**浙江爱信诺细分说明书**

**功能：获取出租行程单信息接口**

**版本号：V1.0**

**受控状态：受控**

**编写：陆健辉**

**审核：**

**批准：**

修订记录

类别：A-增加；M-修改；D-删除

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 修订日期 | 版本号 | 修订类别 | 描述 | 修改人 |
| 2019-08-2 | v1.0 | A |  | 陆健辉 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 接口定义

## 扫码生成地址规则

**[https://fp.of1.cn/business/(行业类型)-(平台ID)?i=(企业税号)！(交易订单号](https://fp.of1.cn/k.action?i=行业类型!平台ID!交易订单号))**

行业类型：001（出租车行业 抛弃之前CZ）

平台ID：001（正尚出租车平台）

交易订单号：车牌号（去掉省份简称）+时间（yyyyMMddHHmms）如：AH95F820190701121121

ENCRYPTStr=encrypt((交易订单号), key) （按之前加密算法加密，加密函数见2.1）

## 拉取行业交易详细信息

**接口URL: /invoice/pull/trade-business-data**

### **请求参数**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 是否必须 | 示例值 | 描述 |
| 1 | tradeNo | String | 是 |  | 交易订单号（密文）  车牌+时间（yyyyMMddHHmmss） |
| 2 | businessPlatformId | String | 是 |  | 行业平台ID(平台方，可以校验此ID是否与其分配的ID相同，只处理相同ID的请求) |
| 3 | taxNo | String | 否 |  | 交易订单为密文是必须传 |

### **请求报文格式示例(Json格式)**

{

"tradeNo": "51ceb23364b29b921c47cb3e9e821cf145daa13e03c1bb04",

"businessPlatformId": "339901999999142",

"taxNo": "339901999999142"

}

### **响应参数**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 示例值 | 描述 | |
| 1 | resultCode | String |  | 结果码（0000：成功，其它均为失败） | |
| 2 | resultMsg | String |  | 结果信息 | |
| 3 | resultBody | Object |  | 返回的结果数据对象 | |
| **返回的结果数据(resultBody)** | | | | | |
| 1 | businessType | String |  | 行业类型（出租车行业类型：001） | |
| 2 | businessPlatformId | String |  | 交易所属行业平台ID | |
| 3 | businessTradeData | String  **/**Ojbect |  | 行业交易详细数据对象(加密的Json字符串/对象) | |
| 4 | cipherType | String |  | 加解密类型：01：DES（目前正尚在用的）  加密非必须，不传或null时“businessTradeData”为对象 | |
| **出租车行业交易详细数据（businessTradeData）** | | | | |
| 1 | tradeNo | String |  | 交易订单号  车牌+时间（yyyyMMddHHmmss） | |
| 2 | payeeTaxNo | String |  | 销方税号(收款方税号) | |
| 3 | amount | String |  | 金额（换算成元） | |
| 4 | licensePlateNo | String |  | 车牌号 | |
| 5 | deviceNo | String |  | 行程设备号 | |
| 6 | startTime | String |  | 上车时间(yyyyMMddHHmmss) | |
| 7 | endTime | String |  | 下车时间(yyyyMMddHHmmss) | |
| 8 | startSite | String |  | 上车地点 | |
| 9 | endSite | String |  | 下车地点 | |
| 10 | startLongitude | String |  | 上车经度 | |
| 11 | startLatitude | String |  | 上车维度 | |
| 12 | startAzimuth | String |  | 上车方位角 | |
| 13 | endLongitude | String |  | 下车经度 | |
| 14 | endLatitude | String |  | 下车维度 | |
| 15 | endAzimuth | String |  | 下车方位角 | |
| 16 | unitPrice | String |  | 单价（换算成元） | |
| 17 | mileage | String |  | 里程（换算成公里） | |
| 18 | waitingTime | String |  | 等候时间（换算成秒） | |
| 19 | travelStatus | String |  | 行程状态 0：行程中，1：行程结束，2：行程异常 | |

### **响应报文格式示例(Json格式)**

加密后，返回的格式

{

"resultCode": "0000",

"resultMsg": "成功",

"resultBody": {

"businessType": "001",

"businessPlatformId": "001",

"businessTradeData": ""

}

}

不加密，返回的格式

{

"resultCode": "0000",

"resultMsg": "成功",

"resultBody": {

"businessType": "001",

"businessPlatformId": "001",

"businessTradeData": {

"tradeNo": "AH95F820190701121121",

"payeeTaxNo": "339901999999142",

"amount": "29",

"licensePlateNo": "浙AH95F8",

"deviceNo": "EFW1324535677",

"startTime": "20190701121151",

"endTime": "20190701125551",

"startSite": "双龙街金色西溪",

"endSite": "城西银泰",

"startLongitude": "104.07642",

"startLatitude": "38.6518",

"startAzimuth": "166.32",

"endLongitude": "104.17642",

"endLatitude": "38.7518",

"endAzimuth": "16.32'",

"unitPrice": "2.5",

"mileage": "21.34",

"travelStatus": "1"

