

IM SDK for HTML5 使用指南

初始化SDK

功能：始化IM SDK 原型：

```
g_yimInstance = new yim.getInstance(dirtyWordsArray, appKey, events);
```

- array **dirtyWordsArray** - 脏字过滤数组，如['色情', '赌博']
- string **appKey** - 在 www.youme.im 注册后获得的app标识
- object **events** - {onLogin: callback, onLogout: callback} 注册相关事件回调
 - **onLogin(YIMErrorcodeOC, event)** - 登录回调
 - **YIMErrorcodeOC**: 错误码，详细描述见错误码定义
 - **event**: 回调事件
 - **code**: 状态，同YIMErrorcodeOC
 - **message**: 相关信息
 - **data**: {userid: 由调用者分配，不可为空字符串，只可由字母或数字或下划线组成}
 - **onLogout(YIMErrorcodeOC, event)** - 登出回调
 - **YIMErrorcodeOC**: 同上
 - **event**: 同上
 - **onJoinChatRoom(YIMErrorcodeOC, event)** - 加入频道回调
 - **YIMErrorcodeOC**: 同上
 - **event**: 回调事件
 - **code**: 状态，同YIMErrorcodeOC
 - **message**: 相关信息
 - **data**: {roomid: 请求加入的频道ID}
 - **onLeaveChatRoom(YIMErrorcodeOC, event)** - 离开频道回调
 - **YIMErrorcodeOC**: 同上
 - **event**: 同上
 - **onSendMessageStatus(YIMErrorcodeOC, event)** - 消息发送状态回调
 - **YIMErrorcodeOC**: 同上
 - **event**: 回调事件
 - **code**: 状态，同YIMErrorcodeOC
 - **message**: 相关信息
 - **data**: {msgid: 消息ID}

- **onVoiceMsgSend(YIMErrorcodeOC, audioInfo)** - 语音消息录制成功发送前接口回调
 - **YIMErrorcodeOC**: 同上
 - **audioInfo**: 语音消息相关信息，方便渲染语音UI表现
 - **strMessageID**: 消息ID
 - **strVoiceID**: 语音ID
 - **intVoiceTime**: 语音时长
- **onRecvMessage(message)**
 - **object message** 消息对象，存储如下字段
 - **intMessageType**: 消息类型
 - **strSenderID**: 消息发送者ID
 - **strRecvID**: 消息接收者ID
 - **strMessageID**: 消息ID
 - **intCreateTime**: 消息创建时间
 - **chatType**: 聊天类型，群聊/私聊
 - **strVoiceID**: 语音消息VoiceID
- **onKickOff** 无任何参数

登录接口

1.登录

功能：指定用户ID登录IM系统 原型：

```
g_yimInstance.login(strYouMeID, strPasswd);
```

- **string strYouMeID**: 由调用者分配，不可为空字符串，只可由字母或数字或下划线组成
- **string strPasswd**: 从服务端获取到的token，请参考《游密后台服务接口文档》->添加用户

返回：错误码，详细描述见错误码定义 备注：这是一个异步操作，操作结果会通过回调接口返回:onLogin

2.登出

功能：登出游密IM云服务器 原型：

```
g_yimInstance.logout();
```

备注：这是一个异步操作，操作结果会通过回调接口返回:onLogout

聊天室接口

1.加入聊天室

功能：加入聊天室进行群组聊天 原型：

```
g_yimInstance.joinRoom(strRoomID);
```

- string **strRoomID**: 请求加入的频道ID

备注：这是一个异步操作，操作结果会通过回调接口返回: `onJoinChatRoom`

2.退出聊天室

功能：退出聊天室 原型：

```
g_yimInstance.leaveRoom(strRoomID);
```

- string **strRoomID**: 请求加入的频道ID

备注：这是一个异步操作，操作结果会通过回调接口返回: `onLeaveChatRoom`

消息接口

1、发送文本消息

功能：发送文本消息到指定接收者 原型：

```
g_yimInstance.sendTextMessage(strRecvID, chatType, strContent);
```

- string **strRecvID**: 接收者ID（用户ID或者群ID）
- int **chatType**: 聊天类型，群聊/私聊
- string **strContent**: 聊天内容，可以用json格式来扩展消息内容

返回：消息ID，与消息状态回调接口的**strMessageID**对应 备注：这是一个异步操作，操作结果会通过回调接口返回， `onSendMessageStatus`

2、发送语音消息

目前语音消息只支持微信内发送，可以通过执行 `yim.isWeixin()` 判断是否是在微信内，再启用语音发送按键

在发送语音之前，需要初始化微信鉴权，示例：先加载微信api js: <http://res.wx.qq.com/open/js/jweixin-1.2.0.js>

```
//微信鉴权，参考：https://mp.weixin.qq.com/wiki  
ajaxGetSignatureWechat(function(weixinConfig){  
    wx.config({  
        debug: false,
```

```

        appId: weixinConfig.appid,
        timestamp: weixinConfig.time,
        nonceStr: weixinConfig.nonce,
        signature: weixinConfig.sign,
        jsApiList:
        ['checkJsApi', 'startRecord', 'stopRecord', 'playVoice', 'stopVoice', 'onVoicePlayE
        nd', 'downloadVoice', 'uploadVoice']
    });

    wx.ready(function(){
        //初始化微信语音
        g_yimInstance.initAudioMedia({
            showProgressTips: 1, //如果需要关闭，上传下载语音提示，可以设置为0
            maxRecordSecond : 60 //默认最大录制语音60秒，最大不能超过
60秒
        }, function(ret){
            console.log("录音类型 :" + ret.type+'; 支持:'+
            (ret.support?1:0));
            if(!ret.support){
                alert('当前版本不支持发送语音! ');
            }
        });
    });
});

