

# Zoom iOS SDK

## Revision History

Date	History	Author
July 30, 2015	First Draft	Robust Hu
Sep 06, 2015	1. Add notification for app become active/inactive; 2. Add ignore signal "SIGPIPE"	Robust Hu
Oct 17, 2015	1. Support to customize the content of "Message Invite" and "Copy URL"; 2. Support to enable/disable "Copy URL"	Robust Hu
Nov 03, 2015	1. Add Feature of Pause/Resume Meeting Audio;	Robust Hu
Jan 15, 2016	1. Support to enable/disable "Invite by Email"; 2. Support to customize the content and subject of "Invite by Email"	Robust Hu
Feb 02, 2016	1. Add Interface to turn on/off cloud record;	Robust Hu
Apr 21, 2016	1. Add Interface to check whether meeting is locked;	Robust Hu
Jul 04, 2016	1. Add Interface "onAppShareSplash" in ZoomSDKMeetingServiceDelegate for app share; 2. Add Interface "onJBHWaitingWithCmd:" in ZoomSDKMeetingServiceDelegate for customizing to show/hide JBH waiting.	Robust Hu
Aug 11, 2016	1. Generate ZoomSDK.framework with Xcode template "Cocoa Touch Framework"; 2. Add Interface for login with work email; 3. Add Interface for login user to schedule/edit/delete/list meeting	Robust Hu
Aug 26, 2016	1. Move ZoomSDKResources.bundle into ZoomSDK.framework; 2. Support to customize strings in Zoom SDK; 3. Support to customize images in Zoom SDK	Robust Hu

## Main class description

- **ZoomSDK**: Zoom SDK base class, take charge of initializing Zoom SDK, and return the instance of ZoomSDKAuthService, ZoomSDKPremeetingService, ZoomSDKMeetingService and ZoomSDKMeetingSettings.
- **ZoomSDKAuthService**: Zoom SDK authenticate service class, before using Zoom SDK, partner app should request authentication with app key and app secret.
- **ZoomSDKAuthDelegate**: Zoom SDK authenticate delegate class, partner app can get the authentication result by this class.
- **ZoomSDKPremeetingService**: Zoom SDK pre-meeting service class, once Zoom SDK user login with work email, he/she can use this class to schedule/edit/delete/list meeting.
- **ZoomSDKPremeetingDelegate**: Zoom SDK pre-meeting delegate class, Zoom SDK user can the event result of schedule/edit/delete/list meeting.
- **ZoomSDKMeetingService**: Zoom SDK meeting service class, take charge of starting/joining meeting.
- **ZoomSDKMeetingServiceDelegate**: Zoom SDK meeting service delegate class, partner app can listen the meeting result and meeting state by the class
- **ZoomSDKMeetingSettings**: Zoom SDK meeting settings class, take charge of setting/getting the meeting audio/video state.
- **ZoomSDKInviteHelper**: Zoom SDK customize invite class, partner app can customize the behavior of inviting participant by the class

