

EZDRM Qencode Configuration Transcoding

Table of Contents

Generating DRM Keys	3
Qencode - Transcoding	6
<i>Qencode API Keys</i>	6
<i>Step 1 - Access Token</i>	7
<i>Step 2 - Create Task</i>	9
<i>Step 3 - Create Job</i>	10
Qencode DRM Encoding - Widevine	11
<i>Key Value Definitions</i>	11
Qencode DRM - PlayReady	13
<i>Key Value Definitions</i>	13
Qencode DRM - Apple Fairplay	15
<i>Key Value Definitions</i>	15
Additional Information	18

Version 1.0

Generating DRM Keys

Below are the steps to create the DRM Keys for CENC-Widevine, CENC-PlayReady and Apple Fairplay Streaming encryption for Qencode.

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 CPIX Web Service

1. Call the EZDRM web service in a browser:

<https://cpix.ezdrm.com/keygenerator/cpix.aspx?k=kid&u=username&p=password&c=resourceName>

The parameters are as follows:

Parameter	Description
k=	Key ID value (client generated) in GUID format
u=	Username
p=	Password
c=	Content Resource ID/name, client generated (such as a stream name or media asset name)

* To generate a GUID for the k value, you can use a GUID generator like the one found here: <https://www.guidgenerator.com/>.

2. The response from EZDRM will look like this:

```
<<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc" id="1">
  <cpix:ContentKeyList>
    <cpix:ContentKey kid="b99ed9e5-XXXX-XXXX-XXX8-43692b686ddb" explicitIV="uZ7ZXXXXXXXXXXXXXXXX2ht2w==">
      <cpix:Data>
        <pskc:Secret>
          <pskc:PlainValue>Zf7JmBnXXXXXXXXMI91SQ==</pskc:PlainValue>
        </pskc:Secret>
      </cpix:Data>
    </cpix:ContentKey>
  </cpix:ContentKeyList>
  <cpix:DRMSysSystemList>
    <cpix:DRMSysSystem kid="b99ed9e5-XXXX-XXXX-XXX8-43692b686ddb" systemId="edef8ba9-79d6-4ace-a3c8-27dcd51d21ed">
      <cpix:PSSH>AAAAA3Bzc2gAAAAA7e+LqXnkSs6jyC<EdcL3FFtNBltmh0Inc9PSIInRyWnrcyI6IyJTRCJdfSoCU0Q</cpix:PSSH>
      </cpix:DRMSysSystem>
    <cpix:DRMSysSystem kid="b99ed9e5-XXXX-XXXX-XXX8-43692b686ddb" systemId="9a04f079-9840-4286-ab92-e65be0885f95">
      <cpix:PSSH>AAADbnBcc2gAAJ<AAG6AC8ALubZAGWLABLAG0AYQ8pA4AbQBpAGfAcgvAMfMbv8mHQALgBJAG8AbQvAEQAUjg8W</cpix:PSSH>
      <cpix:ContentProtectionData>PG1zCH16cH3vPJvNSUFBUV8QVFEY0Fq0Fid0JTOUuQVIBQkZBRUVBUkFCRkFGSUF3QU10QUCwQHJ8QvV8SE1BUFFBauPHZ0FKQIvQuh8Qv9nQXZBz0hBY3dCakPHZ0FauJUJ0QUDfQW3QXVBRzBBVVFcaKfISUF1d0J6Qv</cpix:ContentProtectionData>
      </cpix:DRMSysSystem>
    <cpix:DRMSysSystem kid="b99ed9e5-XXXX-XXXX-XXX8-43692b686ddb" systemId="94ce86fb-07ff-4f43-adb8-93d2fa968ca2">
      <cpix:URIExtXKey>c2tk0i8vzn8zLmV6ZHJ<IDH2OTJ1Njg2ZGRi</cpix:URIExtXKey>
      </cpix:DRMSysSystem>
    </cpix:DRMSysSystemList>
  </cpix:CPIX>
```

