EZDRM WOWZA Configuration

WOWZA Streaming Cloud

Table of Contents

Version 1.0 / June 22, 2021

Introduction

Digital rights management (DRM) technology provides a way, through encryption, for content creators to protect copyrights and unauthorized distribution of their digital media. The Wowza Streaming Cloud REST API provides integration with EZDRM to protect content from unauthorized viewing.

Currently, the following EZDRM key management systems are supported by Wowza Streaming Cloud:

- **EZDRM Universal DRM** – Supports MPEG-DASH playback for Google Widevine and Microsoft PlayReady devices and platforms using a linked Common Encryption (CENC) key.
- **EZDRM Apple FairPlay Streaming** – Supports HLS playback for content to Apple devices with native support for the HTML 5 player in macOS Safari browsers or Safari 11.3 on iOS.

While you can implement DRM for Apple and Widevine/PlayReady individually, in most cases you'll want to complete both of the following tasks to ensure your stream is protected on as many devices and platforms as possible:

- Protect streams for iOS and Apple devices with EZDRM and the Wowza Streaming Cloud REST API
- Protect streams for Google Widevine and Microsoft PlayReady devices with EZDRM and the Wowza Streaming Cloud REST API

Prerequisite

Important: To protect streams using EZDRM, you must have an EZDRM account, configured appropriately for the device types you want to stream to. For documenation and information about account setup, visit: https://www.ezdrm.com/

Universal DRM (CENC-Widevine & PlayReady)

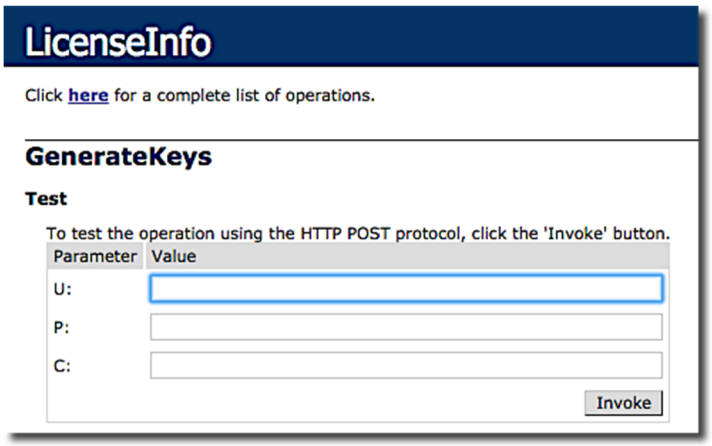
EZDRM Universal DRM is a combination of Google Widevine Modular with Microsoft PlayReady; both using linked CENC keys over DASH streaming. This enables a content owner to encrypt the media once with CENC keys and deliver either a PlayReady License or a Widevine License depending on the player and platform calling for a license.

Generating Keys

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 *EZDRM KeyZ API guide at www.ezdrm.com for more details on calling existing keys with Content_ID.*

3. The response from EZDRM will look like this:

Make note of the following values in the response from EZDRM:

- The **ContentID**. You'll use this to configure the stream protection in Wowza Streaming Cloud.

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=""'
```

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

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 *EZDRM KeyZ API guide at www.ezdrm.com for more details on calling existing keys with Content_ID.*

The example above is for scripts run on Mac. When running on a PC use double quotes as shown below:

```
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 ""

The following is returned from the web service:

```
<EZDRM xmlns="">
        <WideVine diffgr:id="WideVine1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
          <ContentID>6IxXXx0Z8xXXXXXXXXLbg==</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": "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":"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>1Xq+XXXXXX0=</Checksum>
        </PlayReady>
     </EZDRM>
```

Configure the stream for Universal DRM protection

To protect a stream using the EZDRM key you obtained in the previous step, you'll need to set the following EZDRM properties on the transcoder.

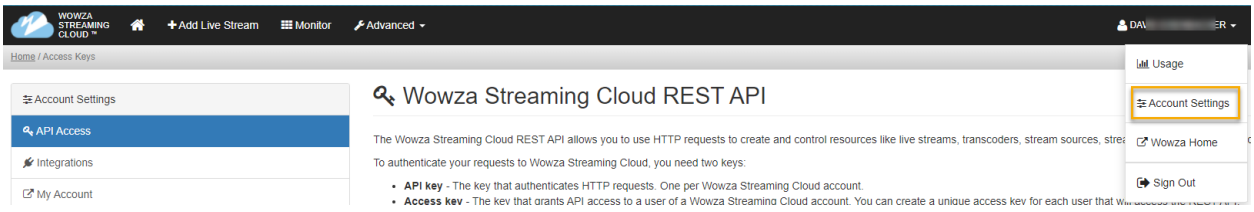
section	key	value	Description
ezdrm	**username**	string	Your EZDRM user name
ezdrm	**password**	string	Your EZDRM password
ezdrm	**wideVineContentId**	string	The content ID you generated from EZDRM. **Note:** While the key name indicates this value is for Widevine, this sets the value for both Widevine and PlayReady DRM.

You can configure the EZDRM properties when you create a transcoder or by updating an existing transcoder.