## Development Description

### Configuration

- iOS version : iOS 7.0 or later
- Support ARC: Yes

### Frameworks

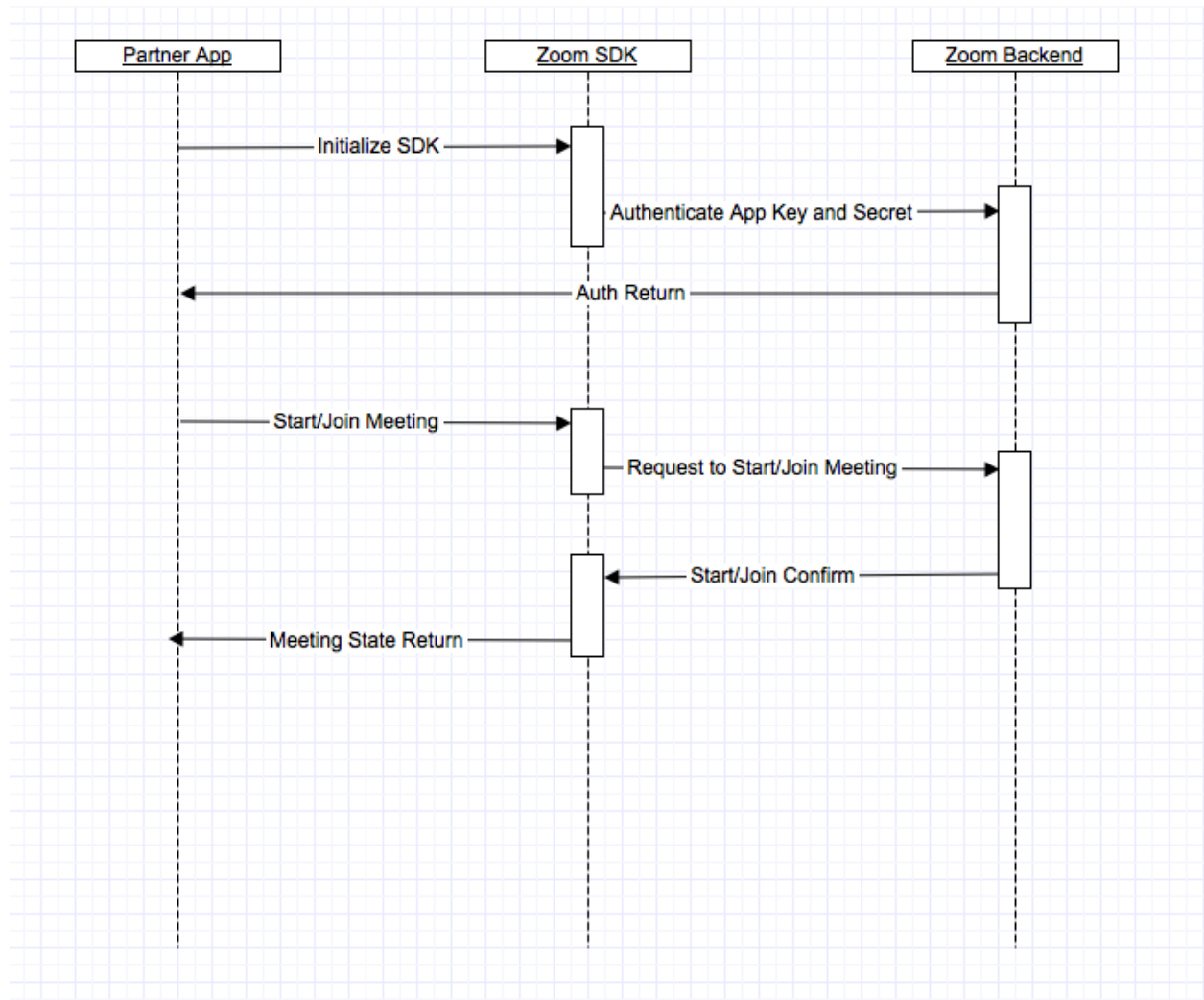
- Zoom SDK provides a ZoomSDK.framework and a ZoomSDKResources.bundle in zoom-ios-sdk/lib directory;

- ZoomSDK.framework should be copied into “Embedded Binaries” firstly, and import ZoomSDK.framework into “Link Binary With Libraries”
- ZoomSDKResources.bundle should be copied into “Copy Bundle Resources”.

## Deployment

- Set “Targeted Device Family” to “iPhone/iPad”;
- Set “iOS Deployment Target” to “iOS 7.0/8.0/9.0”
- Add “-ObjC” in “Other Linker Flags” of Build Settings
- Set “C++ Language Dialect” to “Compile Default”
- Set “C++ Standard Library” to “Compile Default”

## Sequence Diagram



## Code Integration

We provided a project named ZoomSDKSample. For the detailed usage, please refer to the sample project.

### 1. SDK Initialize.

To integrate with Zoom SDK, SDK should be initialized in [application:didFinishLaunchingWithOptions](#): in AppDelegate class, code snippets as following:

```
//1. Set ZoomSDK Domain
[[ZoomSDK sharedSDK] setZoomDomain:@"zoom.us"];

//2. Set ZoomSDK Root Navigation Controller
[[ZoomSDK sharedSDK] setZoomRootController:navController];

//3. ZoomSDK Authenticate
ZoomSDKAuthService *authService = [[ZoomSDK sharedSDK] getAuthService];
if (authService)
{
    authService.delegate = self;

    authService.clientKey = @"xx-xx-xxx";
    authService.clientSecret = @"yy-yy-yyy";

    [authService sdkAuth];
}
```

