

# **EZDRM Bento4 Configuration**

## **Open Source**

---

## Table of Contents

<b>Prerequisites .....</b>	<b>3</b>
<b>Bento 4 Packager Encryption – Widevine &amp; PlayReady .....</b>	<b>3</b>
<i>Generating Keys.....</i>	<i>3</i>
<i>Widevine and PlayReady Encryption .....</i>	<i>7</i>
<b>Bento 4 Packager Encryption – Apple FairPlay Streaming.....</b>	<b>9</b>
<i>Generating Keys.....</i>	<i>9</i>
<i>Apple FairPlay Encryption .....</i>	<i>12</i>
<b>Appendix 1 – Call for Existing EZDRM Keys .....</b>	<b>14</b>
<i>Widevine and PlayReady.....</i>	<i>14</i>
<i>Apple FairPlay Streaming .....</i>	<i>17</i>
<b>Appendix 2 – Installing ARC Plug-in .....</b>	<b>20</b>

Version 1.0

## Prerequisites

Python 2.7 is recommended.

To download Python 2.7: <https://www.python.org/downloads/>

## Bento 4 Packager Encryption – Widevine & PlayReady

### Generating Keys

Below are the steps to create the DRM Keys for CENC-PlayReady or CENC-Widevine encryption for Bento4 (Open Source).

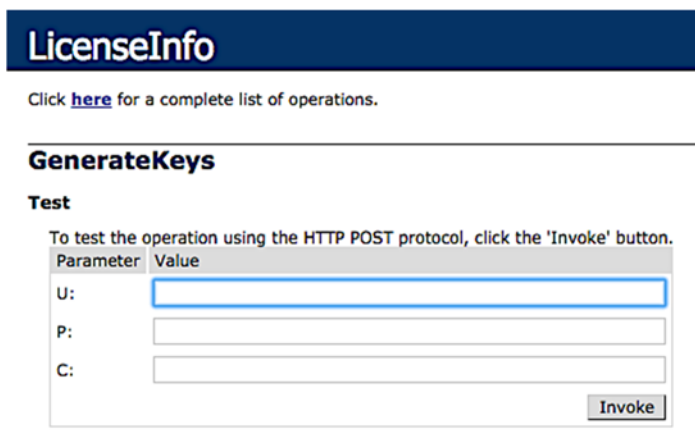
To request the DRM keys from EZDRM to package the media, there are two options, you can call the EZDRM web service in a browser, or you can script this process with curl or other web service calls.

### 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 screenshot shows a web browser window with the title "LicenseInfo". Below the title, there is a link: "Click [here](#) for a complete list of operations." The main content area is titled "GenerateKeys" and contains a "Test" section. The "Test" section includes the instruction: "To test the operation using the HTTP POST protocol, click the 'Invoke' button." Below this instruction is a table with two columns: "Parameter" and "Value". The table has three rows: "U:" with an empty text input field, "P:" with an empty text input field, and "C:" with an empty text input field. At the bottom right of the form is an "Invoke" button.

Parameter	Value
U:	<input type="text"/>
P:	<input type="text"/>
C:	<input type="text"/>

Invoke

The parameters are as follows:

