

航空部分的接口文档

2019年10月7日 10:10

1. <http://{ip:port}/sectorCongestion>

- a. 接口说明：展示在某个月份内扇区的拥挤度情况。
- b. 请求参数：
 - i. date：必填，表示请求的月份。示例：2018-09-01，表示请求2018年9月整个的拥挤度情况
- c. 返回信息：以GeoJson格式返回，其中，geometry字段表示扇区的面数据，properties字段包含的信息如下：
 - i. sector_code：扇区代码
 - ii. name：扇区名称
 - iii. high_end：高度上限
 - iv. low_end：高度下限
 - v. count：在查询时间内，该扇区的飞机总量。
 - vi. level：经过阈值判断后，得到的拥挤度指数，0，1，2分别代表拥挤度低，中，高

d. 返回示例:

2. <http://{ip:port}/airrouteCongestion>

- a. 接口说明：请求航路的拥挤度情况
- b. 请求参数：
 - i. date：必填，表示请求的日期。示例：2019-08-23，表示请求该天的航路拥挤度信息
- c. 返回数据：以GeoJson格式返回，其中，geometry字段表示航路段，properties字段包含的信息如下：
 - i. airroute_id：航路段号。示例“XX0260_1”，其中，“_”符号前面的部分为民航局官方航路段，后面的“1”表示将XX0260航路段按折线点分割后的第一个线段
 - ii. airroute_name：该航路段的名称
 - iii. count：该时间段内飞机的数量
 - iv. level：根据阈值判断之后，得到的拥挤程度指数。0，1，2，3表示基本畅通，轻度拥堵，中度拥堵，重度拥堵

d. 返回示例

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": { "type": "LineString", "coordinates": [[109.395, 16.83333333], [110.3, 17.41583333]] },
      "properties": {
        "airroute_id": "XX0260_1",
        "airroute_name": "栎社NDB-嵊州VOR",
        "count": 455,
        "level": 1
      }
    },
    {
      "type": "Feature",
      "geometry": { "type": "LineString", "coordinates": [[110.3, 17.41583333], [112.245, 18.66166667]] },
      "properties": {
        "airroute_id": "XX0260_2",
        "airroute_name": "栎社NDB-嵊州VOR",
        "count": 320,
        "level": 0
      }
    }
  ],
  .....
}
```

3. <http://ip:port/flightTrack>

- 接口说明：获取航迹信息。前端自己按照时间间隔轮询请求，每次请求后端都只会发送符合所请求条件的航班的信息。
- 请求参数：
 - datetime: 可选，默认为当前时刻。示例：2018-10-08 22:12:22
 - transport_type: 可选，表示运输类型，客运/货运。默认为全部类型。使用“1”代表客运，“0”代表货运
 - aircraft_code_type: 可选，表示飞机的机型。默认为全部类型。示例：“747”
- 返回数据：以GeoJson格式返回，其中，geometry字段包含飞机的位置信息，properties字段包含的信息如下：
 - callsign: 航班号
 - latitude: 纬度
 - longitude: 经度
 - height: 高度
 - heading: 航向角
 - groundspeed: 地速
 - verticalrate: 上升(+)或下降(-)率
 - datetime: 时间戳

d. 返回示例:

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": { "type": "Point", "coordinates": [112.19917, 16.8875] },
      "properties": {
        "callsign": "AXM127",
        "latitude": 16.88750,
        "longitude": 112.19917,
        "height": 945,
        "heading": 216.86990,
        "groundspeed": 847,
        "verticalrate": 0.8,
        "datetime": "2018-10-01 04:27:57"
      }
    },
    {
      "type": "Feature",
      "geometry": { "type": "Point", "coordinates": [116.26, 15.63556] },
      "properties": {
        "callsign": "B8300",
        "latitude": 15.63556,
        "longitude": 111.26000,
        "height": 1219,
        "heading": 216.86240,
        "groundspeed": 856,
        "verticalrate": -0.2,
        "datetime": "2018-10-01 06:08:56"
      }
    },
    .....
  ]
}
```

