

## 平台反查企业数据接入文档

https:// 平台调地址 + order\_info ? params 订单  
https:// 平台调地址 + publish\_info ? params 发布方  
https:// 平台调地址 + service\_info ? params 服务方

平台反查，需平台方提供一个 **页面**

**页面**：不仅包括该模块儿在《《平台系统数据元素定义-无车承运》》文档中涉及到的**该模块儿的所有元素信息**，还可能会有一些企业暂时无法提供的元素信息(具体信息以局方要求为准)

例如：

①企业推送的照片是经过压缩之后的，清晰度可能就会有所降低，在反查的页面中，照片就是企业未经处理的原图；

②企业推送的银行方面信息，可能是服务方在企业的平台上注册的的虚拟账户，反查页面中应该既展示该虚拟账户信息，又展示服务方的真实银行方面信息，对于没有绑定银行信息只有虚拟账户的服务方，在反查页面中需要标明未绑定，并给出服务方对于虚拟账户中资金处理的方式

平台调用地址样例：<https://www.baidu.com/sspt/data/>

①精确到最后一个“/”

②sspt 为平台自定义个性名称 例子：[gxpt](#)

以下说明 地址 使用 样例

1.域名地址（对应下面链接中的 host）

测试：(平台方提供)

正式：(平台方提供)

2 公共请求参数

公共参数	是否必填	描述	示例
app_id	是	平台分配唯一 ID	6aad57d0be7c4658 a8aadec0ba08ecb8
time	是	请求时间戳（精确到秒）	1558679256
sign_key	是	加密请求参数后获得的 32 位字符串（英文字母小写）	8920b9b5797a44ec7d 3475f5675d83f9

注：平台需提供 app\_id 和 app\_key(详见下一页第 3, 4 项内容)

sign\_key 我方提供

2 页面对应地址 支持 **https**（请求方式 GET）

订单: [https://host/gxpt/data/order\\_info](https://host/gxpt/data/order_info)

样例:

[https://www.baidu.com/gxpt/data/order\\_info?app\\_id=xx&sign\\_key=xx&time=xxx&dduid=xxx](https://www.baidu.com/gxpt/data/order_info?app_id=xx&sign_key=xx&time=xxx&dduid=xxx)

业务参数	是否必填	描述	示例
dduid	是	订单唯一 ID	

发布方: [https://host/gxpt/data/publish\\_info](https://host/gxpt/data/publish_info)

样例:

[https://www.baidu.com/gxpt/data/publish\\_info?app\\_id=xx&sign\\_key=xx&time=xxx&fbfuid=xxx](https://www.baidu.com/gxpt/data/publish_info?app_id=xx&sign_key=xx&time=xxx&fbfuid=xxx)

业务参数	是否必填	描述	示例
fbfuid	是	发布方唯一 ID	

服务方信息: [https://host/gxpt/data/service\\_info](https://host/gxpt/data/service_info)

样例:

[https://www.baidu.com/gxpt/data/service\\_info?app\\_id=xx&sign\\_key=xx&time=xxx&fwfuid=xxx](https://www.baidu.com/gxpt/data/service_info?app_id=xx&sign_key=xx&time=xxx&fwfuid=xxx)

业务参数	是否必填	描述	示例
fwfuid	是	用户位置 ID	

### 3.sign\_key 加密方式

```
if(type==0)data.param = "app_id-"+appId+"dduid-"+xuuid+"time-"+time;
if(type==1)data.param = "app_id-"+appId+"fbfuid-"+xuuid+"time-"+time;
if(type==2)data.param = "app_id-"+appId+"fwfuid-"+xuuid+"time-"+time;
```

```
data.appKey = appKey;
data.appId = appId;
```

```
$.ajax({
    type: "post",
    data: data,
    success: function (signKey) {
        if(signKey){
            if(type == 0){ // 订单
                url = url+"order_info?";
                param = "app_id="+appId+"&sign_key="+signKey+"&time="+time+"&dduid="+xuuid;
```

```
public String (...)(@RequestParam Map<String, Object> map) {
```

```
    if(!map.containsKey("appId") || !map.containsKey("appKey")) {
        return null;
    }
```

```
    String str = map.get("param").toString();
```

```
    str = str + map.get("appKey").toString();
```

```
    str = Md5.getMd5(str.getBytes());
```

```
    return str;
```

```
}
```

```

public class Md5 {
    public static String getMD5(byte[] source) {
        String s = null;
        char hexDigits[] = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9',
            'a', 'b', 'c', 'd', 'e', 'f' };
        try {
            java.security.MessageDigest md = java.security.MessageDigest
                .getInstance("MD5");
            md.update(source);
            byte tmp[] = md.digest();
            char str[] = new char[16 * 2];
            int k = 0;
            for (int i = 0; i < 16; i++) {
                byte byte0 = tmp[i];
                str[k++] = hexDigits[byte0 >>> 4 & 0xf];

                str[k++] = hexDigits[byte0 & 0xf];
            }
            s = new String(str);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return s;
    }
}

```

**注意：**

加密内容的格式要求(现阶段)：

app\_id-xxxduuid-xxxtime-xxxapp\_key (订单)

app\_id-xxxfbuuid-xxxtime-xxxapp\_key (发布方)

app\_id-xxxfwuuid-xxxtime-xxxapp\_key (服务方)

加密时候的 time 应和公共参数传的 time 一致。

## 4. app\_id 和 app\_key

app\_id : 32 位随机字符串

app\_key : 16 位随机字符串

例：

app\_id : 6aad57d0be7c4658a8aadec0ba08ecb8

app\_key : a47384c45a231b3f