

PayTrigger(Microfinancing)_PartnerAPI_ZH_20251113

Security Solution Service API Documentation

版本	修改内容	修改日期	作者
1.0	初始版本	2025-11-13	谭旭林

1. API 接入说明

1. 联系商务,提供贵司对应的 IP whitelist, 商务会开通贵司测试账号
2. 在 Dashboard 中获取贵司 apiKey, 便于后续接口调用生成验签
 - a. 登录商户平台 : <https://paytrigger.transsion-os.com/pay-trigger/#/login>
 - b. 进入页面 : Developer Management -> Customize IP



3. 测试联调完成, 联系技术和商务协助部署正式环境(注意区分业务是否印度市场运营)

1.1 地址说明

1.1.1 正式环境说明

A.正式非印度市场:

Formal API address: <https://paytrigger.transsion-os.com/PayTrigger>

Dashboard address: <https://paytrigger.transsion-os.com/pay-trigger/#/login>

B.正式印度市场:

Formal API address: <https://ind-paytrigger.transsion-os.com/PayTrigger>

Dashboard address: <https://ind-paytrigger.transsion-os.com/pay-trigger/#/login>

1.2. 验签具体算法

Plaintext

1. 构造 request body, 如:

```
{ "apiKey": "9j4S70b0sT5h6DVwAez2cPxUKMqBvf8d", "imeiInfo": "apiKey=9j4S70b0sT5h6DVwAez2cPxUKMqBvf8d&imeiInfo=[{"cycleType": 1000, "deeplink": "https://www.baidu.com", "deeplinkPkg": "com.baidu", "expiration": 1621468800, "imei": "353457595430724", "model": "TECNO BD1", "nfcFlag": true, "orderNum": "PT202207181233", "planGaid": "pla9536533", "ram": "1", "rom": "16", "ruleNum": 0}]&preLockFlag=true"
```

2. 将非空参数值参数按照参数名 ASCII 码从小到大排序, 使用 URL 键值对格式 (k1=v1&k2=v2...) 拼接成字符串, 如:

```
apiKey=9j4S70b0sT5h6DVwAez2cPxUKMqBvf8d&imeiInfo=[{"cycleType": 1000, "deeplink": "https://www.baidu.com", "deeplinkPkg": "com.baidu", "expiration": 1621468800, "imei": "353457595430724", "model": "TECNO BD1", "nfcFlag": true, "orderNum": "PT202207181233", "planGaid": "pla9536533", "ram": "1", "rom": "16", "ruleNum": 0}]&preLockFlag=true
```

3. 使用算法 HmacSHA256 和 apiKey 作为加密 key 对步骤 2 的字符串摘要计算, 得到 byte[], 然后对 byte[] 进行 hexString 计算并转换为大写, 最后如:

```
827E2F1B5A8C3AA41757E0C6B22E3F6EB472FDB7A50201921D1D4D91AA5033FF
```

4. Base64 转码后生成 sign:

```
ODI3RTJGMUI1QThDM0FBND E3NTdFMEM2QjIyRTNGNkVCNDcyRkRCN0E1MDIwMTkyMUQxRDREOTFBQTUwMzNGRg==
```

可通过 Portal 页面 Developer Management -> API Debugging 进行联调测试。

Device Management

Custom Management

Developer Management

Apk Management

Customize IP

API Debugging

Settings Management

API Debugging

Use the enroll API as an example for debugging demonstration. Please follow the steps step by step.

Step 1: Constructing the request body

*apikey

Please input apikey

You can get it from Developer Management -> Customize IP -> Your apikey

proLockFlag

False

True means it will be locked immediately after activation.False means you need to provide an expiration time, and it will be locked after the expiration time.