Parameter	Description
<b>u</b>	EZDRM username
<b>p</b>	EZDRM password
<b>c</b>	<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>6Ixxx028xxxxxxxxxLbg=</ContentID>
    <Key>W5XXXXXXXXHxTjhXXXXXvw=</Key>
    <KeyHEX>5bXXXXXXXX9191fXXe38XXXXX56bf</KeyHEX>
    <KeyID>WVXXXXXXXXliBEXXw+XXXX=</KeyID>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
    <KeyIDHEX>5XXXXXX36d85XXXXXXXXXXXXXb24XXXX</KeyIDHEX>
    <PSSH>
      EhXXXXXXXXXXXXXXXX6skGLGXXXXXXXXQ6IebXXX28kSYMXXXXXXXXXXXXXj3JXXXX==
    </PSSH>
    <ServerURL>https://widevine-dash.ezdrm.com/proxy?pX=XXXXXX</ServerURL>
    <ServerGet>
      request={"policy": "", "tracks": [{"type": "SD"}], "content_id": "6Ixxx028xxxxxxxxxLbg="}
    </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": "EhXXXXXXXXXXXXXXXX6skGLGXXXXXXXXQ6IebXXX28kSYMXXXXXXXXXXXXXj3JXXXX="}]}]}
    </ResponseRaw>
  </WideVine>
  <PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <Key>W5XXXXXXXXHxTjhXXXXXvw=</Key>
    <KeyHEX>5bXXXXXXXX9191fXXe38XXXXX56bf</KeyHEX>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
    <LAURL>
      https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX
    </LAURL>
    <Checksum>1Xq+XXXXX0=</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
<b>U</b>	EZDRM username
<b>P</b>	EZDRM password
<b>C</b>	<u>Content ID</u> **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>6lxx0Z8xxxxxxxxLbg==</ContentID>
    <Key>W5xxxxxxZHxTjhxxxxvw==</Key>
    <KeyHEX>5bxxxxxxxx9191fXxe38xxxx56bf</KeyHEX>
    <KeyID>WVxxxxxxliBEXXw+XXXX==</KeyID>
    <KeyIDGUID>5xxxxxx3-36XX-5XX8-8XX1-10xxxxxxxxb</KeyIDGUID>
    <KeyIDHEX>5xxxxxx36d85xxxxxxxxxxxxb24xxx</KeyIDHEX>
    <PSSH>Ehxxxxxxxxxxxxxxxx6skGLGxxxxxxxxO6lebxxxZ8kSYMxxxxxxxxxxxxxxxxj3jxxxx==</PSSH>
    <ServerURL>https://widevine-dash.ezdrm.com/proxy?pX=XXXX</ServerURL>
    <ServerGet>request={"policy": "", "tracks": [ {"type": "SD"}, {"content_id": "6lxx0Z8xxxxxxxxLbg="}]}</ServerG
  et>
  <ResponseRaw>
{"status":"OK","drm":{"type":"WIDEVINE","system_id":"edef8ba979d64acea3c827dcd51d21ed"},"tracks":[{"type":"SD","key_id":
:" WVxxxxxxliBEXXw+XXXX==","key": " W5xxxxxxZHxTjhxxxxvw==","pssh":{"drm_type":"WIDEVINE","data":"Ehxxxxxxxx
xxxxxxxx6skGLGxxxxxxxxQ6lebxxxZ8kSYMxxxxxxxxxxxxxxxxj3jxxxx="}]}]}</ResponseRaw>
```

```
</WideVine>
<PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
<Key>W5XXXXXXXXZHxTjhXXXXvw==</Key>
<KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXX56bf</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>
```

## Widevine and PlayReady Encryption

Once you have the DRM values to encrypt the content, you can add them to the Bento4 open source packager for CENC-PlayReady or CENC-Widevine encryption.

1. First you need to fragment the MP4. Open command prompt and navigate to the MP4 file.

For this example, we are fragmenting the MP4 file named “bigbuckbunny\_450.mp4”. The command would be:

```
mp4fragment.exe inputname.mp4 inputname-frag.mp4
```

Sample command:

```
c:\Users\User\Downloads\Bento4-SDK-1-5-1-622.x86-microsoft-win32-vs2010\bin>  
mp4fragment.exe bigbuckbunny_450.mp4 bigbuckbunny_450-frag.mp4
```

2. This will result in a file called **bigbuckbunny\_450-frag.mp4** in the **bin** folder. Move this file into the **utils** folder.
3. Next you can package the media into the DASH format. Use the following syntax in Command prompt from **utils** folder.

