







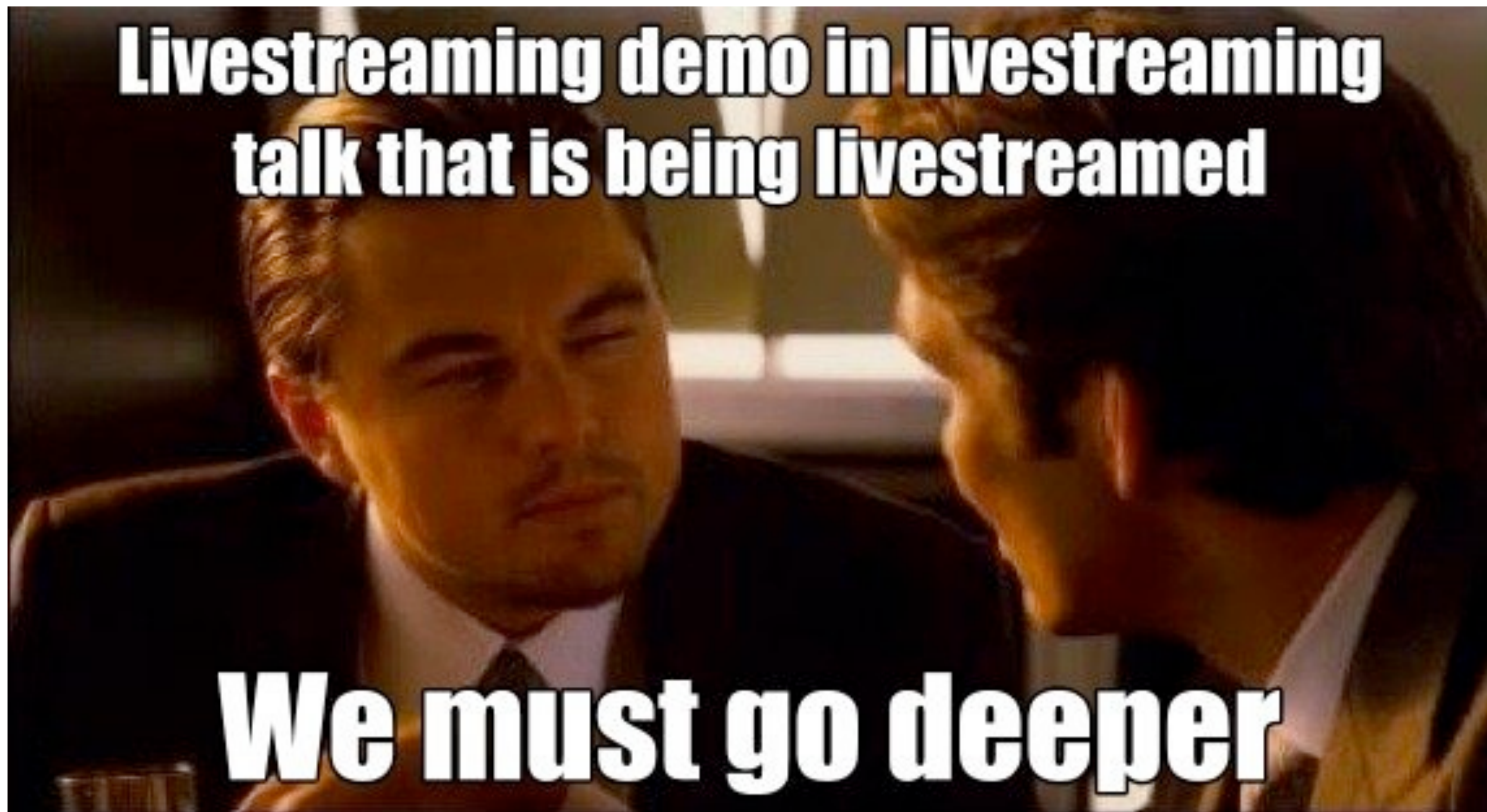
Broadcast Yourself!

Using the YouTube Live APIs to stream to the world

Satyajeet Salgar
PM, YouTube Live

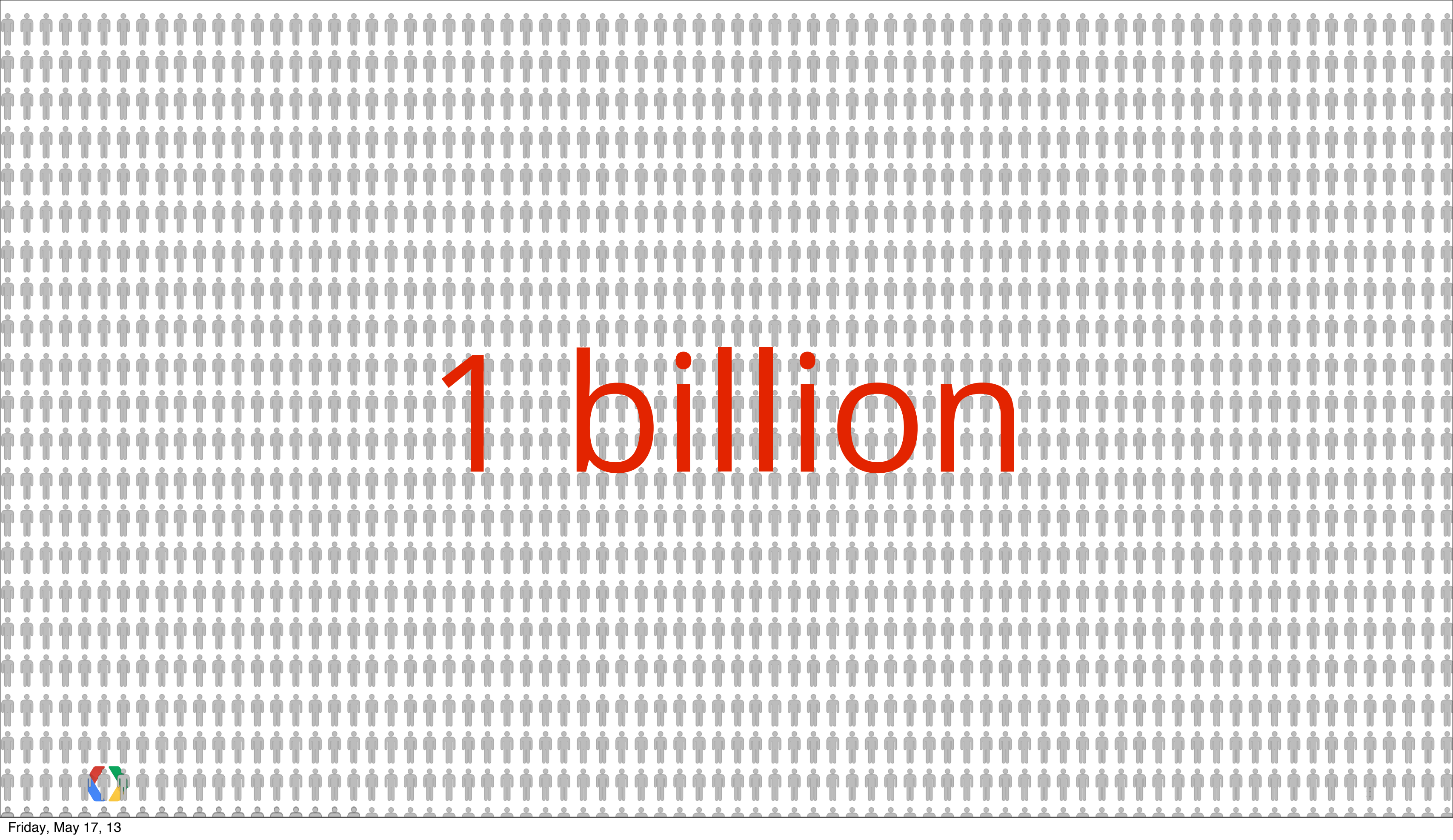
Ibrahim Ulukaya
Developer Relations, YouTube





You  **Tube**





1 billion



6 billion
hours/month



25% on mobile



You Tube







Live on YouTube

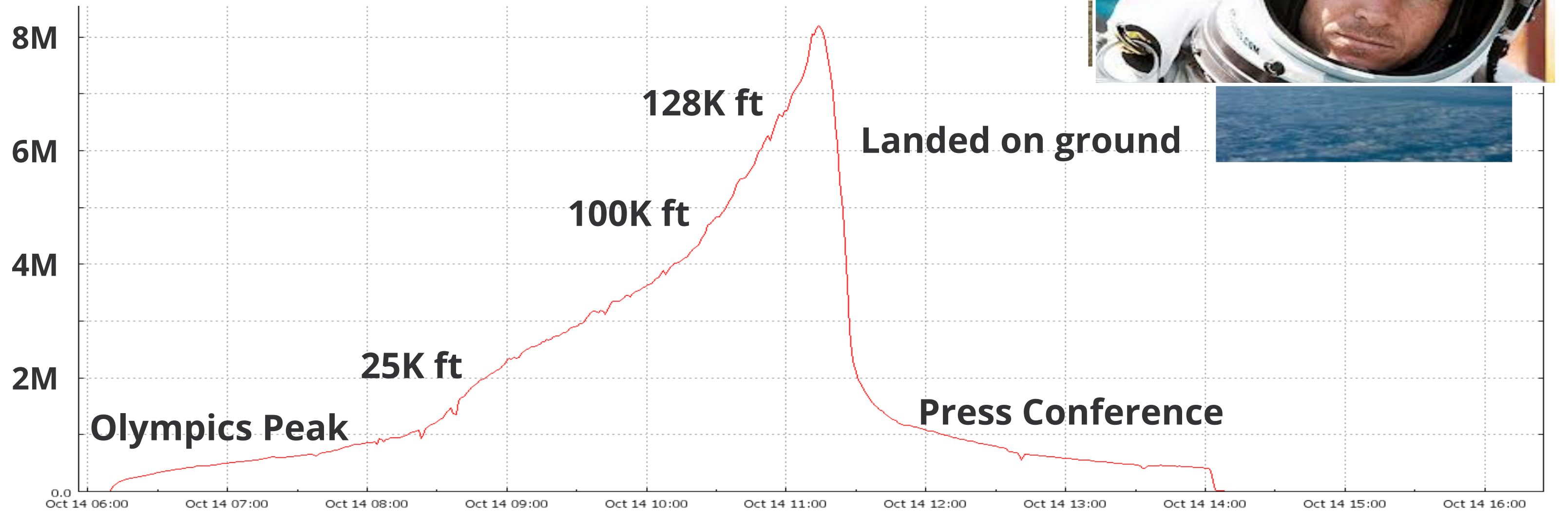


0:52 / 1:30



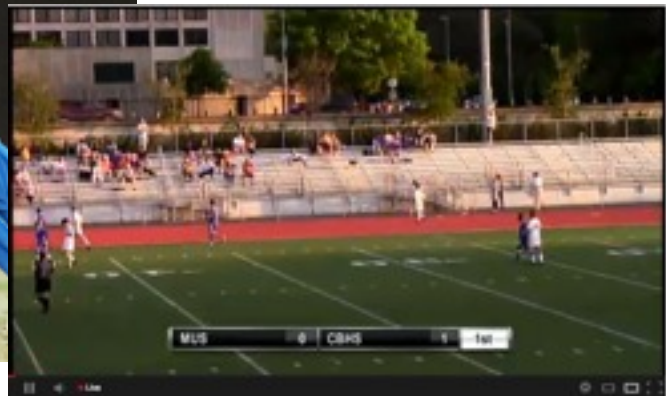
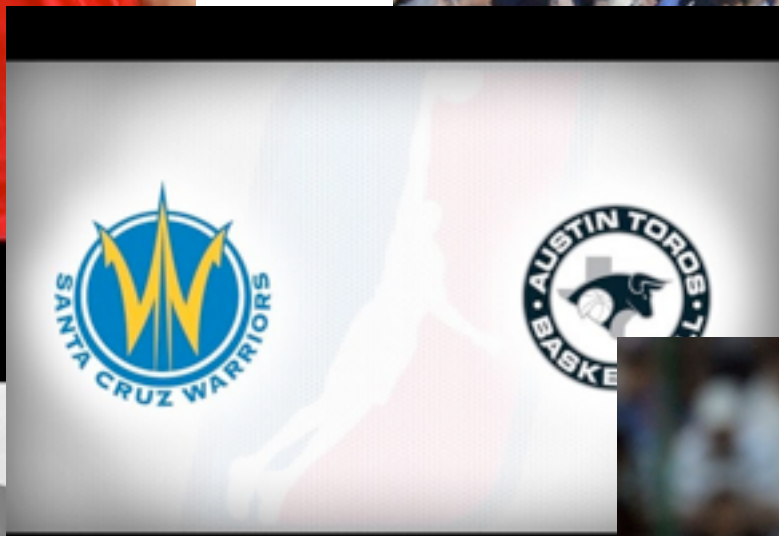
HD



