

# FreeYunApi文档

## 更新版本

- 1.0版本
  - 2021-12-25 22:50:54 更新
  - 增加账号解绑接口

## API列表

### CloudInit

该请求可获取到软件的初始化信息，如软件公告、软件版本号、是否有更新等信息

```
1  /** 传入的初始化信息
2  typedef struct ANTI_FREEYUN_INIT_INFO
3  {
4      std::string Version;           //当前版本号
5      std::string SecretKey;        //软件密钥
6      std::string Rc4Key;           //RC4密钥
7      std::string SaltKey;          //签名盐
8      std::string AppId;            //软件ID
9      std::string MachineID;        //机器码
10     int    ServerLine;             //服务器线路
11
12     ANTI_FREEYUN_INIT_INFO(std::string Version, std::string SecretKey,
13     std::string Rc4Key, std::string SaltKey, std::string AppId, std::string
14     MachineID,int ServerLine)
15     {
16         this->AppId      = AppId;
17         this->Version     = Version;
18         this->SecretKey  = SecretKey;
19         this->Rc4Key     = Rc4Key;
20         this->MachineID  = MachineID;
21         this->SaltKey    = SaltKey;
22
23         this->ServerLine = ServerLine;
24     }
25 }TAG_ANTI_FREEYUN_INIT_INFO, * PTAG_ANTI_FREEYUN_INIT_INFO;
```

```
1  std::tuple<bool,std::string, nlohmann::json>
   CloudInit(PTAG_ANTI_FREEYUN_INIT_INFO pInfo);
```

返回值 元祖

std::get<0> 返回状态 `bool` = true; 初始化成功, `bool` = false; 调用This->GetErrorCode() 返回错误码.

std::get<1> 状态码对应的字符串,也可以调用This->GetErrorStr(errcode); 来获取,都是一样的

std::get<2> 初始化成功后返回json格式.包含以下字段

```

1  "code": 1003, //返回码
2  "timestamp": 1511680582235, //服务器的时间戳
3  //如果code返回不是初始化成功则下面的几个参数不返回，只返回上面两个参数
4  "nowVersion": "v1.0", //当前的版本号
5  "lastVersion": "v1.0", //最新的版本号
6  "needUpdate": 0, //是否强制更新
7  "md5": "9802468381EE92BA0F8B6CCECE1A9A4C", //当前版本的md5
8  "notic": "这里是软件的公告", //软件的公告
9  "baseData": "这里是软件的基础数据" //软件的基础数据

```

## CloudLogin

该接口请求用于账号密码的登陆，将返回token，用于需要认证的接口鉴权访问

```

1  std::tuple<bool, std::string, nlohmann::json> CloudLogin(std::string Account,
   std::string Password, std::string md5);

```

@param:md5 主程序MD5

std::get<2> 初始化成功后返回json格式.包含以下字段

```

{"code":1014,"timestamp":1640438941129,"token":"B6E81D69408D4FF096ECB0D83F17B8D
A"}

```

登陆成功后,会自动把token保存到m\_Token成员变量内

## CloudReg

该接口用户账号模式的账号密码注册

```

1  //注册用户
2  typedef struct ANTI_FREEYUN_REG
3  {
4      std::string Account;           //账号
5      std::string Password;         //密码
6      std::string QQ;               //qq
7      std::string Email;            //邮箱
8      std::string Mobile;           //手机
9      std::string InvitingCode;     //邀请码
10     std::string AgentCode;         //代理账号
11
12
13     ANTI_FREEYUN_REG(std::string Account, std::string Password, std::string
QQ, std::string Email, std::string Mobile, std::string InvitingCode,
std::string AgentCode)
14     {
15         this->Account      = Account;
16         this->Password     = Password;
17         this->QQ           = QQ;
18         this->Email        = Email;
19         this->Mobile       = Mobile;
20         this->InvitingCode = InvitingCode;
21         this->AgentCode    = AgentCode;
22     }

