

德施曼 ios 接口文档

修改日期	版本	修改内容	修改描述	负责人
2017.04.12	1.0		第一版	王斌波
2017.07.12	2.0	智能钥匙接口	第二版	王斌波

目录

1.用户注册登陆	4
1.1 生成验证码	4
1.2 验证码校验	4
1.3 用户注册	5
1.4 用户登陆	5
1.5 修改密码	6
1.6 头像上传	6
1.7 个人信息完善接口	7
1.8 用户注销	7
2.初始化	9
2.1 锁具配对获取 deviceidentifier.....	9
2.2 获取设备信息	9
3.添加设备	11
3.1 添加设备	11
4.手机开门	13
4.1 确定的 mac 地址的手机开门	13
5.个性设置	14
5.1 修改锁具名称	14
5.2 修改用户个性化手机开门方式.....	14
5.3 修改用户个性化开门密码	15
5.4 修改用户手势密码	15
5.5 手势密码开门校验（也可以本地校验）	16
6.高级设置——用户模块	17
6.1 获取当前锁具的用户列表	17
6.2 添加用户	18
6.3 删除用户	18
6.4 用户重命名	19
6.5 设置用户开门方式	19
6.6 设置用户开门时效	20
6.7 添加指纹	21
6.8 删除指纹	21
6.9 指纹重命名	22
6.10 获取指纹列表	22
6.11 高级设置验证	23
7.高级设置——开门密码模块	25
7.1 设置开门密码（锁具上）	25
7.2 生成临时密码	25
7.3 临时钥匙授权	26
7.4 临时密码授权	26

8. 高级设置——锁具信息模块	27
8.1 获取锁具详情	27
8.2 固件更新	28
8.2.1 检测是否有新版本固件	28
8.2.2 新版本固件下载	28
8.2.3 固件更新	29
8.3 注销锁具	29
8.4 获取 WIFI	30
8.5 设置锁具 WIFI	30
8.6 WIFI 推送检测	31
8.7 开门记录上传功能开启与关闭	31
9. 高级设置——亲情设置模块	32
9.1 设置亲情关联	32
9.2 删除亲情关联	32
9.3 获取设备的亲情关联	33
10. 高级设置——报警模块	34
10.1 锁具报警密码设置	34
10.2 锁具报警密码启用 / 关闭	34
10.3 设置指纹报警关联	35
10.4 设置密码报警关联	35
10.5 取消指纹报警关联	36
10.6 取消密码报警关联	36
10.7 查询锁具密码报警关联用户接口	37
10.8 查询锁具指纹报警关联用户接口	37
11. 高级设置——记录查询	39
11.1 查询锁具开门记录	39
11.2 查询锁具亲情记录	39
11.3 查询锁具报警记录	40
12. 智能钥匙（ 长按至灯闪烁 ）	42
12.1 查询设备下的智能钥匙列表	42
12.2 查询智能钥匙的关联信息	42
12.3 扫描智能钥匙	43
12.4 连接钥匙	44
12.5 增加智能钥匙	44
12.6 智能钥匙重命名	45
12.7 启用智能钥匙	45
12.8 禁用智能钥匙	46
12.9 智能钥匙删除锁具	46
12.10 锁具删除智能钥匙	47

1.用户注册登陆

1.1 生成验证码

```
+(void)getAuthCodeRequestWithAccount:(NSString *)account withType:(NSString *)type withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
account	NSString	账号
type	NSString	1 注册 2 忘记密码

消息返回：

```
response: {
    *                status : 1 成功 0 失败
                    msg: 返回话述
                    data: {验证码（例如2399）}
}
error: 错误信息
```

1.2 验证码校验

```
+(void)checkMessageNum:(NSString *)account Type:(NSString *)type Code:code withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
account	NSString	账号
type	NSString	1 注册 2 忘记密码
Code	NSString	验证码

消息返回：

```
response: {
    *                status : 1 成功 0 失败
                    msg: 返回话述
                    data: {}
}
error: 错误信息
```

1.3 用户注册

```
+(void)appRegisterWithAccount:(NSString *)account Password:(NSString *)password withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
account	NSString	账号
password	NSString	密码

消息返回:

```
response: {
    *                status : 1 成功  0 失败
                    msg: 返回话述
                    data:{
                        account: 账号
                        iconurl: 头像
                        nickname: 昵称
                        password: 密码
                    }
}
error: 错误信息
```

1.4 用户登陆

```
+(void)appLoginWithAccount:(NSString *)account PSW:(NSString *)password withCompletionHandler:(void (^)(NSString *response, NSString *error))block
```

参数名	数据类型	意义
account	NSString	账号
password	NSString	密码

消息返回:

```
response: {
    *                status : 1 成功  0 失败
                    msg: 返回话述
                    data:{
```

```

        account:    账号
        iconurl:    头像
        nickname:   昵称
        password:   密码
        mobile:     手机号
        item6:      是否物业业主的标志(1:是 0:不是)
        type:       (1:普通用户)
        item2:      (16:控客)
        cardnum:    身份证号
        name:       姓名
        useraddress: 地址
    }
}
error: 错误信息

```

1.5 修改密码

```

+ (void)updateUserPasswordWithAccount:(NSString *)account Password:(NSString
*)password completionHandler:(void (^)(NSString *response, NSString
*error))block;

```

参数名	数据类型	意义
account	NSString	账号
password	NSString	密码

消息返回:

```

response: {
    *                status : 1 成功  0 失败
                    msg: 返回话述
                    data:{
                        }
    }
error: 错误信息

```

1.6 头像上传

```

+ (void)uploadUserPhotoWithAccount:(NSString *)account Filename:(NSString

```

```
*)filename completionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
account	NSString	账号
Filename	NSString	文件名

消息返回:

```
response: {
    *                status : 1 成功  0 失败
                    msg: 返回话述
                    data:{头像网络路径}
}
error: 错误信息
```

