

文件名称：DLNA接收端SDK接口文档
Android平台
当前版本：V1.2

苏州必捷网络有限公司

修订记录

版本号	拟制/修改人	拟制/修改日期	评审人	修改内容要点
1.0	方一鸣	2019.05.10		初稿
1.1	方一鸣	2019.05.19		新增音视频信息接口
1.2	方一鸣	2019.05.29		新增DLNA厂商信息设置
文档初始拟定时，可不填“评审人”以及“修改内容要点”				

目 录

修订记录.....	2
目录.....	3
1 概述.....	4
1.1 目的.....	4
1.2 读者对象.....	4
1.3 缩略语定义.....	4
2 范围.....	4
2.1 功能.....	4
2.2 SDK框架.....	4
2.3 SDK的DEMO实现.....	4
2.4 SDK交付物.....	4
3 接口.....	5
3.1 DlnaModule类中的接口说明.....	5
3.1.1 设置客户定制的模块实现类接口.....	5
3.1.2 初始化接口.....	5
3.1.3 去初始化接口.....	5
3.1.4 与发射端交互的接口.....	5
3.2 ModuleImplIf接口说明.....	6
3.2.1 会话接入接口.....	6
3.2.2 会话结束接口.....	6
3.3 MediaChannel会话处理接口.....	6
3.3.1 设置窗口句柄.....	6
3.3.2 音量控制接口.....	6
3.3.3 音视频播放控制相关接口.....	7
3.3.4 图片显示接口.....	7
3.3.5 音视频详细信息接口.....	7
4 Demo说明.....	7
5 客户如何使用SDK.....	7

1 概述

1.1 目的

用于指导使用 DLNA 接收端 SDK 的开发人员进行开发及测试。

1.2 读者对象

本文档适用于开发 Android 平台 DLNA 接收端的开发人员。

1.3 缩略语定义

缩写名称	英文	中文
DLNA		数字生活网络联盟

2 范围

2.1 功能

本SDK可以接受来自使用DLNA协议的发射端的投屏请求。

2.2 SDK框架

DLNAReceiver SDK 总体分为两层

- 1) cast_base_lib-1.0.18-release.aar: 它是一个 Android Module, 它定义了基础的 MediaChannel, Module 接口。
- 2) bj_dlna_mediarender-1.0.11-release.aar : 它是一个 Android Module , 定义了 DlnaMediaRenderModule, 以及相关 JNI 接口。

应用程序应基于 bj_dlna_mediarender-1.0.11-release.aar 进行开发。应用程序原则上不应修改 cast_base_lib-1.0.18-release.aar, bj_dlna_mediarender-1.0.11-release.aar 中的内容。

2.3 SDK的DEMO实现

DlnaMediaRenderDemo是接收端的一个参考实现，它基于bj_dlna_mediarender-1.0.10-release.aar 实现了DLNA接收端功能。其中DlnaModuleImp实现了ModuleImpItf接口；DlnaPictureChannel实现了 MediaChannel接口，实现了对DLNA图片会话的处理；DlnaMusicChannel实现了MediaChannel接口，实现了对DLNA音频会话的处理；DlnaVideoChannel实现了MediaChannel接口，实现了对DLNA视频播放会话的处理。

2.4 SDK交付物

- 动态库
- DEMO源代码

- SDK接口文档

3 接口

接口主要在DlnaModule, ModuleImpItf和MediaChannel中;用户可以自己实现相关接口，也可参考我司提供的DEMO源代码实现。

3.1 DlnaModule类中的接口说明

3.1.1 设置客户定制的模块实现类接口

```
public void setImp(ModuleImpItf imp)
```

3.1.2 初始化接口

```
public boolean init(Context context, Properties props);
```

DlnaModule类的Init方法初始化Dlna接收端模块。App在启动做初始化时调用。

其中Properties支持以下属性的设置：

PARA_NAME_DEVICE_NAME: 接收端名称，DLNA接收端显示的名称。

PARA_NAME_DEVICE_UUID: DLNA Media Renderer的唯一标识码，该UUID建议持久化保存，不可随意变化。

PARA_NAME_MANUFACTURER: 厂商名称。

PARA_NAME_MANUFACTURERURL: 厂商网址。

PARA_NAME_MODELDESCRIPTION: 设备描述。

PARA_NAME_MODELNAME: 设备型号。

PARA_NAME_MODELURL: 设备网址。

电脑进入文件管理器然后选择网络就能看见DLNA设备，选择相应设备查看属性可以看到相关的厂商信息。

具体实现可以参考DEMO源代码。

3.1.3 去初始化接口

```
public native void fini();
```

DlnaModule类的fini方法去初始化DLNA模块。App销毁DLNA接收端服务时调用。

3.1.4 与发射端交互的接口

以下接口主要用于和发射端进行音视频播放的进度同步：

```
public void setTransstionEvent(MediaChannel channel);
```

接收端正在准备播放音视频时调用，用于通知发射端播放器进行状态更新。

```
public void sendDurationEvent(MediaChannel channel, int duration);
```