- **id** – c value returned, generic resource name/identifier (client generated)
- **kid** – key_id in GUID format (client generated)*
- **pskc:PlainValue Secret key**– the Secret Content Encryption Key in Base 64 generated by EZDRM and returned as a plain value.
- **explicitIV** – the Apple FairPlay explicit IV value
- **PSSH** – The modular specific protection system specific header (PSSH) data for the encryption process; Base 64 encoded.
 - **Widevine PSSH** – under systemId="edef8ba9-79d6-4ace-a3c8-27dcd51d21ed"
 - **PlayReady PSSH** – under systemId="9a04f079-9840-4286-ab92-e65be0885f95"

Here is the example XML return:

```
<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc" id="1">
  <cpix:ContentKeyList>
    <cpix:ContentKey kid="b99ed9e5-XXXX-XXXX-XXX8-43692b686ddb" explicitIV="uZ7ZXXXXXXXXXXXXXXXX2ht2w==">
      <cpix:Data>
        <pskc:Secret>
          <pskc:PlainValue>Zf7JmBnXXXXXXXXMI91SQ==</pskc:PlainValue>
        </pskc:Secret>
      </cpix:Data>
    </cpix:ContentKey>
  </cpix:ContentKeyList>
  <cpix:DRMSysSystemList>
```

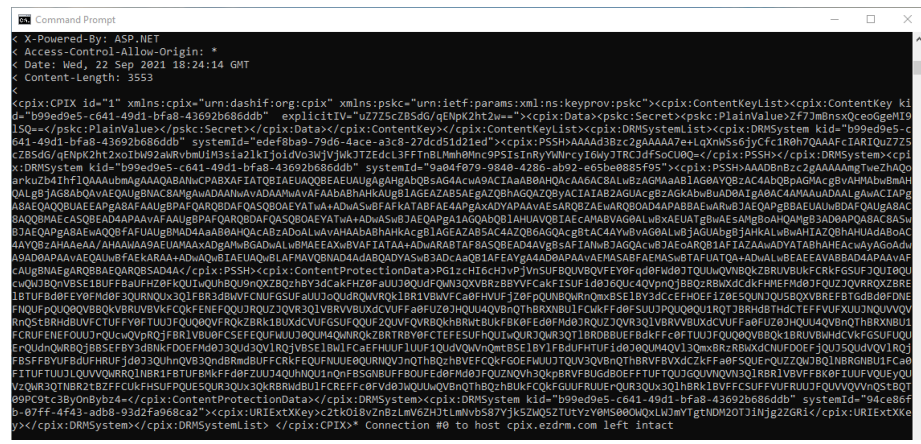
```
<cpix:DRMSystem kid="b99ed9e5-XXXX-XXXX-XXX8-43692b686ddb" systemId="edef8ba9-79d6-4ace-a3c8-27dcd51d21ed"
>
<cpix:PSSH>AAAA3Bzc2gAAAAA7e+LqXnWS6jvCfc1R0XXXXXXXXXXXXXXXXXXXXBSdG/gENpK2ht2xoIbW92awRvbmUiMXXXXXXXXXX
XXXXXXXXXXXXEdcL3FFtNlMmh0Mnc9PSIsInRyYwNrcyI6WYJTRCJdfSoCU00=</cpix:PSSH>
</cpix:DRMSystem>
<cpix:DRMSystem kid="b99ed9e5-XXXX-XXXX-XXX8-43692b686ddb" systemId="9a04f079-9840-4286-ab92-e65be0885f95"
>
<cpix:PSSH>AADBnBzc2gAAAAmgTweZhaOoarkuZb4Ihf1OAAAubmAgAAAOABANwCPABXAFIATOBIAEUAOQBEEAUUgAgAHgAbOBsAG4
AcwA9ACTAaAB0AHOAcAA6AC8ALwBzAGMAaB1AG0AYOBzAC4AbOBpAGMACgBvAHMAbWbMAHOALgBjAG8Ab0AvAEOAUgBNAC8AMgAwADAA
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
EwARwBJAEQAPgBBAEUUwBDAFOAUgA8AC8A0QBMAEcASOBEAD4APAAvFAAUgBPAFOAROBDAFQASOB0AEYATwA+ADwASwBJAEQAPgA1AG0
AbOB1AHUAOVOBIAEcAMABVAG0ALwBxAEUATgBwAEsAMeBoAHOAMgB3AD0APOA8AC8ASwBJAEQAPgA8AEwAOQBFAFUUgBMD4AaAB0AHOAc
ABzADoALwAvAHAAbABhAHKAcgB1AGEAZAB5AC4AZOB6AG0AcgBTAC4AYwBvAG0ALwBjAGUAbgBjAHkALwBwAHIAZQBhAHUAdAbOAC4AYOB
zAHAaAA/AAAAA9AEUAMAAxAdgAmWBGADwALwBMAEFAXwBVAFIATAA+ADwARABTAFA8ASOBEAD4AVgBsAFIAnwBJAG0AcwBJAEoAROB1A
FIATAAwADYATABhAEAcwAvAgoAdwA9AD0APAAvAEOUwBFAEKARAA+ADwAOwBIAEUAOwBLAFMAV0BNAD4AdABOADYASwB3ADcAaOB1AFE
AYgA4AD0APAAvAEMASABFAEMASwBTAFUAT0A+ADwALwBEAEFAVABAD4APAAvFcaUgBNAEgAROBBAEQAROBsAD4A</cpix:PSSH>
```

