



EC SDK

6.1.T2

Demo Guide(iOS)

Issue 01

Date 2018-06-08

Copyright © Huawei Technologies Co., Ltd. 2018. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: <http://e.huawei.com>

Contents

1 Introduction.....	1
2 Quick Experience.....	2
2.1 Applying for a Service Account.....	2
2.2 Development Environment Preparation.....	2
2.3 SDK Import.....	2
2.4 Debug Run Demo.....	3
3 Demo Code Framework.....	4
3.1 Folder Structure.....	5
3.2 Logic Framework.....	7
4 Demo Function List.....	8
4.1 Supporting CloudEC Function.....	12
5 How to Crop.....	18
5.1 Crop Description.....	18
5.2 Example 1:cropping "instant message".....	18
6 FAQ.....	29
7 Change History.....	30

1 Introduction

“CloudEC_Client_API_Demo_iOS” fit to CloudEC 6.1 solution, base on Terminal SDK, provides basic service implementation samples for third-party applications.

Developers can customize interfaces based on the samples to develop their own special services.

2 Quick Experience

2.1 Applying for a Service Account

Use Huawei eSDK RemoteLab Environment

For details on how to schedule and use RemoteLab, see the [Operating Instructions for RemoteLab](#), Obtain the service server address and service account information.

Use the partner's own test or production environment

Contact to your environment administrator to obtain service server address and service account information.

2.2 Development Environment Preparation

Development tools

- Xcode: Xcode 8.0 and above version.

SDK software package

- Access <http://developer.huawei.com>, obtain eSDK API iOS SDK。

eSDK EC API V1.1.RC1.SPC001 iOS



下载

Demo Source code

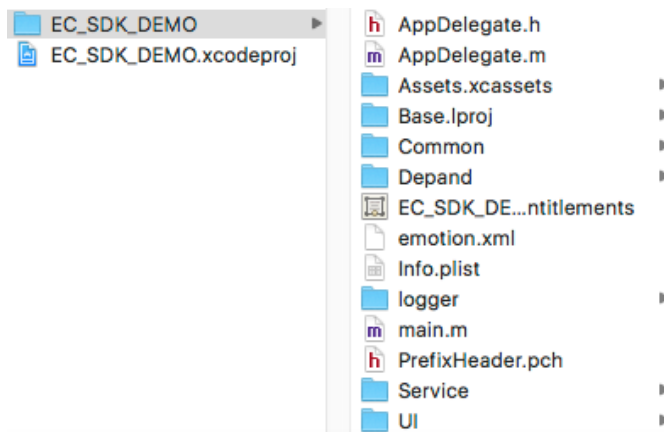
- Access <https://github.com/Huawei>, obtain “CloudEC_Client_API_Demo_iOS”。

2.3 SDK Import

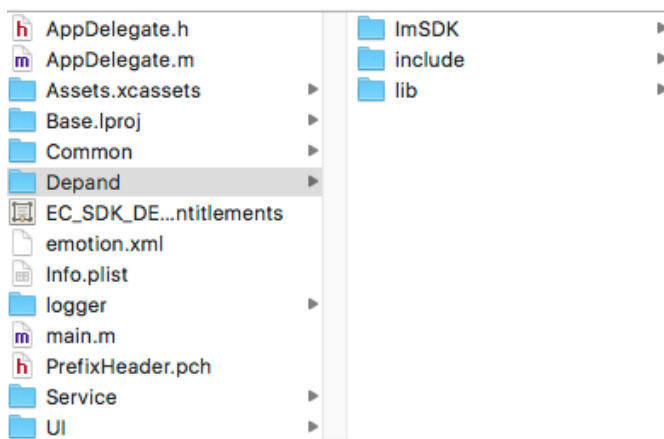
Step 1 Unzip the downloaded eSDK API iOS SDK package to disk:



Step 2 Unzip the downloaded Demo source code and copy it to the working directory:



Step 3 Copy the “Step 1”unpacked package to the “Depand”folder:



----End

2.4 Debug Run Demo

Step 1 Open Xcode, import Demo project processed in capture 2.3.

Step 2 Build



NOTE

In general situation, there is no error in the build process, and if there are errors, please correct them according to the Xcode.

Step 3 Connect iOS to machine debug, process debug.

Step 4 Complete configuration of server address, port, enter login account and password, finish login.