**Note: The following is optional**

The following code snippet was used if partner want to login/logout with work email account:

```
//Login with work email account
ZoomSDKAuthService *authService = [[ZoomSDK sharedSDK] getAuthService];
if (authService)
{
    BOOL ret = [authService loginWithZoom:@"xx@xx.xx" password:@"yyy"];
}

//Logout
ZoomSDKAuthService *authService = [[ZoomSDK sharedSDK] getAuthService];
if (authService)
{
    [authService logoutZoom];
}
```

## 2. Listener for ZoomSDKAuthDelegate.

To check the authentication from Zoom, the listener can be added in AppDelegate, code snippets as following:

```
#pragma mark - Auth Delegate
```

```
//Sink auth event
```

```

- (void)onZoomSDKAuthReturn:(ZoomSDKAuthError)returnValue
{
    NSLog(@"onZoomSDKAuthReturn %d", returnValue);
}

//Sink login event
- (void)onZoomSDKLoginReturn:(NSInteger)returnValue
{
    NSLog(@"onZoomSDKLoginReturn result=%zd", returnValue);

    //Register delegate of pre-meeting service after login with work email account
    ZoomSDKPremeetingService *service = [[ZoomSDK sharedSDK] getPreMeetingService];
    if (service)
    {
        service.delegate = self;
    }
}

//Sink logout event
- (void)onZoomSDKLogoutReturn:(NSInteger)returnValue
{
    NSLog(@"onZoomSDKLogoutReturn result=%zd", returnValue);
}

```

### 3. Start Meeting

After SDK was initialized, partner app can start a zoom meeting, code snippets as following:

```

ZoomSDKMeetingService *ms = [[ZoomSDK sharedSDK] getMeetingService];
if (ms)
{
    ms.delegate = self;

    //For API User
    NSDictionary *paramDict = @{kMeetingParam_UserID:kSDKUserID,
                                kMeetingParam_UserToken:kSDKUserToken,
                                kMeetingParam_UserType:@(ZoomSDKUserType_ZoomUser),
                                kMeetingParam_Username:kSDKUserName,
                                kMeetingParam_MeetingNumber:kSDKMeetNumber,
                                //kMeetingParam_IsAppShare:@(YES)
                                };

    // //For login user start scheduled meeting
    // NSDictionary *paramDict = @{kMeetingParam_MeetingNumber:kSDKMeetNumber,
    //                               //kMeetingParam_IsAppShare:@(YES)
    //                               };
    //

```

```
// //For login user start instant meeting
// NSDictionary *paramDict = @{
//         //kMeetingParam_IsAppShare:@(YES)
//         };

    NSLog(@"onMeetNow ret:%d", ret);
}
```

## 4. Join Meeting

After SDK was initialized, partner app can join a zoom meeting, code snippets as following:

```
ZoomSDKMeetingService *ms = [[ZoomSDK sharedSDK] getMeetingService];
if (ms)
{
    ms.delegate = self;

    //For Join a meeting with password
    NSDictionary *paramDict = @{
        kMeetingParam_Username:kSDKUserName,
        kMeetingParam_MeetingNumber:meetingNumber,
        kMeetingParam_MeetingPassword:meetingPassword,
        //kMeetingParam_ParticipantID:@"111"
    };

    // //For Join a meeting
    // NSDictionary *paramDict = @{
    //         kMeetingParam_Username:kSDKUserName,
    //         kMeetingParam_MeetingNumber:meetingNumber,
    //         //kMeetingParam_ParticipantID:@"111"
    //         };

    ZoomSDKMeetError ret = [ms joinMeetingWithDictionary:paramDict];
    NSLog(@"onJoinaMeeting ret:%d", ret);
}
```

## 5. Listener for Meeting Service

Once Partner App started/joined a zoom meeting, SDK will return the meeting state by Meeting Service Delegate, code snippets as following:

#pragma mark - Meeting Service Delegate

```
- (void)onMeetingReturn:(ZoomSDKMeetError)error internalError:(NSInteger)internalError
{
    NSLog(@"onMeetingReturn:%d, internalError:%zd", error, internalError);
}
```

```
- (void)onMeetingStateChange:(ZoomSDKMeetingState)state
{
    NSLog(@"onMeetingStateChange:%d", state);
}
```

