# iOS SDK API Documentation

## 1. Requirements

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

## 2. Usage

Add the following code to the class method to initialize the 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

### Start Scan

[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:

/// Device unique ID   
@property (nonatomic, copy, readonly ) NSString \* deviceId ;   
  
/// Device mac address@property (nonatomic, copy, nullable) NSString \* deviceMac ;   
  
/// Device connection status   
@property (nonatomic, assign) MBConnectState state;   
  
/// Peripheral that connects or sends or receives data@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;   
  
/// Connect to current device - (void)connect:(nullable void(^)(BOOL, NSError \* \_Nullable))completion;   
  
/// Scanned service/// - Parameter service: Scanned service- (void) didDiscoverCharacteristicsForService :( CBService \*)service;   
  
/// Enable successful notification ( cccd success)   
/// - Parameter characteristic: Enable successful characteristics- (void)didUpdateNotificationStateForCharacteristic:(CBCharacteristic \*)characteristic;   
  
/// Receive data/// - Parameter characteristic: Received data characteristics- (void) didReceivedForCharacteristic :( CBCharacteristic \*)characteristic;   
  
/// Send data to the device/// - Parameters:/// - frame: Device communication protocol frame, the default write data Type is ` CBCharacteristicWriteWithoutResponse `   
/// - type: Write data type ` CBCharacteristicWriteWithoutResponse ` or ` CBCharacteristicWriteWithResponse `   
/// - completion: callback - (void) sendFrame :(id< MBFrameProtocol >)frame completion:(nullable MBTaskCompletion )completion;   
  
/// Disconnect/// - Parameter shouldUnbind : Whether to send the unbinding command   
- (void)disconnect:(BOOL) shouldUnbind ;   
  
@optional   
  
/// Manufacturer field of BLE broadcast information @property (nonatomic, copy, nullable) NSData \* manufacturerData ;   
  
@property (nonatomic, copy, nullable) NSDictionary \* connectOpions ;   
  
/// Successfully open the L2CAP channel/// - Parameter channel:L2CAP channel - (void)didOpenL2CAPChannel:(CBL2CAPChannel \*)channel;   
  
/// Register the callback of the request command received from the peripheral   
- (void) registerCommand :(UInt16)command tag:(nullable NSString \*)tag listener:( MBDidReceivedForCharacteristic )listener;   
/// Remove the request command callback of the peripheral - (void) removeListenForCommand :(UInt16)command tag:(nullable NSString \*)tag;   
/// Cancel the sent data frame. If frames have been sent, it will be invalid - (void) cancelFrames :( NSArray <id< MBFrameProtocol >>\*)frames;

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

### Bind peripheral

Use http://api.cloud.meta-bounds.com/v1/terminals/bind to upload the binding relationship between the user and the peripheral. For details, please refer to: [Cloud Service API Interface Documentation]

### Unbind peripheral

Use http://api.cloud.meta-bounds.com/v1/terminals/unbind to unbind the peripherals. The method to be called at the same time has the parameter value

## Send/Receive Data

### 1. DeviceInfo ( MB GlassesInfoService )

#### 1.1 Obtaining device information

Function prototype: + requestDeviceInfo ::(nullable void(^)(BOOL isEnabledDND , BOOL isEnabledWeather ))completion

Function description: Get device information

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL isEnabledDND , BOOL isEnabledWeather ) | Whether Do Not Disturb mode is on.  Is the weather reminder switch turned on? |

#### 1.2 Setting the peripheral language

Function prototype: + (void) syncAppLanguage :(UInt8)language completion:(nullable void(^)(BOOL))completion

Function description: Set the peripheral language

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| language | UInt8 | Language type; 0x00: Chinese; 0x01: English |
| completion | void(^)(BOOL success) | Is the language setting successful? |

#### 1.3 Setting the time

Function prototype: + (void) syncSystemTime :(nullable void(^)(BOOL))completion

Function description: Synchronize mobile phone time to glasses

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL success) | the mobile phone time synchronized successfully? |

#### 1.4 Set the phone model

Function prototype: + (void) syncPhoneMode :(nullable void(^)(BOOL))completion

Function description: Set the phone model

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL success) | Whether the mobile phone model is synchronized successfully |

#### 1.5 Set the Do Not Disturb mode switch

Function prototype: + (void) syncDNDMode :(BOOL) isQuiet completion:(nullable void(^)(BOOL))completion

Function description: Set the peripheral language

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| isQuiet | BOOL | isQuiet turns on Do Not Disturb mode if it is YES, otherwise it turns off |
| completion | void(^)(BOOL success) | Is the Do Not Disturb mode switch set successfully? |

#### 1.6 Set the Bluetooth name of the peripheral

Function prototype: + (void) syncBluetoothName :( NSString \*)name completion:(nullable void(^)(BOOL))completion

Function description: Set the peripheral language

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| name | NSString | Bluetooth peripheral name |
| completion | void(^)(BOOL success) | Is the Bluetooth name set successfully? |

#### 1.7 Query screen brightness

Function prototype: + (void) requestScreenBrightness :(nullable void(^)( NSInteger ))completion

Function description: Query screen brightness

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)( NSInteger value) | Glasses screen brightness; if acquisition fails, it will be -1 |