imeiInfo

*IMEI

Please input IMEI1

*expiration

☐ Please choose time

If proLockFlag=true,optional

Model

Please input Model, like TECNO B06

Optional

RAM

Please input RAM

Optional

ROM

Please input ROM

Optional

cycleType

Please input Repayment period

Optional, 3000-Daily 3000-Weekly 3000-Monthly

orderNum

Please input orderNum

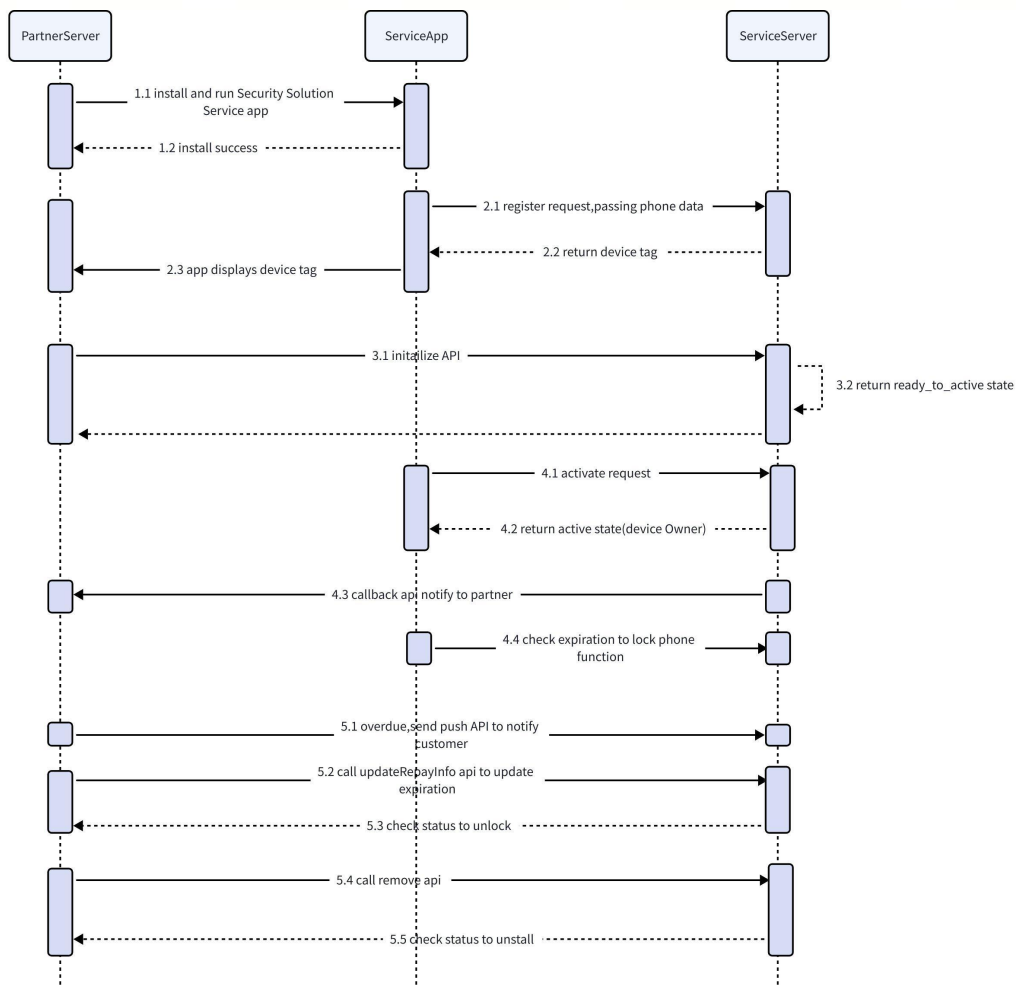
Optional

Other optional information is omitted here. For details, please refer to the API document

Constructing HTTP body

2. 状态流程图

Device Lock Lifecycle:Registered-->Ready_to_active-->Active-->Removable



3. API 列表

3.1 初始化激活设备锁 API

Request:

HTTP

Desc: 设备注册后，需调用初始化激活接口设置首次到期时间

POST /api/partner/lock/v1/initializeLock

Content-Type: application/json; charset=UTF-8

sign:MGJhODAxMTlhOWRhZWRLYjMxN2Q5YmVhMzkzMWFmNjFjOGZkODZjYjQyYTU5NjZkNmFkZDU1ZDM3YzMyMWIyYg==

```
{
  "repayedAmt":1000,
  "totalAmt":3000,
  "deviceTag":"QYMDX4KN",
  "nextRepayTime":1576322185,
  "nextRepayAmt":1000,
  "currencyType":"$",
  "currentTerm":1,
  "totalTerm":3,
  "relatedMerchant":"8qXIGKndpecSDm1LF1HaQ0fN6AREjvs4",
  "description":"repay record update",
  "orderNum":"11111111",
  "phoneNum":"1388188888",
  "deeplink":"halacredit://credit/order?id=123664",
  "deeplinkPkg":"com.hala.fintech.coihub",
  "ruleNum":1,
  "imei":"111124345435432",
  "loanPurpose":"test"
}
```

Parameter	Type	Required	Description
deviceTag	String	T	设备标签号
imei	String	F	设备 imei; 当 deviceTag 和 imei 两个都设置，则校验是否匹配

repayedAmt	BigDecimal	T	已还金额， 两位小数
totalAmt	BigDecimal	T	总金额， 两位小数
nextRepayTime	Long	T	下期还款时间 时间戳 13 位 单位毫秒（需大于当前时间戳）
nextRepayAmt	BigDecimal	T	下期还款金额， 两位小数
currencyType	String	T	货币单位符号， 如 \$
currentTerm	Integer	T	当前期数 （支持范围 1-1000）
totalTerm	Integer	T	总期数（支持范围 1-1000）
relatedMerchant	String	T	关联商户 apiKey（商户后台可查看）
description	String	F	备注
phoneNum	String	T	客户手机号
orderNum	String	T	商户订单号
deeplink	String	F	商户 deeplink
deeplinkPkg	String	F	商户 deeplink 对应包名
ruleNum	Integer	F	锁策略规则 0-5， 不传则为默认规则 0
loanPurpose	String	T	借款目的
sign	String	T	header 中 必传 sign， sign=Base64.encode(HMAC_SHA256(content)) 具体 demo 可咨询技术

Response:

HTTP
Content-Type: application/json; charset=UTF-8

```
{
  "code": 200,
  "data": {
    "orderNum": "11111111"
  },
  "message": "Success"
}
```

字段	字段类型	说明
code	int	结果状态码
message	String	结果描述
orderNum	String	商户订单号

3.2 合作伙伴还款状态变更 API（实际解锁）

Request:

HTTP
Desc: 客户每次还款后，调用还款变更接口解锁，延长到期时间，产生解锁指令

POST /api/partner/lock/v1/updateRepayInfo

Content-Type: application/json; charset=UTF-8

sign:MGJhODAxMTlhOWRhZWRlYjMxN2Q5YmVhMzgzMWFmNjFjOGZkODZjYjQyYTU5NjZkNmFkZDU1ZDM3YzMyMWIyYg==

```
{
  "repayedAmt": 1000,
  "totalAmt": 3000,
  "deviceTag": "YJ72Q18E",
  "imei": "1587105425960",
  "nextRepayTime": 1573371983,
  "nextRepayAmt": 1000,
}
```

```

"currencyType":"$",
"currentTerm":1,
"totalTerm":3,
"relatedMerchant":"8Zlibe3YNpQSEcgjLn0TJVXUCy5IOmBM",
"description":"repay record update",
"orderNum":"11111111",
"phoneNum":"1388188888",
"deeplink":"halacredit://credit/order?id=123664",
"deeplinkPkg":"com.hala.fintech.coihub",
"ruleNum":1
}

```

Parameter	Type	Required	Description
deviceTag	String	F	设备标签
imei	String	F	设备 imei,deviceTag 与 imei 不能同时为 空，必传一个设备 标识，若两者都传 值，则校验两者匹 配性
repayedAmt	BigDecimal	F	已还金额 两位小 数
totalAmt	BigDecimal	F	总金额 两位小数
nextRepayTime	Long	T	下期还款时间 时 间戳（需大于当前 时间戳）
nextRepayAmt	BigDecimal	F	下期还款金额 两 位小数
currencyType	String	F	货币单位符号 例：\$
currentTerm	Integer	F	当前期数（1-

			1000)
totalTerm	Integer	F	总期数 (1-1000)
relatedMerchant	String	T	关联商户 apiKey
description	String	F	备注
phoneNum	String	F	客户手机号
orderNum	String	F	商户订单号
deeplink	String	F	商户 deeplink
deeplinkPkg	String	F	商户 deeplink 对应包名
ruleNum	Integer	F	锁策略规则 0-5, 不传默认 0
sign	String	T	header 中 必传 sign, sign=Base64.encode(HMAC_SHA256(content))

Response:

```

HTTP
Content-Type: application/json; charset=UTF-8

{
  "code": 200
  "message": "Success"
}
```

字段	字段类型	说明
code	int	结果状态码

message	String	结果描述
---------	--------	------

3.3 合作伙伴移除设备锁 API

Request:

HTTP

Desc: 客户还款全部完成后, 可移除设备锁, 解除设备限制

POST /api/partner/lock/v1/removeLock

Content-Type: application/json; charset=UTF-8

sign:MGJh0DAxMTlh0WRhZWRlYjMxN2Q5YmVhMzkzMWFMNjFjOGZkODZjYjQyYTU5NjZkNmFkZDU1ZDM3YzMyMWIyYg==

```
{
  "deviceTag": "QYMDX4KN",
  "imei": "111111111111",
  "apiKey": "8qXIGKndpecSDm1LF1HaQ0fN6AREjvs4"
}
```

Parameter	Type	Required	Description
deviceTag	String	F	设备标签
imei	String	F	设备 imei,deviceTag 与 imei 不能同时为 空, 必传一个设备 标识
apiKey	String	T	关联商户 apiKey
sign	String	T	header 中 必传 sign, sign=Base64.encode(HMAC_SHA256(content))

Response:

```
HTTP
Content-Type: application/json; charset=UTF-8

{
  "code": 200,
  "message": "Success"
}
```

字段	字段类型	说明
code	int	结果状态码
message	String	结果描述

3.4 查询设备锁状态 API

Request:

```
HTTP
Desc: 查询单个设备锁具体信息

POST /api/partner/lock/v1/findLockState

Content-Type: application/json; charset=UTF-8

sign:MGJhODAxMTlhOWRhZWRLYjMxN2Q5YmVhMzkzMWFMNjFjOGZkODZjYjQyYTU5NjZkNmFkZDU1ZDM3YzMyMWIyYg==

{
  "deviceTag": "QYMDX4KN",
  "imei": "1587105425960",
  "orderNum": "D1587105425960",
  "apiKey": "8qXIGKndpecSDm1LF1HaQ0fN6AREjvs4"
}
```

Parameter	Type	Required	Description
deviceTag	String	F	设备标签号
imei	String	F	设备

			imei,deviceTag 与 imei 不能同时为空, 必传一个设备标识
orderNum	String	F	商户关联订单号 -- 可查询已移除设备
apiKey	String	T	关联商户 apiKey
sign	String	T	header 中 必传 sign, sign=Base64.encode(HMAC_SHA256(content))

Response:

HTTP

Content-Type: application/json; charset=UTF-8

```
{
  "code": 200,
  "data": {
    "deviceTag": "FLTZWA7V",
    "lockState": 2000,
    "serverState": 2000,
    "orderNum": "11111111",
    "mobileStatus": 1000,
    "serverLockStatus": 1000,
    "activeTime": 1596698973,
    "lastConnectTime": 1596698973,
    "expiration": 1596698973,
    "lockRuleNum": 1,
    "lockTypeSwitch":
    "[{\"lockFlag\":false,\"lockType\":\"WATERMARK_LOCK\"},{\"lockFlag\":false,\"lockType\":\"CALL_LOCK\"},{\"lockFlag\":false,\"lockType\":\"SMS_LOCK\"},{\"lockFlag\":false,\"lockType\":\"APP_WHITELIST_LOCK\"},{\"lockFlag\":false,\"lockType\":\"APP_BLACKLIST_LOCK\"},{\"lockFlag\":false,\"lockType\":\"AUTO_POPUP_LOCK\"},{\"lockFlag\":false,\"lockType\":\"CALL_IN_LOCK\"},{\"lockFlag\":false,\"lockType\":\"SCREEN_BLOCKED_LOCK\"}]",
    "clientRemoveTime": null,
  }
}
```

```

        "serverRemoveTime": 1575906352,
        "frameworkVersion": "1.3.0.0",
        "apkVersion": "1.3.0.0",
        "imei": "1587105425960",
        "serverAntiTheftFunction": true,
        "clientAntiTheftFunction": false,
        "showTime": 1649301379742,
        "answerDuration": 111,
        "showType": 1,
        "fileNum": 1,
        "model": "TECNO AC8",
        "partnersPkgState": "TYPE_ABNORMAL",
        "ram": 8,
        "rom": 256,
        "buildNumber": "BF6-SE668SABCDE-SGo-OP-230320V442",
        "antiCrackingVer": "P1"
    },
    "message": "Success"
}

```

字段	字段类型	说明
code	int	结果状态码
message	String	结果描述
deviceTag	String	设备标签
lockState	int	客户端实际状态 0: unregistered 1000:registered 2000:ready_to_activate 3000:active 4000:active_and_lock 5000:removable
mobileStatus	int	客户端实际同步的锁定/未锁定的状态，当水印、拨号盘、短信、APP 任一被锁即为锁定状态，都没有锁定时为未锁定状态

		1000-locked/ 2000-unlock
serverState	int	服务端当前下发状态 500:pre_enroll /1000-registered/2000-ready_to_activate/3000-active/4000-active_and_lock/5000-removable
serverLockStatus	int	服务端下发 的锁定/未锁定的状态，当水印、拨号盘、短信、APP 任一被锁即为锁定状态，都没有锁定时为未锁定状态 1000-locked/ 2000-unlock
activeTime	Long	激活时间时间戳 精确到秒
expiration	Long	到期时间时间戳 精确到秒
lastConnectTime	Long	上次联网时间时间戳 精确到秒
lockTypeSwitch	String	客户端执行成功的最新锁指令 lockType 锁类型： WATERMARK_LOCK-水印锁 CALL_LOCK-电话呼出锁 SMS_LOCK-短信锁 APP_WHITELIST_LOCK-白名单应用 APP_BLACKLIST_LOCK-黑名单应用 AUTO_POPUP_LOCK-自动弹窗锁 CALL_IN_LOCK-呼入锁； SCREEN_BLOCKED_LO

		CK 屏幕锁; lockFlag 锁开关: true-开启 false- 关闭
orderNum	String	商户订单号
lockRuleNum	int	锁策略编号
clientRemoveTime	Long	客户端实际移除时间时间 戳 精确到秒
serverRemoveTime	Long	上服务端下发移除时间时 间戳 精确到秒
frameworkVersion	String	设备 frameworkVersion
apkVersion	String	设备 apkVersion
imei	String	设备 imei
serverAntiTheftFunction	Boolean	防盗锁服务端是否开启
clientAntiTheftFunction	Boolean	防盗锁客户端是否开启
showTime	Long	模拟来电客户端展现时间 戳 精确到秒
answerDuration	Long	模拟来电接听时长 精确 到秒
showType	int	接听类型 0- 未接听 1-已 接听 2-已拒接
fileNum	int	最新一次展示的模拟锁消 息规则编号
erverLockTypeSwitch	String	服务器执行成功的最新锁 指令 lockType 锁类型: WATERMARK_LOCK-水

