

EZDRM Unified Streaming

Origin

Table of Contents

Prerequisites	3
Generating DRM Keys	4
<i>Widevine and PlayReady</i>	4
<i>Apple FairPlay Streaming</i>	7
Unified License Key File	10
Unified Origin Configuration	11
<i>Universal (Widevine and PlayReady) DRM protected ISM files</i>	11
<i>Widevine DRM protected ISM files</i>	12
<i>Microsoft PlayReady DRM protected ISM files</i>	13
<i>Apple FairPlay Streaming DRM protected ISM files</i>	13
Playback of Encrypted Content	15
<i>Widevine and PlayReady</i>	15
<i>Apple FairPlay Streaming</i>	19
Appendix 1 – Call for Existing EZDRM Keys	22
<i>Widevine and PlayReady</i>	22
<i>Apple FairPlay Streaming</i>	25
Appendix 2 – Installing ARC Plug-in	28

Version 1

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.

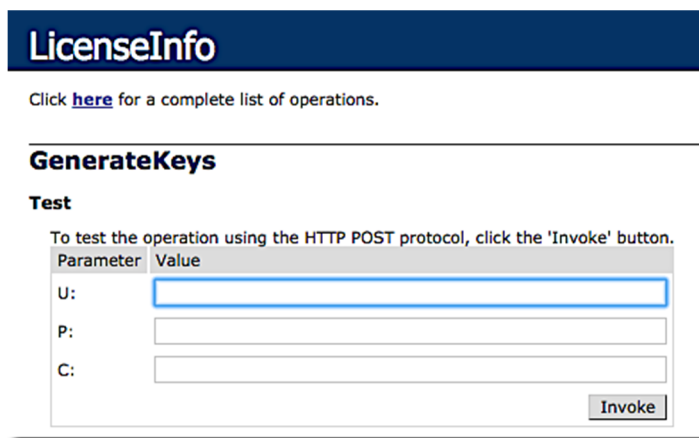
To request the DRM keys from EZDRM to package the media, there are two options, you can call the EZDRM web service in a browser, or you can script this process with curl or other web service calls.

