

EZDRM Unified Streaming Packager / Static Packaging

Table of Contents

Prerequisites	3
Generating DRM Keys	4
<i>Widevine and PlayReady</i>	4
<i>Apple FairPlay Streaming</i>	7
Unified Packaging	10
Universal (Widevine and PlayReady)	10
<i>Step 1 – Encrypt fragmented Audio & Video files</i>	10
<i>Step2 – Package MPD</i>	12
<i>Step 3 – Test Playback</i>	12
Multi-Key Encryption	16
Apple FairPlay Streaming	18
<i>Sample Players for Apple FairPlay Streaming</i>	18
Appendix 1 – Call for Existing EZDRM Keys	21
<i>Widevine and PlayReady</i>	21
<i>Apple FairPlay Streaming</i>	24
Appendix 2 – Installing ARC Plug-in	27

Prerequisites

- A DRM supported license key from Unified Streaming is required, saved as a USP License Key file. You will save in a file in your home directory (where you enter when logging in).
- Mp4Split – used to create the manifest for on the fly encryption.
<http://docs.unified-streaming.com/installation/mp4split.html>

Generating DRM Keys

Widevine and PlayReady

Below are the steps to create the DRM Keys for CENC-Widevine or CENC-PlayReady encryption for Unified Streaming.

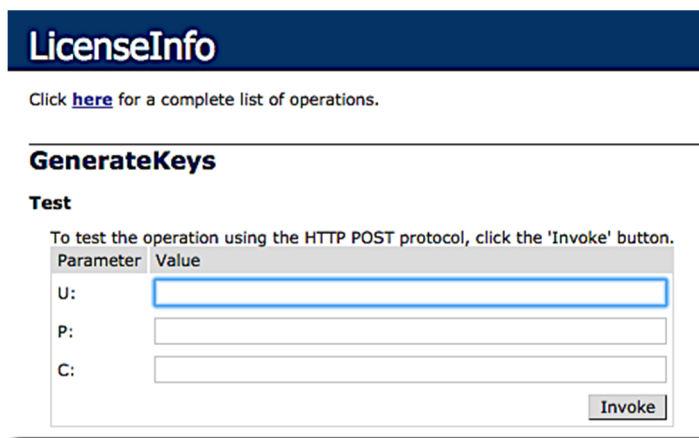
To request the DRM keys from EZDRM to package the media, there are two options, you can call the EZDRM web service in a browser, or you can script this process with curl or other web service calls.

Option 1: Request DRM keys using EZDRM Web Service

1. Call the EZDRM web service in a browser:

<https://wvm.ezdrm.com/ws/LicenseInfo.aspx?op=GenerateKeys>

2. Generate Key values by entering the parameters values and clicking "Invoke".



The parameters are as follows:

Parameter	Description
u	EZDRM username
p	EZDRM password
c	<u>Content ID</u> **optional

Note: The Content_ID is optional. The first time you use this web service it will be blank. For additional calls it can be blank for new keys or use an existing Content_ID. Sending a Content_ID will allow you to encrypt content with the same DRM values as other content and have that content share one license. If you don't send this value, the web service will automatically generate a unique Content_ID. If you call a Content_ID, you will get the back all the DRM key information for that Content_ID. See Appendix 1 for more information on calling existing keys with Content_ID.

3. The response from EZDRM will look like this:

```
<EZDRM xmlns="">
  <WideVine diffgr:id="WideVine1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <ContentID>61xxx028xxxxxxxxLbg==</ContentID>
    <Key>W5XXXXXXXX2HxTjhXXXXVw==</Key>
    <KeyHEX>5bXXXXXXXX9191fXXe38XXXX56bf</KeyHEX>
    <KeyID>WVXXXXXXXX1iBEXXw+XXXX==</KeyID>
    <KeyIDGUID>5XXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
    <KeyIDHEX>5XXXXX36d85XXXXXXXXXXXXb24XXb</KeyIDHEX>
    <PSSH>
      EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6IebXXX28kSYMXXXXXXXXXXXXXj3JXXXX==
    </PSSH>
    <ServerURL>https://widevine-dash.ezdrm.com/proxy?px=XXXXXX</ServerURL>
    <ServerGet>
      request={ "policy": "", "tracks": [ { "type": "SD" } ], "content_id": "61xxx028xxxxxxxxLbg==" }
    </ServerGet>
    <ResponseRaw>
      { "status": "OK", "drm": [ { "type": "WIDEVINE", "system_id": "edef8ba979d64acea3c827dcd51d21ed" } ], "tracks":
        [ { "type": "SD", "key_id": "WVXXXXXXXX1iBEXXw+XXXX==", "key": "WVXXXXXXXX1iBEXXw+XXXX==", "pssh":
          [ { "drm_type": "WIDEVINE", "data": "EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6IebXXX28kSYMXXXXXXXXXXXXXj3JXXXX==" } ] } ] }
    </ResponseRaw>
  </WideVine>
  <PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <Key>W5XXXXXXXX2HxTjhXXXXVw==</Key>
    <KeyHEX>5bXXXXXXXX9191fXXe38XXXX56bf</KeyHEX>
    <KeyIDGUID>5XXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
    <LAURL>
      https://playready.ezdrm.com/cency/preauth.aspx?px=XXXXXX
    </LAURL>
    <Checksum>1Xq+XXXXXX0=</Checksum>
  </PlayReady>
</EZDRM>
```

Option 2: Request DRM keys with curl

The second option to request DRM keys from EZDRM is to script the process with curl or another web service call.

Using EZDRM's web service, the curl script below retrieves the DRM values from the web service.

```
curl -v 'http://wvm.ezdrm.com/ws/LicenseInfo.aspx/GenerateKeys?U=EZDRM_USERNAME&P=EZDRM_PASSWORD&C= ""'
```

The parameters are as follows:

Parameter	Description
U	EZDRM username
P	EZDRM password
C	Content_ID **optional, for blank pass ""

Note: Although Content_ID is optional you must pass a "" for blank if you do not specify a Content_ID.

