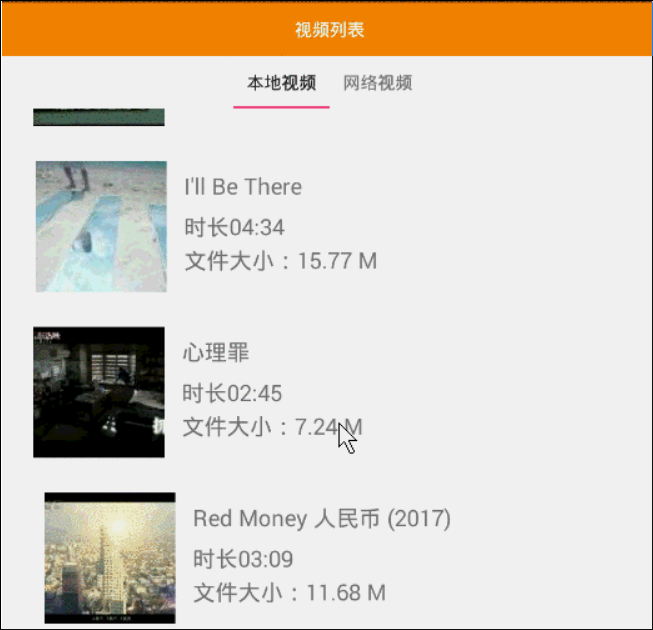
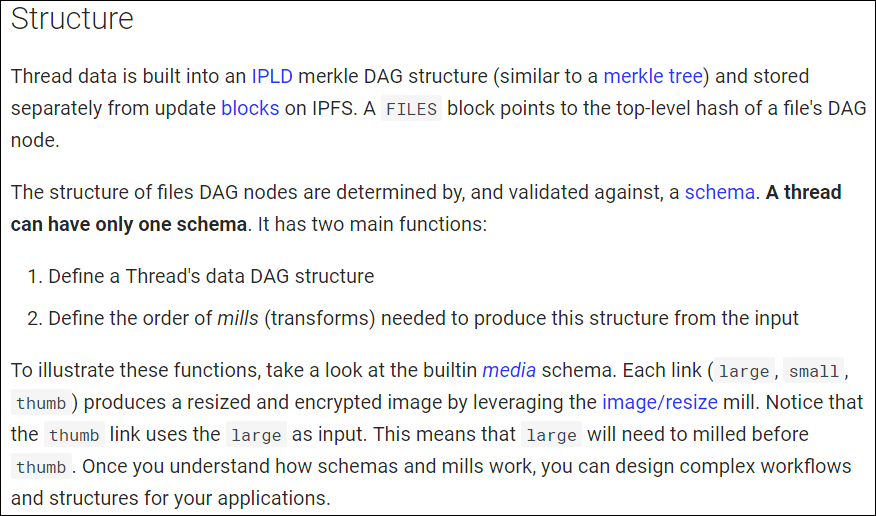
# Textile-Photos需要支持Group加载本地视频文件

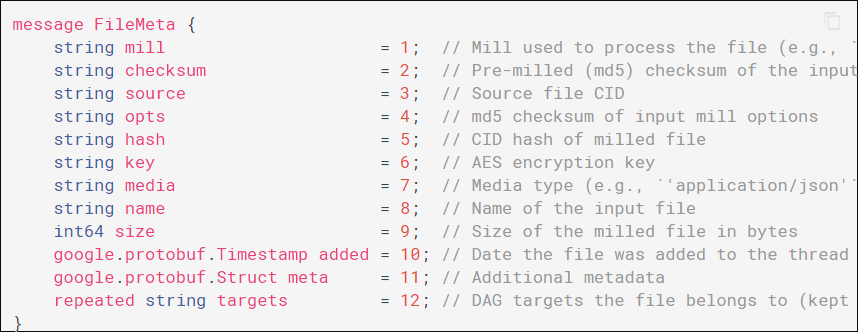
## 当向Group内添加本地视频文件资源时，Textile-Photos需要调用Android视频文件资源（类似于图片文件发现和列表管理）管理API，获取资源列表、视频缩微图，视图呈现类似如下图所示：