----End

3 Demo Code Framework

3.1 Folder Structure

Figure 3-1 folder structure

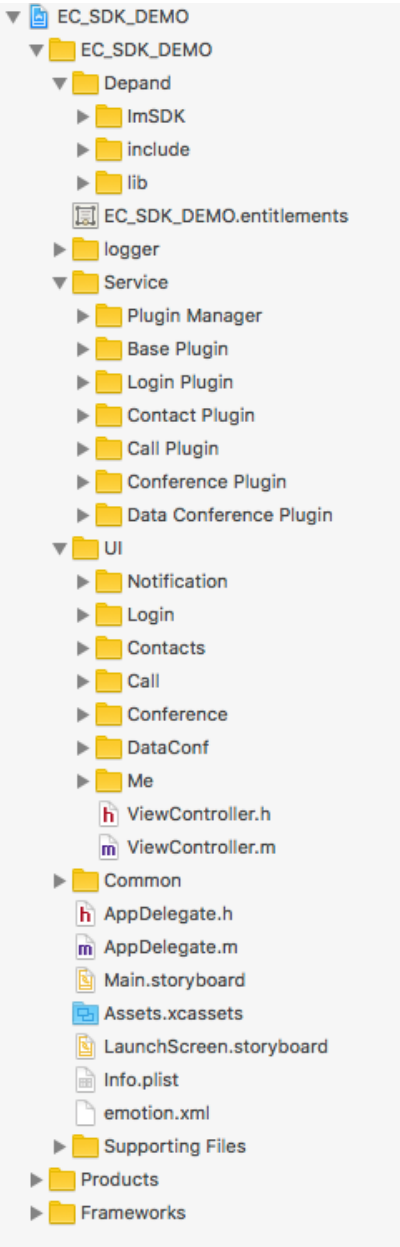
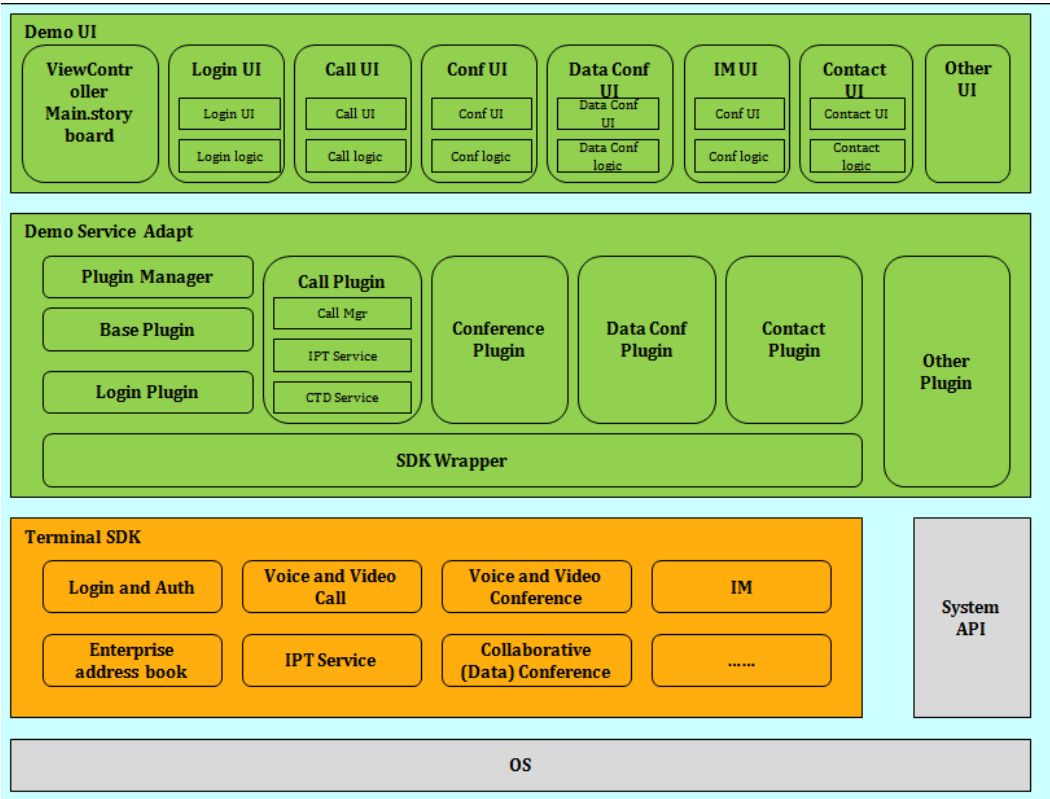


Table 3-1 folder description

Folder	Description
EC_SDK_DEMO\Depand	ImSDK, Tup lib, Tup head file saved path
EC_SDK_DEMO\logger	Log print file saved path

Folder	Description
EC_SDK_DEMO\Service\Plugin Manager	"Service Management" project library folder, containing the implementation code for the demo's configuration of the specific business (whether it supports a service), initialization, and uninitialization of service management.
EC_SDK_DEMO\Service\Base Plugin	"Base Service" project library folder, containing implementation code of management about base message dispatch
EC_SDK_DEMO\Service>Login Plugin	"Login Service" project library folder, containing the implementation code for the overall login entry for all service logic based on the SDK implementation demo.
EC_SDK_DEMO\Service>Contact Plugin	"Contact Service" project library folder, containing service code for enterprise contact (address book) based on SDK.
EC_SDK_DEMO\Service\Call Plugin	"Audio and video call service" project library folder, including the implementation of audio and video calls based on the SDK, IPT service and CTD service code
EC_SDK_DEMO\Service\Conference Plugin	"Audio and video conference services" project library folder, including the implementation of audio and video conference based on the SDK(including data conference) service code.
EC_SDK_DEMO\Service\Data Conference Plugin	"Data conference services" project library folder, including the implementation of data conference service based on the SDK service code.
EC_SDK_DEMO\UI	Every server module page implement code, including IM,login, contact, call, conference, data conference, config and so on UI code.
EC_SDK_DEMO\Common	"Public Service" project library folder, which contains other service dependent public function implementation code.
EC_SDK_DEMO\Main.storyboard	main page xib implementation.

3.2 Logic Framework



4 Demo Function List

Function Item	Function Sub Item	Whether Open	Demo Whether Support
Authentication and Login	normal login	Y	Y
	ticket authentication login	Y	N
	logout	Y	Y
Voice Call	voice Call Setup	Y	Y
	end/cancel/reject call	Y	Y
	accept voice call	Y	Y
	set(or cancel) mute mic	Y	Y
	send DTMF signal	Y	Y
Video Call	video call setup	Y	Y
	end/cancel/reject call	Y	Y
	accept video call	Y	Y
	switch voice call to a video call	Y	Y
	switch video call to a voice call	Y	Y
	accept or reject switch voice call to a video call	Y	Y
	open local video preview	Y	N
	close local video preview	Y	N
	open or close local camera	Y	Y
Device management	switch mic	Y	N

	switch speak	Y	Y
	switch camera	Y	Y
	set call volume	Y	N
Hold&Unhold	hold and unhold voice call	Y	Y
	hold and unhold video call	Y	Y
Divert	call divert	Y	Y
Transfer	blind transfer	Y	Y
Regist IPT service	regist or deregist call wait	Y	Y
	regist or deregist DND	Y	Y
	regist or deregist forward	Y	Y
CTD	set up CTD call	Y	Y
Conference management	book conference	Y	Y
	query conference list	Y	Y
	query conference details	Y	Y
Basic conference control	create instant conference	Y	Y
	get conference info	Y	Y
	invite attendee	Y	Y
	delete attendee	Y	Y
	end conference	Y	Y
	leave conference	Y	Y
	mute conference	Y	Y
	mute attendee	Y	Y
	lock conference	Y	Y
	release chairman	Y	Y
	request chairman	Y	Y
	prolong conference	N	NA
	set or cancel hand up	Y	Y
	display speaker	Y	N
	join a conference from the conference list	Y	Y
	join conference by an access code	Y	Y