#### 1.8 Setting the screen brightness

Function prototype: + (void) syncScreenBrightness :( NSInteger ) value completion:(nullable void(^)(BOOL)) completion

Function description: Set screen brightness

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| value | NSInteger | Screen brightness |
| completion | void(^)(BOOL success) | Is the screen brightness set successfully? |

#### 1.9 Query the automatic screen off time

Function prototype: + (void) requestScreenOffTime :(nullable void(^)( NSInteger ))completion

Function description: Query the automatic screen off time

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)( NSInteger value) | The automatic screen-off time of the glasses; if the acquisition fails, it will be -1 |

#### 1.10 Set the automatic screen off time

Function prototype: + (void) syncScreenBrightness :( NSInteger ) value completion:(nullable void(^)(BOOL)) completion

Function description: Set the automatic screen off time

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| value | NSInteger | Automatic screen off time |
| completion | void(^)(BOOL success) | Is the automatic screen off time set successfully? |

#### 1.11 Query the automatic screen brightness adjustment mode

Function prototype: + (void) requestScreenBrightnessMode :(nullable void(^)(BOOL))completion

Function description: Query the automatic adjustment mode of screen brightness

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL isOn ) | Whether the automatic screen brightness adjustment mode is turned on. YES means it is turned on. |

#### 1.12 Set the screen brightness automatic adjustment mode

Function prototype: + (void) syncScreenBrightnessMode :(BOOL) isAuto completion:(nullable void(^)(BOOL))completion

Function description: Set the screen brightness to automatically adjust mode

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| isAuto | BOOL | Screen brightness automatic adjustment mode, YES: turn on, otherwise turn off |
| completion | void(^)(BOOL success) | Is the automatic screen brightness adjustment mode set successfully? |

#### 1.13 Query screen lighting mode

Function prototype: + (void) requestScreenLighten :(nullable void(^)( NSInteger ))completion

Function description: Query the automatic adjustment mode of screen brightness

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)( NSInteger value) | 0x00: Off; 0x01: Left eye monocular mode; 0x02: Right eye monocular mode |

#### 1.14 Set the screen lighting mode

Function prototype: + (void) syncScreenLighten :( NSInteger ) value completion:(nullable void(^)(BOOL)) completion

Function description: Set the screen brightness to automatically adjust mode

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| value | NSInteger | 0x00: Off; 0x01: Left eye monocular mode; 0x02: Right eye monocular mode |
| completion | void(^)(BOOL success) | Is the automatic screen brightness adjustment mode set successfully? |

#### 1.15 Check BT connection status

Function prototype: + (void) requestBTState :(nullable void(^)(BOOL))completion

Function description: Query the automatic adjustment mode of screen brightness

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL isConnected ) | Is BT connected? |

#### 1.16 Disconnect BT connection

Function prototype: + (void) syncDisconnectBT :(nullable void(^)(BOOL))completion

Function description: Disconnect BT connection

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL success) | Disconnect BT connection and check whether it is set successfully. |

#### 1.17 Reconnect BT

Function prototype: + (void) requestReconnectBT :(nullable void(^)(BOOL))completion

Function description: Reconnect BT

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL success) | Is BT reconnected successfully? |

#### 1.18 Added monitoring of BT connection status

Function prototype: + (void) addBTStateListener :(void(^)(BOOL))listener;

Function description: Set the screen brightness to automatically adjust mode

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL isConnected ) | Is BT connected? |

#### 1.19 Remove BT connection status monitoring

Function prototype: + (void) removeBTStateListener

Function description: Remove the monitoring of BT connection status

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| -- | -- | -- |

#### 1.20 Restore factory settings

Function prototype: + (void) requestFactoryReset :(nullable void(^)(BOOL))completion

Function description: Restore factory settings

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL success) | Is the factory reset successful? |

#### 1.21 Restore factory settings - restart

Function prototype: + (void) requestFactoryReboot :(nullable void(^)(BOOL))completion;

Function description: Restore factory settings - Restart

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL success) | Restore factory settings-restart to see if the settings are successful |

#### 1.22 Sync App Logout

Function prototype: + (void) syncLogout :(nullable void(^)(BOOL))completion;

Function description: Synchronize App logout

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL success) | Synchronize App logout to see if it is set successfully |

#### 1.23 Query the power level of peripherals

Function prototype: + (void) requestBatteryInfo :(nullable void(^)( NSInteger , BOOL))completion

Function description: Query the power of peripherals

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)( NSInteger value, BOOL isCharging ) | Current power of peripherals  Is the peripheral charging? |

#### 1.24 Restore factory settings - restart

Function prototype: + (void) addBatteryListener :(void(^)( NSInteger , BOOL))listener;

Function description: Restore factory settings - Restart

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL success) | Restore factory settings-restart to see if the settings are successful |

#### 1.25 Synchronize App Logout

Function prototype: + (void) removeBatteryListener ;

Function description: Synchronize App logout

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL success) | Synchronize App logout to see if it is set successfully |

#### 1.26 Query low power mode

Function prototype: + (void) requestLowBatteryModel :(nullable void(^)(BOOL))completion;

Function description: Check whether the peripheral is in low power mode

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL isOn ) | Whether the peripheral is in low power mode |

#### 1.27 Set Low Power Mode

Function prototype: + (void) syncLowBatteryModel :(BOOL) isLowBatteryModel completion:(nullable void(^)(BOOL))completion;