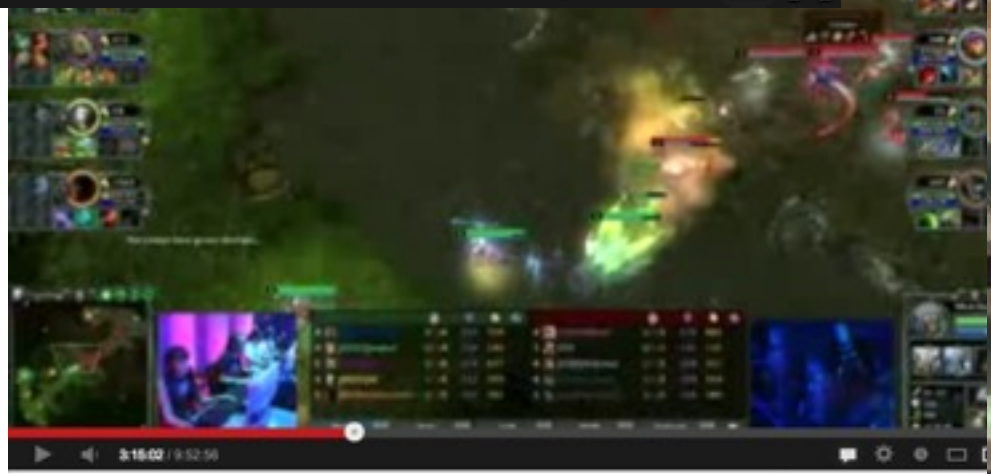


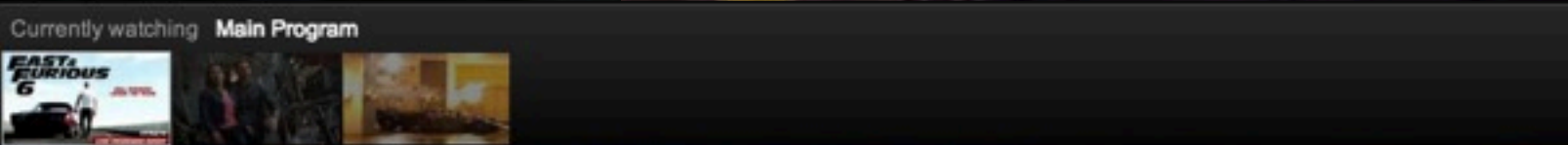
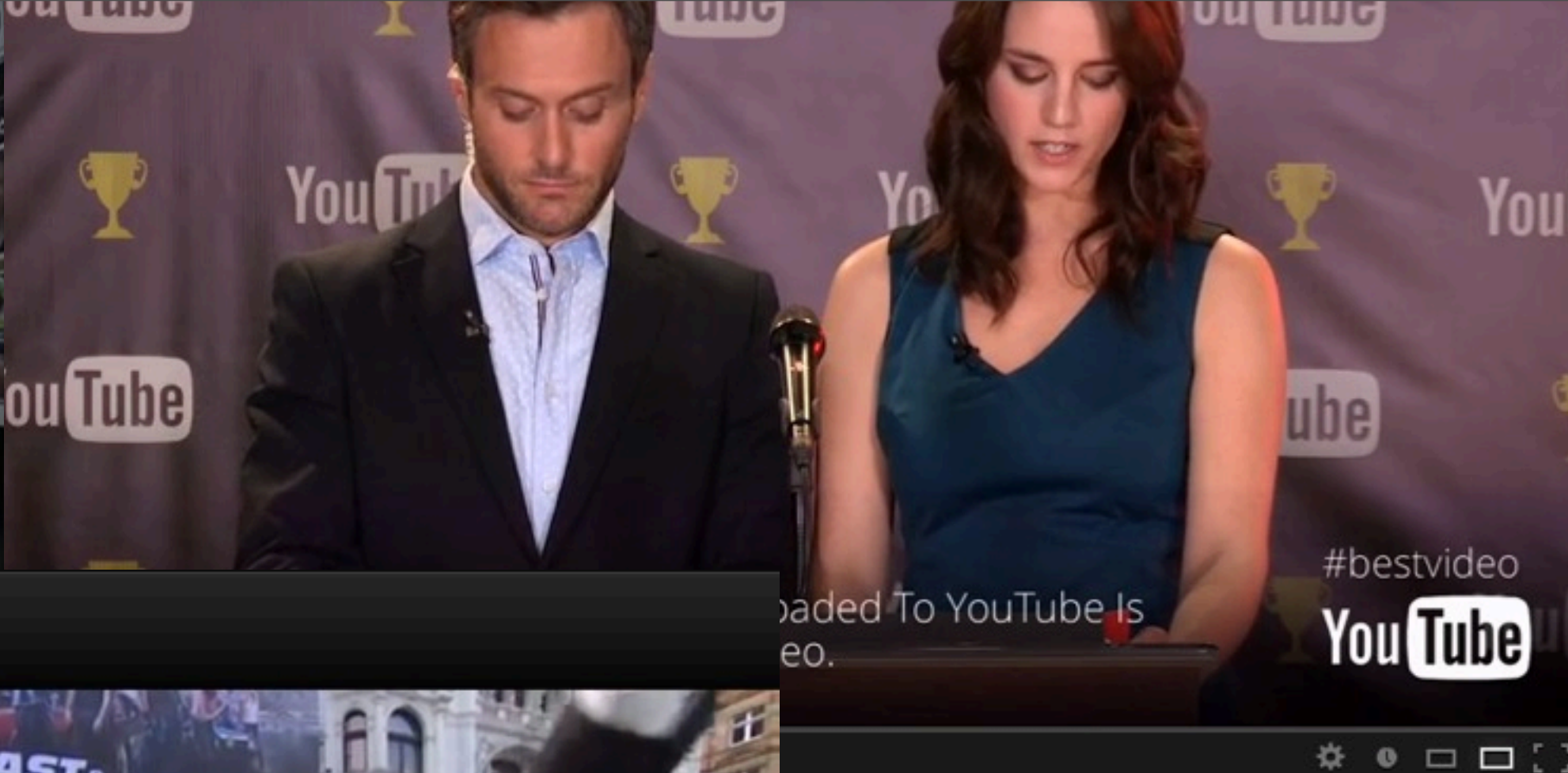
52M playbacks; 8.2M peak concurrent

















The YouTube Live Platform and APIs



Connect viewers with what's
happening now



YouTube offers a powerful live streaming solution



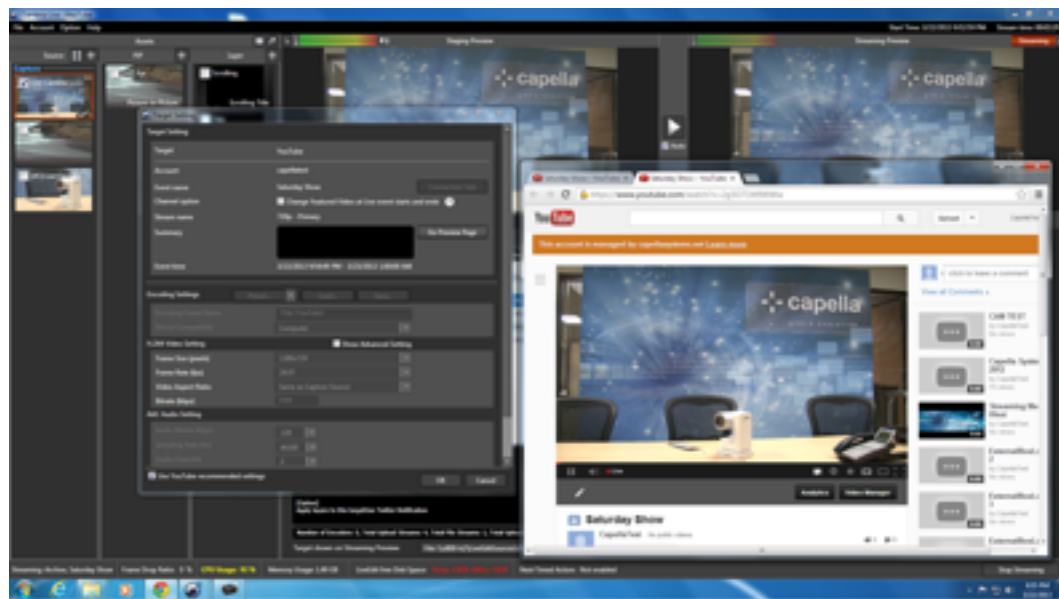
YouTube offers a powerful live streaming solution

- Scale
- Quality
- Transcoding in the cloud
- Multiple Camera angles
- Adaptive bit rate
- Ad Insertion
- Closed Caption

.... and many more



The YouTube Live API lets Partners and Developers build incredible live experiences and enabling tools



Capella



ElGato Game Capture HD



RadiLine

.... and many more



Access to the YouTube Live Platform is expanding making it more compelling to develop for

Live Events



You can stream live events. [Learn more](#)







YouTube Live Streaming API

<https://developers.google.com/youtube/v3/live/>

Agenda



API Overview



Live Streaming Settings



Java Sample



Android Demo



YouTube Live Streaming API



YouTube Live Streaming API

1. Schedule, Update, and Manage live events.



YouTube Live Streaming API

1. Schedule, Update, and Manage live events.
2. Associate events with video streams



YouTube Live Streaming API

1. Schedule, Update, and Manage live events.
2. Associate events with video streams
3. Insert Ads into your live broadcasts. (YouTube content partners)



API Concepts

Featured Live Events



The Big Live Comedy Show
Starts: May 19, 2013

Sun 5:00 PM



LCS 2013 NA Summer Promotion Tournament D1 ...

LIVE NOW 9,564 watching



Show Music Core Live 20130511 - Comeback B1A4...

Starts: May 10, 2013

Fri 11:50 PM

Call of Duty: Black Ops 2



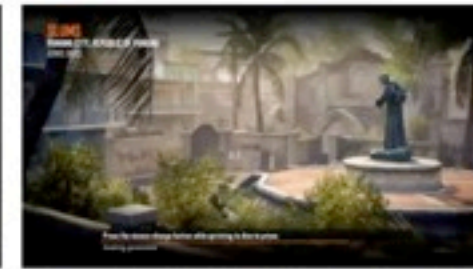
Majestic_Raz0r69 - Black Ops II Multiplayer Live Stream

LIVE NOW 2 watching



LILAXBRO415 - Black Ops II Multiplayer Live Stream

LIVE NOW



Thanatos1890 - Black Ops II Multiplayer Live Stream

LIVE NOW



SIRSAINT76 - Black Ops II Multiplayer Live Stream

LIVE NOW

Live - Sports



Nottingham v Bedford Blues - RFU Championship Semi-Final

LIVE NOW 437 watching



2013 U.S. Masters Swimming Spring National Champions

LIVE NOW 83 watching



2013 U.S. Masters Swimming Spring National Champions

LIVE NOW 84 watching



Canada vs. Germany Men's U19 World Floorball

LIVE NOW 68 watching



API Concepts

Broadcast

Featured Live Events



The Big Live Comedy Show
Starts: May 19, 2013

Sun 5:00 PM



LCS 2013 NA Summer Promotion Tournament D1 ...

LIVE NOW 9,564 watching



Show Music Core Live 20130511 - Comeback B1A4...

Starts: May 10, 2013

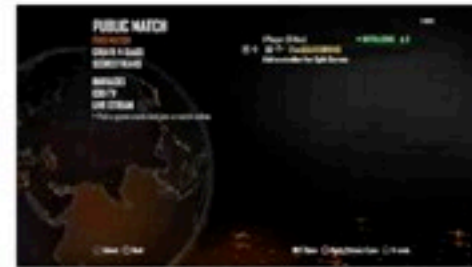
Fri 11:50 PM

Call of Duty: Black Ops 2



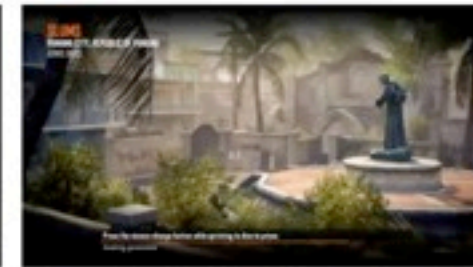
Majestic_Raz0r69 - Black Ops II Multiplayer Live Stream

LIVE NOW 2 watching



LILAXBRO415 - Black Ops II Multiplayer Live Stream

LIVE NOW



Thanatos1890 - Black Ops II Multiplayer Live Stream

LIVE NOW



SIRSAINT76 - Black Ops II Multiplayer Live Stream

LIVE NOW

Live - Sports



Nottingham v Bedford Blues - RFU Championship Semi-Final

LIVE NOW 437 watching



2013 U.S. Masters Swimming Spring National Champions

LIVE NOW 83 watching



2013 U.S. Masters Swimming Spring National Champions

LIVE NOW 84 watching



Canada vs. Germany Men's U19 World Floorball

LIVE NOW 68 watching



API Concepts



API Concepts

Stream



API Concepts



API Concepts

Broadcast

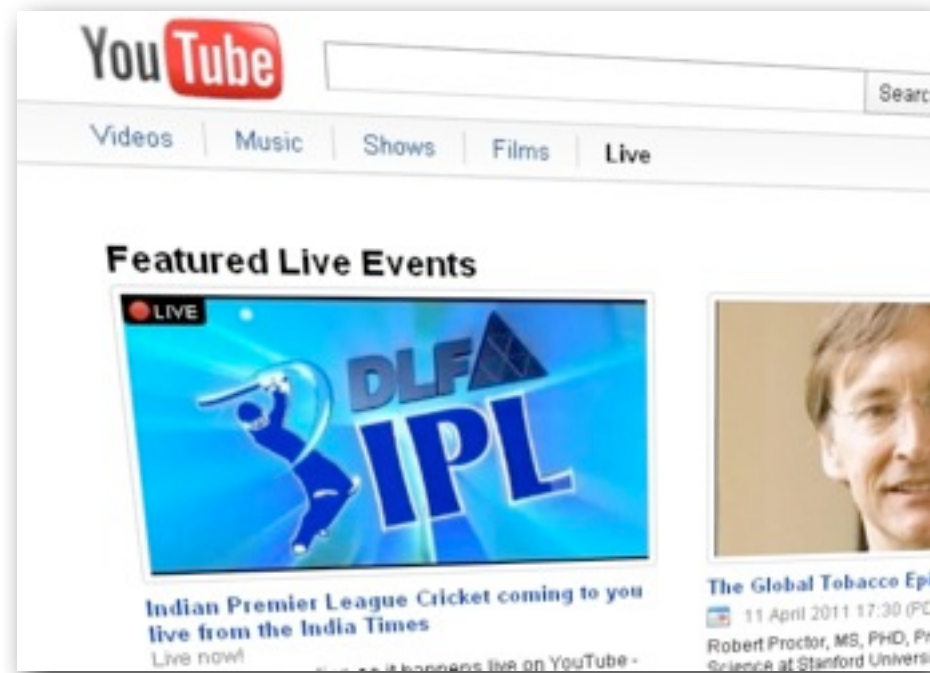


- ▶ Event metadata



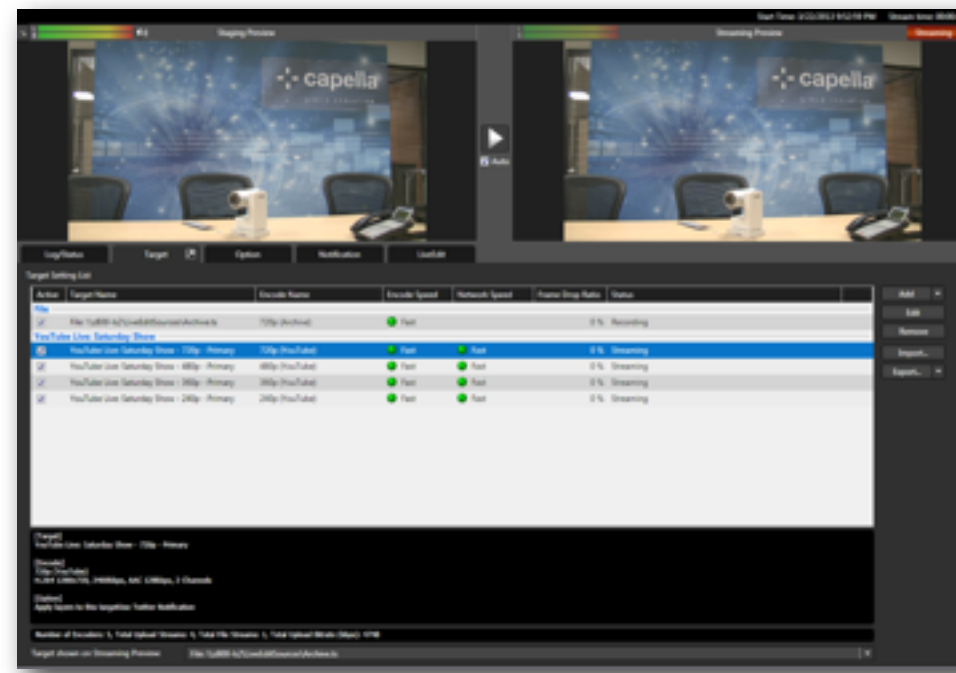
API Concepts

Broadcast



- ▶ Event metadata

Stream

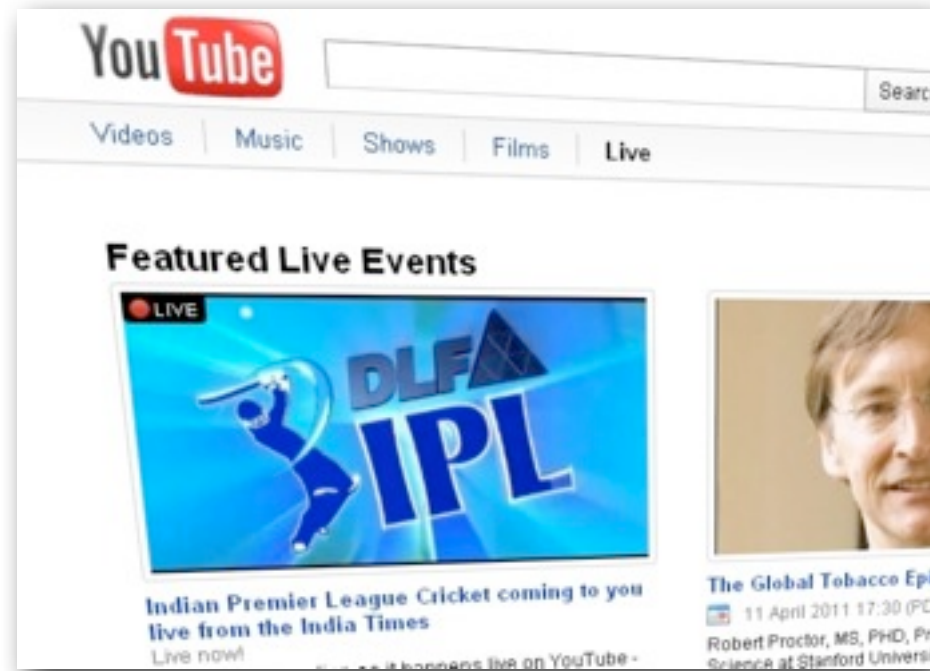


- ▶ Reusable ingestion points with streaming settings.



