Zoom Windows SDK

Modification History

Revision	Date	Originator	Comments
2	08/05/2016	Zoom custom engineering	Support login zoom user
2	10/05/2016	Zoom custom engineering	fix leave meeting wnd cut off issue
2	11/15/2016	Zoom custom engineering	add new interface for get windows handle fix branding issue in setting dialog, whiteboard and airhost window fix participant id don't take effect when start/join meeting
2	01/20/2017	Zoom custom engineering	Support to join Webinar meeting with as Panelist Support to pin/spotlight video Support to pause/resume screen sharing Support H.323/SIP callout directly Support invite by Phone andCall Medirectly Add watermark - Powered by Zoom Support to start/join meeting without audio Support to start/join meeting without video support Multi-share new interface in ISettingService new interface in IMeetingConfiguration refine IMeetingService interface
2	02/10/2017	Zoom custom engineering	Support waiting room Support embedded browser feature Refine the sdk demo project Bug fix
2	03/02/2017	Zoom custom engineering	Enhancement of proxy Enhancement of SSL cert Verification Video/Audio session connection quality API Bug fix

Zoom Windows SDK

| 2	03/10/2017	Zoom custom engineering	host change callback spotlight video change callback self record privilige change callback Low Or Raise Hand Status change callback Bug fix
2	04/12/2017	Zoom custom engineering	Webinar promote and depromote meeting status change notify Join and leave Breakout room meeting status change notify Show or hide invite button in meeting ui and callback invite button click message support Add new interface for H323 password Bug fix

Note that the windows SDK from Zoom is distributed under a separate SDK agreement. Please make sure that you read the terms and conditions of the SDK agreement before using the SDK.

1.Introduction and Pre-Requisite

In order to use the SDK, you need to make sure that the RETS API and the mobile SDK is enabled in your account. If not, please contact your account rep or contact Zoom support. Our windows SDK is written in C++ - if you are writing a C#/.NET application, you need to make sure that you can call the C++ library functions from within your application space.

Note that our SDK support meeting services and do not support messaging currently.

- Get the SDK key and secret from your zoom account. This key/sec is same as the one used for mobile SDK

Single Sign-On >

REST API >

Mobile SDK >
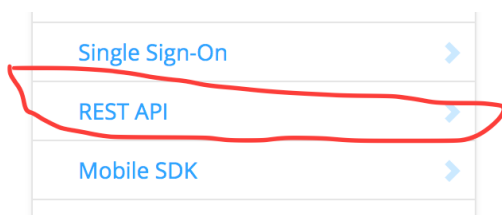
- Get the REST API Key/Sec from your zoom account