1.7 个人信息完善接口

```
+(void)updateUserFullInfoWithUseraddress:(NSString *)useraddress
Nickname:(NSString *)nickname Account:(NSString *)account Cardnum:(NSString *)cardnum WithName:(NSString *)name completionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
Useraddress	NSString	地址
Nickname	NSString	昵称
Account	NSString	账号
Cardnum	NSString	身份证号码
name	NSString	名称

消息返回:

```
response: {
    *                status : 1 成功  0 失败
                    msg: 返回话述
                    data:{ }
}
error: 错误信息
```

1.8 用户注销

```
+(void)logoutWithAccount:(NSString *)account completionHandler:(void
```

(^)(NSString *response, NSString *error))block

参数名	数据类型	意义
account	NSString	账号

消息返回：

```
response: {  
  *          status : 1 成功  0 失败  
             msg: 返回话述  
             data:{ }  
}
```


2.初始化

2.1 锁具配对获取 deviceidentifier

```
+(void)searchBylockseq:(NSString *)lockseq scantype:(NSString *)scantype
withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
lockseq	NSString	锁具 mac
scantype	NSString	11:锁具

消息返回:

```
response: {
    *                status : 1 成功  0 失败
                    msg: 返回话述
                    data: {deviceidentifier数据}
}
error: 错误信息
```

2.2 获取设备信息

```
+(void)showHomePageWithAccount:(NSString*)account
withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
account	NSString	账号

消息返回:

```
response: {
    *                status : 1 成功  0 失败
                    msg: 返回话述
                    data: [ {
                        lockTemppassNum: 临时密码剩余数目
                        phoneuserdnum: 已添加用户数
                        usOpenType: 用户自定义开门类型(当前开门类型)
                        manageaccount: 锁具管理账号
                        metertype: 锁具类型
                    }
```

```

    remark: 备注
    usLockname: 用户自定义的锁具名称
    account: 账号
    lockmac: 锁具mac
    softwareVersion: 锁具上的软件版本
    lockManagepwd: 管理密码
    fingerunuserdnum: 指纹未使用数量
    lockPassword: 锁具密码
    lockLoveuserNum: 亲情用户数量
    lockname: 锁具名称
    shareUser: 分享的用户（临时密码是当前账号）
    lockWifistate: wifi设置状态(1: 启用 0: 禁用)
    channelpassword: 锁具信道密码
    opentypePhone: 手机开门方式
    fingeruserdnum: 指纹使用数量
    opentypeKey: 锁具开门方式
    userHandPassword: 手势密码
    phoneunuserdnum: 用户剩余可添加
    usOpenMusic: 开门音效
    usertimeset: 用户时效
    lockAlarmpassword: 报警密码
    managepassword: 锁具管理密码
    opentypeFinger: 指纹开门方式
    seniorPrivilege: 锁的高级权限（1:有 0:没有）
    unixTime: 锁具的添加时间
    deviceType: 设备类型(11为智能锁, temp是临时钥匙)
    useropentype: 用户的开门方式
    lockWifipwd: wifi密码
    lockWifissid: wifi的ssid
    manageaccountname: 锁具管理账号名
    usOpenPwd: 用户手机开门密码
    lockAlarmpasswordstate: 报警密码状态
    lockLat: 所在经度
    routerid: 是否开启普通开门记录上传
    lockLon: 所在纬度
    updateFlag: 是否为最新版本
    lockHost: 临时钥匙分享者
  }
  { ... }
]
}
error: 错误信息

```

3.添加设备

3.1 添加设备

```
+ (void)addDeviceWithAccount:(NSString *)account nickname:(NSString
*)nickname locknameString:(NSString *)locknameString lockmacString:(NSString
*)lockmacString lockIdentifier:(NSString *)lockIdentifier
deviceType:(NSString*)deviceType lockLatitude:(NSString *)lockLatitude
lockLongitude:(NSString *)lockLongitude softwareVersion:(NSString
*)softwareVersion withCompletionHandler:(void (^)(NSString *response,
NSString *error))block;
```

参数名	数据类型	意义
account	NSString	帐号
nickname	NSString	昵称
locknameString	NSString	锁具名称
lockmacString	NSString	锁具 mac
lockIdentifier	NSString	设备 deviceId
deviceType	NSString	锁具型号(例如: T700、S800)
lockLatitude	NSString	纬度
lockLongitude	NSString	经度

消息返回:

```
response: {
    *          status : 1 成功 0 失败
              msg: 返回话述
              data: {
                    phoneuserdnum: 已添加用户数
                    manageaccount: 锁具管理账号
                    metertype: 锁具类型
                    remark: 备注
                    lockmac: 锁具mac
                    softwareVersion: 锁具上的软件版本
                    fingerunuserdnum: 指纹未使用数量
                    lockPassword: 锁具密码
                    lockname: 锁具名称
                    channelpassword: 锁具信道密码
                    opentypePhone: 手机开门方式
                    fingeruserdnum: 指纹使用数量
                    phoneunuserdnum: 用户剩余可添加
                    lockAlarmpassword: 报警密码
                    managepassword: 锁具管理密码
```

```
opentypeFinger: 指纹开门方式
time: 锁具的添加时间
lockWifipwd: wifi密码
lockWifissid: wifi的ssid
manageaccountname: 锁具管理账号名
lockAlarmpasswordstate: 报警密码状态
lockLat: 所在经度
lockLon: 所在纬度
type: 类型: 11
lockPasswordstate: 锁具密码状态
}
```

```
}
error: 错误信息
```

4.手机开门

4.1 确定的 mac 地址的手机开门

```
+(void)openDoor:(NSString *)lockseq deviceIdentifier:(NSString *)deviceIdentifier channelPwd:(NSString *)channelPwd Mobile:(NSString *)mobile withCompletionHandler:(void (^)(NSString *response, NSString *error))block
```

参数名	数据类型	意义
lockseq	NSString	锁具 mac
deviceIdentifier	NSString	设备标志
channelPwd	NSString	信道密码
Mobile	NSString	账号（临时钥匙传分享用户账号）

消息返回：

```
response: {
    *                status : 1 成功  0 失败
                    msg: 返回话述
                    data: {}
}
error: 错误信息
```

5. 个性设置

5.1 修改锁具名称

