# iOS SDK API文档

## 1. Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Platform | Language | Installation | Notes |
| iOS 14.0+ | Objective-C | Manual | Xcode 14.0+ is required. |

## 2.Usage

在的类方法中添加下面代码初始化SDK

MBCreationOptions \*option = [[MBCreationOptions alloc] init];  
[MBDeviceManager.shared centerManagerByOptions:option];

/// If true, display a warning dialog to the user when the `CBCentralManager` is instantiated if Bluetooth is powered off  
@property (nonatomic, assign) BOOL showPowerAlert;  
/// A unique identifier for the Central Manager that's being instantiated. This identifier is used by the system to  
/// identify a specific CBCentralManager instance for restoration and, therefore, must remain the same for  
/// subsequent application executions in order for the manager to be restored.  
///  
/// -Note: If no `.` Character is included, the Bundle Identifer is automatically added  
@property (nonatomic, copy ) NSString \*restoreIdentifier;

### Scan peripherals

### 开始扫描

[MBDeviceManager.shared scanForOptions:[MBScanOptions new] devices:^(MBScannedDevice \* device) {  
 NSLog(@"扫描到的眼镜MAC: %@", device.mac);   
 MBGlassesDevice \*gDevice = [[MBGlassesDevice alloc] initWithPeripheral:device.peripheral];  
 gDevice.deviceMac = device.mac;  
 self.device = gDevice;  
 }  
 } completion:^(BOOL success, NSError \* error) {  
 NSLog(@"扫描结果:%d, error: %@", success, error);  
 }];

/// Whether or not the scan should filter duplicate peripheral discoveries  
@property (nonatomic, assign) BOOL allowDuplicates;  
  
/// An array of service UUIDs that you want to scan for.  
/// The array is an instance of NSArray, and uses CBUUID objects to represent the UUIDs to scan for.  
/// Specifying this scan option causes the central manager to also scan for peripherals soliciting any of the services contained in the array.  
@property (nonatomic, copy ) NSArray<CBUUID \*> \*solicitedServiceUUIDs;

/// The peripheral to which the central is scanned.  
@property (nonatomic, strong) CBPeripheral \*peripheral;  
/// The Received Signal Strength Indicator (RSSI), in decibels, of the peripheral.  
@property (nonatomic, assign) NSInteger rssi;  
/// The mac of the peripheral.  
@property (nonatomic, copy) NSString \*mac;  
  
/// The name of the peripheral.  
@property (nonatomic, copy) NSString \*name;  
  
/// PID + UID  
@property (nonatomic, copy) NSString \*model;

### 停止扫描

[MBDeviceManager.shared stopScan];

## Start Connect/Bind/Unbind peripheral

### Connect/Bind peripheral

- (void)connectWithMac:(NSString \*)mac  
 userId:(NSInteger)userId  
 token:(NSString \*)token  
 identifier:(nullable NSString \*)identifier  
 completion:(nullable void(^)(\_Nullable id<MBBluetoothDevice> device, NSError \*\_Nullable error))completion;

#### Parameters

The mac of the peripheral.

App登录用户的唯一标识

通过http://api.cloud.meta-bounds.com/v1/terminals/check获取的设备Token

The UUID associated with the peripheral.

The completion handler to call when the connection is complete. This handler is executed on the main queue.

This completion handler takes the following parameters:

/// 设备唯一Id  
@property (nonatomic, copy, readonly) NSString \*deviceId;  
  
/// 设备mac地址  
@property (nonatomic, copy, nullable) NSString \*deviceMac;  
  
/// 设备连接状态  
@property (nonatomic, assign) MBConnectState state;  
  
/// 连接或发送、接收数据的外设  
@property (nonatomic, strong, readonly) CBPeripheral \*peripheral;  
  
/// Returns an id<MBBluetoothDevice> object initialized by the given peripheral.  
+ (id<MBBluetoothDevice>)deviceWithPeripheral:(CBPeripheral \*)peripheral;  
  
/// Returns an id<MBBluetoothDevice> object initialized by the given peripheral.  
- (id<MBBluetoothDevice>)initWithPeripheral:(CBPeripheral \*)peripheral;  
  
/// 连接当前设备  
- (void)connect:(nullable void(^)(BOOL, NSError \* \_Nullable))completion;  
  
/// scan到的服务  
/// - Parameter service: scan到的服务  
- (void)didDiscoverCharacteristicsForService:(CBService \*)service;  
  
/// 使能成功通知(cccd success)  
/// - Parameter characteristic: 使能成功的特征  
- (void)didUpdateNotificationStateForCharacteristic:(CBCharacteristic \*)characteristic;  
  
/// 接收到数据  
/// - Parameter characteristic: 接收到数据的特征  
- (void)didReceivedForCharacteristic:(CBCharacteristic \*)characteristic;  
  
/// 向该设备发送数据  
/// - Parameters:  
/// - frame: 设备通讯协议帧，默认写数据Type为`CBCharacteristicWriteWithoutResponse`  
/// - type: 写数据类型 `CBCharacteristicWriteWithoutResponse` 或 `CBCharacteristicWriteWithResponse`  
/// - completion: 回调  
- (void)sendFrame:(id<MBFrameProtocol>)frame completion:(nullable MBTaskCompletion)completion;  
  
/// 断开连接  
/// - Parameter shouldUnbind: 是否需要发送解绑命令  
- (void)disconnect:(BOOL)shouldUnbind;  
  
@optional  
  
/// BLE广播信息的制造商字段  
@property (nonatomic, copy, nullable) NSData \*manufacturerData;  
  
@property (nonatomic, copy, nullable) NSDictionary \*connectOpions;  
  
/// 成功打开L2CAP通道  
/// - Parameter channel:L2CAP通道  
- (void)didOpenL2CAPChannel:(CBL2CAPChannel \*)channel;  
  
/// 注册接收到外设的请求命令回调  
- (void)registerCommand:(UInt16)command tag:(nullable NSString \*)tag listener:(MBDidReceivedForCharacteristic)listener;  
/// 移除外设的请求命令回调  
- (void)removeListenForCommand:(UInt16)command tag:(nullable NSString \*)tag;  
/// 取消发送的数据帧，如果frames已经发送则无效  
- (void)cancelFrames:(NSArray<id<MBFrameProtocol>>\*)frames;

An error object that indicates why the connection failed, or  if the connection was successful.