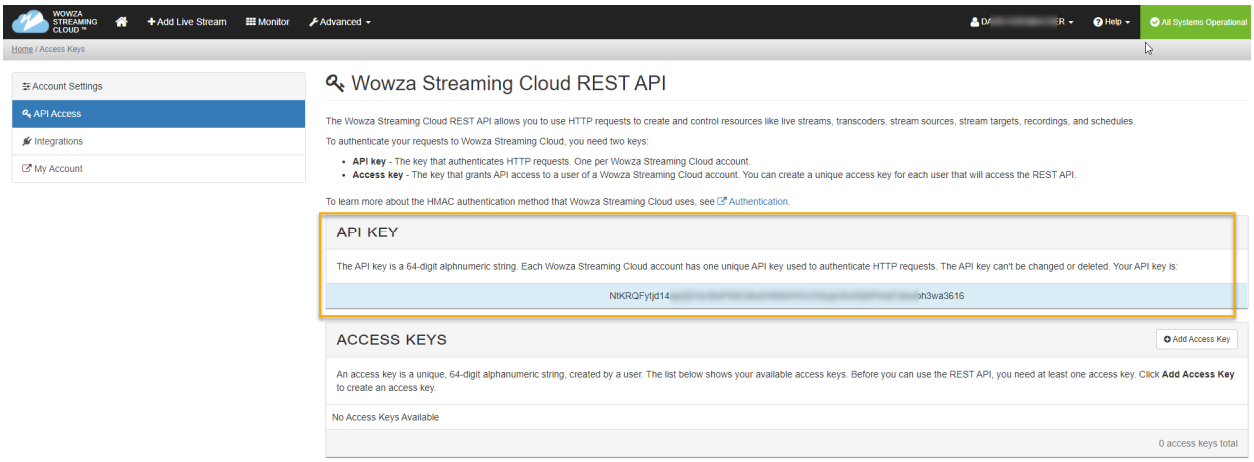
Note: Wowza Streaming Cloud does not validate the EZDRM values you specify in these properties. Make sure you enter the correct values.

WSC API Key and Access Key

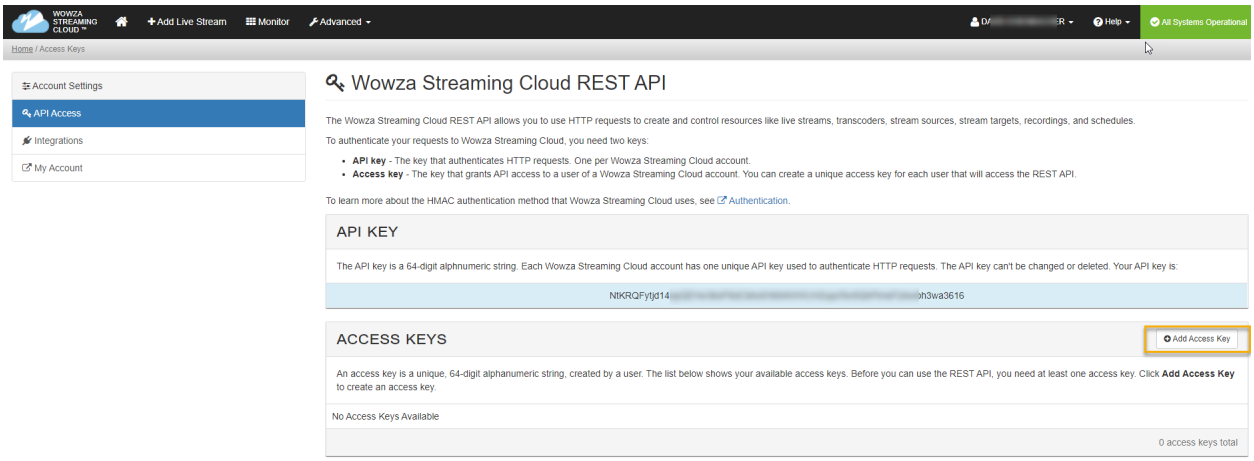
The **API key** is a 64-digit alphanumeric string. Each Wowza Streaming Cloud account has one unique API key used to authenticate HTTP requests. The API key can't be changed or deleted. To find your API Key, under your username menu, select Account Settings.



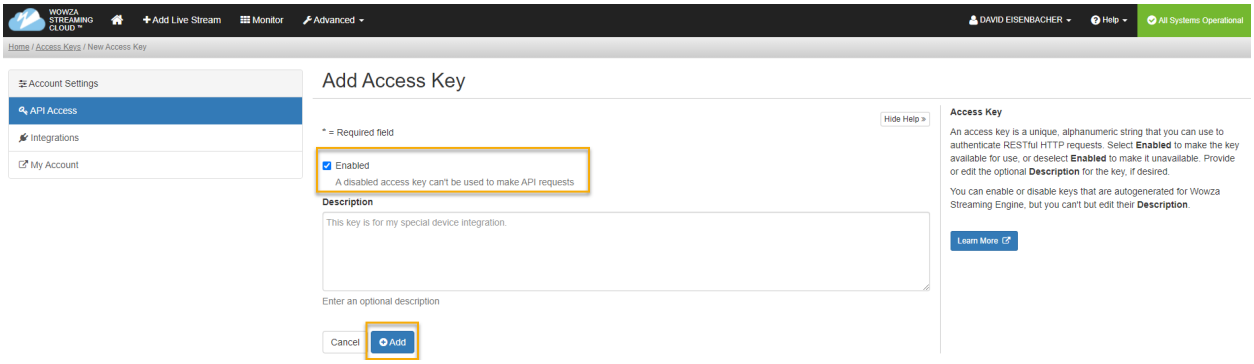
Locate your **API Key** on the API Access Page



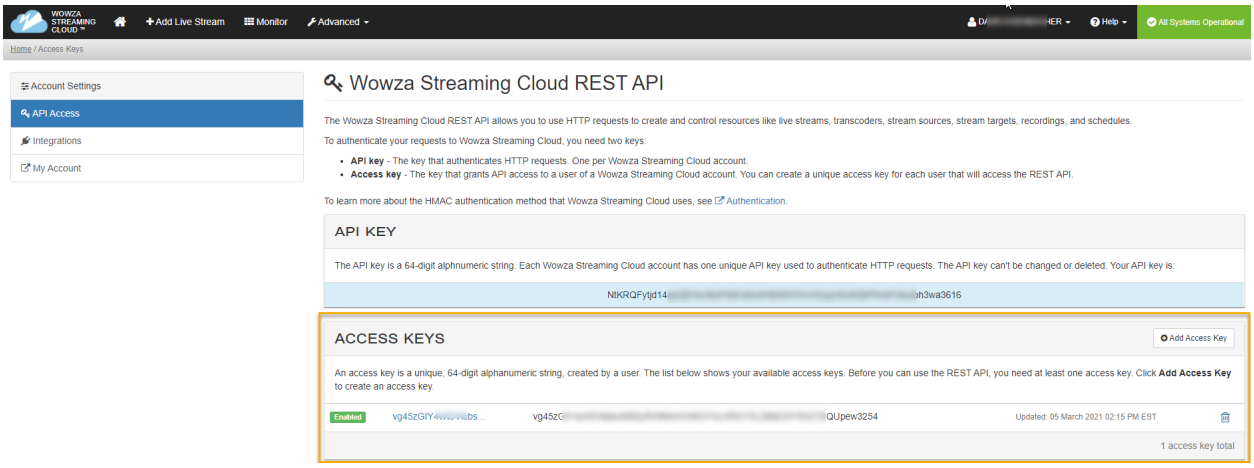
An **Access Key** is a unique, 64-digit alphanumeric string, created by a user. Before you can use the REST API, you need at least one access key. Click **Add Access Key** to create an access key.