Zoom Windows SDK

Credential | Playground

API Key: KuqoKDxjQT2dtdb9Yis47A

API Secret: ************ Show Regenerate

- Call REST API "getbyemail" and this should return the user id and user token

API Playground

API Endpoint: https://api.zoom.us/v1/user/getbyemail

API Key: * KuqoKDxjQT2dtdb9Yis47A

API Secret: * ylQrZ6uPduFwZ2QpPyhtGawPytNqAEdKYdGd

Data Type: ⦿ JSON ◯ XML Clear

User Email Address: * abc@test.us

Login Type: * ⦿ Work Email ◯ Google ◯ Facebook ◯ SSO ◯ API

Send API Request
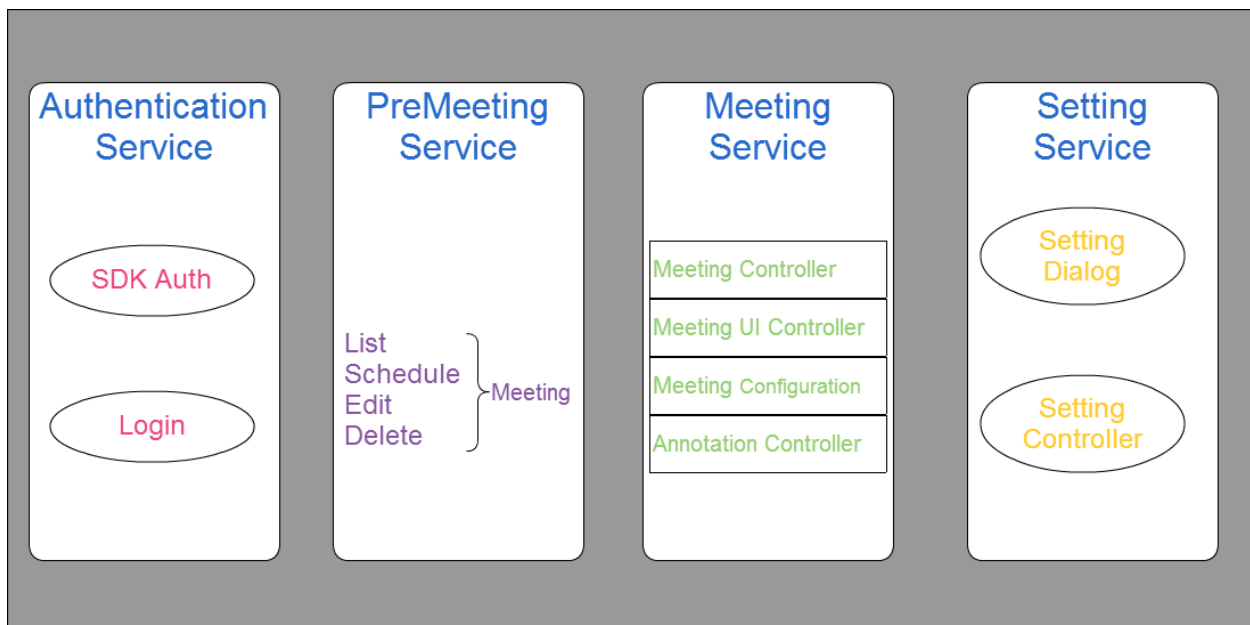
Zoom Windows SDK

Post Data:

```
{
    "id": "LeE4XRa8RkaCxtqgXts3Zw",
    "disable_jbh_reminder": false,
    "enable_cmr": true,
    "enable_auto_recording": true,
    "enable_cloud_auto_recording": t
    "timezone": "America/Los_Angeles",
    "created_at": "2015-07-23T23:40:58Z",
    "token": "mL4KVYD-
8fl2TtnE1nFM5IeEMjNXT4xfgdFj62PmqNg.BgIsb0NFOURZNStNZGJuNXhXSGY5SW
RydGovdCtIdW5oV0tzQ0xwbnBFODRqTT1AN2QxM2I5NWIwZTVlZDliOTE5OTE1ZW
IwNDM4ZTNmMmVlY2E0MmZjZjE5MmUyZDdjNzY0YzQwOWU1NjliMjExYgA"
}
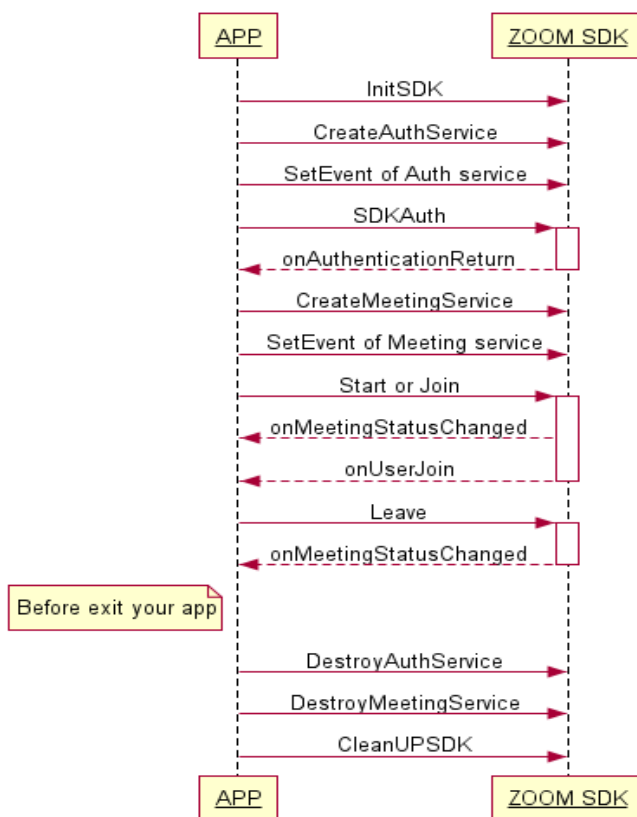```

# 2.SDK Architecture and Workflow

Zoom SDK supports two options for user authentication.

1.       You can pass the user id of the user to the SDK initialization and all the meetings will start on that user's account. This is referred to as API user