4. <http://ip:port/flightDetailInfo>

- 接口说明：获取某一架航班的具体信息。主要的目的是减轻实施航班接口的数据量。只有在点选某个具体航班的时候，才会发起这样的一个请求，获取相对静态的数据。
- 请求参数：
 - callsign: 必填，航班号，示例：AXM127
 - date: 选填，日期，示例：2019-09-10。默认为当前日期。
- 返回数据：以Json格式返回，包含的字段说明如下：
 - departure: 起飞机场
 - arrival: 降落机场
 - aircraft_type: 机型
 - code_group: 飞机所属系列的代号
 - transport_type: 运行类型（客运/货运）
 - airline: 所属航空公司
 - airroute_plan: 显示计划航路全程，以GeoJson的格式返回
 - airroute_history: 显示已有航迹的全程

d. 返回示例:

```
{
  "departure": "南昌昌北",
  "arrival": "深圳宝安",
  "aircraft_type": "A320",
  "code_group": "320",
  "transport_type": "客运",
  "airline": "深圳航空公司",
  "airroute_plan": { "type": "LineString", "coordinates": [[82.89972222, 44.59972222], [82.03666667, 44.01166667], [81.32888889, 43.95805556]] },
  "airroute_history": { "type": "LineString", "coordinates": [[114.0344444, 22.70972222], [113.7416667, 22.92166667], [113.4180556, 23.15305556], [113.515, 23.41166667]] }
}
```

5. <http://ip:port/flightList>

- 接口说明：全国航班列表查询。返回的是所请求条件的全部飞行数据
- 请求参数：

- i. date: 选填, 表示要查询的航班日期, 默认为当前日期。示例: 2019-10-09
 - ii. dep_airport: 选填, 表示起飞机场的代号, 默认为全国所有机场。示例: ZUCK
 - iii. arr_airport: 选填, 表示降落机场的代号, 默认为全国所有机场。示例: ZULS
 - iv. callsign: 选填, 表示航班号, 默认为全部的航班信息。示例: CSN6680
 - v. page: 必填, 表示分页序号, 用于前端分页显示数据。
- c. 返回数据: 以GeoJson格式返回, 其中地理位置信息字段表示飞机的计划航路。properties字段包含的属性如下:
- i. callsign: 航班号
 - ii. departure: 起飞机场
 - iii. arrival: 降落机场
 - iv. dep_datetime_plan: 计划起飞时间
 - v. dep_datetime_real: 实际起飞时间
 - vi. arr_datetime_plan: 计划降落时间
 - vii. arr_datetime_real: 实际降落时间
 - viii. aircraft_type_plan: 计划执飞机型
 - ix. aircraft_type_real: 实际执飞机型
 - x. aircraft_code_plan: 计划执飞的飞机机尾号
 - xi. aircraft_code_real: 实际执飞的飞机机尾号
 - xii. flying_status: 是否处于飞行状态。true表示在飞, false表示落地
 - xiii. transport_type: 运输类型
 - xiv. airline: 所属航空公司
- d. 返回示例:

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": { "type": "LineString", "coordinates": [[82.89972222, 44.59972222], [82.03666667, 44.01166667], [81.32888889, 43.95805556]] },
      "properties": {
        "callsign": "CHH7012",
        "departure": "广州/白云",
        "arrival": "海口/美兰",
        "dep_datetime_plan": "2018-09-30 23:50:00",
        "dep_datetime_real": "2018-10-01 00:00:00",
        "arr_datetime_plan": "2018-10-01 00:57:00",
        "arr_datetime_real": "2018-10-01 01:05:00",
        "aircraft_type_plan": "B738",
        "aircraft_type_real": "B738",
        "aircraft_code_plan": "B1928",
        "aircraft_code_real": "B1928",
        "flying_status": true,
        "transport_type": "客运",
        "airline": "海南航空公司",
      }
    },
    .....
  ]
}
```

6. <http://ip:port/airportTakeoffandLandingRate>

- a. 接口说明: 全国机场起降率查询, 返回的是查询请求时间段内的全国机场的飞机起飞和降落的数量。
- b. 请求参数:
 - i. end_datetime: 必填。表示查询的结束时间。示例: 2018-10-01 00:00:00
 - ii. start_datetime: 选填, 表示查询的开始时间。默认为结束时间前一个小时。示例: 2018-09-30 23:00:00
- c. 返回数据: 以GeoJson数组的方式返回, 其中, geometry字段为机场的地理位置。properties字段包含的属性说明如下:
 - i. airport_ident: 机场代号
 - ii. airport_name: 机场名称
 - iii. takeoff_num: 起飞数量
 - iv. landing_num: 降落数量
- d. 返回数据示例:

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": { "type": "Point", "coordinates": [116.584999084, 40.0801010132] },
      "properties": {
        "airport_ident": "ZBAA",
        "airport_name": "北京首都",
        "takeoff_num": 30,
        "landing_num": 20
      }
    },
    {
      "type": "Feature",
      "geometry": { "type": "Point", "coordinates": [90.9119033813, 29.2978000641] },
      "properties": {
        "airport_ident": "ZULS",
        "airport_name": "拉萨贡嘎",
        "takeoff_num": 3,
        "landing_num": 2
      }
    }
  ]
}
```

```

    }
    .....
}
]
}

