EZDRM Coconut API Encoding

Table of Contents

Generating DRM Keys **4**

Widevine and PlayReady *4*

Apple FairPlay Streaming *7*

Coconut DRM Encryption - Widevine & PlayReady **10**

Key Value Definitions *10*

Widevine only *11*

PlayReady only *11*

Coconut DRM Encryption - Apple Fairplay **12**

Key Value Definitions *12*

Coconut Jobs **14**

Appendix 1 – Call for Existing EZDRM Keys **15**

Widevine and PlayReady *15*

Apple FairPlay Streaming *18*

Appendix 2 – Installing ARC Plug-in **21**

Additional Information **23**

Version 1.0

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.

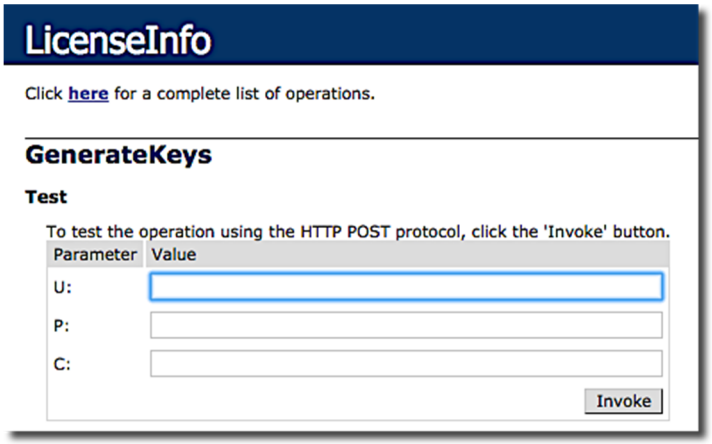
To request the DRM keys from EZDRM to package the media, there are two options, you can call the EZDRM web service in a browser, or you can script this process with curl or other web service calls.