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.

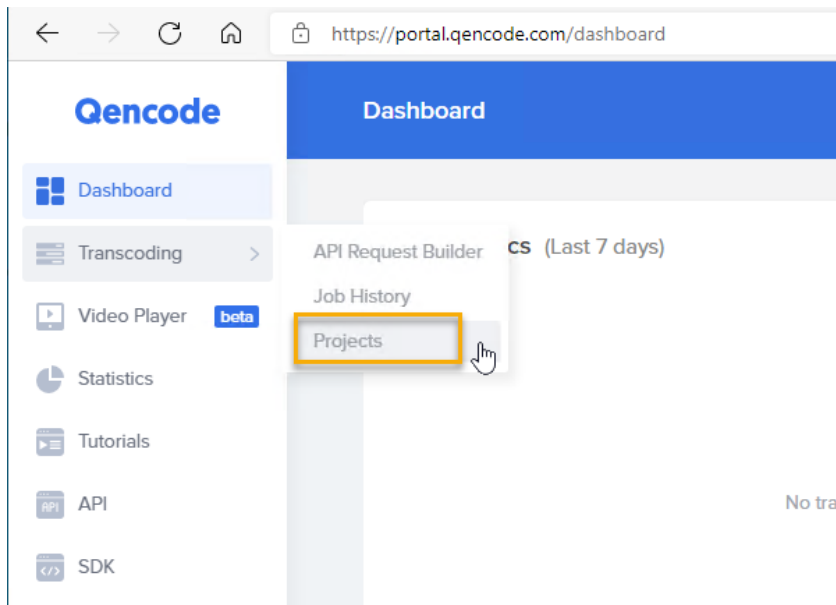
```
curl -v "https://cpix.ezdrm.com/keygenerator/cpix.aspx?k=kid&u=username&p=password&c=1"
```



Qencode - Transcoding

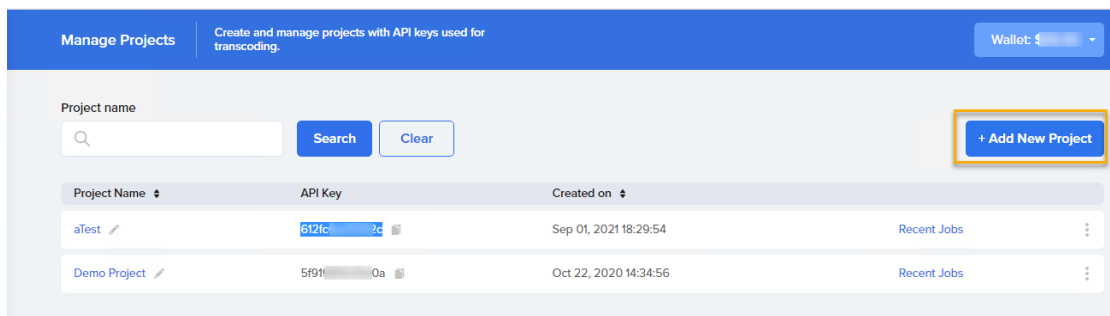
Qencode API Keys

To generate API Requests, copy the **API Key** for authentication from an existing project or create a new project under **Projects** on **Transcoding** Dashboard menu.