```
mp4-dash.py --widevine-header=#PSSH DATA --playready-header="LA_URL:LAURL" --encryption-key="KeyIDHex:KeyHex" video-source.mp4
```

**Note: Be sure to include the # before the PSSH Data value.**

The description for the lines of syntax include:

- Use the command line option **--widevine-header** to specify the Widevine **PSSH** Data value with a # in front of the value.
- Use the command line option **--playready-header** to specify the EZDRM PlayReady **LAURL** value with “LA\_URL:” in front of the LAURL value.

- Use the command line option **--encryption-key** to specify the EZDRM Key. Bento4 refers to encryption-key as KeyID:Key. The KeyID = EZDRM **KeyIDHex** and the Key = EZDRM **KeyHex**.
- **video-source.mp4** is the source file.

Sample command from **utils** folder:

```
mp4-dash.py --widevine-header=#NOTREALDATAEhXXXXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXO6lebXXXZ8kSYMXXXXXXXXXXXXXXXXXX  
XXi3lXXX== --playready-header="LA URL:https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX" --encryption-key="5XX  
XXXX36d85XXXXXXXXXXXXXXXXb24XXb:5bXXXXXXXXXX9191fXXe38XXXXXXXX56bf" bigbuckbunny_450-frag.mp4
```

You can also script this process with curl or other web service calls.

For updated details of the syntax please refer to:

<https://www.bento4.com/developers/dash/encryption-and-drm/>



## Bento 4 Packager Encryption – Apple FairPlay Streaming

EZDRM Apple FairPlay DRM is a hosted Apple FairPlay Streaming (DRM). This enables a content owner to encrypt the media with Apple FPS DRM keys and deliver content Apple devices with native support MAC Safari browser via HTML 5 player or iOS via native App or Safari 11.3.

The packaging process encrypts the media. This is accomplished via a secure web call to the EZDRM Key Servers API. The Key Server API will return an XML response with the DRM key structure.

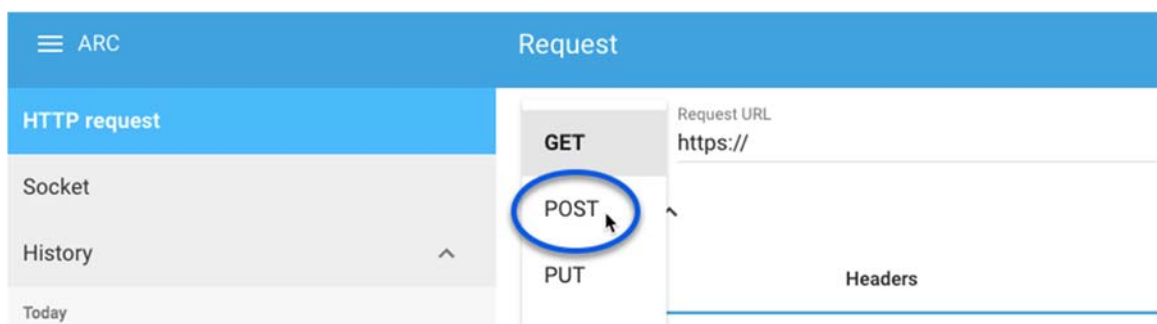
### Generating Keys

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
<b>u</b>	EZDRM username
<b>p</b>	EZDRM password

```

<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>
  <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>0jXXXXXXXXXXXXwDRjUrJ/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
<b>U</b>	EZDRM username
<b>P</b>	EZDRM password

The following is returned:

```
<FairPlay>
  <AssetID>1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXXa25</AssetID>
  <KeyHEX>D230XXXXXXXXXXXX17300XXXXXXXXXXXX4EABXXXXXXXXXXXX4349271XXXXXB</KeyHEX>
  <KeyID>0jXXXXXXXXXXXXwDRjUrJ/XXXXXXXXXXXXkQ0knFXXX=</KeyID>
  <KeyUri>skd://fps.ezdrm.com/:1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXXa25</KeyUri>
  <LicensesUrl>http://fps.ezdrm.com/api/licenses</LicensesUrl>
  <SupportedFPSVersions>1</SupportedFPSVersions>
</FairPlay>
```

## Apple FairPlay Encryption

Once you have the DRM values to encrypt the content, you can add them to the Bento4 open source packager for Apple FairPlay encryption.