## 6. Get/Set Meeting Audio/Video State

After SDK was initialized, partner can change the meeting audio/video state, the change will take action in the subsequent meeting, code snippets as following:

```
ZoomSDKMeetingSettings *settings = [[ZoomSDK sharedSDK] getMeetingSettings];
if (!settings)
    return nil;

BOOL isAutoConnected = [settings autoConnectInternetAudio];

BOOL isAudioMuted = [settings muteAudioWhenJoinMeeting];

BOOL isVideoMuted = [settings muteVideoWhenJoinMeeting];

BOOL disabled = [settings driveModeDisabled];

BOOL disabledCallIn = [settings callInDisabled];

BOOL disabledCallOut = [settings callOutDisabled];

//Set Meeting Audio/Video State
UISwitch *sv = (UISwitch*)sender;

[[[ZoomSDK sharedSDK] getMeetingSettings] setAutoConnectInternetAudio:sv.on];

[[[ZoomSDK sharedSDK] getMeetingSettings] setMuteAudioWhenJoinMeeting:sv.on];

[[[ZoomSDK sharedSDK] getMeetingSettings] setMuteVideoWhenJoinMeeting:sv.on];

[[[ZoomSDK sharedSDK] getMeetingSettings] disableDriveMode:sv.on];

[[[ZoomSDK sharedSDK] getMeetingSettings] disableCallIn:sv.on];

[[[ZoomSDK sharedSDK] getMeetingSettings] disableCallOut:sv.on];
```

## 7. Pause/Resume Meeting Audio

After started/joined meeting, partner can pause/resume the meeting audio, code snippets as following:



```
BOOL isNoAudio = [[[ZoomSDK sharedSDK] getMeetingService] isNoMeetingAudio];
[[[ZoomSDK sharedSDK] getMeetingService] pauseMeetingAudio:!isNoAudio];
```

## 8. Customize to invite participant

Once Partner App started/joined a zoom meeting, Partner app can customize the behavior of inviting participant, code snippets as following:

//To enable to customize Invite Participant, Partner should add the following code at first, the “InviteViewController” is the Partner provided view controller.

```
[ZoomSDKInviteHelper sharedInstance].enableInvite = YES;
[ZoomSDKInviteHelper sharedInstance].inviteVCName = @"InviteViewController";
```

And partner can get meeting ID and join meeting URL by the following properties:

```
NSString *meetingID = [ZoomSDKInviteHelper sharedInstance].meetingID;
NSString *meetingURL = [ZoomSDKInviteHelper sharedInstance].joinMeetingURL;
```

## 9. Customize the content of “Invite by Message”, “Copy URL” and “Invite by Email”

Partner App can customize the content of “Invite by Message (SMS)”, “Copy URL” and “Invite by Email” after the meeting is ongoing, and can enable/disable the feature of “Invite by Message”/ “Copy URL”/“Invite by Email”, code snippets as following:

```
//For Enable/Disable Copy URL
[ZoomSDKInviteHelper sharedInstance].disableCopyURL = YES;
```

```
//For Enable/Disable Invite by Email
[ZoomSDKInviteHelper sharedInstance].disableInviteEmail = YES;
```

```
//For Enable/Disable Invite by Message
[ZoomSDKInviteHelper sharedInstance].disableInviteSMS = YES;
```

```
if (state == ZoomSDKMeetingState_InMeeting)
{
    //For customizing the content of Invite by SMS
    NSString *meetingID = [[ZoomSDKInviteHelper sharedInstance] meetingID];
    NSString *smsMessage = [NSString stringWithFormat:NSLocalizedString(@"Please join meeting with ID: %@, @""), meetingID];
    [[ZoomSDKInviteHelper sharedInstance] setInviteSMS:smsMessage];

    //For customizing the content of Copy URL
    NSString *joinURL = [[ZoomSDKInviteHelper sharedInstance] joinMeetingURL];
```

```

        NSString *copyURLMsg = [NSString stringWithFormat:NSLocalizedString(@"Meeting
URL: %@", @""), joinURL];
        [[ZoomSDKInviteHelper sharedInstance] setInviteCopyURL:copyURLMsg];

        //For customizing "Invite by Email"
        [ZoomSDKInviteHelper sharedInstance].inviteEmailSubject = @"Invite by Email";
        [ZoomSDKInviteHelper sharedInstance].inviteEmailContent = [NSString
stringWithFormat:NSLocalizedString(@"Please join meeting with ID: %@", @""), meetingID];
    }

```

## 10. Turn on/off cloud record

```

ZoomSDKMeetingService *ms = [[ZoomSDK sharedInstance] getMeetingService];
//To check whether support cloud record or not
[ms isCMREnabled];

//To check whether cloud record is in progress
[ms isCMRInProgress];

//Turn on cloud record
[ms turnOnCMR:YES];

//Turn off cloud record
[ms turnOnCMR:NO];