Be sure the **Enabled** checkbox is checked (Enabled makes the key available for use). Then click **Add** to create a new Access Key.



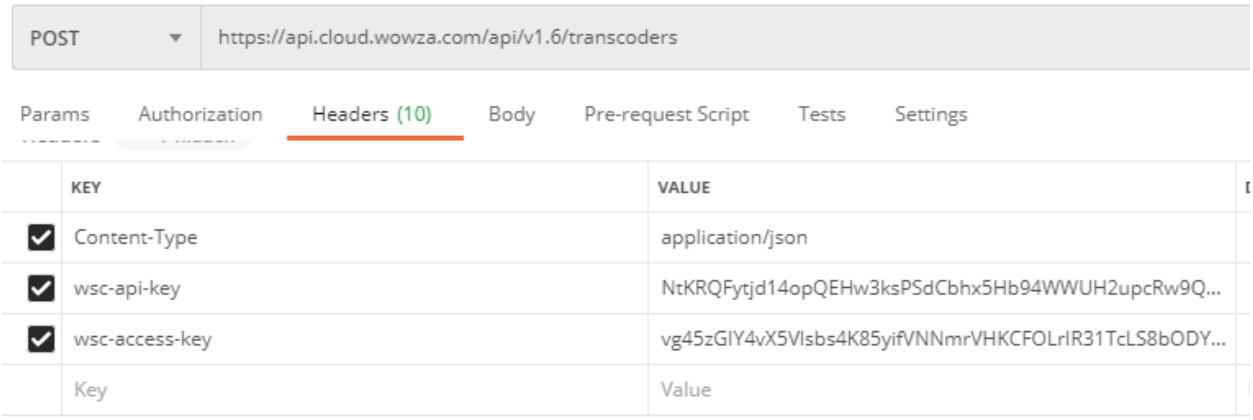
The Access Key will now be available under API Access:

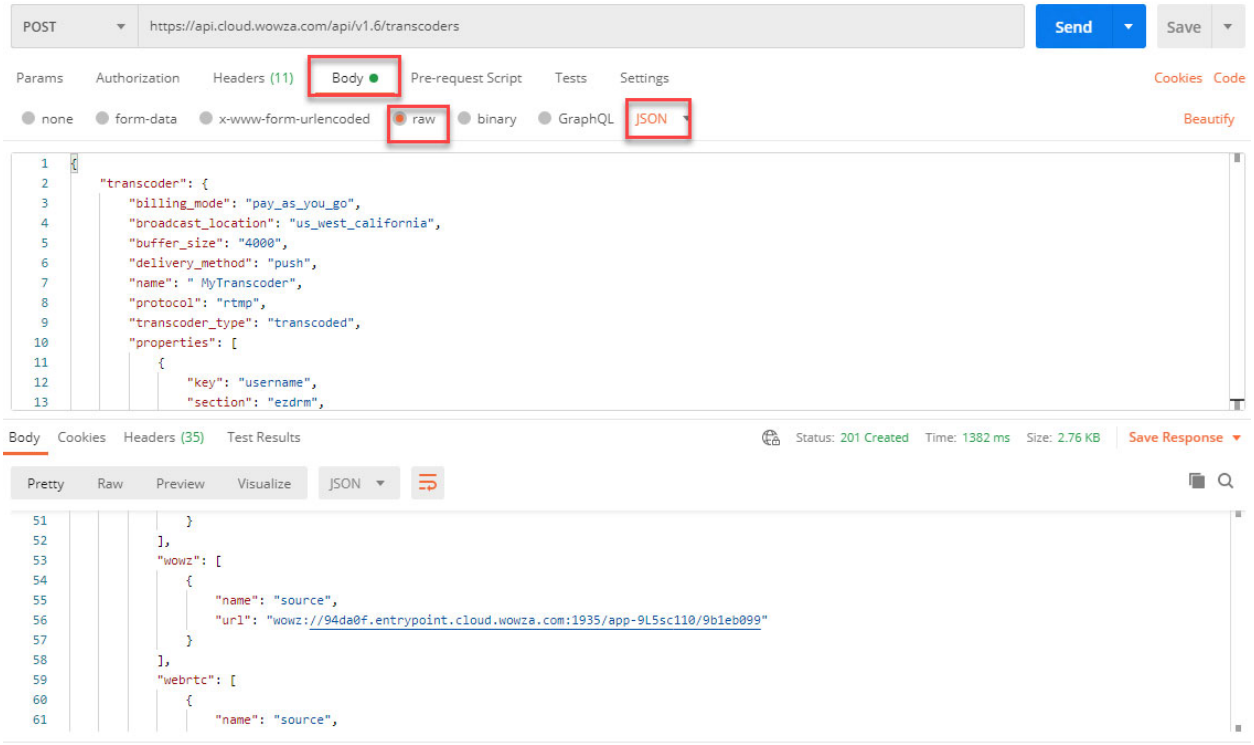


Configure Universal DRM when creating a new transcoder

To configure DRM when creating a new transcoder, utilize curl or emulate in Postman as shown (POST to https://api.cloud.wowza.com/api/v1.6/transcoders) and update the values as shown in the following example:

```
curl -X POST \
-H "Content-Type: application/json" \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
-d '{
   "transcoder": {
     "billing_mode": "pay_as_you_go",
     "broadcast_location": "us_west_california",
     "buffer_size": "4000",
     "delivery_method": "push",
     "name": " MyTranscoder",
     "protocol": "rtmp",
     "transcoder_type": "transcoded",
     "properties": [
     {
        "key": "username",
        "section": "ezdrm",
        "value": "your_ezdrm_username"
      },
      {
        "key": "password",
        "section": "ezdrm",
        "value": "your_ezdrm_password"
      },
      {
        "key": "wideVineContentId",
        "section": "ezdrm",
        "value": "content_id_from_ezdrm"
      }
     ]
   }
}' "${WSC_HOST}/api/${WSC_VERSION}/transcoders"
```