2.       There are times it might be cumbersome to get the user id and you might want to distribute your app to all Zoom users – in this case, you can ask the user to enter the Zoom login credentials (username /password) in your app and then pass it to the SDK. This is referred to as normal user

-       API User join/start and leave

## API user Join/start and leave meeting Sequence

| APP | | ZOOM SDK |
|---|---|---|

InitSDK

CreateAuthService

SetEvent of Auth service

SDKAuth

onAuthenticationReturn

CreateMeetingService

SetEvent of Meeting service

Start or Join

onMeetingStatusChanged

onUserJoin

Leave

onMeetingStatusChanged

Before exit your app

DestroyAuthService

DestroyMeetingService

CleanUPSDK

| APP | | ZOOM SDK |
|---|---|---|

- Normal User join/start and leave meeting

Zoom Windows SDK

## Login(Normal) user Join/start and leave meeting Sequence

APP                ZOOM SDK

InitSDK

CreateAuthService

SetEvent of Auth service

SDKAuth

onAuthenticationReturn

Login

onLoginRet

CreateMeetingService

SetEvent of Meeting service

Start or Join

onMeetingStatusChanged

onUserJoin

Leave

onMeetingStatusChanged

Before exit your app

DestroyAuthService

DestroyMeetingService

CleanUPSDK

APP                ZOOM SDK
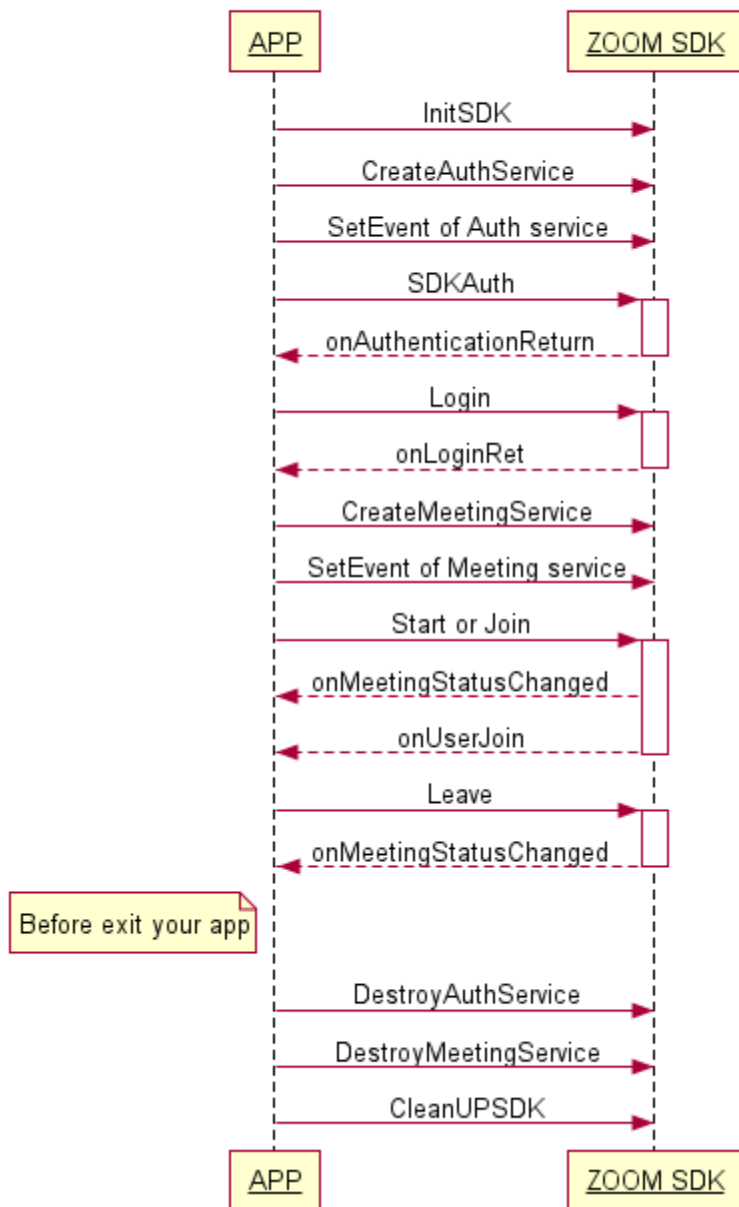
●     Normal User list/schedule/edit/delete meeting

Zoom Windows SDK

## Normal user schedule list edit delete meeting Sequence