```
+(void)resetLockName:(NSString *)lockseq Newlockname:(NSString *)newlockname
withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
lockseq	NSString	锁具 mac
Newlockname	NSString	锁具名称

消息返回：

```
response: {
    *                status : 1 成功  0 失败
                    msg: 返回话述
                    data: {}
}
error: 错误信息
```

5.2 修改用户个性化手机开门方式

```
+(void)upUserOpenType:(NSString *)account Lockseq:(NSString *)lockseq
OpenType:(NSInteger)opentype withCompletionHandler:(void (^)(NSString
*response, NSString *error))block;
```

参数名	数据类型	意义
lockseq	NSString	锁具 mac
OpenType	NSString	开门方式(1: 摇一摇 2: 点一点: 3: 密码 4: 手势)
account	NSString	账号

消息返回：

```
response: {
    *                status : 1 成功  0 失败
                    msg: 返回话述
                    data: {}
}
error: 错误信息
```

5.3 修改用户个性化开门密码

```
+(void)upUserOpenPwd:(NSString *)account Lockseq:(NSString *)lockseq
Password:(NSString *)password withCompletionHandler:(void (^)(NSString
*response, NSString *error))block;
```

参数名	数据类型	意义
lockseq	NSString	锁具 mac
Password	NSString	密码
account	NSString	账号

消息返回：

```
response: {
    *                status : 1 成功  0 失败
                    msg: 返回话述
                    data: {}
}
error: 错误信息
```

5.4 修改用户手势密码

```
+(void)upUserHandOpenPwd:(NSString *)account Lockseq:(NSString *)lockseq
Openpwd:(NSString *)openpwd withCompletionHandler:(void (^)(NSString
*response, NSString *error))block;
```

参数名	数据类型	意义
lockseq	NSString	锁具 mac
Openpwd	NSString	开门密码
account	NSString	账号

消息返回：

```
response: {
    *                status : 1 成功  0 失败
                    msg: 返回话述
                    data: {}
}
error: 错误信息
```

5.5 手势密码开门校验（也可以本地校验）

```
+(void)checkHandOpenPwd:(NSString *)account Lockmac:(NSString *)lockmac  
Openpwd:(NSString *)openpwd withCompletionHandler:(void (^)(NSString  
*response, NSString *error))block;
```

参数名	数据类型	意义
lockseq	NSString	锁具 mac
Openpwd	NSString	开门密码
account	NSString	账号

消息返回：

```
response: {  
    *                status : 1 成功  0 失败  
                    msg: 返回话述  
                    data: {}  
}  
error: 错误信息
```


6.高级设置——用户模块

6.1 获取当前锁具的用户列表

```
+(void)getaccountListBylockseq:(NSString*)lockseq
withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
lockseq	NSString	锁具 mac

消息返回：

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
              data: [ {iconurl: 头像url,
                        item1:受否有高级权限  1为管理员  0为普通用户
                        lockseq: 锁具mac
                        manageaccount:房主账号
                        nickname: 用户昵称
                        useraccount: 用户账号
                        useropentype: 用户开门方式: 1指纹 2蓝牙 3指纹
                        + 蓝牙
                        usertimeset: 用户开门时效
                        } ,
                    {...} ,
                    {...} ]
```

例如：

```
[{"iconurl":"http://cdn.dsmzg.com:80/upload/debug/userphotos/15899138509163
946..png","id":272,"item1":"1","item2":"","level":0,"lockseq":"D8:E1:CB:DC:
59:EB","loveAccountFlag":0,"manageaccount":"15257121663","nickname":"152571
21663","remark":"锁具主人","remoteopenlock":0,"state":1,"time":"2017-04-10
11:12","type":2,"useraccount":"15257121663","useropentype":3,"usertimeset":
""},
{"iconurl":"http://121.43.225.110:12780/upload/debug/userphotos/20170328131
8395151490678436124","id":297,"item1":"0","item2":"","level":0,"lockseq":"D
8:E1:CB:DC:59:EB","loveAccountFlag":1,"manageaccount":"15257121663","nickna
me":"波波","remark":"锁具用户
","remoteopenlock":0,"state":1,"time":"2017-04-11
10:32","type":1,"useraccount":"18969079030","useropentype":3,"usertimeset":
""}]
```

```

    })
    }
error: 错误信息

```

6.2 添加用户

```

+ (void)doSingleUserAddBylockseq: (NSString*)lockseq
withDeviceIdentifier: (NSString*)deviceIdentifier
withManageaccount: (NSString *)manageaccount withName: (NSString *)username
withMobile: (NSString *)usermobile withCompletionHandler: (void (^)(NSString
*response, NSString *error))block;

```

参数名	数据类型	意义
lockseq	NSString	锁具 mac
deviceIdentifier	NSString	设备标志
manageaccount	NSString	操作者账号
username	NSString	添加用户昵称
usermobile	NSString	添加用户账号

```

消息返回:
response: {
    *                status : 1 成功  0 失败
                    msg: 返回话述
}
error: 错误信息

```

6.3 删除用户

```

+ (void)doSingleUserDeleteBylockseq: (NSString *)lockseq
withDeviceIdentifier: (NSString *)deviceIdentifier withMobile: (NSString
*)usermobile withType: (NSString *)type withCompletionHandler: (void
^)(NSString *response, NSString *error))block;

```

参数名	数据类型	意义
lockseq	NSString	锁具 mac
deviceIdentifier	NSString	设备标志

usermobile	NSString	删除用户账号
type	NSString	删除用户的用户类型（1 是房主 2 普通用户）

消息返回：

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
}
error: 错误信息
```

6.4 用户重命名

```
+(void)updateUserLockNickName:(NSString *)account Lockseq:(NSString
*)lockseq Newnickname:(NSString *)newnickname withCompletionHandler:(void
(^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
lockseq	NSString	锁具 mac
account	NSString	账号
newnickname	NSString	新昵称

消息返回：

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
}
error: 错误信息
```

6.5 设置用户开门方式