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>6IxXXx0Z8xXXXXXXXXXlbg==</ContentID>
    <Key>W5XXXXXXXX2HxTjhXXXXXv==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXXXXX56bf</KeyHEX>
    <KeyID>WVXXXXXXXXliBEXXw+XXXX==</KeyID>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
    <KeyIDHEX>5XXXXXXXX36d85XXXXXXXXXXXXb24XXb</KeyIDHEX>
    <PSSH>
      EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6IebXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXXX==
    </PSSH>
    <ServerURL>https://widevine-dash.ezdrm.com/proxy?pX=XXXXXX</ServerURL>
    <ServerGet>
      request={"policy": "", "tracks": [ {"type": "SD"}], "content_id": "6IxXXx0Z8xXXXXXXXXXlbg==" }
    </ServerGet>
    <ResponseRaw>
      {"status":"OK","drm":{"type":"WIDEVINE","system_id":"edef8ba979d64acea3c827dcd51d2led"},"tracks":
      [{"type":"SD","key_id":"WVXXXXXXXXliBEXXw+XXXX==","key":"WVXXXXXXXXliBEXXw+XXXX==","pssh":
      [{"drm_type":"WIDEVINE","data":"EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6IebXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXXX=="}]}]}
    </ResponseRaw>
  </WideVine>
  <PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <Key>W5XXXXXXXX2HxTjhXXXXXv==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXXXXX56bf</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 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.asmx/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": "6IxXXx0Z8xXXXXXXXXXX
  Lbg=="}</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>
```

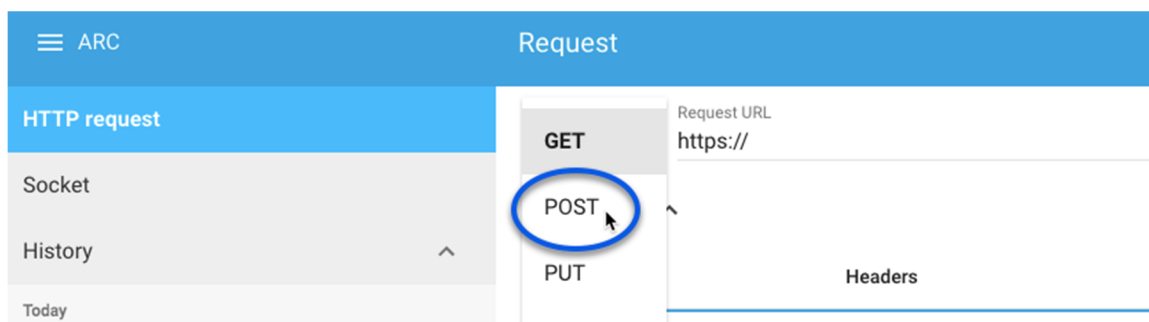
Apple FairPlay Streaming

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**.

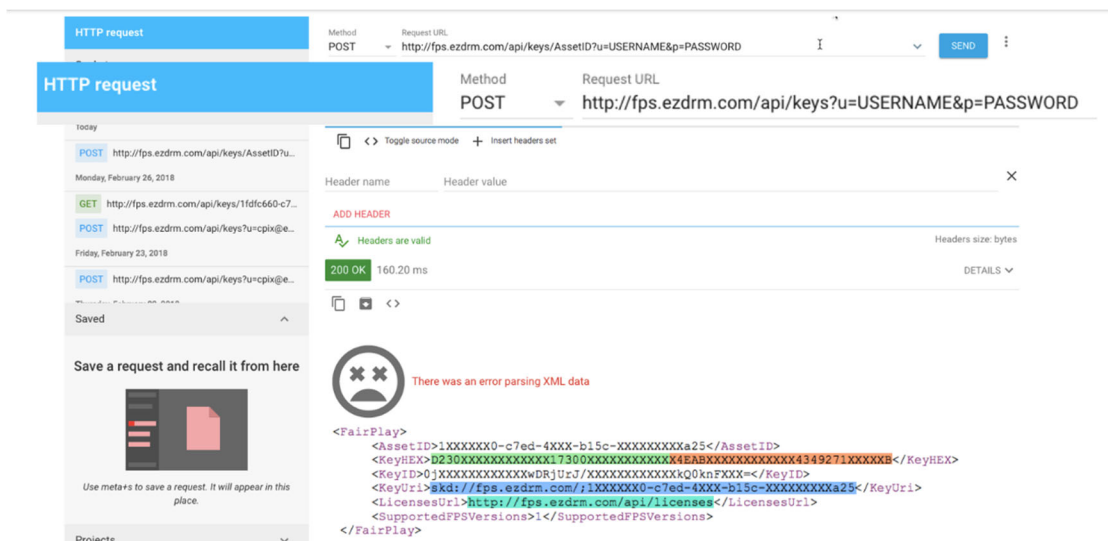


3. Enter the Request URL below updated with your username and password:

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

The parameters are as follows:

Parameter	Description
u	EZDRM username
p	EZDRM password



4. The following is an example of the response:

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

Here are the descriptions of the items returned:

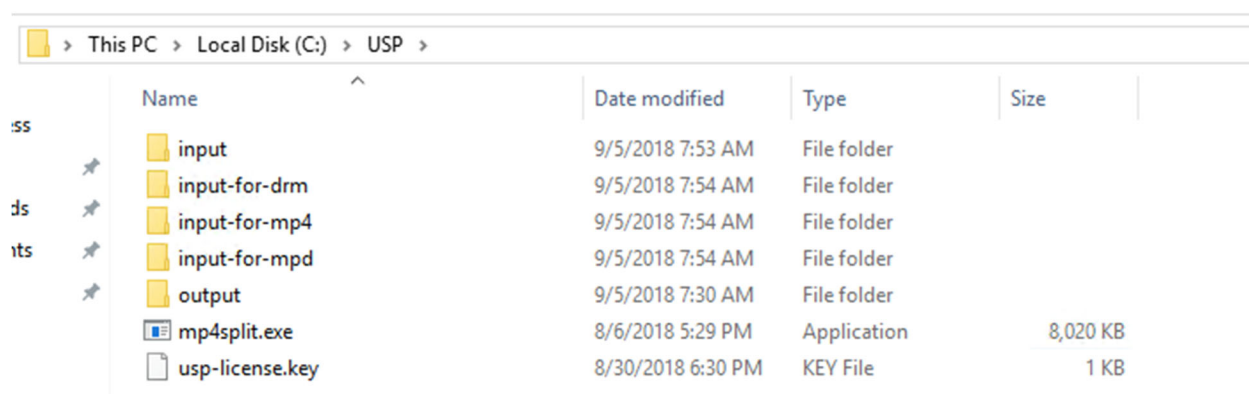
- **KeyHEX:** the DRM Encryption Key. This is given back in two forms: Base64 and HEX. The first 32 characters form KeyHEX are used in the key file for "key" and the last 32 characters are used for "iv".
- **KeyUri:** the License URL for encryption.
- **LicenseURL:** the License URL used for the license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

Unified License Key File

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

Unified Origin Configuration

Universal (Widevine and PlayReady) DRM protected ISM files

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:

```
mp4split --license-key=usp-license.key -o <<OUTPUT FILE>>
--iss.key=KeyIDHEX:EZDRM_KeyHEX
--iss.license_server_url=PlayReady_LAURL
--widevine.key=KeyIDHEX:EZDRM_KeyHEX
--widevine.license_server_url=Widevine_ServerURL
--widevine.drm_specific_data=Widevine_PSSH
<<INPUT FILE>>
```

NOTE: The widevine.key needs a colon (:) between KeyIDHEX and KeyHEX. The iss.key needs a colon (:) between KeyIDHEX and KeyHEX.

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.license.server_url:** This is the URL to the Widevine license server, the EZDRM Widevine **ServerURL** value.
- **widevine.drm_specific_data:** This is the value for the Widevine PSSH box, the EZDRM **PSSH** value.

Example Command

```
mp4split --license-key=usp-license.key -o DASH/output/widevineANDplayready.ism --iss.key=5XXXXXX36d85XXXXX
XXXXXXXXb24XXXb:5bXXXXXXXXXX9191fXXe38XXXXXX56bf --iss.license_server_url=https://playready.ezdrm.com/cenc
y/preauth.aspx?px=XXXXXX --widevine.key=5XXXXXX36d85XXXXXXXXXXXXb24XXXb:5bXXXXXXXXXX9191fXXe38XXXXXX56bf
--widevine.license_server_url=https://widevine-dash.ezdrm.com/proxy?px=BXXXXX --widevine.drm_specific_data
=EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXXXX== input\ren.mp4
```

Widevine DRM protected ISM files

To create DRM protected content, the following EZDRM DRM Key values are required for Widevine. Replace the DRM KEY values into this command:

```
mp4split --license-key=usp-license.key -o <<OUTPUT FILE>>
--widevine.key=KeyIDHEX:KeyHEX
--widevine.license_server_url=Widevine ServerURL
--widevine.drm_specific_data=Widevine PSSH
<<INPUT FILE>>
```

NOTE: The widevine.key needs a colon (:) between KeyIDHEX and KeyHEX.

The parameters of the configuration have the following meaning:

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

Example Command

```
mp4split --license-key=usp-license.key -o widevine/output/widevine.ism --widevine.key=5XXXXXX36d85XXXXXXXXX
XXXXb24XXXb:5bXXXXXXXXXX9191fXXe38XXXXXX56bf --widevine.license_server_url=https://widevine-dash.ezdrm.co
m/proxy?px=BXXXXX --widevine.drm_specific_data=EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXX
XXXXj3JXXXXX== input\ren.mp4
```

Microsoft PlayReady DRM protected ISM files

To create DRM protected content, the following EZDRM DRM Key values are required for PlayReady. Replace the DRM KEY values into this command:

```
mp4split --license-key=usp-license.key -o <<OUTPUT FILE>>
--iss.key=KeyIDHEX:EZDRM KeyHEX
--iss.license_server_url=PlayReady LAURL
<<INPUT FILE>>
```

NOTE: The iss.key needs a colon (:) between KeyIDHEX and KeyHEX.

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.

Example Command

```
mp4split --license-key=usp-license.key -o playready/output/playready.ism --iss.key=5XXXXXX36d85XXXXXXXXXX
XXb24XXXb:5bXXXXXXXXXX9191fXXe38XXXXXXXX56bf --iss.license_server_url=https://playready.ezdrm.com/cency/prea
uth.aspx?pX=XXXXXX input\ren.mp4
```

Apple FairPlay Streaming DRM protected ISM files

To create DRM protected content, the following EZDRM DRM Key values are required for Apple FairPlay Streaming:

```
mp4split --license-key=usp-license.key -o <<OUTPUT FILE>>
--hls.key=EZDRM KeyHEX (first 32):EZDRM KeyHEX (last 32)
--hls.license_server_url=EZDRM LicensesURL
--hls.playout=sample_aes_streamingkeydelivery
test.mp4
```

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 and the last 32 characters in the EZDRM **KeyHEX** value, separated by a colon.
- **hls.license_server_url:** the License URL used for the license key call, this is the EZDRM **LicenseURL** value.

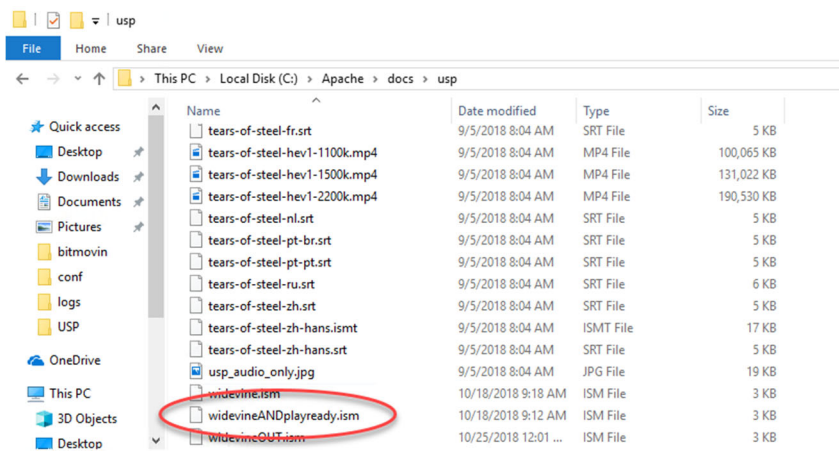
Example Command

```
mp4split --license-key=usp-license.key -o fps/output/fairplay.ism --hls.key=D230XXXXXXXXXXXX17300XXXXXXXXXX:XX:X4EABXXXXXXXXXXXX4349271XXXXXB --hls.license_server_url=http://fps.ezdrm.com/api/licenses --hls.playout=sample_aes_streamingkeydelivery input/ren.mp4
```

Playback of Encrypted Content

Widevine and PlayReady

Once you have your output .ISM files created, upload them to your web folder for playback on the webserver, as shown in the example below:



An example playback player for Universal content is the Shaka Player found here: <https://shaka-player-demo.appspot.com/demo/>

Note: Widevine is for Chrome & Firefox, PlayReady is for IE or Edge. Also https must be used.

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)

Live sim (multi-period)

Custom license server:

Wowza

Big Buck Bunny (Live)

Custom license certificate URL:

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)

Load

Un

2. To use the player for testing you'll need to edit the following:

- The Manifest URL for the MPD file
 - Our example:
<https://origin.ezdrm.com/widevineANDplayready.ism/.mpd>
- The Proxy URL
 - The Widevine **ServerURL** value or PlayReady **LAURL** value.

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



3. 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



When testing playback, you can also test sending your Custom Data. Enter the Custom Data as part of the Proxy URL as shown in the URL above and the example below:

Asset:	(custom asset)
Custom manifest:	https://origin.ezdrm.com/widevineANDplayready.is
Custom license server:	zdrm.com/proxy?pX=Bl[redacted]&customdata=12345
Custom license certificate URL:	

Load
Unload



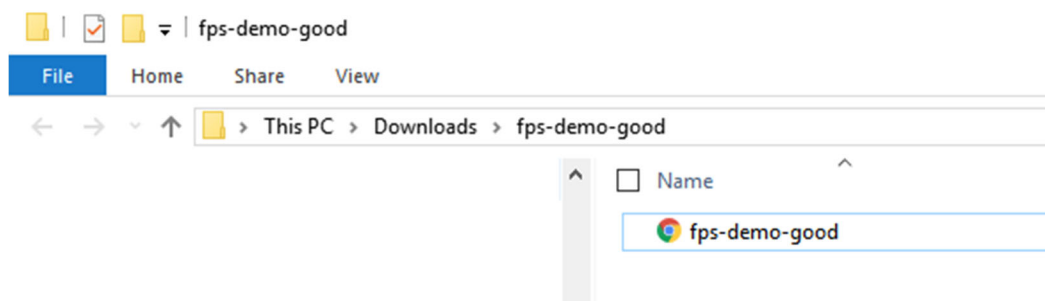
Apple FairPlay Streaming

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 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, stLen=string.length; i<stLen; i++) {
          array[i] = string.charCodeAt(i);
        }
        return array;
      }
    </script>
  </head>
</html>

