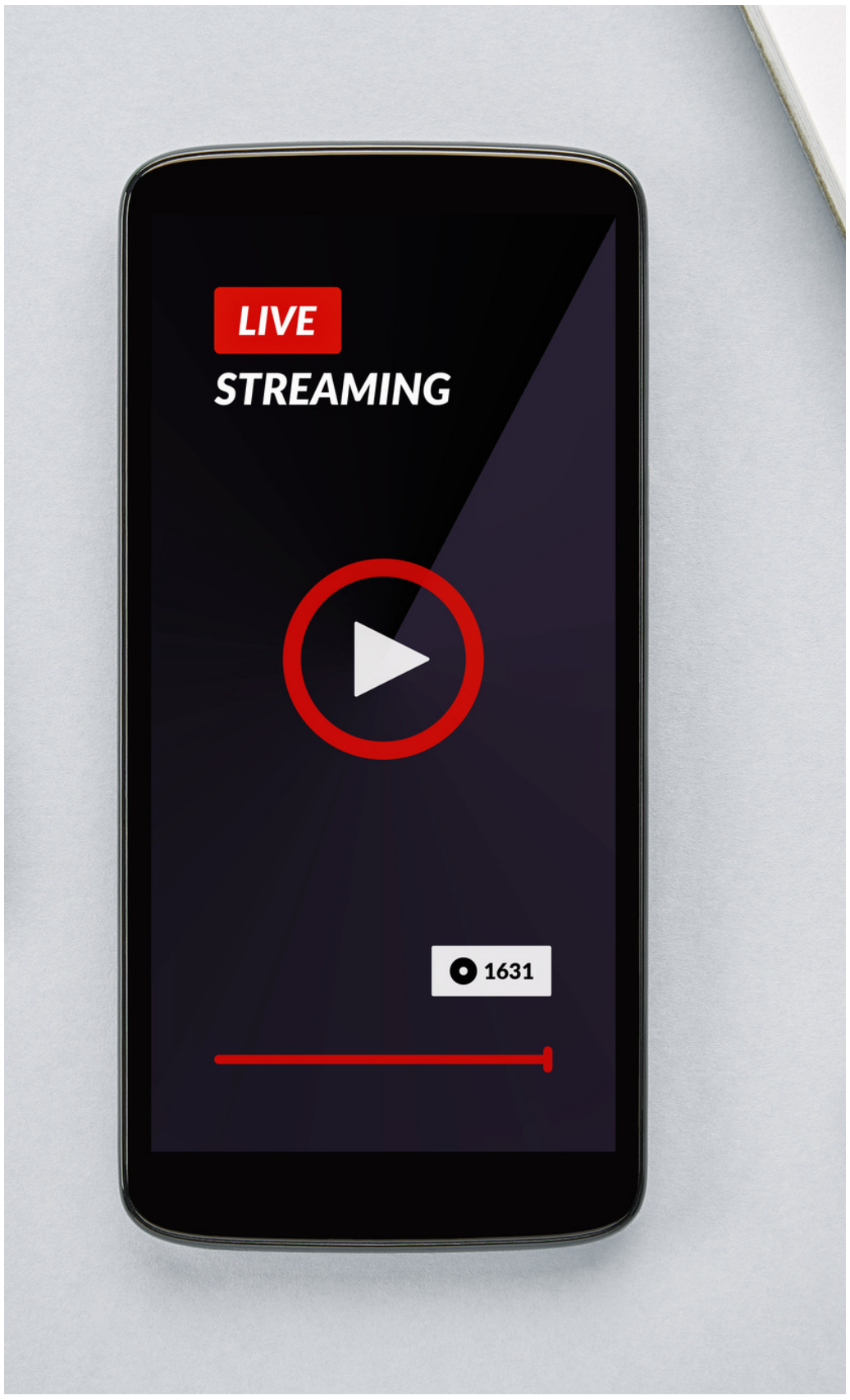

Topic for discussion:

Media Streaming using ExoPlayer in Android

Presentation by Rohan Pambhar
(Android Engineer)

Contents

- What is Media Streaming in Android?
- Types of Media Streaming supported.
- How to integrate Exoplayer into the Android application?

What is Media Streaming in Android?

- Media streaming is an audio/video file that can be watched from any Android device in both ways, remotely and locally.

- The media files can be watched from the internet, or you can watch them locally by downloading the media file to your Android devices.

- You can also watch live media streaming, e.g., Live News, Live Sports games, and other similar live events.

- You can also do video calls on any android app, which is also a good example of live streaming.

