户标识(cid)	zjxl	是
vnos	车牌号_车牌颜色，英文逗号分隔，一次请求最多 100 个车牌号 车牌颜色：1 蓝色，2 黄色	京 A12345_1, 京 B12345_2	是

● 输出参数列表

接口返回数据解码后样例

```
{"result": "1", "status": 1001}
```

数据以 json 格式返回，其中主要包含以下几类信息

参数	说明
----	----

result	1: 成功
status	状态: 详见附录 A;

3.1.11. 区域内停车提醒服务

3.1.11.1. 产品典型场景

生产型企业或者销售型企业, 对于销售有区域限制的时候, 因为价格差异, 经常会出现不法经销商进行串货。

生产型企业的风控部门, 针对已知的部分串货区域, 进行围栏检测, 进入围栏后, 若有停车, 则疑似串货, 实时通知风控部门。企业风控部门进行实时监控和处理。

3.1.11.2. 产品功能说明

平台实时监控货物运行情况, 当货物离线时长达到预设阈值 (10 分钟) 时, 平台将车辆异常离线状态数据推送到用户数据接收接口。

本服务提供方式: 由 API 接口提供。

接口整体技术服务支持, [跳转](#)。

围栏生命周期:

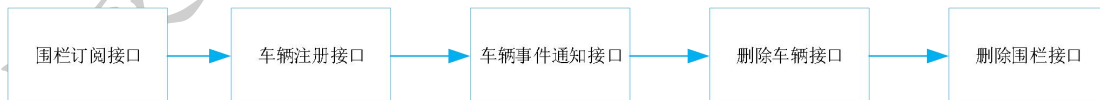
- ① 注册后 10 分钟生效, 生效后 10 分钟 (默认阈值), 开始推送数据
- ② 删除围栏的情况: 客户自己删除。
- ③ 删除围栏的情况: 删除订阅车辆时, 若该围栏下没有绑定的车辆, 围栏会立即删除。
- ④ 删除围栏的情况: 用户账号有效期, 过期时, 围栏会自动删除

注册后 10 分钟生效, 生效后 10 分钟 (默认阈值), 开始推送数据。

提醒: 注册车辆和围栏后, 使用完成请及时删除, 长期不删除, 车辆进出围栏会一直推送消息, 扣费较多 (按推送扣费)。

本服务包含 5 个接口: 区域注册接口, 车辆订阅接口, 车辆停车时间通知接口, 删除车辆订阅接口, 删除区域接口。

多接口按如下顺序调用



3.1.11.3. API 对接说明

3.1.11.3.1. 区域注册接口

- 对接地址

请求地址格式: /save/apis/areaRegV5

- 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是

cid	客户端标识（API 访问凭证）， 对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
areaname	自定义区域名称（名称不能超 25 个字符，不能包含特殊字符，比如下划线）	北仑港	是
lonlats	自定义区域坐标格式：经度，纬度英文逗号分隔，多个用 分割。坐标点的个数，最少 4 个，最多 20 个（多边形）（第一个和最后一个必须一样）； 圆心坐标格式：经度，纬度以英文逗号分隔。	113.796387, 41.19519 118.410645, 41.228249 117.773438, 40.245992 116.015625, 40.212441 114.938965, 39.520992 113.796387, 41.19519（多边形）； 116.015625, 40.212441（圆形）	是
type	订阅类型：1：指定车辆（只允许白名单以内的车辆）	1	是
actiontype	事件类型：5 区域内停车；	5	是
timeout	驶出超时（单位：分钟） 驶出超时时间=驶出时间-驶入时间，（要求正整数据）	20	否
stoptime	区域内停车时长（单位：分钟） 不能小于 5	5	是
coorType	坐标类型	坐标类型 1. 谷歌、2. 百度、3. 高德	是
areatype	区域类型	区域类型 1 多边形 2 圆形	否，不传时默认为 1
radius	半径（单位：米）最小 500M，最大 30000M	1000	否 Areatype 为 2 时，必传

● 输出参数列表

```
{ "result": { "areaid": "48649262BA33928BE050A8C0366804C9", "state": "1" }, "status": "1001" }
```

数据以 json 格式返回，其中主要包含以下几类信息：

参数	说明
result	详见数据说明
status	状态：详见附录 A；

数据体详细字段说明：

result	说明
areaid	自定义区域 ID（当注册成功时返回自定义区域 ID，失败时空）
state	1：注册成功，2：注册失败，3：区域名称被占用

3.1.11.3.2. 车辆订阅接口

● 对接地址

请求地址格式 [/save/apis/vnoReg](#)

● 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识（API 访问凭证），对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
userflag	用户标识(cid)	zjxl	是
vnos	车牌号_车牌颜色，英文逗号分隔 车牌颜色：1 蓝色，2 黄色	京 A12345_1, 京 B12345_2	是（最大支持 50 个）
areaid	自定义区域 ID	48649262BA33928BE050A8C0366804C9	是

● 输出参数列表

接口返回数据解码后样例

```
{ "result": "1", "status": "1001" }
```

数据以 json 格式返回，其中主要包含以下几类信息

参数	说明
result	1: 成功, 2: 自定义区域 ID 不存在
status	状态: 详见附录 A;

3.1.11.3.3. 车辆停车事件通知接口

● 对接地址

提供一个 POST 请求接口，x-www-form-urlencoded 方式接收入参列表的所有参数

<https://www.XXX.cn/areamsg>

● 输入参数列表

拼接参数	参数说明	参数值	是否必选
vno	车牌号	京 xxx	是
vco	颜色（1 蓝色，2 黄色）	1	否
areaId	区域 ID	48649262BA33928BE050A8C0366804C9	事件类型为 1、2 时 必填
type	事件类型：5 区域内停车；	5	是
utc	时间：UTC 格式精确到毫秒	1484978812831	是
vin	车架号	LFV202020009876	否
stopLonlat	停靠点经纬度(84 坐标系，小数点后小于等于 7 位)，格式：经度, 纬度二者以英文逗号分隔	121.84431, 29.89889	否
stopAddr	停靠点位置	河北省保定市正定县 XXX 街道 XX 村	否
stopStartTime	停靠开始时间	1484978812831	否

- 接口返回数据样例

"success"

3.1.11.3.4. 删除车辆订阅接口

- 对接地址

请求地址格式 /save/apis/vnoDel/

- 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识 (API 访问凭证), 对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
userflag	用户标识(cid)	zjxl	是
vnos	车牌号_车牌颜色, 英文逗号分隔, 一次请求最多 100 个车牌号 车牌颜色: 1 蓝色, 2 黄色	京 A12345_1, 京 B12345_2	是
areaid	自定义区域 ID	48649262BA33928B E050A8C0366804C9	是

- 输出参数列表

接口返回数据解码后样例

{"result": "1", "status": "1001"}

数据以 json 格式返回, 其中主要包含以下几类信息

参数	说明
result	1: 成功, 2: 自定义区域 ID 不存在
status	状态: 详见附录 A;

3.1.11.3.5. 删除区域注册接口

- 对接地址

请求地址格式 /save/apis/areaDel/

- 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识 (API 访问凭证), 对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
userflag	用户标识(cid)	zjxl	是
areaid	自定义区域 ID	48649262BA33928B E050A8C0366804C9	是

- 输出参数列表

接口返回数据解码后样例

```
{"result": "1", "status": "1001"}
```

数据以 json 格式返回，其中主要包含以下几类信息

参数	说明
result	1: 成功, 2: 自定义区域 ID 不存在 3: 已删除
status	状态: 详见附录 A;

3.1.12. 偏航预警服务

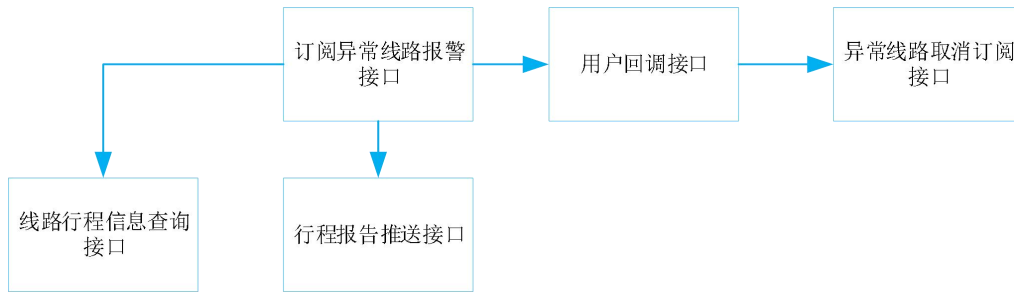
3.1.12.1. 产品典型场景

运输企业对自己商品运输线路有要求，比如高值商品，比如特殊类型的商品；也有一些生产型企业，要求按指定路线行走。但是实际上司机行走线路是否按客户要求，客户根本不知道。