```
+(void)doUserOpenTypeUpdateBylockseq:(NSString *)lockseq
withDeviceIdentifier:(NSString *)deviceIdentifier withAccount:(NSString
*)account withOpentype:(NSInteger )opentype
isChangeTypeByFingerPrint:(BOOL )isChangeTypeByFingerPrint
withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
lockseq	NSString	锁具 mac

deviceIdentifier	NSString	设备标志
account	NSString	账号
opentype	NSString	开门方式 1 指纹 2 蓝牙 3 指纹+蓝牙
isChangeTypeByFingerPrint	BOOL	NO 指纹指纹开门权限无变更时 YES 有变更

消息返回:

```
response: {
    *          status : 1 成功 0 失败
              msg: 返回话述
}
```

error: 错误信息

6.6 设置用户开门时效

```
+(void)doUserTimePeriodUpdateBylockseq:(NSString *)lockseq
withDeviceIdentifier:(NSString *)deviceIdentifier withAccount:(NSString
*)currentUserLockAccount withTimePeriod:(NSString *)
currentUserLockTimePeriod withCompletionHandler:(void (^)(NSString
*response, NSString *error))block;
```

参数名	数据类型	意义
lockseq	NSString	锁具 mac
deviceIdentifier	NSString	设备标志
account	NSString	账号
currentUserLockTimePeriod	NSString	开门时效

消息返回:

```
response: {
    *          status : 1 成功 0 失败
              msg: 返回话述
}
```

error: 错误信息

6.7 添加指纹

```
+(void)doSingleFingerInputBylockseq:(NSString *)lockseq
withDeviceIdentifier:(NSString *)deviceIdentifier rmobile:(NSString
*)rmobile currentUserLockAccount:(NSString *)currentUserLockAccount
withSoftwareVersion:(NSString *)softwareVersion;
```

参数名	数据类型	意义
lockseq	NSString	锁具 mac
deviceIdentifier	NSString	设备标志
rmobile	NSString	当前操作者账号
currentUserLockAccount	NSString	指纹所属账号
SoftwareVersion	NSString	锁具版本号

消息返回：

收到名字为@ “RecordFingerInfo” 的通知

```
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(receiveWeekNotification:)
name:@"RecordFingerInfo" object:nil];
```

```
- (void)receiveWeekNotification:(NSNotification *)noti{
    NSLog(@"noti.object =%@",noti.object);
}
```

```
noti.object: {
    *          status : 1 成功  0 失败  2 录制过程中
              data:fingerTime: 指纹录入次数（status值为2时）
              msg: 返回话述
    }
```

6.8 删除指纹

```
+(void)doSingleFingerDeleteByDeviceIdentifier:(NSString *)deviceIdentifier
withFingerID:fingerID fingerlockid:fingerlockid withCompletionHandler:(void
(^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
deviceIdentifier	NSString	设备标志
fingerID	NSString	指纹 id

fingerlockid	NSString	锁具指纹序列号
--------------	----------	---------

消息返回：

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
}
error: 错误信息
```

6.9 指纹重命名

```
+(void)updateFingerName:(NSString *)fingerid Fingername:(NSString
*)fingername FingerAccount:(NSString *)fingerAccount
withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
fingername	NSString	指纹名称
fingerID	NSString	指纹 id
FingerAccount	NSString	指纹账号

消息返回：

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
}
error: 错误信息
```

6.10 获取指纹列表

```
+(void)getAccountFinger:(NSString *)lockseq FingerAccount:(NSString
*)fingerAccount withCompletionHandler:(void (^)(NSString *response, NSString
*error))block;
```

参数名	数据类型	意义
lockseq	NSString	锁具 mac
FingerAccount	NSString	指纹账号

消息返回：

```

response: {
    *      status : 1 成功  0 失败
           msg: 返回话述
           data: [ {fingerId: 指纹ID,
                    fingerLockId:锁具指纹序列号
                    fingerName: 指纹名称
                    fingerNum:房主账号
                    lockSeq: 锁具mac
                    manageAccount: 房主账号
                    userName: 用户昵称
                    useraccount: 用户账号
                    } ,
                   { ... } ,
                   { ... } ]
}

```

例如:

```

"data":[{"deleteState":0,"fingerId":159,"fingerLockId":"00
00 00 00-00 00 00 01-00 00 00 02","fingerName":"指纹
1","fingerNum":0,"fingerType":1,"item1":"","item2":"0","item3":"0","item4":
"3","item5":"","lockSeq":"D8:E1:CB:DC:59:EB","manageAccount":"15257121663",
"remark":"","userAccont":"15257121663","userName":""},
{"deleteState":0,"fingerId":160,"fingerLockId":"00 00 00
03-00 00 00 04-00 00 00 05","fingerName":"指纹
2","fingerNum":0,"fingerType":1,"item1":"","item2":"0","item3":"0","item4":
"3","item5":"","lockSeq":"D8:E1:CB:DC:59:EB","manageAccount":"15257121663",
"remark":"","userAccont":"15257121663","userName":""}]
}

```

error: 错误信息

6.11 高级设置验证

```

+ (void)seniorCheckByAccount:(NSString *)account Lockseq:(NSString *)lockseq
withCompletionHandler:(void (^)(NSString *response, NSString *error))block;

```

参数名	数据类型	意义
account	NSString	操作者账号
Lockseq	NSString	锁具 mac

消息返回:

```

response: {
    *      status : 1 成功  0 失败
           msg: 返回话述

```

}

error: 错误信息

7.高级设置——开门密码模块

7.1 设置开门密码（锁具上）

```
+(void)lockOpenPwdUpdateByLockSeq:(NSString *)lockseq
withDeviceIdentifier:(NSString *)deviceIdentifier withPwd:(NSString
*)lockOpenPwd withCompletionHandler:(void (^)(NSString *response, NSString
*error))block;
```

参数名	数据类型	意义
lockseq	NSString	锁具 mac
deviceIdentifier	NSString	设备标志

消息返回：

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
}
error: 错误信息
```

7.2 生成临时密码