Option 1: Request DRM keys using EZDRM Web Service

1. Call the EZDRM web service in a browser:

   https://wvm.ezdrm.com/ws/LicenseInfo.asmx?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	Content_ID **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>W5XXXXXXXZHxTjhXXXXXvw==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXXX56bf</KeyHEX>
    <KeyID>WVXXXXXXXliBEXXw+XXXXX==</KeyID>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
    <KeyIDHEX>5XXXXXX36d85XXXXXXXXXXXXXXb24XXXb</KeyIDHEX>
   ▼<PSSH>
      EhXXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXXXj3JXXXX==
    </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":"WVXXXXXXXliBEXXw+XXXXX==","key":"WVXXXXXXXliBEXXw+XXXXX==","pssh":
      [{"drm_type":"WIDEVINE","data":"EhXXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXXXj3JXXXX=="}]}]}
    </ResponseRaw>
  </WideVine>
 ▼<PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <Key>W5XXXXXXXZHxTjhXXXXXvw==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXXX56bf</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 'https://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>6IxXXx0Z8xXXXXXXXXXLbg==</ContentID>
          <Key>W5XXXXXXXZHxTjhXXXXXvw==</Key>
          <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXXX56bf</KeyHEX>
          <KeyID>WVXXXXXXXliBEXXw+XXXXX==</KeyID>
          <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
          <KeyIDHEX>5XXXXXX36d85XXXXXXXXXXXXXb24XXXb</KeyIDHEX>
          <PSSH>EhXXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXXX==</PSSH>
          <ServerURL>https://widevine-dash.ezdrm.com/proxy?pX=XXXXXX</ServerURL>
          <ServerGet>request={"policy": "", "tracks": [ {"type": "SD"}], "content_id": "6IxXXx0Z8xXXXXXXX
XXLbg=="}</ServerGet>
          <ResponseRaw>
{"status":"OK","drm":[{"type":"WIDEVINE","system_id":"edef8ba979d64acea3c827dcd51d21ed"}],"tracks":[{"type
":"SD","key_id":" WVXXXXXXXliBEXXw+XXXXX==","key":" W5XXXXXXXZHxTjhXXXXXvw==","pssh":[{"drm_type":"WIDEVIN
E","data":"EhXXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXXX=="}]}]}</ResponseRaw>
          </WideVine>
          <PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
          <Key>W5XXXXXXXZHxTjhXXXXXvw==</Key>
          <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXXX56bf</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>
```

Apple FairPlay Streaming

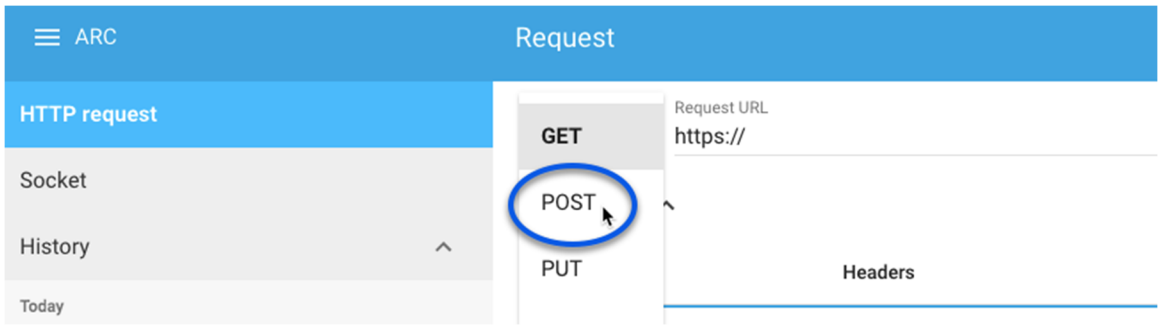
Below are the steps to create the Key Files for Apple FairPlay Streaming.

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**.

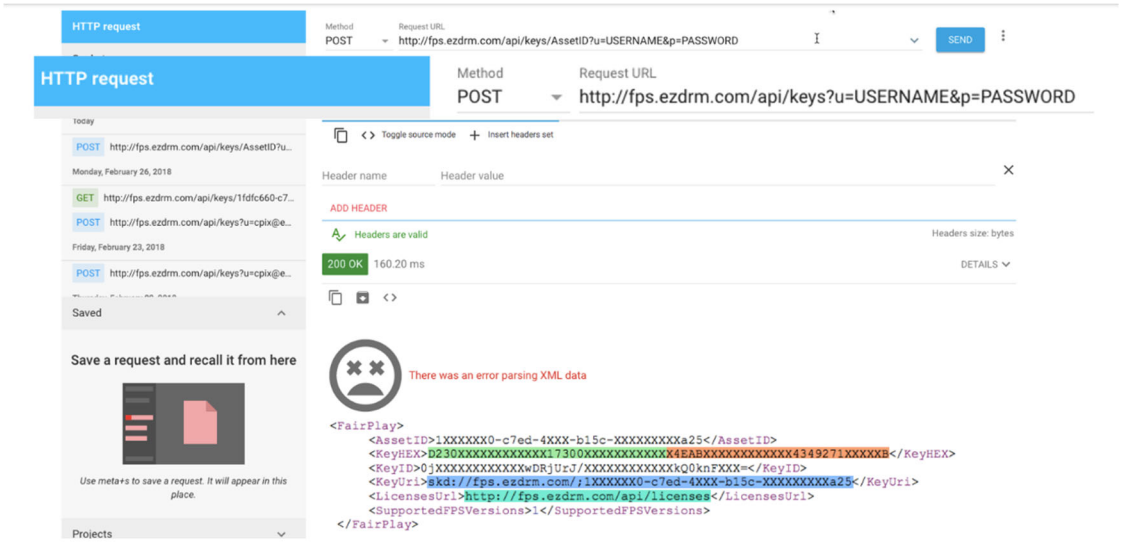


3. Enter the Request URL below updated with your username and password:

   https://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-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>
```

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 'https://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>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>
```

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.

Coconut DRM Encryption - Widevine & PlayReady

The following example is for Widevine and PlayReady, for more details visit https://docs.coconut.co/jobs/outputs-httpstream#drm-and-encryption.

