

# DuerOS DCS Android SDK API使用指南

## sdk初始化相关

- DcsSdkBuilder类

方法	描述
DcsSdkBuilder withSdkConfig(SdkConfigProvider sdkConfigProvider)	提供相关的配置信息
DcsSdkBuilder withOauth(IOauth oauth)	设置登录的方式
DcsSdkBuilder withDeviceId(String deviceId)	设置唯一设备id
DcsSdkBuilder withWakeupProvider(IWakeupProvider wakeupProvider)	设置唤醒相关参数
DcsSdkBuilder withAudioRecorder(BaseAudioRecorder audioRecorder)	设置音频数据来源的实现类
DcsSdkBuilder withProductVersion(String appVersion)	设置应用版本号（可选）
DcsSdkBuilder withMediaPlayer(IMediaPlayer mediaPlayer)	设置sdk内部音乐播放器的实现类（可选）
DcsSdkBuilder withSpeakerDecoder(IDecoder decoder)	设置音频数据解码器（可选）
DcsSdkBuilder withSpeakerPlayer(IMediaPlayer mediaPlayer)	设置音频数据播放器（可选）
IDcsSdk build()	创建一个sdk的具体实现类

## sdk基本接口相关

- IDcsSdk接口

接口名	描述
void run(OnInitListener listener)	启动运行，登录成功后调用，sdk内部用于建立长连接，长连接建立成功后会只收到一次回调
void release()	释放资源，程序退出时调用
IVoiceRequest getVoiceRequest()	获取语音请求对象
void putDeviceModule(BaseDeviceModule deviceModule)	增加自定义端能力
void addConnectionStatusListener(IConnectionStatusListener connectStatusListener)	添加连接状态监听
void removeConnectionStatusListener(IConnectionStatusListener connectStatusListener)	移除连接状态监听

- SdkConfigProvider接口

接口名	描述
String clientId()	产品线的ClientId
int pid()	语音PID，没有取值限制，某些测试PID可能为负值
int regionId()	方言地域位置id
HttpProxy ~~httpProxy~~()	代理设置，该方法已过时

boolean compressPCM()	是否开启语音PCM压缩
boolean asrOnly()	是否仅ASR
String appKey()	语音appkey，和pid一一对应
boolean longSpeech()	是否开启长语音
int asrAudioSource()	设置识别音频的audiosource

• IConnectionStatusListener接口

接口名	描述
void onConnectStatus(ConnectionStatus connectionStatus)	长连接回调接口
ConnectionStatus	DISCONNECTED（长连接断开） PENDING (长连接正在连接中) CONNECTED (长连接连接正常)

• PlaybackControllerDeviceModule.CommandIssued 类

变量	描述
CommandIssuedPlay	播放
CommandIssuedPause	暂停
CommandIssuedPrevious	上一首
CommandIssuedNext	下一首

• IResponseListener 接口

接口名	描述
void onSuccess(int statusCode);	成功回调，statusCode http请求的响应code
void onFailed(DcsErrorCode dcsErrorCode);	失败回调，dcsErrorCode 出错的异常信息类
void onCancel();	取消

• IDirectiveReceivedListener 接口

接口名	描述
void onDirective(Directive directive)	指令回调方法

• Directive类

成员变量	描述
public Header header;	对应返回指令json的header
public Payload payload;	对应返回指令json的payload
private String rawMessage;	对应整个指令json
public String rawPayload;	对应整个指令payload
public JSONObject jsonObjectDirective;	对应整个指令jsonobject

• IDcsRequestBodySentListener接口

接口名	描述
void onDcsRequestBody(DcsRequestBody dcsRequestBody)	收到请求消息体回调

• Location.LocationHandler接口

接口名	描述
double getLongitude();	经度
double getLatitude();	纬度
String getCity();	城市名字
EGeoCoordinateSystem getGeoCoordinateSystem()	坐标