The following is returned from the web service:

```
<EZDRM xmlns="">
  <WideVine diffgr:id="WideVine1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <ContentID>6IxXXx0Z8xXXXXXXXXXXLbg==</ContentID>
    <Key>W5XXXXXXXXZHxTjhXXXXXvw==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXXX56bf</KeyHEX>
    <KeyID>WVXXXXXXXX1iBEXXw+XXXXX==</KeyID>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXXb</KeyIDGUID>
    <KeyIDHEX>5XXXXXXXX36d85XXXXXXXXXXXXb24XXb</KeyIDHEX>
    <PSSH>EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXX06IebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXXX==</PSSH>
    <ServerURL>https://widevine-dash.ezdrm.com/proxy?pX=XXXXXX</ServerURL>
    <ServerGet>request={"policy": "", "tracks": [ {"type": "SD"}], "content_id": "6IxXXx0Z8xXXXXXXXX
XXLbg=="}</ServerGet>
    <ResponseRaw>
{"status":"OK","drm":[{"type":"WIDEVINE","system_id":"edef8ba979d64acea3c827dcd51d21ed"}],"tracks":[{"type
":"SD","key_id":" WVXXXXXXXX1iBEXXw+XXXXX==","key":" W5XXXXXXXXZHxTjhXXXXXvw==","pssh":[{"drm_type":"WIDEVIN
E","data":"EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXX06IebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXXX="}]}]}</ResponseRaw>
    </WideVine>
    <PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <Key>W5XXXXXXXXZHxTjhXXXXXvw==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXXX56bf</KeyHEX>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXXb</KeyIDGUID>
    <LAURL>https://playready.ezdrm.com/cencv/preauth.aspx?pX=XXXXXX</LAURL>
    <Checksum>1Xq+XXXXXX0</Checksum>
    </PlayReady>
  </EZDRM>
```

Here is a description of the items returned for use with encryption:

- **KeyIDHex** – the common content encryption key value, also known as the “kid” value
- **KeyHex** – the common encryption key value, also known as the “cek” value
- **LAURL** – the server URL to the PlayReady license server
- **PSSH** - this is the value for the Widevine PSSH box

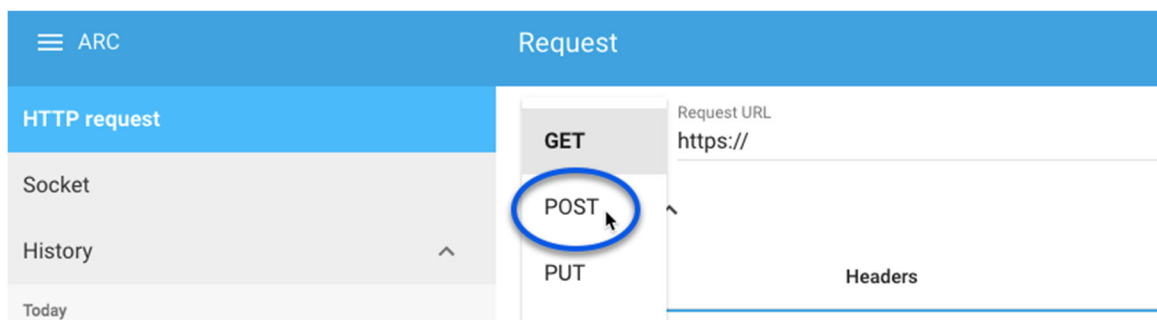
Apple FairPlay Streaming

Below are the steps to create the Key Files for Apple FairPlay Streaming.

To request the DRM keys from EZDRM to package the media, there are two options, you can call the EZDRM Key Servers API, or you can script this process with curl or other web service calls.

Option 1: Request DRM keys using EZDRM Key Servers API

1. To request the DRM keys through Advanced REST client (ARC) API, open a session and select HTTP Request. If you do not have the ARC plug-in for Chrome, see [Appendix 2](#) for instructions.
2. Change the Method dropdown to **POST**.

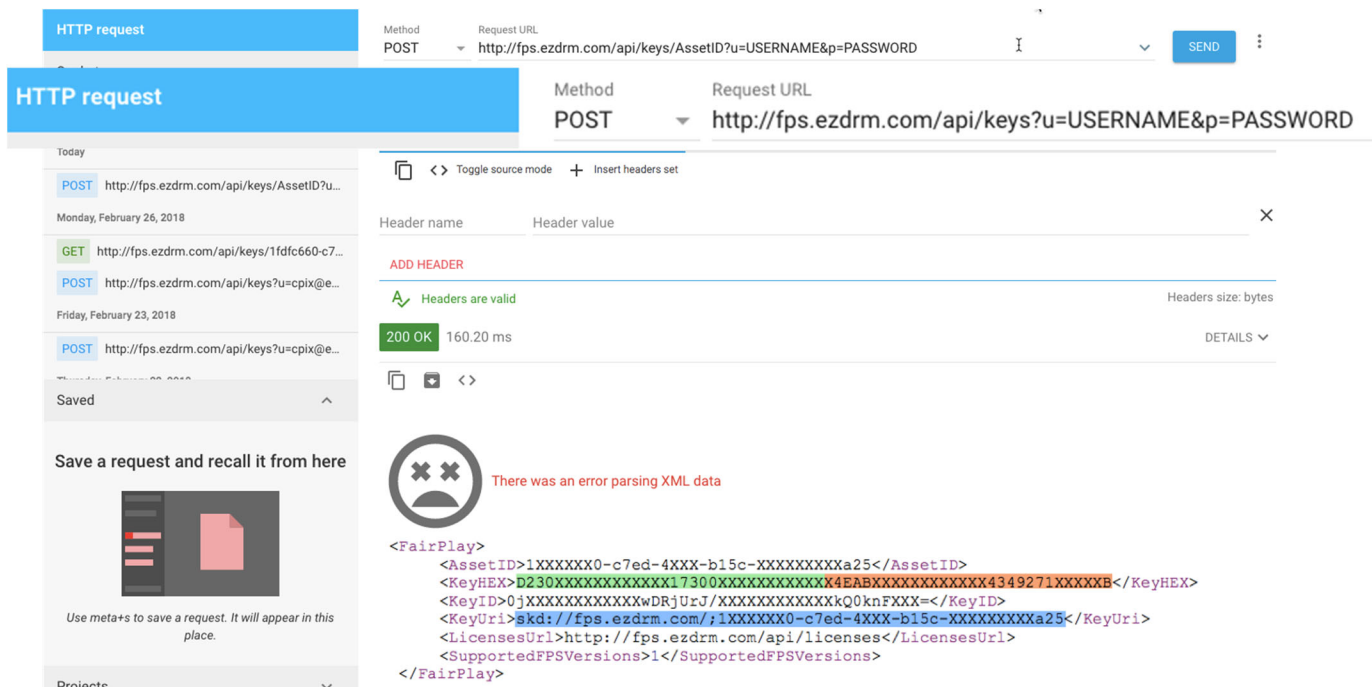


3. Enter the Request URL below updated with your username and password:

<http://fps.ezdrm.com/api/keys?u=USERNAME&p=PASSWORD>

The parameters are as follows:

Parameter	Description
u	EZDRM username
p	EZDRM password



HTTP request

Method POST Request URL http://fps.ezdrm.com/api/keys/AssetID?u=USERNAME&p=PASSWORD

Header name Header value

ADD HEADER

Headers are valid

200 OK 160.20 ms

There was an error parsing XML data

```
<FairPlay>
  <AssetID>1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXXa25</AssetID>
  <KeyHEX>D230XXXXXXXXXXXXX17300XXXXXXXXXXXXX4EABXXXXXXXXXXXXX4349271XXXXXB</KeyHEX>
  <KeyID>0jXXXXXXXXXXXXXwDRjUrJ/XXXXXXXXXXXXXkQ0knFXXX=</KeyID>
  <KeyUri>skd://fps.ezdrm.com/;1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXXa25</KeyUri>
  <LicensesUrl>http://fps.ezdrm.com/api/licenses</LicensesUrl>
  <SupportedFPSVersions>1</SupportedFPSVersions>
</FairPlay>
```