```

## 11. Notify app become active/inactive.

Zoom SDK should notify the common layer that app will become inactive in [applicationWillResignActive](#): in AppDelegate class, code snippets as following:

```
[[ZoomSDK sharedInstance] appWillResignActive];
```

Zoom SDK should notify the common layer that app did enter background in [applicationDidEnterBackground](#): in AppDelegate class, code snippets as following:

```
[[ZoomSDK sharedInstance] appDidEnterBackground];
```

Zoom SDK should notify the common layer that app did become active in [applicationDidBecomeActive](#): in AppDelegate class, code snippets as following:

```
[[ZoomSDK sharedInstance] appDidBecomeActive];
```

## 12. Ignore Signal SIGPIPE.

Zoom SDK should ignore Signal “SIGPIPE” to avoid crash when app switch from background to foreground, the code can be added in function `main(int argc, char * argv[])` in main.m file, code snippets as following:

```
struct sigaction sa;
sa.sa_handler = SIG_IGN;
sigaction(SIGPIPE, &sa, NULL);
```

### 13. Pre-Meeting Service

This class provides support for schedule/edit/delete meeting Zoom SDK once login with work email, code snippets as following:

```
//For Schedule Meeting
ZoomSDKPremeetingService *service = [[ZoomSDK sharedSDK] getPreMeetingService];
if (service)
{
    id<ZoomSDKMeetingItem> item = [service createMeetingItem];
    [item setMeetingTopic:@"xxx"];
    [item setStartTime:[NSDate date]];
    [item setTimeZoneID:[NSTimeZone defaultTimeZone].name];
    [item setDurationInMinutes:60];

    [service scheduleMeeting:item];

    [service destroyMeetingItem:item];
}

//For Edit Meeting
ZoomSDKPremeetingService *service = [[ZoomSDK sharedSDK] getPreMeetingService];
if (service)
{
    id<ZoomSDKMeetingItem> item = [service getMeetingItemByNumber:123456789];
    if (item)
    {
        [item setMeetingTopic:@"xxx yyy"];
        [item setMeetingNumber:123456789];
        [item setStartTime:[NSDate date]];
        [item setMeetingPassword:@"yyy"];

        [service editMeeting:item];
    }
}

//For Delete Meeting
ZoomSDKPremeetingService *service = [[ZoomSDK sharedSDK] getPreMeetingService];
if (service)
{
```

```

        id<ZoomSDKMeetingItem> item = [service getMeetingItemByNumber:123456789];
        if (item)
        {
            [service deleteMeeting:item];
        }
    }

//For List Meeting
ZoomSDKPremeetingService *service = [[ZoomSDK sharedSDK] getPreMeetingService];
if (service)
{
    [service listMeeting];
}

```

## 14. Listener for Pre-Meeting Service

Once Schedule/Edit/Delete/List zoom meeting with ZoomSDKPremeetingService, SDK will return the event state by Pre-Meeting Service Delegate, code snippets as following:

```

- (void)sinkSchedultMeeting:(NSInteger)result
{
    NSLog(@"sinkSchedultMeeting result: %zd", result);
}

- (void)sinkEditMeeting:(NSInteger)result
{
    NSLog(@"sinkEditMeeting result: %zd", result);
}

- (void)sinkDeleteMeeting:(NSInteger)result
{
    NSLog(@"sinkDeleteMeeting result: %zd", result);
}

- (void)sinkListMeeting:(NSInteger)result withMeetingItems:(NSArray*)array
{
    NSLog(@"sinkSchedultMeeting result: %zd items: %@", result, array);
}

```

## 15. For App Share

iOS SDK supports to start a meeting with sharing content in App(App Share), and at the same time the meeting UI can be hidden in the background while app sharing.

```

//Once meeting is started, user can sink "onMeetingReady" and start app share in this method
- (void)onMeetingReady
{