```
    "outputs": {
        "httpstream": {
            "dash": {
                "path": "/dash",
                "widevine_pssh": "Widevine PSSH",
                "playready_laurl": "Playready LAURL",
                "encryption_key": "KeyIDHex:KeyHex"
            }
```

Key Value Definitions

- **widevine_pssh:** This is the value for the Widevine PSSH box, the EZDRM **PSSH** value.
- **playready_laurl**: This is the URL to the PlayReady license server, the EZDRM PlayReady **LAURL** value.
- **encryption_key** = This is the common content encryption key, the EZDRM **KeyIDHEX**:**KeyHEX** values, dashes removed, separated by a colon (:).

The following is a sample with key values:

```
{
    "input": {
        "url": "https://s3.amazonaws.com/yoursample.mp4"
    },
    "storage": {
        "service": "coconut"
    },
    "notification": {
        "url": "https://app.coconut.co/notifications/http/ba1XXXX2"
    },
    "outputs": {
        "httpstream": {
            "dash": {
                "path": "/dash",
```

```
                "widevine_pssh": "CAESEKap1G1+5XXXXxxxxXXXXXxxxxXXXmlkb25lIhA6McJnOOwoQKE1mMAjRKu8KgJTRA==
",
                "playready_laurl": "https://playready.ezdrm.com/cency/preauth.aspx?pX=E0XXXX",
                "encryption_key": "5XXXXXX36d85XXXXXXXXXXXXXb24XXXb:5bXXXXXXXXXX9191fXXe38XXXXXX56bf"
            }
        }
    }
}
```

Widevine only

For only Widevine send the following output DRM keys:

```
    "outputs": {
        "httpstream": {
            "dash": {
                "path": "/dash",
                "widevine_pssh": "Widevine PSSH",
                "encryption_key": "KeyIDHex:KeyHex"
            }
```

PlayReady only

For only PlayReady send the following output DRM keys:

```
    "outputs": {
        "httpstream": {
            "dash": {
                "path": "/dash",
                "playready_laurl": "Playready LAURL",
                "encryption_key": "KeyIDHex:KeyHex"
            }
```

Coconut DRM Encryption - Apple Fairplay

The following example is for Widevine and Playready, for more details visit https://docs.coconut.co/jobs/outputs-httpstream#drm-and-encryption.