4. The following is an example of the response:

```
<FairPlay>
  <AssetID>1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXXa25</AssetID>
  <KeyHEX>D230XXXXXXXXXXXXX17300XXXXXXXXXXXXX4EABXXXXXXXXXXXXX4349271XXXXXB</KeyHEX>
  <KeyID>0jXXXXXXXXXXXXXwDRjUrJ/XXXXXXXXXXXXXkQ0knFXXX=</KeyID>
  <KeyUri>skd://fps.ezdrm.com/;1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXXa25</KeyUri>
  <LicensesUrl>http://fps.ezdrm.com/api/licenses</LicensesUrl>
  <SupportedFPSVersions>1</SupportedFPSVersions>
</FairPlay>
```

5. A unique **AssetID** is assigned the first time you call the API, to get existing keys see [Appendix 1](#).

Option 2: Request DRM keys with curl

Using EZDRM's web service, the curl script below retrieves the DRM values from the web service.

```
curl -X POST 'http://fps.ezdrm.com/api/keys?U=Username&P=Password' -d ' '
```

Note: there is a space between the single quotes at the end of the line.

The parameters are as follows:

Parameter	Description
U	EZDRM username
P	EZDRM password

The following is returned:

```
<FairPlay>
  <AssetID>1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXa25</AssetID>
  <KeyHEX>D230XXXXXXXXXX1730XXXXXXXXXX4EABXXXXXXXXXX4349271XXXXXB</KeyHEX>
  <KeyID>0jXXXXXXXXXXwDRjUrJ/XXXXXXXXXXkQ0knFXXX=</KeyID>
  <KeyUri>skd://fps.ezdrm.com/;1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXa25</KeyUri>
  <LicensesUrl>http://fps.ezdrm.com/api/licenses</LicensesUrl>
  <SupportedFPSVersions>1</SupportedFPSVersions>
</FairPlay>
```

Here are the descriptions of the items returned:

- **KeyHEX:** the DRM Encryption Key. This is given back in two forms: Base64 and HEX. The first 32 characters form KeyHEX are used in the key file for "key" and the last 32 characters are used for "iv".
- **KeyUri:** the License URL for encryption.

Unified Packaging

Universal (Widevine and PlayReady)

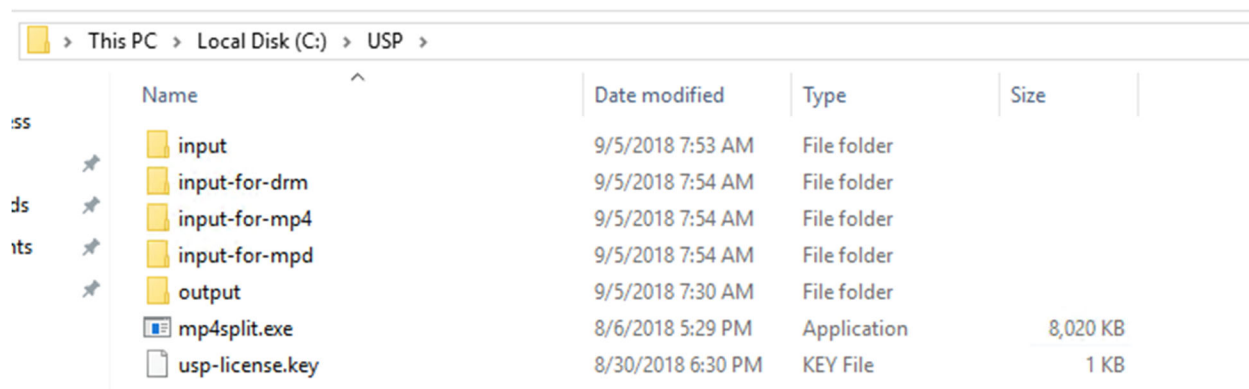
Step 1 – Encrypt fragmented Audio & Video files

License Key

A DRM supported license key from Unified Streaming is required, saved as a USP License Key file.

```
--license-key=usp-license.key
```

You will save in a file in your home directory (where you enter when logging in).



Name	Date modified	Type	Size
input	9/5/2018 7:53 AM	File folder	
input-for-drm	9/5/2018 7:54 AM	File folder	
input-for-mp4	9/5/2018 7:54 AM	File folder	
input-for-mpd	9/5/2018 7:54 AM	File folder	
output	9/5/2018 7:30 AM	File folder	
mp4split.exe	8/6/2018 5:29 PM	Application	8,020 KB
usp-license.key	8/30/2018 6:30 PM	KEY File	1 KB

Encryption

To create Multi-DRM protected content, the following EZDRM DRM Key values are required for Widevine and PlayReady. Replace the DRM KEY values into this command for each audio and video file.

```
mp4split --license-key=usp-license.key
-o <<OUTPUT FILE .CMFA or .CMFV>>
--iss.key=EZDRM_KeyIDHEX:EZDRM_KeyHEX
--iss.license_server_url=PlayReady_LAURL
--widevine.key=EZDRM_KeyIDHEX:EZDRM_KeyHEX
--widevine.drm_specific_data=Widevine_PSSH
<<INPUT FILE .MP4>>
```

The parameters of the configuration have the following meaning:

- **iss.key:** This is the common content encryption key, the EZDRM **KeyIDHEX:KeyHEX** value, divided by a colon (:).
- **iss.license_server_url:** This is the URL to the PlayReady license server, the EZDRM PlayReady **LAURL** value.
- **widevine.key:** This is the common unique identifier for your content key, the EZDRM **KeyIDHEX:KeyHEX** value, divided by a colon (:).
- **widevine.drm_specific_data:** This is the value for the Widevine PSSH box, the EZDRM **PSSH** value.