	IVR join conference	Y	Y
Progressive Conference	Two-Party Call-to-Conference Conversion	Y	N
	upgrade conference	Y	Y
Base control	join data conference	Y	Y
	leave data conference	Y	Y
	end data conference	Y	Y
Video	base video control	Y	Y
	watch video	Y	Y
	set video param	Y	Y
Desktop Sharing	watch desktop sharing	Y	Y
Application Sharing	watch application sharing	Y	Y
Document	watch document sharing	Y	Y
Whiteboard	watch white board sharing	Y	Y
Annotation	watch annotation	Y	Y
Chat	send public instant message	Y	N
Personal information access and setting	get and set user information	Y	Y
	get and set user icon	Y	Y
Personal contact	get contact and group list	Y	Y
	obtain icon	Y	Y
	add friend	Y	Y
	delete contact or friend	Y	Y
Enterprise contact	query enterprise contact	Y	Y
	query user info	Y	Y
Status and remind	set personal status	Y	Y
	subscribe user status	Y	Y
Fixed group	dissolution fixed group	Y	Y
	modify fixed group	Y	Y
	approve fixed group invitation	Y	Y
	quite fixed group	Y	Y
	obtain fixed group list	Y	Y

	obtain fixed group member list	Y	Y
	obtain fixed group detail	Y	Y
Discussion group	create discussion group	Y	Y
	fix discussion group	Y	Y
	modify discussion group	Y	Y
	invite discussion group member	Y	Y
	delete discussion group member	Y	Y
	quite discussion group	Y	Y
	obtain discussion group list	Y	Y
	obtain discussion group member list	Y	Y
	obtain discussion group detail	Y	Y
Text message	send instant message	Y	Y
	get roaming messages	Y	Y
	set message has read	Y	Y
	message notify	Y	Y
Push	open Push	Y	N
	close Push	Y	N
	Push message handle	Y	N
Rich media	send and receive picture	Y	Y
	send and receive audio	Y	Y
	send and receive video	Y	Y
	send and receive normal offline file	Y	Y
Internet address book	query contact	Y	Y
	query department	Y	Y
	get and set contact icon	Y	Y

4.1 Supporting CloudEC Function

Charact	Function Item	Function Sub Item	Whether Open (Host ed)	Demo Whether Support	Demo Description
Login	Authentication and Login	normal login	Y	Y	not support STG at present
		ticket authentication login	Y	N	not support server environment, low priority
		logout	Y	Y	not support STG at present
VoIP	Voice Call	voice Call Setup	Y	Y	
		end/cancel/reject call	Y	Y	
		accept voice call	Y	Y	
		set(or cancel) mute mic	Y	Y	
		send DTMF signal	Y	Y	
	Video Call	video call setup	Y	Y	
		end/cancel/reject call	Y	Y	
		accept video call	Y	Y	
		switch voice call to a video call	Y	Y	
		switch video call to a voice call	Y	Y	
		accept or reject switch voice call to a video call	Y	Y	
		open local video preview	Y	N	not implemented
		close local video preview	Y	N	not implemented

Charact	Function Item	Function Sub Item	Whether Open (Host ed)	Demo Whether Support	Demo Description
	Device management	open or close local camera	Y	Y	
		switch mic	Y	N	use system config
		switch speak	Y	Y	
		switch camera	Y	Y	
		set call volume	Y	N	use system config
IPT	Hold&unhold	hold and unhold voice call	Y	Y	
		hold and unhold video call	Y	Y	
	Divert	call divert	Y	Y	
	Transfer	blind transfer	Y	Y	
	Regist IPT service	regist or deregist call wait	Y	Y	
		regist or deregist DND	Y	Y	
		regist or deregist forward	Y	Y	
	CTD	set up CTD call	Y	Y	
ConfC trl	Conference management	book conference	Y	Y	
		query conference list	Y	Y	
		query conference details	Y	Y	
	Basic conference control	create instant conference	Y	Y	
		get conference info	Y	Y	
		invite attendee	Y	Y	
		delete attendee	Y	Y	
		hang up attendee	Y	Y	

Charact	Function Item	Function Sub Item	Whether Open (Host ed)	Demo Whether Support	Demo Description
		recall attendee	Y	Y	
		end conference	Y	Y	
		leave conference	Y	Y	
		mute conference	Y	Y	
		mute attendee	Y	Y	
		lock conference	Y	Y	
		release chairman	Y	Y	
		request chairman	Y	Y	
		prolong conference	Y	N	not implemented
		set or cancel hand up	Y	Y	
		display speaker	Y	N	not implemented
		join a conference from the conference list	Y	Y	
		join conference by an access code	Y	Y	
		IVR join conference	Y	Y	
	Video control	set conference mode	Y	Y	
		broadcast conference	Y	Y	
		watch conference	N	Y	
	Progressive Conference	Two-Party Call-to-Conference Conversion	Y	N	not implemented
		upgrade conference	Y	Y	
	data sharing	active sharing data	Y	N	not implemented
		watch data	Y	N	not implemented

Charact	Function Item	Function Sub Item	Whether Open (Host ed)	Demo Whether Support	Demo Description
DataC onf	Base control	join data conference	Y	Y	
		leave data conference	Y	Y	
		end data conference	Y	Y	
	Desktop Sharing	watch desktop sharing	Y	Y	
	Application Sharing	watch application sharing	Y	Y	
	Document	watch document sharing	Y	Y	
	Whiteboard	watch white board sharing	Y	Y	
	Annotation	watch annotation	Y	Y	
	Chat	send public instant message	Y	N	not implemented
IM	Personal information access and setting	get and set user information	Y	Y	
		get and set user icon	Y	Y	
	Personal contact	get contact and group list	Y	Y	
		obtain icon	Y	Y	
		add friend	Y	Y	
		delete contact or friend	Y	Y	
	Enterprise contact	query enterprise contact	Y	Y	
		query user info	Y	Y	
	Status and remind	set personal status	Y	Y	
		subscribe user status	Y	Y	
	Fixed group	dissolution fixed group	Y	Y	
		modify fixed group	Y	Y	