```
        APP   ZOOM SDK
         │   InitSDK   │
         │─────────────────────────────▶│
         │   CreateAuthService   │
         │─────────────────────────────▶│
         │   SetEvent of Auth service   │
         │─────────────────────────────▶│
         │   SDKAuth   │
         │─────────────────────────────▶│
         │   onAuthenticationReturn   │
         │◀ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ │
         │   Login   │
         │─────────────────────────────▶│
         │   onLoginRet   │
         │◀ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ │
         │   CreatePreMeetingService   │
         │─────────────────────────────▶│
         │ SetEvent of PreMeeting service│
         │─────────────────────────────▶│
         │   ScheduleMeeting   │
         │─────────────────────────────▶│
         │   onScheduleOrEditMeeting   │
         │◀ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ │
         │   EditMeeting   │
         │─────────────────────────────▶│
         │   onScheduleOrEditMeeting   │
         │◀ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ │
         │   ListMeeting   │
         │─────────────────────────────▶│
         │   onListMeeting   │
         │◀ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ │
         │   DeleteMeeting   │
         │─────────────────────────────▶│
         │   onDeleteMeeting   │
         │◀ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ │
 Before exit your app
         │   DestroyAuthService   │
         │─────────────────────────────▶│
         │   DestroyPreMeetingService   │
         │─────────────────────────────▶│
         │   CleanUPSDK   │
         │─────────────────────────────▶│
        APP   ZOOM SDK
```
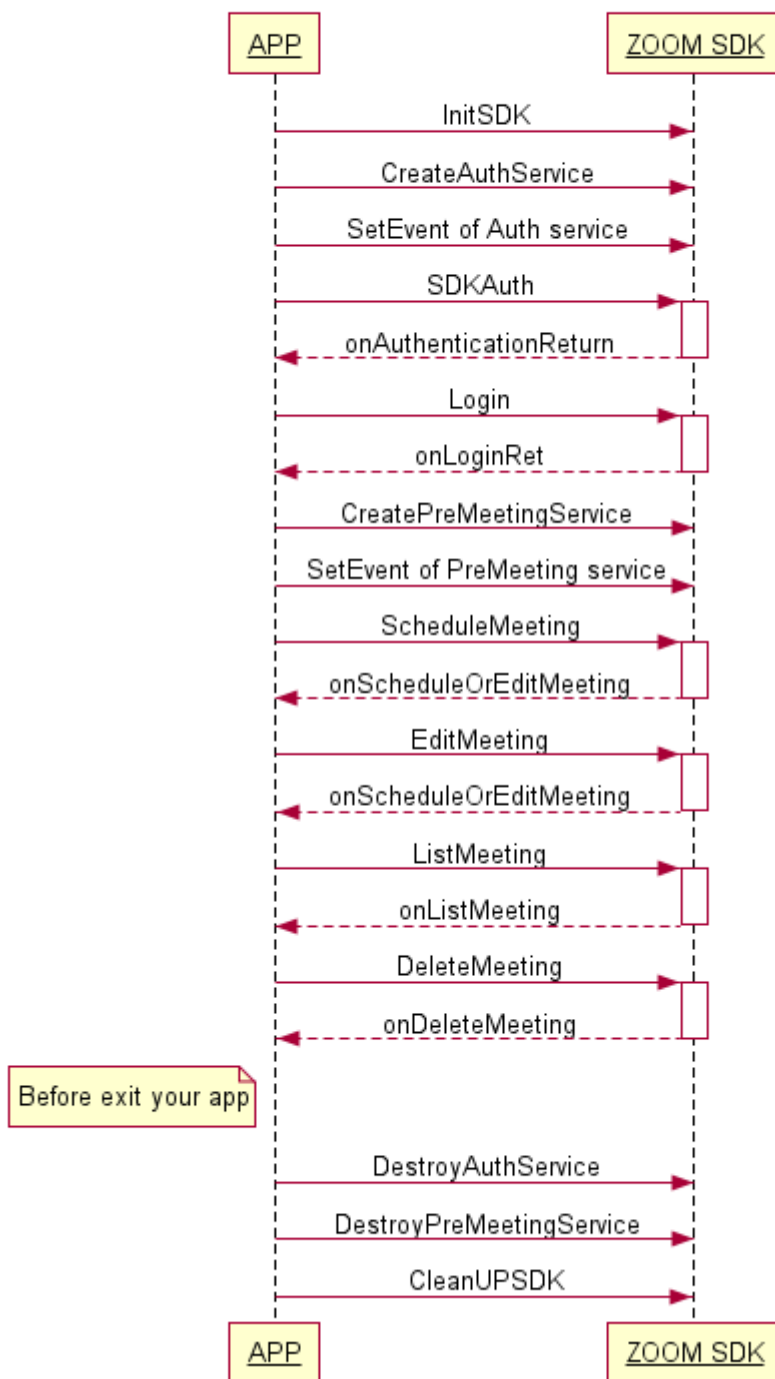
# 3.SDK Functions

## 3.1. Initialization

You can find these APIs in zoom_sdk.h.　Please call zoom sdk APIs in main ui thread for thread safe operation.

Initialize zoom sdk:

```
1.   //Initialization code
2.   ZOOM_SDK_NAMESPACE::InitParam param_;
3.   param_.strWebDomain = _T("https://zoom.us/");
4.   aram_.emLanguageID = ZOOM_SDK_NAMESPACE::LANGUAGE_English;//ZOOM_SDK_NAMESPACE::SDK_LANGUAGE_ID
5.   //branding, if you don't config, use zoom as defeault
6.   param_.strBrandingName = "Zoom";
7.   param_.strSupportUrl = "https://support.zoom.us/";
8.   param_.hResInstance = your resource module handle;
9.   param_.uiWindowIconBigID = your windows title big icon id;
10.  param_.uiWindowIconBigID = your windows title samll icon id;
11.  ZOOM_SDK_NAMESPACE::SDKError err = ZOOM_SDK_NAMESPACE::InitSDK(param_);
12.  switch (err)
13.  {
14.  case ZOOM_SDK_NAMESPACE::SDKERR_SUCCESS:
15.      {
16.      //success
17.      }
18.      break;
19.  default:
20.      {
21.      //error handle
22.      }
23.      break;
24.  }
```

Cleanup zoom sdk before you exit your application.

```
1.   //Cleanup code
2.   ZOOM_SDK_NAMESPACE::CleanUPSDK();
```

3.2. Authentication Service

You should do authentication first when you start to use zoom sdk and include auth_service_interface.h

- 　　　　Create authentication service

Zoom Windows SDK

```
1.   //Create Auth Service
2.   ZOOM_SDK_NAMESPACE::IAuthService* pAuthService(NULL);
3.   ZOOM_SDK_NAMESPACE::IAuthServiceEvent* pAuthServiceEvent;//implement yourself
4.   ZOOM_SDK_NAMESPACE::SDKError err = ZOOM_SDK_NAMESPACE::CreateAuthService(&pAuthService);
5.   if (ZOOM_SDK_NAMESPACE::SDKERR_SUCCESS == err && pAuthService)
6.   {
7.       pAuthService->SetEvent(pAuthServiceEvent);
8.       //do you logic
9.   }
```

- Auth SDK

```
1.   //Auth sdk
2.   ZOOM_SDK_NAMESPACE::AuthParam authParam;
3.   authParam.appKey = L"your app key";
4.   authParam.appSecret = L"your app secret";
5.   pAuthService->SDKAuth(authParam);
6.
7.   //Auth sdk callback
8.   //implement yourself
9.   virtual void IAuthServiceEvent::onAuthenticationReturn(AuthResult ret) = 0;
```

- Login end user

```
1.   //Login end user
2.   ZOOM_SDK_NAMESPACE::LoginParam param_;
3.   param_.bRememberMe = true or false;
4.   param_.userName = L"your account";
5.   param_.password = L"your password";
6.   pAuthService->Login(param_);
7.
8.   //Login callback
9.   //implement yourself
10.  virtual void IAuthServiceEvent::onLoginRet(LOGINSTATUS ret, IAccountInfo* pAccountInfo) = 0;
```

- Logout end user

```
1.   //logout end user
2.   pAuthService->LogOut();
3.
4.   //Logout callback
5.   //implement yourself
6.   virtual void IAuthServiceEvent::onLogout() = 0;
```

- Destroy auth service

```
1.   //Destroy auth service
2.   ZOOM_SDK_NAMESPACE::DestroyAuthService(pAuthService);
```

Zoom Windows SDK

3.3. Pre-meeting Service

If you want to use this service, include premeeting_service_interface.h.

- ● Create Pre-Meeting Service:

```
1.   //Create premeeting service
2.   ZOOM_SDK_NAMESPACE::IPreMeetingService* pPreMeetingService(NULL);
3.   ZOOM_SDK_NAMESPACE::IPreMeetingServiceEvent* pPreMeetingServiceEvent(NULL);//implement yourself
4.   ZOOM_SDK_NAMESPACE::SDKError err = ZOOM_SDK_NAMESPACE::CreatePreMeetingService(&pPreMeetingService);
5.   if (ZOOM_SDK_NAMESPACE::SDKERR_SUCCESS == err && pPreMeetingService)
6.   {
7.       pPreMeetingService->SetEvent(pPreMeetingServiceEvent);
8.       //do your logic
9.   }
```

- ● Schedule Meeting