Audio Example Command

```
mp4split --license-key=usp-license.key -o C:\USP\packager\output\cenc\tos-aac-128k-dash-cenc.cmfa --iss.key=5XXXXXXXX36d85XXXXXXXXXXXXb24XXXb:5bXXXXXXXXX9191fXXe38XXXXXX56bf --iss.license_server_url=https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX --widevine.key=5XXXXXXXX36d85XXXXXXXXXXXXb24XXXb:5bXXXXXXXXX9191fXXe38XXXXXX56bf --widevine.drm_specific_data=EhXXXXXXXXXXXXX6skGLGXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXj3JXXXX== C:\USP\packager\input\tears-of-steel-aac-128k.mp4
```

Video Example Commands

```
mp4split --license-key=usp-license.key -o C:\USP\packager\output\cenc\tos-avc1-400k-dash-cenc.cmfv --iss.key=5XXXXXXXX36d85XXXXXXXXXXXXb24XXXb:5bXXXXXXXXX9191fXXe38XXXXXX56bf --iss.license_server_url=https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX --widevine.key=5XXXXXXXX36d85XXXXXXXXXXXXb24XXXb:5bXXXXXXXXX9191fXXe38XXXXXX56bf --widevine.drm_specific_data=EhXXXXXXXXXXXXX6skGLGXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXj3JXXXX== C:\USP\packager\input\tears-of-steel-avc1-400k.mp4

mp4split --license-key=usp-license.key -o C:\USP\packager\output\cenc\tos-avc1-1000k-dash-cenc.cmfv --iss.key=5XXXXXXXX36d85XXXXXXXXXXXXb24XXXb:5bXXXXXXXXX9191fXXe38XXXXXX56bf --iss.license_server_url=https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX --widevine.key=5XXXXXXXX36d85XXXXXXXXXXXXb24XXXb:5bXXXXXXXXX9191fXXe38XXXXXX56bf --widevine.drm_specific_data=EhXXXXXXXXXXXXX6skGLGXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXj3JXXXX== C:\USP\packager\input\tears-of-steel-avc1-1000k.mp4
```

Step2 – Package MPD

Now that all outputs have been created, run this command to join them into one MPD file using the following sample command:

```
mp4split --license-key=usp-license.key  
        --package_mpd -o <<OUTPUT FILE .MPD>>  
<<INPUT FILE .CMFA>> <<INPUT FILE .CMFV>><<INPUT FILE .CMFV>>
```

Sample command:

```
mp4split --license-key=usp-license.key --package_mpd -o C:\USP\packager\output\cenc\tears-of-steel-cenc.mpd  
d C:\USP\packager\output\cenc\tos-aac-128k-dash-cenc.cmfa C:\USP\packager\output\cenc\tos-avc1-400k-da  
sh-cenc.cmfv C:\USP\packager\output\cenc\tos-avc1-1000k-dash-cenc.cmfv
```

Now Zip or copy contents of \packager\output\cenc for playback on your webserver.

Step 3 – Test Playback

When your packaged files are on the webserver, you can test Universal playback using Shaka Player.

<https://shaka-player-demo.appspot.com/demo/>

Test Widevine playback on Chrome & Firefox, PlayReady on IE and Edge

Note: all calls must be over *https*.

1. Select the **(custom asset)** option from the **Asset** dropdown:

Shaka Player v2.4.5-uncompiled

[uncompiled](#) | [compiled \(debug\)](#) | [compiled \(release\)](#)

This is a demo of Google's Shaka Player, a JavaScript library for adaptive video streaming.
Choose an asset and tap **Load**. (On Android, you may also need to press the play button on the video.)

[Documentation](#) | [Repo](#) | [NPM](#) | [CDNJS](#)

Asset: Angel One (multicodec, multilingual) ▼

Custom manifest: Live sim (6s segments)

Custom license server: Live sim (multi-period)

Custom license certificate URL: Wowza

Load Unload

Big Buck Bunny (Live)

Bitcodin

Art of Motion (DASH)

Art of Motion (HLS, TS)

Sintel (HLS, TS, 4k)

Azure Media Services

Azure Trailer

Tears Of Steel (external text)

GPAC

live profile with five periods

main profile, single file

onDemand profile

Verizon Digital Media Services

Multi DRM - 8 Byte IV

Multi DRM - MultiPeriod - 8 Byte IV

Widevine - 16 Byte IV

Widevine - 16 Byte IV - (mix of encrypted and unencrypted periods)

Offline

(custom asset)

2. To use the player for testing you'll need to edit the following:
- The **Manifest URL** for the packaged MPD
 - The **Proxy URL** - The Widevine **ServerURL** value or PlayReady **LAURL** value from the EZDRM Encryption Key file.

Shaka Player v2.4.5-uncompiled

[uncompiled](#) | [compiled \(debug\)](#) | [compiled \(release\)](#)

This is a demo of Google's Shaka Player, a JavaScript library for adaptive video streaming.

Choose an asset and tap **Load**. (On Android, you may also need to press the play button on the video.)

[Documentation](#) | [Repo](#) | [NPM](#) | [CDNJS](#)

Asset:

Custom manifest:

Custom license server:

Custom license certificate URL:

Load

Unload



- Click the Load button to see the test playback:

Shaka Player v2.4.5-uncompiled

[uncompiled](#) | [compiled \(debug\)](#) | [compiled \(release\)](#)

This is a demo of Google's Shaka Player, a JavaScript library for adaptive video streaming.

Choose an asset and tap **Load**. (On Android, you may also need to press the play button on the video.)

[Documentation](#) | [Repo](#) | [NPM](#) | [CDNJS](#)

Asset:	(custom asset)
Custom manifest:	https://wowza.ezdrm.com/vod/mp4:sample.mp4/r
Custom license server:	https://widevine-dash.ezdrm.com/proxy?pX=XXXX
Custom license certificate URL:	

Load

Unload



For more information, refer to EZDRM documentation on testing playback.

Multi-Key Encryption

To create DRM protected content using different DRM Keys for each audio or video stream, you will want to request separate keys for each file you want to encode. This may be beneficial for analytics tracking, for different audio track languages or video bitrates. You can use as many key sets as needed.

For this example of Universal encryption, we will use one set of DRM keys for the audio track and a separate key set for the video files.

1. Create two separate calls for EZDRM keys to get the two key sets.

Example Set 1

KeyIDHEX: f002XXXXXXXXXXXXXXXXXXXXXXXXXXXXfa

KeyHEX: 6cXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX64

Example Set 2

KeyIDHEX: 497bXXXXXXXXXXXXXXXXXXXXXXXXXXXXb7

KeyHEX: 20XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX09

2. For this example, we use the first key set for the audio file, and the second set for each of the video files.

Audio Example – Set 1 (readable):

```
mp4split --license-key=usp-license.key
-o C:\USP\packager\output\cenc-multikey\tos-aac-128k-dash-cenc.cmfa
--iss.key=f002XXXXXXXXXXXXXXXXXXXXXXXXXXXXfa:6cXXXXXXXXXXXXXXXXXXXXXXXXXXXX64
--iss.license_server_url=https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX
--widevine.key=f002XXXXXXXXXXXXXXXXXXXXXXXXXXXXfa:6cXXXXXXXXXXXXXXXXXXXXXXXXXXXX64
--widevine.drm_specific_data=EhXXXXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXX
X==
C:\USP\packager\input\tears-of-steel-aac-128k.mp4
```