		印锁 CALL_LOCK-电话 呼出锁 SMS_LOCK-短信 锁 APP_WHITELIST_LOCK -白名单应用 APP_BLACKLIST_LOCK -黑名单应用 AUTO_POPUP_LOCK- 自动弹窗锁 CALL_IN_LOCK-呼入锁; SCREEN_BLOCKED_LO CK 屏幕锁; lockFlag 锁开关: true-开启 false- 关闭
lockCreateTime	Long	服务器锁策略生成时间
lockCompleteTime	Long	客户端锁策略执行时间
model	String	设备型号
partnersPkgState	String	商户 APP 状态: TYPE_ABNORMAL-应用 异常; TYPE_NORMAL-正 常; TYPE_PKG_NULL-未 配置; TYPE_UNINSTALLED-未 安装
ram	int	ram
rom	int	rom
buildNumber	String	buildNumber
antiCrackingVer	String	antiCrackingVer

3.5 批量查询设备锁状态 API

Request:

HTTP

Desc:批量查询设备锁信息

POST /api/partner/lock/v1/batchFindLockState

Content-Type: application/json; charset=UTF-8

sign:MGJhODAxMTlhOWRhZWRLYjMxN2Q5YmVhMzgzMWFmNjFjOGZkODZjYjQyYTU5NjZkNmFkZDU1ZDM3YzMyMWIyYg==

```
{
  "deviceTag": "QYMDX4KN,UBK4F8XN,XNAYPCJL",
  "imei": "1587105425960,1587105425960",
  "orderNum": "D1587105425960,D1587105425960",
  "apiKey": "8qXIGKndpecSDm1LF1HaQ0fN6AREjvs4"
}
```

Parameter	Type	Required	Description
deviceTag	String	F	设备标签号,最多一百个不重复的deviceTag,逗号隔开
imei	String	F	设备 imei,最多一百个不重复的imei,逗号隔开 ;deviceTag 与 imei 不能同时为空, 必传一个设备标识;
orderNum	String	F	商户关联订单号,最多一百个不重复的订单号,逗号隔开
apiKey	String	T	关联商户 apiKey
sign	String	T	header 中 必传 sign,

			sign=Base64.encode(HMAC_SHA256(content))
--	--	--	--

Response:

HTTP

Content-Type: application/json; charset=UTF-8

```
{
  "code": 200,
  "message": "Success",
  "data": [
    {
      "deviceTag": "FLTZWA7V",
      "orderNum": null,
      "lockState": 5000,
      "mobileStatus": 2000,
      "serverState": 5000,
      "activeTime": null,
      "lockTypeSwitch":
        "[{\"lockFlag\":false,\"lockType\":\"WATERMARK_LOCK\"},{\"lockFlag\":false,\"lockType\":\"APP_BLACKLIST_LOCK\"},{\"lockFlag\":false,\"lockType\":\"CALL_LOCK\"},{\"lockFlag\":false,\"lockType\":\"SMS_LOCK\"},{\"lockFlag\":false,\"lockType\":\"APP_WHITELIST_LOCK\"},{\"lockFlag\":false,\"lockType\":\"CALL_IN_LOCK\"},{\"lockFlag\":false,\"lockType\":\"AUTO_POPUP_LOCK\"},{\"lockFlag\":false,\"lockType\":\"SCREEN_BLOCKED_LOCK\"}]",
      "lastConnectTime": 1575896686,
      "expiration": 1576322185,
      "serverLockStatus": 2000,
      "lockRuleNum": 0,
      "clientRemoveTime": null,
      "serverRemoveTime": 1575906352,
      "imei": "1111111111111111",
      "frameworkVersion": "1.2.0.0",
      "apkVersion": "1.2.0.0",
      "serverAntiTheftFunction": true,
      "clientAntiTheftFunction": false,
      "showTime": 1649301379742,
      "answerDuration": 111,
      "showType": 1,
      "fileNum": 1,
      "model": "TECNO AC8",
    }
  ]
}
```

```

        "partnersPkgState": "TYPE_ABNORMAL",
        "ram":8,
        "rom":256,
        "buildNumber":"BF6-SE668SABCDE-SGo-OP-
230320V442",
        "antiCrackingVer":"P4"
    }
]
}

```

字段	字段类型	说明
code	int	结果状态码
message	String	结果描述
deviceTag	String	设备标签
lockState	int	客户端实际状态 0: unregistered 1000:registered 2000:ready_to_activate 3000:active 4000:active_and_lock 5000:removable
mobileStatus	int	客户端实际同步的锁定/未锁定的状态，当水印、拨号盘、短信、APP 任一被锁即为锁定状态，都没有锁定时为未锁定状态 1000-locked/ 2000-unlock
serverState	int	服务端当前下发状态 500:pre_enroll / 1000-registered/2000-ready_to_activate/3000-active/4000-active_and_lock/5000-removable

serverLockStatus	int	服务端下发 的锁定/未锁定的状态，当水印、拨号盘、短信、APP 任一被锁即为锁定状态，都没有锁定时为未锁定状态 1000-locked/ 2000-unlock
activeTime	Long	激活时间时间戳 精确到秒
expiration	Long	到期时间时间戳 精确到秒
lastConnectTime	Long	上次联网时间时间戳 精确到秒
lockTypeSwitch	String	客户端执行成功的最新锁指令 lockType 锁类型：WATERMARK_LOCK-水印锁 CALL_LOCK-电话呼出锁 SMS_LOCK-短信锁 APP_WHITELIST_LOCK-白名单应用 APP_BLACKLIST_LOCK-黑名单应用 AUTO_POPUP_LOCK-自动弹窗锁 CALL_IN_LOCK-呼入锁； ==SCREEN_BLOCKED_LOCK 屏幕锁；==lockFlag 锁开关：true-开启 false-关闭
orderNum	String	商户订单号
lockRuleNum	int	锁策略编号

clientRemoveTime	Long	客户端实际移除时间时间戳 精确到秒
serverRemoveTime	Long	服务端下发移除时间时间戳 精确到秒
frameworkVersion	String	设备 frameworkVersion
apkVersion	String	设备 apkVersion
imei	String	设备 imei
serverAntiTheftFunction	Boolean	防盗锁服务端是否开启
clientAntiTheftFunction	Boolean	防盗锁客户端是否开启
showTime	Long	模拟来电客户端展现时间时间戳 精确到秒
answerDuration	Long	模拟来电接听时长 精确到秒
showType	int	接听类型 0- 未接听 1-已接听 2-已拒接
fileNum	int	最新一次展示的模拟锁消息规则编号
serverLockTypeSwitch	String	服务器执行成功的最新锁指令 lockType 锁类型: WATERMARK_LOCK-水印锁 CALL_LOCK-电话呼出锁 SMS_LOCK-短信锁 APP_WHITELIST_LOCK-白名单应用 APP_BLACKLIST_LOCK-黑名单应用 AUTO_POPUP_LOCK-