因此我司提供偏航预警服务，根据客户提供的起点，终点，中转点进行线路预定，之后根据预设线路进行偏航，离线，停车实时通知。

3.1.12.2. 产品功能说明

- 功能详解
 - 根据客户提供的起点，终点，中转点，进行多条线路的设置。并且根据设置的线路进行偏航，离线，停车进行实时通知服务。
 - 先订阅，后监控，订阅后 10 分钟生效。
 - 起始地坐标、目的地坐标，起始地区县编码，目的地区县编码：行政区划编码是必填，坐标为非必填，但若是填写坐标，则以坐标为准。若填写坐标，则默认半径为 3000 米。
 - 可以支撑最多 5 个中转地。
 - 离线时长、停车时长：以 10 的整数倍设置，单位为分钟。不设置默认 30 分钟。
 - 偏离报警频次，默认 5 分钟一次，可设置为最小 5 分钟，最大 60 分钟，大于 5 分钟时，必须是 10 的整数倍。
 - 标识：报警、解除报警。
 - ①报警：离线，就是指系统发现车机离线了；停车，就是系统发现停车了；偏离，就是指系统发现车辆偏移。
 - ②解除报警：离线，就是指系统发现车机上线了；停车，就是指系统发现开车了；偏离，就是指系统发现车辆结束偏离。
 - 行程线路推送接口，必须在“订阅异常线路报警接口”中，send 字段选择了“推送”。
- 本服务提供方式：由 API 接口提供。
- 接口整体技术服务支持，[跳转](#)。
- 本服务包含 5 个接口：订阅异常线路报警接口，用户回调接口，异常线路取消订阅接口，行程线路信息查询接口，行程报告推送接口。
- 多接口按如下顺序调用



3.1.12.3. API 对接说明

3.1.12.3.1. 订阅偏航报警接口

- 对接地址

请求地址格式 </save/apis/subAbnormalLineWarnV2>

- 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识 (API 访问凭证), 对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
vclN	车牌号	京 xxx	是
vco	车牌颜色	车牌颜色, 1: 蓝色; 2: 黄色	是
startCoor	起始地坐标 (谷歌坐标系)	经纬度点, 116. 346985, 39. 856548	否 (若不填, 则出行政区, 算做出发; 若填写, 则以出坐标 3000 米范围就算出发)
startCode	起始地区县编码	222222	是
destCoor	目的地坐标 (谷歌坐标系)	经纬度点, 118. 346985, 39. 856548	否 (若不填, 则进行行政区, 算做到达; 若填写, 则以进入到坐标 3000 米范围才算到达)
destCode	目的地地区县编码	111111	是
stime	任务开始时间	utc 格式	是, 开始时间与结束时间之间不能超过 72 小时
etime	任务结束时间	utc 格式	是, 开始时间与结束时间之间不能超过 72 小时
type	订阅预警类型 type=1 异常线路预警 type=2 异常线路预警+停车 type=3 异常线路预警	用户根据业务需要进行接口订阅	是

	+离线 type=4 异常线路预警 +停车+离线		
transitCode_time	中转地编码_预计到达时间(最多支持 5 个中转地,预计到达时间采用 utc 时间)	360104_1580641200000, 340100_1580706000000, 130600_1580806800000	否 (多个中转地时,必须按路过的先后顺序填写,以英文逗号隔开)
parkMins	停车时长	默认为 30, 单位为分钟, 大于等于 10 的整数	否
outMins	离线时长	默认为 30, 单位为分钟, 大于等于 10 的整数	否
abnMins	偏离报警频次	默认 5 分钟, (单位为分钟), 最小 5 分钟, 当大于 5 分钟时, 必须为 10 的整数倍, 最大值 60 分钟	否
send	任务监控过程中是否推送行程记录	默认 1, 不推送 2: 推送	否

● 输出参数列表

```
{
  "result": [
    {
      "110121_110122": "25f838d8-d67e-48b6-92d9-1c3f9ff59bb0"
    },
    {
      "110121_110122_110123": "35f838d8-d67e-48b6-92d9-1c3f9ff59bb0"
    }
  ],
  "status": 1001
}
```

(注: 信息返回格式为转码格式, 需解码后才可显示以下数据样例格式)

参数	说明
result	结果, JSON 格式返回, 如果订阅成功, 返回结果为订阅的线路与对应的行程 rid
status	返回状态码: 详见附录 A

3.1.12.3.2. 用户回调接口

● 对接地址

你方需为我方开放平台提供车辆事件的通知回调接口, 接口需遵循如下规范:

采用 https 方式对外提供接口服务, 提供一个 POST 请求接口, x-www-form-urlencoded 方式接收入参列表的所有参数, 然后返回字符串 "success" 即可。

接口不支持 IP 调用, 必须使用域名调用。

注意: 该接口使用 post form 格式推送。推送的字段为 data,String 类型。接收的时候, 使用 String data 接收 (data 是参数名), 然后再解析字符串的内容即可 (内容是 json 字符串), 或产品在进行更详细编写注解。

偏离预警

请求地址格式, POST 参数为表单提交方式: https://???/apis/abnormalLineWarn

停车预警

请求地址格式, POST 参数为表单提交方式: https://???/apis/parkWarn

离线预警

请求地址格式, POST 参数为表单提交方式: https://???/apis/offLineWarn

位置信息回调 (不管是否有异常, 每半小时推送一条位置信息)

请求地址格式, POST 参数为表单提交方式: https://???/apis/locationInfo

● 输入参数列表

请求参数

拼接参数	参数说明	参数值	备注
data	回调数据内容，JSON 字符串	详情见	

回调参数说明

偏离：

```
data="{\"adr\":\"天津市东丽区\",\"dLoc\":\"内蒙古自治区乌兰察布市察哈尔右翼前旗\",\"label\":\"1\",\"rid\":\"50f31b74-197c-4ea0-a9f4-1c1986eb3f60\",\"sLoc\":\"天津市滨海新区\",\"time\":1600672391546,\"type\":\"偏离\",\"userName\":\"c8c66fc5-b892-493d-9af1-7bf55d5a4e11\",\"vno\":\"晋 F77002\",\"warnTime\":5}\""
```

离线：

```
data="{\"adr\":\"天津市东丽区\",\"dLoc\":\"内蒙古自治区乌兰察布市察哈尔右翼前旗\",\"label\":\"1\",\"rid\":\"50f31b74-197c-4ea0-a9f4-1c1986eb3f60\",\"sLoc\":\"天津市滨海新区\",\"time\":1600672391546,\"type\":\"离线\",\"userName\":\"c8c66fc5-b892-493d-9af1-7bf55d5a4e11\",\"vno\":\"晋 F77002\",\"warnTime\":30}\""
```

停车：

```
data="{\"adr\":\"天津市东丽区\",\"dLoc\":\"内蒙古自治区乌兰察布市察哈尔右翼前旗\",\"label\":\"1\",\"rid\":\"50f31b74-197c-4ea0-a9f4-1c1986eb3f60\",\"sLoc\":\"天津市滨海新区\",\"time\":1600672391546,\"type\":\"停车\",\"userName\":\"c8c66fc5-b892-493d-9af1-7bf55d5a4e11\",\"vno\":\"晋 F77002\",\"warnTime\":30}\""
```

位置信息：

```
data="{\"adr\":\"天津市东丽区\",\"dLoc\":\"内蒙古自治区乌兰察布市察哈尔右翼前旗\",\"label\":null,\"rid\":\"50f31b74-197c-4ea0-a9f4-1c1986eb3f60\",\"sLoc\":\"天津市滨海新区\",\"time\":1600672391546,\"type\":\"位置信息\",\"userName\":\"c8c66fc5-b892-493d-9af1-7bf55d5a4e11\",\"vno\":\"晋 F77002\",\"warnTime\":null}\""
```

接口返回数据解码后样例：

"success"

字段名称	字段说明	示例数据	数据类型
userName	用户名称		字符串类型
rid	行程 id	25f838d8-d67e-48b6-92d9-1c3f9ff59bb0	字符串类型
vno	车牌号	京 NXXXXX	字符串类型
sLoc	出发地	河北省邯郸市大名县	字符串类型
dLoc	目的地	河北省保定市雄县	字符串类型
adr	报警地点	河南省濮阳市南乐县	字符串类型
type	报警类型：偏离、停车、离线、位置信息	偏离	字符串类型
label	标识（1、报警；2、解除报警；）	2	字符串类型
time	报警发生时间	1493987977000	整型，utc
warnTime	报警持续时间	20	整型（单位分钟）

3.1.12.3.3. 异常线路取消订阅接口

● 对接地址

请求地址格式 </save/apis/delAbLine>

● 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识 (API 访问凭证), 对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
rid	行程 ID	25f838d8-d67e-48b6-92d9-1c3f9ff59bb0	是

● 输出参数列表

接口返回数据解码后样例: {"result": "1", "status": 1001}

参数	说明
result	删除结果 1 成功 2 失败
status	返回状态码: 详见附录 A 1006 行程 ID 不存在

3.1.12.3.4. 线路行程信息查询接口

● 对接地址

请求地址格式 </save/apis/queryLineRecord>

● 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识 (API 访问凭证), 对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
rid	异常线路预警接口返回的行程标识	25f838d8-d67e-48b6-92d9-1c3f9ff59bb0	是

● 输出参数列表

字段名称	字段说明	示例数据	数据类型
Adr	报警地点	河南省濮阳市南乐县	字符串
type	报警类型: 偏离、停车、离线、位置信息	偏离	字符串
time	报警发生时间	1493987977000	整型, utc
outTime	持续时间	20	整型 (单位分钟)