Video Example – Set 2 (readable):

```
mp4split --license-key=usp-license.key
-o C:\USP\packager\output\cenc-multikey\tos-avc1-400k-dash-cenc.cmfv
--iss.key=497bXXXXXXXXXXXXXXXXXXXXXXXXXXXXb7:20XXXXXXXXXXXXXXXXXXXXXXXXXXXX09
```



```
--iss.license_server_url=https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX
--widevine.key=497bXXXXXXXXXXXXXXXXXXXXb7:20XXXXXXXXXXXXXXXXXXXXXXXXXXXX09
--widevine.drm_specific_data=EhXXXXXXXXXXXXX6sKGLGXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXj3JXXX
X==
C:\USP\packager\input\tears-of-steel-avc1-400k.mp4

mp4split --license-key=usp-license.key
-o C:\Sandbox\USP\packager\output\cenc-multikey\tos-avc1-1000k-dash-cenc.cmfv
--iss.key=497bXXXXXXXXXXXXXXXXXXXXb7:20XXXXXXXXXXXXXXXXXXXXXXXXXXXX09
--iss.license_server_url=https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX
--widevine.key=497bXXXXXXXXXXXXXXXXXXXXb7:20XXXXXXXXXXXXXXXXXXXXXXXXXXXX09
--widevine.drm_specific_data=EhXXXXXXXXXXXXX6sKGLGXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXj3JXXX
X==
C:\USP\packager\input\tears-of-steel-avc1-1000k.mp4
```

- Now that you have the encrypted files continue by creating the MPD file same way as illustrated previously:

Sample command:

```
mp4split --license-key=usp-license.key --package_mpd -o C:\USP\packager\output\cenc-multikey\tears
-of-steel-cenc-multikey.mpd
C:\USP\packager\output\cenc-multikey\tos-aac-128k-dash-cenc.cmfa
C:\USP\packager\output\cenc-multikey\tos-avc1-400k-dash-cenc.cmfv
C:\USP\packager\output\cenc-multikey\tos-avc1-1000k-dash-cenc.cmfv
```

- Zip or copy contents of \packager\output\cenc-multikey for playback on your webserver as previously illustrated.

Apple FairPlay Streaming

To create DRM protected content, the following EZDRM DRM Key values are required for Apple FairPlay Streaming. This example is for saving the Unified license key as a key file.

```
mp4split -licence-key=usp-license.key
--package hls
-o playready.ism <<OUTPUT FILE>>
--fragment duration=9000
--hls.key=EZDRM KeyHEX (first 32):EZDRM KeyHEX (last 32)
--hls.license_server_url=EZDRM KeyUri
<<INPUT FILE>>
--track_type=video
```

Here are the descriptions of the items returned:

- **hls.key:** the 128-bit content encryption key. This is the first 32 characters from the EZDRM **KeyHEX** value, a colon, and then the initialization vector (IV) used for encryption, which is the last 32 characters in the EZDRM **KeyHEX** value.
- **hls.license_server_url:** the License URL used for the license key call, this is the EZDRM **KeyUri** value.

Example Command

```
mp4split --license-key=usp-license.key
--package_hls -o C:\USP\packager\output\fps\fairplay.m3u8 --fragment_duration=9000 --hls.key=D230XXXXXX
XXXXX17300XXXXXXXXXXXX:X4EABXXXXXXXXXXXX4349271XXXXXB --hls.license_server_url=skd://fps.ezdrm.com/;1XXXXX
X0-c7ed-4XXX-b15c-XXXXXXXXXXa25 C:\USP\input\ren.mp4 --track_type=video
```

Sample Players for Apple FairPlay Streaming

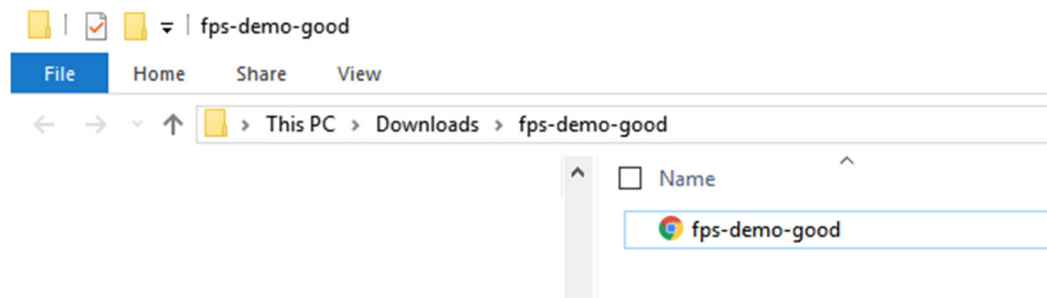
For an iOS offline example, you can download using this link:
<http://www.ezdrm.com/downloads/fps-ios-demo-updates-3.zip>

To use the player for testing in Safari you'll need to:

- Edit the Cert file
- Edit the M3U8

Another player example can be downloaded using this link:

<http://www.ezdrm.com/downloads/fps-demo-good.zip>



To use this player for testing you'll need to modify the following lines in the HTML file:

```

fps-demo-good - Notepad
File Edit Format View Help
<!DOCTYPE html>
<html>
<!-- Portions Copyright (C) 2015 Apple Inc. All Rights Reserved.
See the Apple Developer Program License Agreement for this file's licensing information.
All use of these materials is subject to the terms of the Apple Developer Program License Agreement.
Copyright (C) 2016 DigitalPrimates
-->

<!-- Usage Notes:

Adjust the paths below (marked with '<<' '>>') -- Lines 29 - 31

-->

<head>
<title>FPS Demo Application</title>
<script>
/*
The EME specification (https://dvcs.w3.org/hg/html-media/raw-file/tip/encrypted-media/encrypted-media.html)
is supported starting OSX 10.10 and greater.
*/
var keySystem;
var certificate;

// ##### -----PATHS TO ADJUST----- #####
// -----
// -----

var serverCertificatePath = 'http://URL TO Cert.cer'; // ADAPT: This is the path to the fps certificate on your server.
var serverProcessSPCPath = 'http://fps.ezdrm.com/api/licenses/Edit AssetID'; // ADAPT: This is the path/URL to the keyserver module that processes the SPC and returns a CKC
var serverMediaSourcePath = 'http://URL to Stream.m3u8';

function stringToArray(string) {
var buffer = new ArrayBuffer(string.length*2); // 2 bytes for each char
var array = new Uint16Array(buffer);
for (var i=0, strLen=string.length; i<strLen; i++) {
array[i] = string.charCodeAt(i);
}
return array;
}

```

1. **var serverCertificatePath** – replace with the path to the FPS certificate on your server
2. **var serverProcessSPCPath** – edit the **AssetID** provided with your EZDRM keys (see next section on [Generating Keys](#))

3. **var serverMediaSourcePath** – replace with the URL path to your M3U8 file

Here's an example:

```
var serverCertificatePath = 'http://www.site.com/fps.cer'; // ADAPT: This is the path to the f  
ps certificate on your server.
```

```
var serverProcessSPCPath = 'http://fps.ezdrm.com/api/licenses/1XXXXXX0-c7ed-4XXX-b15c-XXXXXXX  
XXa25&customdata=123'; // ADAPT: This is the path/URL to the keyserver module that processes t  
he SPC and returns a CKC
```

```
var serverMediaSourcePath = 'https://wowza.ezdrm.com/vod/mp4:sample.mp4/manifest.m2u8';
```