Use the following syntax in Command prompt from the **bin** folder:

```
mp4hls --hls-version 7 --output-single-file --encryption-mode=SAMPLE-AES --encryption-key=KeyHEX --encryption-iv-mode=fps  
--encryption-key-format=com.apple.streamingkeydelivery --encryption-key-uri=KeyUri --encryption-key-format-versions=1 vid  
eo-source.mp4
```

Here are the descriptions of the keys returned from the EZDRM Key servers API:

- **KeyHEX:** the DRM Encryption Key. With FairPlay, the key and IV are delivered together in the **KeyHEX** value. The first 32 characters are used in the key file for "key" and the last 32 characters are used for "iv".
- **KeyUri:** the License URL for encryption.

The description for the lines of syntax include:

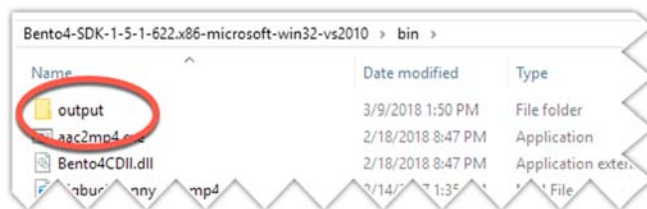
- Use the command line option **--encryption-mode=SAMPLE-AES** to select the SAMPLE-AES encryption mode.
- Use the command line option **--encryption-key** to specify the EZDRM **KeyHEX** (the key and iv as returned above).
- Use the command line option **--encryption-iv-mode=fps** to specify that the IV is delivered with the key.
- Use the command line option **--encryption-key-format=com.apple.streamingkeydelivery** to specify that FairPlay will be used to deliver the key and IV.
- Use the command line option **--encryption-key-uri** to specify the **KeyUri** for the EZDRM server.

- **video-source.mp4** should be replaced with the name of the source file.

Sample Command from the bin folder:

```
mp4hls --hls-version 7 --output-single-file --encryption-mode=SAMPLE-AES --encryption-key=D230XXXXXXXXXXXX17300XXXXXX
XXXXXXXX4EABXXXXXXXXXXXX4349271XXXXXB --encryption-iv-mode=fps --encryption-key-format=com.apple.streamingkeydelive
ry --encryption-key-uri=skd://fps.ezdrm.com/:1XXXXX0-c7ed-4XXX-b15c-XXXXXXXXXa25 --encryption-key-format-versions=1 bj
gbuckbunny_450.mp4
```

You will now have an **output** folder in the **bin** file with the encrypted files.



You can also script this process with curl or other web service calls.

For updated details on the Bento4 Apple FairPlay syntax please refer to this link:  
<https://www.bento4.com/developers/hls/>

## 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

1. Call the EZDRM web service in a browser:

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

2. 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
<b>u</b>	EZDRM username
<b>p</b>	EZDRM password
<b>c</b>	<u>Content_ID</u>