```

```

ZoomSDKMeetingService *ms = [[ZoomSDK sharedSDK] getMeetingService];
if ([ms isDirectAppShareMeeting])
{
    if ([ms isStartingShare] || [ms isViewingShare])
    {
        NSLog(@"There exist an ongoing share");
        [ms showZoomMeeting:nil];
        return;
    }

    BOOL ret = [ms startAppShare];
    NSLog(@"Start App Share... ret:%zd", ret);
}
}

//Notify that user can share the splash in this method
- (void)onAppShareSplash
{
    ZoomSDKMeetingService *ms = [[ZoomSDK sharedSDK] getMeetingService];
    if (ms)
    {
        [ms appShareWithView:self.splashVC.view];
    }
}

//Notify that user clicked the share button in meeting UI, user can hide meeting ui and share
his/her content in the App.
- (void)onClickedShareButton
{
    ZoomSDKMeetingService *ms = [[ZoomSDK sharedSDK] getMeetingService];
    if (ms)
    {
        if ([ms isStartingShare] || [ms isViewingShare])
        {
            NSLog(@"There exist an ongoing share");
            return;
        }

        [ms hideZoomMeeting:^(void){
            [ms startAppShare];
        }];
    }
}

//Notify that there exists no ongoing share, user can determine to share his/her content or not.
- (void)onOngoingShareStopped
{
    NSLog(@"There does not exist ongoing share");
}

```

```
// ZoomSDKMeetingService *ms = [[ZoomSDK sharedSDK] getMeetingService];
// if (ms)
// {
//     [ms startAppShare];
// }
}
```

## 16. Customize waiting UI

When user try to join a meeting which the host has not started, the “Waiting for Host” UI can be shown, and user can customize the waiting UI once received the following callback.

```
//This method is optional
- (void)onJBHWaitingWithCmd:(JBHCmd)cmd
{
    switch (cmd) {
        case JBHCmd_Show:
        {
            UIViewController *vc = [UIViewController new];
            //
            NSString *meetingID = [ZoomSDKInviteHelper sharedInstance].meetingID;
            vc.title = meetingID;
            //
            UIBarButtonItem *leaveItem = [[UIBarButtonItem alloc]
initWithTitle:NSLocalizedString(@"Leave", @"") style:UIBarButtonItemStylePlain target:self
action:@selector(onLeave:)];
            [vc.navigationItem setRightBarButtonItem:leaveItem];
            //
            UINavigationController *nav = [[UINavigationController alloc]
initWithRootViewController:vc];
            nav.modalPresentationStyle = UIModalPresentationFormSheet;
            [self presentViewController:nav animated:YES completion:NULL];
        }
        break;

        case JBHCmd_Hide:
        default:
        {
            [self dismissViewControllerAnimated:YES completion:NULL];
        }
        break;
    }
}
```

### Note:

Zoom iOS SDK should be integrated in the main thread.

**For fixing bug that view will move up about 20px after leaving meeting:**

1. Add the following code in method “viewWillAppear” in the view controller which call the interface of starting/joining meeting

```
- (void)viewWillAppear:(BOOL)animated
{
    [super viewWillAppear:animated];

    //for bug that there exist 20 pixels in the bottom while leaving meeting quickly
    [[UIApplication sharedApplication] setStatusBarHidden:YES
withAnimation:UIStatusBarAnimationNone];
    [[UIApplication sharedApplication] setStatusBarHidden:NO
withAnimation:UIStatusBarAnimationNone];
}
```

2. Keep app continues to run in the background to support audio, add flag in project's info.plist:

```
<key>UIBackgroundModes</key>

<array>

    <string>audio</string>

</array>
```

**For hiding warning from iOS SDK:**

Set “Debug Information Format” = “DWARF” in project's build setting.

**For customize string resource in iOS SDK:**

Partner can define the string resource in Localizable.string file, and the key of string should be matched with the one in Zoom SDK. On the other hand, the key and value of defined string should not be the same, or the customized string will not take place the one in Zoom SDK.

We customized some strings in ZoomSDKSample project, as follows:

```
"Waiting..." = "Sample Waiting...";
```

```
"Zoom" = "SDK Sample";
```

```
"Leave" = "Exit";
```

```
"Leave Meeting" = "Exit Meeting";
```

#### **For customize image resource in iOS SDK:**

Partner can define the image resource by their own, and the image file name should be equal to the one in Zoom SDK, which will take place the one in Zoom SDK.

We customized some images in ZoomSDKSample project, the name of image files as follows:

```
icon_meeting_share.png
```

```
icon_meeting_share@2x.png
```

```
icon_meeting_share@3x.png
```

```
icon_meeting_stopshare.png
```

```
icon_meeting_stopshare@2x.png
```

```
icon_meeting_stopshare@3x.png
```