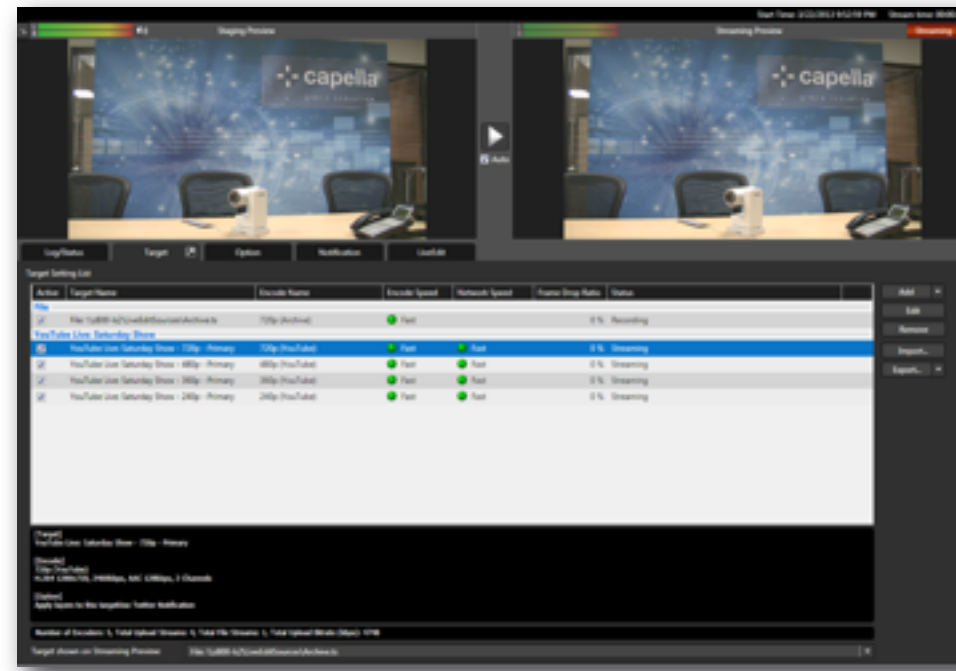
API Concepts

Broadcast



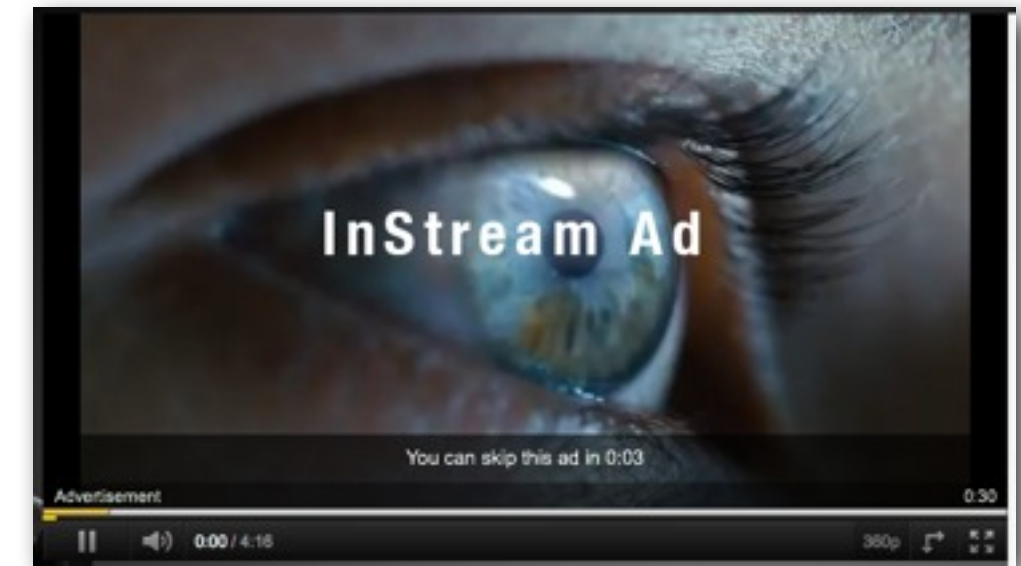
- ▶ Event metadata

Stream



- ▶ Reusable ingestion points with streaming settings.

Cuepoint



Only available to YouTube content partners.

- ▶ Built on Partner API

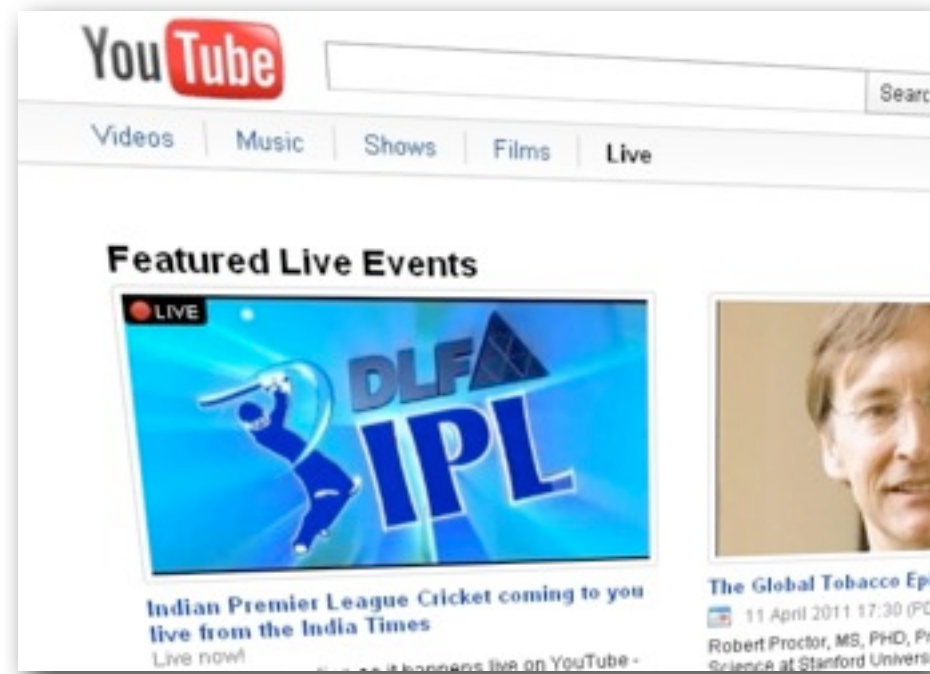


API Interaction



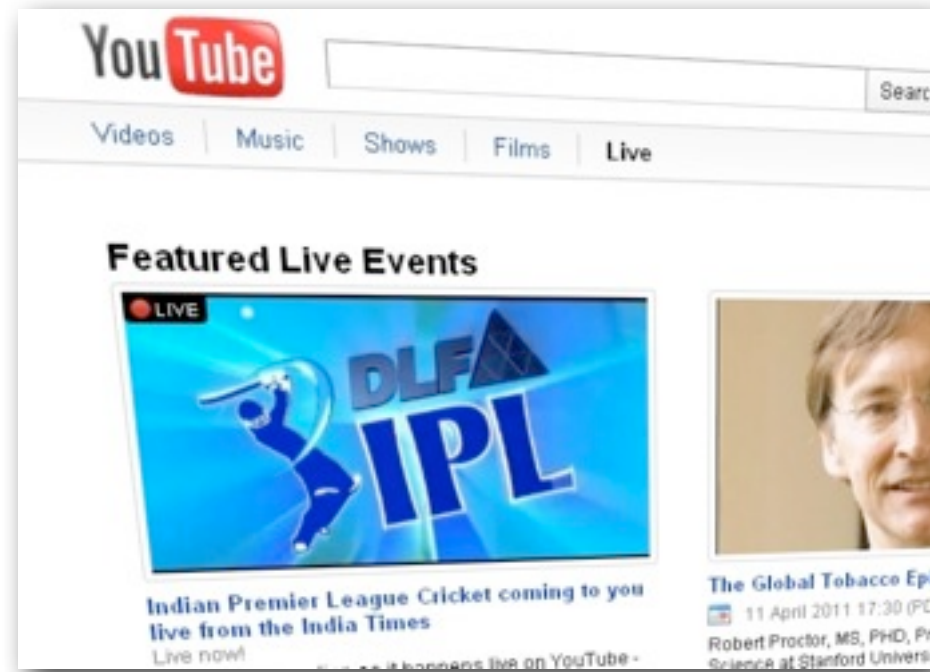
API Interaction

Broadcast



API Interaction

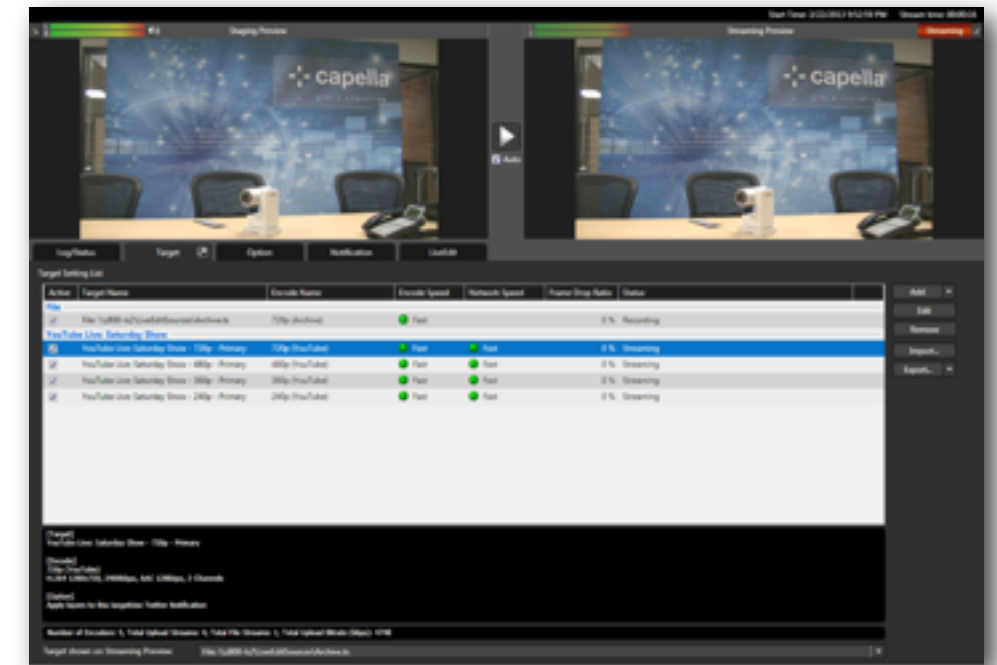
Broadcast



bind()



Stream

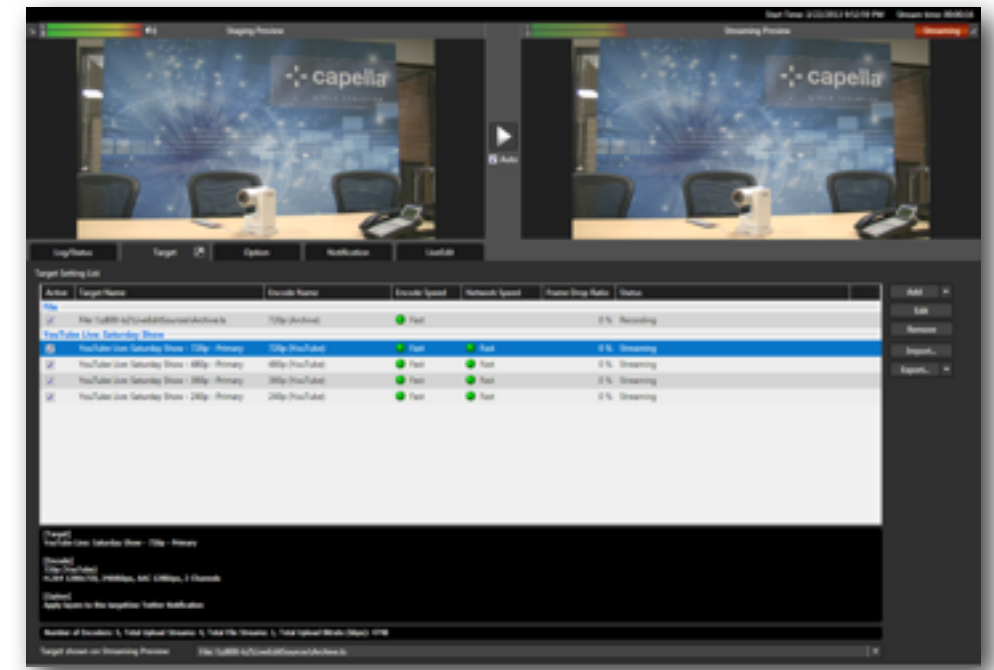


API Interaction

Broadcast



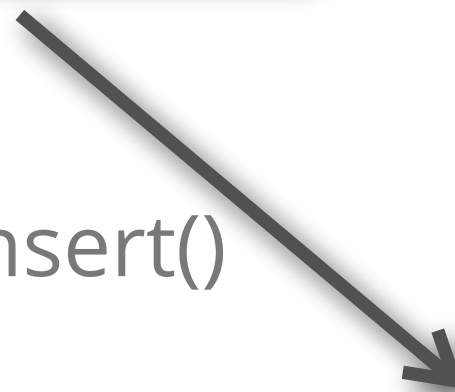
Stream



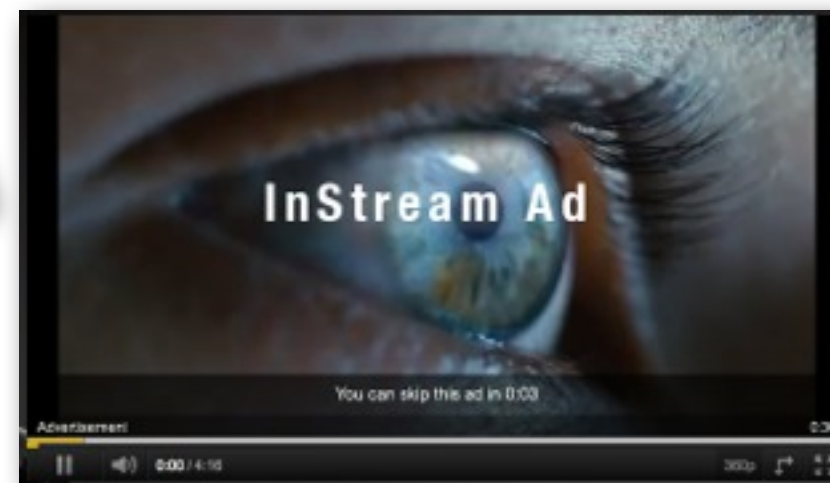
bind()



insert()



Cuepoint



1. Schedule, Update, and Manage live events.

Broadcast

Title

Description

Tags [?]