### Bind peripheral

使用http://api.cloud.meta-bounds.com/v1/terminals/bind上传用户与外设的绑定关系，详情请参考：【云服务API接口文档】

### Unbind peripheral

使用http://api.cloud.meta-bounds.com/v1/terminals/unbind解除外设的绑定关系，同时调用的方法，参数值为

## Send/Receive Data

### 1. DeviceInfo(MBGlassesInfoService)

#### 1.1 获取设备信息

函数原型：+ requestDeviceInfo::(nullable void(^)(BOOL isEnabledDND, BOOL isEnabledWeather))completion

功能描述：获取设备信息

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL isEnabledDND, BOOL isEnabledWeather) | 勿扰模式是否打开.  天气提醒开关是否打开 |

#### 1.2 设置外设语言

函数原型：+ (void)syncAppLanguage:(UInt8)language completion:(nullable void(^)(BOOL))completion

功能描述：设置外设语言

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| language | UInt8 | 语言类型；0x00: 中文; 0x01: 英文 |
| completion | void(^)(BOOL success) | 语言是否设置成功 |

#### 1.3 设置时间

函数原型：+ (void)syncSystemTime:(nullable void(^)(BOOL))completion

功能描述：同步手机时间到眼镜

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL success) | 手机时间是否同步成功 |

#### 1.4 设置手机型号

函数原型：+ (void)syncPhoneMode:(nullable void(^)(BOOL))completion

功能描述：设置手机型号

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL success) | 手机型号是否同步成功 |

#### 1.5 设置勿扰模式开关

函数原型：+ (void)syncDNDMode:(BOOL)isQuiet completion:(nullable void(^)(BOOL))completion

功能描述：设置外设语言

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| isQuiet | BOOL | isQuiet为YES打开勿扰模式，反之关闭 |
| completion | void(^)(BOOL success) | 勿扰模式开关是否设置成功 |

#### 1.6 设置外设蓝牙名称

函数原型：+ (void)syncBluetoothName:(NSString \*)name completion:(nullable void(^)(BOOL))completion

功能描述：设置外设语言

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| name | NSString | 外设蓝牙名称 |
| completion | void(^)(BOOL success) | 蓝牙名称是否设置成功 |

#### 1.7 查询屏幕亮度

函数原型：+ (void)requestScreenBrightness:(nullable void(^)(NSInteger))completion

功能描述：查询屏幕亮度

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(NSInteger value) | 眼镜屏幕亮度；如果获取失败，则为-1 |

#### 1.8 设置屏幕亮度

函数原型：+ (void)syncScreenBrightness:(NSInteger)value completion:(nullable void(^)(BOOL))completion

功能描述：设置屏幕亮度

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| value | NSInteger | 屏幕亮度 |
| completion | void(^)(BOOL success) | 屏幕亮度是否设置成功 |

#### 1.9 查询自动息屏时间

函数原型：+ (void)requestScreenOffTime:(nullable void(^)(NSInteger))completion

功能描述：查询自动息屏时间

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(NSInteger value) | 眼镜自动息屏时间；如果获取失败，则为-1 |

#### 1.10 设置自动息屏时间

函数原型：+ (void)syncScreenBrightness:(NSInteger)value completion:(nullable void(^)(BOOL))completion

功能描述：设置自动息屏时间

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| value | NSInteger | 自动息屏时间 |
| completion | void(^)(BOOL success) | 屏幕自动息屏时间是否设置成功 |

#### 1.11 查询屏幕亮度自动调节模式

函数原型：+ (void)requestScreenBrightnessMode:(nullable void(^)(BOOL))completion

功能描述：查询屏幕亮度自动调节模式

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL isOn) | 屏幕亮度自动调节模式是否打开，YES为打开 |

#### 1.12 设置屏幕亮度自动调节模式

函数原型：+ (void)syncScreenBrightnessMode:(BOOL)isAuto completion:(nullable void(^)(BOOL))completion

功能描述：设置屏幕亮度自动调节模式

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| isAuto | BOOL | 屏幕亮度自动调节模式, YES: 打开, 反之关闭 |
| completion | void(^)(BOOL success) | 屏幕亮度自动调节模式是否设置成功 |

#### 1.13 查询屏幕点亮方式

函数原型：+ (void)requestScreenLighten:(nullable void(^)(NSInteger))completion

功能描述：查询屏幕亮度自动调节模式

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(NSInteger value) | 0x00: 关闭; 0x01: 左眼单目模式; 0x02: 右眼单目模式 |

#### 1.14 设置屏幕点亮方式

函数原型：+ (void)syncScreenLighten:(NSInteger)value completion:(nullable void(^)(BOOL))completion

功能描述：设置屏幕亮度自动调节模式

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| value | NSInteger | 0x00: 关闭; 0x01: 左眼单目模式; 0x02: 右眼单目模式 |
| completion | void(^)(BOOL success) | 屏幕亮度自动调节模式是否设置成功 |