返回参数如下:

```
{"result": {"records": [{"adr": "山东省济南市槐荫区 AA 物流中心", "type": "离开", "time": "1552406400000", "outTime": null}, {"adr": "山东省济南市 xx 区", "type": "驶入", "time": "1552406400000", "outTime": null}, {"adr": "山东省济南市 xx 区", "type": "驶出", "time": "1552406400000", "outTime": null}, {"adr": "山西省 xx 市 xx 区",
```

```
"type": "偏离", "time": "1552406400000", "outTime": null}, {"adr": "山东省济南市 xx 区", "type": "停车", "time": "1552406400000", "outTime": 30}, {"adr": "山东省济南市 xx 区", "type": "离线", "time": "1552406400000", "outTime": 40}, {"adr": "江苏省徐州市云龙区", "type": "驶入", "time": "1552406400000", "outTime": null}, {"adr": "江苏省徐州市云龙区 AA 产业园", "type": "达到", "time": "1552406400000", "outTime": null}]]}, {"status": 1001}
```

参数	说明
result	结果, JSON 格式返回
status	返回状态码: 详见附录 A

3.1.12.3.5. 行程报告推送接口

● 对接地址

你方需为我方开放平台提供车辆事件的通知回调接口, 接口需遵循如下规范:

采用 https 方式对外提供接口服务, 支持 POST 表单提交方式, 我方将发起 https 请求, 接口参数中传递用户标识, 并验证用户标识的合法性和安全性, 然后正确接收通知后, 返回应答消息。该接口在行程结束后, 推送行程报告给客户。

接口不支持 IP 调用, 必须使用域名调用。

请求地址格式, POST 参数为表单提交方式: <https://?/?/?/?/apis/recordInfo>

● 回调参数

回调数据格式

```
{
  "rid": "25f838d8-d67e-48b6-92d9-1c3f9ff59bb0",
  "userName": "25f838d8-d67e-48b6-92d9-1c3f9ff59bb0",
  "records": [
    {
      "adr": "山东省济南市槐荫区 AA 物流中心",
      "type": "离开",
      "time": "1552406400000",
      "outTime": null
    },
    {
      "adr": "山东省济南市 xx 区",
      "type": "驶入",
      "time": "1552406400000",
      "outTime": null
    },
    {
      "adr": "山东省济南市 xx 区",
      "type": "驶出",
      "time": "1552406400000",
      "outTime": null
    },
    {
      "adr": "山西省 xx 市 xx 区",
      "type": "偏离",
      "time": "1552406400000",
      "outTime": null
    },
    {
      "adr": "山东省济南市 xx 区",
      "type": "停车",
      "time": "1552406400000",
      "outTime": 30
    },
    {
      "adr": "山东省济南市 xx 区",
      "type": "离线",
      "time": "1552406400000",
      "outTime": 40
    },
    {
      "adr": "江苏省徐州市云龙区",
      "type": "驶入",
      "time": "1552406400000",
      "outTime": null
    },
    {
      "adr": "江苏省徐州市云龙区 AA 产业园",
      "type": "达到",
      "time": "1552406400000",
      "outTime": null
    }
  ]
}
```

回调参数说明

字段名称	字段说明	示例数据	数据类型	备注
rid	行程 id	25f838d8-d67e-48b6-92d9-1c3f9ff59bb0	字符串	
records	行程信息		数据列表	
adr	地点	山西省 xx 市 xx 区	字符串	
type	事件类型	偏离	字符串	
time	事件发生时间	1552406400000	整型, utc	
outTime	持续时间	30	整型	单位分钟

JSON 格式返回

```
{"result": 1552406400000, "status": 1001}
```

参数	说明
result	结果, utc 时间戳
status	返回状态码

3.1.13. 货车城市溯源服务

3.1.13.1. 产品典型场景

政府或交管等平台需要回溯货运车辆一定时间内轨迹行程时使用。

3.1.13.2. 产品功能说明

实现货车途径城市溯源查询服务,通过传入具体车辆属性来查询近 x 天途径的所有区域(区/县)是否涉及订阅的风险等级区域以及具体停留信息、进出时间等。

本服务以 API 接口的形态提供服务。

接口整体技术服务支持, [跳转](#)。

3.1.13.3. API 对接说明

3.1.13.3.1. 货车城市溯源服务

- 对接地址
请求地址格式/ [save/apis/areaTrace](#)

- 输入参数列表

参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识 (API 访问凭证), 对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
vno	车牌号	京 xxx	是
plateColor	车牌颜色 1 蓝色, 2 黄色 3 黄绿色	2	是
riskRating	风险等级 (1 全部, 2 高, 3 中、4 低)	默认 1 可同时传输 2 个值, 如高中, 用逗号分隔	是
timeDimension	时间维度 (14 天内, 7 天内, 自定义)	默认 14, 单位: 天, 最大 14 天	是

- 输出参数列表
数据以 json 格式返回, 其中主要包含以下几类信息:

参数	说明
result	详见数据说明
status	状态: 详见附录 A;

数据详细字段说明:

输出参数字段	输出参数名	示例数据	备注
--------	-------	------	----

areaName	区域名称（途径省市区县名称，最小到区县）	北京市海淀区	所有途径区域都输出
vInTime	驶入区域时间	1635497914000	所有途径区域都输出
vOutTime	驶出区域时间	1635497914000	所有途径区域都输出
riskRating	风险等级（高，中，低）	高	
duration	停留时长（单位 h）	1.2h	保留 1 位小数

3.1.14. 中途停车服务

3.1.14.1. 产品典型场景

在物流场景下，承运方完成运单后，货主方针对运单中停车地址进行核验查询，防止串货换货发生。

3.1.14.2. 产品功能说明

本接口提供指定车牌号，指定时间段查询车辆停车信息，开始时间和结束时间不能超过 24 小时。

本服务以 API 接口的形态提供服务。

接口整体技术服务支持，[跳转](#)。

● API 服务说明

通过 API 的方式对接，根据传入参数，系统自动计算后，按返回参数，通过 API 的方式回传给客户。

根据输入车牌号，车牌颜色，车辆出发时间和到达时间，始发地，目的地，始发地坐标，目的地坐标，停车时长。返回行程节点，途径城市情况，中途停留点情况。

3.1.14.3. API 对接说明

3.1.14.3.1. 中途停车服务

● 对接地址

请求地址格式 [/save/apis/vQueryPark](#)

● 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识（API 访问凭证），对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
vcIN	车牌号	陕 YH0009	是
vco	车牌颜色	车牌颜色，1：蓝色；2：黄色	是
qryBtm	查询起始时间 (yyyy-MM-dd HH:mm:ss)	2018-09-18 09:00:00 或 2018-09-18（若为 yyyy-MM-dd 格	是

	和 yyyy-MM-dd 格式)	式, 则查询指定日期 24 小时内的数据)	
qryEtm	查询截止时间 (yyyy-MM-dd HH:mm:ss 和 yyyy-MM-dd 格式)	2018-09-18 10:30:25 或 2018-09-18 (若为 yyyy-MM-dd 格式, 则查询指定日期 24 小时内的数据)	是
parkMins	停车时长	默认为 10, 单位为分钟, 大于等于 3 的整数	否

● 输出参数列表

接口返回数据解码后样例

```
{
  "result": {
    "parkInfo": [
      {
        "adr": "天津市蓟县田野农家, 向东方向, 253 米",
        "bte": "2017-06-27 14:03:53",
        "ete": "2017-06-27 14:22:50",
        "lat": "39.82045333333333",
        "lon": "117.29147166666667",
        "parkMins": "18.0"
      },
      {
        "adr": "天津市蓟县田野农家, 向东方向, 232 米",
        "bte": "2017-06-27 15:08:20",
        "ete": "2017-06-27 17:26:46",
        "lat": "39.82045333333333",
        "lon": "117.29171333333333",
        "parkMins": "138.0"
      }
    ],
    "parkSize": 2,
    "status": 1001
  }
}
```

参数	说明
result	查询结果
status	返回状态码: 详见附录 A

数据体详细字段说明:

输出参数名称	输出参数字段	示例数据	备注
所有停靠信息	parkInfo		
停靠次数	parkSize	2	
停车时长	parkMins	35.0	字符串类型
停车开始时间	bte	2018-09-18 10:00:00	字符串类型
停车结束时间	ete	2018-09-18 10:30:00	字符串类型
停车位置经度	lon	123.236672	84 经度坐标信息
停车位置纬度	lat	60.236672	84 纬度坐标信息
停车位置描述	adr	北京海淀西二旗	字符串类型

3.1.15. 行驶里程服务

3.1.15.1. 产品典型场景

在物流场景下, 承运方完成运单后, 货主方查询承运车辆里程进行结算。

3.1.15.2. 产品功能说明

基于中交兴路智能 transmap-mileage 算法, 本接口提供指定车牌号、指定时间段查询车辆实际行驶里程信息, 开始时间和结束时间不能超过 24 小时。

本服务以 API 接口的形态提供服务。

● API 服务说明

通过 API 的方式对接, 根据传入参数, 系统自动计算后, 按返回参数, 通过 API 的方式

回传给客户。

根据输入车牌号，车牌颜色，查询起始和终止时间。返回行驶里程，起点终点等信息。
接口整体技术服务支持，[跳转](#)。

3.1.15.3. API 对接说明

3.1.15.3.1. 行驶里程服务

- 对接地址

请求地址格式 [/save/apis/vQueryMileage](#)