Sample response

```
{
    "transcoder": {
        "id": "gpdXXXzv",
        "name": " MyTranscoder",
        "transcoder_type": "transcoded",
        "billing_mode": "pay_as_you_go",
        "broadcast_location": "us_west_california",
```

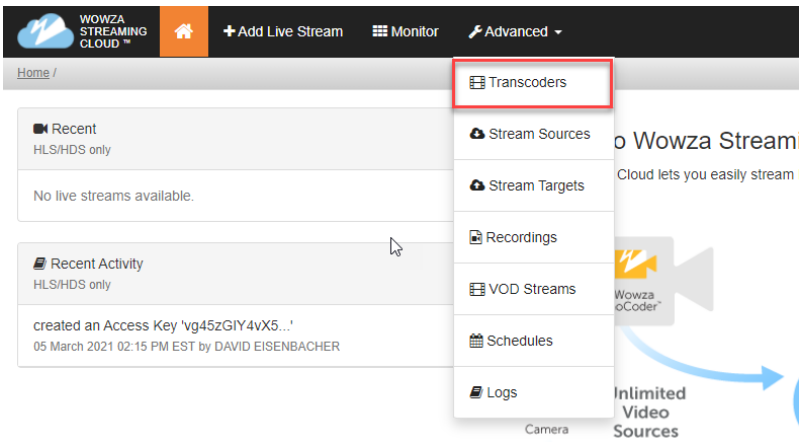
```
        "closed_caption_type": "none",
        "protocol": "rtmp",
        "delivery_method": "push",
        "source_port": 1XX5,
        "domain_name": "94XXXXf.entrypoint.cloud.wowza.com",
        "application_name": "app-9XXXX110",
        "stream_name": "4XXXXX47",
        "playback_stream_name": "9b1XXXX9",
        "delivery_protocols": [
            "rtmp",
            "rtsp",
            "wowz",
            "hls",
            "webrtc"
        ],
        "buffer_size": 4000,
        "low_latency": false,
        "stream_smoother": false,
        "idle_timeout": 1200,
        "play_maximum_connections": 10,
        "disable_authentication": false,
        "username": "client64782",
        "password": "9XXXX20d",
        "watermark": false,
        "created_at": "2021-03-08T13:48:08.000Z",
        "updated_at": "2021-03-08T13:48:08.000Z",
        "direct_playback_urls": {
            "hls": [
                {
                    "name": "default",
                    "url": "https://9XXXXf.entrypoint.cloud.wowza.com/app-9LXXXXX0/ngrp:9bXXXX99_all/playl
ist.m3u8"
                }
            ],
            "rtmp": [
                {
                    "name": "source",
                    "url": "rtmp://9XXXXf.entrypoint.cloud.wowza.com/app-9LXXXXX0/9bXXXX99"
                }
            ],
            "rtsp": [
                {
                    "name": "source",
                    "url": "rtsp://9XXXXf.entrypoint.cloud.wowza.com:1935/app-9LXXXXX0/9bXXXX99"
```

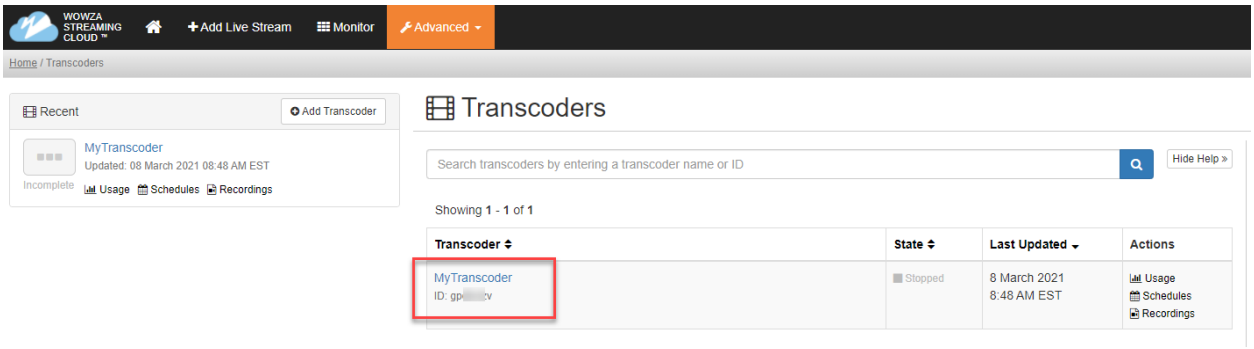
```
                }
            ],
            "wowz": [
                {
                    "name": "source",
                    "url": "wowz://9XXXXf.entrypoint.cloud.wowza.com:1935/app-9LXXXXX0/9bXXXX99"
                }
            ],
            "webrtc": [
                {
                    "name": "source",
                    "url": "wss://9XXXXf.entrypoint.cloud.wowza.com/webrtc-session.json",
                    "application_name": "app-9LXXXXX0",
                    "stream_name": "9bXXXX99"
                }
            ]
        },
        "outputs": []
    }
}
```

Configure DRM on an existing transcoder

To configure DRM on an existing transcoder, utilize curl or emulate in Postman as shown (PATCH to *https://api.cloud.wowza.com/api/v1.6/transcoders/[transcoder_id]*) and update the values as shown in the following example.