#### 1.15 查询BT连接状态

函数原型：+ (void)requestBTState:(nullable void(^)(BOOL))completion

功能描述：查询屏幕亮度自动调节模式

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL isConnected) | BT是否已连接 |

#### 1.16 断开BT连接

函数原型：+ (void)syncDisconnectBT:(nullable void(^)(BOOL))completion

功能描述：断开BT连接

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL success) | 断开BT连接是否设置成功 |

#### 1.17 重新连接BT

函数原型：+ (void)requestReconnectBT:(nullable void(^)(BOOL))completion

功能描述：重新连接BT

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL success) | BT是否重连成功 |

#### 1.18 添加BT连接状态的监听

函数原型：+ (void)addBTStateListener:(void(^)(BOOL))listener;

功能描述：设置屏幕亮度自动调节模式

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL isConnected) | BT是否已连接 |

#### 1.19 移除BT连接状态的监听

函数原型：+ (void)removeBTStateListener

功能描述：移除BT连接状态的监听

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| -- | -- | -- |

#### 1.20 恢复出厂设置

函数原型：+ (void)requestFactoryReset:(nullable void(^)(BOOL))completion

功能描述：恢复出厂设置

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL success) | 恢复出厂设置是否成功 |

#### 1.21 恢复出厂设置-重启

函数原型：+ (void)requestFactoryReboot:(nullable void(^)(BOOL))completion;

功能描述：恢复出厂设置-重启

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL success) | 恢复出厂设置-重启是否设置成功 |

#### 1.22 同步App退出登录

函数原型：+ (void)syncLogout:(nullable void(^)(BOOL))completion;

功能描述： 同步App退出登录

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL success) | 同步App退出登录是否设置成功 |

#### 1.23 查询外设电量

函数原型：+ (void)requestBatteryInfo:(nullable void(^)(NSInteger, BOOL))completion

功能描述：查询外设电量

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(NSInteger value, BOOL isCharging) | 外设当前电量  外设是否在充电 |

#### 1.24 恢复出厂设置-重启

函数原型：+ (void)addBatteryListener:(void(^)(NSInteger, BOOL))listener;

功能描述：恢复出厂设置-重启

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL success) | 恢复出厂设置-重启是否设置成功 |

#### 1.25 同步App退出登录

函数原型：+ (void)removeBatteryListener;

功能描述： 同步App退出登录

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL success) | 同步App退出登录是否设置成功 |

#### 1.26 查询低电量模式

函数原型：+ (void)requestLowBatteryModel:(nullable void(^)(BOOL))completion;

功能描述： 查询外设是否为低电量模式

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL isOn) | 外设是否为低电量模式 |

#### 1.27 设置低电量模式

函数原型：+ (void)syncLowBatteryModel:(BOOL)isLowBatteryModel completion:(nullable void(^)(BOOL))completion;

功能描述：设置低电量模式

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| isLowBatteryModel | BOOL | 是否为低电量模式; YES: 低电量模式，反之正常模式 |
| completion | void(^)(NSInteger success) | 低电量模式是否设置成功 |

#### 1.28 同步天气开关状态

函数原型：+ (void)syncWeatherEnable:(BOOL)isEnable completion:(nullable void(^)(BOOL))completion;

功能描述：同步天气开关状态

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| isEnable | BOOL | 是否打开天气开关; YES: 打开; 反之关闭 |
| completion | void(^)(BOOL success) | 天气开关是否同步成功 |

#### 1.29 同步天气数据

函数原型：+ (void)syncWeatherData:(NSData \*)data completion:(nullable void(^)(BOOL))completion;

功能描述： 同步天气数据

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL success) | 天气数据是否同步成功 |

### 2. DeviceSetting(MBGlassesSettingService)

#### 2.1 设置SimCard状态

函数原型：+ (void)syncSimCardState:(BOOL)available completion:(nullable void(^)(BOOL success))completion;

功能描述：设置SimCard是否可用

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| available | BOOL | SimCard是否可用，YES: 可用；NO: 不可用 |
| completion | void(^)(BOOL success) | SimCard状态是否设置成功 |

#### 2.2 查询快捷桌面开关状态

函数原型：+ (void)requestQuickDesktopSwitch:(nullable void(^)(BOOL))completion;

功能描述：设置SimCard是否可用

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL isOn) | 快捷桌面开关是否打开 |

#### 2.3 设置快捷桌面开关

函数原型：+ (void)syncQuickDesktopSwitch:(BOOL)isOn completion:(nullable void(^)(BOOL success))completion;

功能描述：设置SimCard是否可用

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| isOn | BOOL | 是否打开快捷桌面，YES: 打开；NO: 关闭 |
| completion | void(^)(BOOL success) | 快捷桌面开关是否设置成功 |

#### 2.4 眼镜远程控制

函数原型：+ (void)syncRemoteControlAction:(MBRemoteControlAction)action completion:(nullable void(^)(BOOL))completion;

功能描述：眼镜遥控器

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数** | **类型** | **描述** | | |
| action | MBRemoteControlAction | 远程控制命令 | | |
| completion | void(^)(BOOL success) | 远程控制命令是否设置成功 | | |
| MBRemoteControlAction | | | **数值（**UInt8**）** | **描述** | |
| MBRemoteControlActionTouchBarForward | | | 0x00 | 触控条前滑 | |
| MBRemoteControlActionTouchBarbackward | | | 0x01 | 触控条后滑 | |
| MBRemoteControlActionTouchBarSingleClick | | | 0x02 | 触控条单击 | |
| MBRemoteControlActionTouchBarLongPress | | | 0x03 | 触控条长按 | |
| MBRemoteControlActionHomeSingleClick | | | 0x04 | 多功能键单击 | |
| MBRemoteControlActionHomeDoubleClick | | | 0x05 | 多功能键双击 | |
| MBRemoteControlActionHomeLongPress | | | 0x06 | 多功能键长按 | |
| MBRemoteControlActionHomePress | | | 0x07 | 多功能键按下 | |
| MBRemoteControlActionHomeReleased | | | 0x08 | 多功能键释放 | |
| MBRemoteControlActionTouchBarPress | | | 0x09 | 触摸条按下 | |
| MBRemoteControlActionTouchBarReleased | | | 0x0A | 触摸条释放 | |
| MBRemoteControlActionTouchBarDoubleClick | | | 0x0B | 触控条双击 | |

