Low-Latency HTTP Live Streaming

Roger Pantos, HLS Technical Lead
Why lower latency?
Crucial for Some Content
Crucial for Some Content

Sports
Crucial for Some Content

Sports

Late-breaking news
Crucial for Some Content

Sports

Late-breaking news

Real-time interactive gaming broadcasts
Crucial for Some Content

Sports
Late-breaking news
Real-time interactive gaming broadcasts
Award ceremonies and other social media events
How Low?
How Low?

2-8 seconds matches cable satellite TV
How Low?

2-8 seconds matches cable satellite TV

Designing for 1-2 delay (at scale!)
It’s Still HLS
It’s Still HLS

Rate adaptation
It’s Still HLS

Rate adaptation

Encryption, ads, metadata
It’s Still HLS

Rate adaptation

Encryption, ads, metadata

Scales with regular CDNs
It’s Still HLS

Rate adaptation

Encryption, ads, metadata

Scales with regular CDNs

Backward-compatible
How To Do It
Regular HLS Latency
Regular HLS Latency

HLS was designed to be simple and robust
Regular HLS Latency

HLS was designed to be simple and robust

But simplicity has a cost
Regular HLS Latency

HLS was designed to be simple and robust

But simplicity has a cost

HLS typically lags 20-30s behind live
Client

Server

Encode segment

frame #1

6s
Client

Playlist request
(best case)

Server

Encode segment

frame #1

6s
Playlist request (best case)

Playlist request (worst case)

Encode segment

frame #1

6s
Client

Playlist request (best case)

6s

Playlist request (worst case)

~5.9s

Server

Encode segment

frame #1

Playlist response
- Playlist request (best case): ~6s
- Playlist request (worst case): ~5.9s
- Segment request:

Server:
- Encode segment
- Playlist response

Client:
- Frame #1
Client

Playlist request (best case)

Playlist request (worst case)
Segment request

Server

Encode segment

6s

~5.9s

Playlist response
Segment response

frame #1
Client

Playlist request (best case)

6s

Playlist request (worst case)

1 RTT

~5.9s

Segment request

Segment response

Server

Encode segment

frame #1

Playlist response

Segment response
Client

Playlist request
(best case)

Server

6s

Encode segment

Playlist request
(worst case)

~5.9s

1 RTT

Playlist response

Segment request

Segment response
Client

Encode segment

Server

frame #1

Playlist request (best case)

~5.9s

Playlist request (worst case)

1 RTT

Segment request

Segment response

Playlist response

1 RTT
Too Much Caching Hurts Latency

Client 1

Client 2

Origin

Playlist (1-3)

Segment1

Segment2

Segment3

CDN edge

CDN edge

CDN edge
Too Much Caching Hurts Latency
Too Much Caching Hurts Latency
Too Much Caching Hurts Latency
Considerations
Considerations

HTTP is still the king
Considerations

HTTP is still the king

Keeping HTTP means keeping segment encode delay
Considerations

HTTP is still the king

Keeping HTTP means keeping segment encode delay

• Segments must become smaller: ~250 milliseconds
Considerations

HTTP is still the king

Keeping HTTP means keeping segment encode delay
• Segments must become smaller: ~250 milliseconds

CDNs are essential to scale
Considerations

HTTP is still the king

Keeping HTTP means keeping segment encode delay
• Segments must become smaller: ~250 milliseconds

CDNs are essential to scale
• Caches are gonna cache
Considerations

HTTP is still the king

Keeping HTTP means keeping segment encode delay
• Segments must become smaller: ~250 milliseconds

CDNs are essential to scale
• Caches are gonna cache

Runway to react is much shorter
Considerations

HTTP is still the king

Keeping HTTP means keeping segment encode delay
  • Segments must become smaller: ~250 milliseconds

CDNs are essential to scale
  • Caches are gonna cache

Runway to react is much shorter
  • Must switch in ~600 milliseconds instead of 10 seconds
5 Changes
Reduce Publishing Latency

Origin
Reduce Publishing Latency

Origin

1.1.ts
Reduce Publishing Latency
Reduce Publishing Latency

Origin

1.1.ts
1.2.ts
1.3.ts
1.ts
Optimize Segment Discovery

Client

Origin
Optimize Segment Discovery
Optimize Segment Discovery
Optimize Segment Discovery