```

```

23     ANTI_FREEYUN_REG(std::string Account, std::string Password)
24     {
25         this->Account      = Account;
26         this->Password     = Password;
27         this->QQ           = "";
28         this->Email        = "";
29         this->Mobile       = "";
30         this->InvitingCode = "";
31         this->AgentCode    = "";
32     }
33
34
35 }TAG_ANTI_FREEYUN_REG, * PTAG_ANTI_FREEYUN_REG;
36 std::tuple<bool, std::string, nlohmann::json> CloudReg(PTAG_ANTI_FREEYUN_REG
pInfo);

```

- 1 -返回的json Code
- 2 1001、表示软件不存在或软件密钥不对
- 3 1043、表示软件关闭了注册功能
- 4 1004、表示该用户已经存在了
- 5 1005、表示机器码已经被注册了，如果开启了一台机器只能注册一个，如果机器码被注册会返回该值
- 6 1006、表示账号注册成功

## CloudPay

该接口用于账号模式的账号充值，不可用于单码模式的充值

```

1 std::tuple<bool, std::string, nlohmann::json> CloudPay(std::string Account,
std::string cardNo);

```

- 1 1001、表示软件不存在或软件密钥不对
- 2 1027、表示充值卡不存在
- 3 1045、表示充值卡被封停
- 4 1028、表示充值卡已被使用
- 5 1008、表示用户不存在
- 6 1029、表示充值成功

## CloudQueryUserInfo

用于查询用户信息

```

1 std::tuple<bool, std::string, nlohmann::json> CloudQueryUserInfo(std::string
Account);

```

- 1001、表示软件不存在或软件密钥不对
- 1015、账号登陆过期
- 1016、用户登陆后被删除即账号不存在
- 1017、用户信息查询成功

成功后返回的JSON信息

```
1 | {"account": "121231", "code": 1017, "email": "", "invitingcode": "w18nso", "mobile": "", "point": 0, "qq": "", "timeout": "2021-12-25 21:51:22", "timestamp": 1640439826604}
```

## CloudBlackLst

该接口检测到破解者后拉入机器码或IP到黑名单，后端将不返回任何数据

```
1 | std::tuple<bool, std::string, nlohmann::json> CloudBlackLst(std::string Account, ULONG Type = 2);
```

@Param:Type 加入黑名单的类型 1 = IP黑名单 2 = 机器码黑名单

## CloudChangePassword

该接口可用于账号密码模式的密码修改

```
1 | std::tuple<bool, std::string, nlohmann::json> CloudChangePassword(std::string Account, std::string oldPwd, std::string newPwd);
```

```
1 | code
2 | 1001、软件不存在或密钥错误
3 | 1008、用户不存在
4 | 1025、修改密码原始密码错误
5 | 1026、密码修改成功
```

## CloudExit

退出登录

```
1 | std::tuple<bool, std::string, nlohmann::json> CloudExit(std::string Account);
```

## CloudGetVersionInfo

取版本信息或更新

```
1 | std::tuple<bool, std::string, nlohmann::json> CloudGetVersionInfo();
```

```
1 | 返回的JSON
2 | {"code": 1033, "describe": "更新", "md5": "", "name": "时间模式", "url": ""}
```

## CloudGetPayCardList

取充值卡列表

```
1 | std::tuple<bool, std::string, nlohmann::json> CloudGetPayCardList();
```

```
1 {"cardList":[{"id":9031,"name":"月卡","price":12000,"value":3},
{"id":8420,"name":"年卡","price":1215752092,"value":100},{"id":8419,"name":"月卡","price":4000,"value":1},{"id":8406,"name":"周卡","price":1000,"value":1},
{"id":8405,"name":"天卡","price":500,"value":1}],"code":1035,"timestamp":1640440776576}
```

## CloudGetUserStatus

取用户状态接口

```
1 std::tuple<bool, std::string, nlohmann::json> CloudGetUserStatus(std::string
Account);
```

- 1 1001、软件不存在或密钥错误
- 2 1015、账号登陆过期
- 3 1016、用户登陆后被删除即账号不存在
- 4 1039、用户状态正常
- 5 1040、用户被锁定
- 6 1011、账号过期
- 7 1012、账号点数不足