- 输入参数列表

拼接参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识（API 访问凭证），对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
vclN	车牌号	陕 YH0009	是
vco	车牌颜色	车牌颜色，1：蓝色；2：黄色	是
qryBtm	查询起始时间 (yyyy-MM-dd HH:mm:ss 和 yyyy-MM-dd 格式)	2018-09-18 09:00:00 或 2018-09-18（若为 yyyy-MM-dd 格式，则查询指定日期 24 小时内的数据）	是
qryEtm	查询截止时间 (yyyy-MM-dd HH:mm:ss 和 yyyy-MM-dd 格式)	2018-09-18 10:30:25 或 2018-09-18（若为 yyyy-MM-dd 格式，则查询指定日期 24 小时内的数据）	是

- 输出参数列表

接口返回数据解码后样例

```
{
  "result": {
    "end": "湖南省岳阳市临湘市张家大屋，向东南方向，178 米",
    "endLonLat": "113.49648166666667,29.473815",
    "endTime": "2018-09-18 09:00:00",
    "mileage": "1.3",
    "start": "湖南省岳阳市临湘市李家坡，向东北方向，99 米",
    "startLonLat": "113.483715,29.470065",
    "startTime": "2018-09-18 10:30:00",
    "status": "1001"
  }
}
```

参数	说明
result	查询结果
status	返回状态码：详见附录 A

数据体详细字段说明：

输出参数名称	输出参数字段	示例数据	备注
行驶总里程数	mileage	30.3	本段行程中车辆总行驶里程数。单位：公里，支持 1 位小数，四舍五入
起点时间	startTime	2018-09-18 09:00:00	精确到年月日时分秒
终点时间	endTime	2018-09-18 10:30:00	精确到年月日时分秒

起点位置经纬度	startLonLat	116.346985, 39.856548	本段行程中起点的位置 84 经纬度坐标信息
起点位置描述	start	北京市丰台区万寿寺加油站, 向西北方向, 57 米	本段行程中起点的位置文字地理信息
终点位置经纬度	endLonLat	118.346985, 39.856548	本段行程中终点的位置 84 经纬度坐标信息
终点位置描述	end	河北省唐山市滦县大贺庄子村, 向东北方向, 474 米	本段行程中终点的位置文字地理信息

3.1.16. 车辆完整信息验证服务

3.1.16.1. 产品典型场景

物流场景下, 在运输前承运方需要协调或调度车辆信息进行查询, 判断车辆是否符合运输要求。

3.1.16.2. 产品功能说明

基于中交货运大数据基础上, 经过数据清洗、整合、处理后, 输出道路运输相关业务场景中数据。根据输入的车牌号、车牌颜色列表, 返回车辆完整信息验证结果。

本服务以 API 接口的方式提供服务。

● API 服务说明

通过 API 的方式对接, 根据传入参数, 系统自动计算后, 按返回参数, 通过 API 的方式回传给客户。

本接口提供按照车牌号, 车牌颜色, 批量验证是否在有效期内。批量最多 50 台车。

重复验证, 重复扣费, 请验证后做好相应备份。

接口整体技术服务支持, [跳转](#)。

3.1.16.3. API 对接说明

3.1.16.3.1. 车辆完整信息验证服务 (车牌号)

● 对接地址

请求地址格式 </save/apis/vehicleInfoCompleteV21>

● 输入参数列表

拼接参数	参数说明	参数值	是否必选
vc1Ns	车牌号_车牌颜色, 列表 (以半角逗号连接) (最多 50 辆车) 车牌颜色, 1: 蓝色; 2: 黄色	京 xxx_1, 京 yyy_2, 京 zzz_2, 京 mmm_2	是
token	用户令牌	xxx	是
cid	客户端标识 (API 访问凭证), 对应账密中的 client_id	xxx	是
srt	私钥	xxx	是

● 输出参数列表

接口返回数据解码后样例

```
{
  "result": [
    {
      "ldTn": "11990",
      "vclLng": "9600",
      "vclWdt": "2380",
      "enTn": "12130",
      "carType": "货五",
      "vclTn": "24315",
      "clDrwTn": "24315",
      "engine": "A44D1711878",
      "pltClr": "2",
      "vclHgt": "800",
      "vclN": "云 E17361",
      "state": "1001",
      "vclAxs": "4",
      "vclBrand": "大地牌 RX5240ECCQB",
      "vclTyreNumber": "10",
      "vehicleType": "重型仓栅式货车",
      "drivingLicenseRegisterDate": "20100322",
      "dateOfissue": "20160523",
      "natureOfUsage": "货运",
      "vin": "LGF19AJLXD*****",
      "areaName": "云南省昆明市",
      "ldTn": "12895",
      "vclLng": "5600",
      "vclWdt": "2300",
      "enTn": "11910",
      "carType": "货五",
      "vclTn": "25000",
      "clDrwTn": "25000",
      "engine": "MH4E3C00199",
      "pltClr": "2",
      "vclHgt": "1150",
      "vclN": "鄂 C93316",
      "state": "1001",
      "vclAxs": "3",
      "vclBrand": "东风牌 DFL3258A3",
      "vclTyreNumber": "10",
      "vehicleType": "重型自卸货车",
      "drivingLicenseRegisterDate": "20130423",
      "dateOfissue": "20180714",
      "natureOfUsage": "货运",
      "vin": "LGF19AJLU*****",
      "areaName": "湖北省仙桃市",
      "ldTn": "",
      "vclLng": "",
      "vclWdt": "",
      "enTn": "5830",
      "carType": "货五",
      "vclTn": "15490",
      "clDrwTn": "15490",
      "engine": "51418130",
      "pltClr": "2",
      "vclHgt": "",
      "vclN": "浙 B8A362",
      "state": "1001",
      "vclAxs": "2",
      "vclBrand": "解放牌 CA4158K2R5EA80",
      "vclTyreNumber": "6",
      "vehicleType": "牵引车",
      "drivingLicenseRegisterDate": "20090810",
      "dateOfissue": "20080810",
      "natureOfUsage": "货运",
      "vin": "LGF19AJ*****",
      "areaName": "河北省张家口市"
    }
  ],
  "status": "1001"
}
```

数据以 json 格式返回，其中主要包含以下几类信息

参数	说明
result	查询结果（对于中交不存在的车，则不返回查询结果）
status	返回状态码：详见附录 A

数据体详细字段说明

输出参数名称	输出参数字段	类型	示例数据	备注
车牌号	vclN	String	京 P52656	字符串类型
车牌颜色	pltClr	String	2	字符串类型：1 蓝 2 黄
总质量	vclTn	String	31000	总质量（单位：kg）
准牵引质量	clDrwTn	String	31000	准牵引质量（单位：kg）
额定载质量	ldTn	String	21000	额定载质量（单位：kg）
整备质量	enTn	String	10000	整备质量（单位：kg）
轴数	vclAxs	String	3	车辆轴数（单位：轴）
外廓长	vclLng	String	10655	外廓长（单位：毫米）
外廓宽	vclWdt	String	2496	外廓宽（单位：毫米）
外廓高	vclHgt	String	3450	外廓高（单位：毫米）
货车分类	carType	String 1: 货一 2: 货二 3: 货三 4: 货四 5: 货五	1	车型（货一到货五）定义逻辑： 2 吨以下（含 2 吨）- 货一 2-5 吨（含 5 吨）- 货二 5-10 吨（含 10 吨）- 货三 10-15 吨（含 15 吨）- 货四 15 吨以上- 货五
品牌型号	vclBrand	String	豪沃牌	字符串类型

行驶证注册日期	drivingLicenseRegisterDate	String	20130507	字符串类型
行驶证发证日期	dateOfissue	String	20120815	字符串类型
发动机号	engine	String	17A00198759	字符串类型
车辆类型	vehicleType	String	重型自卸货车	字符串类型
车轮数	vclTyreNumber	String	6	字符串类型
使用性质	natureOfUsage	String	货运	字符串类型
车架号	vin	String	LFWRMU9J9* *****	字符串类型
车籍地	areaName	String	河北省张家口市	字符串类型，所属地区（或地级市）Code 编码
车辆状态	state	String	1001	1001 有数据 1006 无数据

3.1.16.3.2. 车辆完整信息验证服务（车架号）

● 对接地址

请求地址格式 </save/apis/vehicleInfoComVinV21>

● 输入参数列表

拼接参数	参数说明	参数值	是否必选
vins	车架号列表（以半角逗号连接）（最多 50 辆车）	LGF19AJLXDF539086, LGF19AJLUYF537776, LGF19AJLXDF132976	是
token	用户令牌	xxx	是
cid	客户端标识（API 访问凭证），对应账密中的 client_id	xxx	是
srt	私钥	xxx	是