Appendix 1 – Call for Existing EZDRM Keys

Widevine and PlayReady

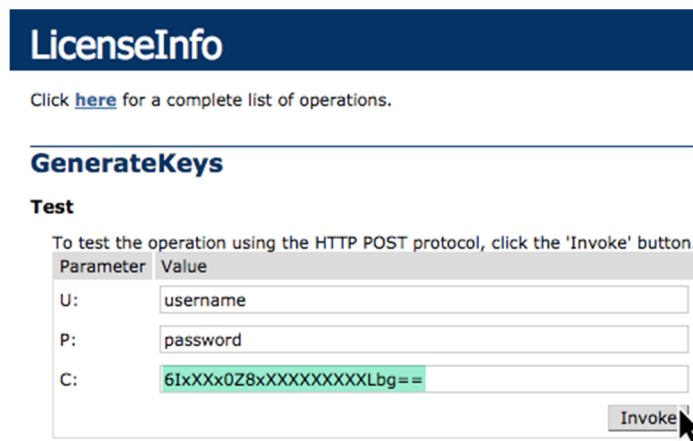
Sending a Content_ID for existing keys will allow you to encrypt content with the same DRM values as other content and have that content share one license. If you call a Content_ID you will get the back all the DRM key information for that Content_ID.

Option 1: Request existing DRM keys using EZDRM Web Service

- a. Call the EZDRM web service in a browser:

<https://wvm.ezdrm.com/ws/LicenseInfo.aspx?op=GenerateKeys>

- b. Return Key values by entering the parameter values including the existing Content_ID and click “Invoke”.



LicenseInfo

Click [here](#) for a complete list of operations.

GenerateKeys

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
U:	username
P:	password
C:	6IxXXx0Z8XXXXXXXXXXLbg==

Invoke

The parameters are as follows:

Parameter	Description
u	EZDRM username
p	EZDRM password
c	<u>Content ID</u>

- c. Your return results will always return the existing DRM keys that are tied to the Content_ID.

```
<EZDRM xmlns="">
  <WideVine diffgr:id="WideVine1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <ContentID>6Ixxx0Z8xxxxxxxxLbg=</ContentID>
    <Key>W5xxxxxxxxZhxTjhxxxxxvw=</Key>
    <KeyHEX>5bxxxxxxxx9191fxxe38xxxxx56bf</KeyHEX>
    <KeyID>WVxxxxxxxxliBEXxw+xxxx=</KeyID>
    <KeyIDGUID>5xxxxxx3-36xx-5xx8-8xx1-10xxxxxxxxxb</KeyIDGUID>
    <KeyIDHEX>5xxxxxx36d85xxxxxxxxxxb24xxxb</KeyIDHEX>
    <PSSH>
      EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXXX==
    </PSSH>
    <ServerURL>https://widevine-dash.ezdrm.com/proxy?pX=XXXXXX</ServerURL>
    <ServerGet>
      request={"policy": "", "tracks": [ {"type": "SD"}], "content_id": "6Ixxx0Z8xxxxxxxxLbg="}
    </ServerGet>
    <ResponseRaw>
      {"status":"OK","drm":[{"type":"WIDEVINE","system_id":"edef8ba979d64acea3c827dcd51d21ed"}],"tracks":
      [{"type":"SD","key_id":"WVxxxxxxxxliBEXxw+xxxx","key":"WVxxxxxxxxliBEXxw+xxxx","pssh":
      [{"drm_type":"WIDEVINE","data":"EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXXX="}]}]}
    </ResponseRaw>
  </WideVine>
  <PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <Key>W5xxxxxxxxZhxTjhxxxxxvw=</Key>
    <KeyHEX>5bxxxxxxxx9191fxxe38xxxxx56bf</KeyHEX>
    <KeyIDGUID>5xxxxxx3-36xx-5xx8-8xx1-10xxxxxxxxxb</KeyIDGUID>
    <LAURL>
      https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX
    </LAURL>
    <Checksum>1Xq+XXXXXX0=</Checksum>
  </PlayReady>
</EZDRM>
```

Option 2: Request existing DRM keys with curl

The second option to request existing DRM keys from EZDRM is to script the process with curl or another web service call.

Using EZDRM's web service, the curl script below retrieves the DRM values from the web service with the existing DRM keys based on the Content_ID.

```
curl -v 'http://wvm.ezdrm.com/ws/LicenseInfo.asmx/GenerateKeys?U=EZDRM_USERNAME&P=EZDRM_PASSWORD&C=CONTENT_ID''
```

The parameters are as follows:

Parameter	Description
U	EZDRM username
P	EZDRM password
C	<u>Content ID</u>

The web service will always return the existing DRM keys that are tied to the Content_ID.

Sample script with Content_ID:

```
curl -v 'http://wvm.ezdrm.com/ws/LicenseInfo.asmx/GenerateKeys?U=EZDRM_USERNAME&P=EZDRM_PASSWORD&C=6Ixx0Z8xXXXXXXXXLbg==Z8xXXXXXXXXLbg=='
```

Existing DRM Keys returned:

```
<EZDRM xmlns="">
  <WideVine diffgr:id="WideVine1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <ContentID>6Ixx0Z8xXXXXXXXXLbg==</ContentID>
    <Key>W5XXXXXXXXZHxTjhXXXXvw==</Key>
    <KeyHEX>5bXXXXXXXX9191fXXe38XXXX56bf</KeyHEX>
    <KeyID>WVXXXXXXXXliBEXw+XXXX==</KeyID>
    <KeyIDGUID>5XXXXX3-36XX-5XX8-8XX1-10XXXXXXXXb</KeyIDGUID>
    <KeyIDHEX>5XXXXX36d85XXXXXXXXXXXXb24XXb</KeyIDHEX>
    <PSSH>EhXXXXXXXXXXXXX6skGLGXXXXXXXXQ6IebXXZ8kSYMXXXXXXXXXXXXXj3JXXXX==</PSSH>
    <ServerURL>https://widevine-dash.ezdrm.com/proxy?pX=XXXXXX</ServerURL>
    <ServerGet>request={"policy": "", "tracks": [ {"type": "SD"}], "content_id": "6Ixx0Z8xXXXXXXXX
  XXLbg=="}</ServerGet>
    <ResponseRaw>
      {"status":"OK", "drm":[{"type":"WIDEVINE", "system_id":"edef8ba979d64acea3c827dcd51d21ed"}], "tracks":[{"type":
      "SD", "key_id": " WVXXXXXXXXliBEXw+XXXX=", "key": " W5XXXXXXXXZHxTjhXXXXvw=", "pssh":[{"drm_type":"WIDEVIN
      E", "data": "EhXXXXXXXXXXXXX6skGLGXXXXXXXXQ6IebXXZ8kSYMXXXXXXXXXXXXXj3JXXXX="}]}]}</ResponseRaw>
    </WideVine>
    <PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
      <Key>W5XXXXXXXXZHxTjhXXXXvw==</Key>
      <KeyHEX>5bXXXXXXXX9191fXXe38XXXX56bf</KeyHEX>
      <KeyIDGUID>5XXXXX3-36XX-5XX8-8XX1-10XXXXXXXXb</KeyIDGUID>
      <LAURL>https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX</LAURL>
      <Checksum>1Xq+XXXXXX0=</Checksum>
    </PlayReady>
  </EZDRM>
```

Apple FairPlay Streaming

Sending a AssetID for existing keys will allow you to encrypt content with the same DRM values as other content and have that content share one license. If you call an AssetID you will get the back all the DRM key information for that AssetID.