Client

Origin

1.1.ts

...
Improve Caching Behavior

Client 1

CDN

Origin

Optimize Discovery
Improve Caching Behavior

Client 1

CDN

Origin
Improve Caching Behavior

Client 1 → Question Mark → CDN → Origin

Optimize Discovery
Improve Caching Behavior

Client 1

CDN

Origin

Optimize Discovery
Improve Caching Behavior

Client 1

CDN

Client 2

Optimize Discovery
Improve Caching Behavior

Origin

Client 1

CDN

1.1.ts

Client 2

Origin

1.1.ts
Improve Caching Behavior

Client 1

Client 2

CDN

Origin

Optimize Discovery
Improve Caching Behavior

Client 1

CDN

1.1.ts

Origin

1.1.ts

Client 2

Optimize Discovery
Improve Caching Behavior

Client 1

? 1.1.ts

CDN

Origin

Client 2
Improve Caching Behavior

Client 1

CDN

1.1.ts

Origin

1.1.ts

Client 2
Improve Caching Behavior

Client 1

CDN

Client 2

Origin

1.1.ts

1.2.ts

1.1.ts

1.2.ts
Eliminate Segment Round Trip

Client 1

CDN

Origin

1.1.ts
Eliminate Segment Round Trip

Client 1 → ? → CDN (1.1.ts) → Origin (1.1.ts)
Eliminate Segment Round Trip

Client 1

+ 1.1.ts

? 

CDN

1.1.ts

Origin

1.1.ts
Reduce Playlist Transfer Overhead

Client 1

CDN
Reduce Playlist Transfer Overhead
Reduce Playlist Transfer Overhead

Client 1 —> CDN —> ? —> Client 1
Reduce Playlist Transfer Overhead

Client 1

CDN
Reduce Playlist Transfer Overhead
Reduce Playlist Transfer Overhead

Client 1

CDN
Reduce Playlist Transfer Overhead
Switch Tiers Quickly

CDN

1 Mbps
- 1.1.ts
- 1.2.ts
- 1.3.ts

2 Mbps
- 2Mb 1.1.ts
- 2Mb 1.2.ts
- 2Mb 1.3.ts

Client 1
Switch Tiers Quickly

CDN

1 Mbps
- 1.1.ts
- 1.2.ts
- 1.3.ts

2 Mbps
- 2Mb 1.1.ts
- 2Mb 1.2.ts
- 2Mb 1.3.ts

Client 1
Switch Tiers Quickly

Client 1

CDN

1 Mbps
- 1.1.ts
- 1.2.ts
- 1.3.ts

2 Mbps
- 2Mb 1.1.ts
- 2Mb 1.2.ts
- 2Mb 1.3.ts
Reduce Publishing Latency
Optimize Discovery
Eliminate Round Trips
Reduce Transfer Overhead
Switch Tiers Quickly
Origin API

Enabled by EXT-X-SERVER-CONTROL
Origin API

Enabled by EXT-X-SERVER-CONTROL

Small number of server directives
Origin API

Enabled by EXT-X-SERVER-CONTROL

Small number of server directives

Carried as query parameters on Playlist URL
Origin API

Enabled by EXT-X-SERVER-CONTROL

Small number of server directives

Carried as query parameters on Playlist URL

GET https://example.com/live.m3u8?_HLS_skip=YES
Origin API

Enabled by EXT-X-SERVER-CONTROL

Small number of server directives

Carried as query parameters on Playlist URL

GET https://example.com/live.m3u8?_HLS_skip=YES

_HLS* parameters are reserved on m3u8 URLs
Origin API

Enabled by EXT-X-SERVER-CONTROL

Small number of server directives

Carried as query parameters on Playlist URL

GET https://example.com/live.m3u8?_HLS_skip=YES

_HLS* parameters are reserved on m3u8 URLs

Sorted within URL to improve CDN hit ratio
Partial Segments
Partial Segments

Partial Segment is a subset of a regular Segment
Partial Segments

Partial Segment is a subset of a regular Segment

- CMAF Chunks, or short TS, audio or VTT
Partial Segments

Partial Segment is a subset of a regular Segment

- CMAF Chunks, or short TS, audio or VTT
- Less than a full GOP
Partial Segments

Partial Segment is a subset of a regular Segment
- CMAF Chunks, or short TS, audio or VTT
- Less than a full GOP

Playlists update every Partial Segment
Partial Segments

Partial Segment is a subset of a regular Segment
• CMAF Chunks, or short TS, audio or VTT
• Less than a full GOP

Playlists update every Partial Segment

Publishing latency becomes the Partial Segment duration
Partial Segments

Partial Segment is a subset of a regular Segment
- CMAF Chunks, or short TS, audio or VTT
- Less than a full GOP