Find your Transcoder IDs under the Advanced menu / Transcoders:






```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
-d '{
    "transcoder": {
      "properties": [
        {
          "key": "username",
          "section": "ezdrm",
          "value": "your_ezdrm_username"         },
        {
          "key": "password",
          "section": "ezdrm",
          "value": "your_ezdrm_password"
        },
        {
          "key": "wideVineContentId",
          "section": "ezdrm",
          "value": "content_id_from_ezdrm"
        }
      ]
   }
}' "${WSC_HOST}/api/${WSC_VERSION}/transcoders/[transcoder_id]"
```

Enable MPEG-DASH streaming

EZDRM Universal DRM encrypts MPEG-DASH streams, and MPEG-DASH is only available on Fastly stream targets. HLS is the default delivery protocol for Fastly stream targets, so you must enable MPEG-DASH.

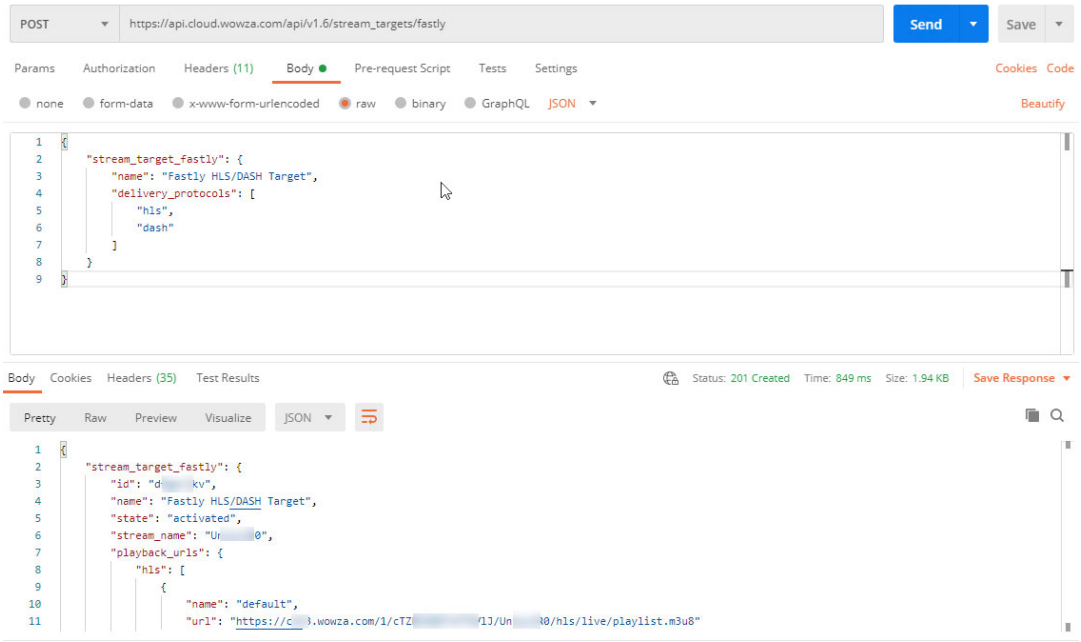
Note: Enabling MPEG-DASH will incur additional egress charges. Egress is incurred per protocol.

You can enable MPEG-DASH when you create a Fastly stream target or by updating an existing Fastly stream target.

Note: You can only enable MPEG-DASH using v 1.7 (beta) of the REST API.

Enable when creating a new Fastly stream target

```
curl -X POST \
-H "Content-Type: application/json" \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
-d '{
     "stream_target_fastly": {
        "name": "Fastly HLS/DASH Target",
        "delivery_protocols": [
            "hls",
            "dash"
          ]
      }
}' "${WSC_HOST}/api/${WSC_VERSION}/stream_targets/fastly"
```



Sample response

Make note of the DASH playback URL in the response, because you'll use this when you test playback.

```
{
    "stream_target_fastly": {
        "id": "zfqXXv4f",
        "name": "My Target",
        "state": "activated",
        "stream_name": "OGXXYnNQ",
        "delivery_protocols": [
            "hls",
            "dash"
        ],
        "playback_urls": {
            "hls": [
                {
                    "name": "default",
                    "url": "https://domain.wowza.com/1/VGl5YXXXXJXSCt4/OG40YnNQ/hls/live/playlist.m3u8"
                }
            ],
            "dash": [
                {
                    "name": "default",
                    "url": "https://domain.wowza.com/1/VGl5YXXXXJXSCt4/OG40YnNQ/dash/live/manifest.mpd"
                }
            ]
        },
        ...
    }
}
```

Tip: Make sure the MPEG-DASH enabled stream target is added to the transcoder you configured for stream protection.

Enable when updating an existing Fastly stream target

```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
-d '{
     "stream_target_fastly": {
        "delivery_protocols": [
            "hls",
            "dash"
          ]
      }
}' "${WSC_HOST}/api/${WSC_VERSION}/stream_targets/fastly/[ID]"
```

Sample response

Make note of the DASH playback URL in the response, because you'll use this when you test playback.

```
{
    "stream_target_fastly": {
        "id": "zfqXXv4f",
        "name": "My Target",
        "state": "activated",
        "stream_name": "OGXXYnNQ",
        "delivery_protocols": [
            "hls",
            "dash"
        ],
        "playback_urls": {
            "hls": [
                {
                    "name": "default",
                    "url": "https://domain.wowza.com/1/VGl5YVpjNjJXSCt4/OG40YnNQ/hls/live/playlist.m3u8"
                }
            ],
            "dash": [
                {
                    "name": "default",
                    "url": "https://domain.wowza.com/1/VGl5YVpjNjJXSCt4/OG40YnNQ/dash/live/manifest.mpd"
                }
            ]
        },
```

```
        ...
    }
}
```