Starts:

 at :

Ends:

 at :

Privacy Settings [?]

Basic Info

Event Settings

Advanced Settings

Recording [?]

- Enable (Maximum 4 hours)
- Disable (Video will not be automatically recorded)

DVR

- Enable DVR (Maximum 36-hour DVR window)



Broadcast Life Cycle

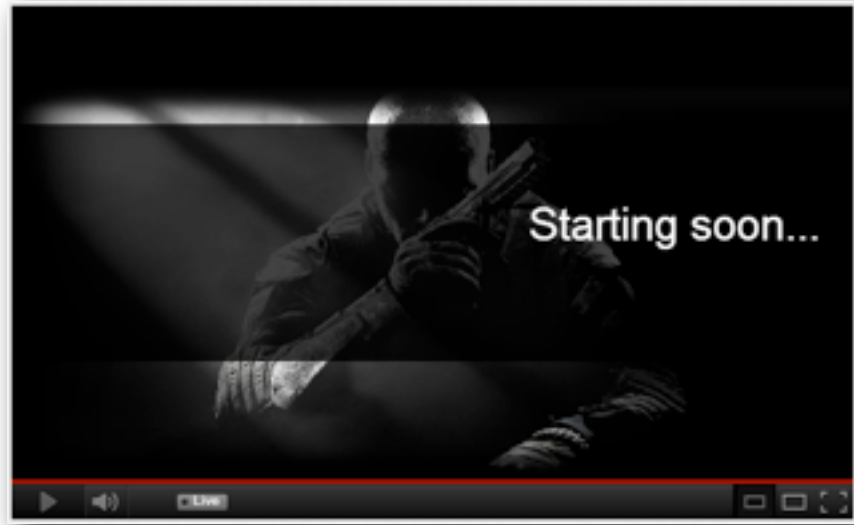


Broadcast Life Cycle

1.Created



Broadcast Life Cycle

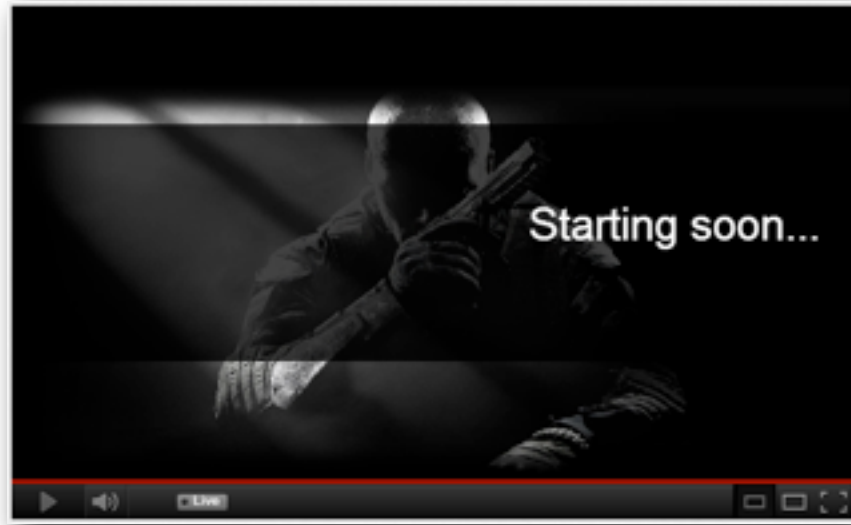


1.Created

2.Ready



Broadcast Life Cycle



1.Created

2.Ready

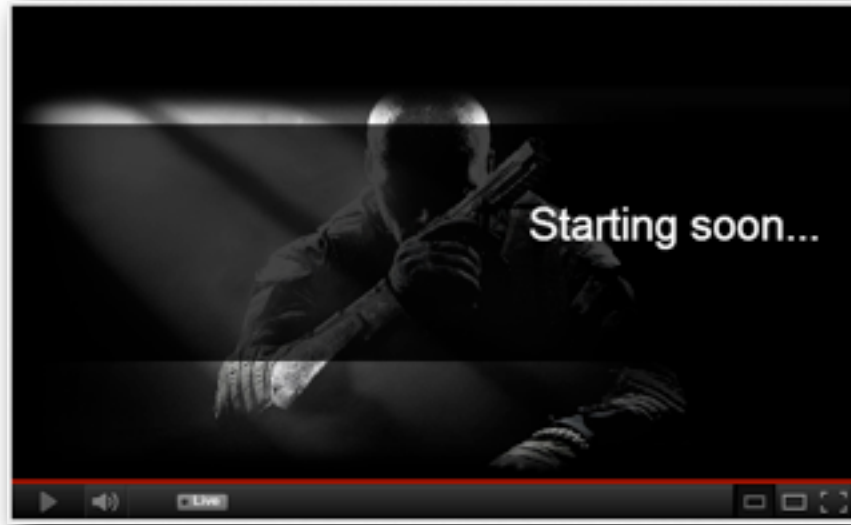
3.Preview



A screenshot of a web dashboard for stream management. At the top, there's a "Preview" button and a timer showing "00:00:00". Below that, a "STREAM STATUS" section shows "GOOD" in green text, with "480P STREAM" underneath. To the right, it says "May 6, 2013 03:22 PM (PDT)" and "The health is good." There are two tabs: "Manage" (selected) and "Analytics". At the bottom, there's a "PREVIEW" section with a video player area.



Broadcast Life Cycle

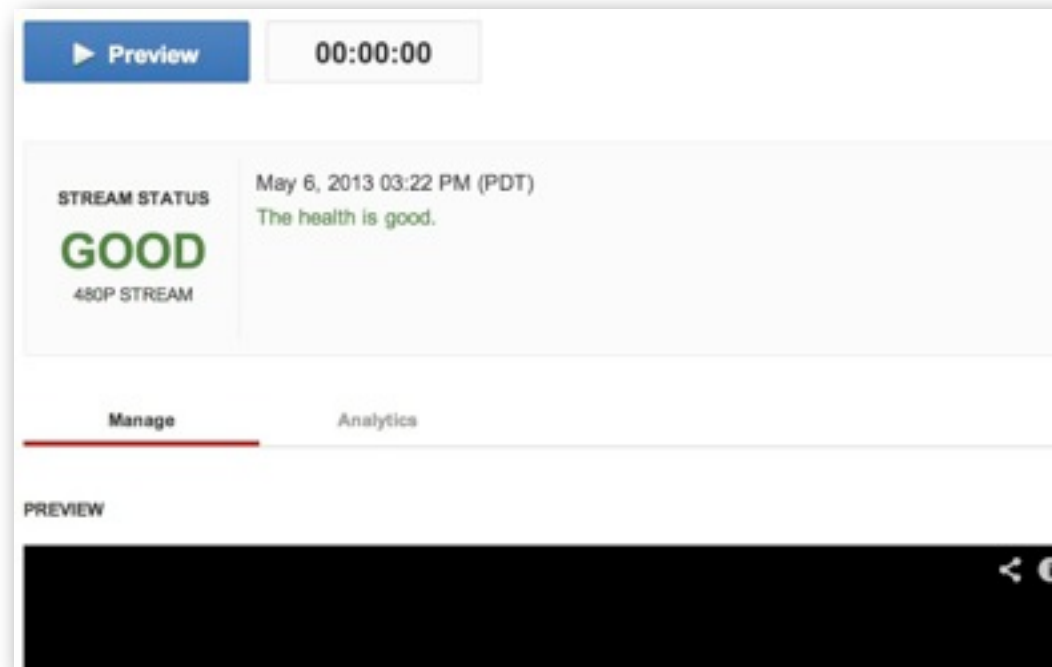


1.Created

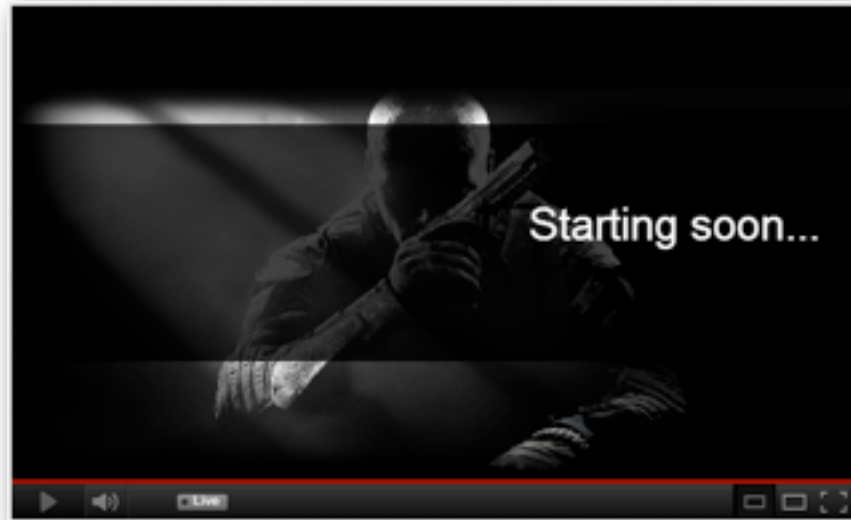
2.Ready

3.Preview

4.LIVE



Broadcast Life Cycle



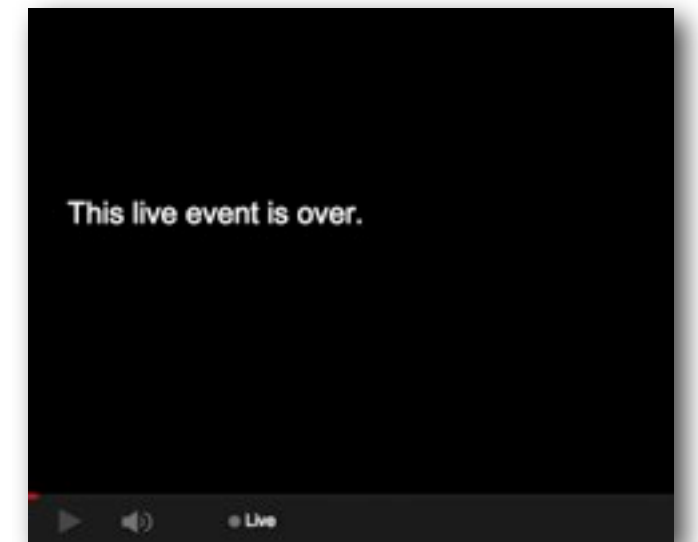
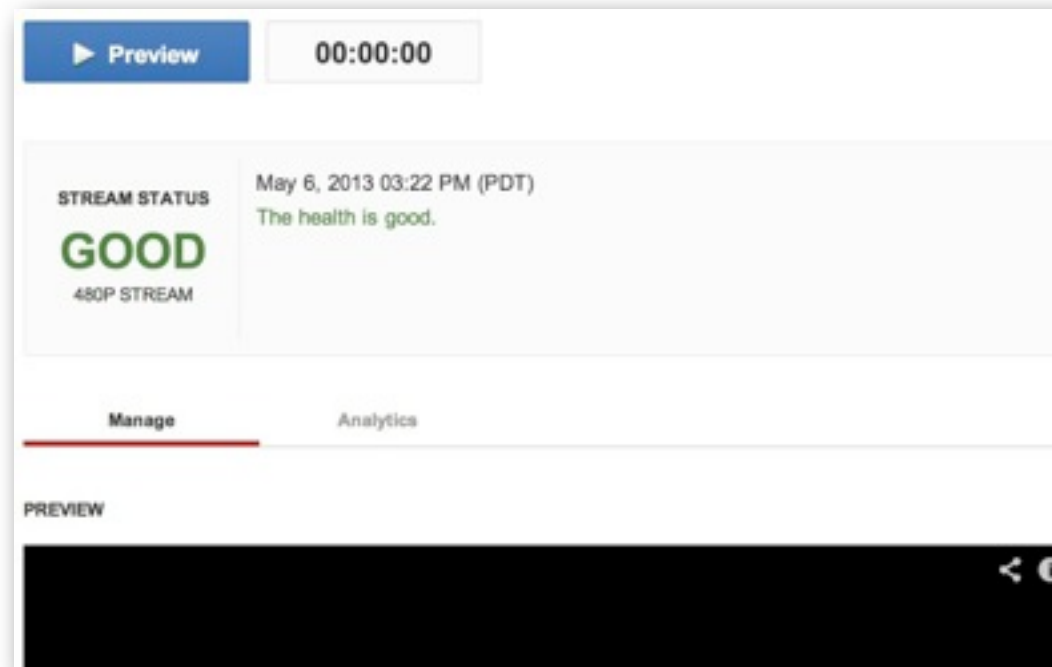
1.Created

2.Ready

3.Preview

4.LIVE

5.Completed



2. Associate events with video streams

Stream

Create stream

Name

128 character(s) remaining

Description

Maximum sustained bitrate that you can support:

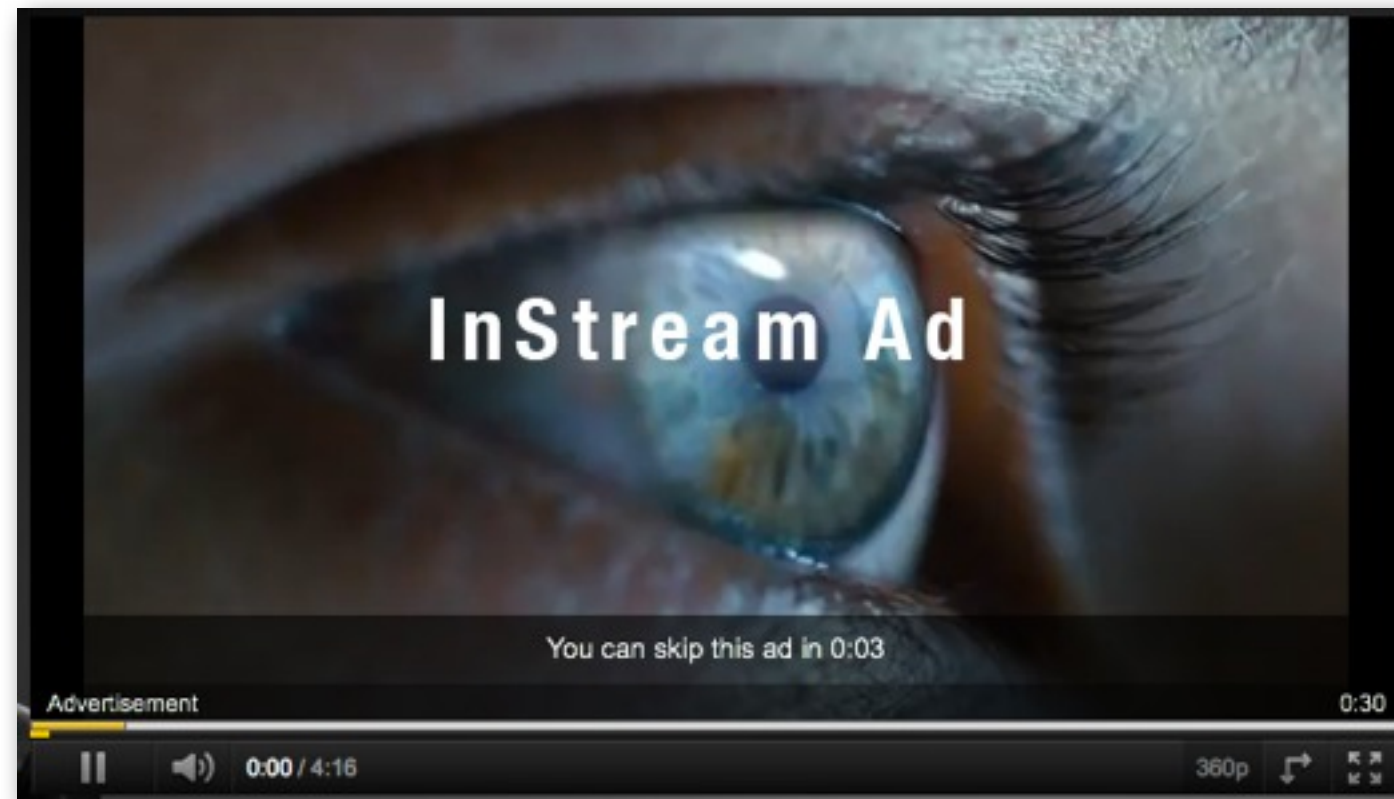
3000Kbps - 6000Kbps (1080p) ▼

- RTMP URL
- Stream Name



3. Insert Ads into your live broadcasts

Cuepoint



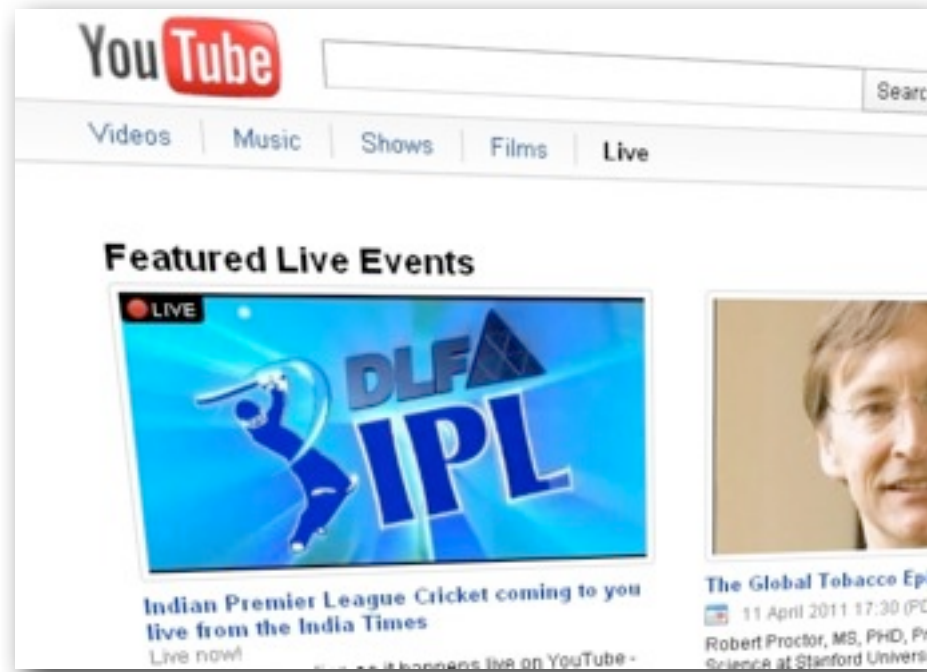
- Offset Time
- Duration



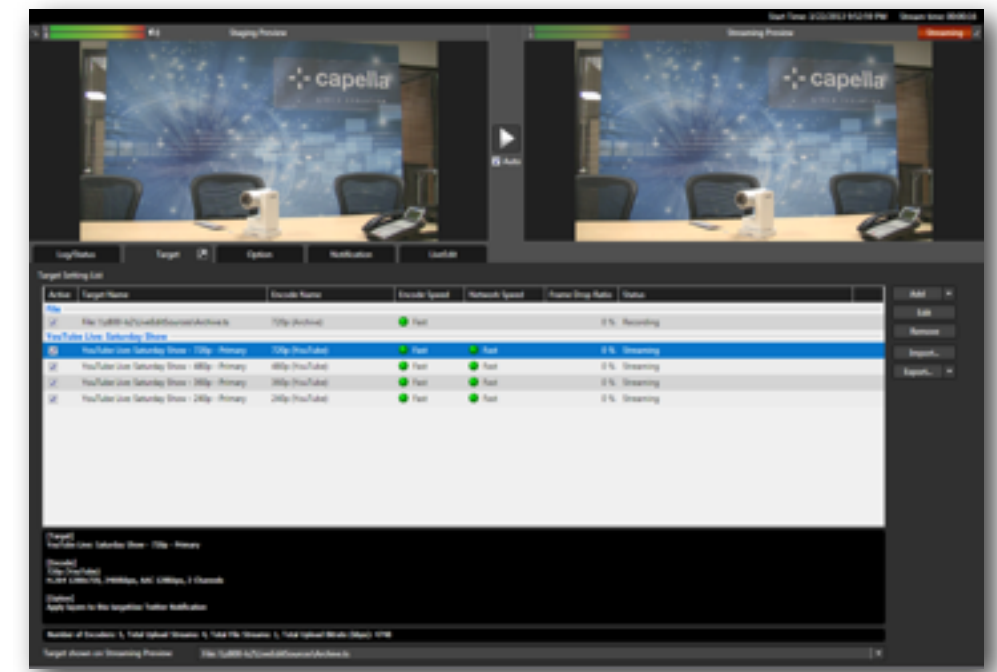
YouTube Live Streaming API Resource Interaction

Broadcast

Stream



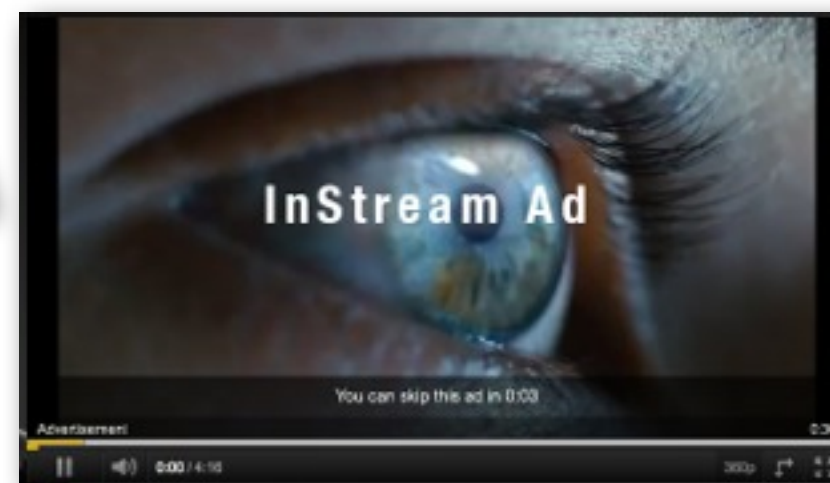
bind()



insert()



Cuepoint



Agenda



API Overview



Live Streaming Settings



Java Sample



Android Demo



Live Streaming Settings

► Resolutions and Bitrates

	240p	360p	480p	720p	1080p
Resolution	426 x 240	640 x 360	854 x 480	1280 x 720	1920 x 1080
Suggested Video Bitrate	400 Kbps	750 Kbps	1000 Kbps	2500 Kbps	4500 Kbps

**With the new platform, choose only one of the above.
All lower qualities will be automatically created from that one stream.**



Live Streaming Settings

▶ Encoder Settings

Protocol: RTMP Flash Streaming

Video codec: H.264, Main 4.1

Frame rate: 30 FPS

Keyframe frequency: 2 seconds

Audio codec: AAC-LC (Audio required)

Audio sample rate: 44.1 KHz

Audio bitrate: 128 Kbps stereo



Live Streaming Settings

▶ Encoder Settings

Protocol: RTMP Flash Streaming

Video codec: H.264, Main 4.1

Frame rate: 30 FPS

Keyframe frequency: 2 seconds

Audio codec: AAC-LC (Audio required)





Audio sample rate: 44.1 KHz

Audio bitrate: 128 Kbps stereo

Mobile streaming: Live events will be automatically available on **iOS & Android 4.0+** mobile devices via m.youtube.com as well as **Google TV**.



Agenda

-  **API Overview**
-  **Live Streaming Settings**
-  **Java Sample**
-  **Android Demo**







YouTube Live API Samples

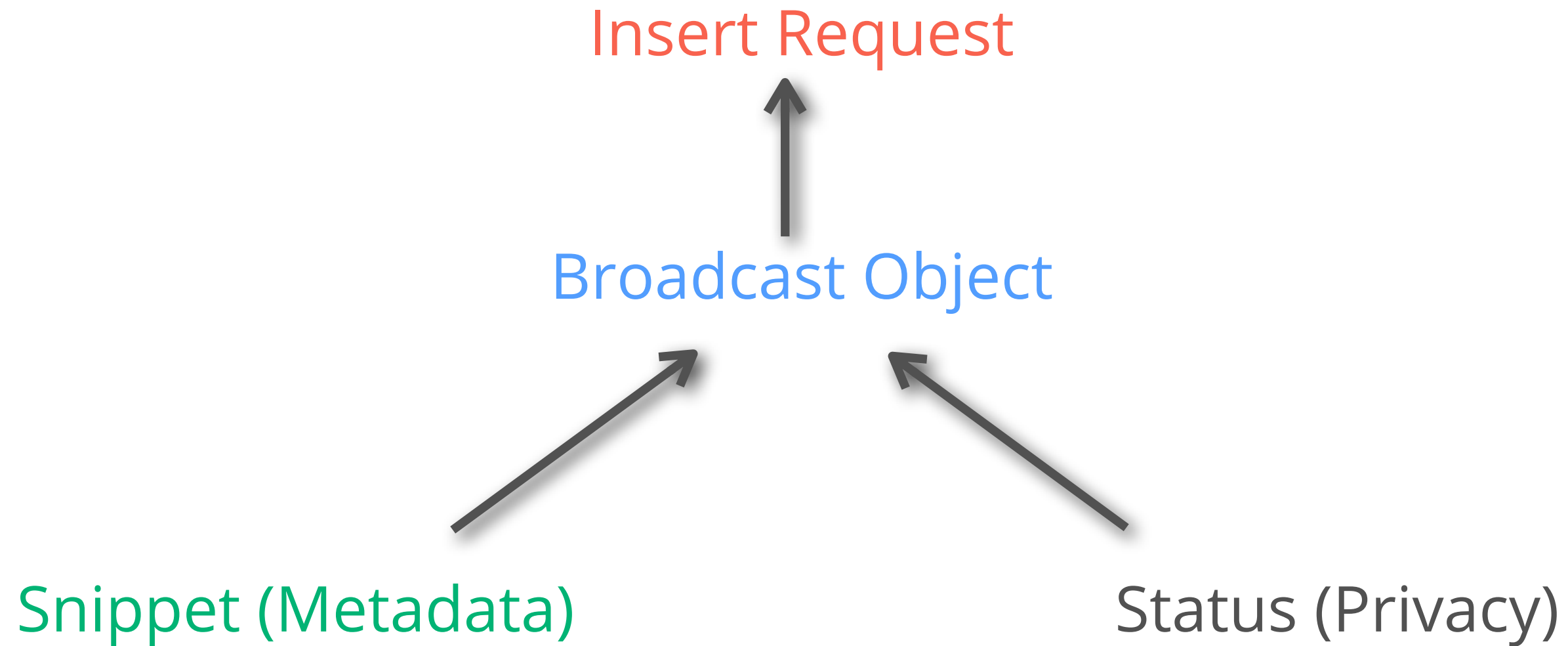
<https://code.google.com/p/youtube-api-samples/>

YouTube API v3 Client Libraries

- <https://developers.google.com/youtube/v3/libraries>



Create Broadcast (Event)



Create live event (broadcast)

▶ Create snippet with Title and Scheduled Time

```
LiveBroadcastSnippet broadcastSnippet = new LiveBroadcastSnippet();  
  
broadcastSnippet.setTitle(title);  
broadcastSnippet.setScheduledStartTime(new DateTime("2024-01-30T00:00:00.000Z"));  
broadcastSnippet.setScheduledEndTime(new DateTime("2024-01-31T00:00:00.000Z"));
```

Java



Create live event (broadcast)

▶ Create privacy status

Java

```
LiveBroadcastStatus status = new LiveBroadcastStatus();  
status.setPrivacyStatus("private");
```



Create live event (broadcast)

▶ Create broadcast object with Snippet and Status

Java

```
LiveBroadcast broadcast = new LiveBroadcast();  
  
broadcast.setKind("youtube#liveBroadcast");  
broadcast.setSnippet(broadcastSnippet);  
broadcast.setStatus(status);
```



Create live event (broadcast)

▶ Create Insert Broadcast request

Java

```
YouTube.LiveBroadcasts.Insert liveBroadcastInsert =  
    youtube.liveBroadcasts().insert("snippet,status", broadcast);
```



Create live event (broadcast)