● 输出参数列表

接口返回数据解码后样例

```
{
  "result": [
    {
      "ldTn": "11990",
      "vclLng": "9600",
      "vclWdt": "2380",
      "enTn": "12130",
      "carType": "货五",
      "vclTn": "24315",
      "clDrwTn": "24315",
      "engine": "A44D1711878",
      "pltClr": "2",
      "vclHgt": "800",
      "vclN": "云 E17361",
      "state": "1001",
      "vclAxs": "4",
      "vclBrand": "大地牌",
      "RX5240ECCQB",
      "vclTyreNumber": "10",
      "vehicleType": "重型仓栅式货车",
      "drivingLicenseRegisterDate": "20100322",
      "dateOfissue": "20160523",
      "natureOfUsage": "货运",
      "vin": "LGF19AJLXDF5390876",
      "areaName": "云南省昆明市",
      {
        "ldTn": "12895",
        "vclLng": "5600",
        "vclWdt": "2300",
        "enTn": "11910",
        "carType": "货五",
        "vclTn": "25000",
        "clDrwTn": "25000",
        "engine": "MH4E3C00199",
        "pltClr": "2",
        "vclHgt": "1150",
        "vclN": "鄂 C93316",
        "state": "1001",
        "vclAxs": "3",
        "vclBrand": "东风牌 DFL3258A3",
        "vclTyreNumber": "10",
        "vehicleType": "重型自卸货车",
        "drivingLicenseRegisterDate": "20130423",
        "dateOfissue": "20180714",
        "natureOfUsage": "货运",
        "vin": "LGF19AJLUYF537776",
        "areaName": "湖北省仙桃市",
        {
          "ldTn": "",
          "vclLng": "",
          "vclWdt": "",
          "enTn": "5830",
          "carType": "货五",
          "vclTn":
```

```
"15490", "clDrwTn": "15490", "engine": "51418130", "pltClr": "2", "vclHgt": "", "vclN": "浙B8A362", "state": 1001, "vclAxs": "2", "vclBrand": "解放牌 CA4158K2R5EA80", "vclTyreNumber": "6", "vehicleType": "牵引车", "drivingLicenseRegisterDate": "20090810", "dateOfissue": "20080810", "natureOfUsage": "货运", "vin": "LGF19AJLXDF132976", "areaName": "河北省张家口市"}], "status": 1001}
```

数据以 json 格式返回，其中主要包含以下几类信息

参数	说明
result	查询结果（对于中交不存在的车，则不返回查询结果）
status	返回状态码：详见附录 A

数据体详细字段说明

输出参数名称	输出参数字段	类型	示例数据	备注
车牌号	vclN	String	京 P12346	字符串类型
车牌颜色	pltClr	String	2	字符串类型：1 蓝 2 黄
总质量	vclTn	String	31000	总质量（单位：kg）
准牵引质量	clDrwTn	String	31000	准牵引质量（单位：kg）
额定载质量	ldTn	String	21000	额定载质量（单位：kg）
整备质量	enTn	String	10000	整备质量（单位：kg）
轴数	vclAxs	String	3	车辆轴数（单位：轴）
外廓长	vclLng	String	10655	外廓长（单位：毫米）
外廓宽	vclWdt	String	2496	外廓宽（单位：毫米）
外廓高	vclHgt	String	3450	外廓高（单位：毫米）
货车分类	carType	String 1: 货一 2: 货二 3: 货三 4: 货四 5: 货五	1	车型（货一到货五）定义逻辑： 2 吨以下（含 2 吨）- 货一 2-5 吨（含 5 吨）- 货二 5-10 吨（含 10 吨）- 货三 10-15 吨（含 15 吨）- 货四 15 吨以上- 货五
品牌型号	vclBrand	String	豪沃牌	字符串类型
行驶证注册日期	drivingLicenseRegisterDate	String	20130507	字符串类型
行驶证发证日期	dateOfissue	String	20120815	字符串类型
发动机号	engine	String	17A00198759	字符串类型
车辆类型	vehicleType	String	重型自卸货车	字符串类型
车轮数	vclTyreNumber	String	6	字符串类型
使用性质	natureOfUsage	String	货运	字符串类型
车架号	vin	String	LGF19AJLXDF132976	字符串类型
车籍地	areaName	String	河北省张家口市	字符串类型，所属地区（或地级市）Code 编码
车辆状态	state	String	1001	1001 有数据 1006 无数据

3.1.17. 智能电子围栏 v1.0

3.1.17.1. 产品典型场景

- 场景 1:

用户在新画电子围栏时,不确定围栏是否准确。此时用户可通过近期在此围栏场地进行过业务的车辆,使用其行程数据,校验围栏是否可被触发,以此快速印证围栏的准确性。

- 场景 2:

用户已经使用了电子围栏,但对于电子围栏的使用效果,客户难以进行评估,比如围栏是否准确、车辆触碰围栏总体情况等,因此难以更进一步加强对车辆监控的管理。此时用户可使用电子围栏评估功能。

围栏评估功能通过对订阅围栏车辆近期行程进行大数据运算,根据计算车辆触碰等数据,进行分析和汇总,输出围栏触碰数据、评估等分析结果,使用户可以直观了解其围栏使用情况,指导下一步改善方向,以提高用户企业对运输业务的管理能力。

3.1.17.2. 产品功能说明

智能电子围栏 v1.0 使用可视化的界面,应用基于车辆行程数据及地图区域数据的大数据算法,在为客户提供电子围栏新增,删除,订阅车辆等管理功能外,还提供电子围栏准确性校验及总体使用情况评估功能。

- API 服务说明

通过 API 的方式对接,根据传入参数,系统自动计算后,按返回参数,通过 API 的方式回传给客户。

接口整体技术服务支持, [跳转](#)。

本服务包含 4 个接口:电子围栏时空回溯,电子围栏准确性评估,电子围栏评估完成事件通知,查询电子围栏评估报告。

3.1.17.3. API 对接说明

3.1.17.3.1. 电子围栏时空回溯

- 描述:

根据车辆+时间段的轨迹+停车点,与围栏(圆型或多边形)进行匹配,并返回发生的驶入/驶出事件,且对未触碰围栏的轨迹进行区分,区分如下:

- 1、车辆无轨迹
- 2、未触碰围栏,且轨迹最近点距离围栏 $\leq$ 【围栏灵敏度】
- 3、未触碰围栏,且轨迹最近点距离围栏 $>$ 【围栏灵敏度】

- 接口调用示例:

请求地址格式 <https://openapi-test.sinoiov.cn/save/apis/areaReview>

- 参数列表

参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识(API 访问)	xxx	是

	凭证), 对应账密中的 client_id		
srt	私钥	xxx	是
vc1N	车牌号	陕 YH0008	是
vco	车牌颜色 (1 蓝色、2 黄色、3 黄绿色)	2	是
qryBtm	开始时间, 格式: yyyy-MM-dd HH:mm:ss 或 yyyy-MM-dd (系统 自动转换为 yyyy-MM-dd 00:00:00)	2021-07-28 09:00:00	是
qryEtm	结束时间, 格式: yyyy-MM-dd HH:mm:ss 或 yyyy-MM-dd (系统 自动转换为 yyyy-MM-dd 23:59:59)	2021-07-28 19:20:30	是
areaId	电子围栏 id	12312	是
precisionDis	围栏灵敏度设置, 输入为正数 (小数位最多 1 位), 单位 km, 默认为 5km	3	否

● 输出参数列表

参数	说明
result	查询结果
status	返回状态码: 详见附录 A

接口返回数据解码后样例:

```
{
  "result": {
    "areaId": "",
    "code": "0",
    "eventList": [
      {
        "vco": "2",
        "vc1N": "陕 YH0010",
        "lon": "118.318352",
        "eventType": "1",
        "time": "1653809924000",
        "lat": "31.101822"
      },
      {
        "vco": "2",
        "vc1N": "陕 YH0010",
        "lon": "118.339845",
        "eventType": "2",
        "time": "1653810044000",
        "lat": "31.152925"
      }
    ],
    "areaName": "北仑港",
    "nearestPoint": null
  },
  "status": 1001
}
```

● 数据体详细字段说明:

输出参数名称	输出参数字段	示例数据	备注
回溯结果编码	code	0	0: 请求正常; 1: 请求错误; 2: 轨迹异常; 3: 未搜索到轨迹数据; 4: 未触碰围栏, 且轨迹最近点距离围栏 ≤ 【围栏灵敏度】, 最

			近的点【x, y】; 5: 未触碰围栏, 且轨迹最近点 距离围栏>【围栏灵敏度】, 最 近的点【x, y】;
围栏 id	areaId	123445	
围栏名称	areaName	北仑港	
轨迹触碰围栏 数据列表	eventList	[{"vco": "2", "vclN": "陕 YH0010", "lon": "118. 318352", "eventType": "1", "time": "165380 9924000", "lat": "31. 101822"}]	code 为 0 时有值
车牌号	vclN	陕 YH0008	车牌号
车牌颜色	vco	2	车牌颜色 (1 蓝色、2 黄色、3 黄绿色)
事件类型	eventType	1	1、进区域, 2、出区域
触碰围栏时间	time	1648742400000	时间戳, 精确到毫秒
经度	lon	118.318352	保留 6 位小数
纬度	lat	31.101822	保留 6 位小数
距离围栏最近 轨迹点信息	nearestPoint	{"distance": "0.057", "lon": "116.332515", "time": "1654503355 000", "lat": "40.0420 47"}	当 code 为 4、5 时有值
经度	lon	116.332515	保留 6 位小数
纬度	lat	40.042047	保留 6 位小数
轨迹点时间	time	1654503355000	时间戳, 精确到毫秒
距离围栏的距 离	distance	0.057	单位 km, 保留三位小数

3.1.17.3.2. 电子围栏准确性评估

● 描述:

根据围栏 ID, 轮询最新发起订阅的最多 500 个车辆, 计算时间段为近 72 小时, 使用大数据算法进行评估运算。

● 接口调用示例:

请求地址格式 <https://openapi-test.sinoiov.cn/save/apis/areaEvaluate>

● 参数列表

参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识 (API 访问凭证), 对应账密中的 client_id	xxx	是

srt	私钥	xxx	是
areaId	围栏 id	xxxx	是
precisionDis	围栏灵敏度设置，输入为正数（小数位最多 1 位），单位 km，默认为 5km	3	否

● 输出参数列表

参数	说明
result	围栏评估报告 id（查询评估报告时使用）
status	返回状态码：详见附录 A

接口返回数据解码后样例：

```
{"result": "0c4d84d35a9c44c1a008638246e9489a", "status": 1001}
```

3.1.17.3.3. 电子围栏评估完成事件通知

● 描述：

电子围栏精确性评估完成后会通过此接口通知调用方评估状态

● 接口调用方式

你方需为我方开放平台提供评估完成事件的通知回调接口，接口需遵循如下规范：

采用 https 方式对外提供接口服务，提供一个 POST 请求接口，application/json 方式接收入参列表的所有参数，然后返回字符串“success”即可。

接口不支持 IP 调用，必须使用域名调用。

● 接口调用示例

提供一个 POST 请求接口，application/json 方式接收入参列表的所有参数

<https://www.XXX.cn/areaEvaluateMessage>

● 参数列表

参数	参数说明	参数值	是否必选
id	评估报告 id	0c4d84d35a9c44c1a008638246e9489a	是
state	评估状态，2 成功，3 失败	2	是

● 返回数据样例：

```
success
```

3.1.17.3.4. 查询电子围栏评估报告

● 描述：

查询 6.1.2 接口生成的评估报告

● 接口调用示例：

请求地址格式 <https://openapi-test.sinoiov.cn/save/apis/queryAreaEvaluate>

● 参数列表

参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识 (API 访问凭证), 对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
id	评估报告 id	0c4d84d35a9c44c1a008638246e9489a	是

● 输出参数列表

参数	说明
result	返回结果
status	返回状态码: 详见附录 A

接口返回数据解码后样例:

```
{
  "result": {
    "areaId": "5115302853006141464",
    "areaName": "5594028530060801933",
    "evaluateResult": {
      "eventEnterCount": 0,
      "vehicleTotalCount": 2,
      "vehicleTouchedCount": 0,
      "vehicleNoTouchedCount": 2,
      "vehicleExitCount": 0,
      "vehicleOutPrecisionCount": 2,
      "vehicleEnterCount": 0,
      "eventExitCount": 0,
      "vehicleNoTrackCount": 0,
      "vehicleInPrecisionCount": 0
    },
    "state": "2",
    "failMessage": ""
  },
  "status": 1001
}
```

● 数据体详细字段说明:

输出参数名称	输出参数字段	示例数据	备注
围栏 id	areald	5115302853006141464	
围栏名称	areaName	北仑港	
评估状态	state	2	0 待评估, 1 评估中, 2 评估成功, 3 评估失败
失败信息	failMessage		当 state 为 3 时有值
评估结果数据	evaluateResult		数据集
评估车辆总数	vehicleTotalCount	100	
触碰围栏车辆数	vehicleTouchedCount	80	触碰围栏车辆数
触发驶入事件车辆数	vehicleEnterCount	65	触发驶入事件车辆数
触发驶出事件车辆数	vehicleExitCount	30	触发驶出事件车辆数
触发驶入事件数量	eventEnterCount	75	
触发驶出事件数量	eventExitCount	22	
未触发围栏车	vehicleNoTouchedCount	10	

辆数			
无轨迹车辆数	vehicleNoTrackCount	2	
未触碰围栏,且 轨迹/停车点最 近点距离围栏 ≤【围栏灵敏 度】车辆数	vehicleInPrecisionCount	22	
未触碰围栏,且 轨迹/停车点最 近点距离围栏 >【围栏灵敏 度】车辆数	vehicleOutPrecisionCount	12	

3.1.18. 历史业务质量分析服务

3.1.18.1. 产品典型场景

企业年度内部审计时对历史财年的业务情况需进行内部质量管理控制等相关数据质量分析。当发生货运纠纷需要取证、或应对税务机关的稽查或者是网络货运企业的监管需求等场景。

历史业务质量分析服务为以上场景提供辅助服务或证据支持。

3.1.18.2. 产品功能说明

基于实时计算引擎分析运输起点终点信息,提供企业业务核验管理。可追溯 6 个月前近三年内的行程,单次最长可追溯 72 小时行程。

输入车牌号,车牌颜色,车辆出发时间和到达时间,始发地,目的地,始发地坐标,目的地坐标提交分析任务请求。(接口最大支持 10 辆车)。提交信息后 T+3 工作日可收到我方主动推送的车辆提交任务的行程报告。

● API 服务说明

通过 API 的方式对接,根据传入参数,系统自动计算后,按返回参数,通过 API 的方式回传给客户。

接口整体技术服务支持, [跳转](#)。

业务规则:

- 1) 时间段: 最长时间差是 72 小时, 最长历史时间是 3 年以内。
- 2) 支持蓝色车牌, 黄色车牌和黄绿色车牌。
- 3) 提交任务成功后异步收到服务报告推送 (最晚 T+3 日)

3.1.18.3. API 对接说明

3.1.18.3.1. 服务发起订阅

- **对接地址**
请求地址格式 </save/apis/vHisQualityTask>

- **输入参数列表**

参数	参数说明	参数值	是否必选
token	用户令牌	xxx	是
cid	客户端标识 (API 访问凭证), 对应账密中的 client_id	xxx	是
srt	私钥	xxx	是
vclNs	车牌号_车牌颜色, 列表 (以半角逗号连接), 车牌颜色 1: 蓝色; 2: 黄色, 3: 黄绿色	京 xxx_1, 京 yyy_2, 京 zzz_2, 京 mmm_2	是 (最多 10 辆车)
qryBtm	开始时间	2021-09-05 15:56:46	是, 6 个月之前自然月
qryEtm	结束时间	2021-09-06 15:56:46	是, 与开始时间相差 72 小时之内
startLonlat	始发地坐标 (WGS84 坐标系)	116.317953, 30.425221	否, 始发地坐标和目的地坐标为一组, 始发地行政区划编码和目的地的行政区划编码为一组, 都输入的时候以始发地坐标、目的地坐标为准;
endLonlat	目的地坐标 (WGS84 坐标系)	110.994681, 25.908981	
startAreaCode	始发地行政区划编码	431102	
endAreaCode	目的地行政区划编码	450324	

- **输出参数列表**

参数	说明
result	查询结果
status	返回状态码: 详见附录 A

3.1.18.3.2. 服务报告推送

- **描述**: 推送“发起订阅车辆”的服务报告
- **接口调用方式**
你方需为我方开放平台提供报告推送的回调接口, 接口需遵循如下规范:
采用 https 方式对外提供接口服务, 支持 application/json 接收方式, 我方将发起 https 请求, 你方正确接收报告后, 返回应答消息。
接口不支持 IP 调用, 必须使用域名调用。
- **接口调用示例**
请求地址格式 <http://XXX.XXX.XX/save/>

● 数据体详细字段说明：

输出参数名称	输出参数字段	示例数据	备注
纬度	lat	23986841	字符串类型单位：1/600000.0 WGS84 坐标系
经度	lon	69807550	字符串类型单位：1/600000.0 WGS84 坐标系
GPS 时间	gtm	20140325/104954	字符串类型格式：yyyyMMdd/HHmmss
GPS 速度	spd	602	字符串类型单位 (1/10.0) 百米/时 保留 1 位小数
里程	mlg	1500	字符串类型当大于 0 时，需乘以 0.1 换算成实际里程，单位为 km，其他按 实际值显示
海拔	hgt	3250	字符串类型单位为 m
正北方向夹角	agl	90	字符串类型 (0—359，正北为 0，顺 时针)
是否经过始发地	isOri	1	1:是 0:否
是否经过目的地	isDes	1	1:是 0:否

● 接口返回数据解码后样例

```
{
  "taskId": "88499669-d69b-48be-88eb-d238e2857714",
  "vehicleList": [
    {
      "isDes": "0",
      "isOri": "0",
      "trackList": [
        {
          "agl": "82",
          "gtm": "20220117/170326",
          "hgt": "33",
          "lat": "14058765",
          "lon": "68061481",
          "mlg": "1253081",
          "spd": "650"
        },
        {
          "agl": "82",
          "gtm": "20220117/170341",
          "hgt": "33",
          "lat": "14058955",
          "lon": "68063013",
          "mlg": "1253084",
          "spd": "640"
        }
      ],
      "vco": "2",
      "vno": "陕 YH0009"
    }
  ]
}
```

车辆状态结果文件（每辆车是否成功分析历史业务）

数据以 json 格式返回

4. 技术服务支持

4.1. 术语和定义

◆ https

HTTPS（全称：Hyper Text Transfer Protocol over Secure Socket Layer），是以安全为目标的 HTTP 通道，简单讲是 HTTP 的安全版。即 HTTP 下加入 SSL 层，HTTPS 的安全基础是 SSL，因此加密的详细内容就需要 SSL。它是一个 URI scheme（抽象标识符体系），句法类同 http：体系。用于安全的 HTTP 数据传输。

◆ SDK（软件开发工具包）

软件开发工具包（外语首字母缩写：SDK、全称：Software Development Kit）一般都是一些软件工程师为特定的软件包、软件框架、硬件平台、操作系统等建立应用软件时的开发工具的集合。