		自动弹窗锁 CALL_IN_LOCK-呼入锁; SCREEN_BLOCKED_LOCK 屏幕锁; lockFlag 锁开关: true-开启 false- 关闭
lockCreateTime	Long	服务器锁策略生成时间
lockCompleteTime	Long	客户端锁策略执行时间
model	String	设备型号
partnersPkgState	String	商户 APP 状态: TYPE_ABNORMAL-应用 异常; TYPE_NORMAL-正 常; TYPE_PKG_NULL-未 配置; TYPE_UNINSTALLED-未 安装
ram	int	ram
rom	int	rom
buildNumber	String	buildNumber
antiCrackingVer	String	antiCrackingVer

3.6 设备锁激活/移除信息回传通知合作方 API

Request:

HTTP

Desc: 当设备端实际已激活/移除时会通知到合作方的回调地址

POST https://xxx/callbackUrl

Content-Type: application/json; charset=UTF-8

sign:MGJhODAxMTlhOWRhZWRlYjMxN2Q5YmVmMzkzMWFmNjFjOGZkODZjYjQyYTU5N

jZkNmFkZDU1ZDM3YzMyMWIyYg==

示例 1:

```
{
  "productModel": "TECNO",
  "mobileStatus": 1000,
  "serverState": 1000,
  "orderNum": "1111111",
  "activeTime": 1572969600,
  "deviceTag": "IST3PDUZ",
  "imei": "111111111111",
  "expiration": 1572969600,
  "state": 1000,
  "clientRemoveTime": 1572969600,
  "serverRemoveTime": 1572969600,
  "notifyType": 1000
}
```

示例 2:

如果是强限制锁指令超限后提醒:

```
{
  "notifyType": 4000,
  "deviceTag": "3WGHPKCJ",
  "orderNum": "LOAN1000074720201203071002",
  "imei": "",
  "tip": "The strong restriction order issued by the equipment did not take effect due to control."
}
```

Parameter	Type	Required	Description
productModel	String	T	设备型号
deviceTag	String	T	设备 tag
imei	String	T	设备 imei
orderNum	String	T	商户订单号
expiration	Long	T	过期时间时间戳
activeTime	Long	T	激活时间时间戳

state	int	T	客户端当前状态 0-unregistered/1000-registered/2000-ready_to_activate/3000-active/4000-active_and_lock/5000-removable
mobileStatus	int	T	客户端锁定/未锁定的状态，当水印、拨号盘、短信、APP 任一被锁即为锁定状态，都没有锁定时为未锁定状态 1000-locked/2000-unlock
serverState	int	T	服务端当前状态 500-enrolled/1000-registered/2000-ready_to_activate/3000-active/4000-active_and_lock/5000-removable
clientRemoveTime	Long	F	客户端实际移除时间时间戳 精确到秒
serverRemoveTime	Long	F	上服务端下发移除时间时间戳 精确到秒
==notifyType==	==int==	==T==	==回调通知类型： 1000-客户端激活状态回调通知； 2000-客户端移除状态回调通知； 4000-设备锁指令

			因强限制超限 回调通知==
==tip==	==String==	==F==	==强限制超限后提醒，不下发强锁指令==
sign	String	T	header 中 必传 sign, sign=Base64.encode(HMAC_SHA256(content))

Response:

HTTP

Desc:回调通知接收成功后，你们须通过此格式告知我们处理成功，否则会默认你们处理失败，我们会采取补偿机制再次通知，直到通知成功或达到相应次数后不再继续通知

Content-Type: application/json; charset=UTF-8

```
{
  "code": 200,
  "message": "Success"
}
```

字段	字段类型	说明
code	int	结果状态码
message	String	结果描述

3.7 合作方推送消息 API

Request:

HTTP

Desc:针对单个设备，发送定制化的推送消息

POST /api/partner/push/v1/sendPushInfo

Content-Type: application/json; charset=UTF-8

sign:MGJhODAxMTlhOWRhZWRIYjMxN2Q5YmVhMzgzMWFmNjFjOGZkODZjYjQyYTU5NjZkNmFkZDU1ZDM3YzMyMWIyYg==

```
{
  "deviceTag": "QYMDX4KN",
  "imei": "",
  "apiKey": "8qXIGKndpecSDm1LF1HaQ0fN6AREjvs4",
  "content": "12434",
  "title": "12223",
  "pushType": 1,
  "h5link": "www.baidu.com",
  "deeplink": "halacredit://credit/order?id=123664",
  "deeplinkPkg": "com.hala.fintech.coihub"
}
```

Parameter	Type	Required	Description
apiKey	String	T	关联商户的 apiKey
content	String	T	推送内容，限制在 500 字符以内；如果是 pushType=3，则内容传商户后台上传后产生的录音文件编号 0-3
deviceTag	String	F	设备标签
imei	String	F	设备 imei,deviceTag 与 imei 不能同时为空，必传一个设备标识
title	String	T	推送标题，限制在 80 字符以内;如果是 pushType=3,则

			传模拟来电显示的电话号码
pushType	int	T	发送推送信息 pushType=2 ; 发送弹窗信息 pushType=1;发送模拟来电消息的录音文件 pushType=3
==deeplink==	==String==	==F==	==商户 deeplink 点击消息后的跳转地址，仅支持 pushType=1 和 pushType=2 的消息类型。deeplink 跳转到 deeplinkPkg 的 app==
==h5link==	==String==	==F==	==商户 h5link, h5 link 浏览器打开 h5 link address ,仅支持 pushType=1 和 pushType=2 的消息类型==
==deeplinkPkg==	==String==	==F==	==商户 deeplink 对应包名，传的是 deeplink，该字段必填==
sign	String	T	header 中 必传 sign, sign=Base64.encode(HMAC_SHA256(content))