Function description: Set low power mode

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| isLowBatteryModel | BOOL | Is it low power mode? YES: low power mode, otherwise normal mode |
| completion | void(^)( NSInteger success) | Is the low power mode set successfully? |

#### 1.28 Synchronize weather switch status

Function prototype: + (void) syncWeatherEnable :(BOOL) isEnable completion:(nullable void(^)(BOOL))completion;

Function description: Synchronize weather switch status

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| isEnable | BOOL | Whether to turn on the weather switch; YES: turn on; otherwise turn off |
| completion | void(^)(BOOL success) | Whether the weather switch is synchronized successfully |

#### 1.29 Synchronize weather data

Function prototype: + (void) syncWeatherData :( NSData \*)data completion:(nullable void(^)(BOOL))completion;

Function description: Synchronize weather data

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL success) | Whether the weather data is synchronized successfully |

### 2. DeviceSetting ( MB GlassesSettingService )

#### 2.1 Set SimCard status

Function prototype: + (void) syncSimCardState :(BOOL)available completion:(nullable void(^)(BOOL success))completion;

Function description: Set whether SimCard is available

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| available | BOOL | SimCard available ? YES: available; NO: not available |
| completion | void(^)(BOOL success) | Is the SimCard status set successfully? |

#### 2.2 Query the status of the shortcut desktop switch

Function prototype: + (void) requestQuickDesktopSwitch :(nullable void(^)(BOOL))completion;

Function description: Set whether SimCard is available

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL isOn ) | Is the shortcut desktop switch turned on? |

#### 2.3 Set the shortcut desktop switch

Function prototype: + (void) syncQuickDesktopSwitch :(BOOL) isOn completion:(nullable void(^)(BOOL success))completion;

Function description: Set whether SimCard is available

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| isOn | BOOL | Whether to open the shortcut desktop, YES: open; NO: close |
| completion | void(^)(BOOL success) | Is the quick desktop switch set successfully? |

#### 2.4 Remote control of glasses

Function prototype: + (void) syncRemoteControlAction :( MBRemoteControlAction )action completion:(nullable void(^)(BOOL))completion;

Function description: glasses remote control

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **parameter** | **type** | **describe** | | |
| action | MBRemoteControlAction | Remote control commands | | |
| completion | void(^)(BOOL success) | Whether the remote control command is set successfully | | |
| MBRemoteControlAction | | | **Numeric value (** UInt8 **)** | **describe** | |
| MBRemoteControlActionTouchBarForward | | | 0x00 | Touch bar slide forward | |
| MBRemoteControlActionTouchBarbackward | | | 0x01 | Touch bar slide back | |
| MBRemoteControlActionTouchBarSingleClick | | | 0x02 | Touch Bar Click | |
| MBRemoteControlActionTouchBarLongPress | | | 0x03 | Touch bar long press | |
| MBRemoteControlActionHomeSingleClick | | | 0x04 | Multi-function key click | |
| MBRemoteControlActionHomeDoubleClick | | | 0x05 | Double-click the multi-function button | |
| MBRemoteControlActionHomeLongPress | | | 0x06 | Long press the multi-function button | |
| MBRemoteControlActionHomePress | | | 0x07 | Multi-function button pressed | |
| MBRemoteControlActionHomeReleased | | | 0x08 | Multi-function button release | |
| MBRemoteControlActionTouchBarPress | | | 0x09 | Touch bar pressed | |
| MBRemoteControlActionTouchBarReleased | | | 0x0A | Touch Bar Release | |
| MBRemoteControlActionTouchBarDoubleClick | | | 0x0B | Touch Bar Double Click | |

#### 2.5 Query the intelligent power saving switch

Function prototype: + (void) requestSmartPowerSaving :(nullable void(^)(BOOL))completion;

Function description: Query the intelligent power saving switch

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL isOn ) | Is the smart power saving switch turned on? |

#### 2.6 Set up smart power saving switch

Function prototype: + (void) syncSmartPowerSaving :(BOOL) isOn completion:(nullable void(^)(BOOL))completion;

Function description: Set the intelligent power saving switch

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| isOn | BOOL | Smart power saving switch status |
| completion | void(^)(BOOL success) | Is the intelligent power saving switch set successfully? |

#### 2.7 Query the smart wear detection switch

Function prototype: + (void) requestWearCheck :(nullable void(^)(BOOL))completion;

Function description: Query the smart wear detection switch

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL isOn ) | Is the smart wear detection switch turned on? |

#### 2.8 Setting the smart wear detection switch

Function prototype: + (void) syncWearCheck :(BOOL) isOn completion:(nullable void(^)(BOOL))completion;

Function description: Synchronize App logout

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| isOn | BOOL | Smart wear detection switch status |
| completion | void(^)(BOOL success) | Is the smart wear detection switch set successfully? |

#### 2.9 Check the screen light switch when wearing

Function prototype: + (void) requestWearBrightness :(nullable void(^)(BOOL))completion;

Function description: Check the screen light switch when wearing

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL isOn ) | Wearing screen switch status |

#### 2.10 Set the screen-on switch when wearing

Function prototype: + (void) syncWearBrightness :(BOOL) isOn completion:(nullable void(^)(BOOL))completion;