4.2. 产品服务技术联调流程

产品服务通过 **https** 方式对外提供服务，遵循服务规范，发送 **https** 请求，支持 POST，数据交换服务将验证用户的合法性和安全性，然后提供服务，数据采用 UTF-8 格式编码。

4.2.1. 第一步：获取账密

由你方商务人员向我方商务人员发起申请，申请成功后将为您提供调用服务所需的服务账号、密码、私钥及客户端 ID 信息。

4.2.2. 第二步：编写服务

编写调用服务的客户端代码，按照服务定义的 **https** 调用方式及传输转换后的参数进行调用，调用示例代码详见附录 B 及调用示例工程。

4.2.3. 第三步：联调测试

联调测试环境域名为：`openapi-test.sinoiov.cn`，请根据文档中的示例参数或示例程序进行联调，联调环境为非正式数据，需使用我方指定的参数进行联调。服务调用成功且返回数据可正常解析后，表示联调成功。

联调过程中如遇到问题，请与我方运营人员联系，联调通过后如需使用正式环境则与我方商务人员联系，会为您提供正式环境域名及账号、密码、客户端 ID 等信息。

开放平台对外测试联调环境现对外提供如下：陕 YH0008、陕 YH0009，这两辆车为实时数据。车架号：LSSS6PEB6KT008136，这是陕 YH0009 的车架号。

4.3. 调用要求

4.3.1. 服务调用方式

产品服务通过 **https** 方式对外提供产品服务，遵循产品服务规范，发送 **https** 请求，支持 POST，数据格式为 `form`，数据交换产品服务将验证用户的合法性和安全性，然后提供产品服务，数据采用 UTF-8 格式编码。



4.3.2. 用户认证和服务安全

将根据第三方开发者访问的范围，进行用户授权等权限验证，保障用户数据的安全性。

用户首次调用产品服务，须先登录，认证通过后生成令牌。

令牌长期有效，登录后之前的令牌将立即失效，同一用户多部门调用产品服务时，建议由统一服务调用登录，将令牌缓存起来，多个服务统一从共享缓存中获取令牌。令牌失效后再调用登录，统一获取令牌，避免频繁调用登录，建议一天内登录次数不超过 10 次，超过 10 次将触发安全系统报警。

另外，为了您账号的安全，建议每月定时更新令牌。

根据国家法律法规，并考虑数据安全的需要，系统会有产品服务访问上限的限制。

示例：

请求地址格式

<https://openapi-test.sinoiov.cn/save/apis/login/>该地址为测试环境地址。

<https://openapi.sinoiov.cn/save/apis/login/>该地址为生产环境地址。

测试账号仅限测试联调环境使用，获取正式账号后则需更换正式环境地址，两套环境不混用。

参数列表

拼接参数	参数说明	参数值	是否必选
user	账号	xxx	是
pwd	密码	xxx	是

cid	客户端 id	xxx	是
srt	私钥	xxx	是

返回数据解码后样例：

```
{"result":"301f4556-f9bd-4e9c-9a3d-c0cd7a6ad7db","status":1001}
```

参数	说明
result	令牌
status	登录状态：详见附录 A

5. 常见问题

(1) 什么是令牌？令牌失效时间是多长？

答：令牌是指用户访问智运数字化产品的动态标示，长期有效，登录后会产生新令牌，旧的令牌会立即失效。为了您账号的安全，建议每月定时更新令牌，请勿频繁调用登录，否则会触发我们系统的安全防护体系。

(2) 你们的服务响应时间是多少？服务可承受的最大并发数是多少？

答：服务响应时间为百毫秒左右，我们的服务支持线性扩容，根据用户的需求会弹性增加。目前服务运行稳定，性能可靠，支持多线程并发调用。（“运输节点服务（批量）”每次可同时提交介质数为 100，建议并发 10 线程以下，如有特殊需求可专项沟通）

(3) 开通服务时，需要客户提供的客户端 IP 是指什么？

答：需要提供该 IP 的目的是用于确认请求是从客户合法的服务器发起，所以需要提供的是客户调用服务时的服务器外网出口 IP。

(4) 服务中返回的经纬度是什么坐标系？经纬度是否是偏移的？如何将你们返回的经纬度信息在其它平台的地图上进行准确展示？

答：服务中返回的坐标系是 84 坐标系（要除以 60 万），经纬度为车机终端上报的原始坐标，没有经过偏移处理，不同平台的地图都支持从 84 坐标系的经纬度转换到自家地图坐标系，相关问题请咨询地图提供商。客户使用的第三方地图，在解析地图时，需要按照该地图的标准做偏移处理，我们服务返回的是车机原始经纬度，没有经过偏移加工。

(5) 技术白皮书后提供的是 Java 示例，请问有别的语言示例吗？

答：签约后我方会为您提供专业的服务，白皮书中提供 Java 版本的代码示例，其他版本代码咨询及遇到技术问题时可以在工作群里及时沟通解决。在后续的数据展示环节我方也会给予一定帮助。关注我们平台微信公众号“车旺智运”以便了解更多关于产品服务及相关服务的动态及业务通知。

(6) 为什么会返回 1002 参数不正确？

答：首先检查是否按白皮书中调用示例操作的，有可能是程序员粗心引起的大小写错误、空格、

字符问题。还有以下情况：

- 1、字段拼接：文档中拼接参数为必选，输入时没加上该字段参数。
- 2、格式错误：车牌 京 A12345 一个汉字+六位字母数字，输入时缺位；文档中字段限定只有 1~9 可选，用户输入 10。
- 3、时间不正确：历史节点等需要输入时间段参数的服务，一次调用的时间段不能超过 24 小时；服务器上保存包含当月的 6 个自然月的信息，比如今天 6 月 13 日，可检索到 1 月 1 日的信息，超出或格式不对则报错。

（7）为什么会返回 1006 无结果？

答：返回 1006 两种情况：一是所检索的未入网；二是该条件下无信息，比如运输行程服务，某时间区间内，没有向平台上传信息，会返回无结果；或其他有输入条件参数的服务，无该类型的检索不到结果。

（生产环境检索真实车牌不能用测试环境域名）

（8）为什么返回 1010 用户名或密码不正确？

答：账户开通后用户名密码发送到客户邮箱，用户自己妥善保管。如有此情况请立即联系中交工作人员重置密码。

（9）为什么返回 1014 无权限？

答：无权限表示该服务没有开通，可以联系商务同事申请开通。

（10）为什么返回 1016 令牌失效？为什么会一下正常一下失效？

答：生成新令牌则旧令牌失效。

令牌频繁失效原因是用户有两个服务程序在获取令牌，甲程序获取新令牌使乙程序的令牌失效，乙程序继续获取令牌，则甲程序失效。

附录 A：状态码说明

状态码	说明
1001	服务执行成功
1002	参数不正确（参数为空、检索时间范围不正确、参数数量不正确）
1003	服务调用数量已达上限
1004	服务调用次数已达上限
1005	该账号未授权指定所属行政区划范围
1006	无结果
1010	用户名或密码不正确
1011	IP 不在白名单列表
1012	账号已禁用
1013	账号已过有效期
1014	无服务权限
1015	用户认证系统已升级，请使用令牌访问
1016	令牌失效
1017	账号欠费

1018	授权的服务已禁用
1019	授权的服务已过期
1020	该车调用次数已达上限
1021	client_id 不正确
1031	签名验证失败
1050	没有购买此类套餐
1051	套餐服务能力已用尽
1052	用户不存在
1053	服务无对应有效套餐
9001	系统异常
1060	新能源车辆已订阅，暂时无结果

附录 B: Java 示例代码

SDK 示例代码

```
package com.sinoiov.test;

import com.openapi.sdk.service.DataExchangeService;
import java.util.HashMap;
import java.util.Map;

/**
 * 使用 SDK 服务客户端调用示例（如不使用 SDK，可通过标准 HTTPS 方式调用服务，忽略正式信任。）
 * 该文件使用 UTF-8 编码格式
 */
public class DemoMain{
    /** 用户登陆
     * 用户首次调用服务，须先登录，认证通过后生成令牌。
     * 令牌，多服务调用业务时，建议由统一服务调用登录，将令牌缓存起来，多个服务统一从共享缓存中获取令牌。
     * 为安全考虑，建议用户自己每月至少更新一次令牌。
     * 令牌失效后再调用登录服务获取令牌，避免频繁调用登录服务，建议一天内登录次数不超过 10 次，超过 10 次将触发安全系统报警。
     */
    public void login() {
        try {
            Map<String, String> map = new HashMap<String, String>(4);
            map.put("user", "您的 API 账号");
            map.put("pwd", "您的 API 账号密码");
            map.put("srt", "您的私钥");
            map.put("cid", "您的客户端 ID");
            String url = "https://openapi-test.sinoiov.cn/save/apis/login";
            // 设置 http 读写超时
            DataExchangeService des = new DataExchangeService(5000, 8000);
            System.out.println("请求地址:"+url);
            // 通过 https 方式调用，此方法内部会使用私钥生成签名参数 sign, 私钥不会发送
            String res = des.postHttps(url, map);
            System.out.println("返回:"+ res);
        } catch (Exception e) {
            System.out.println("e: " + e.getMessage());
        }
    }

    /** 一、 运输节点服务
     * 本服务提供指定介质的运输节点服务。
     */
}
```

```

    * */
    public static void transTimeManage() {
        try {
            Map<String, String> map = new HashMap<String, String>(3);
            map.put("token", "您的令牌");
            map.put("cid", "您的客户端 ID");
            map.put("srt", "您的私钥");
            map.put("vcln", "陕 YH0009");
            map.put("timeNearby", "30");

            String url = "https://openapi-test.sinoiov.cn/save/apis/vLastLocationV3";

            DataExchangeService des = new DataExchangeService(5000, 8000);
            System.out.println("请求地址:"+url);
            // 通过 https 方式调用, 此方法内部会使用私钥生成签名参数 sign, 私钥不会发送
            String res = des.postHttps(url, map);
            System.out.println("返回:"+ res);
        } catch (Exception e) {
            System.out.println("e:" + e.getMessage());
        }
    }
}