Playlists update every Partial Segment

Publishing latency becomes the Partial Segment duration

Partial Segments appear in parallel to regular Segments in Playlist
Partial Segments Disappear Quickly

Reduce Publishing Latency
Partial Segments Disappear Quickly

Partial Segments only at live edge of Playlist
Partial Segments Disappear Quickly

Partial Segments only at live edge of Playlist

• Removed once Parent Segments are established
Partial Segments Disappear Quickly

Partial Segments only at live edge of Playlist
• Removed once Parent Segments are established
• Keeps Playlists small
Partial Segments Disappear Quickly

Partial Segments only at live edge of Playlist
• Removed once Parent Segments are established
• Keeps Playlists small
Partial Segments Disappear Quickly

Partial Segments only at live edge of Playlist
• Removed once Parent Segments are established
• Keeps Playlists small
# The new EXT-X-PART Tag

EXTM3U

#EXT-X-TARGETDURATION:6.0

#EXT-X-PART-INF:PART-TARGET=0.5

#EXTINF 6.0,

segment43.ts
# The new EXT-X-PART Tag

#EXTM3U
#EXT-X-TARGETDURATION:6.0
#EXT-X-PART-INF:PART-TARGET=0.5
#EXTINF 6.0,
segment43.ts
#EXT-X-PART:URI="segment44.1.ts",DURATION=0.5,INDEPENDENT=YES
# The new EXT-X-PART Tag

#EXTM3U

#EXT-X-TARGETDURATION:6.0

#EXT-X-PART-INF:PART-TARGET=0.5

#EXTINF 6.0,

segment43.ts

#EXT-X-PART:URI="segment44.1.ts",DURATION=0.5,INDEPENDENT=YES

#EXT-X-PART:URI="segment44.2.ts",DURATION=0.5
# The new EXT-X-PART Tag

```m3u
#EXTM3U

## The new EXT-X-PART Tag

```
# The new EXT-X-PART Tag

#EXTM3U

#EXT-X-TARGETDURATION:6.0

#EXT-X-PART-INF:PART-TARGET=0.5

#EXTINF 6.0,

segment43.ts

#EXT-X-PART:URI="segment44.1.ts",DURATION=0.5,INDEPENDENT= YES

#EXT-X-PART:URI="segment44.2.ts",DURATION=0.5

# The new EXT-X-PART Tag

#EXTM3U

#EXT-X-TARGETDURATION:6.0

#EXT-X-PART-INF:PART-TARGET=0.5

#EXTINF 6.0,

segment44.ts

#EXT-X-PART:URI="segment44.12.ts",DURATION=0.5
# The new EXT-X-PART Tag

#EXTM3U

#EXT-X-TARGETDURATION:6.0

#EXT-X-PART-INF:PART-TARGET=0.5

#EXTINF 6.0,

segment43.ts

#EXT-X-PART:URI="segment44.1.ts",DURATION=0.5,INDEPENDENT=YES

#EXT-X-PART:URI="segment44.2.ts",DURATION=0.5

:

#EXT-X-PART:URI="segment44.12.ts",DURATION=0.5

#EXTINF 6.0,

segment44.ts

#EXT-X-PART:URI="segment45.1.ts",DURATION=0.5,INDEPENDENT=YES
# The new EXT-X-PART Tag
#EXTM3U
#EXT-X-TARGETDURATION:6.0
#EXT-X-PART-INF:PART-TARGET=0.5
#EXTINF 6.0,
    segment43.ts
#EXT-X-PART:URI="segment44.1.ts",DURATION=0.5,INDEPENDENT=YES
#EXT-X-PART:URI="segment44.2.ts",DURATION=0.5
    :
#EXT-X-PART:URI="segment44.12.ts",DURATION=0.5
#EXTINF 6.0,
    segment44.ts
#EXT-X-PART:URI="segment45.1.ts",DURATION=0.5,INDEPENDENT=YES
#EXT-X-PART:URI="segment45.2.ts",DURATION=0.5
    :
#EXT-X-PART:URI="segment45.12.ts",DURATION=0.5
#EXTINF 6.0,
    segment45.ts
# The new EXT-X-PART Tag

EXTM3U
EXT-X-TARGETDURATION:6.0
EXT-X-PART-INF:PART-TARGET=0.5
EXTINF 6.0,
    segment43.ts
EXTINF 6.0,
    segment44.ts
EXT-X-PART:URI="segment45.1.ts",DURATION=0.5,INDEPENDENT=YES
EXT-X-PART:URI="segment45.2.ts",DURATION=0.5
...:
EXT-X-PART:URI="segment45.12.ts",DURATION=0.5
EXTINF 6.0,
    segment45.ts
Blocking Playlist Reload