• EGeoCoordinateSystem类

成员变量	描述
WGS84	球面坐标
BD09LL	百度球面坐标
BD09MC	百度墨卡托坐标

语音识别相关

• IVoiceRequest接口

接口名	描述
void beginVoiceRequest(boolean enableVad)	开始语音流请求,enableVad true:开启Vad（语音尾点检测）， false: touch模式（按下开始发送语音流，抬起结束语音流）
void endVoiceRequest(final IVoiceRequestListener listener)	结束语音流请求,等待度秘后端返回结果
void cancelVoiceRequest(final IVoiceRequestListener listener)	取消语音请求，忽略度秘后端返回结果
void cancelVoiceRequest(boolean initiative, IVoiceRequestListener listener)	取消语音请求，忽略度秘后端返回结果，true：主动取消，false：被动取消，回调中发起语音请求
void addDialogStateListener(IDialogStateListener dialogStateListener)	添加会话状态listener
void removeDialogStateListener(IDialogStateListener dialogStateListener)	移除会话状态listener

• IVoiceRequestListener 接口

接口名	描述
void onSuccess();	结束语音请求或者取消语音请求的异步成功回调

• IDialogStateListener接口

接口名	描述
void onDialogStateChanged(DialogState dialogState)	会话状态变化回调，参数dialogState 会话状态
DialogState	IDLE (等待语音输入) LISTENING (正在语音输入) THINKING (语音请求结束，正在等待度秘后端返回结果) SPEAKING (正在进行(语音)tts播报)

扩展端能力相关

• BaseDeviceModule类

成员变量	描述
protected final String nameSpace	DeviceModule对应的唯一标示
protected final IMessageSender messageSender	event上传处理类

方法	描述
public abstract ClientContext clientContext()	端状态：服务端在处理某些事件时，需要了解在请求当时设备端各模块所处的状态。比如端上是否正在播放音乐，是否有闹钟在响，是否正在播报等等
public abstract void handleDirective(Directive directive) throws HandleDirectiveException;	处理服务端下发的指令
public abstract HashMap< supportPayload()	加入该deviceModule需要处理的指令和指令对应的Payload,@return HashMap> key为指令 NameSpace+指令的名字，value为改指令对应的payload的数据结构类的class
public abstract void release();	释放一些资源，比如监听器，在sdk.release的时候，sdk内部会调用
public String getNameSpace()	返回DeviceModule的nameSpace

唤醒相关

• IWakeupProvider 接口

接口名	描述
WakeUpConfig wakeupConfig();	获取唤醒配置
boolean enableWarning();	唤醒成功后是否播放提示音
String warningSource();	唤醒成功后的提示音资源，建议是android下的assets目录下的短小资源；如果是assets下的，url格式为 assets://filename；每次在播放唤醒提示音前调用该方法
float volume();	播放提示音的音量大小 0<=返回值<=1;每次在播放唤醒提示音前调用该方法
boolean wakeAlways()	是否开启唤醒
BaseWakeup wakeupImpl()	唤醒默认实现
int audioType()	传入AudioManager的STREAM_SYSTEM的参数

• WakeUpConfig.Builder类

方法	描述
Builder resPath(String resPath)	(必选) Snowboy .res文件的路径（相对assets）
Builder umdlPath(String umdlPath)	(必选) Snowboy .umdl文件的路径（相对于assets）
Builder sensitivity(String sensitivity)	(必选) Snowboy唤醒敏感度，格式为','分隔的浮点数，使用默认模型时值需为 "0.35,0.35,0.45"
Builder wakeUpWords(List wakeUpWords)	(必选) 唤醒词列表。需要注意的是，列表中的元素需要和Snowboy模型文件中一致。例如：默认模型文件中有3个唤醒词，分别为不同语速的【小度小度】，则每个元素的index字段分别应为1、2、3
Builder highSensitivity(String highSensitivity)	(可选) Snowboy唤醒敏感度-参数2，格式为','分隔的浮点数，使用默认模型时值需为 "0.45,0.45,0.55"
Builder onPlayingSensitivity(String onPlayingSensitivity)	(可选) MediaPlayer播放时Snowboy唤醒敏感度，格式为','分隔的浮点数
Builder onPlayingHighSensitivity(String onPlayingHighSensitivity)	(可选) MediaPlayer播放时Snowboy唤醒敏感度-参数2，格式为','分隔的浮点数
Builder audioGain(float audioGain)	(可选) Snowboy唤醒音量增益。若麦克风录制音量较小，可以尝试设为 >1 的值