#### 2.5 查询智能省电开关

函数原型：+ (void)requestSmartPowerSaving:(nullable void(^)(BOOL))completion;

功能描述： 查询智能省电开关

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL isOn) | 智能省电开关是否打开 |

#### 2.6 设置智能省电开关

函数原型：+ (void)syncSmartPowerSaving:(BOOL)isOn completion:(nullable void(^)(BOOL))completion;

功能描述：设置智能省电开关

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| isOn | BOOL | 智能省电开关状态 |
| completion | void(^)(BOOL success) | 智能省电开关是否设置成功 |

#### 2.7 查询智能佩戴检测开关

函数原型：+ (void)requestWearCheck:(nullable void(^)(BOOL))completion;

功能描述：查询智能佩戴检测开关

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL isOn) | 智能佩戴检测开关是否打开 |

#### 2.8 设置智能佩戴检测开关

函数原型：+ (void)syncWearCheck:(BOOL)isOn completion:(nullable void(^)(BOOL))completion;

功能描述： 同步App退出登录

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| isOn | BOOL | 智能佩戴检测开关状态 |
| completion | void(^)(BOOL success) | 智能佩戴检测开关是否设置成功 |

#### 2.9 查询佩戴亮屏开关

函数原型：+ (void)requestWearBrightness:(nullable void(^)(BOOL))completion;

功能描述： 查询佩戴亮屏开关

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL isOn) | 佩戴亮屏开关状态 |

#### 2.10 设置佩戴亮屏开关

函数原型：+ (void)syncWearBrightness:(BOOL)isOn completion:(nullable void(^)(BOOL))completion;

功能描述：设置佩戴亮屏开关

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| isOn | BOOL | 佩戴亮屏开关状态 |
| completion | void(^)(BOOL success) | 低电量模式是否设置成功 |

#### 2.11 未佩戴超过10分钟监听

函数原型：+ (void)addNoWear:(NSString \*)tag listener:(void(^)(void))listener;

功能描述：未佩戴超过10分钟监听

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| tag | NSString | 标签 |
| listener | void(^)(void) |  |

#### 2.12 同步QQ音乐授权状态

函数原型：+ (void)syncQQMusicAuthentication:(BOOL)isAuthenticated completion:(nullable void(^)(BOOL))completion;

功能描述： 同步QQ音乐授权状态

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| isAuthenticated | BOOL | qq音乐授权是否授权 |
| completion | void(^)(BOOL success) | qq音乐授权状态是否同步成功 |

### 3. Contacts

#### 3.1 获取通讯录设备端联系人个数

函数原型：+ (void)requestContactsCount:(nullable void(^)(NSInteger))completion;

功能描述：获取通讯录设备端联系人个数

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL success) | NSInterger  获取通讯录设备端联系人个数 |

#### 3.2 通讯录上传联系人

函数原型：+ (void)syncContacts:(NSArray \*)contacts completion:(nullable void(^)(BOOL))completion;

功能描述：通讯录上传联系人

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| contacts | NSArray | 联系人数组 |
| completion | void(^)(BOOL success) | 通讯录人员名单上传是否成功 |

#### 3.3 根据id号获取单个联系人详细信息

函数原型：+ (void)requestContactsAtIndex:(NSInteger)index completion:(nullable void(^)(NSArray<NSDictionary<NSString \*, id> \*> \*))completion;

功能描述：根据id号获取单个联系人详细信息

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| index | NSInteger | 根据id号获取单个联系人序号 |
| completion | (nullable void(^)(NSArray<NSDictionary<NSString \*, id> \*> \*))completion | 获取单个联系人的详细信息字典 |

#### 3.4 同步单个联系人姓名、电话

函数原型：+ (void)syncContactName:(NSString \*)name telephone:(NSString \*)telephone completion:(nullable void(^)(BOOL))completion;

功能描述：同步单个联系人姓名、电话

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| name | NSInteger | 姓名 |
| telephone | NSString | 电话 |
| completion | (nullable void(^)(BOOL))completion | 同步单个联系人信息是否成功 |

### 4. Map

#### 4.1 进入导航

函数原型：+ (void)syncEnterNavigation:(nullable void(^)(BOOL))completion;

功能描述：进入导航

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| completion | (nullable void(^)(BOOL))completion | 是否进入导航功能模块 |

#### 4.2 开始导航

函数原型：+ (void)syncStartNavigation:(nullable void(^)(BOOL))completion;

功能描述：开始导航

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| completion | (nullable void(^)(BOOL))completion | 是否开始导航 |

#### 4.3 结束导航

函数原型：+ (void)syncStopNavigation:(nullable void(^)(BOOL))completion;

功能描述：结束导航

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| completion | (nullable void(^)(BOOL))completio | 是否结束导航 |

#### 4.4 设置导航时屏幕常亮

函数原型：+ (void)syncNavigationKeepBrightState:(BOOL)isOn completion:(nullable void(^)(BOOL))completion;

功能描述：设置导航时屏幕常亮

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| completion | (nullable void(^)(BOOL))completio | 是否设置导航时屏幕常亮成功 |
| isOn | BOOL | 设置导航时屏幕常亮 0x00 关闭 0x01 开启 |

#### 4.5 导航下发页面枚举值

函数原型：+ (void)syncNavigationPageName:(MBMapNavigationPageName)pageName

completion:(nullable void(^)(BOOL))completion ;