```
+(void)doGenerateTempSecretKeysByLocseq:(NSString *)lockseq
withDeviceIdentifier:(NSString *)deviceIdentifier withAccount:(NSString
*)account withsoftwareVersion:(NSString *)softwareVersion
withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
lockseq	NSString	锁具 mac
deviceIdentifier	NSString	设备标志
account	NSString	账号
softwareVersion	NSString	锁具版本号

消息返回：

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
              data: 十组临时密码 以“|”间隔
    例如 data:
    537357|773752|707068|252535|711193|070312|932770|301205|164664|418774
}
error: 错误信息
```

7.3 临时钥匙授权

```
+(void)insertLockTempUserItemWithAccount:(NSString *)account
AuthMobile:(NSString *)authMobile Lockmac:(NSString *)lockmac
nickName:(NSString *)nickname completionHandler:(void (^)(NSString
*response, NSString *error))block;
```

参数名	数据类型	意义
account	NSString	操作者账号
authMobile	NSString	被授权人手机号
lockmac	NSString	锁具 Mac
nickName	NSString	用户昵称

消息返回:

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
}
error: 错误信息
```

7.4 临时密码授权

```
+(void)getDistributeAuthToMate:(NSString *)account Lockseq: (NSString
*)lockseq completionHandler:(void (^)(NSString *response, NSString
*error))block;
```

参数名	数据类型	意义
account	NSString	操作者账号
Lockseq	NSString	锁具 mac

消息返回:

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
              data: 临时密码
}
error: 错误信息
```

8. 高级设置——锁具信息模块

8.1 获取锁具详情

```
+(void)getLockMsgByMac:(NSString *)lockmac completionHandler:(void
(^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
Lockseq	NSString	锁具 mac

消息返回：

```
response: {
    *      status : 1 成功  0 失败
    msg: 返回话述
    data: [ {channelpassword:
              fingerunuserdnum: 剩余可使用指纹数
              lockAlarmpassword: 锁具报警密码
              lockAlarmpasswordstate: 锁具报警功能是否启用
            softwareVersion: 锁具版本号
            metertype: 锁具型号
            lockseq: 锁具mac
            lockWifipwd: Wi-Fi密码
            lockWifissid: wifi名字
            lockWifistate: Wi-Fi功能是否启用
            lockLat: 经度
            lockLon: 纬度
          } ]
```

例如：

```
data":[{"channelpassword":"14a62334","fingerunuserdnum":78,"fingeruserdnum":2,"hardwareVersion":"","id":169,"item1":"","item2":"","keyunyserdnum":0,"keyuserdnum":0,"lockAlarmpassword":"","lockAlarmpasswordstate":0,"lockLat":"30.177273","lockLon":"120.204981","lockManagepwd":"","lockPassword":"123546","lockPasswordstate":0,"lockSalescode":"","lockSalesname":"","lockWifipwd":"","lockWifissid":"","lockWifistate":0,"lockmac":"D8:E1:CB:DC:59:EB","lockname":"指纹锁","lockseq":"D8:E1:CB:DC:59:EB","manageaccount":"15257121663","manageaccountname":"15257121663","managepassword":"","metertype":"T700_0","opentypeFinger":1,"opentypeKey":0,"opentypePhone":1,"parentid":0,"phoneunuserdnum":148,"phoneuserdnum":2,"remark":"","routerid":0,"softwareVersion":"A0.1.007_20161201","state":1,"time":"2017-04-10 11:12","type":11}]
}
```

error: 错误信息

8.2 固件更新

8.2.1 检测是否有新版本固件

```
+(void)lockFirewareUpdateCheckBylockseq:(NSString *)lockseq
withDeviceIdentifier:(NSString *)deviceIdentifier
withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
lockseq	NSString	锁具 Mac
deviceIdentifier	NSString	设备标志

消息返回：

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
              data: [ {
                      appName: 文件名
                      appRefreshtime: 更新时间
                      appUrl: 网络下载路径
                      appVersion: 固件版本号
                    } ]
```

例如：

```
[{"appName":"T700更新程序","appRefreshtime":"2016-08-11
09:15","appState":1,"appType":2,"appUrl":"http://dsmzg.com:8
080/base/softUp/T700_app20161201.zip","appVersion":"V0.1.003
_20161201","id":8,"item1":"T700","item2":"1.wifi开启关闭
<br>2. 临时秘钥<br>3. 推送检测","remark":"T700_0"}]
```

```
}
```

error: 错误信息

8.2.2 新版本固件下载

```
+(void)downloadFirmwareFile:(NSString *)fileUrl Filename:(NSString
*)filename withCompletionHandler:(void (^)(NSString *response, NSString
*error))block
```

参数名	数据类型	意义
fileUrl	NSString	网络下载路径

Filename	NSString	文件名字
----------	----------	------

消息返回：

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
    }
error: 错误信息
```

8.2.3 固件更新

```
+(void)UpdateLockseq:(NSString *)lockseq withDeviceIdentifier:(NSString
*)deviceIdentifier withAppName:(NSString *)appName withAppVersion:(NSString
*)appVersion withCompletionHandler:(void (^)(NSString *response, NSString
*error))block;
```

参数名	数据类型	意义
lockseq	NSString	锁具 Mac
deviceIdentifier	NSString	设备标志
appName	NSString	文件名
appVersion	NSString	固件版本

消息返回：

error: 错误信息

8.3 注销锁具

```
+(void)doLockLogoutByLocseq:(NSString *)lockseq
withDeviceIdentifier:(NSString *)deviceIdentifier
withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
lockseq	NSString	锁具 Mac
deviceIdentifier	NSString	设备标志

消息返回：

```
response: {
```

```

*           status : 1 成功  0 失败
           msg: 返回话述
        }
error: 错误信息

```

8.4 获取 WIFI

```

+(void)getCurrentWifiNameWithCompletionHandler:(void (^)(NSString *response,
NSString *error))block;

```

参数名	数据类型	意义
-----	------	----

```

消息返回:
response: {
*           status : 1 成功  0 失败
           msg: 返回话述
           data: Wi-Fi名称
        }
error: 错误信息