Charact	Function Item	Function Sub Item	Whether Open (Host ed)	Demo Whether Support	Demo Description
		approve fixed group invitation	Y	Y	not implemented
		quite fixed group	Y	Y	
		obtain fixed group list	Y	Y	
		obtain fixed group member list	Y	Y	
		obtain fixed group detail	Y	Y	
	Discussion group	create discussion group	Y	Y	
		fix discussion group	Y	Y	
		modify discussion group	Y	Y	
		invite discussion group member	Y	Y	
		delete discussion group member	Y	Y	
		quite discussion group	Y	Y	
		obtain discussion group list	Y	Y	
		obtain discussion group member list	Y	Y	
		obtain discussion group detail	Y	Y	
	Text message	send instant message	Y	Y	
		get roaming messages	Y	Y	
		set message has read	Y	Y	
		message notify	Y	Y	

Charact	Function Item	Function Sub Item	Whether Open (Host ed)	Demo Whether Support	Demo Description
	Push	open Push	Y	N	not implemented
		close Push	Y	N	not implemented
		Push message handle	Y	N	not implemented
	Rich media	send and receive picture	Y	Y	
		send and receive audio	Y	Y	
		send and receive video	Y	Y	
		send and receive normal offline file	Y	Y	
eAddr	Internet address book	query contact	Y	Y	
		query department	Y	Y	
		get and set contact icon	Y	Y	

5 How to Crop

5.1 Crop Description

EC SDK Demo(iOS) devided module base on the functional charact, can be based on the actual needs of developers, in the modification of a small number of code, the need to cut the function.

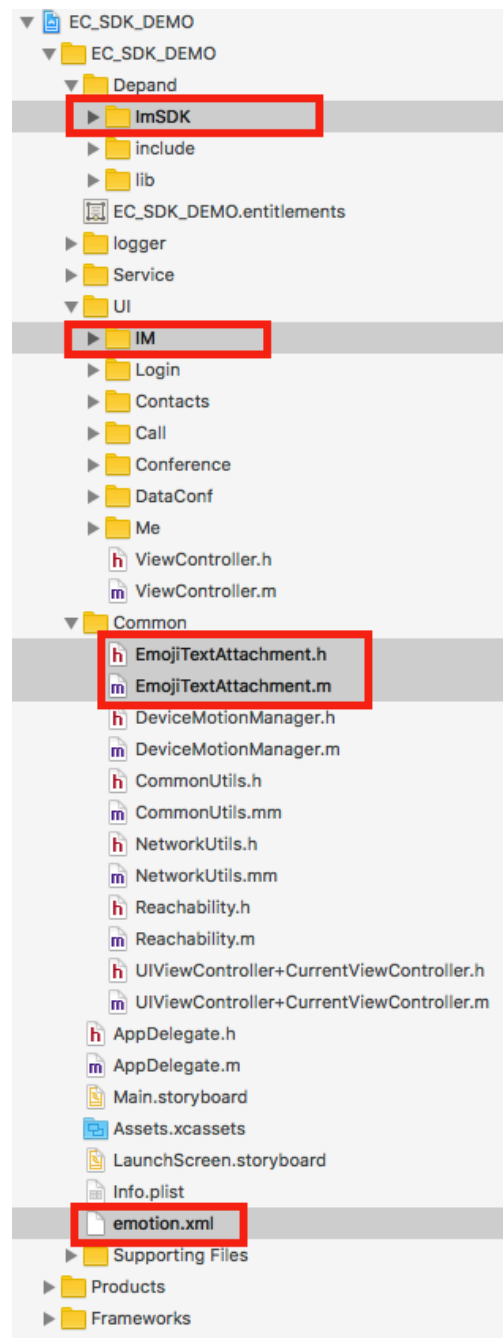
5.2 Example 1:cropping "instant message"



No special instructions, red box  in the content indicates that needs to be deleted.

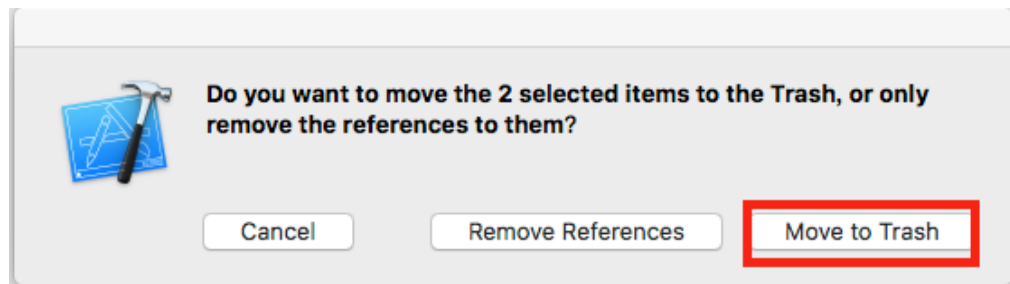
Step 1 Delete the unused project, packages, files, and file directories.

Figure 5-1

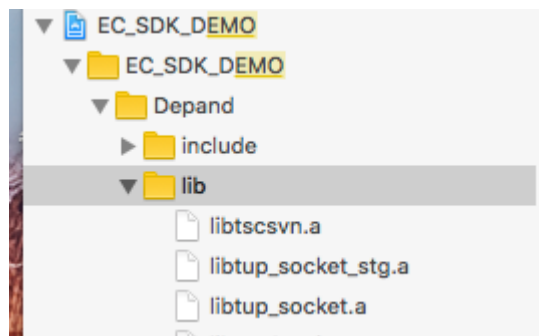


Circled part in Figure1, right click Delete, choose move to trash as figure2

Figure 5-2



Step 2 Add relative library file



right click lib, click Add File to "xxxx", choose library as follow, click add
libtscsvn.a ,libtup_socket_stg.a ,libtup_socket.a ,libtupService.a ,libuspsdk.a