功能描述：导航下发页面枚举值

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| completion | (nullable void(^)(BOOL))completio | 是否设置导航时屏幕常亮成功 |
| pageName | MBMapNavigationPageName | MBMapNavigationPageName类型  代表设备端展示页面  （1）语音搜索地址或选择常用地址MBMapNavigationPageNameSearchAddress  （2）选择导航方式：`步行`或`骑行MBMapNavigationPageNameSelectTraffic  （3）提醒用户设置常用地址`回家`和`公司`地址MBMapNavigationPageNameRemaindSetAddress  （4）从地址列表中选择地址MBMapNavigationPageNameAddresses  （5）回到App首页MBMapNavigationPageNameHomepage  （6）正在导航界面MBMapNavigationPageNameNavigation  （7） 用户未同意高德地图隐私协议MBMapNavigationPageNameUnAuthenticated  （8） 设备标定页面MBMapNavigationPageNameCalibration |

#### 4.6 导航下发页面

函数原型：+ (void)syncNavigationPageName:(MBMapNavigationPageName)pageName

completion:(nullable void(^)(BOOL))completion ;

功能描述：导航下发页面枚举值

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| completion | (nullable void(^)(BOOL))completio | 是否设置导航下发页面成功 |
| pageName | MBMapNavigationPageName | MBMapNavigationPageName类型  代表设备端展示页面  （1）语音搜索地址或选择常用地址MBMapNavigationPageNameSearchAddress  （2）选择导航方式：`步行`或`骑行MBMapNavigationPageNameSelectTraffic  （3）提醒用户设置常用地址`回家`和`公司`地址MBMapNavigationPageNameRemaindSetAddress  （4）从地址列表中选择地址MBMapNavigationPageNameAddresses  （5）回到App首页MBMapNavigationPageNameHomepage  （6）正在导航界面MBMapNavigationPageNameNavigation  （7） 用户未同意高德地图隐私协议MBMapNavigationPageNameUnAuthenticated  （8） 设备标定页面MBMapNavigationPageNameCalibration |

#### 4.7 同步导航数据

函数原型：+ (void)syncAmapBinaryPacket:(NSString \*)guideInfo

timeRemaing:(NSString \*)timeRemaing

flag:(UInt32)flag

highlighted:(NSString \*)highlighted

image:(UIImage \*)image

completion:(nullable void(^)(BOOL))completion;

功能描述：同步导航数据

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| guideInfo | NSString | 导航信息 （左转、右转、直行等） |
| timeRemaing | NSString | 航剩余时间 |
| flag | UInt32 | 导航类型 |
| highlighted | NSString | 导航高亮 |
| image | UIImage | 导航图标 |
| completion | (nullable void(^)(BOOL))completion | 是否同步导航数据成功 |

#### 4.8 获取地图导航标定结果

函数原型：+ (void)requestCalibStatus:(nullable void(^)(BOOL))completion;

功能描述： 获取地图导航标定结果

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| completion | (nullable void(^)(BOOL))completion | 是否获取地图导航标定结果成功 |

#### 4.9 添加开始导航监听

函数原型：+ (void)addStartNavigationListener:(void(^)(void))listener;

功能描述： 添加开始导航监听

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| listener | (void(^)(void))listener | 无返回值  监听导航开始状态 |

#### 4.10 添加地图导航标定结果监听

函数原型：+ (void)addCalibStatusListener:(void(^)(BOOL))listener;

功能描述： 添加地图导航标定结果监听

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| listener | (void(^)(void))listener | 监听设备导航标定状态 |

#### 4.10 监听导航页面确定选择

函数原型：+ (void)addFlipListener:(void(^)(NSString \* \_Nullable, NSInteger))listener;

功能描述： 监听导航页面确定选择

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| listener | (void(^)(NSString \* \_Nullable, NSInteger))listener | 监听返回 block 参数如下 |
| pageName | NSString | 当前设备显示页面 （4.6 MBMapNavigationPageName） |
| option | NSInteger | 操作类型  voiceAssistantOptionBLEData :  voice\_Guester\_None  voice\_Guester\_NaviHome\_Walk\_ReturnHome  voice\_Guester\_NaviCompany\_Ride\_Back  voice\_Guester\_NaviSelected(Int) |

### 5. Translator

#### 5.1 提词器-发送文本

函数原型：+ (void)sendTeleprompterText:(NSString \*)text

pageIndex:(NSInteger)pageIndex

totalSize:(NSInteger)totalSize

completion:(nullable void(^)(BOOL))completion;

功能描述： 同步文本至设备端

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| text | NSString | 文本内容（单页数据） |
| pageIndex | NSInteger | 当前第几页 |
| totalSize | NSInteger | 总共多少页 |
| completion | (nullable void(^)(BOOL))completion | 同步是否成功 |

#### 5.2 提词器-页面翻页监听

函数原型：+ (void)addTeleprompterFlip:(NSString \*)tag listener:(void(^)(NSInteger))listener;

功能描述： 同步文本至设备端

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| tag | NSString | 监听 tag 标志 |
| listener | (void(^)(NSInteger))listener | 返回 数据：uint8\_t：0---page up / 1---page-down |

#### 5.3 提词器-移除页面翻页监听

函数原型：+ (void)removeTeleprompterFlipListener:(NSString \*)tag;

功能描述： 移除页面翻页监听

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| tag | NSString | 监听 tag 标志 |

#### 5.4 提词器-获取提词器支持字体尺寸列表

函数原型：+ (void)requestTeleprompterFontList:(nullable void(^)(NSArray<NSString \*> \*))completion;

功能描述： 获取提词器支持字体尺寸列表

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| completion | (nullable void(^)(NSArray<NSString \*> \*))completion | struct meta\_font\_size... |

#### 5.5 提词器-获取提词器当前字体序号

函数原型：+ (void)requestTeleprompterFont:(nullable void(^)(NSInteger))completion;

功能描述： 获取提词器当前字体序号

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| completion | (nullable void(^)(NSInteger))completion; | NSInteger ：字号 |