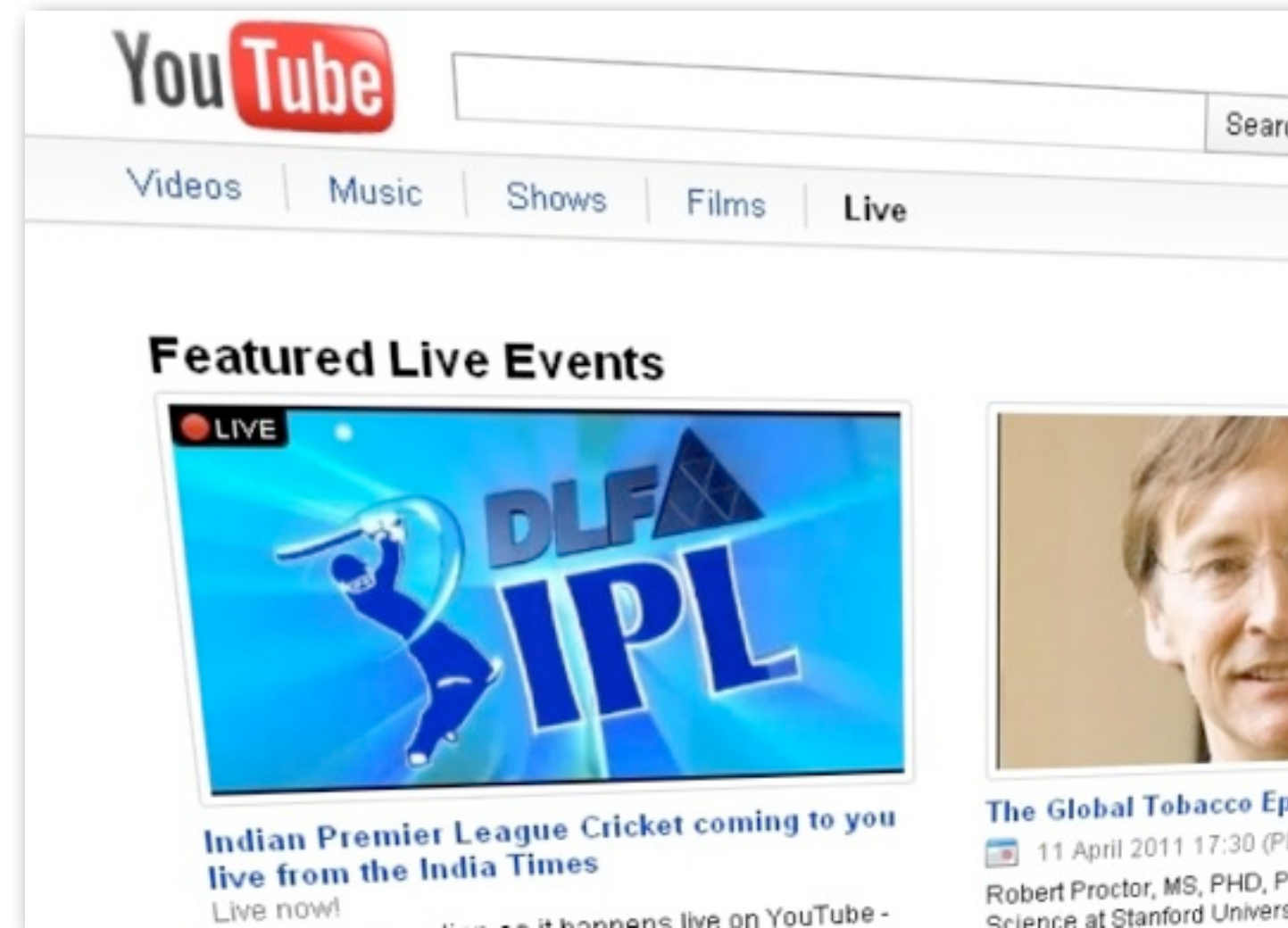
▶ Create Insert Broadcast request and Execute

Java

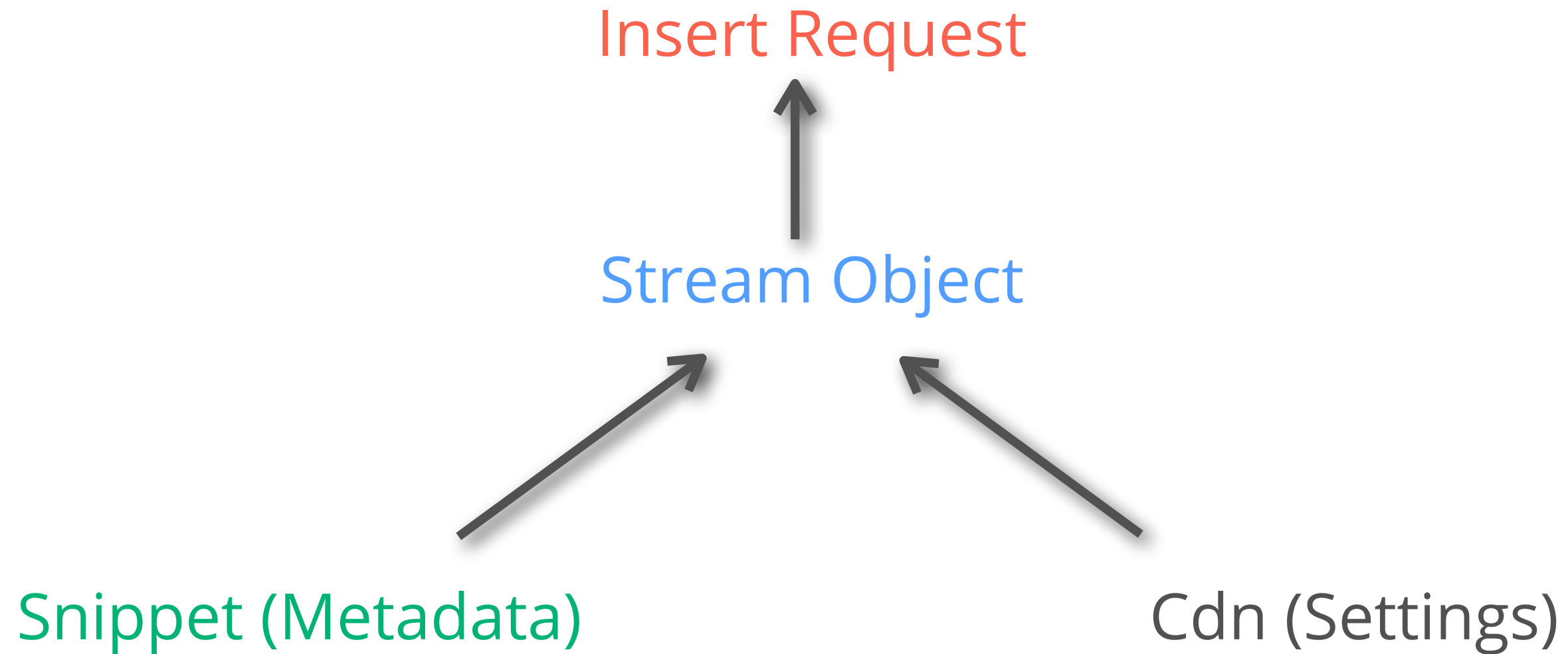
```
YouTube.LiveBroadcasts.Insert liveBroadcastInsert =  
    youtube.liveBroadcasts().insert("snippet,status", broadcast);  
  
LiveBroadcast returnedBroadcast = liveBroadcastInsert.execute();
```



We created the Broadcast (event)



Create Stream (Ingestion Point)



Create ingestion point (stream)

▶ Create snippet with Title

Java

```
LiveStreamSnippet streamSnippet = new LiveStreamSnippet();  
  
streamSnippet.setTitle(title);
```



Create ingestion point (stream)

▶ Create ingestion settings with Format and Type

Java

```
LiveStreamCdn cdn = new LiveStreamCdn();  
  
cdn.setFormat("1080p");  
cdn.setIngestionType("rtmp");
```



Create ingestion point (stream)

▶ Create Stream object with Snippet and Ingestion settings.

Java

```
LiveStream stream = new LiveStream();  
  
stream.setKind("youtube#liveStream");  
stream.setSnippet(streamSnippet);  
stream.setCdn(cdn);
```



Create ingestion point (stream)

▶ Create Insert Stream request

Java

```
YouTube.LiveStreams.Insert liveStreamInsert =  
    youtube.liveStreams().insert("snippet,cdn", stream);
```



Create ingestion point (stream)

▶ Create Insert Stream request, **execute**

Java

```
YouTube.LiveStreams.Insert liveStreamInsert =  
    youtube.liveStreams().insert("snippet,cdn", stream);  
  
LiveStream returnedStream = liveStreamInsert.execute();
```



Ingestion Info

Stream

Create stream

Name

128 character(s) remaining

Description

Maximum sustained bitrate that you can support:

3000Kbps - 6000Kbps (1080p) ▼

- **Ingestion Address**
- **Stream Name**



Create ingestion point (stream)

▶ Get ingestion info

Java

```
YouTube.LiveStreams.Insert liveStreamInsert =  
    youtube.liveStreams().insert("snippet,cdn", stream);  
  
LiveStream returnedStream = liveStreamInsert.execute();  
  
ingestionAddress = returnedStream.getCdn().getIngestionInfo().getIngestionAddress();  
streamName = returnedStream.getCdn().getIngestionInfo().getStreamName();
```



We created the Stream

Start Time: 3/22/2013 9:52:59 PM Stream time: 00:00:16

Staging Preview Streaming Preview

Log/Status Target Option Notification LiveEdit

Auto

Target Setting List

Active	Target Name	Encode Name	Encode Speed	Network Speed	Frame Drop Ratio	Status
<input checked="" type="checkbox"/>	File: \\u800-h2\Live\Edit\Source\Archive.ts	720p (Archive)	Fast		0 %	Recording
<input checked="" type="checkbox"/>	YouTube Live: Saturday Show					
<input checked="" type="checkbox"/>	YouTube Live: Saturday Show - 720p - Primary	720p (YouTube)	Fast	Fast	0 %	Streaming
<input checked="" type="checkbox"/>	YouTube Live: Saturday Show - 480p - Primary	480p (YouTube)	Fast	Fast	0 %	Streaming
<input checked="" type="checkbox"/>	YouTube Live: Saturday Show - 360p - Primary	360p (YouTube)	Fast	Fast	0 %	Streaming
<input checked="" type="checkbox"/>	YouTube Live: Saturday Show - 240p - Primary	240p (YouTube)	Fast	Fast	0 %	Streaming

[Target]
YouTube Live: Saturday Show - 720p - Primary

[Encode]
720p (YouTube)
6,254 1280x720, 2400kbps, AAC 128kbps, 2 Channels

[Option]
Apply servers to this target: Twitter Notification

Number of Encoders: 5, Total Upload Streams: 4, Total File Streams: 1, Total Upload Bitrate (Mbps): 47MB

Target shown on Streaming Preview: File: \\u800-h2\Live\Edit\Source\Archive.ts

Add
Edit
Remove
Import...
Export...

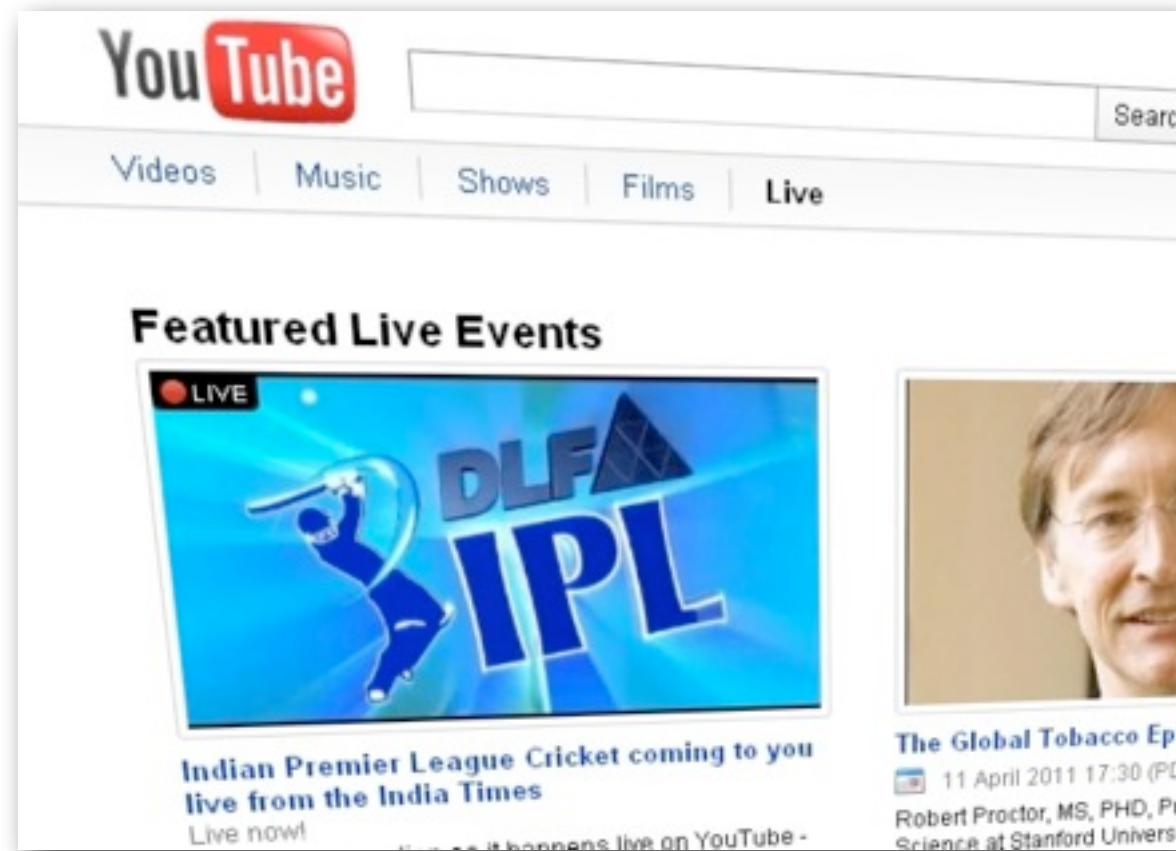


What's missing?

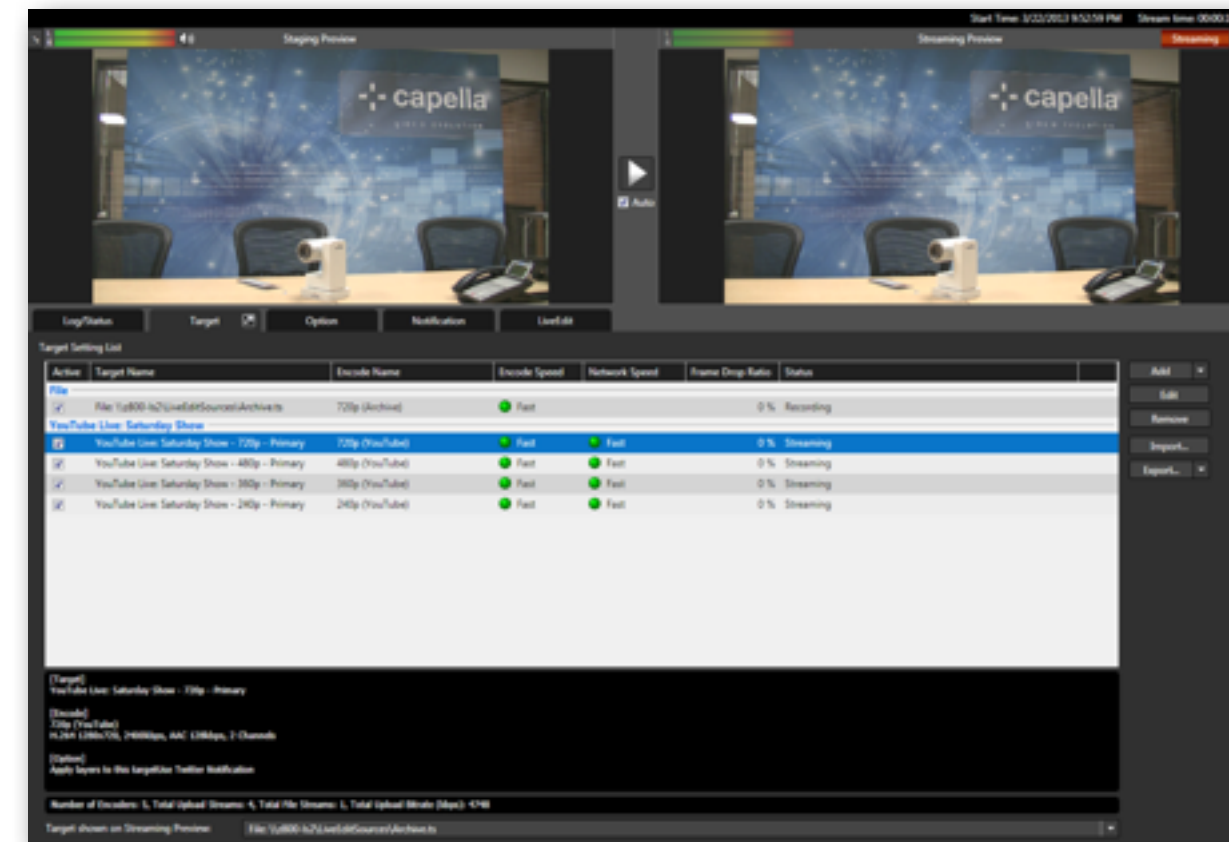


What's missing?

