

fastlane接口文档

测试环境信息：

minio地址： minio.xxxx.com

用户名： 111

密码： 222

两台签名机：

A机: 192.168.34.23

对应下载地址： <https://download.xxxx.com> 对应目录 /dev/shm

B机： 192.168.34.24

对应下载地址： <https://download.xxxx.com> 对应目录 /dev/shm

fastlane工作目录： [/dev/shm/wood/resign](#)（固定）

原包目录 /dev/shm/wood/appdata/wechat-vi， 里面有最新两个版本

```
/dev/shm/wood/appdata/ymapp-vi  
[root@test-yunqueA ymapp-vi]# ls  
0.6.8 0.7.1
```

新超级签系统2.0 测试账号

Lorne.Anoshenko@mail.com----Xyd002345----EezczLM6Kt----生日:1988年10月1日

密保:朋友x111工作y112父母d113

9032845590

Team ID: WNB7JUH2M

contentProviderId: 123103233

Issuer ID: f4ecceb6-7721-420c-8393-d5a79596036a 26569R2627



Kendra.Chuchina@mail.com----Xyd002345----0jzeudbwX5----生日:1988年10月1日密

保:朋友x111工作y112父母d113

9032845590

Team ID: 6M626T54XW

contentProviderId: 123102846

Issuer ID: 509cdda4-698a-46c8-ad5d-c4f4b6c957dc 5ZD8THM52B



初次使用时需要验证码，会发送到绑定的手机上，认证通过后会保存session，过期时间3~4周

所以上生产之前统一用脚本初始化所有账号

测试环境已经初始化了Lorne.Anoshenko@mail.com，在23,24两台机器，另一个没有，准备走完整的流程

说明：正式环境账号投入使用时要求是新账号，不允许里面账号中没有注册任何UDID，但却生成了release证书，因为发布证书最多就两个，别的应用生成了fastlane拿不到p12，用的时候带着UDID去创建发布证书，不允许占坑位

```
cd /dev/shm/wood/resign
```

```
fastlane ios resign_app
  channelcode: '98767221048515904'
  udid: '00008030-001A44DE2160802E'
  app_identifier: 'com.erdeng.suiyi'
  apple_id: 'Lorne.Anoshenko@mail.com'
  team_id: 'WNB7JUH2M'
  version: '0.6.8'
  bundle_id: 'com.wechatvn.app'
  icon: 'https://xxx57x57.png'
  timestamp: '20210713213356652'
```

channelcode:代理渠道号，代理链接下载需要，官网下载的不要不要不要传这个字段，整块都不要

udid:设备UDID，多个设备用英文逗号分隔不要留空

格 '00008030-001A44DE2160802E,00008030-001A44DE2160802E'

传一个UDID过来，会生该账号下成所有UDID的profile，意味着该包可以供该账号下所有人使用

app_identifier:用于签名的bundle identifier，第一次创建随机生成，为了区别于机器人操作限定规则

com.{组织名称}.{App名称}，[A-Za-z]{8,20}，然后存起来，同一个APP下次还是这个ID，例如该账号

Lorne.Anoshenko@mail.com我随机已经随机指定了

com.erdeng.suiyi，那么全站APP（wechat-vi）会一直是这个ID

apple_id:开发者账号

team_id:team id

version:app版本号, Jenkins发版时会同步到/dev/shm/wood/appdata/wechat-vi/{version}/wechat-vi.ipa下面

bundle_id:这个要和app_identifier区分开, 这是最终的包名, 例如全站APP产品定的包名是com.wechatvn.app, 最终强制修改为这个包名

icon:APP安装时显示logo, 规格57x57.png

timestamp:时间戳用作目标APP路径, 格式20210713213356652

key_id:p8认证信息, 如果传了就表示使用p8方式认证, 否则就是用appleID方式认证

api_key_path:p8认证信息(不建议用p8)

如果用p8的话就传这个字段, 不用就不传, apple_id任何时候都要传, go把文件Redis中读取的信息按照以下格式生成一个json文件放到指定目录中, 如果目录不存在就新建, 如果存在了不要覆盖

文件名就是{key_id}.json, 因为key是换行的命令行不好操作/dev/shm/wood/resign/cert/{apple_id}/{key_id}.json

```
{
  "key_id": "26569R2627",
  "issuer_id": "f4ecceb6-7721-420c-8393-d5a79596036a",
  "key": "-----BEGIN PRIVATE KEY-----
\nMIGTAqEAMBMGBYqGSM49AgEGCCqGSM49AwEHBHkwdwIBAQQggJ99hoC
XTadhSHNp\nbrt3ZwmM/
jyAI9e9FmITw9A8wECgCgYIKoZIzj0DAQehRANCAAT9SJAcUC2a3QN\n
96ji1+zsJF6PXWw9cE0rJcoqk04QLZvHxVViDkQef52wMskBb3zSPFRhZ
jQAD+bp\nKPzfYsK8\n-----END PRIVATE KEY-----",
  "duration": 1200,
  "in_house": false
}
```