#### 5.6 提词器-设置提词器当前字体序号

函数原型：+ (void)syncTeleprompterFont:(NSInteger)fontIndex completion:(nullable void(^)(BOOL))completion;

功能描述： 获取提词器当前字体序号

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| fontIndex | NSInteger | 字体字号 |
| completion | (nullable void(^)(BOOL))completion; | 设置字体是否成功 |

#### 5.7 提词器-获取提词器显示区域尺寸

函数原型：+ (void)requestTeleprompterArea:(nullable void(^)(CGSize))completion;

功能描述： 获取提词器显示区域尺寸

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| completion | (nullable void(^)(CGSize))completion; | 获取CGSize 类型屏幕尺寸 |

#### 5.8 提词器-监听开始演讲

函数原型：+ (void)addTeleprompterStartSpeech:(nullable NSString \*)tag listener:(void(^)(void))listener;

功能描述： 监听开始演讲

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| listener | (void(^)(void))listener | block 执行收到监听后的业务操作 |
| tag | NSString | 监听标志位 |

#### 5.9 提词器-提词器结束演讲

函数原型：+ (void)teleprompterStopSpeech:(nullable void(^)(BOOL))completion;

功能描述： 提词器结束演讲

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| completion | (nullable void(^)(BOOL))completion | 是否结束题词演讲 |

#### 5.10 提词器-提词器结束演讲监听

函数原型：+ (void)teleprompterStopSpeech:(NSString \*)tag listener:(void(^)(void))listener;

功能描述： 提词器结束演讲监听

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| completion | (nullable void(^)(BOOL))completion | 是否结束题词演讲 |

#### 5.11 提词器-获取演讲时间

函数原型：+ (void)requestTeleprompterSpeechTime:(nullable void(^)(NSInteger))completion;

功能描述： -获取演讲时间

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| completion | (nullable void(^)(NSInteger))completion | 获取演讲时间 |

#### 5.12 提词器-iOS开启设备HFP通道

函数原型：+ (void)requestOpenHFP:(nullable void(^)(BOOL))completion;

功能描述： -iOS开启设备HFP通道

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| completion | (nullable void(^)(BOOL))completion | iOS开启设备HFP通道 |

### 6. Teleprompter

#### 6.1 翻译-开始翻译

函数原型：+ (void)startTranslation:(NSInteger)role language:(NSInteger)language completion:(nullable void(^)(BOOL))completion;

功能描述： -开始翻译

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| role | NSInteger | 设置演讲角色：0-眼镜端 1-手机端 |
| language | NSInteger | 设置语言：0-中文 1-英语 |
| completion | (nullable void(^)(BOOL))completion | 开始翻译是否成功 |

#### 6.2 翻译-结束翻译

函数原型：+ (void)stopTranslation:(nullable void(^)(BOOL))completion;

功能描述： -结束翻译

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| completion | (nullable void(^)(BOOL))completion | 结束翻译是否成功 |

#### 6.3 翻译-指定讲话语言

函数原型：+ (void)syncTranslateLanguage:(NSInteger)language completion:(nullable void(^)(BOOL))completion;

功能描述： -指定讲话语言

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| language | NSInteger | 设置语言：0-中文 1-英语 |
| completion | (nullable void(^)(BOOL))completion | 设置讲话翻译语言成功 |

#### 6.4 翻译-指定讲话使能

函数原型：+ (void)syncTranslationSpeak:(NSInteger)role completion:(nullable void(^)(BOOL))completion;

功能描述： -指定讲话语言

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| role | NSInteger | 设置演讲角色：0-眼镜端 1-手机端 |
| completion | (nullable void(^)(BOOL))completion | 设置讲话翻译语言成功 |

#### 6.5 翻译-翻译讲话使能

函数原型：+ (void)syncTranslationSpeak:(NSInteger)role completion:(nullable void(^)(BOOL))completion;

功能描述： -翻译讲话使能

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| role | NSInteger | 设置演讲角色：0-眼镜端 1-手机端 |
| completion | (nullable void(^)(BOOL))completion | 设置讲话翻译语言使能 |

#### 6.6 翻译-文本同步

函数原型：+ (void)updateTranslateTextRole:(NSInteger)role sessionId:(NSInteger)sessionId session:(NSString \*)session completion:(nullable void(^)(BOOL))completion；

功能描述： -文本同步

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| role | NSInteger | 设置演讲角色：0-眼镜端 1-手机端 |
| sessionId | NSInteger | 翻译会话id |
| session | NSString | 翻译内容（翻译前，识别的文本内容） |
| completion | (nullable void(^)(BOOL))completion | 文本同步至设备端是否成功 |

#### 6.7 翻译-监听录音控制开关

函数原型：+ (void)addMicrophoneListener:(void(^)(BOOL))listener;

功能描述： -监听录音控制开关

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| listene | (void(^)(BOOL))listene | 监听录音控制开关状态 |

#### 6.8 翻译-移除录音控制开关监听

函数原型：+ (void)removeMicrophoneListener;

功能描述： -移除录音控制开关监听

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| 无 | 无 | 无 |

#### 6.9 翻译-监听语言实时翻译结束

函数原型：+ (void)addEndSpeechListener:(void(^)(BOOL))listener;

功能描述： -监听语言实时翻译结束

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| listener | (void(^)(BOOL))listener | 监听语言实时翻译结束 |

#### 6.10 翻译-移除语言实时翻译结束监听

函数原型：+ (void)removeEndSpeechListener;

功能描述： -移除语言实时翻译结束监听

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| 无 | 无 | 无 |

### 7. Schedule

#### 7.1 获取全部日程key

函数原型：+ (void)requestAllScheudles:(nullable void(^)(NSArray<NSString \*> \*))completion;