Function description: Set the screen-on switch when wearing

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| isOn | BOOL | Wearing screen switch status |
| completion | void(^)(BOOL success) | Is the low power mode set successfully? |

#### 2.11 Not wearing the monitor for more than 10 minutes

Function prototype: + (void) addNoWear :( NSString \*)tag listener:(void(^)(void))listener;

Function description: Monitoring when not worn for more than 10 minutes

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| tag | NSString | Label |
| listener | void(^)(void) |  |

#### 2.12 Synchronize QQ Music authorization status

Function prototype: + (void) syncQQMusicAuthentication :(BOOL) isAuthenticated completion:(nullable void(^)(BOOL))completion;

Function description: Synchronize QQ music authorization status

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| isAuthenticated | BOOL | QQ music authorized? |
| completion | void(^)(BOOL success) | Is the QQ music authorization status synchronized successfully? |

### 3. Contacts

#### 3.1 Get the number of contacts in the address book device

Function prototype: + (void) requestContactsCount :(nullable void(^)( NSInteger ))completion;

Function description: Get the number of contacts in the address book device

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL success) | NSInterger  Get the number of contacts in the address book device |

#### 3.2 Upload contacts from address book

Function prototype: + (void) syncContacts :( NSArray \*)contacts completion:(nullable void(^)(BOOL))completion;

Function description: Upload contacts from address book

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| Contacts | NSArray | Contacts array |
| completion | void(^)(BOOL success) | Whether the address book list is uploaded successfully |

#### 3.3 Get detailed information of a single contact by ID number

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

Function description: Get detailed information of a single contact based on the ID number

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| index | NSInteger | Get a single contact number based on the id number |
| completion | (nullable void(^)( NSArray < NSDictionary < NSString \*, id> \*> \*))completion | Get the detailed information dictionary of a single contact |

#### 3.4 Synchronize a single contact's name and phone number

Function prototype: + (void) syncContactName :( NSString \*)name telephone:( NSString \*)telephone completion:(nullable void(^)(BOOL))completion;

Function description: Synchronize a single contact name and phone number

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| name | NSInteger | Name |
| telephone | NSString | Telephone |
| completion | (nullable void(^)(BOOL))completion | Whether the synchronization of a single contact information is successful |

### 4. Map

#### 4.1 Enter the navigation

Function prototype: + (void) syncEnterNavigation :(nullable void(^)(BOOL))completion;

Function description: Enter navigation

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| completion | (nullable void(^)(BOOL))completion | Whether to enter the navigation function module |

#### 4.2 Start Navigation

Function prototype: + (void) syncStartNavigation :(nullable void(^)(BOOL))completion;

Function description: Start navigation

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| completion | (nullable void(^)(BOOL))completion | Start navigation? |

#### 4.3 End navigation

Function prototype: + (void) syncStopNavigation :(nullable void(^)(BOOL))completion;

Function description: End navigation

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| completion | (nullable void(^)(BOOL)) completio | End navigation? |

#### 4.4 Set the screen to stay on during navigation

Function prototype: + (void) syncNavigationKeepBrightState :(BOOL) isOn completion:(nullable void(^)(BOOL))completion;

Function description: Set the screen to always be on during navigation

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| completion | (nullable void(^)(BOOL)) completio | Whether to set the screen to stay on during navigation successfully |
| isOn | BOOL | Set the screen to stay on during navigation 0x00 Off 0x01 On |

#### 4.5 Navigation page enumeration value

Function prototype: + (void) syncNavigationPageName :( MBMapNavigationPageName ) pageName

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

Function description: Navigation page enumeration value

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| completion | (nullable void(^)(BOOL)) completio | Whether to set the screen to stay on during navigation successfully |
| pageName | MBMapNavigationPageName | MBMapNavigationPageName Type  Represents the device display page  (1) Search the address by voice or select a common address MBMapNavigationPageNameSearchAddress  (2) Select navigation mode: Walking or CyclingMBMapNavigationPageNameSelectTraffic  (3) Remind users to set frequently used addresses "home" and "company" addresses MBMapNavigationPageNameRemaindSetAddress  (4) Select the address from the address list MBMapNavigationPageNameAddresses  (5) Return to the App homepage MBMapNavigationPageNameHomepage  (6) Navigating the interface MBMapNavigationPageNameNavigation  (7) The user has not agreed to the Amap privacy agreement MBMapNavigationPageNameUnAuthenticated  (8) Device calibration page MBMapNavigationPageNameCalibration |

#### 4.6 Navigation delivery page

Function prototype: + (void) syncNavigationPageName :( MBMapNavigationPageName ) pageName

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

Function description: Navigation page enumeration value

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| completion | (nullable void(^)(BOOL)) completio | Whether to set the navigation page to be sent successfully |
| pageName | MBMapNavigationPageName | MBMapNavigationPageName Type  Represents the device display page  (1) Search the address by voice or select a common address MBMapNavigationPageNameSearchAddress  (2) Select navigation mode: Walking or CyclingMBMapNavigationPageNameSelectTraffic  (3) Remind users to set frequently used addresses "home" and "company" addresses MBMapNavigationPageNameRemaindSetAddress  (4) Select an address from the address list MBMapNavigationPageNameAddresses  (5) Return to the App homepage MBMapNavigationPageNameHomepage  (6) Navigating the interface MBMapNavigationPageNameNavigation  (7) The user has not agreed to the Amap privacy agreement MBMapNavigationPageNameUnAuthenticated  (8) Device calibration page MBMapNavigationPageNameCalibration |