## 当选中指定的视频文件以后，该文件被加入Group组内，即在Group试图内能够显示已经加入的视频文件，视频文件的视图显示可以采取上图中“文件名字+视频时长+文件大小”的显示格式，使得用户方便的辨识该文件为视频文件，而不是图片文件，或者也可以其他的区别标示方法

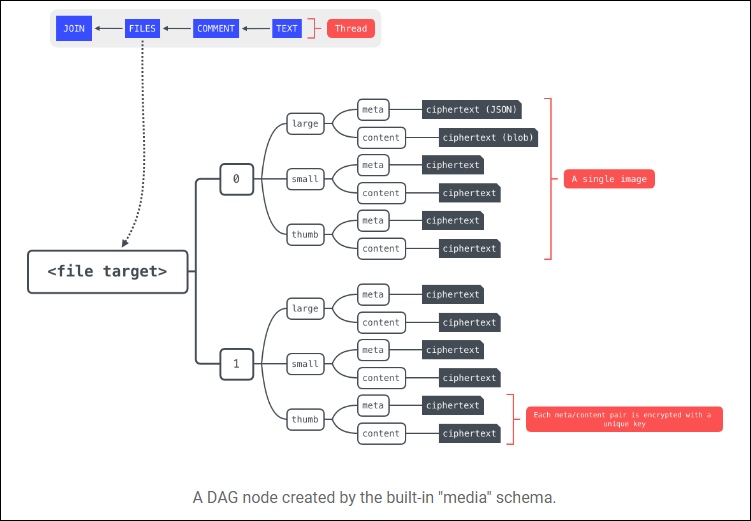
## 加入Group内的视频文件，现有的数据集Thread数据结构需要支持视频文件格式，类似原有的图片文件数据结构可能需要进行修改和扩展：