• WakeUpWord类

字段	描述
int index	唤醒词在模型文件中的索引，一般为1...N
String word	唤醒词字符串，该字符串可由调用者定义，在命中对应索引的唤醒后，会原封不动传出来
byte[] wakeUpWordAudioData	唤醒词对应的语音数据
long time	唤醒词对应的时间ms
boolean isOneshot	是否是oneshot

• BaseWakeup类

方法	描述
public void initWakeup(WakeUpConfig wakeUpConfig)	初始化唤醒配置
public abstract void startWakeup()	开始唤醒
public abstract void stopWakeup(IStopWakeupListener l)	停止唤醒
public void release()	释放资源，比如调用底层库释放资源等
public void addWakeupListener(IWakeupListener listener)	唤醒结果的回调监听
public void removeWakeupListener(IWakeupListener listener)	移除唤醒结果回调
protected void fireOnWakeUpSucceed(WakeUpWord wakeUpWord)	回调唤醒成功
protected void fireOnInitWakeUpSucceed()	回调唤醒初始化成功

protected void <code>fireOnInitWakeUpFailed</code> (String errorMessage)	回调唤醒初始化失败，errorMessage 错误信息
public void <code>setSupportOneShot</code> (boolean supportOneShot)	设置是否支持oneshot
public void <code>setSensitivity</code> (String sensitivity)	设置灵敏度，默认空实现
public void <code>setHighSensitivity</code> (String highSensitivity)	设置灵敏度参数2，默认空实现

• **IWakeUpAgent**接口

接口名	描述
void <code>addWakeupAgentListener</code> (IWakeupAgentListener listener)	添加唤醒listener
void <code>removeWakeupAgentListener</code> (IWakeupAgentListener listener)	移除唤醒listener

• IWakeupAgentListener 接口

接口名	描述
void <code>onInitWakeUpSucceed</code> ()	初始化唤醒成功。只有接收到该回调后，方可开始唤醒
void <code>onInitWakeUpFailed</code> (String errMsg)	初始化唤醒失败时的回调
void <code>onWakeupSucceed</code> (WakeUpWord wakeUpWord)	唤醒成功，wakeUpWord 唤醒词
void <code>onWarningCompleted</code> ()	播放完提示音后回调，如果有提示音的话
void <code>onWarningError</code> (String error, IMediaPlayer.ErrorType errorType)	播放提示音失败时的回调

• **IWakeUpListener** 接口

接口名	描述
void <code>onInitWakeUpSucceed</code> ()	初始化成功后回调,比如拷贝唤醒词需要的文件到应用的私用目录等
void <code>onInitWakeUpFailed</code> (String errorMessage)	初始化失败后回调，这一般出现在拷贝唤醒资源失败的时候，errorMessage 错误信息
void <code>onWakeupSucceed</code> (WakeUpWord wakeUpWord)	唤醒成功后回调，wakeUpWord 唤醒词

• **IStopWakeupListener** 接口

接口名	描述
void <code>onStopWakeup</code> ()	停止唤醒后的回调

## 上传通讯录相关

• **IUpload**接口

接口名	描述
void <code>uploadPhoneContacts</code> (String contacts, boolean forceUpload, IUploadListener listener)	上传手机通讯录
void <code>uploadWechatContacts</code> (String contacts, IUploadListener listener)	上传微信通讯录

## 登录相关

• **IOauth**接口

--

接口名	描述
void getToken(IOauthCallback listener)	获取accessToken接口
void clearAccessToken()	清理accessToken接口

• IOauthCallback接口

接口名	描述
void onSuccess(String accessToken)	accessToken获取成功
void onError(String errorMsg)	accessToken获取失败
void onCancel()	accessToken获取被取消，如果是百度Oauth的code的方式，如果用户取消了会回调该方法

• ILoginListener接口

接口名	描述
void onSuccess(String accessToken)	登录成功
void onFailed(String errorMessage)	登录失败
void onCancel()	登录被取消了