Response:

HTTP

Content-Type: application/json; charset=UTF-8

```
{
  "code": 200,
  "message": "Success"
}
```

字段	字段类型	说明
code	int	结果状态码
message	String	结果描述

3.8 合作方批量推送消息 API

Request:

HTTP

Desc: 针对多个设备，批量发送定制化的推送消息, 推送/弹窗/模拟来电消息类型

POST /api/partner/push/v1/sendBatchPushInfo

Content-Type: application/json; charset=UTF-8

sign:MGJhODAxMTlhOWRhZWRLYjMxN2Q5YmVhMzgzMWFmNjFjOGZkODZjYjQyYTU5NjZkNmFkZDU1ZDM3YzMyMWIyYg==

```
{
  "pushInfo": "[{\"content\": \"12434\",
    \"deviceTag\": \"QYMDX4KN\",
    \"imei\": \"1111\",
    \"title\": \"12223\",
    \"h5link\": \"www.baidu.com\",
    \"deeplink\": \"halacredit://credit/order?id=123664\",
    \"deeplinkPkg\": \"com.hala.fintech.coihub\"}]",
  "apiKey": "8qXIGKndpecSDm1LF1HaQ0fN6AREjvs4",
  "pushType": 1
}
```

Parameter	Type	Required	Description
pushInfo	String	T	pushInfo 值格式应为字符串，字符串里包含的是一个数组，基础数组的排序方式也是 ASCLL 排序，如示例，这样 sign 就会匹配
apiKey	String	T	关联商户的 apiKey
content	String	T	推送内容，限制在 500 字符以内;如果是 pushType=3,则传商户后台上传后产生的录音文件编号 0-3
deviceTag	String	F	设备标签,每次最多 100 个
imei	String	F	设备 imei,deviceTag 与 imei 不能同时为空，必传一个设备标识
title	String	T	推送标题，限制在 80 字符以内;发送模拟来电消息的录音文件 pushType=3
pushType	int	T	发送模拟来电信息 pushType=3 ; 发送推送信息 pushType=2 ; 发

			送弹窗信息 pushType=1
==deeplink==	==String==	==F==	==商户 deeplink 点击消息后的跳转 地址, 仅支持 pushType=1 和 pushType=2 的消 息类型。deeplink 跳转到 deeplinkPkg 的 app==
==h5link==	==String==	==F==	==商户 h5link, 浏 览器打开 h5 link address ,h5 link 仅 支持 pushType=1 和 pushType=2 的 消息类型==
==deeplinkPkg==	==String==	==F==	==商户 deeplink 对应包名, 传的是 deeplink, 该字段 必填==
sign	String	T	header 中 必传 sign, sign=Base64.encode(HMAC_SHA256(content))

Response:

```

HTTP
Content-Type: application/json; charset=UTF-8

{
  "code": 200,
  "data": [
    {
      "code": 30020,

```

```
        "deviceTag": "QYMDX4KN",
        "imei": "11111",
        "message": "ClientId is null"
    }
],
"message": "Success"
}
```

字段	字段类型	说明
code	int	结果状态码
data	List	对应的失败的 deviceTag
message	String	结果描述

3.9 合作伙伴临时解除设备锁 API

Request:

HTTP

Desc: 在用户被锁机情形下，临时允许用户设备解锁一定时间

POST /api/partner/unlock/v1/tempUnlock

Content-Type: application/json; charset=UTF-8

sign:MGJhODAxMTlhOWRhZWRLYjMxN2Q5YmVhMzgzMWFmNjFjOGZkODZjYjQyYTU5NjZkNmFkZDU1ZDM3YzMyMWIyYg==

```
{
  "deviceTag": "QYMDX4KN",
  "imei": "1587105425960",
  "apiKey": "8qXIGKndpecSDm1LF1HaQ0fN6AREjvs4",
  "tempLockTime": 72,
  "timeUnit": "HOUR"
}
```

Parameter	Type	Required	Description
-----------	------	----------	-------------

deviceTag	String	F	设备标签
imei	String	F	设备 imei,deviceTag 与 imei 不能同时为 空，必传一个设备 标识
apiKey	String	T	关联商户 apiKey
tempLockTime	Integer	F	临时解锁时间，默 认 24 小时，单 位：小时
timeUnit	String	F	临时解锁时间单 位，如果未传，则 默认： HOURS ； 枚举值： HOURS-小时 MINUTES-分钟
sign	String	T	header 中 必传 sign,sign=Base64. encode(HMAC_S HA256(content))

Response:

```

HTTP
Content-Type: application/json; charset=UTF-8

{
  "code": 200,
  "message": "Success"
}
```

字段	字段类型	说明
----	------	----

code	int	结果状态码
message	String	结果描述

3.10 PIN Unlock 离线解锁

Request:

HTTP

Desc: 用户逾期锁机后，通过离线方式还款后，从其金融方获得一个离线解锁 pin，用户在 PT 界面输入 pin，即可解锁。

POST /api/partner/unlock/v1/verifyCode

Content-Type: application/json; charset=UTF-8

sign:MGJh0DAxMTlh0WRhZWRlYjMxN2Q5YmVhMzkzMWFMNjFjOGZkODZjYjQyYTU5NjZkNmFkZDU1ZDM3YzMyMWIyYg==

```
{
  "apiKey": "3w1mZa2pK79Wye0k5UfLCHPVbiodNv4F",
  "imei": "112233445566",
  "captcha": "1234",
  "deviceTag": "A23DFTP8"
}
```

Parameter	Type	Required	Description
imei	String	F	设备 imei 号
deviceTag	String	F	deviceTag
captcha	String	F	4 位随机数
apiKey	String	T	关联商户 apiKey
sign	String	T	header 中 必传 sign, sign=Base64.encode(HMAC_SHA25

			6(content))
--	--	--	-------------

Response:

```

HTTP
Content-Type: application/json; charset=UTF-8

{
  "code": 200,
  "message": "Success",
  "data": {
    "verifyCode": "123456789"
  }
}

```

字段	字段类型	说明
code	int	结果状态码
message	String	结果描述
verifyCode	String(9)	9 位验证码

3.11 商户 license 信息查询接口 API

Request:

```

HTTP
Desc: 查询商户 license 信息

POST /api/partner/company/v1/checkLicense

Content-Type: application/json; charset=UTF-8

sign:MGJhODAxMTlhOWRhZWRlYjMxN2Q5YmVhMzkzMWFMNjFjOGZkODZjYjQyYTU5NjZkNmFkZDU1ZDM3YzMyMWIyYg==

{
  "apiKey": "8qXIGKndpecSDm1LF1HaQ0fN6AREjvs4"
}