#### 4.7 Synchronizing Navigation Data

Function prototype: + (void) syncAmapBinaryPacket :( NSString \*) guideInfo

timeRemaing :( NSString \*) timeRemaing

flag:(UInt32)flag

highlighted:( NSString \*)highlighted

image:( UIImage \*)image

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

Function description: Synchronize navigation data

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| guideInfo | NSString | Navigation information (turn left, turn right , go straight, etc. ) |
| timeRemaing | NSString | Remaining flight time |
| flag | UInt32 | Navigation Type |
| highlighted | NSString | Navigation Highlight |
| image | UIImage | Navigation Icons |
| completion | (nullable void(^)(BOOL))completion | Whether the navigation data synchronization is successful |

#### 4.8 Obtaining map navigation calibration results

Function prototype: + (void) requestCalibStatus :(nullable void(^)(BOOL))completion;

Function description: Get map navigation calibration results

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| completion | (nullable void(^)(BOOL))completion | Whether the map navigation calibration result is obtained successfully |

#### 4.9 Add start navigation listener

Function prototype: + (void) addStartNavigationListener :(void(^)(void))listener;

Function description: Add start navigation listener

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| listener | (void(^)(void))listener | No return value  Listen for navigation start status |

#### 4.10 Add map navigation calibration result monitoring

Function prototype: + (void) addCalibStatusListener :(void(^)(BOOL))listener;

Function description: Add map navigation calibration result monitoring

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| listener | (void(^)(void))listener | Monitor device navigation calibration status |

#### 4.10 Listen to the navigation page to confirm the selection

Function prototype: + (void) addFlipListener :(void(^)( NSString \* \_Nullable, NSInteger ))listener;

Function description: Listen to the navigation page to confirm the selection

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| listener | (void(^)( NSString \* \_Nullable, NSInteger ))listener | The listener returns the following block parameters: |
| pageName | NSString | The current device display page (4.6 MBMapNavigationPageName ) |
| option | NSInteger | Operation Type  voiceAssistantOptionBLEData :  voice\_Guester\_None  voice\_Guester\_NaviHome\_Walk\_ReturnHome  voice\_Guester\_NaviCompany\_Ride\_Back  voice\_Guester\_NaviSelected (Int) |

### 5. Translator

#### 5.1 Teleprompter-Send Text

Function prototype: + (void) sendTeleprompterText :( NSString \*)text

pageIndex :( NSInteger ) pageIndex

totalSize :( NSInteger ) totalSize

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

Function description: Synchronize text to device

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| text | NSString | Text content (single page data) |
| pageIndex | NSInteger | Current page |
| totalSize | NSInteger | How many pages in total |
| completion | (nullable void(^)(BOOL))completion | Whether the synchronization is successful |

#### 5.2 Teleprompter-Page Turning Monitoring

Function prototype: + (void) addTeleprompterFlip :( NSString \*)tag listener:(void(^)( NSInteger ))listener;

Function description: Synchronize text to device

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| tag | NSString | Listening tag flag |
| listener | (void(^)( NSInteger ))listener | Return data: uint8\_t: 0---page up / 1---page down |

#### 5.3 Teleprompter-Remove page turning monitoring

Function prototype: + (void) removeTeleprompterFlipListener :( NSString \*)tag;

Function description: Remove page turning monitoring

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| tag | NSString | Listening tag flag |

#### 5.4 Teleprompter - Get the list of font sizes supported by the teleprompter

Function prototype: + (void) requestTeleprompterFontList :(nullable void(^)( NSArray < NSString \*> \*))completion;

Function description: Get the list of font sizes supported by the teleprompter

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| completion | (nullable void(^)( NSArray < NSString \*> \*))completion | struct meta\_font\_size ... |

#### 5.5 Teleprompter-Get the current font number of the teleprompter

Function prototype: + (void) requestTeleprompterFont :(nullable void(^)( NSInteger ))completion;

Function description: Get the current font number of the teleprompter

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| completion | (nullable void(^)( NSInteger ))completion; | NSInteger : font size |

#### 5.6 Teleprompter-Set the current font number of the teleprompter

Function prototype: + (void) syncTeleprompterFont :( NSInteger ) fontIndex completion:(nullable void(^)(BOOL))completion;

Function description: Get the current font number of the teleprompter

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| fontIndex | NSInteger | Font Size |
| completion | (nullable void(^)(BOOL))completion; | Set font successfully |

#### 5.7 Teleprompter - Get the display area size of the teleprompter

Function prototype: + (void) requestTeleprompterArea :(nullable void(^)( CGSize ))completion;

Function description: Get the display area size of the teleprompter

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| completion | (nullable void(^)( CGSize ))completion; | Get the screen size of CGSize type |

#### 5.8 Teleprompter-Monitoring Start Speech

Function prototype: + (void) addTeleprompterStartSpeech :(nullable NSString \*)tag listener:(void(^)(void))listener;

Function description: Listen to the start of the speech

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| listener | (void(^)(void))listener | block executes the business operation after receiving the monitoring |
| tag | NSString | Monitor flag |