Step 3 Delete unused page and related entrance from Main.storyboard

Figure 5-3

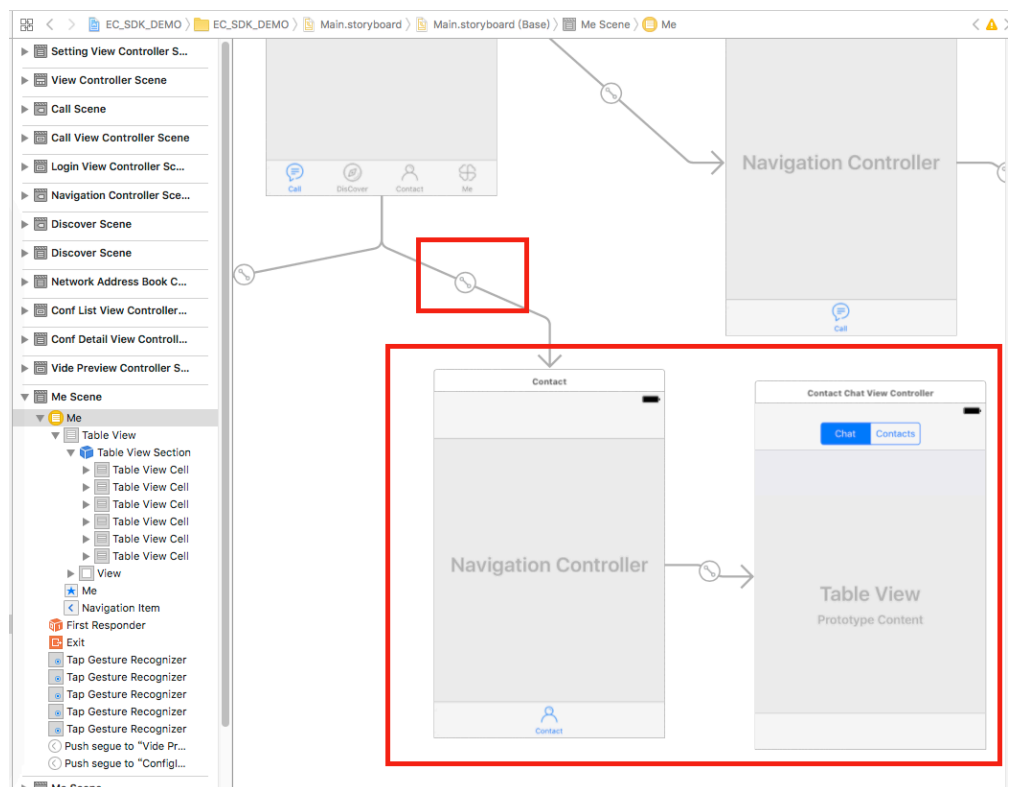
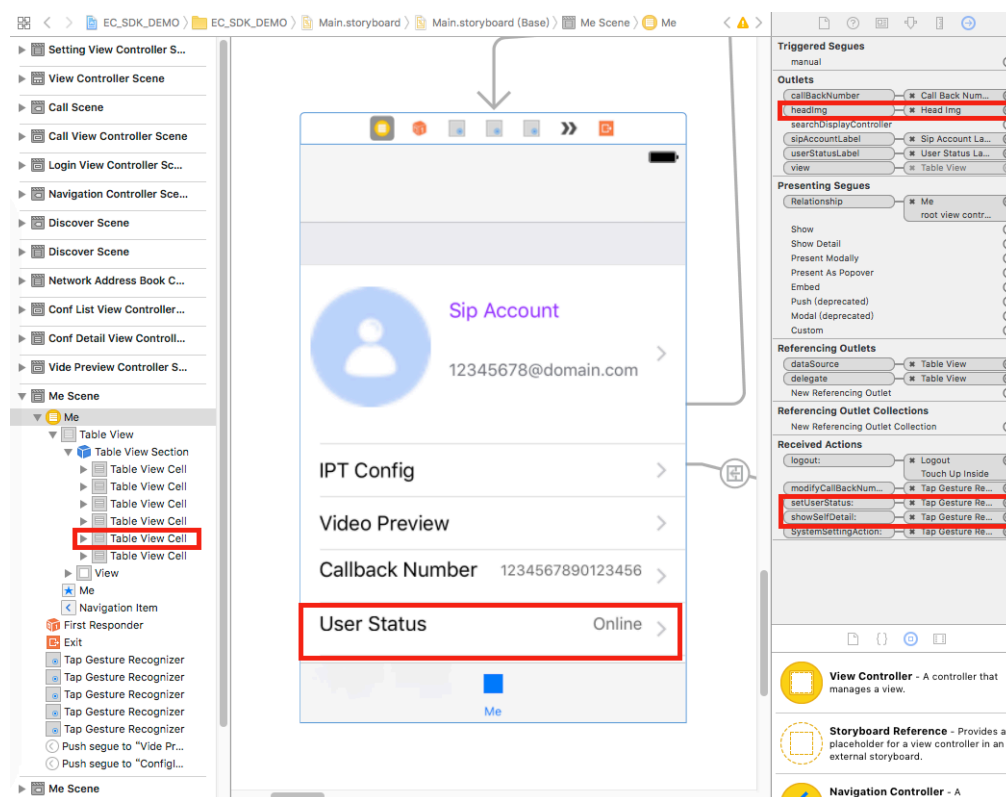


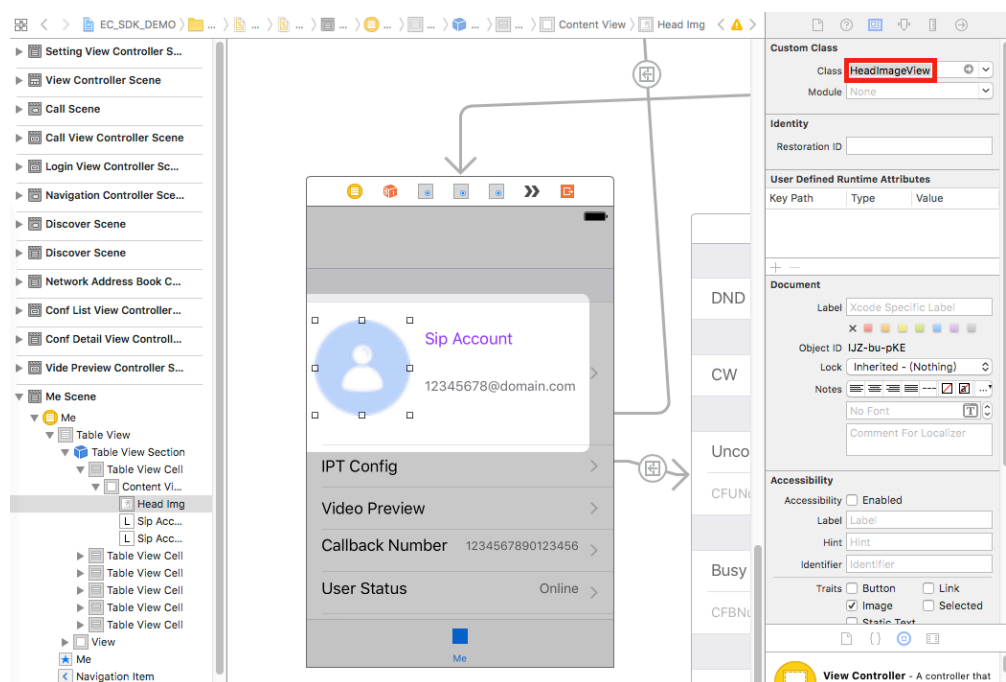
figure3 delete red circle link and page.