EXT-X-SERVER-CONTROL with CAN-BLOCK-RELOAD attribute
Blocking Playlist Reload

EXT-X-SERVER-CONTROL with CAN-BLOCK-RELOAD attribute

Clients ask for its next Playlist update in advance
Blocking Playlist Reload

EXT-X-SERVER-CONTROL with CAN-BLOCK-RELOAD attribute

Clients ask for its next Playlist update in advance

Server holds request until next Segment or Partial Segment appears
#EXTM3U
#EXT-X-TARGETDURATION:6
#EXT-X-MEDIA-SEQUENCE:1800
#EXTINF 6.0,
segment-abc.ts
#EXTINF 6.0,
segment-def.ts
#EXT-X-DISCONTINUITY
#EXTINF 6.0,
ad-xyz.ts

Media Sequence Number 1800
Media Sequence Number 1801
#EXTM3U
#EXT-X-TARGETDURATION:6
#EXT-X-MEDIA-SEQUENCE:1800
#EXTINF 6.0,
segment-abc.ts
#EXTINF 6.0,
segment-def.ts
#EXT-X-DISCONTINUITY
#EXTINF 6.0,
ad-xyz.ts

Media Sequence Number 1800
Media Sequence Number 1801
Media Sequence Number 1802
#EXTM3U
#EXT-X-TARGETDURATION:6
#EXT-X-MEDIA-SEQUENCE:1800
#EXTINF 6.0,
segment-abc.ts
#EXTINF 6.0,
segment-def.ts
#EXT-X-DISCONTINUITY
#EXTINF 6.0,
ad-xyz.ts
???

Media Sequence Number 1800
Media Sequence Number 1801
Media Sequence Number 1802
Media Sequence Number 1803
# Blocking Playlist Reload

# Block until Media Sequence Number 1803 is in Playlist

GET https://example.com/live.m3u8?_HLS_msn=1803
# Blocking Playlist Reload

# Block until Media Sequence Number 1803 is in Playlist

GET https://example.com/live.m3u8?_HLS_msn=1803

# Block until first part of Media Sequence Number 1803 is in Playlist

GET https://example.com/live.m3u8?_HLS_msn=1803&_HLS_part=0&_HLS_push=1
# Blocking Playlist Reload
# Block until Media Sequence Number 1803 is in Playlist

GET https://example.com/live.m3u8?_HLS_msn=1803

# Block until first part of Media Sequence Number 1803 is in Playlist

GET https://example.com/live.m3u8?_HLS_msn=1803&_HLS_part=0&_HLS_push=1
# Blocking Playlist Reload
# Block until Media Sequence Number 1803 is in Playlist

GET https://example.com/live.m3u8?_HLS_msn=1803

# Block until first part of Media Sequence Number 1803 is in Playlist

GET https://example.com/live.m3u8?_HLS_msn=1803&_HLS_part=0&_HLS_push=1
HTTP/2

Eliminate Round Trips
HTTP/2

Successor to HTTP 1.1
HTTP/2

Successor to HTTP 1.1

RFC 7540 published in 2015
HTTP/2

Successor to HTTP 1.1

RFC 7540 published in 2015

Widely supported
HTTP/2

Successor to HTTP 1.1

RFC 7540 published in 2015

Widely supported

Required for Low Latency HLS
HTTP/2

Successor to HTTP 1.1

RFC 7540 published in 2015

Widely supported

Required for Low Latency HLS

HTTP/2 Push
HTTP/2

Successor to HTTP 1.1

RFC 7540 published in 2015

Widely supported

Required for Low Latency HLS

HTTP/2 Push

• GET response can include secondary resources
Segment Push

Eliminate Round Trips
Segment Push

GET of Playlist also pushes new Segment
Segment Push

GET of Playlist also pushes new Segment

• Eliminates round trip request
Reduce Publishing Latency
Optimize Discovery
Eliminate Round Trips

Client

Server

~5.9s

6s

Playlist request

Segment request
Client → Server

- Playlist request
- Segment request

Server to LL Client

- Playlist request

Optimize Discovery
Reduce Publishing Latency
Eliminate Round Trips
Client

Server

LL Client

LL Server

Playlist request

Segment request

6s

~5.9s

1s

Reduce Publishing Latency
Optimize Discovery
Eliminate Round Trips
Playlist request
Segment request