#### 5.9 Teleprompter-Teleprompter ends the speech

Function prototype: + (void) teleprompterStopSpeech :(nullable void(^)(BOOL))completion;

Functional description: The teleprompter ends the speech

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| completion | (nullable void(^)(BOOL))completion | End of the inscription speech |

#### 5.10 Teleprompter-Teleprompter End of Speech Monitoring

Function prototype: + (void) teleprompterStopSpeech :( NSString \*)tag listener:(void(^)(void))listener;

Function description: The teleprompter ends the speech monitoring

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| completion | (nullable void(^)(BOOL))completion | End of the inscription speech |

#### 5.11 Teleprompter-Get speech time

Function prototype: + (void) requestTeleprompterSpeechTime :(nullable void(^)( NSInteger ))completion;

Function description: -Get speech time

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| completion | (nullable void(^)( NSInteger ))completion | Get the lecture time |

#### 5.12 Teleprompter-iOS opens the device HFP channel

Function prototype: + (void) requestOpenHFP :(nullable void(^)(BOOL))completion;

Function description: -iOS opens the device HFP channel

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| completion | (nullable void(^)(BOOL))completion | iOS opens the device HFP channel |

### 6. Teleprompter

#### 6.1 Translation - Start translating

Function prototype: + (void) startTranslation :( NSInteger )role language:( NSInteger )language completion:(nullable void(^)(BOOL))completion;

Function description: - Start translating

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| role | NSInteger | Set the speaking role: 0-glasses 1-mobile phone |
| language | NSInteger | Set language: 0-Chinese 1-English |
| completion | (nullable void(^)(BOOL))completion | Start translation successfully |

#### 6.2 Translate-End Translation

Function prototype: + (void) stopTranslation :(nullable void(^)(BOOL))completion;

Function description: -End translation

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| completion | (nullable void(^)(BOOL))completion | Whether the translation is completed successfully |

#### 6.3 Translation - Specifying the Speech Language

Function prototype: + (void) syncTranslateLanguage :( NSInteger )language completion:(nullable void(^)(BOOL))completion;

Function description: -Specify the speaking language

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| language | NSInteger | Set language: 0-Chinese 1-English |
| completion | (nullable void(^)(BOOL))completion | Set speech translation language successfully |

#### 6.4 Translation-Specified Speech Enable

Function prototype: + (void) syncTranslationSpeak :( NSInteger )role completion:(nullable void(^)(BOOL))completion;

Function description: -Specify the speaking language

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| role | NSInteger | Set the speaking role: 0-glasses 1-mobile phone |
| completion | (nullable void(^)(BOOL))completion | Set speech translation language successfully |

#### 6.5 Translation - Translation speech enablement

Function prototype: + (void) syncTranslationSpeak :( NSInteger )role completion:(nullable void(^)(BOOL))completion;

Function description: -Translation speech enable

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| role | NSInteger | Set the speaking role: 0-glasses 1-mobile phone |
| completion | (nullable void(^)(BOOL))completion | Set speech translation language enable |

#### 6.6 Translation-Text Synchronization

Function prototype: + (void) updateTranslateTextRole :( NSInteger )role sessionId :( NSInteger ) sessionId session:( NSString \*)session completion:(nullable void(^)(BOOL))completion;

Function description: -Text synchronization

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| role | NSInteger | Set the speaking role: 0-glasses 1-mobile phone |
| sessionId | NSInteger | Translation session id |
| session | NSString | Translation content (identified text content before translation) |
| completion | (nullable void(^)(BOOL))completion | Whether the text is synchronized to the device successfully |

#### 6.7 Translation-Monitoring Recording Control Switch

Function prototype: + (void) addMicrophoneListener :(void(^)(BOOL))listener;

Function description: -Monitor recording control switch

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| listene | (void(^)(BOOL)) listen | Monitor recording control switch status |

#### 6.8 Translation-Remove recording control switch monitoring

Function prototype: + (void) removeMicrophoneListener ;

Function description: -Remove the recording control switch monitoring

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| none | none | none |

#### 6.9 Translation-Monitoring language real-time translation ends

Function prototype: + (void) addEndSpeechListener :(void(^)(BOOL))listener;

Function description: - Listen to the language and translate it in real time

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| listener | (void(^)(BOOL))listener | Monitoring language real-time translation ends |

#### 6.10 Translation-Remove language real-time translation end monitoring

Function prototype: + (void) removeEndSpeechListener ;

Function description: -Remove the language real-time translation end monitoring

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| none | none | none |

### 7. Schedule

#### 7.1 Get all schedule keys

Function prototype: + (void) requestAllScheudles :(nullable void(^)( NSArray < NSString \*> \*))completion;

Function description: Get all schedule keys

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)( NSArray < NSString \*> \*keys) | Schedule key |

#### 7.2 Get the specified schedule information

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

Function description: Get the switch status of all notifications

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| key | UInt16 | Schedule key |
| completion | void(^)( NSDictionary < NSString \*, id> \*dictionary) | Schedule Information |