Figure 5-4



In Figure 4, firstly select the ME controller, click the x in middle of the circle part to delete, and the table view cell in the left circle(see if it corresponds to the middle circle).

Figure 5-5



In Figure 5, firstly select the head view of the pair, delete the circled part, and then click Enter.

Step 4 Delete unused code from AppDelegate.m

Figure 5-6

```
#import "AppDelegate.h"
#import "NetworkUtils.h"
#import <TUPIOSSDK/TUPIOSSDK.h>
#import "ServiceManager.h"
#import "CallWindowController.h"
#import "ConfRunningViewController.h"
#import "ECSLogger.h"
#import "LocalNotificationCenter.h"
#import "ConfStatus.h"

@interface AppDelegate ()

@end
```

Figure 5-7

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)
launchOptions {

    [[UINavigationController appearance] setBarTintColor:[UIColor colorWithRed:44.0/255 green:110.0/255 blue:
    232.0/255 alpha:1]];
    [[UINavigationController appearance] setTintColor:[UIColor whiteColor]];
    [[UINavigationController appearance] setTitleTextAttributes:@{NSForegroundColorAttributeName : [UIColor
    whiteColor]}];
    // Override point for customization after application launch.

    // DDLog init (write log to file)
    [[ECSLogger sharedInstance] addFileLogger];
    [NetworkUtils sharedInstance];

    // Config TUPIOSSDK log path
    NSString *logPath = [[NSHomeDirectory() stringByAppendingPathComponent:@"Documents"]
    stringByAppendingString:@"TUPO69log"];
    [ECSUserDefaults sharedInstance].logFileSuperPath = logPath;

    // init TUPIOSSDK (module: Contact/IM/Group/MALogin)
    [ECSAppConfig sharedInstance].appLogLevel = kECSLogDebug;
    [ECSAppConfig sharedInstance].isLogEnabled = YES;
    [ECSAppConfig sharedInstance].version = @"V3.0.4.5";
    if ([ECSAppConfig sharedInstance].isFirstUsed)
    {
        [[ECSAppConfig sharedInstance] initializeSecurityRandomKey];
    }
    [TUPIOSSDKService start];

    // start up tup module.(module: Call/Conference/SipLogin)
    [ServiceManager startup];

    // other UI Things
    [CallWindowController sharedInstance];
    [[LocalNotificationCenter sharedInstance] start];

    return YES;
}
```

Figure 5-8

```
- (void)applicationWillResignActive:(UIApplication *)application {
    // Sent when the application is about to move from active to inactive state. This can occur for
    // certain types of temporary interruptions (such as an incoming phone call or SMS message) or
    // when the user quits the application and it begins the transition to the background state.
    // Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering
    // callbacks. Games should use this method to pause the game.

    [[LOCAL_DATA_MANAGER managedObjectContext] saveToPersistent];
    [[ECSAppConfig sharedInstance] save];
}

- (void)applicationDidEnterBackground:(UIApplication *)application {
    // Use this method to release shared resources, save user data, invalidate timers, and store
    // enough application state information to restore your application to its current state in case
    // it is terminated later.
    // If your application supports background execution, this method is called instead of
    // applicationWillTerminate: when the user quits.

    __block UIBackgroundTaskIdentifier taskID = [[UIApplication sharedApplication]
        beginBackgroundTaskWithExpirationHandler:^(
            [[UIApplication sharedApplication] endBackgroundTask:taskID];
            taskID = UIBackgroundTaskInvalid;
        )];

    [[LOCAL_DATA_MANAGER managedObjectContext] saveToPersistent];
    [[ECSAppConfig sharedInstance] save];
}
```

Figure 5-9

```
- (void)applicationWillTerminate:(UIApplication *)application {
    // Called when the application is about to terminate. Save data if appropriate. See also
    // applicationDidEnterBackground:.

    [[LOCAL_DATA_MANAGER managedObjectContext] saveToPersistent];
    [[ECSAppConfig sharedInstance] save];
}

- (void)application:(UIApplication *)application didRegisterForRemoteNotificationsWithDeviceToken:
(NSData *)deviceToken {
    // Config APNS type and device token to MAA for APNS Notification.
    const unsigned *tokenBytes = (const unsigned *) [deviceToken bytes];
    NSString *hexToken = [NSString stringWithFormat:@"%08x%08x%08x%08x%08x%08x%08x%08x",
        ntohs(tokenBytes[0]), ntohs(tokenBytes[1]), ntohs(tokenBytes[2]),
        ntohs(tokenBytes[3]), ntohs(tokenBytes[4]), ntohs(tokenBytes[5]),
        ntohs(tokenBytes[6]), ntohs(tokenBytes[7])];

    [ECSAppConfig sharedInstance].deviceToken = hexToken;
    [ECSAppConfig sharedInstance].apnsType = APNS_DEV;
}
```

Delete circled part code in figure6、7、8、9.

Step 5 Delete unused code from CommonUtils.mm.

Figure 5-10

```
// CommonUtils.m
// EC_SDK_DEMO
//
// Created by EC Open support team
// Copyright (C), 2017, Huawei Tech
//

#import "CommonUtils.h"
#include <netdb.h>
#include <net/if.h>
#include <ifaddrs.h>
#include <arpa/inet.h>
#include <dlfcn.h>
#include <svs/svsctl.h>
#include <TUPIOSSDK/TUPIOSSDK.h>
```

Figure 5-11