3. 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>6Ixxx028xxxxxxxxxLbg==</ContentID>
    <Key>W5XXXXXXXX2HxTjhXXXXVw==</Key>
    <KeyHEX>5bXXXXXXXXX9191fXx38XXXXXXXX56bf </KeyHEX>
    <KeyID>WVXXXXXXXXliBEXw+XXXX==</KeyID>
    <KeyIDGUID>5XXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
    <KeyIDHEX>5XXXXX36d85XXXXXXXXXXXXb24XXb</KeyIDHEX>
    <PSSH>
      EhXXXXXXXXXXXXXXXX6skGLGXXXXXXXXQ6IebXXX28kSYMXXXXXXXXXXXXXXXXj3JXXXX==
    </PSSH>
    <ServerURL>https://widevine-dash.ezdrm.com/proxy?pX=XXXXXX</ServerURL>
    <ServerGet>
      request={"policy": "", "tracks": [{"type": "SD"}], "content_id": "6Ixxx028xxxxxxxxxLbg==" }
    </ServerGet>
    <ResponseRaw>
      {"status": "OK", "drm": [{"type": "WIDEVINE", "system_id": "edef8ba979d64acea3c827dcd51d21ed"}], "tracks": [{"type": "SD", "key_id": "WVXXXXXXXXliBEXw+XXXX==", "key": "WVXXXXXXXXliBEXw+XXXX==", "pssh": [{"drm_type": "WIDEVINE", "data": "EhXXXXXXXXXXXXXXXX6skGLGXXXXXXXXQ6IebXXX28kSYMXXXXXXXXXXXXXXXXj3JXXXX=="}]}]}
    </ResponseRaw>
  </WideVine>
  <PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <Key>W5XXXXXXXX2HxTjhXXXXVw==</Key>
    <KeyHEX>5bXXXXXXXXX9191fXx38XXXXXXXX56bf </KeyHEX>
    <KeyIDGUID>5XXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
    <LAURL>
      https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX
    </LAURL>
    <Checksum>1Xq+XXXXX0=</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.aspx/GenerateKeys?U=EZDRM_USERNAME&P=EZDRM_PASSWORD&C=CONTENT_ID'
```

The parameters are as follows:

Parameter	Description
<b>U</b>	EZDRM username
<b>P</b>	EZDRM password
<b>C</b>	<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.aspx/GenerateKeys?U=EZDRM_USERNAME&P=EZDRM_PASSWORD&C=6lxXXx0Z8xXXXXXXXXXlbg=='
```