Option 1: Request existing DRM keys using EZDRM Key Servers API

- To request existing DRM keys through Advanced REST client (ARC) API, open a session and select HTTP Request. If you do not have the ARC plug-in for Chrome, see [Appendix 2](#) for instructions.
- Change the Method dropdown to **GET**.

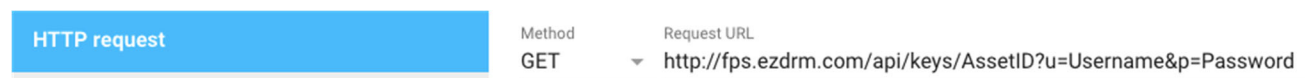


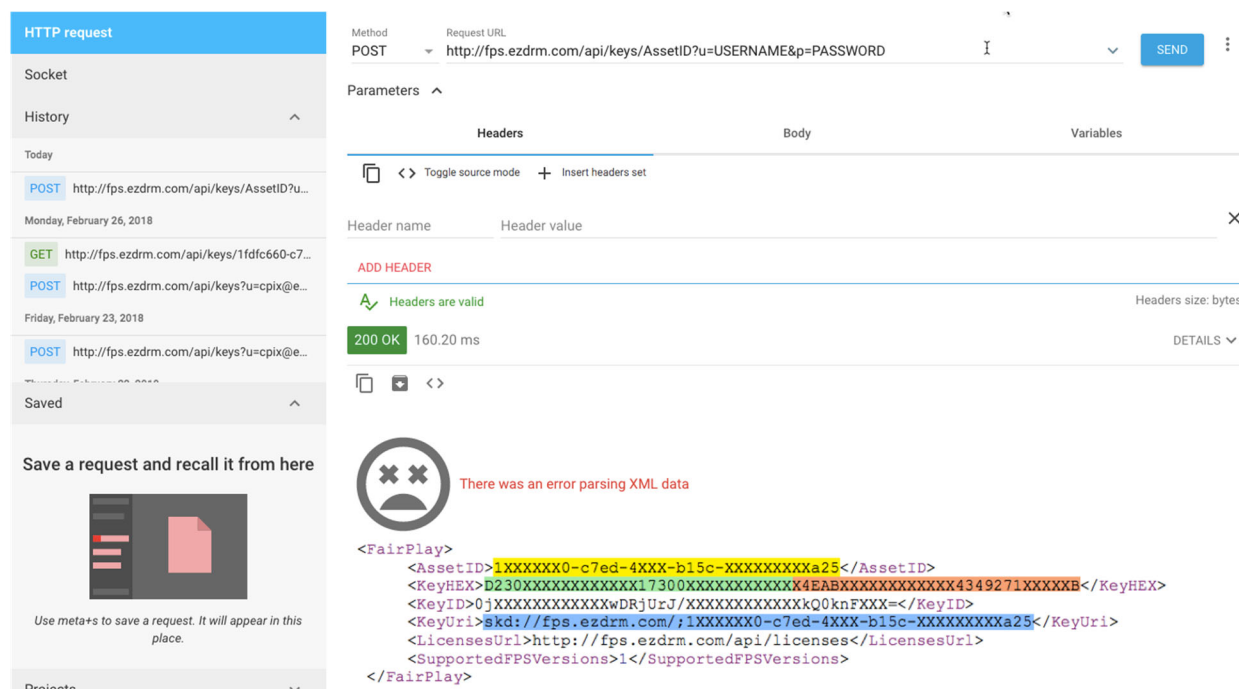
- Enter the Request URL below updated with your AssetID, username, and password:

<http://fps.ezdrm.com/api/keys/AssetID?u=Username&p=Password>

The parameters are as follows:

Parameter	Description
<u>AssetID</u>	Existing FairPlay AssetID
u	EZDRM username
p	EZDRM password





HTTP request

Method: POST, Request URL: http://fps.ezdrm.com/api/keys/AssetID?u=USERNAME&p=PASSWORD

Parameters: Headers, Body, Variables

Header name: Header value

ADD HEADER

Headers are valid

200 OK, 160.20 ms

There was an error parsing XML data

```
<FairPlay>
  <AssetID>1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXXa25</AssetID>
  <KeyHEX>B230XXXXXXXXXXXXX17300XXXXXXXXXXXXX4EABXXXXXXXXXXXXX4349271XXXXXB</KeyHEX>
  <KeyID>0jXXXXXXXXXXXXXDRjUrJ/XXXXXXXXXXXXXkQ0knFXXX=</KeyID>
  <KeyUri>skd://fps.ezdrm.com/;1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXXa25</KeyUri>
  <LicenseUrl>http://fps.ezdrm.com/api/licenses</LicenseUrl>
  <SupportedFPSVersions>1</SupportedFPSVersions>
</FairPlay>
```

The GET will always return the existing DRM keys that are tied to the **AssetID**.

Option 2: Request DRM keys with curl

Using EZDRM's web service, the curl script below retrieves the DRM values from the web service.

```
curl -X POST 'http://fps.ezdrm.com/api/keys/AssetID?U=Username&P=Password' -d ' '
```

Note: there is a space between the single quotes at the end of the line.

The parameters are as follows:

Parameter	Description
AssetID	Existing FairPlay AssetID
U	EZDRM username
P	EZDRM password

The response will always return the existing DRM keys that are tied to the **AssetID**.