```

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 ISM file

Here's an example:

```
var serverCertificatePath = 'http://www.site.com/fps.cer'; // ADAPT: This is the path to the fps certificate on your server.  
var serverProcessSPCPath = 'http://fps.ezdrm.com/api/licenses/1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXa25&custom  
data=12'; // ADAPT: This is the path/URL to the keyserver module that processes the 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

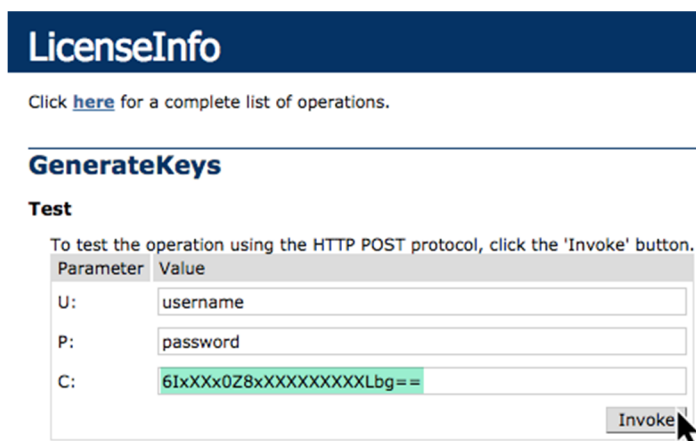
Sending a Content_ID for existing keys will allow you to encrypt content with the same DRM values as other content and have that content share one license. If you call a Content_ID you will get the back all the DRM key information for that Content_ID.

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>5bxxxxxxxx9191fxxe38xxxxxxxx56bf</KeyHEX>
    <KeyID>WVxxxxxxxxliBEXXw+XXXX=</KeyID>
    <KeyIDGUID>5xxxxxx3-36xx-5xx8-8xx1-10xxxxxxxxxb</KeyIDGUID>
    <KeyIDHEX>5xxxxxx36d85xxxxxxxxxxb24xxxb</KeyIDHEX>
    <PSSH>
      EhXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXj3JXXXX==
    </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":"EhXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXj3JXXXX="}]}]}
    </ResponseRaw>
  </WideVine>
  <PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <Key>W5xxxxxxxxZhxTjhxxxxxvw=</Key>
    <KeyHEX>5bxxxxxxxx9191fxxe38xxxxxxxx56bf</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=6IxXXx0Z8xXXXXXXXXXlbg==&Z8xXXXXXXXXXlbg=='
```