```

/**
 *This method is used to get Image from disk
 *从图片文件夹获取图片
 *@param imageFile image file path
 *@return image
 */
-(UIImage *) attachImageFile:(NSString*) imageFile {
    UIImage* image = [[ESpaceImageCache sharedInstance] imageWithKey:imageFile];
    if (image) {
        return image;
    } else {
        NSData* imageData = [[NSFileManager defaultManager] contentsAtPath:imageFile];
        UIImage* image = nil;
        if (imageData) {
            image = [UIImage imageWithData:imageData];
            return image;
        }
    }
    return nil;
}

```

Delete unused code from figure10、 11

Step 6 Delete unused code from LoginService.mm.

Figure 5-12

```

1 #import "LoginService.h"
2 #import "LoginServerInfo+uportalInfo.h"
3 #include "string.h"
4 #include <netdb.h>
5 #include <net/if.h>
6 #include <ifaddrs.h>
7 #include <arpa/inet.h>
8 #include <dlfcn.h>
9 #include <sys/sysctl.h>
10 #import "login_interface.h"
11 #import "LoginInfo.h"
12 #import "Initializer.h"
13 #import "login_def.h"
14 #import "LoginCenter.h"
15 #import "CommonUtils.h"
16 #import "ManagerService.h"
17 #import <TUPIOSSDK/TUPMAALoginService.h>
18 #import <TUPIOSSDK/ECSAppConfig.h>
19 #import <TUPIOSSDK/eSpaceDBService.h>
20 #import <TUPNetworkSDK/ECSSocketController.h>
21 #import "call_interface.h"
22
23 #define NEEDMAALOGIN 1 // 是否需要MAA登陆
24 @interface LoginService()

```

Delete circled part code in figure 12, and modify define NEEDMAALOGIN as 0.

Step 7 Delete unused code from MeViewController.m.

Figure 5-13

```
#import "MeViewController.h"
#import "ManagerService.h"
#import "CommonUtils.h"
#import "HeadImageView.h"

#import <TUPIOSSDK/eSpaceDBService.h>
#import <TUPContactSDK/TupContactService.h>
#import <TUPIOSSDK/TUPUserService.h>

#import <TUPIOSSDK/eSpaceDBService.h>
#import "PersonDetailViewController.h"

#define NEEDREGISTERMAALOGOUT 1 //是否需要MAA注销
@interface MeViewController ()
@property(weak, nonatomic) IBOutlet UILabel *sipAccountLabel;
@property(weak, nonatomic) IBOutlet UILabel *callBackNumber;
@property(weak, nonatomic) IBOutlet UILabel *userStatusLabel;
@property(weak, nonatomic) IBOutlet HeadImageView *headImg;
@end

@implementation MeViewController

- (void)viewDidLoad {
    [super viewDidLoad];
    sipAccountLabel.text = [ManagerService callService].sipAccount;
    NSString *account = [[eSpaceDBService sharedInstance].localDataManager
        userAccount];
    ESspaceUserOnlineStatus* status = [[TupContactService sharedInstance]
        onlineStatusForUser:account];
    [self reloadUserStatus:status.userStatus];
    [self updateCallBackNumber];
    [self.headImg setContactEntity:LOCAL_DATA_MANAGER.currentUser];
    // Do any additional setup after loading the view.
}

- (void)viewWillLayoutSubviews {
    [super viewWillLayoutSubviews];
    self.headImg.layer.cornerRadius = 50.0f;
}
```

Delete circled part code in figure 13, and modify define NEEDREGISTERMAALOGOUT as 0.

Figure 5-14

```
- (IBAction) showSelfDetail:(id)sender {}

- (IBAction) setUserStatus:(UITapGestureRecognizer *)sender {}

- (void) setSelfUserStatus:(ESpaceUserStatus) status
{ }

- (void) reloadUserStatus:(ESpaceUserStatus) status { }
```

Delete circled part code in figure 14.

Step 8 Delete unused code from viewController.m.

Figure 5-15

```
#import "ViewController.h"
#import "AppDelegate.h"
#import "LoginViewController.h"
#import <TUPIOSSDK/TUPIOSSDK.h>
#import "ManagerService.h"
#import "UIViewController+CurrentViewController.h"
#import "DataConfBaseViewController.h"

@interface ViewController ()<LoginServiceDelegate>
@property (nonatomic,assign)BOOL isBeKickOut;
@end

@implementation ViewController

-(void)viewWillAppear:(BOOL)animated
{
    [super viewWillAppear:animated];
    [ManagerService_loginService].delegate = self;
    [[TUPMAALoginService sharedInstance].authType = 4;
}

-(void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    _isBeKickOut = NO;
    self.title = @"Main";
    [[TUPMAALoginService sharedInstance].loginService addObserver:self forKeyPath:@"serviceStatus"
    options:NSKeyValueObservingOptionNew context:NULL];
}
```

Figure 5-16

```
-(void)observeValueForKeyPath:(NSString *)keyPath
ofObject:(id)object
change:(NSDictionary<NSKeyValueChangeKey,id> *)change
context:(void *)context {}

-(void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

-(void)loginEventCallback:(TUP_LOGIN_EVENT_TYPE)loginEvent result:(NSDictionary *)resultDictionary
{
    switch (loginEvent)
    {
        case LOGINOUT_EVENT:
        {
            if (ECSLogin == [TUPMAALoginService sharedInstance].loginService.serviceStatus) {
                [[TUPMAALoginService sharedInstance] logout:^(NSError *error) {
                    dispatch_async(dispatch_get_main_queue(), ^{
                        if (error) {
                            [self showMessage:@"Logout failed!"];
                        }else {
                            [[ECSAppConfig sharedInstance] save];
                            [[LOCAL_DATA_MANAGER managedObjectContext] saveToPersistent];
                            [self goToLoginViewController];
                        }
                    });
                }
            }
        }
        else {
            dispatch_async(dispatch_get_main_queue(), ^{
                [self goToLoginViewController];
            });
        }
        break;
    }
    default:
        break;
    }
}
```

Figure 5-17

```
-(void)dealloc
{
    [ManagerService_loginService].delegate = nil;
    [[TUPMAALoginService sharedInstance].loginService removeObserver:self forKeyPath:@"serviceStatus"
    context:NULL];
}
```

Delete circled part code from figure 15、16、17

Step 9 Delete unused code from ContactService.m.

Figure 5-18

