

Meeing SDK(Web)说明文档

1 集成 Meeing SDK

```
package  
  
{  
  
    /**  
     * Dispatched when the connect server Success.  
     */  
  
    [Event(name="ConnectServerSuccess", type="MeetingSDK.ConnectServerEvent")]  
  
    /**  
     * Dispatched when the connect server Closed.  
     */  
  
    [Event(name="ConnectServerClosed", type="MeetingSDK.ConnectServerEvent")]  
  
    /**  
     * Dispatched when the connect server Failed.  
     */  
  
    [Event(name="ConnectServerFailed", type="MeetingSDK.ConnectServerEvent")]  
  
    /**  
     * Dispatched when the connect server Rejected.  
     */  
  
    [Event(name="ConnectServerRejected", type="MeetingSDK.ConnectServerEvent")]
```

```
/**
 *
 * @author whw
 *
 */

public class AVSession extends EventDispatcher
{
    public function AVSession(listener:ClientListener)
    {
    }

    /**
     *SDK
     * @param clientCode
     * @param key
     * @return
     *
     */

    public function Init(clientCode:String,key:String):Boolean
    {}

    /**
     * 链接服务器
```

* @param ip webservice ip 地址 192.168.0.121 不需要带 http 头

* @param port webservice 端口 80

* @param roomid 要加入的会议号

* @param nickname 进入会议后显示的名称

* @param usertype 进入会议的角色 0: 普通用户 1:主席 2:直播用户或者旁听用户

* @param thirdUserId 第三方调用时 可传 自己的用户 id 在用户属性

m_BuddyID 字段

*

*/

public function

EnterMeeting(ip:String, tcpport:int, udpport:int, roomid:String, nickname:String, usertype:int, t

thirdUserId:uint):void

{}

/**

*断开服务器

*

*/

public function LeaveMeeting():void

{}

/**

* 设置服务器分配的 id 当收到 ClientFunc_EnablePresence 回调后 取到

m_MeetBuddyID 需调用此方法

```
* @param muserid
```

```
*
```

```
*/
```

```
public function setMyPeerID(muserid:uint):void
```

```
{
```

```
public function getMyPeerID():uint
```

```
{
```

```
/**
```

```
*
```

```
* @return
```

```
*
```

```
*/
```

```
public function getVideo():CCamera
```

```
{
```

```
/**
```

```
*
```

```
* @return
```

```
*
```

```
*/
```

```
public function getAudio():CMicrophone
```

```
{
```

```
/**
```

* 播放某个用户视频

* @param muserid mediaserver 为用户分配的 id 会在

ClientFunc_EnablePresence 回调函数中拿到 key 为 m_MeetBuddyID

* @param videoid 非多流时 传 0 多流时传具体的视频 id

* @param window 视频播放的窗口

*

*/

```
public function PlayVideo(muserid:uint,videoid:int>window:Video):void
```

```
}
```

```
/**
```

* 取消播放某个用户视频

* @param muserid mediaserver 为用户分配的 id 会在

ClientFunc_EnablePresence 回调函数中拿到 key 为 m_MeetBuddyID

* @param videoid 非多流时 传 0 多流时传具体的视频 id

*

*/

```
public function UnPlayVideo(muserid:uint,videoid:int):void
```

```
}
```

```
/**
```

*

* @param host

* @param window

```

*
*/

public function PlayScreen(host:uint,window:Video):void

{}

/**
*
* @param host
*
*/

public function UnplayScreen(host:uint):void

{}

/**
* 发布自己的视频

* @param muserid 传自己的 id mediaserver 为用户分配的 id 会在
ClientFunc_EnablePresence 回调函数中拿到 key 为 m_MeetBuddyID

* @param videoid 非多流时 传 0 多流时传具体的视频 id
*
*/

public function PublishVideo(videoid:int):void

{}

/**
* 取消发布自己的视频

```

* @param muserid 传自己的 id mediaserver 为用户分配的 id 会在

ClientFunc_EnablePresence 回调函数中拿到 key 为 m_MeetBuddyID

* @param videoid 非多流时 传 0 多流时传具体的视频 id

*

*/

```
public function UnPublishVideo(videoid:int):void
```

```
{}
```

```
/**
```

* 播放某个用户音频

* @param muserid mediaserver 为用户分配的 id 会在

ClientFunc_EnablePresence 回调函数中拿到 key 为 m_MeetBuddyID

*

*/

```
public function PlayAudio(muserid:uint):void
```

```
{}
```

```
/**
```

* 取消播放某个用户音频

* @param muserid mediaserver 为用户分配的 id 会在

ClientFunc_EnablePresence 回调函数中拿到 key 为 m_MeetBuddyID

*

*/

```
public function UnPlayAudio(muserid:uint):void
```

```
}

/**
 * 发布自己的音频
 *
 */

public function PublishAudio():void

}

/**
 * 取消发布自己的音频
 *
 */

public function UnPublishAudio(muserid:uint):void

}

/**
 *
 * @param funcName
 * @param toid
 * @param obj
 *
 */

public function CallClientFunction(funcName:String,toid:uint,obj:Object):void

}
```



```
/**
```

```
*
```

```
* @param proKey
```

```
* @param toid
```

```
* @param obj
```

```
*
```

```
*/
```

```
public function SetClientProperty(proKey:String,obj:Object,toid:uint):void
```

```
}
```

```
/**
```

```
*
```

```
* @param name
```

```
* @param toID
```

```
* @param associatedUserID
```

```
* @param body
```

```
* @param id
```

```
* @param associatedMsgID
```

```
*
```

```
*/
```

```
public function publishMessage(name:String, toID:uint=0,
```

```
associatedUserID:int=0, body:Object=null, id:String="", associatedMsgID:String=""):void
```

```
}
```

```
/**
```

```
*
```

```
* @param name
```

```
* @param toID
```

```
* @param associatedUserID
```

```
* @param body
```

```
* @param id
```

```
* @param associatedMsgID
```

```
*
```

```
*/
```

```
public function deleteMessage(name:String, toID:uint=0, associatedUserID:int=0,
```

```
body:Object=null, id:String="", associatedMsgID:String=""):void
```

```
}
```

```
/**
```

```
*
```

```
* @param text
```

```
* @param toid
```

```
* @param textFromat
```

```
*
```

```
*/
```

```
        public function sendTextMessage(text:String,toid:uint,textFromat:Object):void
        {
        }
    }
}
```

```
package
```

```
{
```

```
    public interface CMicrophone
```

```
    {
```

```
        function getMicCount():int;
```

```
        function getMicNameByIndex(index:int):String;
```

```
        function getMicList():Array;
```

```
        function setMicrophone(index:int):void;
```

```
    }
```

```
}
```

```
package
```

```
{
```

```
    public interface CCamera
```

```
    {
```

```
        function getCameraCount():int;
```

```
        function getCameraNameByIndex(index:int):String;
```

```
        /**
```

```
        *
```

```
* return [{DeviceIndex:0,DeviceName:"name"},{},{}.....]
```

```
*/
```

```
function getCameraList():Array;
```

```
function setMode(index:int,w:int,h:int):void
```

```
}
```

```
}
```