```
1.   //Schedule meeting
2.   IScheduleMeetingItem* pScheduleMeetingItem = ZOOM_SDK_NAMESPACE::IPreMeetingService::CreateScheduleMee
     tingItem();
3.   if (pScheduleMeetingItem)
4.   {
5.       //optional
6.       pScheduleMeetingItem->SetAllowJoinBeforeHost(true);
7.       pScheduleMeetingItem->SetMeetingPassword(password);
8.       pScheduleMeetingItem->SetUsePMIAsMeetingID(true);
9.       pScheduleMeetingItem->TurnOffVideoForHost(true);
10.      pScheduleMeetingItem->TurnOffVideoForAttendee(true);
11.      //
12.
13.      pScheduleMeetingItem->SetStartTime(starttime)
14.      pScheduleMeetingItem->SetDurationInMinutes(duration);
15.      pScheduleMeetingItem->SetMeetingTopic(topic);
16.      pPreMeetingService->ScheduleMeeting(pScheduleMeetingItem);
17.      ZOOM_SDK_NAMESPACE::IPreMeetingService::DestoryScheduleMeetingItem(pScheduleMeetingItem);
18.  }
19.
20.  //Schedule meeting callback
21.  //implement yourself
22.  IPreMeetingServiceEvent
23.  virtual void onScheduleOrEditMeeting(PremeetingAPIResult result, UINT64 meetingUniqueID) = 0;
```

- ● Edit Meeting

Zoom Windows SDK

```
1.   //Edit meeting
2.   IScheduleMeetingItem* pEditMeetingItem = ZOOM_SDK_NAMESPACE::IPreMeetingService::CreateScheduleMeeting
     Item();
3.   if (pEditMeetingItem)
4.   {
5.       //optional
6.       pEditMeetingItem->SetAllowJoinBeforeHost(true);
7.       pEditMeetingItem->SetMeetingPassword(password);
8.       pEditMeetingItem->SetUsePMIAsMeetingID(true);
9.       pEditMeetingItem->TurnOffVideoForHost(true);
10.      pEditMeetingItem->TurnOffVideoForAttendee(true);
11.      //
12.
13.      pEditMeetingItem->SetStartTime(starttime)
14.      pEditMeetingItem->SetDurationInMinutes(duration);
15.      pEditMeetingItem->SetMeetingTopic(topic);
16.      pPreMeetingService->EditMeeting(meetingUniqueID, pEditMeetingItem);
17.      ZOOM_SDK_NAMESPACE::IPreMeetingService::DestoryScheduleMeetingItem(pEditMeetingItem);
18.  }
19.
20.  //Edit meeting callback
21.  //implement yourself
22.  IPreMeetingServiceEvent
23.  virtual void onScheduleOrEditMeeting(PremeetingAPIResult result, UINT64 meetingUniqueID) = 0;
```

- Delete Meeting

```
1.   //Delete meeting
2.   pPreMeetingService->DeleteMeeting(meetingUniqueID);
3.   //Delete meeting callback
4.   //implement yourself
5.   IPreMeetingServiceEvent
6.   virtual void onDeleteMeeting(PremeetingAPIResult result) = 0;
```

- List Meeting

```
1.   //List meeting
2.   pPreMeetingService->ListMeeting();
3.   //List meeting callback
4.   //implement yourself
5.   IPreMeetingServiceEvent
6.   virtual void onListMeeting(PremeetingAPIResult result, IList<UINT64 >* lstMeetingList) = 0;
```

- Get meeting item by ID

Refer the header file for the details.

- Destroy Pre-Meeting Service

```
1.   //Destroy premeeting service
2.   ZOOM_SDK_NAMESPACE::DestroyPreMeetingService(pPreMeetingService);
```

3.4. Meeting Service

If you want to use this service, include meeting_service_interface.h.

- **Create meeting service**

```
1.   //Create meeting service
2.   ZOOM_SDK_NAMESPACE::IMeetingService* pMeetingService(NULL);
3.   ZOOM_SDK_NAMESPACE::IMeetingServiceEvent* pMeetingServiceEvent;//implement yourself
4.   ZOOM_SDK_NAMESPACE::SDKError err = ZOOM_SDK_NAMESPACE::CreateMeetingService(&pMeetingService);
5.   if (ZOOM_SDK_NAMESPACE::SDKERR_SUCCESS == err && pMeetingService)
6.   {
7.       pMeetingService->SetEvent(pMeetingServiceEvent);
8.       //do your logic
9.   }
```

- **Destroy meeting service**

```
1.   //Destroy meeting service
2.   ZOOM_SDK_NAMESPACE::DestroyMeetingService(pMeetingService);
```

3.4.1.   Meeting controller

- Start meeting for API user