如上绿色加亮的“string media”字段目前支持“image/json”，需要考虑支持“video/json”类型以及相关的处理流程；

## 目前已经支持的定义不同图片尺寸，需要考虑是否需要支持不同的视频清晰度和窗口尺寸，即支持类似图片文件的DAG+Thread数据组织管理模式，生成对应的视频文件数据管理模式：



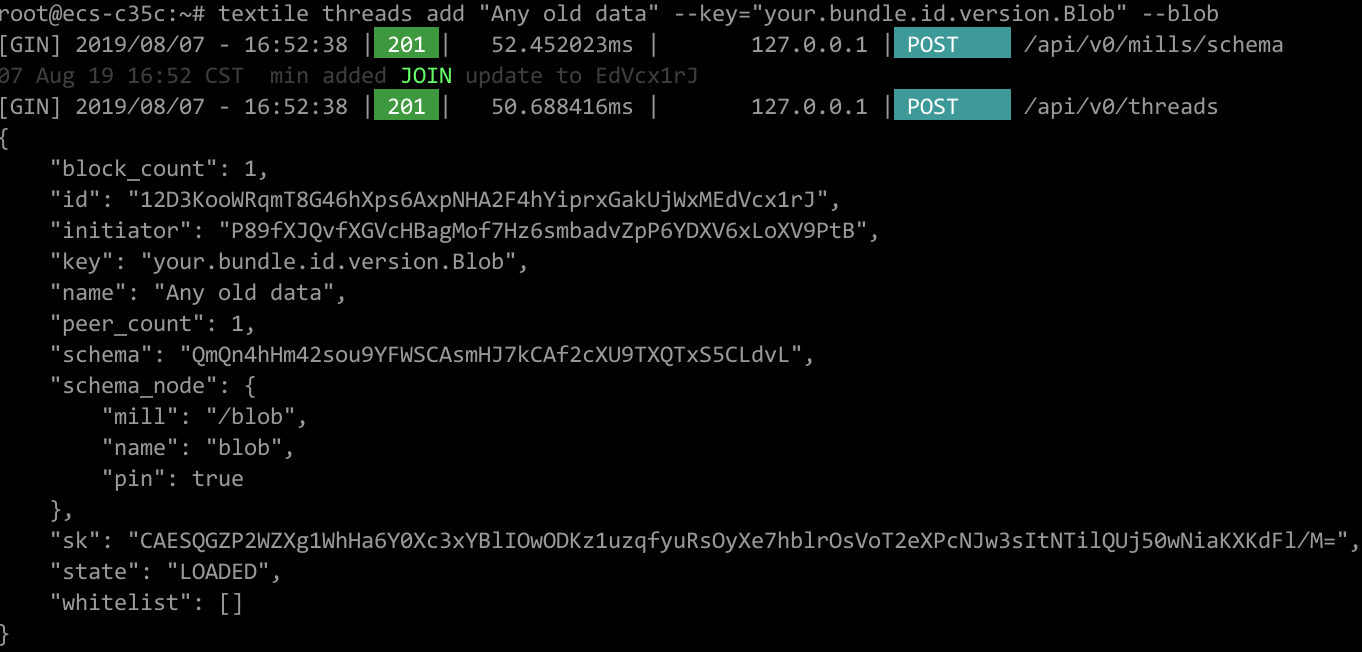


## 目前Textile使用Thread进行数据管理，内部文件都需要支持加密，新增支持视频文件以后，也需要支持加密保护，同时需要考虑加密、解密对处理性能的影响：

## 目前向Textile Thread中加入图片的相关处理，部分涉及媒体流程的命令行、功能处理（函数）需要梳理，针对新增的视频文件进行扩展支持，如下流程需要梳理：

二进制大文件Thread数据：

textile threads add "Any old data" --key="your.bundle.id.version.Blob" –blob



建立相册Thread数据：

textile thread add "Dogs" --type="open" --sharing="shared" –media

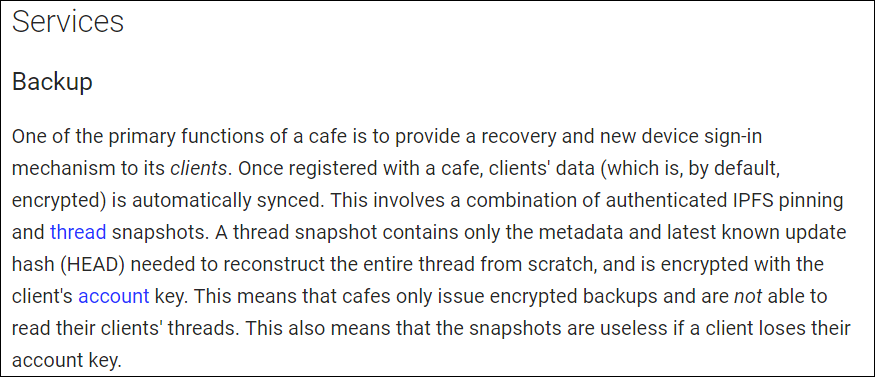


向Thread中添加图片：

textile files add <thread-id> ~/Downloads/william-milliot-510766-unsplash.jpg --caption="Dog at work”