#### 7.3 Add Schedule

Function prototype: + (void) addScheduleWithTitle :( NSString \*)title

startDate :( NSDate \*) startDate

endDate : ( NSDate \*) endDate

remindTime :( NSInteger ) remindTime

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

Function description: Get the switch status of all notifications

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| title | UInt16 | Agenda Title |
| startDate | NSDate \* | Schedule start time |
| endDate | NSDate \* | Schedule end time |
| remindTime | NSInteger | Schedule reminder time |
| completion | void(^)( NSInteger key) | Schedule key |

#### 7.4 Modify schedule

Function prototype: + (void) updateScheduleWithTitle :( NSString \*)title

startDate :( NSDate \*) startDate

endDate : ( NSDate \*) endDate

remindTime :( NSInteger ) remindTime

taskId : (UInt16) taskId

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

Function description: Get the switch status of all notifications

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| title | UInt16 | Agenda Title |
| startDate | NSDate \* | Schedule start time |
| endDate | NSDate \* | Schedule end time |
| remindTime | NSInteger | Schedule reminder time |
| key | UInt16 | Schedule key |
| completion | void(^)(BOOL success) | Whether the schedule key is modified successfully |

#### 7.5 Deleting a Schedule

Function prototype: + (void) deleteScheudleByKey :(UInt16)key completion:(nullable void(^)(BOOL))completion;

Function description: Get all schedule keys

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| key | UInt16 | Schedule key |
| completion | void(^)(BOOL success) | Whether the schedule key is deleted successfully |

### 8. OTA

### 9. ChatGPT

#### 9.1 Add start recording monitoring

Function prototype: + (void) addListenChatGPTStartRecoding :(void(^)(BOOL))listener;

Function description: Add start recording monitoring

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL success) | Whether the schedule key is deleted successfully |

#### 9.2 Add end recording monitoring

Function prototype: + (void) addListenChatGPTEndRecoding :(void(^)(BOOL))listener;

Function description: Add end recording monitoring

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL success) | Whether the schedule key is deleted successfully |

#### 9.3 Start recording

Function prototype: + (void) requestChatGPTStartRecoding :(nullable void(^)(BOOL))completion;

Function description: Add start recording monitoring

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL success) | Whether the schedule key is deleted successfully |

#### 9.4 End Recording

Function prototype: + (void) requestChatGPTEndRecoding :(nullable void(^)(BOOL))completion;

Function description: Add end recording monitoring

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL success) | Whether the schedule key is deleted successfully |

#### 9.5 Start recording

Function prototype: + (void) requestChatGPTStartDialogue :(nullable void(^)(BOOL))completion;

Function description: Add start recording monitoring

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL success) | Whether the schedule key is deleted successfully |

#### 9.6 End Recording

Function prototype: + (void) requestChatGPTEndDialogue :(nullable void(^)(BOOL))completion;

Function description: Add end recording monitoring

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL success) | Whether the schedule key is deleted successfully |

#### 9.7 Start GPT Q&A

Function prototype: + (void) requestChatGPTStartDialogue :(nullable void(^)(BOOL))completion;

Function description: Add start recording monitoring

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL success) | Whether the schedule key is deleted successfully |

#### 9.8 End of GPT Q&A

Function prototype: + (void) requestChatGPTEndDialogue :(nullable void(^)(BOOL))completion;

Function description: Add end recording monitoring

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)(BOOL success) | Whether the schedule key is deleted successfully |

#### 9.9 Remove the start of recording monitoring

Function prototype: + (void) removeListenChatGPTStartRecoding ;

Function description: Add start recording monitoring

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| -- | -- | -- |

#### 9.10 Remove the end of recording monitoring

Function prototype: + (void) removeListenChatGPTEndRecoding ;

Function description: Add end recording monitoring

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| -- | -- | -- |

### 10.Notification

#### 10.1 Set App notification switch status

Function prototype: + (void) syncAppSwitch :( MBAppBundleIdentifer ) identifer isOn :(BOOL) isOn completion:(nullable void(^)(BOOL))completion;

Function description: Set the App notification switch status

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| identifer | MBAppBundleIdentifer | app bundle identifier |
| completion | void(^)(BOOL success) | The App notifies you whether the switch status is set successfully. |

#### 10.2 Get the switch status of all notifications

Function prototype: + (void) requestAllSwitchs :(nullable void(^)( NSArray < MBAppNotification \*>\*))completion;

Function description: Get the switch status of all notifications

|  |  |  |
| --- | --- | --- |
| **parameter** | **type** | **describe** |
| completion | void(^)( NSArray < MBAppNotification \*>\*notifications) | Switch all notifications |

/// app bundle identifier   
@property (nonatomic, copy) MBAppBundleIdentifer bundleIdentifier ;   
  
/// URL scheme of third-party App@property (nonatomic, copy) NSString \*scheme;   
  
@property (nonatomic, copy) NSString \* appName ;   
  
@property (nonatomic, copy) NSString \* imageName ;   
  
@property (nonatomic, copy) NSString \* sortKey ;   
  
/// Is the switch of message notification turned on?@property (nonatomic, assign) BOOL isOn ;   
  
/// Is the third-party App installed on the phone?   
@property (nonatomic, assign) BOOL isInstalled ;   
  
@property (nonatomic, assign) BOOL isEnabled ;

