



一键通服务端接口文档

作者：北京创世华信科技有限公司

完成日期：2019-11-10

更新日期：2019-12-3

响应：bool

请求地址：<http://120.133.26.47:15001/v1/send>

请求头：contentType="application/json ; charset=utf-8"

请求方式：post

1.1 参数说明

参数名称	是否必须	类型	描述	说明
Appkey	是	string	应用唯一标识	
DataContent	是	string	AES 加密令牌	Json = {"oclToken": "令牌"}; Key = MessageSecret AES(json, key)
Signature	是	string	签名, MessageSecret 为用户请求加密的密钥	string sign=Appkey+TimeStamp+MessageSecret+Nonce; Md5(sign) 【注】“+” 仅表示连接
TimeStamp	是	string	UTC 时间戳精确到秒	当前 UTC 时间-(1970-01-01 00:00:00) 时间间隔转换为秒 数的字符串
Nonce	是	string	UUID	可以使用 guidid+时间戳

1.2 成功返回值 operatorType: 手机运营商 CT 电信, CU 联通, CM 移动

```
{
  "success": true,
  "result": {
    "phoneNum": "15754851326",
    "operatorType": "CM"
  },
  "errorCode": null,
  "errorMsg": null,
}
```

1.3 失败返回值

```
{
```



```
"status":false,
"message":"SignatureError",
"body":null
}
```

1.4 错误码

错误码	说明
Unknown	未知
OK	成功
AppConfigNotError	应用配置信息错误
SignatureError	签名错误
RequestInvalid	请求失效
DecodeError	解析错误
ExceedRate	频率限制
ChannelValidFail	运营商验证失败
ParamValidNotPass	参数校验不通过
AppKeyError	AppKey 错误
TimeStampError	时间戳错误
Arrears	余额不足
SystemError	系统错误

1.5 AES 加解密

IV="DEVICE-AES000000"

```
/**
 * 加密
 */
public static string Encrypt(string toEncrypt, string key, string iv)
{
    byte[] keyArray = UTF8Encoding.UTF8.GetBytes(key);
    byte[] ivArray = UTF8Encoding.UTF8.GetBytes(iv);
    byte[] toEncryptArray = UTF8Encoding.UTF8.GetBytes(toEncrypt);
    RijndaelManaged rDel = new RijndaelManaged();
    rDel.Key = keyArray;
    rDel.IV = ivArray;
    rDel.Mode = CipherMode.CBC;
    rDel.Padding = PaddingMode.PKCS7;
    ICryptoTransform cTransform = rDel.CreateEncryptor();
    byte[] resultArray = cTransform.TransformFinalBlock(toEncryptArray, 0, toEncryptArray.Length);
    return Convert.ToBase64String(resultArray, 0, resultArray.Length);
}
```



一键通服务端接口文档

```
/**
 * 解密
 */
public static string Decrypt(string toDecrypt, string key, string iv)
{
    byte[] keyArray = UTF8Encoding.UTF8.GetBytes(key);
    byte[] ivArray = UTF8Encoding.UTF8.GetBytes(iv);
    byte[] toEncryptArray = Convert.FromBase64String(toDecrypt);
    RijndaelManaged rDel = new RijndaelManaged();
    rDel.Key = keyArray;
    rDel.IV = ivArray;
    rDel.Mode = CipherMode.CBC;
    rDel.Padding = PaddingMode.PKCS7;
    ICryptoTransform cTransform = rDel.CreateDecryptor();
    byte[] resultArray = cTransform.TransformFinalBlock(toEncryptArray, 0,
toEncryptArray.Length);
    return UTF8Encoding.UTF8.GetString(resultArray);
}
```