```
1.   //Start meeting for api user
2.   ZOOM_SDK_NAMESPACE::StartParam startParam;
3.   startParam.userType = ZOOM_SDK_NAMESPACE::SDK_UT_APIUSER;
4.   ZOOM_SDK_NAMESPACE::StartParam4APIUser& apiuserParam = startParam.param.apiuserStart;
5.   apiuserParam.userID = L"your user id, get this via RestAPI";
6.   apiuserParam.userToken = L"your user id, get this via RestAPI";
7.   apiuserParam.userName = L"your display name";
8.   apiuserParam.meetingNumber = 123456789;///you can schedule meeting via RestAPI
9.   apiuserParam.isDirectShareDesktop = false;///direct share desktop or not when you start meeting
10.  apiuserParam.hDirectShareAppWnd = NULL;///direct share window or not when you start meeting
11.  pMeetingService->Start(startParam);
12.
13.  //Start meeting callback
14.  //implement yourself, for more details please read comments of IMeetingServiceEvent in head file
15.  IMeetingServiceEvent
16.  virtual void onMeetingStatusChanged(MeetingStatus status, int iResult = 0) = 0;
17.  virtual void onUserJoin(IList<unsigned int >* lstUserID, const wchar_t* strUserList = NULL) = 0;
18.  virtual void onUserAudioStatusChange(IList<IUserAudioStatus* >* lstAudioStatusChange, const wchar_t* s
     trAudioStatusList = NULL) = 0;
```

Zoom Windows SDK

- ● **Start meeting for End user**

```
1.   //Start meeting for end user
2.   ZOOM_SDK_NAMESPACE::StartParam startParam;
3.   startParam.userType = ZOOM_SDK_NAMESPACE::SDK_UT_NORMALUSER;
4.   ZOOM_SDK_NAMESPACE::StartParam4NormalUser& normalParam = startParam.param.normaluserStart;
5.   normalParam.meetingNumber = 0;//your meeting number or 0 for instance meeting
6.   normalParam.isDirectShareDesktop = false;///direct share desktop or not when you start meeting
7.   normalParam.hDirectShareAppWnd = NULL;///direct share window or not when you start meeting
8.   pMeetingService->Start(startParam);
9.
10.  //Start meeting callback
11.  //implement yourself, for more details please read comments of IMeetingServiceEvent in head file
12.  IMeetingServiceEvent
13.  virtual void onMeetingStatusChanged(MeetingStatus status, int iResult = 0) = 0;
14.  virtual void onUserJoin(IList<unsigned int >* lstUserID, const wchar_t* strUserList = NULL) = 0;
15.  virtual void onUserAudioStatusChange(IList<IUserAudioStatus* >* lstAudioStatusChange, const wchar_t* s
     trAudioStatusList = NULL) = 0;
```

- ● **Join Meeting for API user**

```
1.   //Join meeting for api user
2.   ZOOM_SDK_NAMESPACE::JoinParam joinParam;
3.   joinParam.userType = ZOOM_SDK_NAMESPACE::SDK_UT_APIUSER;
4.   ZOOM_SDK_NAMESPACE::JoinParam4APIUser& apiParam = joinParam.param.apiuserJoin;
5.   apiParam.meetingNumber = 123456789;//meeting number you want to join
6.   apiParam.userName = L"your display name";
7.   apiParam.psw = L"Meeting's password";
8.   apiParam.isDirectShareDesktop = false;///direct share desktop or not when you start meeting
9.   apiParam.hDirectShareAppWnd = NULL;///direct share window or not when you start meeting
10.  apiParam.toke4enfrocelogin = L"token for enforce login meeting";//if the meeting is just for signed us
     er.
11.  apiParam.participantId = L"participant Id";///for meeting participant report list, need web backend en
     able.
12.  pMeetingService->Join(joinParam);
13.
14.  //join meeting callback
15.  //implement yourself, for more details please read comments of IMeetingServiceEvent in head file
16.  IMeetingServiceEvent
17.  virtual void onMeetingStatusChanged(MeetingStatus status, int iResult = 0) = 0;
18.  virtual void onUserJoin(IList<unsigned int >* lstUserID, const wchar_t* strUserList = NULL) = 0;
19.  virtual void onUserAudioStatusChange(IList<IUserAudioStatus* >* lstAudioStatusChange, const wchar_t* s
     trAudioStatusList = NULL) = 0;
```

- ● **Join meeting for end user**