Broadcast



Stream



bind()



Connect broadcast to stream

▶ Create Bind Request

Java

```
YouTube.LiveBroadcasts.Bind liveBroadcastBind =  
    youtube.liveBroadcasts().bind(returnedBroadcast.getId(), "id,contentDetails");
```



Connect broadcast to stream

▶ Set stream id to bind

Java

```
YouTube.LiveBroadcasts.Bind liveBroadcastBind =  
    youtube.liveBroadcasts().bind(returnedBroadcast.getId(), "id,contentDetails");  
  
liveBroadcastBind.setStreamId(returnedStream.getId());
```



Connect broadcast to stream





▶ Execute Bind request

Java

```
YouTube.LiveBroadcasts.Bind liveBroadcastBind =  
    youtube.liveBroadcasts().bind(returnedBroadcast.getId(), "id,contentDetails");  
  
liveBroadcastBind.setStreamId(returnedStream.getId());  
  
returnedBroadcast = liveBroadcastBind.execute();
```

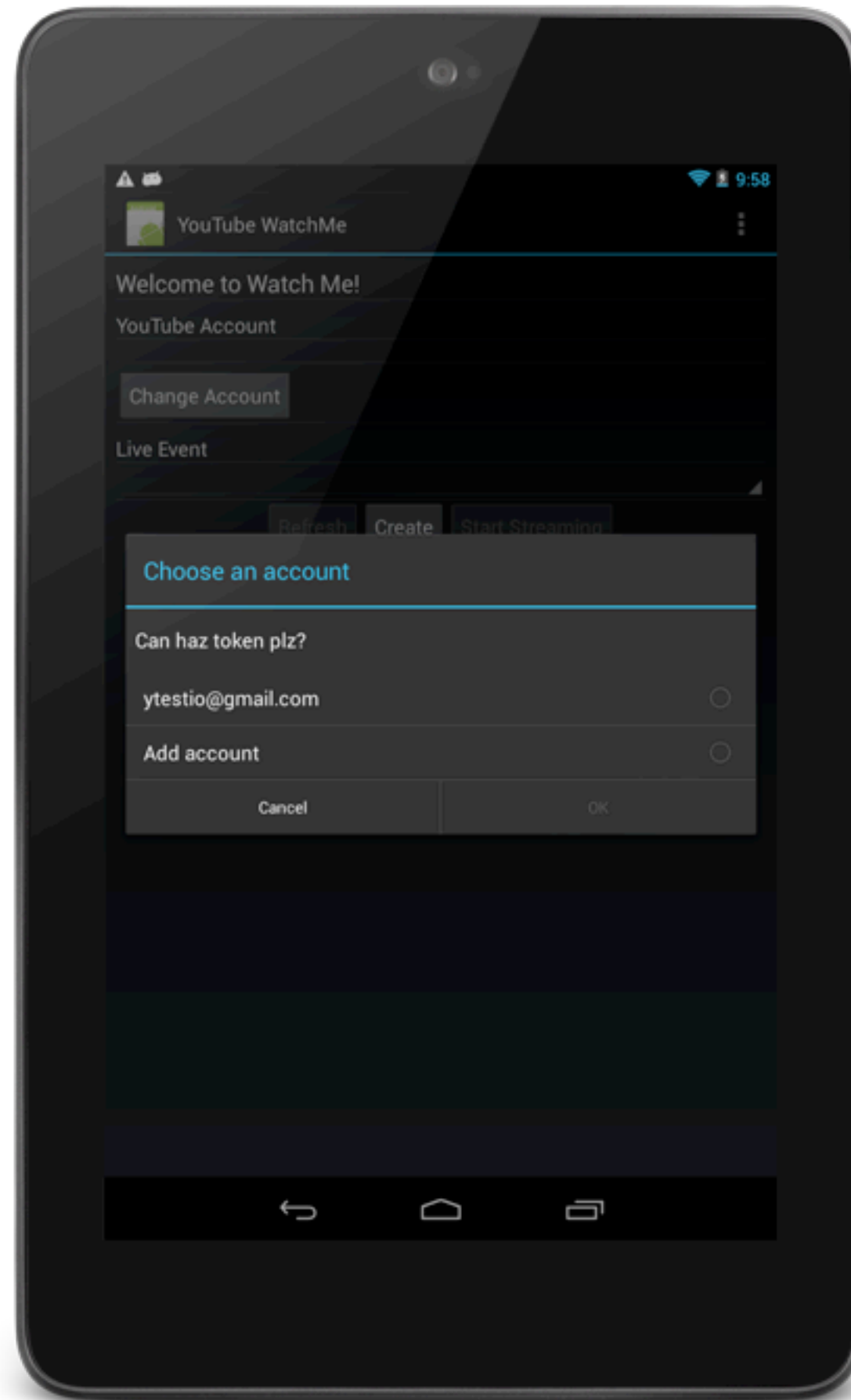


Agenda

-  **API Overview**
-  **Live Streaming Settings**
-  **Java Sample**
-  **Android Demo**