```
{
  "outputs": {
    "httpstream": {
      "hls": {
        "path": "/hls",
        "encryption_mode": "SAMPLE-AES",
        "encryption_key": "KeyHEX (keyiv)",
        "encryption_key_uri": "KeyUri"
      }
    }
  }
}
```

Key Value Definitions

Here are the descriptions of the items returned:

- **encryption_mode:** Must be **Sample-AES**

- **encryption_key:** the DRM Encryption Key. With FairPlay, the **key** and **IV** are delivered together. The first 32 characters from the **KeyHEX** value are used in the key file for **key** and the last 32 characters are used for **iv**, **separated by a colon.**

  Example:
  D230XXXXXXXXXXXX17300XXXXXXXXXXX:X4EABXXXXXXXXXXXX4349271XXXXXB

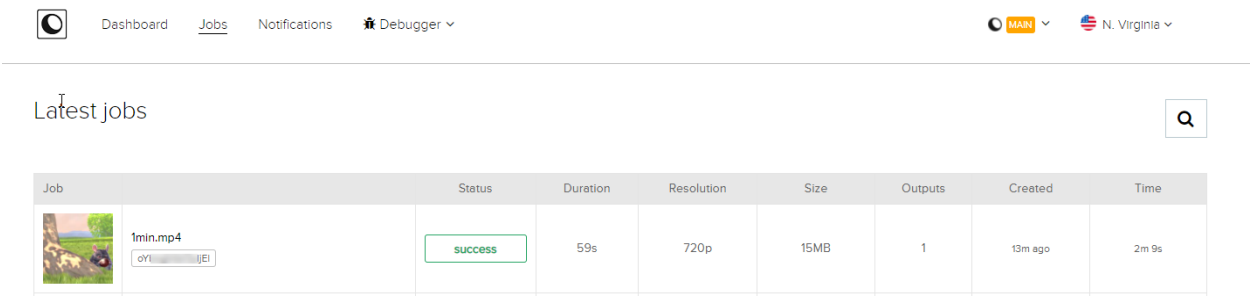
- **encryption_key_uri**: the **KeyUri** for encryption.

The following is a sample with key values:

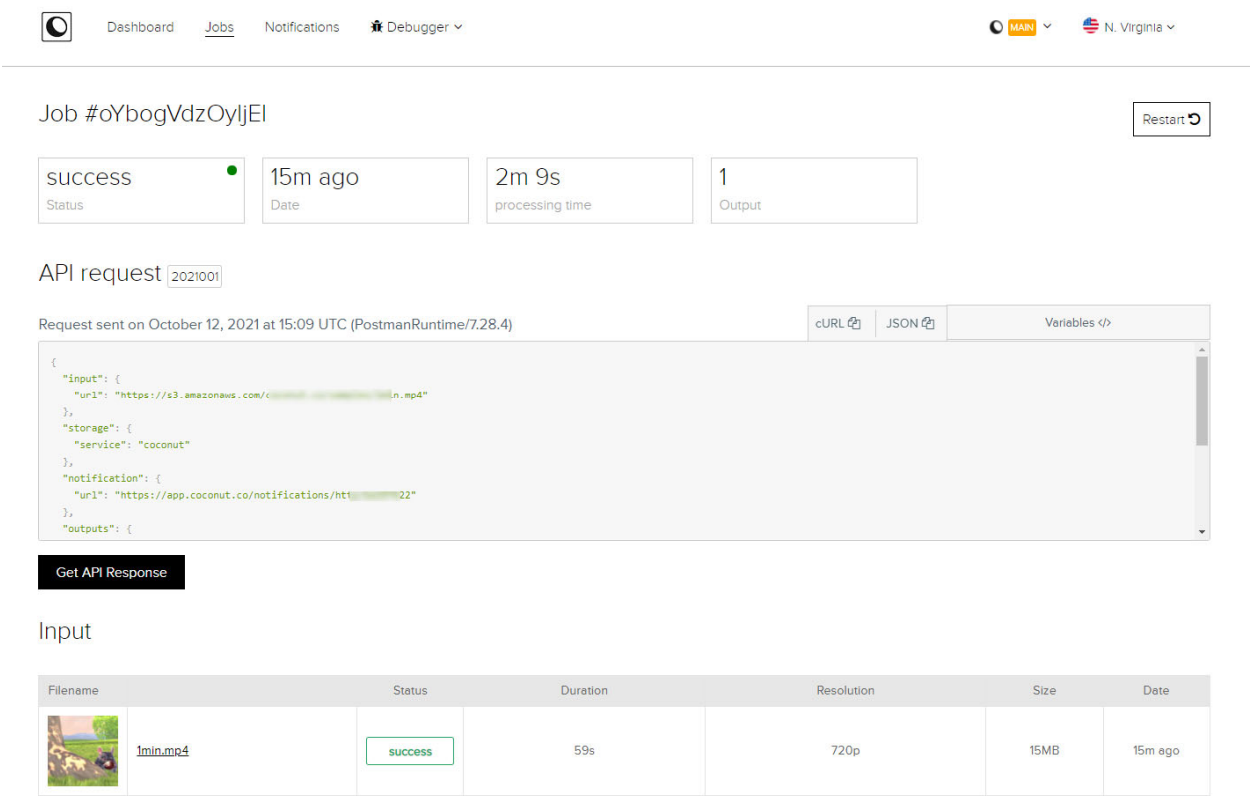
```
{
    "input": {
        "url": "https://s3.amazonaws.com/yoursample.mp4"
    },
    "storage": {
        "service": "coconut"
    },
    "notification": {
        "url": "https://app.coconut.co/notifications/http/baXXXX22"
    },
    "outputs": {
    "httpstream": {
      "hls": {
        "path": "/hls",
        "encryption_mode": "SAMPLE-AES",
        "encryption_key": "D230XXXXXXXXXXXX17300XXXXXXXXXXX:X4EABXXXXXXXXXXXX4349271XXXXXB",
        "encryption_key_uri": "skd://fps.ezdrm.com/;1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXa25"
      }
    }
  }
}
```

Coconut Jobs

See completed Jobs at https://app.coconut.co/jobs.



The Jobs dashboard will show details for the job.



Appendix 1 – Call for Existing EZDRM Keys

Widevine and PlayReady

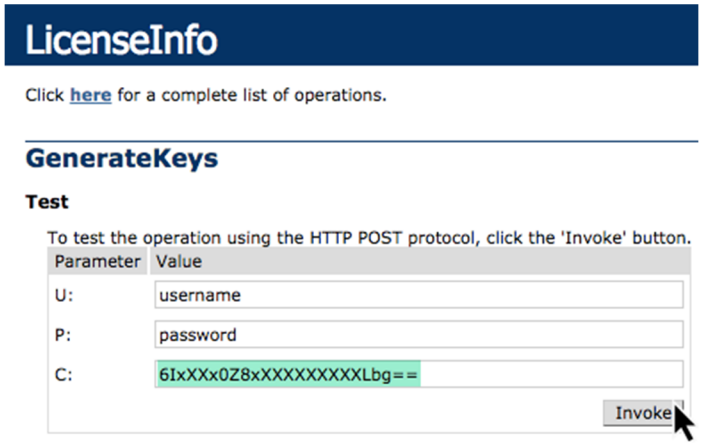