```
#import "Initializer.h"
#import "ContactService.h"
#import "LoginServerInfo.h"
#import "SearchParam.h"
#import "SearchResultInfo.h"
#import "DeptInfo.h"
#import "ContactInfo.h"

#import <UIKit/UIKit.h>
#import "tup_eaddr_def.h"
#import "tup_eaddr_interface.h"
#import <TUPIOSSDK/EmployeeEntity.h>
#import <TUPIOSSDK/eSpaceDBService.h>
#import <TUPIOSSDK/ESpaceImageCache.h>
```

Figure 5-19

```
/**
 * This method is used to set system head image (0~9) (if completionBlock result is YES,
 * set self head ID with sysIconID)
 * 设置系统头像
 * @param sysIconID          Indicates system head image ID
 *                            系统头像id
 * @param completionBlock    Indicates callback(result: set head image result. YES
 *                            or NO)
 *                            回调，返回设置头像成功与否
 */
- (void)setSystemHead:(int)sysIconID withCompletion:(void(^)(BOOL result))completionBlock {
    TUP_RESULT set_sys_result = tup_eaddr_set_sysicon((TUP_UINT32)sysIconID);
    BOOL result = set_sys_result == TUP_SUCCESS;
    if (completionBlock) {
        completionBlock(result);
    }
    if (result) {
        [self setHeadID:[NSString stringWithFormat:@"%d", sysIconID]];
    }
}

/**
 * headId's set method
 * 头像id 的set方法
 * @param headId NSString
 */
- (void)setHeadID:(NSString *)headId {
    EmployeeEntity *selfEntity = LOCAL_DATA_MANAGER.currentUser;
    selfEntity.headId = headId;
}
```

Figure 5-20

```
* This method is used to set custom head image (if completionBlock result is YES, set
  self hedID with headID)
* 设置自定义头像
@param image                Indicates custom image
*                            自定义头像
@param completionBlock      Indicates callback(result: set head image result. YES
  or NO)
*                            回调, 返回设置头像成功与否
*/
- (void)setHeadImage:(UIImage *)image completion:(void(^)(BOOL result, NSString *headID))
  completionBlock {
    //自定义头像接口需要上传三种尺寸的图片: 52x52 120x120 320x320
    NSData *minImg = [self imgWithSize:SIZE52 image:image];
    NSData *midImg = [self imgWithSize:SIZE120 image:image];
    NSData *maxImg = [self imgWithSize:SIZE320 image:image];

    TUP_EADDR_S_ICON* icon_info = (TUP_EADDR_S_ICON*)malloc(sizeof(TUP_EADDR_S_ICON));
    memset(icon_info, 0, sizeof(TUP_EADDR_S_ICON));
    icon_info->pcLargeIcon_data = (TUP_CHAR *)maxImg.bytes;
    icon_info->pcMediumIcon_data = (TUP_CHAR *)midImg.bytes;
    icon_info->pcSmallIcon_data = (TUP_CHAR *)minImg.bytes;
    icon_info->ulLargeIcon_len = (TUP_UINT32)maxImg.length;
    icon_info->ulMediumIcon_len = (TUP_UINT32)midImg.length;
    icon_info->ulSmallIcon_len = (TUP_UINT32)minImg.length;

    TUP_CHAR *modifyTime = (TUP_CHAR *)malloc(16);
    memset_s(modifyTime, 16, 0, 16);
    TUP_UINT32 length = 16;
    TUP_RESULT ret_set_eficon = tup_eaddr_set_deficon(icon_info, modifyTime, length);
    free(icon_info);
    // 出参modifyTime时间戳, 作为联系人headId
    NSString *mTime = [NSString stringWithUTF8String:modifyTime];
    DDLogInfo(@"set image ret: %d modify time: %@", ret_set_eficon, mTime);
    BOOL result = ret_set_eficon == TUP_SUCCESS;
    if (completionBlock) {
        completionBlock(result, mTime);
    }
    if (result) {
        [self setHeadID:mTime];
    }
}
```

Figure 5-21

```
65 - (void)contactModule:(TUP_MODULE)module notification:(Notification *)notification {
66     if (CONTACT_MODULE == module) {
67         switch (notification.msgId) {
68             // 联系人搜索结果
69             case TSDK_E_EADDR_EVT_SEARCH_CONTACTS_RESULT: {
70                 DDLogInfo(@"TSDK_E_EADDR_EVT_SEARCH_CONTACTS_RESULT");
71                 BOOL result = notification.param1 == TSDK_SUCCESS;
72                 if (!result) {
73                     DDLogError(@"TSDK_E_EADDR_EVT_SEARCH_CONTACTS_RESULT,error:%@", [NSString stringWithUTF8String:
74                         (TSDK_CHAR *)notification.data]);
75                     return;
76                 }
77                 TSDK_S_SEARCH_CONTACTS_RESULT *searchContactorResult = (TSDK_S_SEARCH_CONTACTS_RESULT
78                     *)notification.data;
79                 if (searchContactorResult == NULL) {
80                     DDLogWarn(@"handleSearchContact result is empty.");
81                     return;
82                 }
83                 int pageIndex = searchContactorResult->page;
84                 int totalNum = searchContactorResult->total_num;
85                 TSDK_S_CONTACTS_INFO *pstContactorInfo = searchContactorResult->contact_info;
86                 //查询配置自己软终端号
87                 if (notification.param2 == 100) {
88                     if (pstContactorInfo != NULL) {
89                         ContactInfo *contactInfo = [ContactInfo contactInfoTransformFrom:pstContactorInfo[0]];
90                         [[ManagerService callService] configBusinessAccount:contactInfo.terminal token:nil];
91                         [eSpaceDBService sharedInstance].localDataManager = [[ESpaceLocalDataManager alloc]
92                             initWithUserAccount:contactInfo.staffAccount];
93                     }
94                 }
95                 return;
96             }
97         }
98     }
99 }
```

----End

6 FAQ

NA

7 Change History

Publish Date	Document Version	Revise Declare
2018-05-31	01	Cooperate 6.1.0 version(at present only fit to convergent meeting network), document first published.