Existing DRM Keys returned:

```
<EZDRM xmlns="">
  <WideVine diffgr:id="WideVine1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <ContentID>6IxXXx0Z8xXXXXXXXXXlbg==</ContentID>
    <Key>W5XXXXXXXXZHxTjhXXXXXvw==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXXX56bf</KeyHEX>
    <KeyID>WVXXXXXXXX1iBEXXw+XXXXX==</KeyID>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXXb</KeyIDGUID>
    <KeyIDHEX>5XXXXXX36d85XXXXXXXXXXXXXXb24XXXb</KeyIDHEX>
    <PSSH>EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXXX==</PSSH>
    <ServerURL>https://widevine-dash.ezdrm.com/proxy?pX=XXXXXX</ServerURL>
    <ServerGet>request={"policy": "", "tracks": [ {"type": "SD"}], "content_id": "6IxXXx0Z8xXXXXXXXXXlbg=="}</ServerGet>
    <ResponseRaw>
      {"status":"OK","drm":[{"type":"WIDEVINE","system_id":"edef8ba979d64acea3c827dcd51d21ed"}],"tracks":[{"type":"SD","key_id":" WVXXXXXXXX1iBEXXw+XXXXX==","key":" W5XXXXXXXXZHxTjhXXXXXvw==","pssh":[{"drm_type":"WIDEVINE","data":"EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXXX="}]}]}</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/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**.

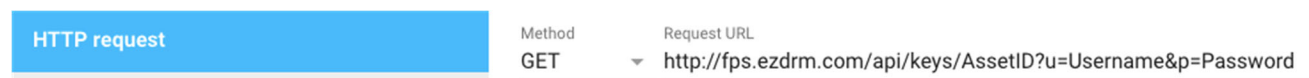


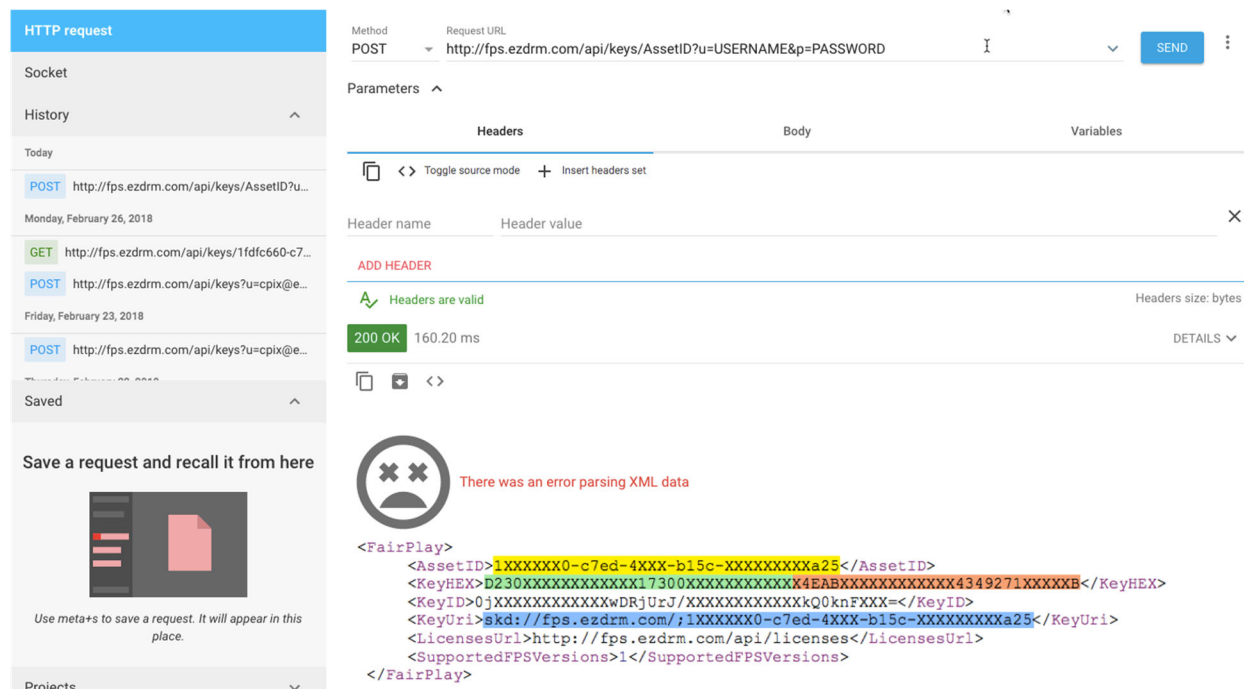
- Enter the Request URL below updated with your AssetID, username, and password:

<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>
  <LicensesUrl>http://fps.ezdrm.com/api/licenses</LicensesUrl>
  <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:

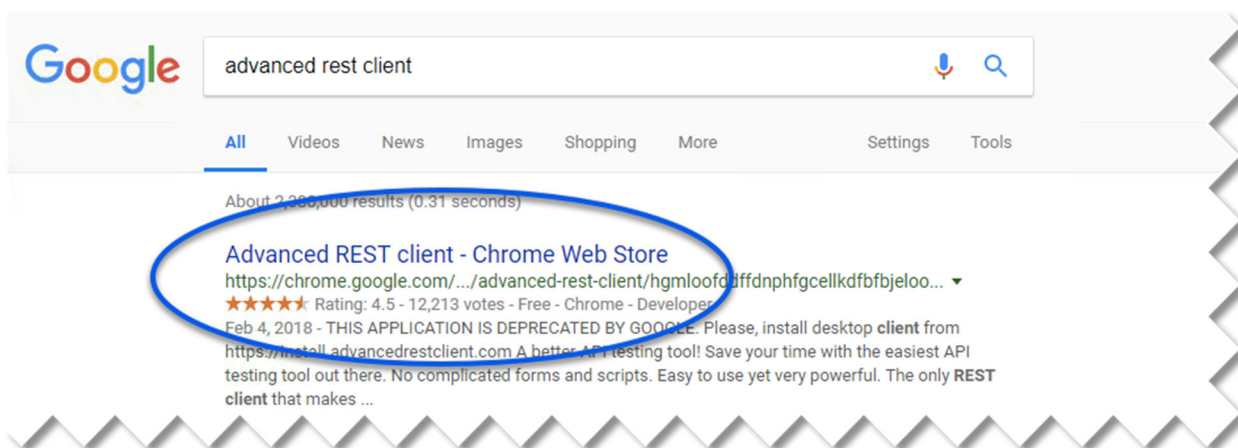
```
curl -X POST 'http://fps.ezdrm.com/api/keys/1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXa25?U=Username&P=Password' -d  
' '
```

Existing DRM Keys returned:

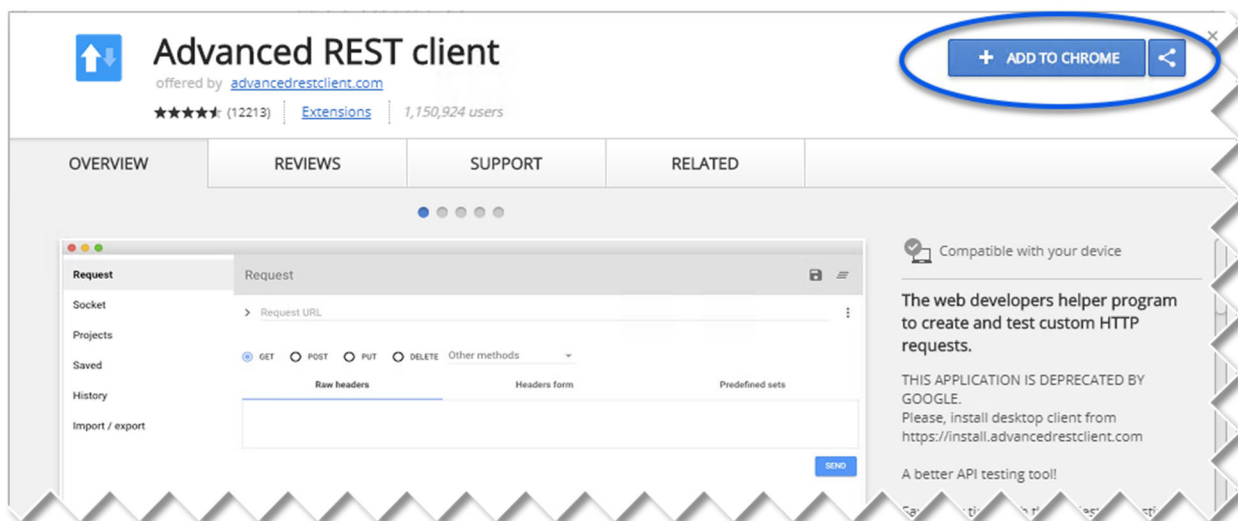
```
<FairPlay>  