```

Parameter	Type	Required	Description
apiKey	String	T	关联商户的 apiKey
sign	String	T	header 中 必传 sign, sign=Base64.encode(HMAC_SHA256(content))

Response:

```

HTTP
Content-Type: application/json; charset=UTF-8

{
  "code": 200,
  "message": "Success",
  "data": {
    "totalAmountOfLicense": 20000,
    "amountUsedOfLicense": 0,
    "remainingAmountOfLicense": 20000
  }
}

```

字段	字段类型	说明
code	int	结果状态码
message	String	结果描述
totalAmountOfLicense	Long	购买总激活额度
amountUsedOfLicense	Long	已使用激活额度
remainingAmountOfLicense	Long	剩余可用激活额度

3.12 查询客户反馈信息

Request:

HTTP

Desc: 查询客户反馈信息

POST /api/partner/feedback/v1/query

Content-Type: application/json; charset=UTF-8

sign:N0I2QzRGNTZDNUYzOTIyNjc2RDhRDMjAwNzZEQTg1MDFBFQ0NCNkI3N0RGQTA3RUJFRjIxQUJEQkMxRUVBNTkFQg==

```
{
  "apiKey": "8qXIGKndpecSDm1LF1HaQ0fN6AREjvs4",
  "deviceTag": "2H8DAUNM",
  "endTime": "2024-01-20",
  "imei": "855512222222226",
  "feedbackType": "Complaint",
  "pageNum": 1,
  "pageSize": 100,
  "startTime": "2024-01-01",
  "status": "1"
}
```

Parameter	Type	Required	Description
apiKey	String	T	关联商户的 apiKey
deviceTag	String	F	设备标签号
imei	String	F	设备 imei
feedbackType	String	F	反馈类型
pageNum	Integer	T	分页查询-页码
pageSize	Integer	T	分页查询-页面大小
startTime	String	T	Start time, utc0

endTime	String	T	End Time, utc
status	String	T	状态: 1-有效; 0-无效
sign	String	T	header 中 必传 sign, sign=Base64.encode(HMAC_SHA256(content))

Response:

HTTP
Content-Type: application/json; charset=UTF-8

```
{
  "code": 200,
  "message": "Success",
  "data": {
    "count": 1,
    "list": [
      {
        "imei": "85551222222226",
        "deviceTag": "2H8DAUNM",
        "problemDesc": "test",
        "phoneNum": "",
        "email": "",
        "feedbackType": "Complaint",
        "fileUrl": null,
        "orderNum": null,
        "ct": "2024-01-16 10:05:44"
      }
    ]
  }
}
```

字段	字段类型	说明
code	int	结果状态码

message	String	结果描述
data.count	int	分页查询记录总数
data.list	List	分页查询，当页记录列表
data.list.deviceTag	String	设备标签号
data.list.imei	String	设备 imei,deviceTag 与 imei 不能同时为空，必传一个设备标识
data.list.problemDesc	String	用户反馈描述
data.list.phoneNum	String	用户手机号码
data.list.email	String	用户 email
data.feedbackType	String	反馈类型
data.list.fileUrl	String	用户反馈上传的文件
data.list.orderNum	String	三方订单号
data.list.ct	String	用户反馈的时间

4. 错误码

返回码	返回码描述	说明
200	Success	成功
201	Non-existent data	
400	bad request	检查请求头是否包含 sign; 请求必填参数是否符合 文档入参类型及个数

500	Server Error	系统错误
510	Frequent requests, try again later	QPS 限流, 请求频率限制,需要降低请求频率
513	argument invalid: {0}	
515	sys_config error, {0}	
516	process failure, {0}	
517	encrypt data failure	
518	decrypt data failure	
519	data processing	
520	export data error, data empty	
530	error sign	
10011	DeviceTag is null	设备标签为空
10012	Expire time is null	
10014	Imei format error	
10036	Param format error	参数格式错误
10042	DeviceTag quantities are out of range	设备数超限
10045	Error,the function is turned off.	功能开关已关闭
20001	DeviceTag or imei does not exist, or the two do not match	设备标签或 imei 不存在, 或两者不匹配
20002	Expire time not exist or	过期时间不存在或已失效

	not valid	
20003	Apikey not exist or expired	apiKey 不存在或已失效
20004	Illegal state change	非法状态请求转换
20005	Apikey has no permission	ApiKey 无权限操作，或联系商务开通该功能
20015	This account does not have permission to activate the current country/model/A/S device	ip 国家/机型 /apkVersion/frameworkVersion 不支持，核对 apk 请求网络是否在允许运营国家名单中；核对机型，apkVersion,frameworkVersion 是否加入运营白名单
30004	Param error	参数错误
30011	Insufficient remaining available licenses	License 额度不足
30021	This device has been registered by another company, please change the device and try again	设备已被其他商户占用
30022	Push type error	推送类型错误
40000	Sign error	验签签名错误
40001	Request ip is not in whitelist ip:{0}	需设置 ip 白名单，联系商务添加服务器 ip 到白名单
40002	Sign is null	验签签名为空

40003	Please configure whitelist ip	需设置 ip 白名单，联系商务添加服务器 ip 到白名单
50000	DetailId error	
50002	ApiKey is null	商户 apiKey 为空
50003	ContractMachine simNumber or phoneNumber is null	
50004	ContractMachine simNumber,phoneNumber and lockSwitch have no change	
50005	Required simNumber format error	
50006	Required phoneNumber format error	
50007	Repay state error	还款状态错误
50008	The number of calls to a single device exceeds the limit. Please try again after 24 hours.	单个设备消息或临时解锁接口超过最大次数
50009	Lock state error	设备锁状态错误
50010	The device is not overdue	设备未逾期，不可调用临时解锁功能
50011	Your company has not activated this function temporarily	贵公司暂未开通临时解锁功能
50012	Both deeplink and deeplinkPkg are empty or have values at the same	需同时配置 deeplink 和包名，要么同时存在，要

	time	么同时为空
50013	Rule number does not exist or is not effective	入参传的规则编号不存在或不生效
50014	The device has opened the anti-theft lock	设备已开启防盗锁，不可临时解锁
50015	This IMEI has been enrolled or activated and cannot be enrolled again.	预录入时，IMEI 已经存在，不包含已经取消的预录入数据
50016	The maximum number of batch operations is 2000. Please reduce the number and try again.	批量操作时，设备数最大条数不能超过 2000，包含批量预录入、批量移除、批量 updateRepay、批量取消。
50017	Partner not exist	合作方不存在，检查 apiKey 是否正确
50020	The imei length must be 12 to 20 digits	imei 长度需 12-20 位数字
50021	Some IMEI entry failed	批量录入时，部分失败 imei 返回
50022	Expiration is null	预录入时，选择激活后不立即锁定，需设置到期时间
50023	The device does not support SMS unlocking	低版本设备不支持短信解锁功能，需升级
50024	The order num not exist	订单号错误或不存在
50025	The device is not in removable state	设备未移除，不能通过订单号查询