## 错误信息

• IErrorListener接口

接口名	描述
void onErrorCode(DcsErrorCode errorCode)	发生错误时回调

• DcsErrorCode

错误码由：error（主错误码），suberror（子错误码），errorMsg(错误基本描述信息)，originMsg (原始的错误信息)组成。接入端需根据主错误码，和子错误码进行判断是否通知用户。

error	suberror	说明
VOICE_REQUEST_EXCEPTION	NETWORK_UNAVAILABLE	网络不可用
VOICE_REQUEST_EXCEPTION	TOKEN_IS_EMPTY	语音请求失败，没有有效的accessToken
VOICE_REQUEST_EXCEPTION	DIRECTIVE_IS_PENDING	长连接建立中
VOICE_REQUEST_EXCEPTION	1003、1005	网络连接超时
VOICE_REQUEST_EXCEPTION	2000、2002、2003、2004、2005、2006、2100	网络连接错误
VOICE_REQUEST_EXCEPTION	3001	麦克风异常，或输入的音频流异常
VOICE_REQUEST_EXCEPTION	3100	VAD资源设置不正确
VOICE_REQUEST_EXCEPTION	3101	VAD未检测到说话
VOICE_REQUEST_EXCEPTION	3102	VAD检测到语音过短
VOICE_REQUEST_EXCEPTION	4001、4002、4003、4004	服务端异常

VOICE_REQUEST_EXCEPTION	5001	客户端无法加载动态库
VOICE_REQUEST_EXCEPTION	6001	语音过长
VOICE_REQUEST_EXCEPTION	7001	无匹配识别结果
VOICE_REQUEST_EXCEPTION	8001	识别引擎忙
VOICE_REQUEST_EXCEPTION	9001	无录音权限
VOICE_REQUEST_EXCEPTION	10002-10012	离线识别错误
VOICE_REQUEST_EXCEPTION	11002-11010	唤醒相关错误
VOICE_REQUEST_EXCEPTION	负值	服务端返回错误码
LOGIN_EXCEPTION	LOGIN_FAILED	登录失败
LOGIN_EXCEPTION	LOGIN_CANCEL	登录取消
DECODER_FAILED	DECODER_FAILED	解码失败
REQUEST_HINT	INVALID_REQUEST	请求格式错误
REQUEST_HINT	UNAUTHORIZED_REQUEST	请求认证失败
REQUEST_HINT	THROTTLING_EXCEPTION	请求数量太多/频率太高
REQUEST_HINT	INTERNAL_SERVICE_EXCEPTION	服务端内部错误
REQUEST_HINT	NA	服务端不可用
EVENT_REQUEST_EXCEPTION	EVENT_TOKEN_ISNULL	事件请求时，token为空
EVENT_REQUEST_EXCEPTION	EVENT_ISNULL	事件请求时事件为空
JSON_PARSE_EXCEPTION	JSON_PARSE_EXCEPTION	json解析出错
DIRECTIVE_EXCEPTION	DIRECTIVE_EXCEPTION	长连接请求异常

修改语音播报相关-音色等

- TtsOnlineInfo类

字段	描述
volume	可选，声音大小 0-9
speed	可选，语速 0-9
pitch	可选，语调0-9
speaker	可选，声音 0:普通女声 1:普通男声 2:特别男声 3:情感男声"度逍遥" 4:情感儿童声"度丫丫"
aue	可选，音频格式，不要随意改动，目前为MP3,常量值:具体参考该类的常量AUDIO_FORMAT_
rate	可选，码率，常量值:具体参考该类的常量AUDIO_BITRATE_
xml	可选，是否为xml

- TtsInfoHelper.IUpdateTtsCallback类

接口名	描述



void <code>success()</code>	设置成功
void <code>error</code> (String errMsg)	设置失败，errMsg为错误信息