Sending a Content_ID for existing keys will allow you to encrypt content with the same DRM values as other content and have that content share one license. If you call a Content_ID you will get the back all the DRM key information for that Content_ID.

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.asmx?op=GenerateKeys

b. Return Key values by entering the parameter values including the existing Content_ID and click "Invoke".



The parameters are as follows:

Parameter	Description
u	EZDRM username
p	EZDRM password
c	Content_ID

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>6IxXXx0Z8xXXXXXXXXXLbg==</ContentID>
    <Key>W5XXXXXXXZHxTjhXXXXXvw==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXXX56bf</KeyHEX>
    <KeyID>WVXXXXXXXliBEXXw+XXXXX==</KeyID>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
    <KeyIDHEX>5XXXXXX36d85XXXXXXXXXXXXXb24XXXb</KeyIDHEX>
   ▼<PSSH>
      EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXXX==
    </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":"WVXXXXXXXliBEXXw+XXXXX==","key":"WVXXXXXXXliBEXXw+XXXXX==","pssh":
      [{"drm_type":"WIDEVINE","data":"EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXXX=="}]}]}
    </ResponseRaw>
  </WideVine>
 ▼<PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
    <Key>W5XXXXXXXZHxTjhXXXXXvw==</Key>
    <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXXX56bf</KeyHEX>
    <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
   ▼<LAURL>
      https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX
    </LAURL>
    <Checksum>1Xg+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 ‘https://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	Content_ID

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

Sample script with Content_ID:

```
curl -v ‘https://wvm.ezdrm.com/ws/LicenseInfo.asmx/GenerateKeys?U=EZDRM USERNAME&P=EZDRM PASSWORD&C=6IxXXx0Z8xXXXXXXXXXLbg==”’
```

Existing DRM Keys returned:


```
<EZDRM xmlns="">
        <WideVine diffgr:id="WideVine1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
          <ContentID>6IxXXx0Z8xXXXXXXXXXLbg==</ContentID>
          <Key>W5XXXXXXXZHxTjhXXXXXvw==</Key>
          <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXXX56bf </KeyHEX>
          <KeyID>WVXXXXXXXliBEXXw+XXXXX==</KeyID>
          <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb</KeyIDGUID>
          <KeyIDHEX>5XXXXXX36d85XXXXXXXXXXXXXXb24XXXb</KeyIDHEX>
          <PSSH>EhXXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXXX==</PSSH>
          <ServerURL>https://widevine-dash.ezdrm.com/proxy?pX=XXXXXX</ServerURL>
          <ServerGet>request={"policy": "", "tracks": [ {"type": "SD"}], "content_id": "6IxXXx0Z8xXXXXXXX
XXLbg=="}</ServerGet>
          <ResponseRaw>
{"status":"OK","drm":[{"type":"WIDEVINE","system_id":"edef8ba979d64acea3c827dcd51d21ed"}],"tracks":[{"type