```

8.5 设置锁具 WIFI

```

+(void)doLockWifiUpdateByLocseq:(NSString *)lockseq
withDeviceIdentifier:(NSString *)deviceIdentifier withSSID:(NSString *)ssid
withPwd:(NSString *)pwd withCompletionHandler:(void (^)(NSString *response,
NSString *error))block;

```

参数名	数据类型	意义
lockseq	NSString	锁具 Mac
deviceIdentifier	NSString	设备标志
SSID	NSString	Wi-Fi 名称
Pwd	NSString	Wi-Fi 密码

```

消息返回:
response: {
*           status : 1 成功  0 失败
           msg: 返回话述
        }
error: 错误信息

```

8.6 WIFI 推送检测

```
+(void)doLockWifiTestByDeviceIdentifier:(NSString *)deviceIdentifier
withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
deviceIdentifier	NSString	设备标志

消息返回：

```
response: {
    *          status : 1 发送检测命令成功  0 发送检测命令失败
              msg: 返回话述
}
error: 错误信息
```

8.7 开门记录上传功能开启与关闭

```
+(void)doLockOpenDoorRecordControlBylockseq:(NSString *)lockseq
withDeviceIdentifier:(NSString *)deviceIdentifier
isUploadOpenDoorRecord:(BOOL )isUploadOpenDoorRecord
withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
lockseq	NSString	锁具 Mac
deviceIdentifier	NSString	设备标志
isUploadOpenDoorRecord	BOOL	是否启用

消息返回：

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
}
error: 错误信息
```

9.高级设置——亲情设置模块

9.1 设置亲情关联

```
+ (void)doinsertLoveAccountRelationItem:(NSString *)lockMac
deviceIdentifier:(NSString *)deviceIdentifier account:(NSString *)account
accountName:(NSString *)accountName senderAccount:(NSString *)senderAccount
senderName:(NSString *)senderName receiverAccount:(NSString
*)receiverAccount receiverName:(NSString *)receiverName
loveTimeset:(NSString *)loveTimeset completionHandler:(void (^)(NSString
*response, NSString *error))block;
```

参数名	数据类型	意义
lockMac	NSString	锁具 Mac
deviceIdentifier	NSString	设备标志
account	NSString	操作者账号
accountName	NSString	操作者昵称
senderAccount	NSString	发送者账号
senderName	NSString	发送者昵称
receiverAccount	NSString	接受者账号
receiverName	NSString	接受者昵称
loveTimeset	NSString	亲情时效

消息返回：

```
response: {
    *          status : 1 成功 0 失败
              msg: 返回话述
}
error: 错误信息
```

9.2 删除亲情关联

```
+ (void)doLoveRelationDeleteByLockSeq:(NSString *)lockseq
withDeviceIdentifier:(NSString *)deviceIdentifier
withLoveRelationID:(NSString *)loveRelationID completionHandler:(void
(^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
-----	------	----

lockseq	NSString	锁具 Mac
deviceIdentifier	NSString	设备标志
loverelationID	NSString	亲情关联 id

消息返回:

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
}
error: 错误信息
```

9.3 获取设备的亲情关联

```
+(void)getLoveAccountRelationByLockSeq:(NSString *)lockseq
withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
lockseq	NSString	锁具 Mac

消息返回:

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
              data: [ {"lvAccount":新增亲情关联的用户账号
                      "lvAccountname":新增亲情关联的用户姓名
                      "lvId":亲情关联ID
                      "lvLockseq":锁具Mac
                      "lvLoveaccounttimeset":亲情关联时间
                      "lvPassiverelationaccount":接收者账号
                      "lvPassiverelationaccountname":接收者姓名
                      "lvRelationaccount":发送者账号
                      "lvRelationaccountname":发送者姓名
                      "lvRelationtime":新增关联时间
                    } ,
                    { ... } ,
                    { ... } ]
}
error: 错误信息
```

10.高级设置——报警模块

10.1 锁具报警密码设置

```
+ (void)doAlarmPwdUpdateByLockSeq:(NSString *)lockseq
withDeviceIdentifier:(NSString *)deviceIdentifier withAlarmPwd:(NSString
*)alarmpwd withCompletionHandler:(void (^)(NSString *response, NSString
*error))block;
```

参数名	数据类型	意义
deviceIdentifier	NSString	设备标识
lockseq	NSString	锁具 mac
alarmpwd	NSString	报警密码

消息返回：

```
response: {
    *           status : 1 成功  0 失败
                msg: 返回话述
}
error: 错误信息
```

10.2 锁具报警密码启用 / 关闭

```
+ (void)doAlarmPwdEnableByLockSeq:(NSString *)lockseq
withDeviceIdentifier:(NSString *)deviceIdentifier withStatus:(NSString
*)status alarmPassword:(NSString *)alarmPassword
withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
deviceIdentifier	NSString	设备标识
lockseq	NSString	锁具 mac
alarmPassword	NSString	报警密码
status	NSString	0:关闭 1:开启

消息返回：

```
response: {
    *           status : 1 成功  0 失败
                msg: 返回话述
}
error: 错误信息
```

10.3 设置指纹报警关联

```
+ (void)doFingerAlarmRelationInsertByLockSeq:(NSString *)lockseq
Alarmaccount:(NSString *)alarmAccount Alarmaccountname:(NSString
*)alarmAccountname Alarmalarmrelation:(NSString *)alarmAlarmrelation
Alarmpassiverelation:(NSString *)alarmPassiverelation
Alarmpassiverelationname:(NSString *)alarmPassiverelationname
FingerName:(NSString *)fingerName deviceIdentifier:(NSString
*)deviceIdentifier withCompletionHandler:(void (^)(NSString *response,
NSString *error))block;
```

参数名	数据类型	意义
Alarmaccount	NSString	操作者账号
lockseq	NSString	锁具 mac
Alarmaccountname	NSString	操作者昵称
Alarmalarmrelation	NSString	指纹 id
Alarmpassiverelation	NSString	接收者账号
Alarmpassiverelationname	NSString	接收者昵称
FingerName	NSString	指纹名称
deviceIdentifier	NSString	设备标识