最终生成的下载链接:

<https://download.ymartindia.xyz/wood/appdata/wechat-vi/{version}/{timestamp}/manifest.plist>

若命令成功执行, go要检查 /dev/shm/wood/appdata/wechat-vi/

`{version}/{timestamp}/manifest.plist` 是否存在，然后拿着上面的链接通知新超级签系统，新超级签系统要记录下来，这个包是包含了该账号下所有UDID，如果用户安装完删了再来是不需要重新打包的，以前的用户再来下载也是不需要打包的，因为【这个包是包含了该账号下所有UDID】这么做极大的减轻签名机压力

账号一键修复

适用于账号没被封，人工revoke或证书同步异常导致证书失效，可以通过该方式修复它

```
fastlane ios nuke_cert app_identifier:com.udwsxa.app  
apple_id:Kendra.Chuchina@mail.com team_id:6M626T54XW
```

目的是清理签名机和minio上的数据，重置，设置为新账号

证书检测 & profile 检测

方式一：从minio中获取证书*.cer，检测是否过期或失效，路径是yqtest/{team_id}/certs/distribution

Minio上的文件是最新最全的，例如profile中的devices就是苹果后台的UDID列表，可以用来检查该账号使用了多个设备

注意：这些文件都是加密的，需要按照指定规则解密 AES-256-CBC

[yqtest](#) / [WNBJ7JUH2M](#) / [certs](#) / [distribution](#) / 

Used: 21.43 GB

 Search Objects...

Name



X685PS2WTL.p12



X685PS2WTL.cer

[yqtest](#) / [WNBJ7JUH2M](#) / [profiles](#) / [adhoc](#) / 

Used: 21.43 GB

 Search Objects...

Name



AdHoc_com.erdeng.suiyi.mobileprovision

方式二：fastlane检测证书后发送结果给go

Fastlane也是通过以上方式检测的，先同步minio最新文件到本地再和服务器比对
需要讨论shell和go的交互方式

证书过期回调:

```
http://127.0.0.1:6011/app/revoke post
'Content-Type' => 'application/json'

{channelcode=1111111,
udid=7c06db4023a02098ea7826d1c4c3cf4d394dc5d7,
app_identifier=com.soo.yiappvi,
apple_id=Madalin.Lickli@mail.com, team_id=SGRLKBH73N,
version=1.0.0, cert_info=[[User ID, SGRLKBH73N], [Common
Name, iPhone Distribution: Staci Lynn Jackson
(SGRLKBH73N)], [Organisation Unit, SGRLKBH73N],
[Organisation, Staci Lynn Jackson], [Country, US], [Start
Datetime, 2021-07-10 12:05:31 UTC], [End Datetime,
2022-07-10 12:05:30 UTC]]}
```

profile信息查询:

```
http://127.0.0.1:6011/app/profile post
'Content-Type' => 'application/json'

{
  "AppIDName": "comsooyiappvi",
  "ApplicationIdentifierPrefix": [
    "SGRLKBH73N"
  ],
  "CreationDate": "2021-07-10T12:15:39+00:00",
  "Platform": [
    "iOS"
  ],
  "IsXcodeManaged": false,
  "DeveloperCertificates": [
    "#<StringIO:0x00007fd973616f58>"
  ],
  "DER-Encoded-Profile": "#<StringIO:0x00007fd973616cd8>",
  "Entitlements": {
    "application-
```

```
identifier": "SGRLKBH73N.com.soo.yiappvi",
  "keychain-access-groups": [
    "SGRLKBH73N.*",
    "com.apple.token"
  ],
  "get-task-allow": false,
  "com.apple.developer.team-identifier": "SGRLKBH73N"
},
"ExpirationDate": "2022-07-10T12:05:30+00:00",
"Name": "match AdHoc com.soo.yiappvi",
"ProvisionedDevices": [
  "00008030-000438E21AC0802E",
  "00008020-001675C81141002E",
  "00008101-001C244E0A28001E",
  "00008020-0018351C0290003A",
  "00008101-001968521A44001E",
  "00008020-0011499A22F2002E",
  "00008030-000634403C03802E",
  "2e94dbf1600c458854499223a82b6ccda07c933a",
  "2ab98c80cd14b01515f8b5a7c89eeea3c3a1229e",
  "c349da21968a71b7732d4cc699d5def95ec6bcb3",
  "7c06db4023a02098ea7826d1c4c3cf4d394dc5d7",
  "00008101-001A415C212A001E",
  "00008030-001215362639802E",
  "1fb7a45738f2954d0fc171166c9910119e739139",
  "00008020-001204843E12002E",
  "a82997aa14f61dbd6eedecb41b94e161c6fcbc71",
  "e2d76d2e6baf21aeea09ea4556690b6658f111d8",
  "00008020-001A292A36F8003A",
  "00008030-000154D60E30802E",
  "045eaa36f3b38aaee2c52f6ab5a9467178087558",
  "d13b10a7ff0f2a10694367721303d7cb1f1c74b0",
  "00008101-001621443620001E",
  "6c5b947be4f509ee9d1739567e4a097d6630ea72",
  "00008101-0015714C2E3A001E",
  "00008030-000960AC11D2402E",
  "00008030-001A44DE2160802E",
  "00008030-001145543420402E",
  "dd5de468de235dc8a49560dced12a3f3e6f8381c"
],
"TeamIdentifier": [
  "SGRLKBH73N"
],
"TeamName": "Staci Lynn Jackson",
```

```
"TimeToLive":364,  
"UUID":"b6e5493e-fe4d-41e2-a8cf-49fbc3ab9717",  
"Version":1  
}
```