```
1.   //Join meeting for end user
2.   ZOOM_SDK_NAMESPACE::JoinParam joinParam;
3.   joinParam.userType = ZOOM_SDK_NAMESPACE::SDK_UT_NORMALUSER;
4.   ZOOM_SDK_NAMESPACE::JoinParam4NormalUser& normalParam = joinParam.param.normaluserJoin;
5.   normalParam.meetingNumber = 123456789;//meeting number you want to join
6.   normalParam.userName = L"your display name";
7.   normalParam.psw = L"Meeting's password";
8.   normalParam.isDirectShareDesktop = false;///direct share desktop or not when you start meeting
9.   normalParam.hDirectShareAppWnd = NULL;///direct share window or not when you start meeting
10.  pMeetingService->Join(joinParam);
11.
12.  //join meeting callback
13.  //implement yourself, for more details please read comments of IMeetingServiceEvent in head file
14.  IMeetingServiceEvent
15.  virtual void onMeetingStatusChanged(MeetingStatus status, int iResult = 0) = 0;
16.  virtual void onUserJoin(IList<unsigned int >* lstUserID, const wchar_t* strUserList = NULL) = 0;
17.  virtual void onUserAudioStatusChange(IList<IUserAudioStatus* >* lstAudioStatusChange, const wchar_t* s
     trAudioStatusList = NULL) = 0;
```

- Other interfaces

Please refer the Zoom SDK.chm

3.4.2. Meeting UI controller

You can get this controller after you have been in the meeting. If not, you can't use this controller.

- Usage

```
1.   //How to user meeting ui controller
2.   ZOOM_SDK_NAMESPACE::IMeetingUIController* pMeetingUIController = pMeetingService->GetUIController();
3.   if (pMeetingUIController)
4.   {
5.       //control meeting ui, such as
6.       pMeetingUIController->EnterFullScreen(true, false); //Enter full screen mode
7.       pMeetingUIController->ExitFullScreen(true, false);//Exit full screen mode
8.       pMeetingUIController->ShowChatDlg(param_);//show chat
9.       pMeetingUIController->SwitchToVideoWall();//switch to video wall mode
10.      //for more details, you can read the IMeetingUIController comments or Zoom SDK.chm
11.  }
```

3.4.3. Meeting Configuration controller

You can set configuration before you call start/join meeting API. And if the meeting is ended or left, the configuration will be reset automatically.

- Usage

Zoom Windows SDK

```
1.   //How to user meeting configuration
2.   ZOOM_SDK_NAMESPACE::IMeetingConfiguration* pMeetingConfig = pMeetingService->GetMeetingConfiguratio
     n();
3.   if (pMeetingConfig)
4.   {
5.       //popup wrong password error dialog or not when join meeting.
6.       pMeetingConfig->DisablePopupMeetingWrongPSWDlg(true);
7.       //don't show zoom meeting id in meeting ui title
8.       pMeetingConfig->HideMeetingInfoFromMeetingUITitle(true);
9.       //popup waiting for host join dialog or not when join meeting.
10.      pMeetingConfig->DisableWaitingForHostDialog(true);
11.      //for more details, you can read the IMeetingConfiguration comments in head file or Zoom SDK.chm
12.  }
```

3.4.4.　　Meeting Annotation controller

You can get this controller when someone or you are sharing. If not, you can't use this controller.

- Usage

```
1.   //how to use the meeting annotation
2.   ZOOM_SDK_NAMESPACE::IAnnotationController* pAnnoCtrl = pMeetingService->GetAnnotationController();
3.   if (pAnnoCtrl)
4.   {
5.       //show annotation bar
6.       pAnnoCtrl->StartAnnotation(ZOOM_SDK_NAMESPACE::SDK_FIRST_VIEW, left, top);
7.       //set color of tools
8.       pAnnoCtrl->SetColor(ZOOM_SDK_NAMESPACE::SDK_FIRST_VIEW, color);
9.       //set tool of annotation
10.      pAnnoCtrl->SetTool(ZOOM_SDK_NAMESPACE::SDK_FIRST_VIEW, type);
11.      //for more details you can read the IAnnotationController comments in head file or zoom_sdk.chm
12.  }
```

- Meeting Event

All of meeting event define at IMeetingServiceEvent. You can refer the header file or Zoom SDK.chm

3.5. Setting Service

If you want to user this service, include setting_service_interface.h.

- Usage

Zoom Windows SDK

```
1.   //How to use setting service
2.   ZOOM_SDK_NAMESPACE::ISettingService* pSettingService(NULL);
3.   ZOOM_SDK_NAMESPACE::SDKError err = ZOOM_SDK_NAMESPACE::CreateSettingService(&pSettingService);
4.   if (ZOOM_SDK_NAMESPACE::SDKERR_SUCCESS == err && pSettingService)
5.   {
6.       pSettingService->ShowSettingDlg(param_);//show zoom setting dialog
7.       pSettingService->SetRecordingPath(path);//set record path
8.       pSettingService->EnableAutoJoinAudio(true);//auto join audio when you join/start meeting
9.       //other interface, you can read the ISettingService comments in head file or Zoom SDK.chm
10.  }
11.
12.  //Destroy setting service
13.  ZOOM_SDK_NAMESPACE::DestroySettingService(pSettingService);
```