消息返回：

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
}
```

error: 错误信息

10.4 设置密码报警关联

```
+ (void)insertPasswdAlarmAccountRelationItem:(NSString *)alarmLockseq
Account:(NSString *)account Accountname:(NSString *)accountname
AlarmAccount:(NSString *)alarmAccount AlarmAccount:(NSString *)alarmName
withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
Account	NSString	操作者账号
alarmLockseq	NSString	锁具 mac
Accountname	NSString	操作者昵称
AlarmAccount	NSString	接收者账号
AlarmAccount	NSString	接收者昵称

消息返回：

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
}
error: 错误信息
```

10.5 取消指纹报警关联

```
+ (void)doFingerAlarmRelationDeleteByLockSeq:(NSString *)lockseq
withDeviceIdentifier:(NSString *)deviceIdentifier withFingerlockid:(NSString
*)fingerlockid withCompletionHandler:(void (^)(NSString *response, NSString
*error))block;
```

参数名	数据类型	意义
lockseq	NSString	锁具 mac
fingerlockid	NSString	关联 id
deviceIdentifier	NSString	设备标识

消息返回：

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
}
error: 错误信息
```

10.6 取消密码报警关联

```
+(void)delPwdAlarmRelation:(NSString *)lockseq AlarmAccount:(NSString
*)alarmAccount withCompletionHandler:(void (^)(NSString *response, NSString
*error))block;
```

参数名	数据类型	意义
Alarmaccount	NSString	密码报警关联账号
lockseq	NSString	锁具 mac

消息返回:

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
}
error: 错误信息
```

10.7 查询锁具密码报警关联用户接口

```
+(void)getPwdAlarmUserList:(NSString *)lockmac withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
lockmac	NSString	锁具 mac

消息返回:

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
              data:[
                  {
                      isLockAdmin: 是否锁具管理员: 0不是 1是
                      isLockOwner: 是否屋主: 0不是 1是
                      isPwdAlarmRelate: 是否密码报警关联用户 0不是 1是
                      lockUserId: 密码关联id
                      lockseq: 锁具mac
                      nickname: 用户昵称
                      useraccount: 用户账号
                  }
                  {...}
              ]
}
error: 错误信息
```

10.8 查询锁具指纹报警关联用户接口

```
+(void)getFingerAlarmUserList:(NSString *)lockmac
withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
lockmac	NSString	锁具 mac

消息返回:

```
response: {
  *      status : 1 成功  0 失败
        msg: 返回话述
        data:[
          {
            alarmAccount: 指纹报警添加人账号
            alarmAccountname: 指纹报警添加人名称
            alarmAlarmrelation: 锁具指纹ID
            alarmAlarmrelationname: 指纹名称
            alarmId: 指纹报警关联ID
            alarmItem1: 指纹来源: 1: 手机录入指纹 2: 锁具录入指纹
            alarmLockseq: 锁具mac
            alarmPassiverelation: 指纹报警关联用户账号
            alarmPassiverelationname: 指纹报警关联用户姓名
          }
          {...}
        ]
      }
error: 错误信息
```

11.高级设置——记录查询

11.1 查询锁具开门记录

```
+(void)getOpenDoorRecordList:(NSString *)lockMac withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
lockMac	NSString	锁具 mac

消息返回：

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
              data:[
                  {
                      account: 开门用户账号
                      content: 内容
                      iconurl: 头像
                      lockName: 锁具名称
                      lockseq: 锁具Mac
                      opentime: 开门时间
                      openway: 开门方式 （1:指纹 2:手机 3:锁具密码）
                      openwayName: 开门方式名称
                      recordId: 记录id
                      username: 开门用户昵称
                  }
                  {...}
              ]
    }
error: 错误信息
```

11.2 查询锁具亲情记录

```
+(void)getFamilyDoorRecordList:(NSString *)lockMac
withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
lockMac	NSString	锁具 mac

消息返回：

```
response: {
  *      status : 1 成功  0 失败
        msg: 返回话述
        data:[
            {
                account: 开门用户账号
                content: 亲情指纹名称
                iconurl: 头像
                lockName: 锁具名称
                lockseq: 锁具Mac
                opentime: 记录时间
                recordId: 记录id
                username: 开门用户昵称
            }
            {...}
        ]
    }
error: 错误信息
```

11.3 查询锁具报警记录

```
+(void)getFamilyDoorRecordList:(NSString *)lockMac
withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
lockMac	NSString	锁具 mac

消息返回：

返回类型值：

alarmType: 11: 撬锁报警 12: 假锁报警 13: 试开报警
 151: 指纹报警 152: 密码报警

alarmType =151 指纹报警 content: 指纹名称

alarmType =152 密码报警 content: ”密码开门”

alarmType = 11,12,13 锁具报警类型 content: 为空或者”锁具报警”

```
response: {
  *      status : 1 成功  0 失败
```



```
msg: 返回话述
data:[
    {
        content: 报警内容
        lockName: 锁具名称
        lockseq: 锁具Mac
        opentime: 记录时间
        recordId: 记录id
        username: 开门用户昵称
        alarmType: 报警记录类型
        alarmTypeName: 报警记录类型名称
    }
    {...}
]
error: 错误信息
```

12.智能钥匙（长按至灯闪烁）

12.1 查询设备下的智能钥匙列表

```
+(void)queryLockSmartkey:(NSString *)lockMac withCompletionHandler:(void (^)(NSString *response, NSString *errorMessage))block;
```

参数名	数据类型	意义
lockMac	NSString	锁具 mac

消息返回：

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
              data:[
                  {
                      id: 序号
                      keyPass: 智能钥匙密钥
                      state: 状态
                      lockMac: 锁具MAC地址
                      addTime: 添加时间
                      keyCount: 该智能钥匙已绑定锁具数目
                      keyName: 智能钥匙别名
                      keyCode: 智能钥匙编号
                  }
                  {...}
              ]
    }
error: 错误信息
```

12.2 查询智能钥匙的关联信息