```

非 SDK 示例代码-Java

```

1. import javax.crypto.Mac;
2. import javax.crypto.spec.SecretKeySpec;
3. import javax.net.ssl.*;
4. import java.io.BufferedReader;
5. import java.io.InputStreamReader;
6. import java.io.OutputStreamWriter;
7. import java.io.PrintWriter;
8. import java.net.URL;
9. import java.nio.charset.StandardCharsets;
10. import java.security.SecureRandom;
11. import java.security.cert.X509Certificate;
12. import java.util.*;
13.
14. /**
15.  * 示例程序
16.  */
17. public class KfptDemo {
18.     public static final String HMAC_SHA1 = "HmacSHA1";

```



```

19.     public static final char[] DIGITAL = "0123456789ABCDEF".toCharArray();
20.
21.     /**
22.      * <pre>
23.      * 请求流程:
24.      * 1、登录, 获取 token
25.      * 2、其他 API, token + 私钥 + 业务参数
26.      * </pre>
27.      */
28.     public static void main(String[] args) throws Exception {
29.         // URL 前缀
30.         String baseUrl = "https://openapi-test.sinoiov.cn/save/apis/";
31.         // 登录
32.         Map<String, String> loginParam = new HashMap<>();
33.         // 请填写自己的账号信息
34.         // 账号
35.         loginParam.put("user", "");
36.         // 密码
37.         loginParam.put("pwd", "");
38.         // 私钥
39.         loginParam.put("srt", "");
40.         // 客户端 id
41.         loginParam.put("cid", "");
42.         String loginUrl = baseUrl + "login";
43.         String loginResult = httpsCall(loginUrl, loginParam, false);
44.         // 此处需要将登录服务中返回的 token 存储下来, 以供其他服务使用
45.         System.out.println(loginResult);
46.
47.         // 检索最新节点
48.         String vLastLocationV3Url = baseUrl + "vLastLocationV3";
49.         Map<String, String> vLastLocationParam = new HashMap<>();
50.         // TOKEN, 从登录服务获取
51.         vLastLocationParam.put("token", "");
52.         // 私钥
53.         vLastLocationParam.put("srt", "");
54.         // 业务参数
55.         vLastLocationParam.put("vc1n", "陕 YH0009");
56.         vLastLocationParam.put("timeNearby", "24");
57.         String vLastLocationV3Result = httpsCall(vLastLocationV3Url, vLastLo
            cationParam, false);
58.         // 处理业务服务返回
59.         System.out.println(vLastLocationV3Result);
60.     }
61.

```

```

62.     /**
63.      * https 调用, 1、处理参数: ①移除 srt, ②计算签名, ③增加 sign; 2、将 map 参数拼接
        成 string 格式; 3、发起调用; 4、处理结果
64.      *
65.      * @param url    url
66.      * @param param   请求参数
67.      * @param strict 严格模式 true 校验证书等; false 不校验
68.      */
69.     public static String httpsCall(String url, Map<String, String> param, boolean strict) throws Exception {
70.         // 1、处理参数
71.         processParam(param);
72.         // 2、将 map 参数拼接成 string 格式
73.         String reqBody = convertMapToString(param);
74.         System.out.printf("Request - url: %s, reqBody: %s\n", url, reqBody);
75.         if (!strict) {
76.             // 信任证书
77.             trustAllCerts();
78.             // 信任域名
79.             trustAllHosts();
80.         }
81.         // 3、发起调用:
82.         HttpURLConnection conn = (HttpURLConnection) new URL(url).openConnection();
83.         conn.setDoInput(true);
84.         conn.setDoOutput(true);
85.         conn.setUseCaches(false);
86.         conn.setRequestMethod("POST");
87.         conn.setRequestProperty("Content-Length", String.valueOf(reqBody.getBytes(StandardCharsets.UTF_8).length));
88.         conn.setRequestProperty("charset", "UTF-8");
89.         conn.connect();
90.
91.         PrintWriter writer = new PrintWriter(new OutputStreamWriter(conn.getOutputStream(), StandardCharsets.UTF_8));
92.         // 注意这里不能有换行
93.         writer.print(reqBody);
94.         writer.flush();
95.         writer.close();
96.
97.         BufferedReader reader = new BufferedReader(new InputStreamReader(conn.getInputStream(), StandardCharsets.UTF_8));
98.         String line;

```

```

99.         StringBuilder result = new StringBuilder();
100.         while ((line = reader.readLine()) != null) {
101.             result.append(line);
102.         }
103.
104.         reader.close();
105.         System.out.printf("Response - code: %s, message: %s, respBody: %s\n",
            conn.getResponseCode(), conn.getResponseMessage(), result);
106.         // 4、处理结果
107.         return result.toString();
108.     }
109.
110.     /**
111.      * 处理参数: ①移除 srt, ②计算签名, ③增加 sign;
112.      *
113.      * @param param 原始参数
114.      */
115.     private static void processParam(Map<String, String> param) throws Exception {
116.         if (!param.containsKey("srt")) {
117.             throw new NullPointerException();
118.         }
119.         String srt = param.remove("srt");
120.         List<String> paramValueList = new ArrayList<>();
121.         for (Map.Entry<String, String> entry : param.entrySet()) {
122.             paramValueList.add(entry.getKey() + entry.getValue());
123.         }
124.         Collections.sort(paramValueList);
125.         String[] data = new String[paramValueList.size()];
126.         paramValueList.toArray(data);
127.         byte[] signature = hmacSha1(data, srt.getBytes(StandardCharsets.UTF_8));
128.         String sign = encodeHexStr(signature);
129.         param.put("sign", sign);
130.     }
131.
132.     /**
133.      * 将参数转换为字符串
134.      *
135.      * @param param 请求参数
136.      */
137.     private static String convertMapToString(Map<String, String> param) {
138.         StringBuilder sb = new StringBuilder();
139.         if (param != null && !param.isEmpty()) {

```

```

140.         for (Map.Entry<String, String> entry : param.entrySet()) {
141.             sb.append(entry.getKey()).append("=").append(entry.getValue
142.             ()).append("&");
143.         }
144.         sb.deleteCharAt(sb.length() - 1);
145.     }
146.     return sb.toString();
147. }
148. /**
149.  * 信任所有证书
150.  */
151. private static void trustAllCerts() throws Exception {
152.     TrustManager[] trustAllCerts = new TrustManager[]{new X509TrustMana
153.     ger() {
154.         public X509Certificate[] getAcceptedIssuers() {
155.             return null;
156.         }
157.         public void checkClientTrusted(X509Certificate[] certs, String
158.         authType) {
159.         }
160.         public void checkServerTrusted(X509Certificate[] certs, String
161.         authType) {
162.         }
163.     }};
164.     SSLContext sc = SSLContext.getInstance("SSL");
165.     sc.init(null, trustAllCerts, new SecureRandom());
166.     HttpsURLConnection.setDefaultSSLSocketFactory(sc.getSocketFactory())
167.     ;
168. }
169. /**
170.  * 信任所有域名
171.  */
172. private static void trustAllHosts() {
173.     HostnameVerifier trustAllHosts = (hostname, session) -> true;
174.     HttpsURLConnection.setDefaultHostnameVerifier(trustAllHosts);
175. }
176. /**
177.  * HmacSha1 定制算法
178.  *

```

```

179.     * @param strings 内容
180.     * @param key    key
181.     * @return 签名
182.     */
183.     private static byte[] hmacSha1(String[] strings, byte[] key) throws Exc
    eption {
184.         SecretKeySpec signingKey = new SecretKeySpec(key, HMAC_SHA1);
185.         Mac mac = Mac.getInstance(HMAC_SHA1);
186.         mac.init(signingKey);
187.         for (String data : strings) {
188.             mac.update(data.getBytes(StandardCharsets.UTF_8));
189.         }
190.         return mac.doFinal();
191.     }
192.
193.     /**
194.     * 字节转为 Hex 字符串
195.     *
196.     * @param bytes 字节数组
197.     * @return Hex 字符串
198.     */
199.     private static String encodeHexStr(final byte[] bytes) {
200.         if (bytes == null) {
201.             return null;
202.         }
203.         char[] result = new char[bytes.length * 2];
204.         for (int i = 0; i < bytes.length; i++) {
205.             result[i * 2] = DIGITAL[(bytes[i] & 0xf0) >> 4];
206.             result[i * 2 + 1] = DIGITAL[bytes[i] & 0x0f];
207.         }
208.         return new String(result);
209.     }
210. }
211.

```