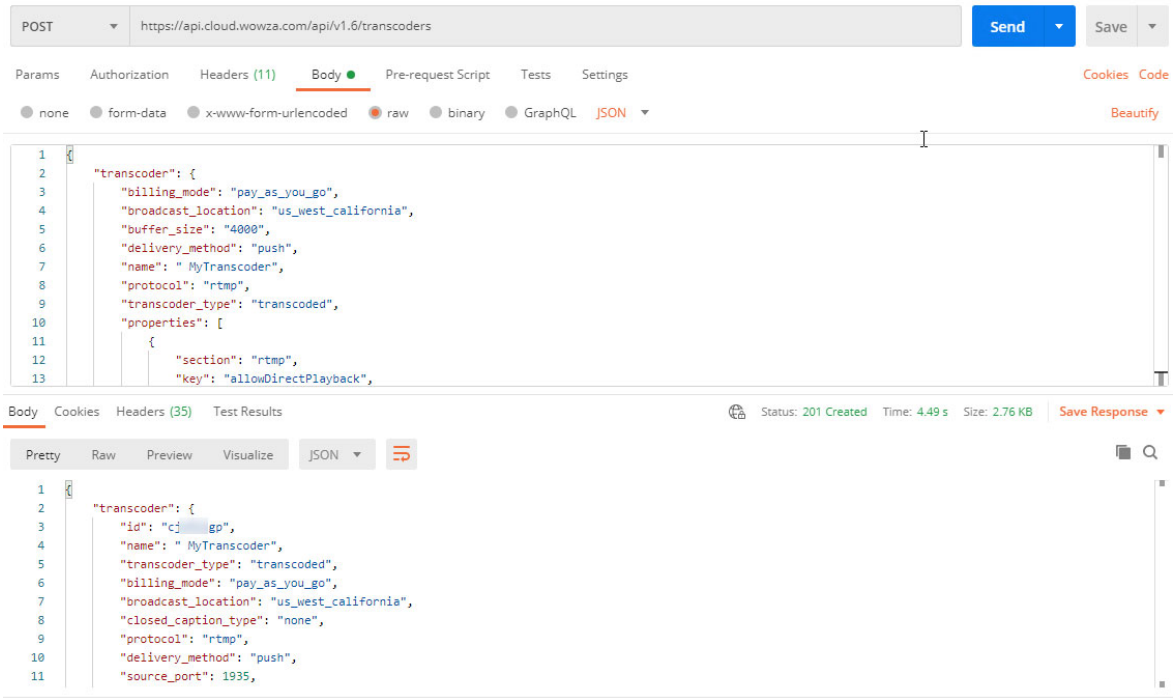
Tip: Make sure the MPEG-DASH enabled stream target is added to the transcoder you configured for stream protection.

(Optional) Block RTMP direct playback for enhanced security

Direct playback through RTMP is enabled by default, but you might want to block RTMP direct playback to ensure only devices and platforms that can decrypt your stream can access it.

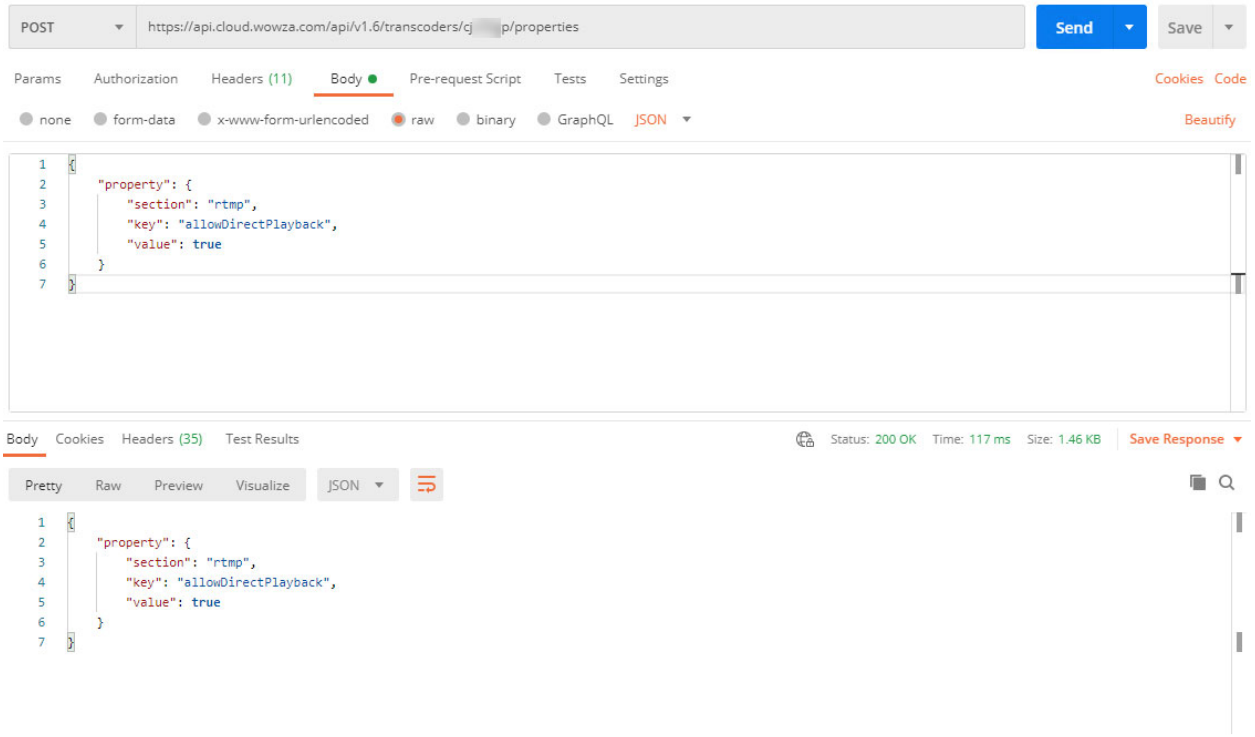
Configure RTMP playback when creating a new transcoder

```
curl -X POST \
-H "Content-Type: application/json" \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
-d '{
   "transcoder": {
     "billing_mode": "pay_as_you_go",
     "broadcast_location": "us_west_california",
     "buffer_size": "4000",
     "delivery_method": "push",
     "name": " MyTranscoder",
     "protocol": "rtmp",
     "transcoder_type": "transcoded",
     "properties": [
     {
        "section": "rtmp",
        "key": "allowDirectPlayback",
        "value": false
      }
     ]
   }
}' "${WSC_HOST}/api/${WSC_VERSION}/transcoders"
```