```

开始录音（注意：目前语音只支持微信客户端，请先判断 **yim.isWeixin()**）

功能：开始录音 原型：

```

g_yimInstance.startAudioMessage(strRecvID, chatType, function(YIMErrorcodeOC,
errInfo){
    if(YIMErrorcodeOC === yim.YIMErrorcode.YIMErrorcode_Fail){
        if(errInfo.errCode === yim.YIMAUDIOCODE.INVOKE_TOOFAST) // 接口调用过快
        if(errInfo.errCode === yim.YIMAUDIOCODE.WAITFOR_LASTINVOKE) //上次调用未
        结束，可能还在录音或发送录音中
    }
});

```

- **strRecvID**: 接收者ID（用户ID或者群ID）
- **chatType**: 聊天类型，群聊/私聊
- **YIMErrorcodeOC**: 返回错误码，判断微信询问语音授权是否成功，以显示相应的展示UI

取消录音

功能：取消录音 原型：

```

g_yimInstance.cancelAudioMessage();

```

停止录音并发送

功能：停止录音并发送出去，当停止调用距开始调用不足一秒时会延迟1秒停止 原型：

```
g_yimInstance.stopAudioMessage(function(YIMErrorcodeOC, errInfo){
    if(YIMErrorcodeOC === yim.YIMErrorcode.YIMErrorcode_Fail){
        if(errInfo.errCode === yim.YIMAUDIOCODE.VOICE_TOOSHORT)    // 语音太短，
        小于1秒
        if(errInfo.errCode === yim.YIMAUDIOCODE.INVALID_INVOKE)    // 无效调用，
        可能未调用开始录音
        if(errInfo.errCode === yim.YIMAUDIOCODE.DUPLICATE_INVOKE) // 重复调用停
        止录音
        if(errInfo.errCode === yim.YIMAUDIOCODE.INVOKE_OUTTIME)    // 停止录音微
        信返回超时，微信坑
    }
});
```

- **YIMErrorcodeOC**: 错误码，详细描述见错误码定义
- **object ret** 相关调用信息

备注：是一个异步操作，操作结果会通过初始回调事件接口返回: onVoiceMsgSend

```
onVoiceMsgSend: function(YIMErrorcodeOC, audioInfo){
    audioInfo = {
        strMessageID,    //消息ID
        strVoiceID,      //语音ID
        intVoiceTime     //语音时长(s)
    };
}
```

播放语音消息

功能：将指定语音消息下载到本地 原型：

```
g_yimInstance.playAudioMessage(strVoiceID, function(YIMErrorcodeOC, errInfo){
    if(YIMErrorcodeOC === yim.YIMErrorcode.YIMErrorcode_Fail){
        if(errInfo.errCode === yim.YIMAUDIOCODE.INVOKE_TOOFAST)    // 调用接口太
        频繁，请等1秒
    }
});
```

- **strVoiceID**: 语音ID
- **YIMErrorcodeOC**: 语音播放完成时回调错误码

停止播放语音

功能：停止正在播放的语音信息 原型：

```
g_yimInstance.stopPlayAudioMessage(strVoiceID, function(YIMErrorcodeOC,
errInfo){
    if(YIMErrorcodeOC === yim.YIMErrorcode.YIMErrorcode_Fail){
        if(errInfo.errCode === yim.YIMAUDIOCODE.INVOKE_TOOFAST)    // 调用接口太
        频繁，请等1秒
        if(errInfo.errCode === yim.YIMAUDIOCODE.INVALID_INVOKE)    // 无效调用，
        可能没有调用开始播放接口
    }
});
```

- **YIMErrorcodeOC**: 停止播放回调错误码

自动播放队列

功能：当收到语音时需要自动按队列播放，输入语音时暂停可以参考 原型：

```
//先初始化自动播放回调方法及事件
g_yimInstance.initAutoPlayVoiceQueue({
    beginPlay: function(msg){
        //当消息开始播放时回调
    },endPlay: function(msg){
        //当消息结束播放时回调
    }
});

//当收到语音消息时，判断为语音消息，加入自动播放队列
onRecvMessage: function(message){
    //接收到语音消息
    if(message.msgType == yim.MessageBodyType.MessageBodyType_Voice){
        var content = JSON.parse(message.content);

        //如果是微信语音
        if(content.recordmode == 1){
            //添加到自动播放列表
            g_yimInstance.addToAutoPlayVoiceQueue(message);
        }
    }
}
```

3、接收消息

功能：消息接收采用异步回调的方式：**onRecvMessage**。

示例：

```
onRecvMessage: function(message){
    if(message.msgType == yim.MessageBodyType.MessageBodyType_Text){
        //添加文本消息到当前页面
        _addTextDom(message);
    }else if(message.msgType == yim.MessageBodyType.MessageBodyType_Voice)
    {
        //接收到语音消息
        var content = JSON.parse(message.content);
        if(content.recordmode == 1){
            //微信录音
            _addVoiceDom(content);
        } else { //普通语音
            _addVoiceDom(content);
        }
    }
}
```