### 11.Notification

#### 11.1 Voice Assistant - Microphone Switch - Settings

Function prototype: + (void) syncMicrophoneSwitch :(BOOL) isOn

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

Function description: Microphone switch setting

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| isOn | BOOL | Disable 0x00 Enable 0x01 |
| completion | (nullable void(^)(BOOL))completion | Set the microphone switch successfully |

#### 11.2 Voice Assistant - Microphone Switch - Get

Function prototype: + (void) requestMicrophoneSwitch :(nullable void(^)(BOOL))completion;

Function description: Microphone switch - get

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| completion | (nullable void(^)(BOOL))completion | Set the microphone switch status  0x00 Close  0x01 Enable |

#### 11.3 Voice Assistant - APP Open (Disable the 2S long press wake-up switch) - Settings

Function prototype: + (void) syncVoiceWakeupSwitch :(BOOL) isOn

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

Function description: APP open (turn off the long press 2S wake-up switch) - Settings

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| isOn | BOOL | 0x00 Close  0x01 Enable |
| completion | (nullable void(^)(BOOL))completion | Set the APP to open (turn off the long press 2S wake-up switch) status |

#### 11.4 Voice Assistant - APP Open (Close the long press 2S wake-up switch) - Get

Function prototype: + (void) requestVoiceWakeupSwitch :(nullable void(^)(BOOL))completion;

Function description: APP open (turn off the long press 2S wake-up switch) - get

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| completion | (nullable void(^)(BOOL))completion | Set the APP to open (turn off the long press 2S wake-up switch) status  0x00 Close  0x01Wake up |

#### 11.5 Voice Assistant - Setting Status - Wake Up

Function prototype: + (void) syncVoiceWakeupStatus :(BOOL) isOn

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

Function description: Set status-wake up

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| isOn | BOOL | 0: End wake-up 1: Wake up |
| completion | (nullable void(^)(BOOL))completion | Set Status-Wake Up  0x00 Close  0x01Wake up |

#### 11.6 Voice Assistant Reply Wake-up State

Function prototype: + (void) responseVoiceWakeupResultCompletion :(nullable void(^)(BOOL))completion;

Function description: Resume wake-up status

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| completion | (nullable void(^)(BOOL))completion | Set Status-Wake Up  0x00 Close  0x01Wake up |

#### 11.7 Voice Assistant-Monitoring Setting Status (Two-Way Command)

Function prototype: + (void) addVoiceAssistantStatusListener :(void(^)(BOOL))listener;

Function description: Resume wake-up status

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| listener | (void(^)(BOOL))listener | Set Status-Wake Up  0x00 Close  0x01Wake up |

#### 11.8 Voice Assistant - Monitoring whether the state is in the voice application

Function prototype + (void) addVoiceStreamListener :(void(^)(BOOL))listener;

Function description: Monitor whether the voice application is in the state

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| listener | (void(^)(BOOL))listener | Whether the "voice wake-up" module is on the device side  0x00 is not present  0x01 |

#### 11.9 Voice Assistant - Voice Conversation Content

Function prototype + (void) syncVoiceBinaryPacket : (UInt16) type

data:( NSData \*)data

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

Function description: Voice conversation content

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| type | UInt16 | See VoiceAssistantType enumeration |
| data | NSData | Convert text data utf8 to NSString |
| isEnd | BOOL | Is this session ended? |
| completion | (nullable void(^)(BOOL))completion | Whether the "voice wake-up" module is on the device side  0x00 is not present  0x01 |

typedef NS\_ENUM(uint16\_t, VoiceAssistantType ) {  
 VoiceAssistantTypeNone = 0,  
 VoiceAssistantTypeQuestion = 1,  
 VoiceAssistantTypeAnswer = 2,  
 VoiceAssistantTypePrompt = 3,  
 VoiceAssistantTypeOpen = 4,  
 VoiceAssistantTypeClose = 5,  
 VoiceAssistantTypeLcdOn = 6, //Turn on the screen  
 VoiceAssistantTypeLcdOff = 7, // Turn off the screen  
 VoiceAssistantTypeGoBack = 16, // Return to the previous level  
 VoiceAssistantTypeExit = 17, //Exit the current module  
 VoiceAssistantTypeChoice = 18,  
 VoiceAssistantTypePlayTTS = 19,   
/// QQ music control VoiceAssistantTypeMusicPlay = 20,  
 VoiceAssistantTypeMusicPause = 21,  
 VoiceAssistantTypeMusicNext = 22,  
 VoiceAssistantTypeMusicPrev = 23,   
//View or create a schedule VoiceAssistantTypeScheduleView = 26,  
 VoiceAssistantTypeScheduleCreate = 27,   
// Make a call VoiceAssistantTypePhoneByName = 28, // Make a call by name  
 VoiceAssistantTypePhoneByNumber = 29, //Call by number  
 VoiceAssistantTypeWakeUp = 30, //Wake-up voice callback   
};

#### 11.10 Voice Assistant - Synchronize Navigation List Address to Device

Function prototype + (void)syncVoiceMapBinaryPacket:(NSArray<MBGlassesNavigationObject\*> \*)array

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

Function description: Voice synchronization navigation list address to the device side

|  |  |  |
| --- | --- | --- |
| parameter | type | describe |
| array | NSArray < MBGrassesNavigationObject \*> | See MBGlassesNavigationObject type |
| completion | (nullable void(^)(BOOL))completion | Synchronize data successfully |

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