":"SD","key_id":" WVXXXXXXXliBEXXw+XXXXX==","key":" W5XXXXXXXZHxTjhXXXXXvw==","pssh":[{"drm_type":"WIDEVIN
E","data":"EhXXXXXXXXXXXXXXXXXX6skGLGXXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXXX=="}]}]}</ResponseRaw>
          </WideVine>
          <PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
          <Key>W5XXXXXXXZHxTjhXXXXXvw==</Key>
          <KeyHEX>5bXXXXXXXXXX9191fXXe38XXXXXX56bf</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>
```

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

a. 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.

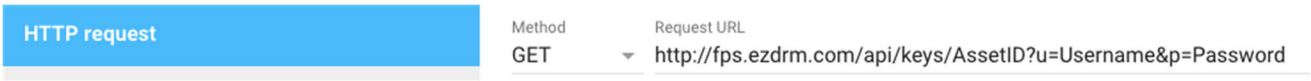
b. Change the Method dropdown to **GET**.

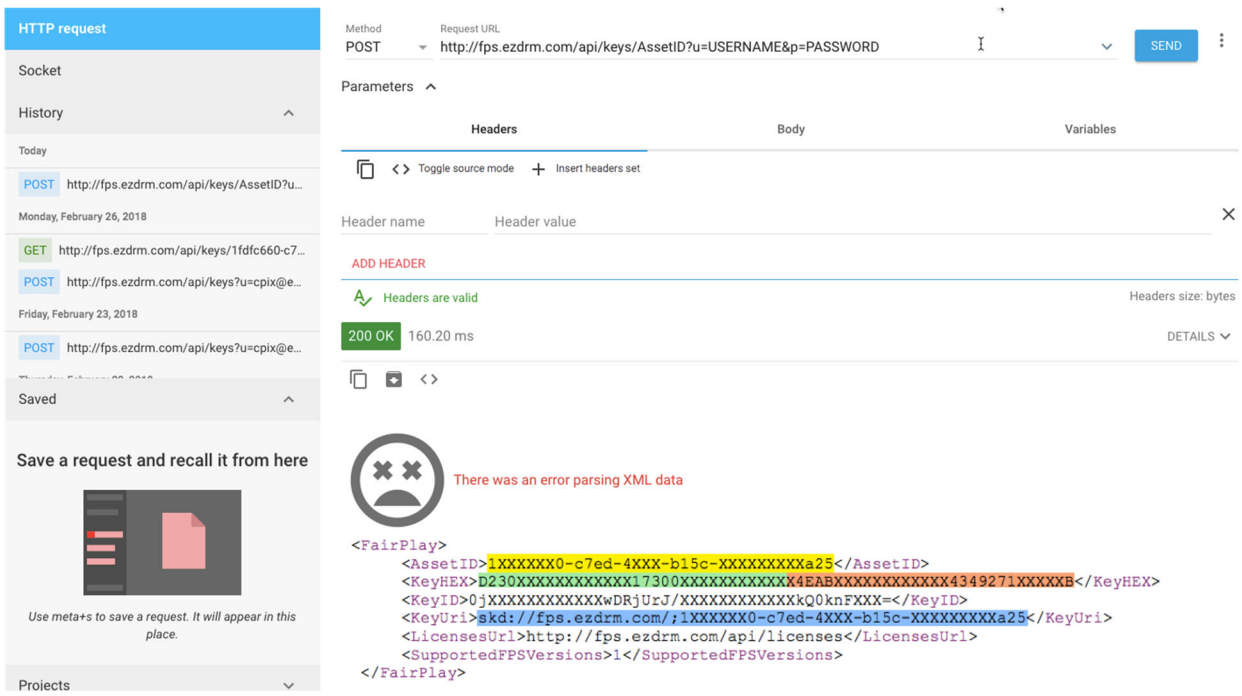
c. Enter the Request URL below updated with your AssetID, username, and password:

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

The parameters are as follows:

Parameter	Description
AssetID	Existing FairPlay AssetID
u	EZDRM username
p	EZDRM password





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

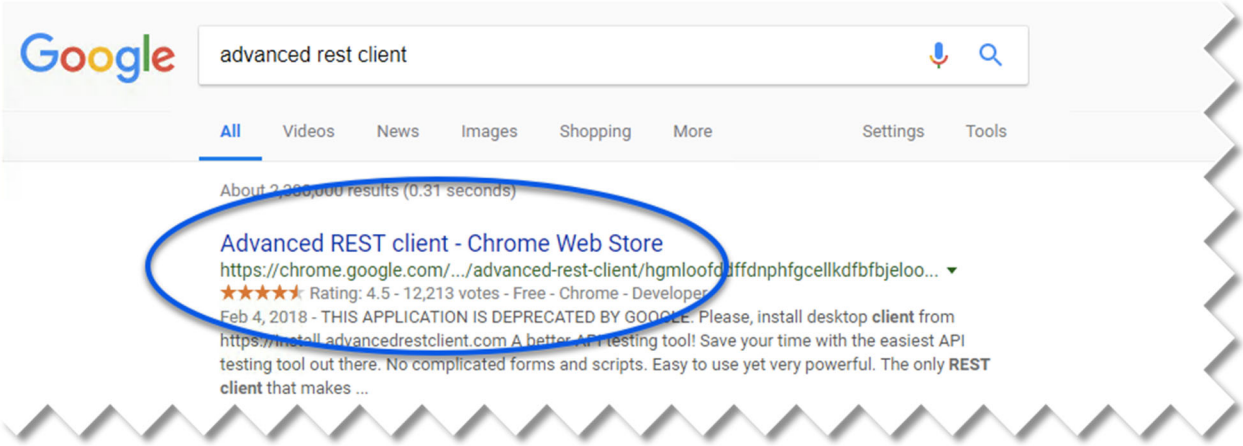
Existing DRM Keys returned:

```
<FairPlay>
    <AssetID>1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXa25</AssetID>
    <KeyHEX>D230XXXXXXXXXXXXX17300XXXXXXXXXXXXX4EABXXXXXXXXXXXXX4349271XXXXXB</KeyHEX>
    <KeyID>0jXXXXXXXXXXXXXwDRjUrJ/XXXXXXXXXXXXXkQ0knFXXX=</KeyID>
    <KeyUri>skd://fps.ezdrm.com/;1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXXa25</KeyUri>
    <LicensesUrl>http://fps.ezdrm.com/api/licenses</LicensesUrl>
    <SupportedFPSVersions>1</SupportedFPSVersions>
 </FairPlay>
```

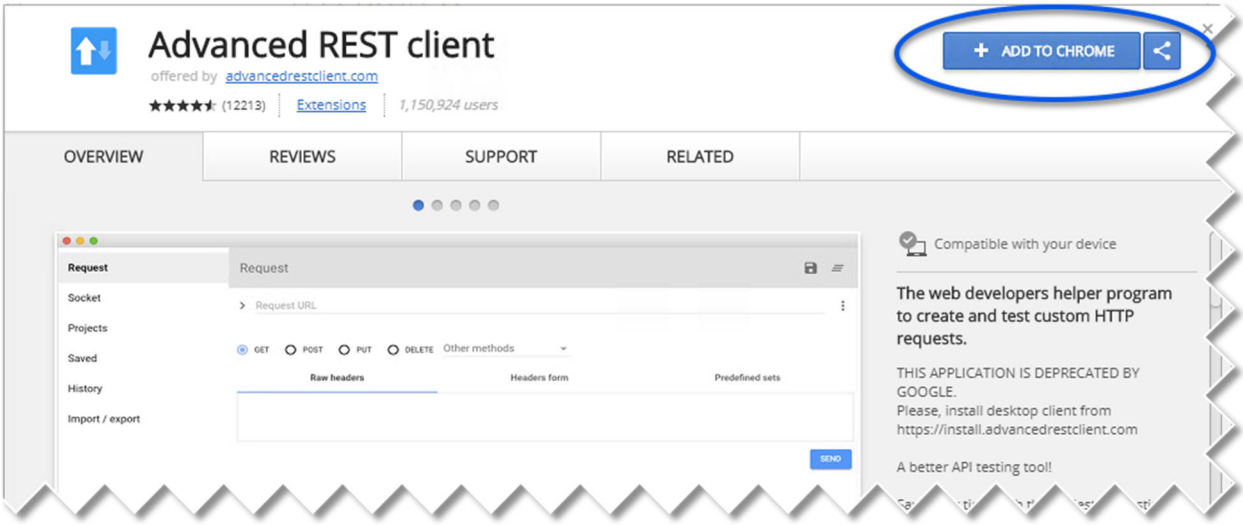
Appendix 2 – Installing ARC Plug-in

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.

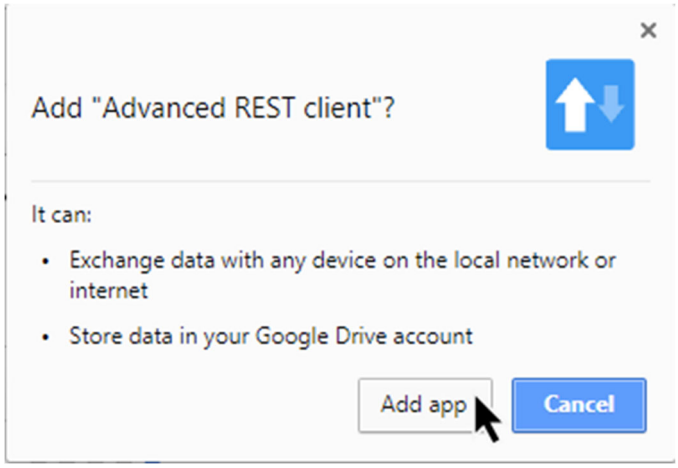
b. Click the "Advanced REST client – Chrome Web Store" link.



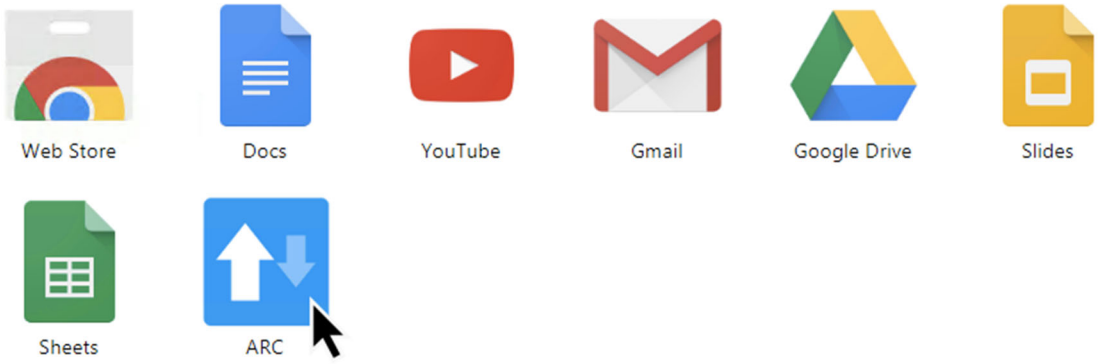
c. Click on the **+Add to Chrome** button to install the plug-in.



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.

Additional Information

For additional questions and comments please contact: simplify@ezdrm.com