To add a new Project:

1. In **Manage Projects** page, click **Add New Project**.



2. Enter Project Name and click **Add project**.



Add new project

Choose a name for your Project

3. Copy the **API Key** for the new Project.

Manage Projects
Create and manage projects with API keys used for transcoding.

Project name

Project Name	API Key	Created on
DemoProject2 <small>✎</small>	6138...ba <small>🗑</small>	Sep 08, 2021 13:21:33

Step 1 - Access Token

In Postman, create **POST** request to the API https://api.gencode.com/v1/access_token

Under **Body**, select **form-data** and add the following values:

Key	Value
api_key	The Project API Key value

POST https://api.qencode.com/v1/access_token

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

none **form-data** x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> api-key	61381...da			
Key	Value	Description		

The response will be the Authentication **token** and expiration.

POST https://api.qencode.com/v1/access_token

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies

none **form-data** x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> api_key	61: 42c	API Key		
Key	Value	Description		

Body Cookies (1) Headers (16) Test Results 200 OK 1819 ms 771 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "error": 0,
3    "token": "6cfa: 5cc6f598",
4    "expire": "2021-09-09T17:22:12"
5  }

```

This **token** will be used to create a Task.

Note: Every Job needs a Task. You cannot run multiple Jobs using the same Task.

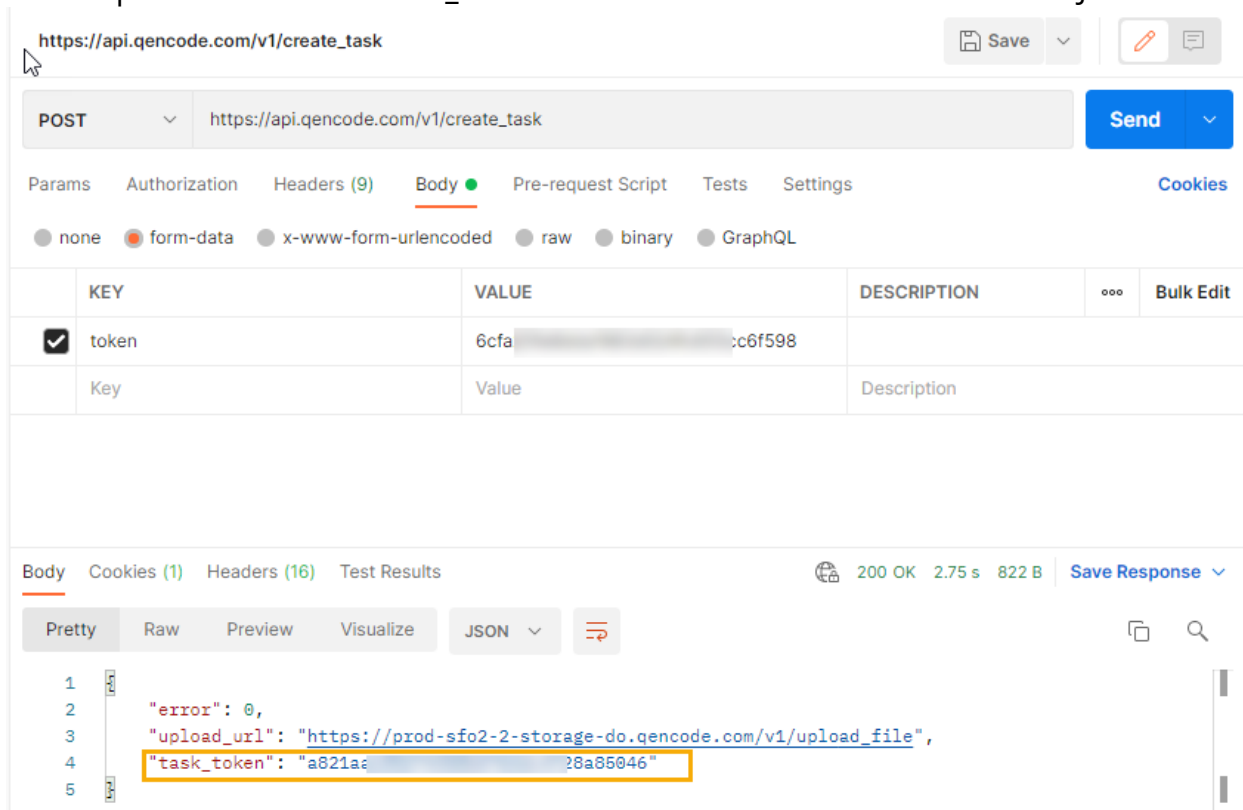
Step 2 - Create Task

In Postman, create **POST** request to the API https://api.qencode.com/v1/create_task

Under **Body**, select **form-data** and add the following values:

Key	Value
token	The token value from the Access Token POST

The response will be the **task_token** which will be needed to create the Job.



The screenshot shows a Postman interface for a POST request to `https://api.qencode.com/v1/create_task`. The request body is set to `form-data` with a single key-value pair: `token` with a value of `6cfa[redacted]c6f598`. The response status is `200 OK` with a response time of `2.75 s` and a size of `822 B`. The response body is shown in JSON format:

```

1  {
2    "error": 0,
3    "upload_url": "https://prod-sfo2-2-storage-do.qencode.com/v1/upload_file",
4    "task_token": "a821a[redacted]28a85046"
5  }

```

The `"task_token"` field is highlighted with a yellow box.

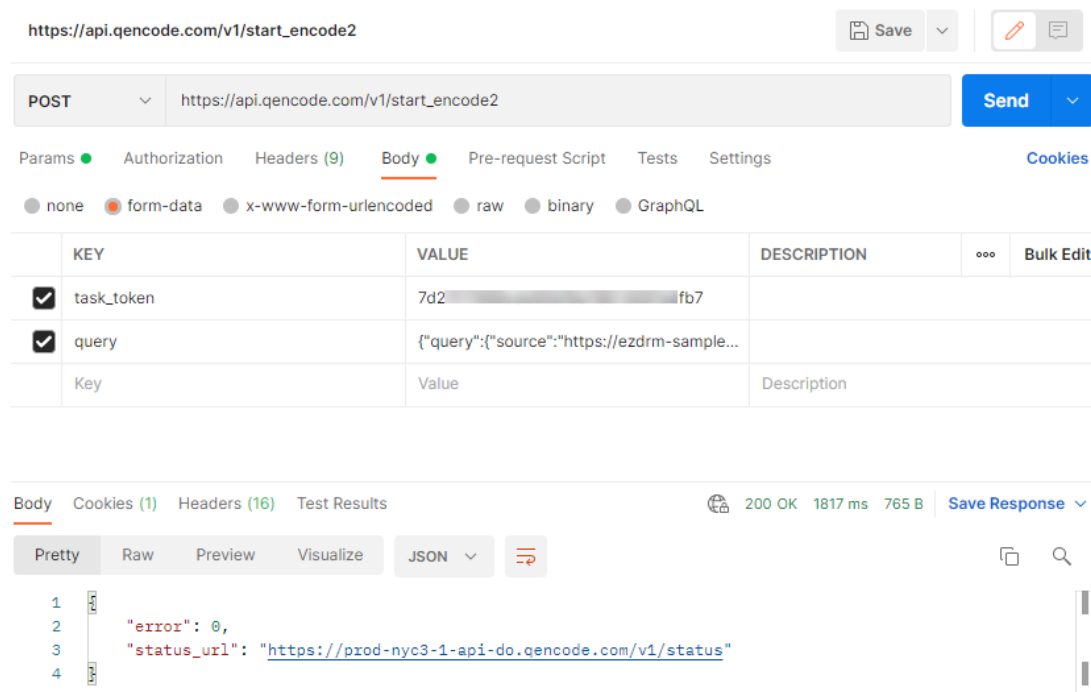
Step 3 - Create Job

In Postman, create **POST** request to the API

https://api.qencode.com/v1/start_encode2

Under **Body**, select **form-data** and add the following values:

Key	Value
task_token	The task_token value from Create Task POST
query	JSON format for Job Details



The JSON query format for Job Details is outlined at

https://portal.qencode.com/request_builder.

The sample base output format includes:

```
{
  "query": {
    "format": [
      {
        "output": "mp4",
        "video_codec": "libx264",
        "height": 2160,
        "quality": 23,
        "audio_bitrate": 128
      }
    ],
    "encoder_version": "2",
    "source": "https://nyc3.s3.qencode.com/qencode/samples/1080-sample.mov"
  }
}
```

The next sections of this document detail the **drm** object formatting for Widevine, PlayReady and FairPlay.

Qencode DRM Encoding - Widevine

For DRM encoding, add a 'cenc_drm' object to your Qencode API request JSON query as shown in this example:

```
"cenc_drm": {
  "key_id" : "b99ed9e5XXXXXXXXXX843692b686ddb",
  "key" : "65fecXXXXecc5071xxXXXXe308f6549",
  "pssh" : "AAAAad3Bzc2gAAAAA7e+LqXnWSs6ivCfc1R0XXXXXXXXXXXXXXXXXXXXXSdG/aENpK2ht2xoIbw92aWRvbmUiMXXXXXXXXXXXXXXXXXXXXXXXXXEdcL3FFTnBLMmh0Mnc9PSIsInRyYWNRcyI6WvJTRCJdfSoCU00="
}
```

Key Value Definitions

- **key_id**: The **kid** used for encryption (also known as KID); Base 64 encoded with no dashes.

Note: Please make sure you remove dash - characters from key_id value when submitting in Qencode API request since it should be in hex format.

- **key**: The DRM content encryption key (128 bit key); Base 64 encoded.

For the **key** value use the **pskc:Secret key** value and decode the Plain Value tag from Base 64 to HEX format in lowercase (no dashes). An example decoder can be found at:

https://tomeko.net/online_tools/base64.php?lang=en

pskc:Secret key (Base 64) = Zf7JmBnxxXXXXXXXXMI9ISQ==



(KeyHEX) = 65fecXXXXecc5071xxXXXXe308f6549

- **PSSH** – The modular specific protection system specific header (PSSH) data for the encryption process; Base 64 encoded.

Note: For Widevine use *cpix:PSSH xml tag from systemId="edef8ba9-79d6-4ace-a3c8-27dcd51d21ed"*

You need to add 'cenc_drm' object as a direct child of the format object. Here's the full example:

```
{
  "query": {
    "source": "https://yourserver.com/video.mp4"
    "format": [
      {
        "output": "advanced_dash",
        "destination": {
          ...
        },
        "stream": [
          {
            "video_codec": "libx264",
            "height": 720,
            "width": 1280,
            "audio_bitrate": 128,
            "keyframe": 25,
            "bitrate": 3100
          }
        ],
        "cenc_drm": {
          "key_id" : "b99ed9e5XXXXXXXXXX843692b686ddb",
          "key" : "65fecXXXXecc5071xxXXxe308f6549",
          "pssh" : "AAAd3Bzc2gAAAAA7e+LqXnWSs6jyCfc1R0XXXXXXXXXXXXXXXXXXXXBSdG/qENpK2ht2xoIbw92awRvbmUiMXXXXXXXXXXXXXXXXXXXXEdcL3FFTnBLMmh0Mnc9PSIsInRvYWNrcyI6WvJTRCJdfSoCU00="
        }
      }
    ]
  }
}
```

Qencode DRM - PlayReady

For DRM encoding, add a 'cenc_drm' object to your Qencode API request JSON query as shown in this example:

```
"cenc_drm": {
  "key_id" : "b99ed9e5XXXXXXXXXX843692b686ddb",
  "key" : "65fecXXXXecc5071xxXXXe308f6549",
  "pssh" : "AAADBnBzc2gAAAAAmgTweZhaQoarkuZb4Ihf1OAAUbmAgAAA0ABANwCPABXAFIATOBIAEUAOBFAEUAgAgAHgAbOBsAG
4AcwA9ACTAaAB0AH0AcAA6AC8ALwBzAGMAaAB1AG0AYOBzAC4AbOBpAGMACgBvAHMAbWbMAHQALgBjAG8AB0AvAF0AUgBNAC8AMgAwADAA
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
AEwARwBJAE0APgBBAEUAUwBDAFOAUgA8AC8A00BMAEcAS0BEAD4APAAvAFAAUgBPAFOAR0BDAFOAS0B0AEYATwA+ADwASwBJAE0APgA1AG
OAb0B1AHUAVOBIAEcAMABVAG0ALwBxAEUATgBwAEsAMgBoAH0AMgB3AD0APOA8AC8ASwBJAE0APgA8AEwA00BFAFUAGBMAD4AaAB0AH0A
cABzADoALwAvAHAAbABhAHkAcgB1AGEAZAB5AC4AZ0B6AG0AcgBtAC4AYwBvAG0ALwBjAGUAbgBjAHkALwBwAHIAZ0BhAHUAdAB0AC4AYO
BzAHAAeAA/AHAAWAA9AEUAMAAxADgAMwBGADwALwBMAEEFAXwBVAFIATAA+ADwARABTAF8AS0BEAD4AVgBsAFIANwBJAG0AcwBJAEoAROB1
AFIAZAAwADYATABhAHEAcwAvAGoAdwA9AD0APAAvAE0AUwBFAEKARAA+ADwAQwBIAEUAOwBLAFMAVQBNA4AdABOADYASwB3ADcAa0B1AF
EAYgA4AD0APAAvAEMASABFAEMASwBTAUFUAT0A+ADwALwBEAEFAVABBAD4APAAvAFcAUgBNAEgAR0BBAE0AR0BSAD4A"
```

Key Value Definitions

- **key_id:** The **kid** used for encryption (also known as KID); Base 64 encoded with no dashes.

Note: Please make sure you remove dash - characters from key_id value when submitting in Qencode API request since it should be in hex format.

- **key:** The DRM content encryption key (128 bit key); Base 64 encoded.

For the **key** value use the **pskc:Secret key** value and decode the Plain Value tag from Base 64 to HEX format in lowercase (no dashes). An example decoder can be found at:

https://tomeko.net/online_tools/base64.php?lang=en

pskc:Secret key (Base 64) = Zf7JmBnxxXXXXXXXXMI9ISQ==



(KeyHEX) = 65fecXXXXecc5071xxXXXXe308f6549

- **PSSH** – The modular specific protection system specific header (PSSH) data for the encryption process; Base 64 encoded.

Note: For PlayReady use cpix:**PSSH** xml tag with systemId="9a04f079-9840-4286-ab92-e65be0885f95"

You need to add 'cenc_drm' object as a direct child of the format object. Here's the full example:

```
{
  "query": {
    "source": "https://yourserver.com/video.mp4"
    "format": [
      {
        "output": "advanced_dash",
        "destination": {
          ...
        },
        "stream": [
          {
            "video_codec": "libx264",
            "height": 720,
            "width": 1280,
            "audio_bitrate": 128,
            "keyframe": 25,
            "bitrate": 3100
          }
        ],
        "cenc_drm": {
          "key_id" : "b99ed9e5XXXXXXXXXX843692b686ddb",
          "key" : "65fecXXXXecc5071xxXXXXe308f6549",
          "pssh" : "AAADBnBzc2gAAAAAmgTweZhaOoarkuZb4Thf1OAAAubmAgAAAQABANwCPABXAFIATOBIAEUAO0BEAEUAUgAgA
HgAb0BsAG4AawA9ACIAaAB0AH0AcAA6AC8ALwBzAGMAaABlAG0AYOBzAC4Ab0BpAGMAcgBvAHMAbwBmAHQALgBjAG8AbQAvAEQAUgBNAC8
AMgAwADAAXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXAEwARwBJAE0APgBBAEUAUwBDAFOAUgA8AC8A00BMAEcASOBEAD4APAAvFAAAUgBPAFOAR0BDAFOASOBAEYATwA+ADwASwBJA
```