Configure RTMP playback on an existing transcoder

```
curl -X POST \
-H "Content-Type: application/json" \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
-d '{
     "property": {
        "section": "rtmp",
        "key": "allowDirectPlayback",
        "value": true
      }
}' "${WSC_HOST}/api/${WSC_VERSION}/transcoders/[transcoder_id]/properties"
```



Test playback with encryption

1. Start your transcoder and your stream.
2. Using the MPEG-DASH playback URL returned in the response when you enabled MPEG-DASH, verify that the stream encryption works as you expect on a player or platform that requires a PlayReady or Widevine license. To test your playback, you'll need a test player and some other information. The tips below are based on the user interface for https://demo.theoplayer.com/ezdrm-demo on Chrome (Widevine) or Internet Explorer (PlayReady):
    - **Streaming protocol** – Set to **MPEG-DASH**.
    - **Stream URL** – The URL for your protected stream.
    - **License Acquisition URL** – This URL is returned in the EZDRM response when you generated the content ID.
        - **Widevine** – The value from the ServerURL parameter. The format is `https://widevine-dash.ezdrm.com/proxy?pX=[XXXXXX]`.
        - **PlayReady** – The value from the LAURL parameter. The format is `https://playready.ezdrm.com/cency/preauth.aspx?pX=[XXXXXX]`.

Refer to the **EZDRM Universal DRM Setup and Playback** guides at www.ezdrm.com under **Resources > Documentation > EZDRM Implementation** for information about how to deliver the Widevine or PlayReady license and approve viewers, proxy URLs you'll need for playback, and sample players.

3. Stop your transcoder when your testing is complete.

More resources

- *EZDRM KeyZ API* – Refer to the ***EZDRM KeyZ API*** guide at www.ezdrm.com under **Resources > Documentation > EZDRM Implementation** for information about generating DRM keys and detailed information about responses returned in the key generation process.
- *EZDRM Testing Playback* – Refer to the ***EZDRM Testing Playback*** guide at www.ezdrm.com under **Resources > Documentation > EZDRM Implementation** for information about sample players and proxy URLs.
- For more documentation about digital rights management in Wowza Streaming Cloud please visit wowza.com/docs/wowza-streaming-cloud

Apple FairPlay Streaming

Playback of protected streams on iOS or Apple TV devices requires Apple's Fairplay DRM. You can access this DRM through our integration with EZDRM and configure stream encryption using the Wowza Streaming Cloud REST API.

You'll use your EZDRM user name and password, as well as a FairPlay asset ID, to configure your stream for DRM protection with Wowza Streaming Cloud and EZDRM FairPlay DRM.

Tip: In addition to completing the steps in this topic, you might also want to use EZDRM Universal to protect streams on Google or Microsoft devices or players.

An EZDRM key contains the **Asset ID** you'll use to configure your stream for DRM protection.

This step assumes:

- You do not already have an Asset ID. If you have one, you can skip to the *Configure the stream for DRM protection*.
- You do not want to pass an existing asset ID in the key generation request. EZDRM allows for passing existing asset IDs, but you should refer *EZDRM KeyZ API guide at* www.ezdrm.com for reasons why you'd want to and the correct syntax for the call should you choose to.

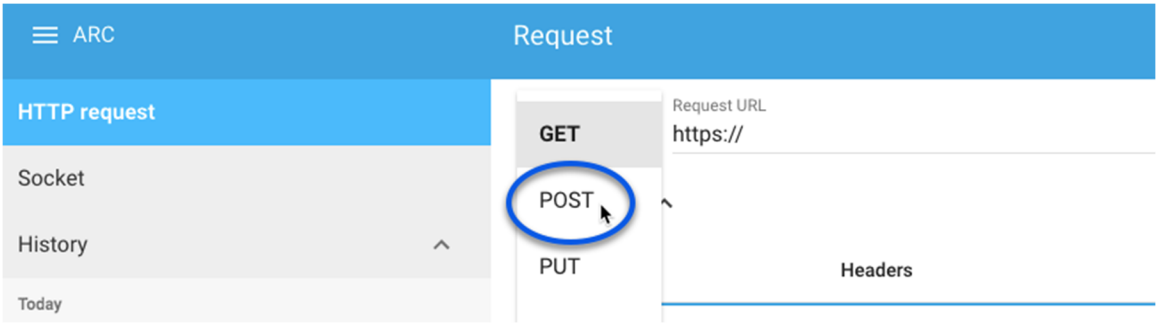
Generating Keys

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.