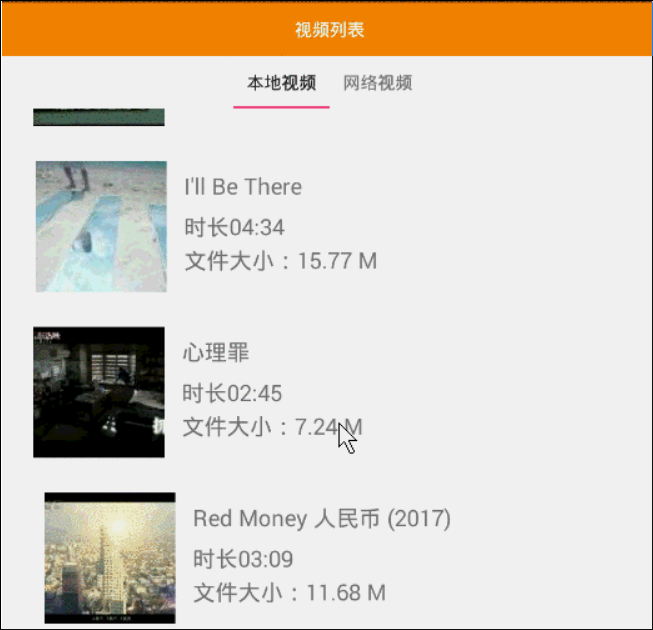
# Textile-Photos需要支持自动备份本地视频文件

## 当视频文件加入Group组内，需要支持自动将该文件“以特定数据集Thread ID”管理的向Cafe节点进行备份，该备份流程依托于目前Textile的自动备份功能



# Textile-Photos需要支持Group自动共享、获取共享的视频文件缩微图

## Thread内数据会自动向Cafe节点同步，由于视频文件Size较大，无法短时间快速上传，因此需要支持“缩微图+“名称+格式+视频时长+文件大小”等信息”首先快速上传Cafe节点，便于Group内其他成员快速感知此共享资源：

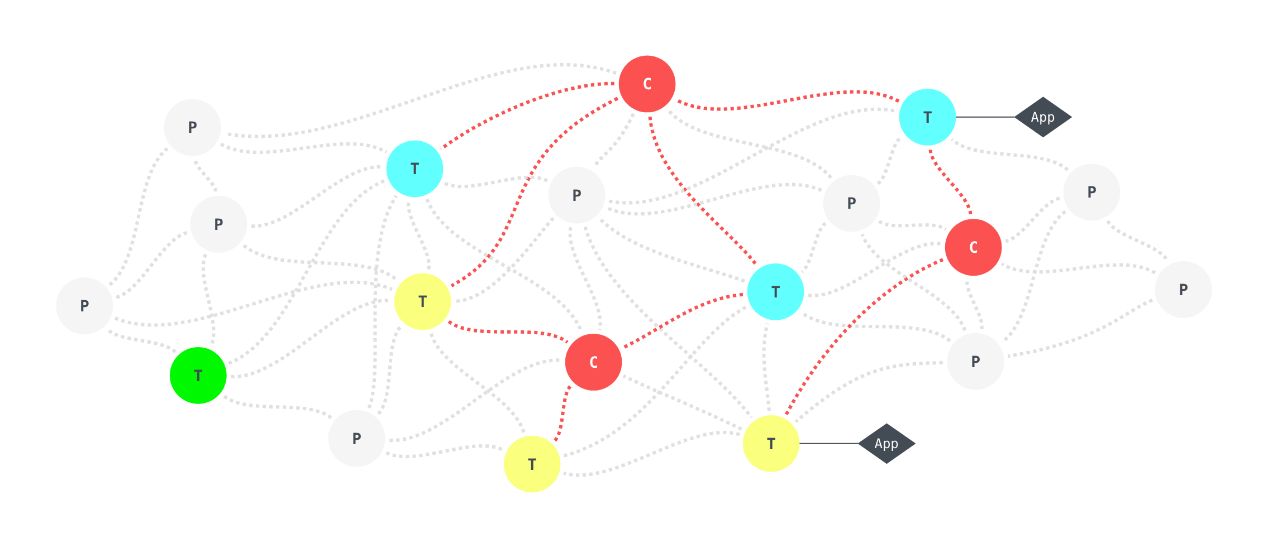


即视频文件由成员加入Group以后，该视频文件的缩微图和视频信息“名称+格式+视频时长+文件大小”等信息需要通过Texile数据同步机制快速的向Group内其他成员同步，其他的成员的Group视图界面应该能正确的显示共享视频的缩微图+“名称+格式+视频时长+文件大小”等信息；