Client

Server

Playlist request

~5.9s

LL Client

LL Server

Playlist response
+ Segment push

Reduce Publishing Latency
Optimize Discovery
Eliminate Round Trips
Client
Playlist request
Segment request
Server
6s
~5.9s
LL Client
LL Server
Playlist request
1s
Playlist response
+ Segment push

Reduce Publishing Latency
Optimize Discovery
Eliminate Round Trips
Client

Server

Playlist request

Segment request

LL Client

LL Server

Playlist request

Next Playlist request

6s

~5.9s

1s

Reduce Publishing Latency

Optimize Discovery

Eliminate Round Trips
Playlist Delta Updates
Playlist Delta Updates

EXT-X-SERVER-CONTROL with CAN-SKIP-UNTIL attribute
Playlist Delta Updates

EXT-X-SERVER-CONTROL with CAN-SKIP-UNTIL attribute

Clients asks for Delta Update explicitly
Playlist Delta Updates

EXT-X-SERVER-CONTROL with CAN-SKIP-UNTIL attribute

Clients asks for Delta Update explicitly

Update skips the earlier part of Playlist client already has
# The new EXT-X-SKIP Tag

GET https://example.com/1M/live.m3u8?_HLS_skip=YES

#EXTM3U
#EXT-X-VERSION:9
#EXT-X-SERVER-CONTROL:CAN-SKIP-UNTIL=36.0
#EXT-X-TARGETDURATION:6
#EXT-X-MEDIA-SEQUENCE:100
#EXT-X-SKIP:SKIPPED-SEGMENTS=1700
#EXTINF 6.0,
segment1800.ts
#EXTINF 6.0,
segment1801.ts
#EXTINF 6.0,
segment1802.ts
...

Reduce Transfer Overhead
# The new EXT-X-SKIP Tag

GET https://example.com/1M/live.m3u8?_HLS_skip=YES

#EXTM3U
#EXT-X-VERSION:9
#EXT-X-SERVER-CONTROL:CAN-SKIP-UNTIL=36.0
#EXT-X-TARGETDURATION:6
#EXT-X-MEDIA-SEQUENCE:100
#EXT-X-SKIP:SKIPPED-SEGMENTS=1700
#EXTINF 6.0,
segment1800.ts
#EXTINF 6.0,
segment1801.ts
#EXTINF 6.0,
segment1802.ts
...

Reduce Transfer Overhead
# The new EXT-X-SKIP Tag

GET https://example.com/1M/live.m3u8?_HLS_skip=YES

#EXTM3U
#EXT-X-VERSION:9
#EXT-X-SERVER-CONTROL:CAN-SKIP-UNTIL=36.0
#EXT-X-TARGETDURATION:6
#EXT-X-MEDIA-SEQUENCE:100
#EXT-X-SKIP:SKIPPED-SEGMENTS=1700
#EXTINF 6.0,
segment1800.ts
#EXTINF 6.0,
segment1801.ts
#EXTINF 6.0,
segment1802.ts
...

Reduce Transfer Overhead
Rendition Reports

Playlist updates can carry up-to-date reports on peer Playlists
Rendition Reports

Playlist updates can carry up-to-date reports on peer Playlists

• Last Media Sequence Number and last Partial Segment number
Rendition Reports

Playlist updates can carry up-to-date reports on peer Playlists
• Last Media Sequence Number and last Partial Segment number
• Allows client to load latest Playlist when switching bit rates
# Requesting and receiving Rendition Reports

GET https://example.com/1M/live.m3u8?_HLS_report=2M/live.m3u8
# Requesting and receiving Rendition Reports

GET https://example.com/1M/live.m3u8?_HLS_report=/2M/live.m3u8

#EXTM3U
#EXT-X-RENDITION-REPORT:URI="/2M/live.m3u8",LAST-MSN=1801,LAST-PART=0
#EXT-X-TARGETDURATION:6
#EXT-X-MEDIA-SEQUENCE:1800
#EXTINF 6.0,
segment73.ts
...