功能描述：获取全部日程key

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(NSArray<NSString \*> \*keys) | 日程key |

#### 7.2 获取指定日程信息

函数原型：+ (void)requestScheduleByKey:(UInt16)key completion:(nullable void(^)(NSDictionary<NSString \*, id> \*))completion;

功能描述： 获取所有通知的开关状态

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| key | UInt16 | 日程key |
| completion | void(^)(NSDictionary<NSString \*, id> \*dictionary) | 日程信息 |

#### 7.3 添加日程

函数原型：+ (void)addScheduleWithTitle:(NSString \*)title

startDate:(NSDate \*)startDate

endDate:(NSDate\*)endDate

remindTime:(NSInteger)remindTime

completion:(nullable void(^)(NSInteger))completion;

功能描述： 获取所有通知的开关状态

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| title | UInt16 | 日程标题 |
| startDate | NSDate\* | 日程开始时间 |
| endDate | NSDate\* | 日程结束时间 |
| remindTime | NSInteger | 日程提醒时间 |
| completion | void(^)(NSInteger key) | 日程key |

#### 7.4 修改日程

函数原型：+ (void)updateScheduleWithTitle:(NSString \*)title

startDate:(NSDate \*)startDate

endDate:(NSDate\*)endDate

remindTime:(NSInteger)remindTime

taskId:(UInt16)taskId

completion:(nullable void(^)(BOOL))completion;

功能描述： 获取所有通知的开关状态

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| title | UInt16 | 日程标题 |
| startDate | NSDate\* | 日程开始时间 |
| endDate | NSDate\* | 日程结束时间 |
| remindTime | NSInteger | 日程提醒时间 |
| key | UInt16 | 日程key |
| completion | void(^)(BOOL success) | 日程key是否修改成功 |

#### 7.5 删除日程

函数原型：+ (void)deleteScheudleByKey:(UInt16)key completion:(nullable void(^)(BOOL))completion;

功能描述：获取全部日程key

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| key | UInt16 | 日程key |
| completion | void(^)(BOOL success) | 日程key是否删除成功 |

### 8. OTA

### 9. ChatGPT

#### 9.1 添加开始录音监听

函数原型：+ (void)addListenChatGPTStartRecoding:(void(^)(BOOL))listener;

功能描述：添加开始录音监听

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL success) | 日程key是否删除成功 |

#### 9.2 添加结束录音监听

函数原型：+ (void)addListenChatGPTEndRecoding:(void(^)(BOOL))listener;

功能描述：添加结束录音监听

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL success) | 日程key是否删除成功 |

#### 9.3 开始录音

函数原型：+ (void)requestChatGPTStartRecoding:(nullable void(^)(BOOL))completion;

功能描述：添加开始录音监听

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL success) | 日程key是否删除成功 |

#### 9.4 结束录音

函数原型：+ (void)requestChatGPTEndRecoding:(nullable void(^)(BOOL))completion;

功能描述：添加结束录音监听

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL success) | 日程key是否删除成功 |

#### 9.5 开始录音

函数原型：+ (void)requestChatGPTStartDialogue:(nullable void(^)(BOOL))completion;

功能描述：添加开始录音监听

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL success) | 日程key是否删除成功 |

#### 9.6 结束录音

函数原型：+ (void)requestChatGPTEndDialogue:(nullable void(^)(BOOL))completion;

功能描述：添加结束录音监听

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL success) | 日程key是否删除成功 |

#### 9.7 开始GPT问答

函数原型：+ (void)requestChatGPTStartDialogue:(nullable void(^)(BOOL))completion;

功能描述：添加开始录音监听

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL success) | 日程key是否删除成功 |

#### 9.8 结束GPT问答

函数原型：+ (void)requestChatGPTEndDialogue:(nullable void(^)(BOOL))completion;

功能描述：添加结束录音监听

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(BOOL success) | 日程key是否删除成功 |

#### 9.9 移除开始录音监听

函数原型：+ (void)removeListenChatGPTStartRecoding;

功能描述：添加开始录音监听

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| -- | -- | -- |

#### 9.10 移除结束录音监听

函数原型：+ (void)removeListenChatGPTEndRecoding;

功能描述：添加结束录音监听

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| -- | -- | -- |

### 10.Notification

#### 10.1 设置App通知开关状态

函数原型：+ (void)syncAppSwitch:(MBAppBundleIdentifer)identifer isOn:(BOOL)isOn completion:(nullable void(^)(BOOL))completion;

功能描述：设置App通知开关状态

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| identifer | MBAppBundleIdentifer | app bundle identifier |
| completion | void(^)(BOOL success) | App通知开关状态是否设置成功 |

#### 10.2 获取所有通知的开关状态

函数原型：+ (void)requestAllSwitchs:(nullable void(^)(NSArray<MBAppNotification\*>\*))completion;

功能描述： 获取所有通知的开关状态

|  |  |  |
| --- | --- | --- |
| **参数** | **类型** | **描述** |
| completion | void(^)(NSArray<MBAppNotification\*>\*notifications) | 所有通知的开关 |

/// app bundle identifier  
@property (nonatomic, copy) MBAppBundleIdentifer bundleIdentifier;  
  
/// 第三方App的URL scheme  
@property (nonatomic, copy) NSString \*scheme;  
  
@property (nonatomic, copy) NSString \*appName;  
  
@property (nonatomic, copy) NSString \*imageName;  
  
@property (nonatomic, copy) NSString \*sortKey;  
  
/// 消息通知的开关是否打开  
@property (nonatomic, assign) BOOL isOn;  
  
/// 第三方App是否安装在手机上  
@property (nonatomic, assign) BOOL isInstalled;  
  
@property (nonatomic, assign) BOOL isEnabled;