# Textile-Photos需要支持共享视频文件的高速下载

## 共享视频下载性能首先依托现有的Cafe节点能力，目前本地视频文件加入Textile Thread以后，会自动向Cafe节点同步，待同步完成以后Group内其他成员有下载需求在启动下载，目前分析该下载过程性能较慢

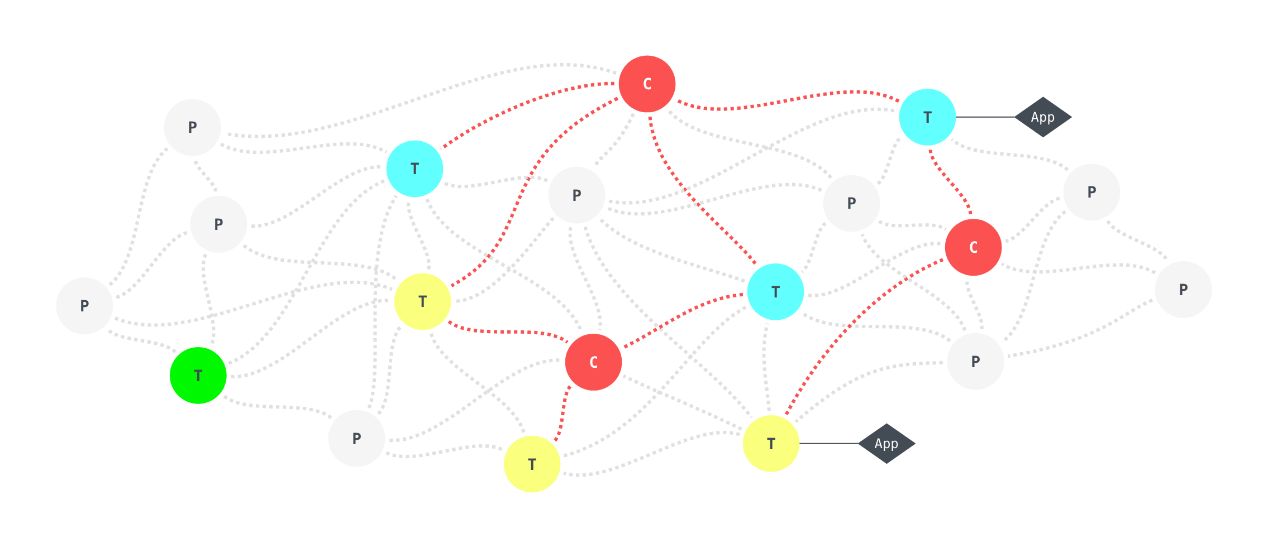
## 需要考虑支持类似：“一边上传 + 一边下载”的模式，即数据发布节点将原始资源文件切割为特定大小的数据块，一块一块的向Cafe节点同步，同时订阅节点（Group内其他节点）可以一块一块的并行下载，即订阅节点不依赖上传节点完整上传视频文件结束；