50026	Your company has not uploaded or deleted the recording file	贵司未上传模拟来电录音文件 或已删除此文件
50027	The file address is incorrect or invalid. Please check and retransmit	文件地址不正确或无效, 请核对后重新输入
50028	The simulated incoming call num not exist	模拟来电消息规则编号不存在, 请核对后重新输入
50029	Device unavailable	
50051	Imei not exist	
50052	IMEI is already in use	
50053	Repeated activation is not allowed. Please change the device and try again.	未打开重复预录入的开关, 进行重复预录入了
50054	The Switch is null	
50055	The plan gaid is consistent with the device gaid, and activation is not allowed	
50056	The pre entered model is inconsistent with the equipment model and cannot be activated	
50057	The pre entered rom is inconsistent with the equipment rom and cannot be activated	
50058	The pre entered ram is inconsistent with the	

	equipment ram and cannot be activated	
50059	The pre entered nfcFlag is inconsistent with the equipment nfcFlag and cannot be activated	
50060	Wrong time range or format	
50061	The order no existst	
50062	The three parameters, deeplink, deeplinkPkg and h5link, cannot exist at the same time. You can only select one type of link	选择一种 link 类型
50063	The APK version is {0},so the key code is required.	
71107	Pin config is null.	离线解锁配置不存在
9900	Server Error	
9999	Server Error, {0}	

频繁度限制说明

接口	限制说明	异常返回码	返回码描述
/api/partner/lock/v1/findLockState	24 小时内单个设备查询上限 100 次	510	Frequent requests, try again later
/api/partner/lock/v1/batchFindLockState	24 小时内批次查询上限 100 次	510	Frequent requests, try again later

/api/partner/push/v1/sendPushInfo	- 弹窗信息 pushType=1 24 小时内单个设备查询上限 3 次; - 推送信息 pushType=2, 24 小时内单个设备查询上限 3 次; - 模拟来电 pushType=3, 24 小时内单个设备查询上限 1 次;	50008	The number of calls to a single device exceeds the limit. Please try again after 24 hours.
-----------------------------------	---	-------	--

5. 验签具体算法

Plaintext

- 将非空参数值参数按照参数名 ASCII 码从小到大排序，使用 URL 键值对格式（k1=v1&k2=v2...）拼接成字符串，如：
apiKey=9j4S70b0sT5h6DVwAez2cPxUKMqBvf8d&imeiInfo=[{"expiration":null,"imei":"353457595430724"}]&preLockFlag=true
- HMAC_SHA256+apiKey 加密后全部转大写：
4AEC1192B7AD077767FBB323F8B3D5B7342FF66841AF13CA79F78F07E4DC1CE7
- Base64 转码后生成
sign:NEFFQzExOTJCN0FEMDc3NzY3RkJCMzIzRjhCM0Q1Qjc zNDJGRjY2ODQxQUYxM0NBnzlGNzhGMDdFNERDMUNFNw==

6. Q&A

Q1:When using Python syntax to interface with Java API, what to do if the boolean type distinguishes the first letter from the upper case, resulting in an error in signature verification?

A1: Python and Java are different in Boolean:

In Python, letters indicating true and false must start with capital letters, for example

a = True;

b = False;

Java means true and false;

Therefore, I suggest that when you transmit true / false, you use the string format. In

this way, the string to be signed is uniform, and the final signature verification sign will be consistent

The Payload is : {'imeiInfo':

```
u'[{ "expiration":1662136801,"imei":"359581820772412","model":"infinix SMART 6","ram":"2","rom":"32"}]', 'preLockFlag': 'false', 'apiKey': '8qXIGKndpecSDmILF1HaQ0fN6AREjvs4'}
```

Plaintext

```
payload string is
apiKey=8qXIGKndpecSDmILF1HaQ0fN6AREjvs4&imeiInfo=[{"expiration":1662136801,"imei":"359581820772412","model":"infinix SMART 6","ram":"2","rom":"32"}]&preLockFlag=false
```

Plaintext

```
upperStr: =====
BB76C48237CCC2A3B6DF5EF9283E91ACFE983EE964C9B34F5F3E1691F3831AA7
```

Plaintext

```
sign:QkI3NkM0ODIzN0NDQzJBM0I2REY1RUY5MjgzRTkxQUNGRTk4M0VFOTY0Qz1CMzRGNUYzRTE2OTFGMzgzMUFBNw==
```

Q2:HmacSHA256 in PHP VS JAVA VS Golang

A2:

The Java algorithm is slightly different from the PHP encryption algorithm. You should check it

It can be roughly divided into the following parts for analysis:

- \1. Get sha256 instance
- \2. Generate an encrypted key
- \3. Initialize the sha256 instance with this encryption key
- \4. Use this instance to encrypt and generate hash according to the provided string
- \5. Finally, the whole is converted to hexadecimal and then output the string

The PHP part :

```
function (plain, secretKey){
return bin2hex(hash_hmac("sha256",utf8_encode(plain),utf8_encode(secretKey),true));
}
```

=====Golang SHA256 HMAC Hash=====

```
package main

import (
    "crypto/hmac"
    "crypto/sha256"
    "encoding/hex"
    "fmt"
)

func main() {
    secret := "mysecret"
    data := "data"
    fmt.Printf("Secret: %s Data: %s\n", secret, data)
    // Create a new HMAC by defining the hash type and the key (as byte array)
    h := hmac.New(sha256.New, []byte(secret))
    // Write Data to it
    h.Write([]byte(data))
    // Get result and encode as hexadecimal string
    sha := hex.EncodeToString(h.Sum(nil))
    fmt.Println("Result: " + sha)
}
```