```

7. [http://\(ip:port\)/groundAircraft](http://(ip:port)/groundAircraft)

- a. 接口说明: 显示全国地面航班的情况。前端轮询, 时间间隔由前端决定, 一般考虑1min以上。
- b. 请求参数:
 - i. datetime: 必填, 表示查询的时间。默认为当前时间。格式示例: "2019-10-07 17:25:36"
- c. 返回参数: 以Json数组的方式返回, 每一个Json对象为一个机场的数据, 包含的属性字段如下:
 - i. airport_ident: 机场的代号
 - ii. position: 机场的位置, 以GeoJson的格式返回
 - iii. flights: 一个数组, 包含正停在机场的航班信息。数组的每一项为一个Json对象, 包含的字段信息如下:
 - 1) callsign: 将要执行的航班号
 - 2) arrival: 到达机场
 - 3) dep_datetime_plan: 计划起飞时间
 - 4) arr_datetime_plan: 计划到达时间
 - 5) aircraft_type_plan: 计划执飞的飞机型号
 - 6) code_group_play: 计划执飞的飞机所属系列型号
 - 7) aircraft_code_plan: 计划执飞的飞机机尾号
 - 8) transport_type_plan: 运输类型
 - 9) airline: 所属航空公司
- d. 返回示例:

```

[
  {
    "airport_ident": "ZBAA",
    "position": { "type": "Point", "coordinates": [116.5972222, 40.07222222] },
    "flights": [
      {
        "callsign": "CHH7012",
        "arrival": "海口/美兰",
        "dep_datetime_plan": "2018-09-30 23:50:00",
        "arr_datetime_plan": "2018-10-01 01:05:00",
        "aircraft_type_plan": "B738",
        "code_group_plan": "737",
        "aircraft_code_plan": "B1928",
        "transport_type_plan": "客运",
        "airline": "海南航空公司",
      },
      {
        "callsign": "UEA2728",
        "arrival": "温州/龙湾",
        "dep_datetime_plan": "2019-07-18 11:00:00",
        "arr_datetime_plan": "2019-07-18 12:40:00",
        "aircraft_type_plan": "A320",
        "code_group_plan": "320",
        "aircraft_code_plan": "B8878",
        "transport_type_plan": "货运",
        "airline": "中国邮政航空公司",
      }
    ]
  },
  {
    "airport_ident": "ZBZJ",
    "position": { "type": "Point", "coordinates": [114.9308333, 40.73916667] },
    "flights": [
      {
        "callsign": "CSN312",
        "arrival": "广州/白云",
        "dep_datetime_plan": "2019-07-17 13:30:00",
        "arr_datetime_plan": "2019-07-18 04:04:00",
        "aircraft_type_plan": "B77W",
        "code_group_plan": "777",
        "aircraft_code_plan": "B209Y",
        "transport_type_plan": "客运",
        "airline": "上海航空公司",
      }
    ]
  }
],
.....
]

```

8. [http://\(ip:port\)/capacity](http://(ip:port)/capacity)

- a. 接口说明: 运力测算。
- b. 请求参数:
 - i. start_datetime: 必填, 表示查询的开始时间。
 - ii. end_datetime: 必填, 表示查询的结束时间。
 - iii. airports: 必填, 表示起飞的机场, 或者查询的机场, 内容为一个机场代号的数组。
 - iv. destination: 可选, 表示考虑方向性, 从当前查询的机场到目的地机场。内容为一个机场代号的数组。
- c. 返回结果: 以Json数组的形式返回, 每一个Json对象, 包含的字段信息如下:
 - i. airport: 起飞机场的代号

ii. capacity: 运力情况。格式为一个Json数组，数组的每一项包含的字段说明如下：

1) aircraft_code_group: 飞机所属的机型系列

2) num: 架次数量

d. 返回结果示例:

```
[
  {
    "airport": "ZBAA",
    "capacity": [{ "aircraft_code_group": "737", "num": 28 }, { "aircraft_code_group": "320", "num": 20 }, { "aircraft_code_group": "747", "num": 15 } ]
  },
  {
    "airport": "ZBZJ",
    "capacity": [{ "aircraft_code_group": "747", "num": 2 }, { "aircraft_code_group": "738", "num": 5 }, { "aircraft_code_group": "321", "num": 6 } ]
  }
]
```