Group成员A: 向组内共享视频文件，一片片向Cafe节点同步

Group成员B： 与A并行着从Cafe节点一片片下载视频文件

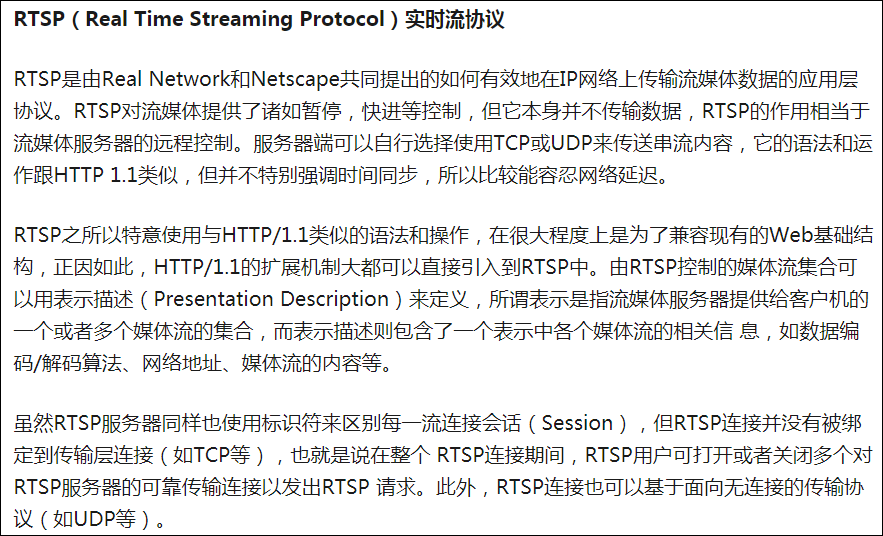
## 由于Textile底层的IPFS支持并行多点下载，既可以按照Merkel-DAG的各个数据块的Hash-ID并行从多个节点获取数据块，然后在本地进行拼装，因此可以考虑：Group成员从Cafe节点、数据发布者等多个节点同时下载视频文件数据，加速下载性能；



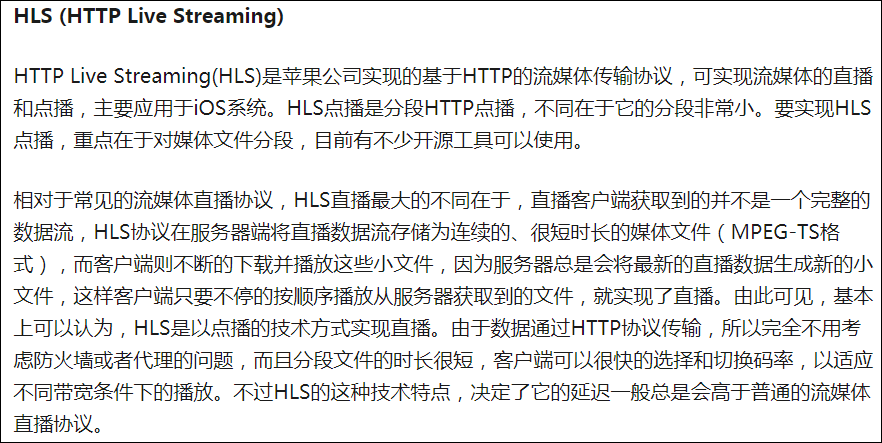
Group成员B：从A、B、Cafe节点同时并行下载视频文件

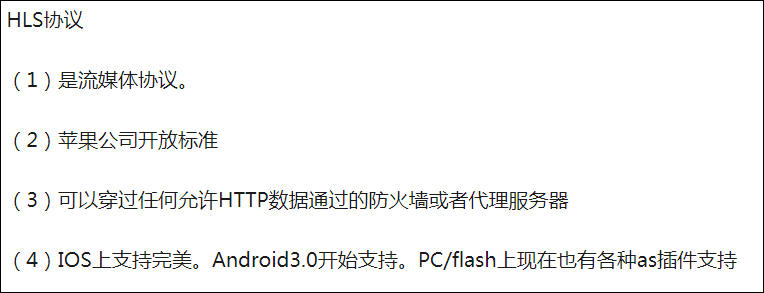
Group成员A: 向组内共享视频文件

## Textile支持的视频文件下载需要依托IPFS的数据块获取能力，很难支持类似现有互联网中类似“RTSP（Real Time Streaming Protocol）实时流协议”，IPFS本质上还是基于特定大小的文件来生成唯一的Hash-ID，然后进行路由的发布、查找、下载