## CloudUserSubPoint

客户端扣点

扣点数 非必传，扣点数必须大于0，如不传则默认扣软件设置的值

```
1 std::tuple<bool, std::string, nlohmann::json> CloudUserSubPoint(std::string
Account, int value);
```

- 1 1001、软件不存在或密钥错误
- 2 1015、账号登陆过期
- 3 1041、该软件为时间验证不支持客户端扣点
- 4 1016、用户登陆后被删除即账号不存在
- 5 1012、账号点数不足
- 6 1042、客户端扣点成功

```
1 false errorCode:1041 该软件为时间验证不支持客户端扣点
```

## CloudCardLogin

卡号登陆

与账号密码登陆的返回值一样

```
1 std::tuple<bool, std::string, nlohmann::json> CloudCardLogin(std::string
Card, std::string md5);
```

```
1 {"code":1014,"timestamp":1640441503180,"token":"1D2C39F3AACE403E81A83C72E1715
136"}
```

## CloudExecTelnetCode

### 执行远程代码

@Param:用户的账号

@Param:标签名

@Param:js函数方法名

@Param:远程js代码的参数,如果js代码无参数可不填, 如: function test(a,b);则这里参数填写格式为: 参数类型=参数值,参数类型=参数值, 注意每个参数格式为: 数据类型=参数, 每个参数用英文", "隔开, 支持的数据类型有: 1、整数型、2、文本型、3、长整型、4、双精度型、5、单精度型。(1=15,1=10)

```
1 std::tuple<bool, std::string, nlohmann::json> CloudExecTelnetCode(std::string Account, std::string labName, std::string funcName, std::string params);
```

- 1 1001、表示软件不存在或软件密钥不对
- 2 1015、表示账号登陆过期
- 3 1044、表示远程代码调用次数已达上限
- 4 1020、表示远程JS代码不存在
- 5 1021、表示远程JS执行失败
- 6 1022、表示远程JS执行成功

### 例:

```
info = yun.CloudExecTelnetCode("0114F37E0AFA5CE9E2F9A48A8BE5682E",  
"test","myFunction","1=15,1=10");
```

```
1 {"code":1022,"result":"25.0","timestamp":1640441504553}
```

## CloudGetTeletVar

### 取远程变量

```
1 std::tuple<bool, std::string, nlohmann::json> CloudGetTeletVar(std::string Account, std::string keyName);
```

- 1 1001、表示软件不存在或软件密钥不对
- 2 1015、表示账号登陆过期
- 3 1019、表示远程变量获取成功
- 4 1018、表示远程变量不存在

```
1 {"code":1019,"timestamp":1640441813162,"variable":"75 10 E8 ?? ?? ?? ?? E8 ??  
?? ?? ?? 8B 0D ?? ?? ?? ??"}
```

## CloudHeartBeat

### 心跳维持

```
1 std::tuple<bool, std::string, nlohmann::json> CloudHeartBeat(std::string Account);
```

- 1 1001、表示软件不存在或软件密钥不对
- 2 1015、表示账号登陆过期
- 3 1046、表示心跳成功

```
1 {"code":1046,"timestamp":1640443109087}
```

## CloudChangeMachine

修改机器码

```
1 std::tuple<bool, std::string, nlohmann::json> CloudChangeMachine(std::string Account, std::string Password = "");
```

- 1 1001、表示软件不存在或软件密钥不对
- 2 1015、表示账号登陆过期
- 3 1008、表示用户不存在
- 4 1009、表示登陆密码有误
- 5 1031、表示机器码已是绑定，无需更改
- 6 1012、表示账号点数不足
- 7 1011、表示账号过期
- 8 1032、表示软件换绑成功

## CloudFeedback

留言反馈

@Param:留言内容

@Param:联系方式

```
1 std::tuple<bool, std::string, nlohmann::json> CloudFeedback(std::string Context, std::string links);
```