## Existing DRM Keys returned:

```
<EZDRM xmlns="">
  <WideVine diffgr:id="WideVine1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <ContentID>6lxXXx0Z8xXXXXXXXXXlbg==</ContentID>
    <Key>W5XXXXXXXXZHxTjhXXXXvw==</Key>
    <KeyHEX>5bXXXXXXXXX9191fXXe38XXXXX56bf </KeyHEX>
    <KeyID>WVXXXXXXXXliBEXXw+XXXXX==</KeyID>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
    <KeyIDHEX>5XXXXXX36d85XXXXXXXXXXXXXb24XXb</KeyIDHEX>
    <PSSH>EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6lebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3jXXX==</PSSH>
    <ServerURL>https://widevine-dash.ezdrm.com/proxy?pX=XXXXX</ServerURL>
    <ServerGet>request={"policy": "", "tracks": [ {"type": "SD"}, {"content_id": "6lxXXx0Z8xXXXXXXXXXlbg=="}]}</ServerGet>
  et>
  <ResponseRaw>
{"status":"OK","drm":[{"type":"WIDEVINE","system_id":"edef8ba979d64acea3c827dcd51d21ed"},"tracks":[{"type":"SD","key_id":"WVXXXXXXXXliBEXXw+XXXXX==","key":"W5XXXXXXXXZHxTjhXXXXvw==","pssh":{"drm_type":"WIDEVINE","data":"EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXXQ6lebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3jXXX=="}]}]}</ResponseRaw>
  </WideVine>
  <PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <Key>W5XXXXXXXXZHxTjhXXXXvw==</Key>
    <KeyHEX>5bXXXXXXXXX9191fXXe38XXXXX56bf</KeyHEX>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
    <LAURL>https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXX</LAURL>
    <Checksum>1Xq+XXXXX0=</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

1. 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.
2. Change the Method dropdown to **GET**.



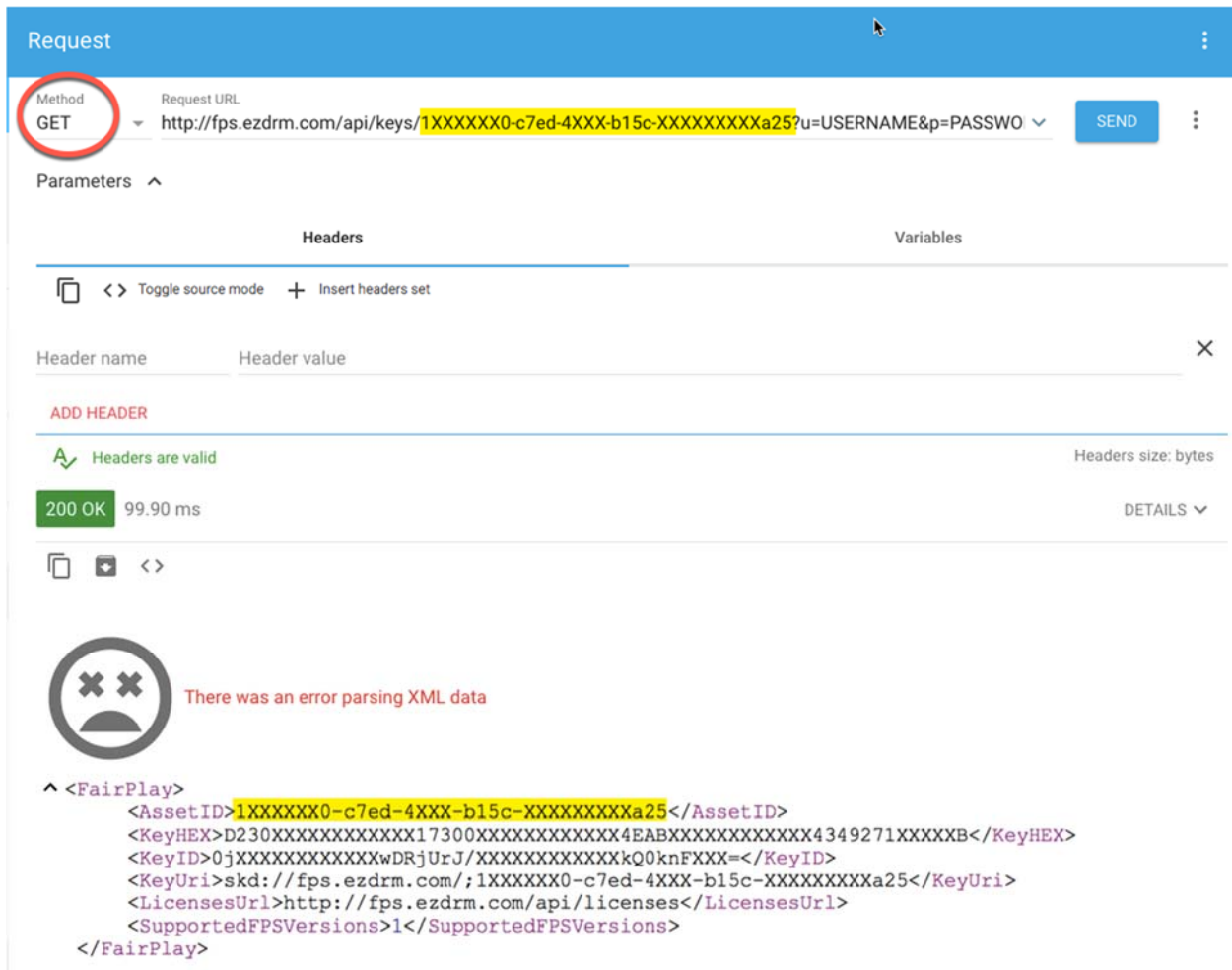
3. Enter the Request URL below updated with your AssetID, username, and password:

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

The parameters are as follows:

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





Request

Method: GET Request URL: http://fps.ezdrm.com/api/keys/1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXa25?u=USERNAME&p=PASSWO

Parameters: Headers Variables

200 OK 99.90 ms

There was an error parsing XML data

```

^ <FairPlay>
  <AssetID>1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXa25</AssetID>
  <KeyHEX>D230XXXXXXXXXXXX17300XXXXXXXXXXXX4EABXXXXXXXXXXXX4349271XXXXXB</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>

```

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
<b>AssetID</b>	Existing FairPlay AssetID
<b>U</b>	EZDRM username
<b>P</b>	EZDRM password

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