### 11.Notification

#### 11.1 语音助手一麦克风开关-设置

函数原型：+ (void)syncMicrophoneSwitch:(BOOL)isOn

completion:(nullable void(^)(BOOL))completion;

功能描述：麦克风开关设置

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| isOn | BOOL | 关闭 0x00 开启 0x01 |
| completion | (nullable void(^)(BOOL))completion | 设置麦克风开关成功 |

#### 11.2 语音助手一麦克风开关-获取

函数原型：+ (void)requestMicrophoneSwitch:(nullable void(^)(BOOL))completion;

功能描述：麦克风开关-获取

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| completion | (nullable void(^)(BOOL))completion | 设置麦克风开关状态获取  0x00关闭  0x01开启 |

#### 11.3 语音助手一APP打开(关闭长按2S唤醒开关)-设置

函数原型：+ (void)syncVoiceWakeupSwitch:(BOOL)isOn

completion:(nullable void(^)(BOOL))completion;

功能描述：APP打开(关闭长按2S唤醒开关)-设置

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| isOn | BOOL | 0x00关闭  0x01开启 |
| completion | (nullable void(^)(BOOL))completion | 设置APP打开(关闭长按2S唤醒开关) 状态 |

#### 11.4 语音助手一APP打开(关闭长按2S唤醒开关)-获取

函数原型：+ (void)requestVoiceWakeupSwitch:(nullable void(^)(BOOL))completion;

功能描述：APP打开(关闭长按2S唤醒开关)-获取

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| completion | (nullable void(^)(BOOL))completion | 设置APP打开(关闭长按2S唤醒开关) 状态  0x00关闭  0x01唤醒 |

#### 11.5 语音助手一设置状态-唤醒

函数原型：+ (void)syncVoiceWakeupStatus:(BOOL)isOn

completion:(nullable void(^)(BOOL))completion;

功能描述：设置状态-唤醒

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| isOn | BOOL | 0:结束唤醒 1:唤醒 |
| completion | (nullable void(^)(BOOL))completion | 设置状态-唤醒  0x00关闭  0x01唤醒 |

#### 11.6 语音助手一回复唤醒状态

函数原型：+ (void)responseVoiceWakeupResultCompletion:(nullable void(^)(BOOL))completion;

功能描述：回复唤醒状态

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| completion | (nullable void(^)(BOOL))completion | 设置状态-唤醒  0x00关闭  0x01唤醒 |

#### 11.7 语音助手一监听设置状态（双向指令）

函数原型：+ (void)addVoiceAssistantStatusListener:(void(^)(BOOL))listener;

功能描述：回复唤醒状态

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| listener | (void(^)(BOOL))listener | 设置状态-唤醒  0x00关闭  0x01唤醒 |

#### 11.8 语音助手一监听是否在语音应用内的状态

函数原型+ (void)addVoiceStreamListener:(void(^)(BOOL))listener;

功能描述：监听是否在语音应用内的状

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| listener | (void(^)(BOOL))listener | 是否在设备端“语音唤醒”模块  0x00不在  0x01在 |

#### 11.9 语音助手一语音会话内容

函数原型+ (void)syncVoiceBinaryPacket:(UInt16)type

data:(NSData \*)data

completion:(nullable void(^)(BOOL))completion \_\_attribute\_\_((deprecated("Use syncVoiceBinaryPacket:isEnd:data:completion:")));

功能描述：语音会话内容

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| type | UInt16 | 参见VoiceAssistantType 枚举 |
| data | NSData | 文本数据utf8 转换成NSString |
| isEnd | BOOL | 本次会话是否结束 |
| completion | (nullable void(^)(BOOL))completion | 是否在设备端“语音唤醒”模块  0x00不在  0x01在 |

typedef NS\_ENUM(uint16\_t, VoiceAssistantType) {  
 VoiceAssistantTypeNone = 0,  
 VoiceAssistantTypeQuestion = 1,  
 VoiceAssistantTypeAnswer = 2,  
 VoiceAssistantTypePrompt = 3,  
 VoiceAssistantTypeOpen = 4,  
 VoiceAssistantTypeClose = 5,  
 VoiceAssistantTypeLcdOn = 6, //打开屏幕  
 VoiceAssistantTypeLcdOff = 7, //关闭屏幕  
 VoiceAssistantTypeGoBack = 16, //返回上一级  
 VoiceAssistantTypeExit = 17, //退出当前模块  
 VoiceAssistantTypeChoice = 18,  
 VoiceAssistantTypePlayTTS = 19,  
 /// QQ音乐控制  
 VoiceAssistantTypeMusicPlay = 20,  
 VoiceAssistantTypeMusicPause = 21,  
 VoiceAssistantTypeMusicNext = 22,  
 VoiceAssistantTypeMusicPrev = 23,  
 //查看或创建日程  
 VoiceAssistantTypeScheduleView = 26,  
 VoiceAssistantTypeScheduleCreate = 27,  
 //打电话  
 VoiceAssistantTypePhoneByName = 28,//按名称打电话  
 VoiceAssistantTypePhoneByNumber = 29, //按号码打电话  
 VoiceAssistantTypeWakeUp = 30, //唤醒语音的回调  
};

#### 11.10 语音助手一语音同步导航列表地址到设备端

函数原型+ (void)syncVoiceMapBinaryPacket:(NSArray<MBGlassesNavigationObject\*> \*)array

completion:(nullable void(^)(BOOL))completion;

功能描述：语音同步导航列表地址到设备端

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 描述 |
| array | NSArray<MBGlassesNavigationObject\*> | 参见MBGlassesNavigationObject 类型 |
| completion | (nullable void(^)(BOOL))completion | 同步数据成功 |

@interface MBGlassesNavigationObject : NSObject  
  
@property (nonatomic, copy) NSString \*poi;  
@property (nonatomic, copy) NSString \*address;  
@property (nonatomic, assign) double distance;  
  
@end