}

}

}

## 推送行业交易详细信息

**接口URL: /invoice/push/trade-business-data**

### **请求参数**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **名称** | **类型** | **示例值** | | **描述** |
| 1 | businessType | String |  | | 行业类型（出租车行业类型：001） |
| 2 | businessPlatformId | String |  | | 交易所属行业平台ID | |
| 3 | businessTradeData | String  **/**Ojbect |  | 行业交易详细数据对象(加密的Json字符串/对象) | | |
| 4 | cipherType | String |  | 加解密类型：01：DES（目前正尚在用的）  加密非必须，不传或null时“businessTradeData”为对象 | | |
| **出租车行业交易详细数据（businessTradeData）** | | | | | |
| 1 | tradeNo | String |  | | 交易订单号  车牌+时间（yyyyMMddHHmmss） |
| 2 | payeeTaxNo | String |  | | 销方税号(收款方税号) |
| 3 | amount | String |  | | 金额 |
| 4 | licensePlateNo | String |  | | 车牌号 |
| 5 | deviceNo | String |  | | 行程设备号 |
| 6 | startTime | String |  | | 上车时间(yyyyMMddHHmmss) |
| 7 | endTime | String |  | | 下车时间(yyyyMMddHHmmss) |
| 8 | startSite | String |  | | 上车地点 |
| 9 | endSite | String |  | | 下车地点 |
| 10 | startLongitude | String |  | | 上车经度 | |
| 11 | startLatitude | String |  | | 上车维度 | |
| 12 | startAzimuth | String |  | | 上车方位角 | |
| 13 | endLongitude | String |  | | 下车经度 | |
| 14 | endLatitude | String |  | | 下车维度 | |
| 15 | endAzimuth | String |  | | 下车方位角 | |
| 16 | unitPrice | String |  | | 单价（换算成元） |
| 17 | mileage | String |  | | 里程（换算成公里） |
| 18 | waitingTime | String |  | | 等候时间（换算成秒） |
| 19 | travelStatus | String |  | | 行程状态 0：行程中，1：行程结束，2：行程异常 | |

### **请求报文格式示例(Json格式)**

加密后，请求的格式

{

"businessType": "001",

"businessPlatformId": "001",

"businessTradeData": ""

}

不加密，请求的格式

{

"businessType": "001",

"businessPlatformId": "001",

"businessTradeData": {

"tradeNo": "AH95F820190701121121",

"payeeTaxNo": "339901999999142",

"amount": "29",

"licensePlateNo": "浙AH95F8",

"deviceNo": "EFW1324535677",

"startTime": "20190701121151",

"endTime": "20190701125551",

"startSite": "双龙街金色西溪",

"endSite": "城西银泰",

"startLongitude": "104.07642",

"startLatitude": "38.6518",

"startAzimuth": "166.32",

"endLongitude": "104.17642",

"endLatitude": "38.7518",

"endAzimuth": "16.32",

"unitPrice": "2.5",

"mileage": "21.34",

"travelStatus": "1"

}

}

### **响应参数**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **示例值** | **描述** |
| resultCode | String |  | 结果码（0000：接收成功，其它均为接收失败） |
| resultMsg | String |  | 结果信息 |

### **请求报文格式示例(Json格式)**

{

"resultCode": "0000",

"resultMsg": "接收数据成功"

}

# 加密算法定义

## 加解密类型：01的加解密类

import javax.crypto.Cipher;

import javax.crypto.SecretKey;

import javax.crypto.SecretKeyFactory;

import javax.crypto.spec.DESKeySpec;

import java.security.SecureRandom;

public class ZSDESUtil {

private static final String CHARSET = "UTF-8";

/\*\*

\* 加密算法

\*/

private static final String ALGORITHM = "DES";

private static String byte2hex(byte[] bytes) {

StringBuilder stringBuilder = new StringBuilder();

for (byte b : bytes) {

String s = Integer.toHexString(b & 0XFF);

if (s.length() == 1) {

stringBuilder.append("0").append(s);

} else {

stringBuilder.append(s);

}

}

return stringBuilder.toString().toLowerCase();

}

private static byte[] encrypt(byte[] data, byte[] key) throws Exception {

SecureRandom sr = new SecureRandom();

DESKeySpec dks = new DESKeySpec(key);

SecretKeyFactory keyFactory = SecretKeyFactory.getInstance(ALGORITHM);

SecretKey secretKey = keyFactory.generateSecret(dks);

Cipher cipher = Cipher.getInstance(ALGORITHM);

cipher.init(Cipher.ENCRYPT\_MODE, secretKey, sr);

return cipher.doFinal(data);

}

/\*\*

\* 对用DES加密过的数据进行加密.

\*

\* @param data 待加密字符串

\* @param key 密钥

\* @return 加密后的字符串

\* @throws Exception Exception

\*/

public static String encrypt(String data, String key) throws Exception {

return byte2hex(encrypt(data.getBytes(CHARSET), key.getBytes(CHARSET)));

}

public static void main(String[] args) {

String key = "12345678abcdefgh";

try {

String encryptValue = encrypt("5.8:328:2.5:22.0",key);

System.out.println(encryptValue);

} catch (Exception e) {

e.printStackTrace();

}

}

}