```
+(void)querySmartkeyRelation:(NSString *)keyCode withCompletionHandler:(void (^)(NSString *response, NSString *errorMessage))block;
```

参数名	数据类型	意义
keyCode	NSString	智能钥匙编号

消息返回:

```
response: {
  *      status : 1 成功  0 失败
        msg: 返回话述
        data:[
            {
                keyCode: 智能钥匙编号
                keyName: 智能钥匙别名
                lockMac: 锁具mac
                id: 序号
                keyPass: 智能钥匙密钥
                state: 状态
                addTime: 添加时间
            }
            {...}
        ]
    }
error: 错误信息
```

12.3 扫描智能钥匙

```
+ (void)scanBleWithCompletionHandler:(void (^)(NSMutableArray *Array,
NSString *error))block;
```

参数名	数据类型	意义

消息返回:

```
Array: {
    [
        NSObject:{
            peripheral(CBPeripheral): 蓝牙周边信息
            lockname: 智能钥匙名字
            lockIdentifier: 智能钥匙identity
        }
        {...}
    ]
}
error: 错误信息
```

12.4 连接钥匙

```
+(void)connect:(NSString *)smartKeyName BleArray:(NSMutableArray *)BleArray;
```

参数名	数据类型	意义
BleArray	NSMutableArray	12.3 中获取的 Array

消息返回：

需要添加通知，通知名字为 **smartKey**

通知中添加：

```
NSUserDefaults *defaults = [NSUserDefaults standardUserDefaults];
int type = [[defaults objectForKey:@"smartKeysss"] intValue];
type: 0:未连接，需要重试
      1:连接成功
      2:连接错误，需要重试
```

12.5 增加智能钥匙

```
+(void)addSmartKey:(NSMutableArray *)BleArray lockmac:(NSString *)lockmac
deviceIdentifier:(NSString *)deviceIdentifier channelpassword:(NSString
*)channelpassword withCompletionHandler:(void (^)(NSString *response,
NSString *error))block;
```

参数名	数据类型	意义
BleArray	NSMutableArray	12.3 中获取的 Array
lockmac	NSString	锁具 mac
deviceIdentifier	NSString	设备 deviceIdentifier
channelpassword	NSString	锁具信道密码

需要锁具和智能钥匙同时进入设置状态

消息返回：

```
response: {
    *          status : 1 成功 0 失败
              msg: 返回话述
              data:[]
            }
error: 错误信息
```

12.6 智能钥匙重命名

```
+(void)updateSmartKeyName:(NSString *)lockMac keyCode:(NSString *)keyCode
keyPass:(NSString *)keyPass keyName:(NSString *)keyName
withCompletionHandler:(void (^)(NSString *response, NSString
*errorMessage))block;
```

参数名	数据类型	意义
lockMac	NSString	锁具 MAC 地址
keyPass	NSString	智能钥匙密钥
keyCode	NSString	智能钥匙编号
keyName	NSString	智能钥匙别名

消息返回：

```
response: {
    *          status : 1 成功  0 失败
              msg: 返回话述
              data:[]
            }
error: 错误信息
```

12.7 启用智能钥匙

```
+(void)updateSmartkeyStateOn:(NSString *)lockmac deviceIdIdentifier:(NSString
*)deviceIdIdentifier keyPass:(NSString *)keyPass keyCode:(NSString *)keyCode
withCompletionHandler:(void (^)(NSString *response, NSString *error))block;
```

参数名	数据类型	意义
lockmac	NSString	锁具 MAC 地址
deviceIdIdentifier	NSString	设备 deviceIdIdentifier
keyPass	NSString	智能钥匙密钥
keyCode	NSString	智能钥匙编号

锁具进入设置状态

消息返回：

```
response: {
```

```

*          status : 1 成功  0 失败
          msg: 返回话述
          data:[]
        }
error: 错误信息

```

12.8 禁用智能钥匙

```

+(void) updateSmartkeyStateOff:(NSString *)lockmac
deviceIdentifier:(NSString *)deviceIdentifier keyPass:(NSString *)keyPass
keyCode:(NSString *)keyCode withCompletionHandler:(void (^)(NSString
*response, NSString *error))block;

```

参数名	数据类型	意义
lockmac	NSString	锁具 MAC 地址
deviceIdentifier	NSString	设备 deviceIdentifier
keyPass	NSString	智能钥匙密钥
keyCode	NSString	智能钥匙编号

锁具进入设置状态

```

消息返回:
response: {
*          status : 1 成功  0 失败
          msg: 返回话述
          data:[]
        }
error: 错误信息

```

12.9 智能钥匙删除锁具

```

+(void) deleteLockFromSmartKey:(NSString *)lockmac deviceIdentifier:(NSString
*)deviceIdentifier keyCode:(NSString *)keyCode withCompletionHandler:(void
(^)(NSString *response, NSString *error))block;

```

参数名	数据类型	意义
lockmac	NSString	锁具 MAC 地址
deviceIdentifier	NSString	钥匙 deviceIdentifier
keyPass	NSString	智能钥匙密钥
keyCode	NSString	智能钥匙编号

消息返回：

```
response: {  
    *          status : 1 成功  0 失败  
              msg: 返回话述  
              data: []  
            }  
error: 错误信息
```

12.10 锁具删除智能钥匙

```
+(void)deleteSmartKeyFromLock:(NSString *)lockmac deviceIdentifier:(NSString  
*)deviceIdentifier keyCode:keyCode keyPass:(NSString *)keyPass  
withCompletionHandler:(void (^)(NSString *response, NSString *error))block
```

参数名	数据类型	意义
lockmac	NSString	锁具 MAC 地址
deviceIdentifier	NSString	锁具 deviceIdentifier
keyPass	NSString	智能钥匙密钥
keyCode	NSString	智能钥匙编号

锁具进入设置状态

消息返回：

```
response: {  
    *          status : 1 成功  0 失败  
              msg: 返回话述  
              data: []  
            }  
error: 错误信息
```