Switch Tiers Quickly
Putting It All Together
Demo
How To Get Started
App Adoption
// Discover and adjust distance from live

var item = myAVPlayerItem()
var howFarNow = item.configuredTimeOffsetFromLive
// Discover and adjust distance from live

var item = myAVPlayerItem()
var howFarNow = item.configuredTimeOffsetFromLive
// Discover and adjust distance from live

var item = myAVPlayerItem()
var howFarNow = item.configuredTimeOffsetFromLive
var recommended = item.recommendedTimeOffsetFromLive
// Discover and adjust distance from live

var item = myAVPlayerItem()
var howFarNow = item.configuredTimeOffsetFromLive
var recommended = item.recommendedTimeOffsetFromLive

if (howFarNow < recommended) {
    item.configuredTimeOffsetFromLive = recommended
}
// Discover and adjust distance from live

var item = myAVPlayerItem()
var howFarNow = item.configuredTimeOffsetFromLive
var recommended = item.recommendedTimeOffsetFromLive

if ( howFarNow < recommended ) {
    item.configuredTimeOffsetFromLive = recommended
}

// Maintain position of playhead relative to live edge after rebuffer
item.automaticallyPreservesTimeOffsetFromLive = YES
// Discover and adjust distance from live

var item = myAVPlayerItem()
var howFarNow = item.configuredTimeOffsetFromLive
var recommended = item.recommendedTimeOffsetFromLive

if ( howFarNow < recommended ) {
    item.configuredTimeOffsetFromLive = recommended
}

// Maintain position of playhead relative to live edge after rebuffer
item.automaticallyPreservesTimeOffsetFromLive = YES
Configure Your CDN
Configure Your CDN

Deliver HLS via HTTP/2
Configure Your CDN

Deliver HLS via HTTP/2
• Support HTTP/2 Push
Configure Your CDN

Deliver HLS via HTTP/2
• Support HTTP/2 Push
• Support HTTP/2 dependency and weighting
Configure Your CDN

Deliver HLS via HTTP/2
• Support HTTP/2 Push
• Support HTTP/2 dependency and weighting

Each server must vend all bit rate tiers
Configure Your CDN

Deliver HLS via HTTP/2
• Support HTTP/2 Push
• Support HTTP/2 dependency and weighting

Each server must vend all bit rate tiers

CDN must aggregate duplicate pending requests to origin
Configure Your CDN

Deliver HLS via HTTP/2
• Support HTTP/2 Push
• Support HTTP/2 dependency and weighting

Each server must vend all bit rate tiers

CDN must aggregate duplicate pending requests to origin
• "Reader while writer” in Apache Traffic Server
Implement Your Origin
Implement Your Origin

Spec for Low-Latency HLS is available today
Implement Your Origin

Spec for Low-Latency HLS is available today
• Currently a separate draft
Implement Your Origin

Spec for Low-Latency HLS is available today
• Currently a separate draft
• Will be merged into core HLS spec later this year
Implement Your Origin

Spec for Low-Latency HLS is available today
• Currently a separate draft
• Will be merged into core HLS spec later this year

Server Configuration Profile
Implement Your Origin

Spec for Low-Latency HLS is available today
• Currently a separate draft
• Will be merged into core HLS spec later this year

Server Configuration Profile
• Required features of Low-Latency servers
Reference Implementation
Reference Implementation

“Low-Latency HLS Beta Tools”
Reference Implementation

“Low-Latency HLS Beta Tools”

Create Playlists with Segments and Partial Segments
Reference Implementation

“Low-Latency HLS Beta Tools”

Create Playlists with Segments and Partial Segments

PHP front-end implements Origin API
Reference Implementation

“Low-Latency HLS Beta Tools”

Create Playlists with Segments and Partial Segments

PHP front-end implements Origin API
  • Blocking Playlist Reload
Reference Implementation

“Low-Latency HLS Beta Tools”

Create Playlists with Segments and Partial Segments

PHP front-end implements Origin API
- Blocking Playlist Reload
- Playlist Delta Updates
Reference Implementation

“Low-Latency HLS Beta Tools”

Create Playlists with Segments and Partial Segments

PHP front-end implements Origin API
  • Blocking Playlist Reload
  • Playlist Delta Updates
  • Rendition Reports
Beta in 2019
Beta in 2019

App entitlement `com.apple.developer.coremedia.hls.low-latency`
Beta in 2019

App entitlement `com.apple.developer.coremedia.hls.low-latency`

• Development
Beta in 2019

App entitlement `com.apple.developer.coremedia.hls.low-latency`

• Development
• TestFlight distribution
Summary
Summary

Read the spec

  protocol_extension_for_low_latency_hls
Summary

Read the spec


Try out the beta
Summary

Read the spec

Try out the beta

Build your packager and origin
HTTP Live Streaming Lab
Thursday, 4:00

HTTP Live Streaming Lab 2
Friday, 11:00