```
<FairPlay>
  <AssetID>1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXa25</AssetID>
  <KeyHEX>D230XXXXXXXXXXXX17300XXXXXXXXXXXX4EABXXXXXXXXXXXX4349271XXXXXB</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>
```

Sample curl script with AssetID:

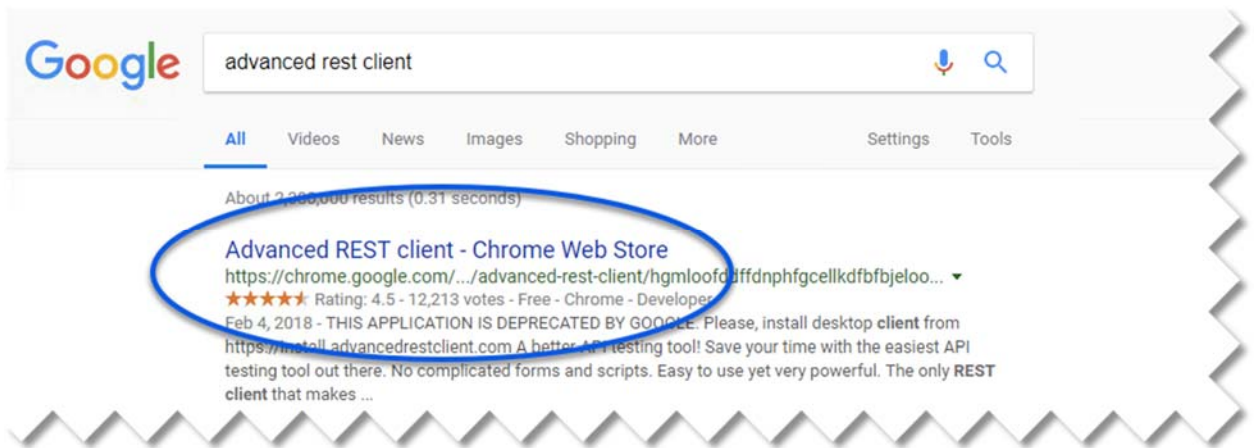
```
curl -X POST 'http://fps.ezdrm.com/api/keys/1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXa25?U=Username&P=Password' -d ''
```

Existing DRM Keys returned:

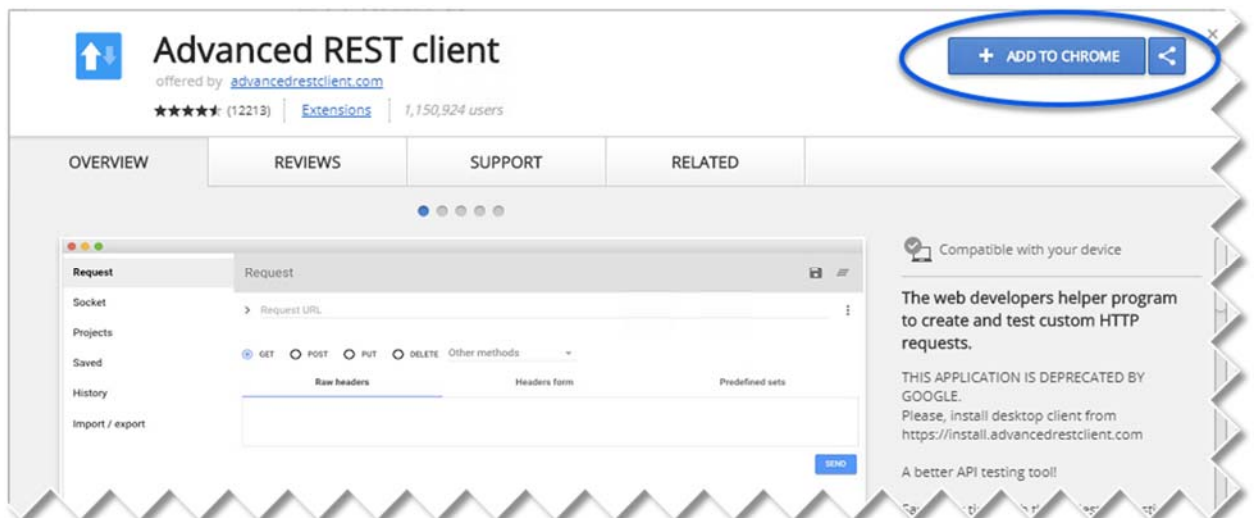
```
<FairPlay>
  <AssetID>1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXa25</AssetID>
  <KeyHEX>D230XXXXXXXXXXXX17300XXXXXXXXXXXX4EABXXXXXXXXXXXX4349271XXXXXB</KeyHEX>
  <KeyID>0jXXXXXXXXXXXXwDRjUrJ/XXXXXXXXXXXXkQ0knFXXX=</KeyID>