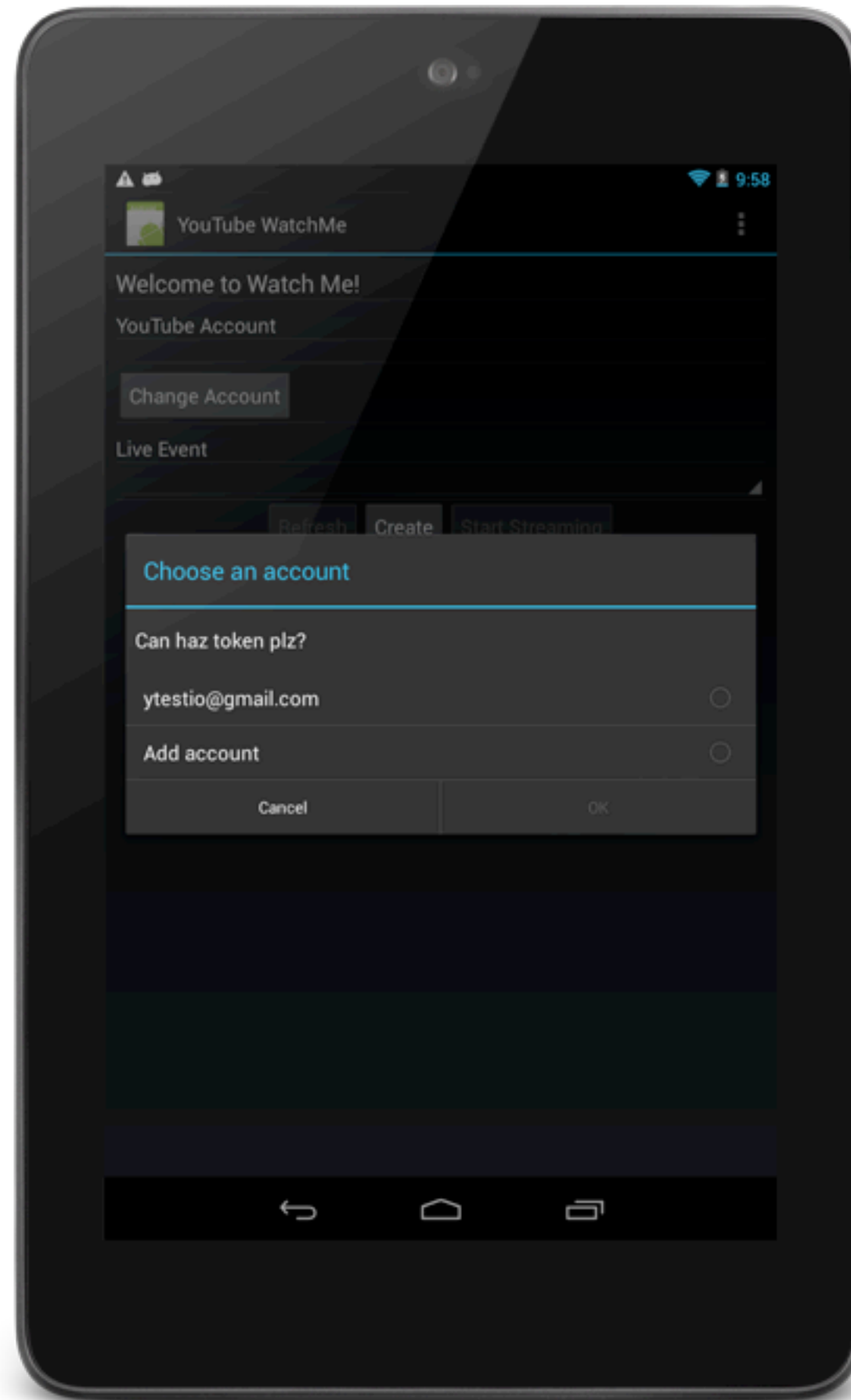


Android Demo



Android Demo

Pick Account

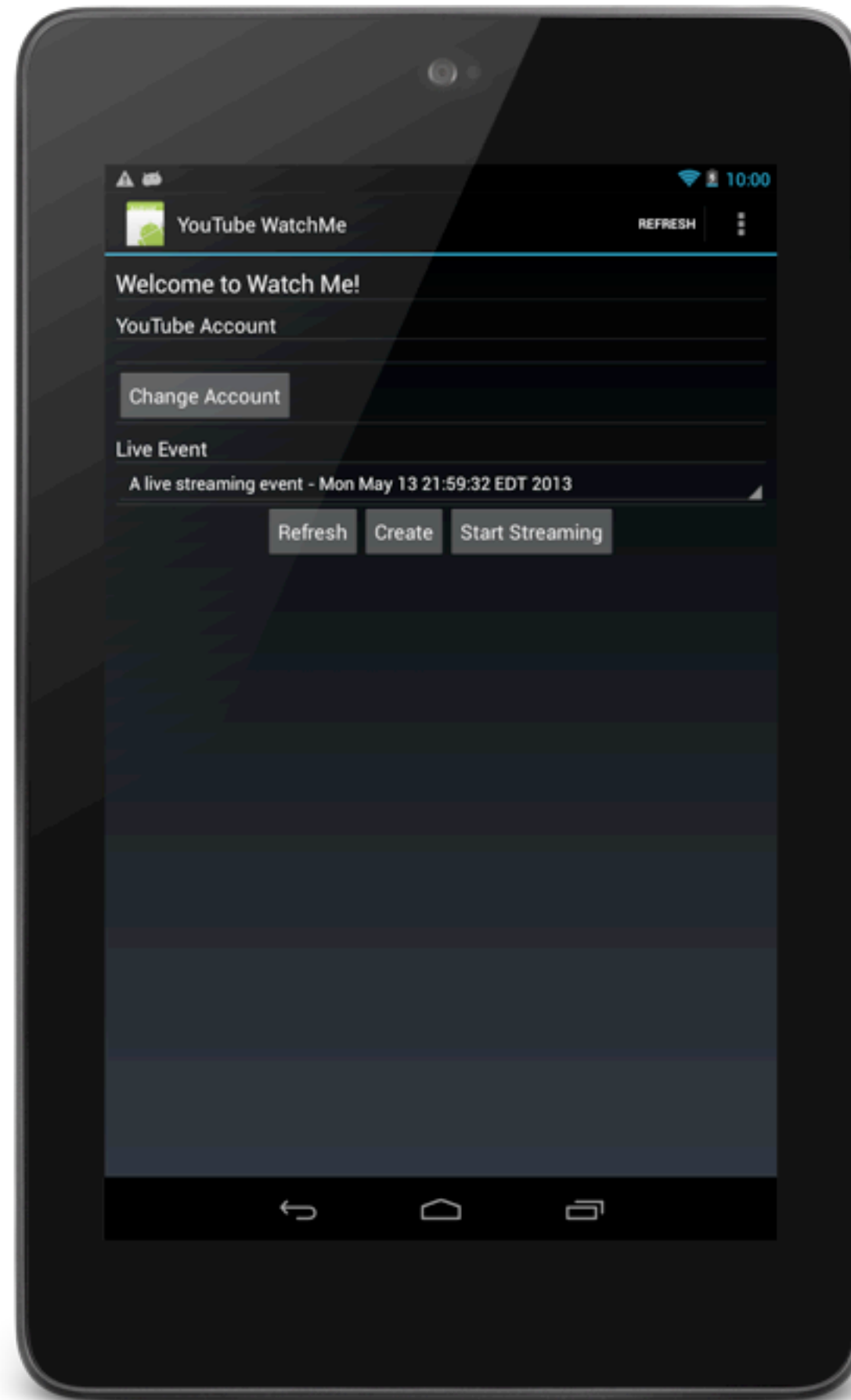


Android Demo

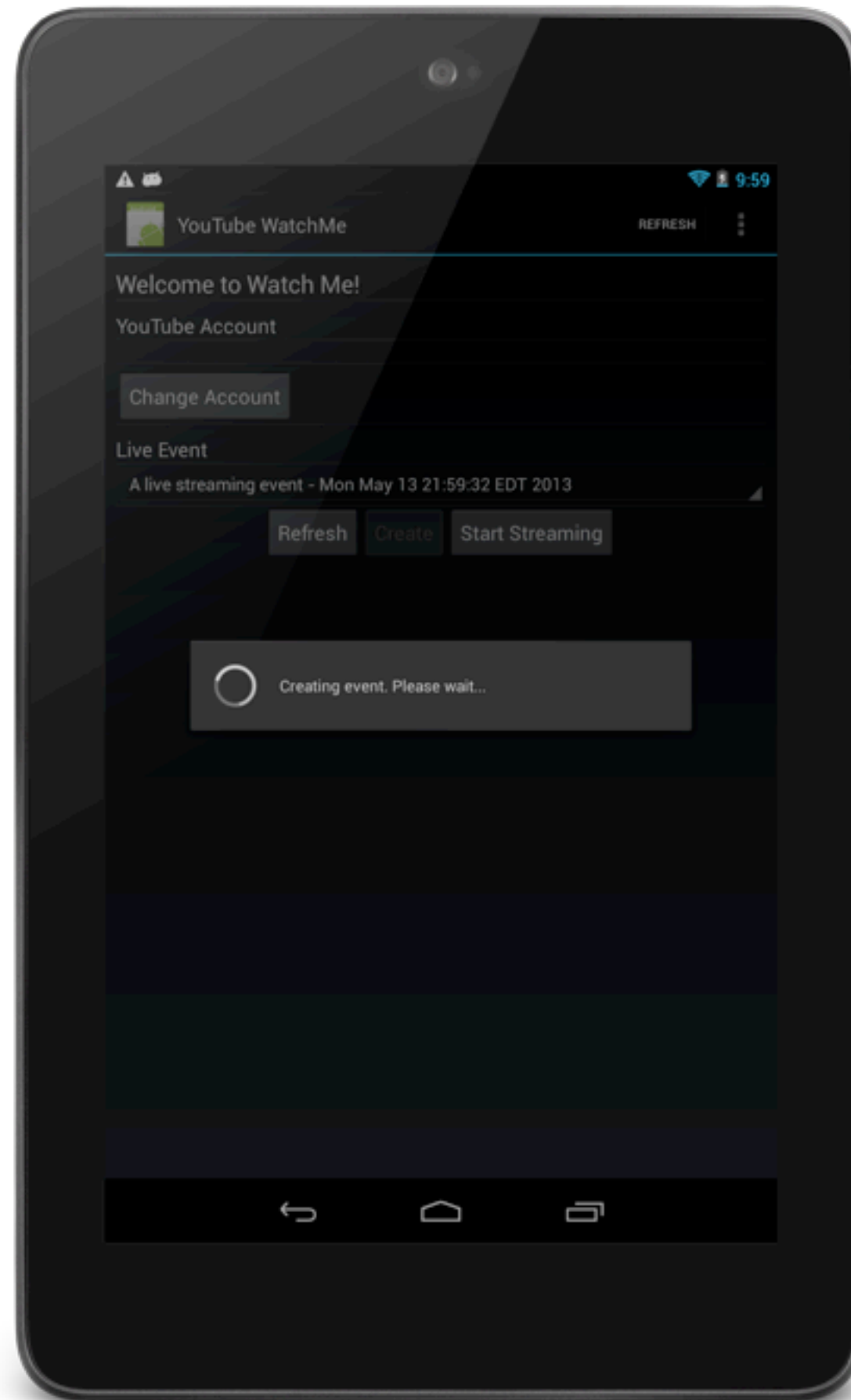


Android Demo

Watch Me

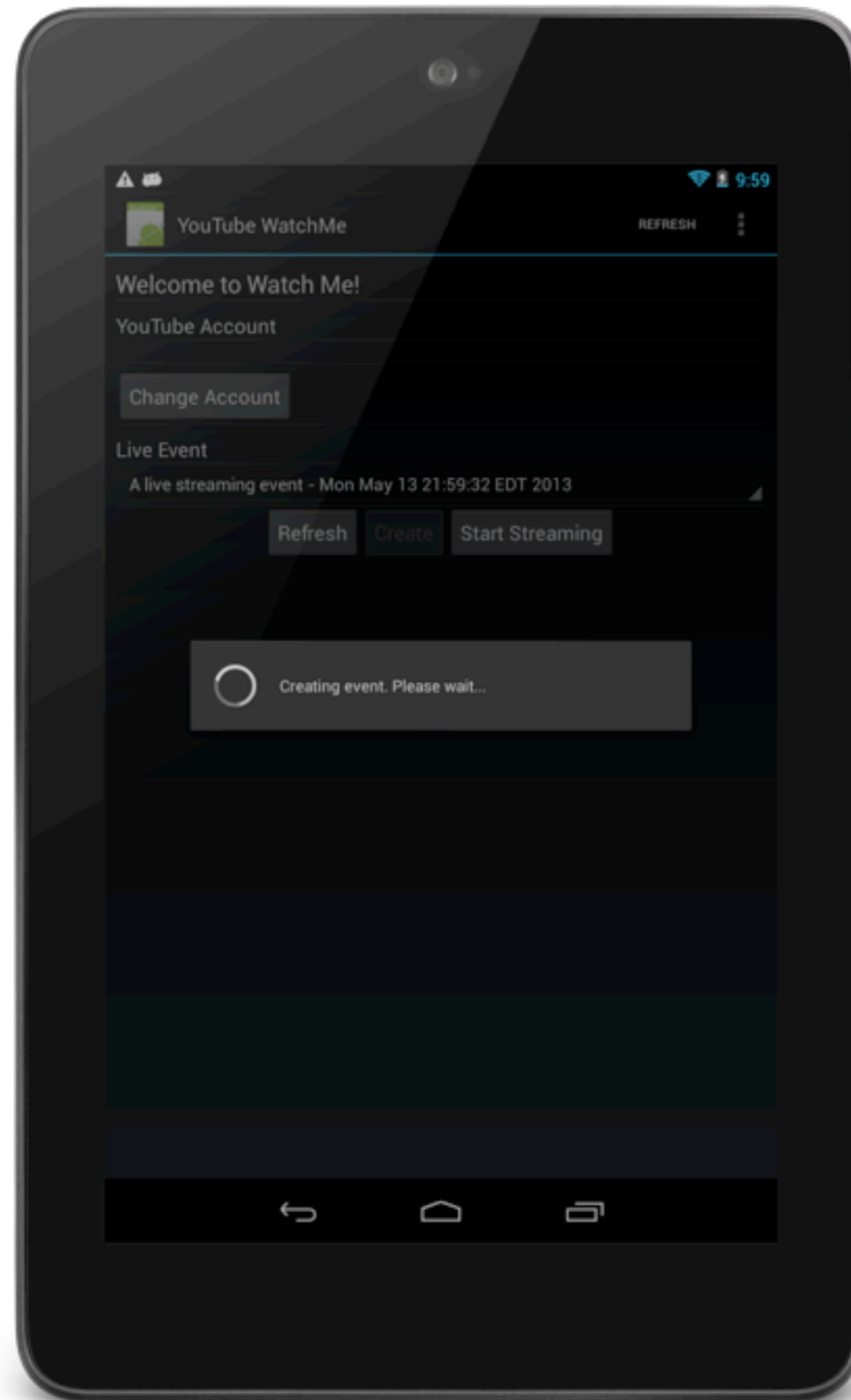


Android Demo



Android Demo

Create Event



Android Demo



Android Demo

List Events



Android Demo

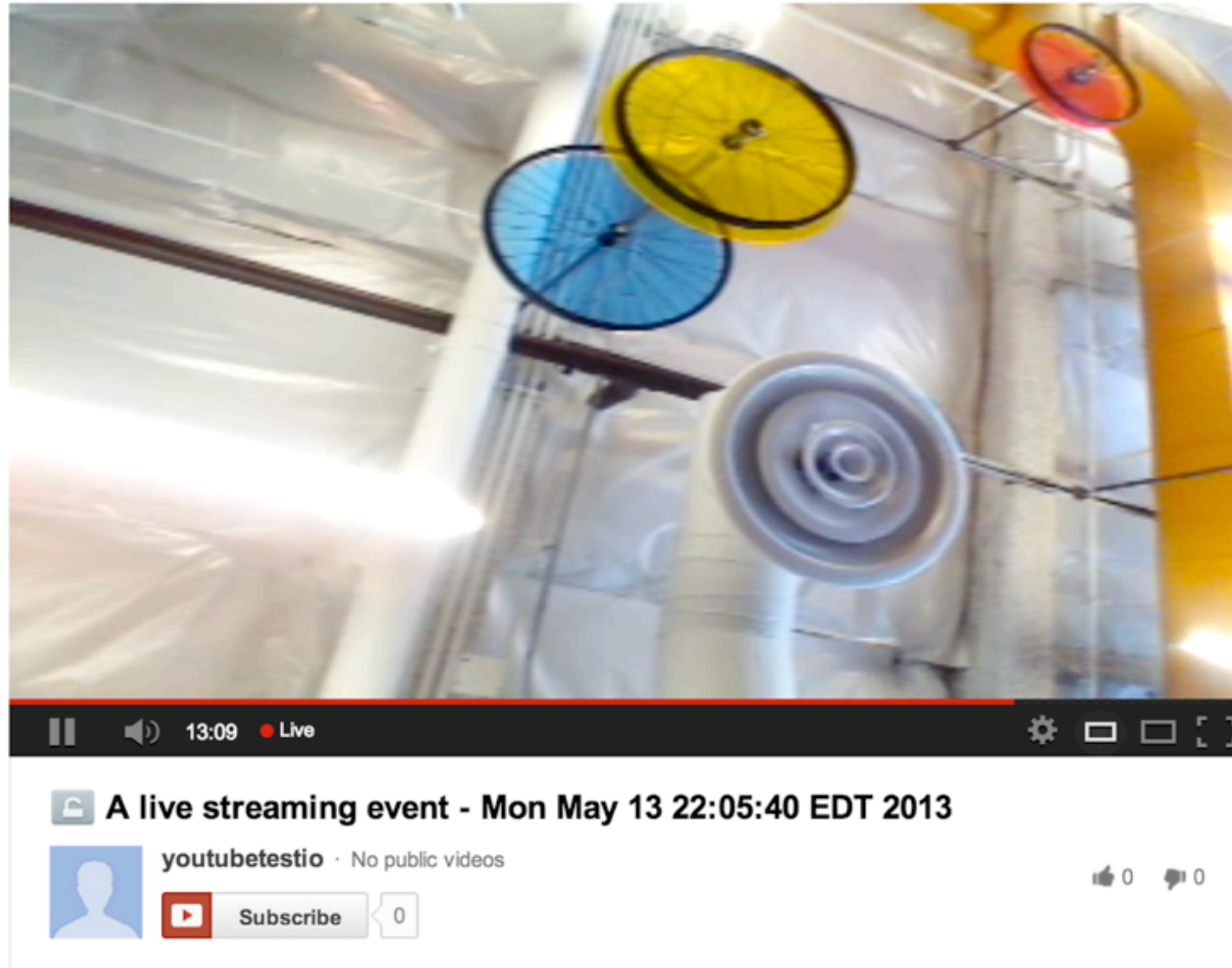


Android Demo

Streaming



Android Demo

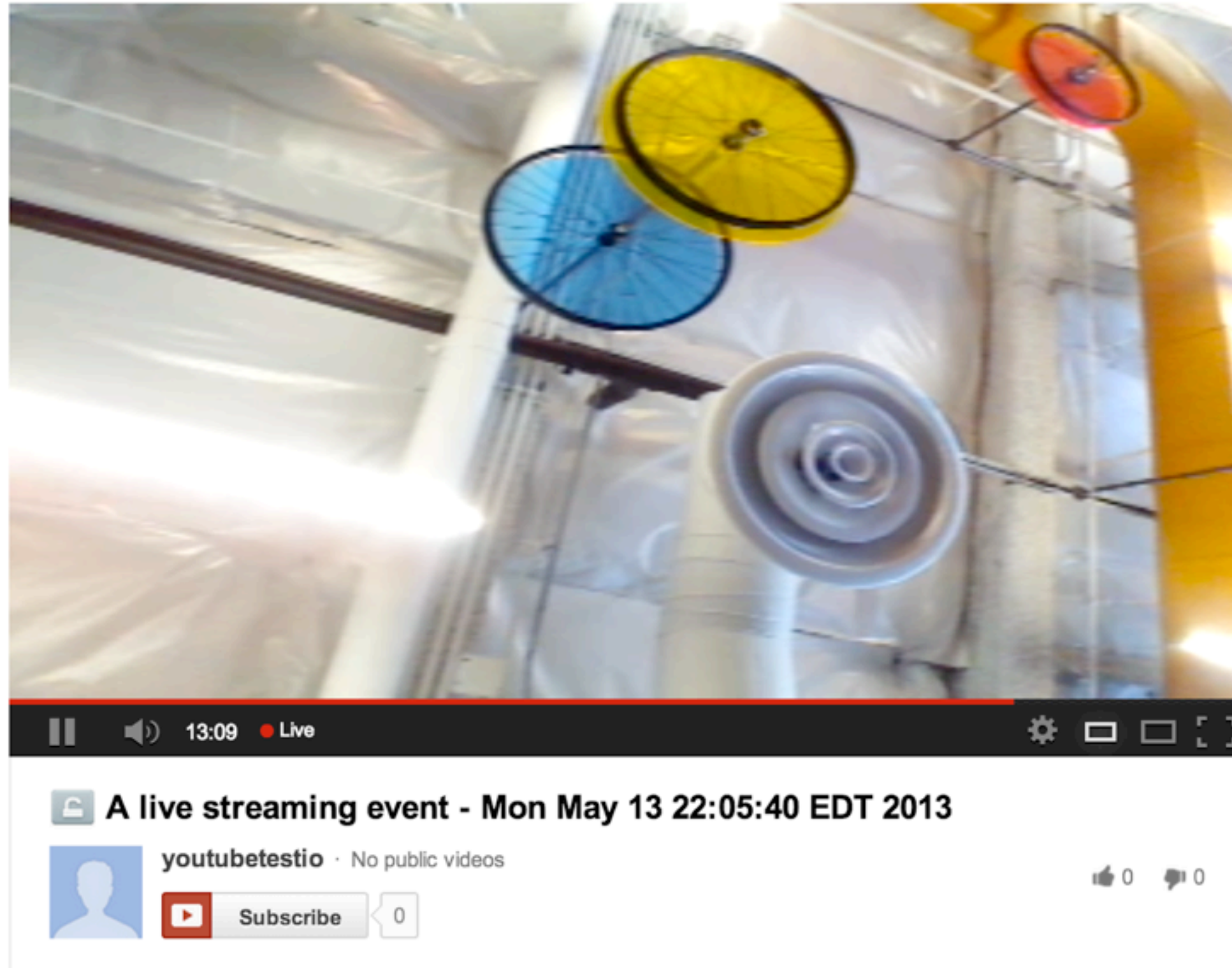


A YouTube video player interface. The video content shows a room with several bicycle wheels (yellow, blue, red, and white) hanging from the ceiling. The player controls at the bottom include a pause button, a volume icon, a timer showing 13:09, a 'Live' indicator, and icons for settings, full screen, and share. Below the video, the title is 'A live streaming event - Mon May 13 22:05:40 EDT 2013'. The channel name is 'youtubetestio' with the note 'No public videos'. There are 0 likes and 0 dislikes. A 'Subscribe' button is visible with a '0' next to it.



Android Demo

On YouTube



A screenshot of a YouTube video player. The video content shows a room with several colorful ceiling fans (yellow, blue, red, and white) hanging from the ceiling. The video player interface includes a progress bar at 13:09, a 'Live' indicator, and control icons for play/pause, volume, settings, and full screen. Below the video, the title is 'A live streaming event - Mon May 13 22:05:40 EDT 2013'. The channel name is 'youtubetestio' with the note 'No public videos'. There are 0 likes and 0 dislikes. A 'Subscribe' button is visible with a '0' next to it.







developers.google.com/youtube/





<Thank You!>



+SatyajeetSalgar
@salgar

+IbrahimUlukaya
@ulukaya



Google
Developers