## API试用接口（后继可能有变化，请谨慎使用）

• **InternalApi**类

方法	描述
public <code>InternalApi</code> (DcsSdkImpl dcsSdkimp)	构造，传入IDcsSdk接口实现类
public void <code>sendQuery</code> (String query)	发送文本请求,即把语音改为文字进行query
public void <code>speakRequest</code> (String text)	在线tts播报请求
public void <code>speakOfflineRequest</code> (String text)	离线-本地文本tts播报请求
public void <code>stopSpeaker</code> ()	停止在线或者离线的tts的播放,停止后不在进行播放
public void <code>pauseSpeaker</code> ()	暂停所有的音频播放(离线文本tts，在线tts，音乐)
public void <code>resumeSpeaker</code> ()	恢复所有的音频播放(离线文本tts，在线tts，音乐)
public void <code>interruptSpeaker</code> ()	临时暂停所有的音频播放(离线文本tts，在线tts，音乐)，再次有任意speaker play指令下发都可以继续播放，同pauseSpeaker必须调用resumeSpeaker才能播放有区别
public void <code>updateOnlineTtsInfo</code> (String clientId, TtsOnlineInfo info, TtsInfoHelper.IUpdateTtsCallback callback)	修改在线tts配置
public void <code>initWakeUp</code> ()	初始化唤醒，如果IWakeupProvider.wakeAlways 返回true的情况下并开启录音
public void <code>stopWakeup</code> (IStopWakeupListener I)	停止唤醒
public void <code>startWakeup</code> ()	开始唤醒
public void <code>uninitWakeUp</code> (IStopWakeupListener listener)	停止唤醒并关闭录音
public AbsDcsClient <code>getDcsClient</code> ()	获取AbsDcsClient
public BaseDeviceModule <code>getDeviceModule</code> (String namespace)	获取DeviceModule
public void <code>removeDeviceModule</code> (String nameSpace)	移除DeviceModule
public IMessageSender <code>getMessageSender</code> ()	获取内部sdk的发送上报event的辅助类
public IWakeupAgent <code>getWakeupAgent</code> ()	获取唤醒的辅助类，用于添加唤醒listener等
public float <code>getCurrentVolume</code> ()	获取当前播放器的音量，最高优先级播放器的音量
public void <code>setVolume</code> (float volume)	设置播放器的音量,0<=volume<=1
public void <code>sendCommandIssuedEvent</code> (PlaybackControllerDeviceModule.CommandIssued commandIssued)	音乐播放控制，上一首，下一首，暂停，播放
public String <code>getDeviceID</code> ()	获取当前设备的唯一id
public void <code>postEvent</code> (Event event, IResponseListener responseListener)	发送event
public void <code>postEvent</code> (JSONObject event, IResponseListener responseListener)	发送event

public void <code>handleDirective</code> (Directive directive)	处理指令
public void <code>addDirectiveReceivedListener</code> (IDirectiveReceivedListener listener)	添加指令listener
public void <code>removeDirectiveReceivedListener</code> (IDirectiveReceivedListener listener)	移除指令listener
public void <code>addRequestBodySentListener</code> (IDcsRequestBodySentListener listener)	添加事件listener
public void <code>removeRequestBodySentListener</code> (IDcsRequestBodySentListener listener)	移除事件listener
public Location.LocationHandler <code>getLocationHandler</code> ()	获取位置信息
public void <code>setLocationHandler</code> (Location.LocationHandler handler)	设置位置handler
public void <code>setAsrMode</code> (int asrMode)	asr的模式
public void <code>setSupportOneshot</code> (boolean isOneshot)	是否开启oneshot
public void <code>setInteractionStrategy</code> (IInteractionStrategy wakeUpStrategy)	设置唤醒-语音请求交互策略
public void <code>login</code> (ILoginListener loginListener)	登录
public IUpload <code>getUpload</code> ()	上传模块(通讯录&微信联系人)
public ISmartHomeManager <code>getSmartHomeManager</code> ()	智能设备管理
public void <code>addErrorListener</code> (IErrorListener listener)	添加sdk错误监听
public void <code>removeErrorListener</code> (IErrorListener listener)	移除sdk错误监听
public void <code>logout</code> ()	退出登录
public void <code>addAudioPlayListener</code> (IMediaPlayer.IMediaPlayerListener listener)	注册提供音乐播放器开始播放 暂停 播放结束 以及播放进度等接口
public void <code>removeAudioPlayListener</code> (IMediaPlayer.IMediaPlayerListener listener)	解除注册 音乐播放器开始播放 暂停 播放结束 以及播放进度等接口
public void <code>addFinishedDirectiveListener</code> (IFinishedDirectiveListener listener)	注册一次ASR/TextInput所有返回指令处理完毕回调
public void <code>removeFinishedDirectiveListener</code> (IFinishedDirectiveListener listener)	解除注册一次ASR/TextInput所有返回指令处理完毕回调
public void <code>addTTSPositionInfoListener</code> (ITTSPositionInfoListener listener)	添加语音播报（TTS）文本同步监听
public void <code>removeTTSPositionInfoListener</code> (ITTSPositionInfoListener listener)	移除语音播报（TTS）文本同步监听
public void <code>exitMultiScene</code> ()	退出当前的多伦场景
public IOauth <code>getOauth</code> ()	获取登录的实现
public boolean <code>isLogin</code> ()	是否登录
public void <code>addListenerToAllMediaPlayer</code> (IMediaPlayerListener listener)	给所有的MediaPlayer添加监听
public void <code>removeListenerToAllMediaPlayer</code> (IMediaPlayerListener listener)	移除给所有MediaPlayer添加的监听
public void <code>setDebugBot</code> (String botId)	调试技能, botId 技能id