错误

访问苹果接口错误回调，主要用于监控

401: 可以当做账号被封的依据

403: 账户密码错误或账号有问题

502: 稍后重试

412: 需要两步验证，人工处理手动输入验证码

code: 苹果接口响应码

msg: 格式化后的错误信息和修复错误的方案

data:

- params: 命令行调用原始参数
- body: 苹果接口response body

```
http://127.0.0.1:6011/app/error post  
'Content-Type' => 'application/json'  
  
{  
  "code": 401,  
  "msg": "Authentication credentials are missing or  
invalid. - Provide a properly configured and signed  
bearer token, and make sure that it has not expired.  
Learn more about Generating Tokens for API Requests  
https://developer.apple.com/go/?id=api-generating-tokens",  
  "data": {  
    "params": {  
      "channelcode": "1111111",  
      "udid": "7c06db4023a02098ea7826d1c4c3cf4d394dc5d7",  
      "app_identifier": "com.soo.yiappvi",  
      "apple_id": "Madalin.Lickli@mail.com",  
      "team_id": "SGRLKBH73N",  
      "version": "1.0.0",  
      "key_id": "8CWW42495P"  
    },  
    "body": {
```



```

    "errors": [
      {
        "status": "401",
        "code": "NOT_AUTHORIZED",
        "title": "Authentication credentials are
missing or invalid.",
        "detail": "Provide a properly configured and
signed bearer token, and make sure that it has not
expired. Learn more about Generating Tokens for API
Requests https://developer.apple.com/go/?id=api-
generating-tokens"
      }
    ]
  }
}
}

```

重点关注的错误代码：

```

msg = "Apple 302 detected - this might be temporary
server error, check https://developer.apple.com/system-
status/ to see if there is a known downtime"
error_callback(302, msg)

```

```

msg = "Apple 502 detected - this might be temporary
server error, try again later"
error_callback(502, msg)

```

```

when 401
  msg = "Auth lost"

```

```

when 403
  msg = "Access forbidden"

```

```

when 429
  error_callback(response.status, "TooManyRequestsError")

```

```

when 403
  msg = "Invalid username and password combination. Used
'#{user}' as the username."
  error_callback(response.status, msg)

```

```
when 409
  error_callback(response.status, "2 step/factor is
enabled for this account, first handle that")
```

```
# Need to acknowledge Apple ID and Privacy statement -
https://github.com/fastlane/fastlane/issues/12577
# Looking for status of 412 might be enough but might be
safer to keep looking only at what is being reported
error_callback(response.status, "Need to acknowledge
Apple ID and Privacy statement - https://github.com/
fastlane/fastlane/issues/12577")
```

```
Utils.error_callback(603, "UnexpectedResponse Error,
http.status=#{response.status}", body)
```

服务器错误5xx

```
error_msg = "Received an internal server error from App
Store Connect / Developer Portal, please try again later"
Utils.error_callback(status, error_msg, body)
```

服务器超时5xx

```
error_msg = "Received a gateway timeout error from App
Store Connect / Developer Portal, please try again later"
Utils.error_callback(status, error_msg, body)
```

```
error_msg = "#{body['userString']} Please manually log
into your Apple Developer account to review and accept
the updated agreement."
Utils.error_callback(601, error_msg, body)
```

```
Your developer account needs to be updated
error_msg = "#{body['userString']} account close 有可能封号
了."
Utils.error_callback(602, error_msg, body)
```

其他UnexpectedResponse错误

```
Utils.error_callback(603, "UnexpectedResponse Error,
http.status=#{response.status}", body)
```

升级

方法同上，udid传该账号下所有UDID用英文逗号分隔

tips:实测传该账号下任一UDID或不传都是OK的，每次签名都是那最新所有UDID来签的