Types of Media Streaming supported

1. **DASH**
    - The full form is **D**ynamic **A**daptive **S**treaming over **H**TTP.
    - It uses an adaptive bitrate streaming technique to deliver high-quality content from HTTP web servers.
    - It breaks the media content into small segments (i.e., **MPEG- DASH** content**)** and maintains the sequence of those segments to deliver them from HTTP web servers.
    - Those segments are also converted into bit rates such as 420p, 720p, 2K, 4K, and many more to adapt the bitrates of the video based on the internet's bandwidth.
    -
2. **HLS**
    - The full form is **H**TTP **L**ive **S**treaming.
    - It was developed by Apple Inc.
    - It is an HTTP-based adaptive bitrate streaming communication protocol.
    - HLS also breaks the overall stream into a sequence of small HTTP-based file downloads and sends them to the client using an extended m3u8 playlist.

3. **SmoothStreaming**
    - Smooth Streaming is an IIS Media Services extension that enables adaptive media streaming to clients over HTTP.
    - It is developed by Microsoft.
    - This technology is based on HTTP and MP4 file format standards.

4. **Progressive download**
    - A progressive download transfers digital media files from a server to a client, typically using the HTTP protocol when initiated from a computer.
    - The consumer may begin playback of the media before the download is complete.
    - The key difference between streaming media and progressive download is how the digital media data is received and stored by the end-user device accessing the digital media.

5. **RTSP**
    - The Real Time Streaming Protocol (RTSP) is an application-level network protocol designed for multiplexing and packetizing multimedia transport streams (such as interactive media, video, and audio) over a suitable transport protocol.
    - RTSP is used in entertainment and communications systems to control streaming media servers.
    - The protocol is used for establishing and controlling media sessions between endpoints.
    - Clients of media servers issue commands such as play, record, and pause, to facilitate real-time control of the media streaming from the server to a client (video on demand) or from a client to the server (voice recording).

How to integrate ExoPlayer library in to Android application?

Add ExoPlayer modules:

The easiest way to get started using ExoPlayer is to add it as a Gradle dependency in the ***build.gradle*** file of your app module. The following will add a dependency to the full library:

```
"com.google.android.exoplayer:exoplayer:${versions.exoplayer}",
```

where 2.X.X is your preferred version (the latest version can be found by consulting the **release notes**).

As an alternative to the full library, you can depend on only the library modules you need. For example, the following will add dependencies on the Core, DASH, and UI library modules, as might be required for an app that only plays DASH content:

```
exoplayerCore: "com.google.android.exoplayer:exoplayer-core:${versions.exoplayer}",
exoplayerDash: "com.google.android.exoplayer:exoplayer-dash:${versions.exoplayer}",
exoplayerUi  : "com.google.android.exoplayer:exoplayer-ui:${versions.exoplayer}",
exoplayerHls : "com.google.android.exoplayer:exoplayer-hls:${versions.exoplayer}"
```

Usage of ExoPlayer:

You can create an ExoPlayer instance using **ExoPlayer.Builder**, which provides a range of customization options.
The code below is the simplest example of creating an instance.

```
// Initialize the player
val player = ExoPlayer.Builder(requireContext()).build()

// Bind the player to the view.
viewBinding.videoPlayerView.player = player

// Build the media item.
val mediaItem: MediaItem = MediaItem.fromUri(videoUri)

// Set the media item to be played.
player.setMediaItem(mediaItem)

// Prepare the player.
player.prepare()

// Start the playback.
player.play()
```

Make sure to release the instance of **exoplayer,** when your view get destroyed.

Based on types of Media Streamings:

You can also play different media streamings such as HLS, DASH, Progressive, and many others. For that, you have to initialize the media source accordingly. Here are a couple of examples given below.

```
private fun getHlsMediaSource(video: Video): HlsMediaSource {
    val dataSourceFactory: DataSource.Factory = DefaultHttpDataSource.Factory()
    val mediaItem = getMediaItem(video)
    return HlsMediaSource.Factory(dataSourceFactory).createMediaSource(mediaItem)
}

private fun getDashMediaSource(video: Video): DashMediaSource {
    val dataSourceFactory: DataSource.Factory = DefaultHttpDataSource.Factory()
    val mediaItem = getMediaItem(video)
    return DashMediaSource.Factory(dataSourceFactory)
        .createMediaSource(mediaItem)
}

private fun getMediaItem(video: Video): MediaItem {
    val mediaItem = MediaItem.Builder()
    val drmUuid = Util.getDrmUuid(video.drmType)
    mediaItem.setUri(video.streamingURL)
    if (drmUuid != null) {
        mediaItem.setDrmConfiguration(
            DrmConfiguration.Builder(drmUuid)
                .setLicenseUri(video.drmLicenseServer)
                .setLicenseRequestHeaders(ImmutableMap.copyOf(video.drmHeaderMessage))
                .build()
        )
    }
    return mediaItem.build()
}
```

GitHub Repository Link:

https://github.com/itAgenturen/media-stream-with-exoplayer

it Agenturen