接收端音视频播放准备完成时调用，用于通知发射端播放器的进度条总时长更新。

```
public void setPositionEvent(MediaChannel channel, int position);
```

接收端音视频播放过程中调用，用于通知发射端播放器进行播放进度更新。

```
public void setPlayStateEvent(MediaChannel channel);
```

接收端音视频开始播放时调用，用于通知发射端播放器进行状态更新。

`public void setPauseStateEvent(MediaChannel channel);`

接收端音视频暂停播放时调用，用于通知发射端播放器进行状态更新。

`public void setStopStateEvent(MediaChannel channel);`

接收端音视频停止播放时调用，用于通知发射端播放器进行状态更新。

具体实现可以参考DEMO源代码。

3.2 ModuleImplIf接口说明

ModuleImplIf定义了一个接口类，需要用户去实现这些接口。

3.2.1 会话接入接口

`public MediaChannel reqMediaChannel(MediaChannelInfo channelInfo,UserInfo userInfo);`

输入：会话信息，描述其业务类型。

channelInfo描述了当前会话的类型信息。

userInfo描述了发射端的IP，设备型号，设备名称等信息。

输出：创建的 MediaChannel对象。

这是一个回调接口，当DLNA协议栈发现有会话接入时，JNI层会主动调用此接口，应用层需要实现相关逻辑，需要在客户自身的ModuleItf实现类中去实现该逻辑。

具体实现可以参考DEMO源代码。

3.2.2 会话结束接口

`public void relMediaChannel(MediaChannel channel)`

回调接口，当底层收到会话结束时调用。应用层实现相关逻辑。

具体实现可参考DEMO源代码。

3.3 MediaChannel会话处理接口

当会话建立成功后，协议栈会调用MediaChannel类中的相应接口回吐数据或者获取状态。以下接口需要在MediaChannel子类中实现，客户需要根据自身实际情况实现下列接口。

MediaChannel子类中重要的接口如下，DLNA接收端需要重点关注以下接口。

3.3.1 设置窗口句柄

`public void setSurface(Surface surface)`

配置窗口句柄，用于render输出。Surface由用户创建的播放器视图获取。参考Demo中的实现。

3.3.2 音量控制接口

`public void setVolume(int volume)`

volume取值范围为0-100。0表示静音，100表示音量为最高。

`public void setVolumeMute()`

静音

`Public void setVolumeUnmute()`

取消静音

以上接口涉及DLNA音频投屏会话，视频投屏会话。参考Demo中的实现。

3.3.3 音视频播放控制相关接口

`public void pause()`

该接口通知播放器暂停视频播放。

`public void play()`

该接口通知播放器开始或继续视频播放。

`public void seek(int sec)`

通知播放器进度条拖放到的位置。单位为秒。

`public int getPts()`

获取播放器当前播放到的位置。单位为毫秒。

`public int getDuration()`

获取当前播放的视频URL的时长。单位为毫秒。

以上接口涉及DLNA音视频投屏播放场景。

3.3.4 图片显示接口

`public void showPicture(String pic)`

JNI回调该接口回吐图片的URL。

该接口涉及图片投屏播放场景。

3.3.5 音视频详细信息接口

MediaModel包含作者、标题、所属专辑以及专辑封面信息，该对象可通过MediaChannelInfo的getMediaModel()方法获取，而MediaChannelInfo可以通过MediaChannel的getMediaInfo()方法获取，即MediaModule model = MediaChannel.getMediaInfo().getMediaModel()。需要注意该功能需要发射端播放器支持。

标题名称可通过MediaModule的getTitle()获得；

作者名字可通过MediaModule的getArtist()获得；

专辑名称可通过MediaModule的getAlbum()获得；

专辑封面图可通过MediaModule的getAlbumUri()获得。

4 Demo说明

Demo中DlnaModuleImp实现了ModuleImpItf接口，实现了relMediaChannel方法，此处可以控制是否接入某个会话的逻辑控制。其中reqMediaChannel中创建了与会话类型对应的MediaChannel实现，并启动播放相关的View，创建用于播放的Surface。 DlnaPictureChannel, DlnaVideoChannel, DlnaMusicChannel分别对应DLNA图片投屏，视频投屏，纯音频播放这几种业务类型，实现了MediaChannel中相关功能接口。

5 客户如何使用SDK

- 1) 导入aar，在app目录下新建libs目录并将aar文件放在该目录下，然后在build.gradle（app）中dependencies上方及内部分别添加如下代码：repositories{ flatDir { dirs 'libs' } }, compile (name: 'bj_dlna_mediarender_1.0.10_release.aar', ext: 'aar') compile (name: 'cast_base_lib-1.0.18-release.aar', ext: 'aar')，可参考Demo。

- 2) 实现ModuleImpItf接口，参考DlnaModuleImp。reqMediaChannel接口返回自定义的MediaChannel实现类的实例，并启动播放界面。
- 3) 实现MediaChannel相关功能接口，参考DlnaPictureChannel, DlnaVideoChannel, DlnaMusicChannel。
- 4) 实现自身的播放界面，可参考view这个package中的view的实现，将播放界面的Surface设置到MediaChannel中。