- 1 1001、表示软件不存在或软件密钥不对
- 2 1023、表示5分钟内你已提交留言，无需再次提交
- 3 1024、表示留言反馈成功

## CloudUploadClientExceptionInfo

上传客户端异常信息

@Param:异常标签

@Param:异常内容

@Param:操作系统

--测试发现 返回为空 但是后台显示上传成功了！

```
1 std::tuple<bool, std::string, nlohmann::json> CloudUploadClientExceptionInfo(std::string ExceptionTag, std::string Context, std::string operaSystem);
```

## CloudGetUserPermission

取用户权限

该接口用于取后台设置的用户角色对应的程序权限标识，（备注：所有注册用户默认为普通用户权限为空）

```
1 | std::tuple<bool, std::string, nlohmann::json> CloudGetUserPermission(std::string Account);
```

```
1 | {"authority":"2","code":1048,"timestamp":1640443375692}
```

## CloudRemoteAlgRelay

远程算法转发

@Param:账号

@Param:远程ID

@Param:请求的参数

```
1 | std::tuple<bool, std::string, nlohmann::json> CloudRemoteAlgRelay(std::string Account, std::string remoteId, std::string params);
```

## CloudUsersOnlineCount

取用户在线人数

```
1 | std::tuple<bool, std::string, nlohmann::json> CloudUsersOnlineCount();
```

```
1 | {"code":1054,"onlineNum":44,"timestamp":1640443565628}
```

## CloudUserUnBind

账号解绑

```
1 | std::tuple<bool, std::string, nlohmann::json> CloudUserUnBind(std::string Account, std::string Password = "");
```

```
1 | {"code":1032,"timestamp":1640443613816}
```

## 返回码列表



-1000	接口弃用-请更新	该返回码系统不可更改返回码，记得做处理
-1001	该服务已经停止	该返回码表示调用的该接口功能已经下线不再提供服务或该接口弃用需要更新
-1	软件尚未初始化	该返回码表示客户端未执行【FreeYun初始化】进行参数设置初始化
-106	表示客户端网络异常	
-107	表示网络数据包错误	表明数据包非法串改，或者sdk升级
1012	账号点数不足	
1011	账号过期	
1028	充值卡已被使用	
1045	充值卡被封停	
1053	该充值卡不允许降级充值	
1027	充值卡不存在	
1034	该软件卡种类为空，请后台添加	
1036	卡种类型不存在	
1042	客户端扣点成功	
1024	留言反馈成功	
1021	远程JS执行失败	
1044	远程代码调用次数已达上限	
1022	远程JS执行成功	
1023	5分钟内你已提交留言，无需再次提交	
1046	心跳成功	
1003	软件获取初始化信息成功	
1020	远程JS代码不存在	
1013	软件使用人数已达多开上限	
1014	登陆成功	
1015	账号登陆过期	
1010	机器码有误或在非绑定电脑登陆	
1005	机器码已经存在	

-1000	接口弃用-请更新	该返回码系统不可更改返回码，记得做处理
1031	机器码已是绑定，无需更改	
1032	软件换绑成功	
1026	密码修改成功	
1030	软件不允许修改机器码	
1038	未开通在线收款功能	
1025	修改密码原始密码错误	
1009	登陆密码有误	
1037	支付创建成功	
1047	黑名单加入成功	
1035	卡种类列表获取成功	
1054	查询在线成员数成功	
1033	获取更新版本信息成功	
1006	账号注册成功	
1048	获取用户软件权限成功	
1007	软件已经关闭验证	
1043	软件已经关闭注册	
1001	软件不存在或密钥错误	
1049	远程转发URL不存在	
1052	远程转发已关闭	
1050	远程转发请求异常	
1051	远程转发请求成功	
1029	充值成功	
1004	用户名已经存在	
1016	用户登陆后被删除即账号不存在	
1017	用户信息查询成功	
1008	用户不存在	
1040	用户被锁定	
1039	用户状态正常	
1018	远程变量不存在	

-1000	接口弃用-请更新	该返回码系统不可更改返回码，记得做处理
1019	远程变量获取成功	
1041	该软件为时间验证不支持 客户端扣点	
1002	软件版本不存在	