  <KeyUri>skd://fps.ezdrm.com/;1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXa25</KeyUri>
  <LicensesUrl>http://fps.ezdrm.com/api/licenses</LicensesUrl>
  <SupportedFPSVersions>1</SupportedFPSVersions>
</FairPlay>
```

## Appendix 2 – Installing ARC Plug-in

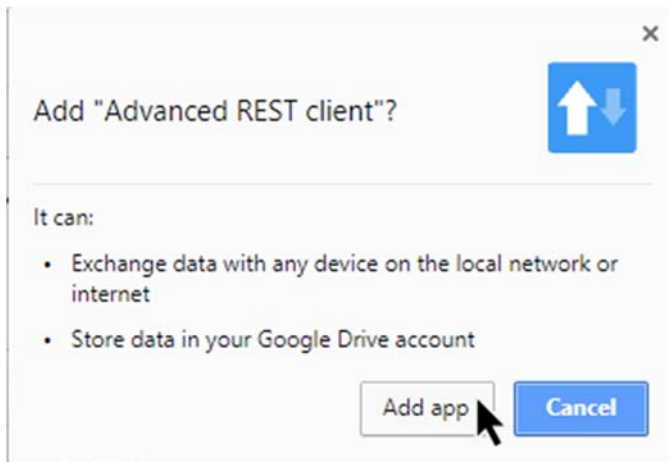
- a. To install the Advanced REST client (ARC) plug-in for Chrome in order to call the EZDRM Key Servers API, search for “Advanced REST client” using the Google search engine.
- b. Click the “Advanced REST client – Chrome Web Store” link.



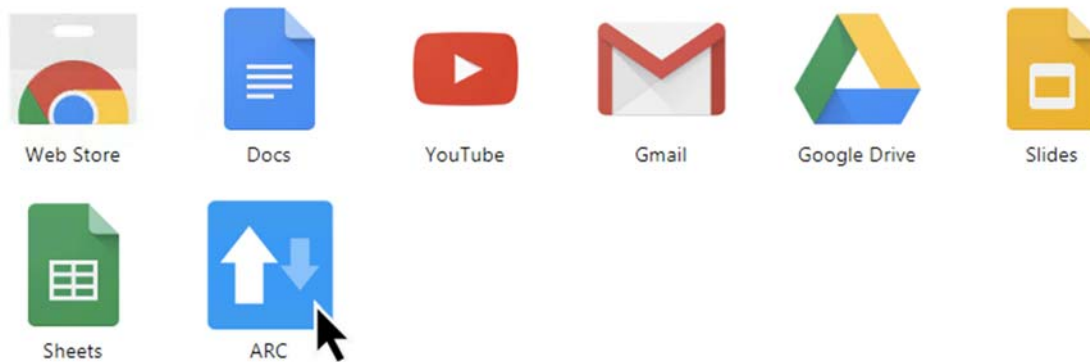
- c. Click on the **+Add to Chrome** button to install the plug-in.



- 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.