Sample curl script with AssetID:

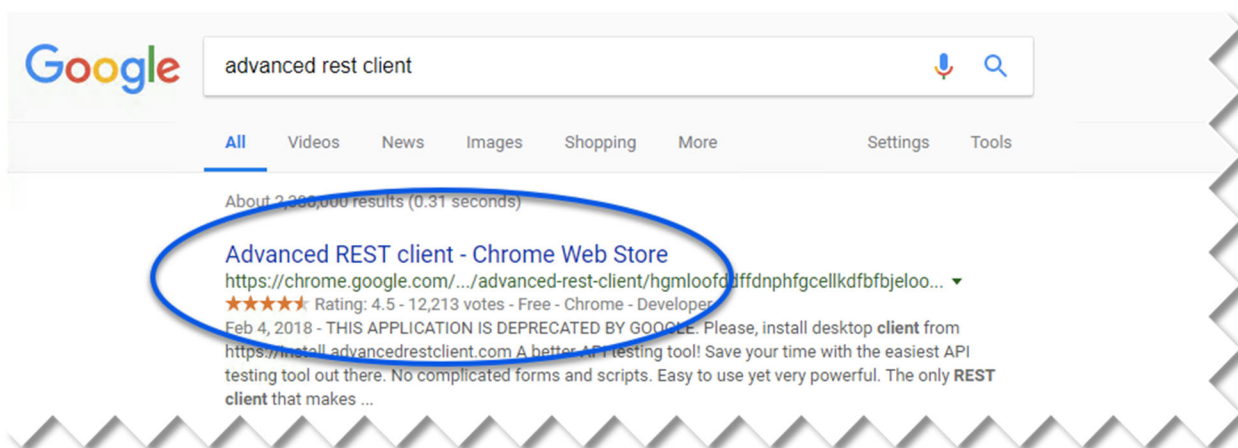
```
curl -X POST 'http://fps.ezdrm.com/api/keys/1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXa25?U=Username&P=Password' -d  
_ '
```

Existing DRM Keys returned:

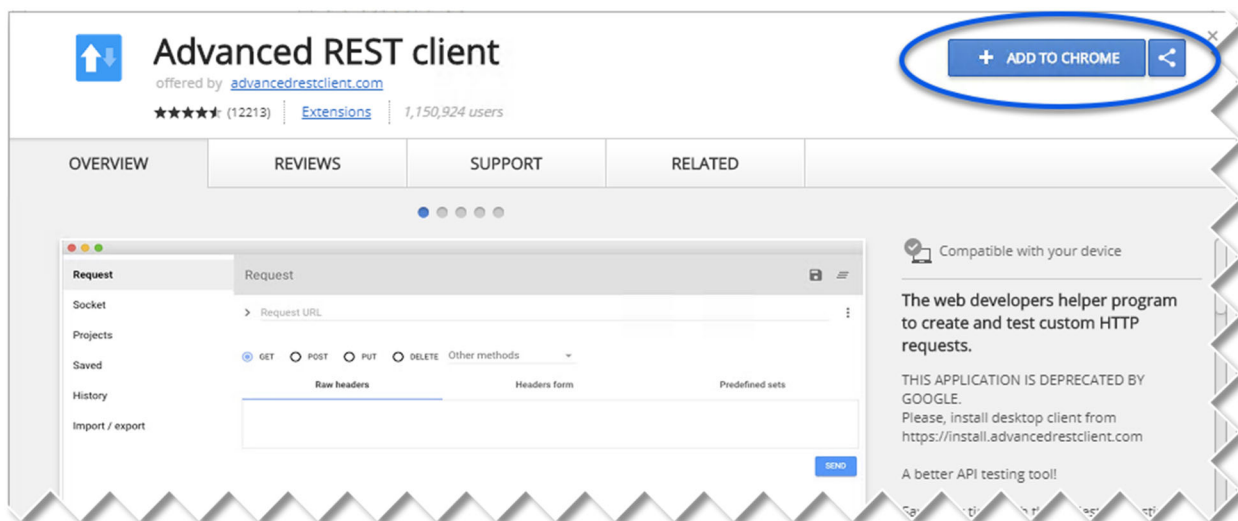
```
<FairPlay>  
  <AssetID>1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXa25</AssetID>  
  <KeyHEX>D230XXXXXXXXXXXX1730XXXXXXXXXXXX4EABXXXXXXXXXXXX4349271XXXXXB</KeyHEX>  
  <KeyID>0jXXXXXXXXXXXXwDRjUrJ/XXXXXXXXXXXXkQ0knFXXX=</KeyID>  
  <KeyUri>skd://fps.ezdrm.com/;1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXa25</KeyUri>  
  <LicensesUrl>http://fps.ezdrm.com/api/licenses</LicensesUrl>  
  <SupportedFPSVersions>1</SupportedFPSVersions>  
</FairPlay>
```

Appendix 2 – Installing ARC Plug-in

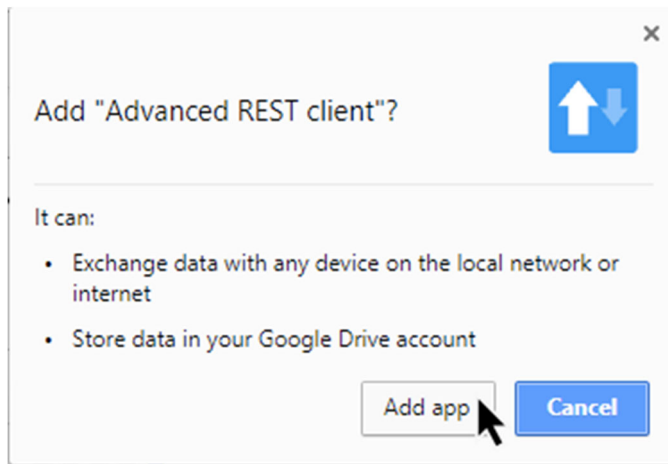
- To install the Advanced REST client (ARC) plug-in for Chrome in order to call the EZDRM Key Servers API, search for “Advanced REST client” using the Google search engine.
- Click the “Advanced REST client – Chrome Web Store” link.



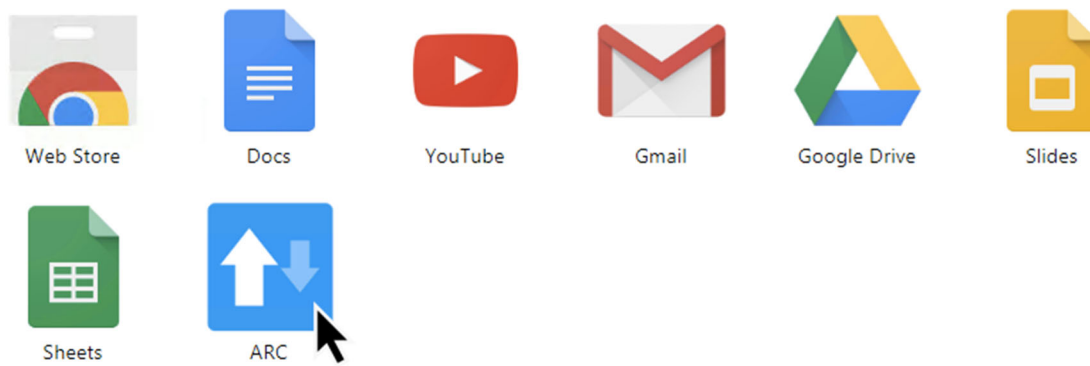
- Click on the **+Add to Chrome** button to install the plug-in.



- d. Click the **Add app** button to confirm installation.



- e. Open the plug-in by clicking the ARC app button.



Return to [Apple FairPlay Streaming](#) section to continue instructions for calling the EZDRM Key servers API.