  <AssetID>1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXa25</AssetID>  
  <KeyHEX>D230XXXXXXXXXX17300XXXXXXXXXX4EABXXXXXXXXXX4349271XXXXXB</KeyHEX>  
  <KeyID>0jXXXXXXXXXXwDRjUrJ/XXXXXXXXXXkQ0knFXXX=</KeyID>  
  <KeyUri>skd://fps.ezdrm.com/;1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXa25</KeyUri>  
  <LicensesUrl>http://fps.ezdrm.com/api/licenses</LicensesUrl>  
  <SupportedFPSVersions>1</SupportedFPSVersions>  
</FairPlay>
```

Appendix 2 – Installing ARC Plug-in

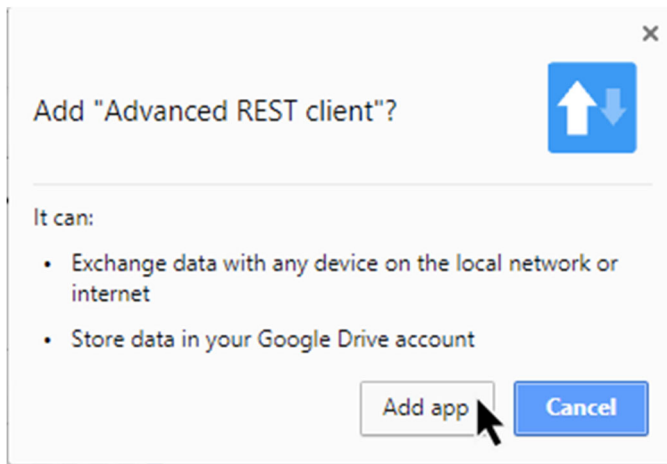
- To install the Advanced REST client (ARC) plug-in for Chrome in order to call the EZDRM Key Servers API, search for “Advanced REST client” using the Google search engine.
- Click the “Advanced REST client – Chrome Web Store” link.



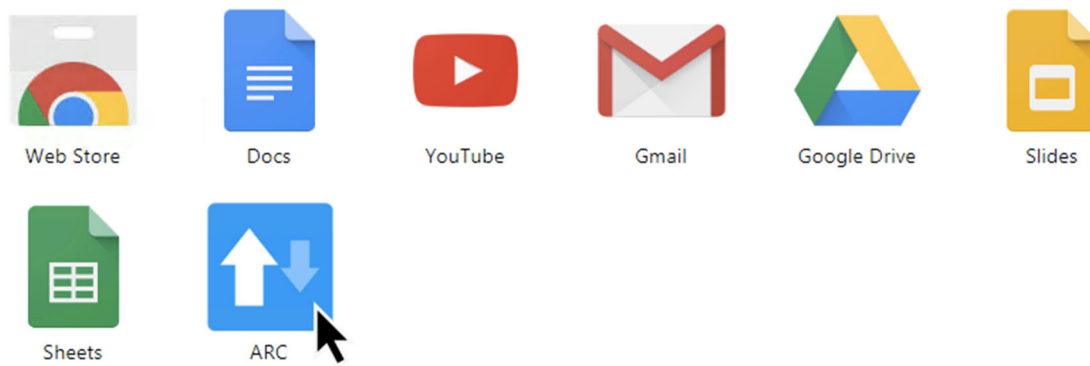
- Click on the **+Add to Chrome** button to install the plug-in.



- 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.