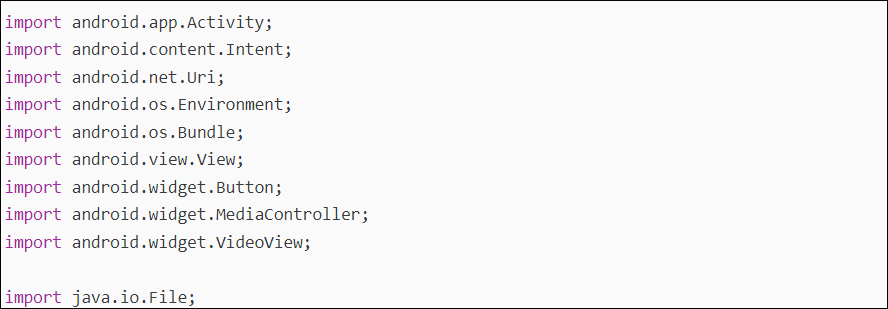
## 如果需要支持实时流协议，可以参考苹果开放标准的：HLS协议，Android版本从3.0版本也已经支持，目前IPFS和Textile的处理流程更加面向文件系统，RTSP、HLS等是依托现有的HTTP-WEB架构实现视频实时文件的快速安全传递、播放、控制等，Textile当前可以考虑，但是优先支持静态视频文件的共享，对实时视频的支持作为扩展考虑；





# Textile-Photos需要支持视频文件的播放

## 调用系统默认的播放器





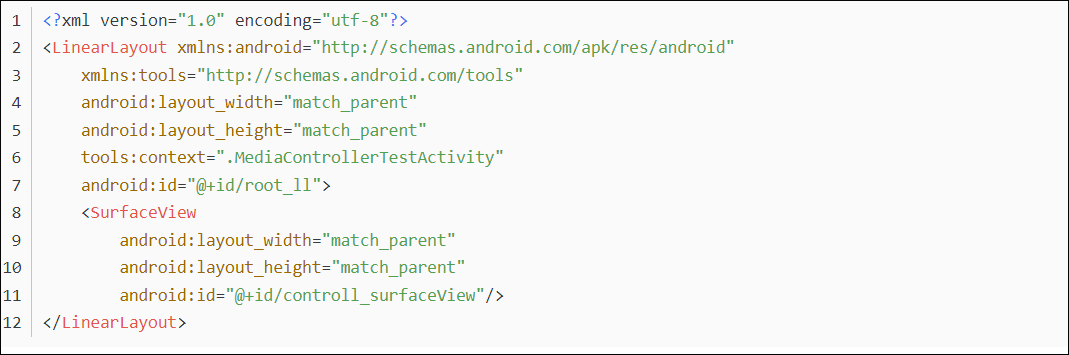
## MediaController+VideoView实现方式

使用Android自带的MediaPlayer、MediaController等类可以很方便的实现视频播放的功能。支持的视频格式有MP4和3GP等。VideoView继承了SurfaceView同时实现了MediaPlayerControl接口，MediaController则是安卓封装的辅助控制器，带有暂停，播放，停止，进度条等控件。通过VideoView+MediaController可以很轻松的实现视频播放、停止、快进、快退等功能:



## MediaPlayer+SurfaceView+MediaController

MediaPlayer+SurfaceView也可以使用系统自带的MediaController控制器。使用这个方式实现，布局文件只需一个SurfaceView即可，其他的控件都交给MediaController控制器，布局文件如下：





## 视频播放这块目前不是难点，目前网上也有很多公开的方法，上面三点是从CSDN中获取，供参考，具体的链接为：

<https://blog.csdn.net/liuzhi0724/article/details/81318816>

<https://blog.csdn.net/youmingyu/article/details/52914513>

<https://blog.csdn.net/tzl0322/article/details/77414978>