```

EOAPgA1AG0AbOB1AHUAVOBIAFcAMABVAG0ALwBxAEUATgBwAEsAMgBoAHOAMgB3AD0APOA8AC8ASwBJAEQAPgA8AEwAOQBfAFUUAUgBMAD4
AaAB0AHOAcABzADoALwAvAHAAbABhAHkAcgB1AGEAZAB5AC4AZOB6AG0AcgBtAC4AYwBvAG0ALwBjAGUAbgBjAHkALwBwAHIAZOBhAHUAd
ABoAC4AYOBzAHAAeAA/AHAAWAA9AEUAMAAxADgAMwBGADwALwBMAEEAXwBVAFIATAA+ADwARABTAF8ASOBEAD4AVgBsAFIANwBJAGOAcwB
JAEoAROB1AFIAZAawADYATABhAHEAcwAvAGoAdwA9AD0APAAvAEQAUwBfAEkARAA+ADwAQwBIAEUAQwBLAFMAVOBNAD4AdABOADYASwB3A
DcAaOB1AFEAYgA4AD0APAAvAEMASABFAEMASwBTAFUATQA+ADwALwBEAEEAVABBAD4APAAvAFcAUgBNAEgAROBBAEQAROBSD4A"
    }
    }
  ]
}
}

```

Qencode DRM - Apple Fairplay

For DRM encoding, add a 'fps_drm' object to your Qencode API request JSON query as shown in this example:

```

"fps_drm" : {
  "key" : "65fecXXXXXecc5071xxXXXe308f6549",
  "iv" : "b99ed9e5cXXXXxxXXXXxx692b686ddb",
  "key_url" : "skd://fps.ezdrm.com/;b99ed9e5-xxx-XXxX-xxxX-43692b686ddb"
}

```

Key Value Definitions

Here are the descriptions of the items returned:

- **key**: This is the common content encryption key in HEX format (32 hexadecimal characters), the EZDRM **KeyHEX** value.
 - Use the **pskc:Secret key** value and decode from Base 64 to HEX format in lowercase (no dashes). An example decoder can be found at: https://tomeko.net/online_tools/base64.php?lang=en

pskc:Secret key (Base 64) = Zf7JmBnsxXxxxXgeMI9ISQ==



key (HEX) = 65fecXXXXxxxXX071ea0681e308f6549

- **iv**: the initialization vector (IV) used for encryption. This is **explicitIV** value in HEX with no dashes.

Decode the **explicitIV** Plain Value Base 64 to HEX format. An example decoder can be found at:

https://tomeko.net/online_tools/base64.php?lang=en

explicitIV (Base 64) = uZ7Z5cXXXXXXXXXXpK2ht2w==



iv (HEX no dashes) = b99ed9e5cXXXXxxXXXXxx692b686ddb

- **key_url**: the License URL used for the license key call, this is the EZDRM **KeyURI**.

Build by appending the **kid** value to base URL "skd://fps.ezdrm.com/;<kid>" for example:

skd://fps.ezdrm.com/;b99ed9e5-xXXX-XXxX-xxxX-43692b686ddb

You need to add 'fps_drm' object as a direct child of the format object. Here's the full example:

```
{
  "query": {
    "source": "https://yourserver.com/video.mp4"
    "format": [
      {
        "output": "advanced_hls",
        "destination": {
          ...
        },
        "stream": [
          {
            "size": "640x360",
            "audio_bitrate": "128"
          },
          {
            "size": "852x480",
            "audio_bitrate": "128"
          },
          {
            "size": "1280x720",
            "audio_bitrate": "320"
          }
        ],
        "fps_drm" : {
          "key" : "65fecXXXXecc5071xxXXXe308f6549",
          "iv" : "b99ed9e5cXXXXxxXXXXxx692b686ddb",
          "key_url" : "skd://fps.ezdrm.com/b99ed9e5-xxx-xxx-xxx-43692b686ddb"
        }
      }
    ]
  }
}
```

Additional Information

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