public BaseAudioInput <code>getAudioInput()</code>	获取audioinput
public IDcsInternalProvider <code>getDcsInternalProvider()</code>	获取DCS内置的信息获取管理类
public void <code>addNearAsrEndListener</code> (INearAsrEndListener listener)	touch模式下asr识别完后回调
public void <code>removeNearAsrEndListener</code> (INearAsrEndListener listener)	移除touch模式下asr识别完后回调
public void <code>switchAsrToNear</code> (final BaseAudioRecorder audioRecorder, final ISwitchAsrNearFarListener listener)	切换到近场asr
public void <code>switchAsrToFar</code> (final BaseAudioRecorder audioRecorder, final ISwitchAsrNearFarListener listener)	切换到远场asr
public TtsImpl <code>initLocalTts</code> (Context context, String pid, String key, String apiKey, String secretkey)	初始化本地tts
public TtsImpl <code>initLocalTts</code> (Context context, String pid, String key, String apiKey, String secretkey,String appld, String licenseFile)	初始化本地tts
public TtsImpl <code>initLocalTts</code> (Context context, int streamType, String pid, String key, String apiKey,String secretkey, String appld, String licenseFile)	初始化本地tts
public void <code>setBDuss</code> (String bduss)	设置bduss
public void <code>setSpeakerStreamType</code> (int streamType)	设置AudioTrack的StreamType
public void <code>setSpeakerUsage</code> (int usage)	设置AudioTrack的Usage
public ICES <code>getStatistics</code> ()	获取对外开发的打点统计接口
public String <code>getActiveDialogRequestId</code> ()	获取当前的会话id

## 其它接口

• **IInteractionStrategy**接口

接口名	描述
void <code>onInitWakeUpFinished</code> ()	唤醒初始化后自定义策略，默认策略为：开启唤醒
void <code>onWakeUpFinished</code> (WakeUpWord wakeUpWord)	唤醒完成后自定义策略，默认策略为：开始语音请求
void <code>onVoiceRequestFinished</code> ()	语音请求结束后自定义策略，默认实现：开启唤醒
void <code>onVoiceRequestStart</code> ()	语音请求开始后自定义策略，默认实现：停止唤醒

• **IFinishedDirectiveListener**接口

接口名	描述
void <code>onFinishedDirective</code> ()	ASR/TextInput所有返回指令处理完毕回调

• **ITTSPositionInfoListener**接口

接口名	描述
void <code>onPositionInfo</code> (long pos, long playTimeMs, long mark)	tts文本同步，pos 播放到第几个字，playTimeMs 当前播放时间，mark 页数，如果Speak指令中没有 andsubContentsInCharacters 则 mark 为-1

# 专业术语

---

名词	术语解释
DCS	DuerOS服务端与设备端之间的通讯协议的简称
CLIENT_ID	产品ID，用户在开放平台申请设备时获得
PID	语音识别的产品ID
ASR	语音识别
TTS	语音合成
VAD模式	语音尾点检测
TOUCH模式	按下开始发送语音流，抬起结束语音流
oneshot	唤醒后不用等待，可以立即进行语音输入
Kitt	SDK 内置的唤醒库，全名Kitt Snowboy
device id	设备指纹/设备 ID，能够标识设备的唯一性
longspeech	长语音识别，没有60s时长的限制，可识别数小时的音频