2. Change the Method dropdown to **POST**.

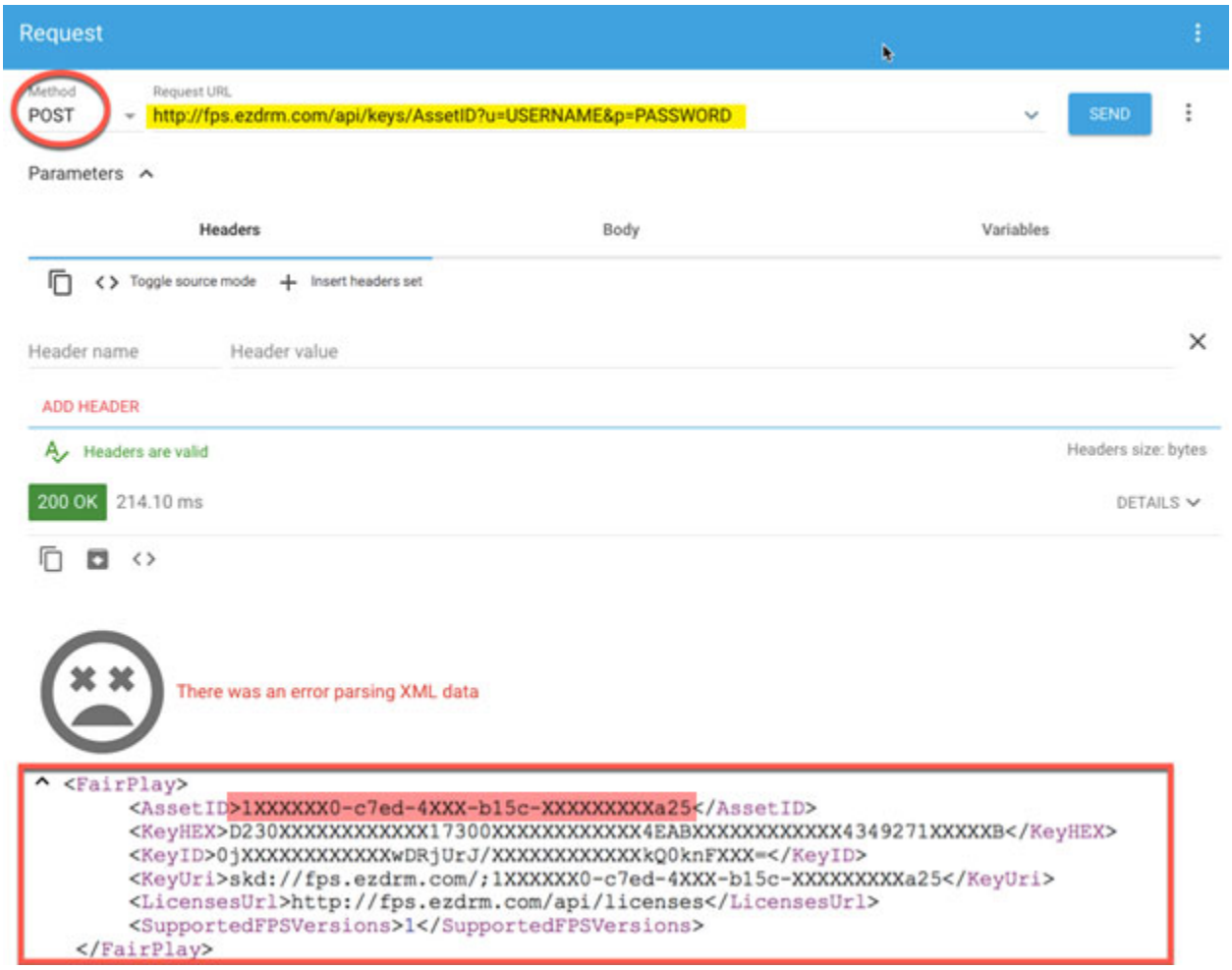


3. Enter the Request URL below updated with your username and password:

   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>
```

See ***EZDRM KeyZ API*** *guide at www.ezdrm.com under* ***Resources > Documentation > EZDRM Implementation*** *for more details on calling existing keys with* ***Asset_ID***.

Configure the stream for FairPlay DRM protection

To protect a stream using the EZDRM key you obtained in the previous step, you'll need to set the following EZDRM properties on the transcoder.

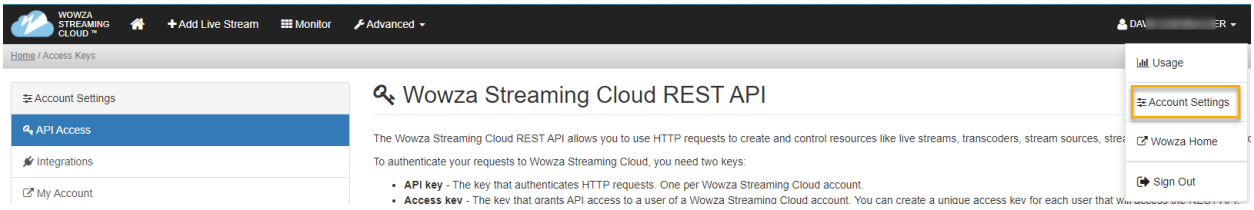
section	key	value	Description
ezdrm	username	string	Your EZDRM user name
ezdrm	password	string	Your EZDRM password
ezdrm	fairPlayAssetID	string	The FairPlay asset ID you generated from EZDRM

You can configure the EZDRM properties when you create a transcoder or by updating an existing transcoder.

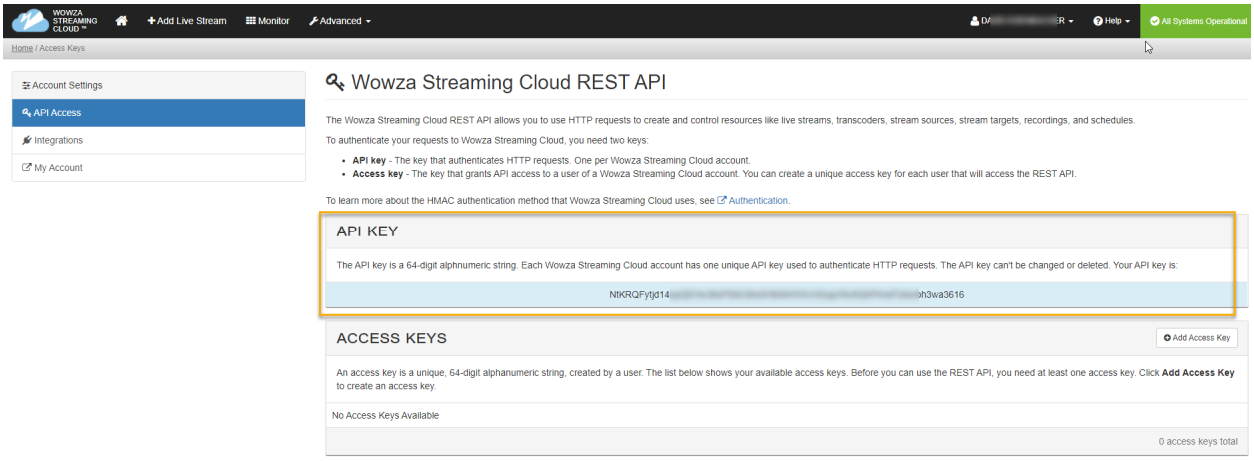
Note: Wowza Streaming Cloud does not validate the EZDRM values you specify in these properties. Make sure you enter the correct values.

WSC API Key and Access Key

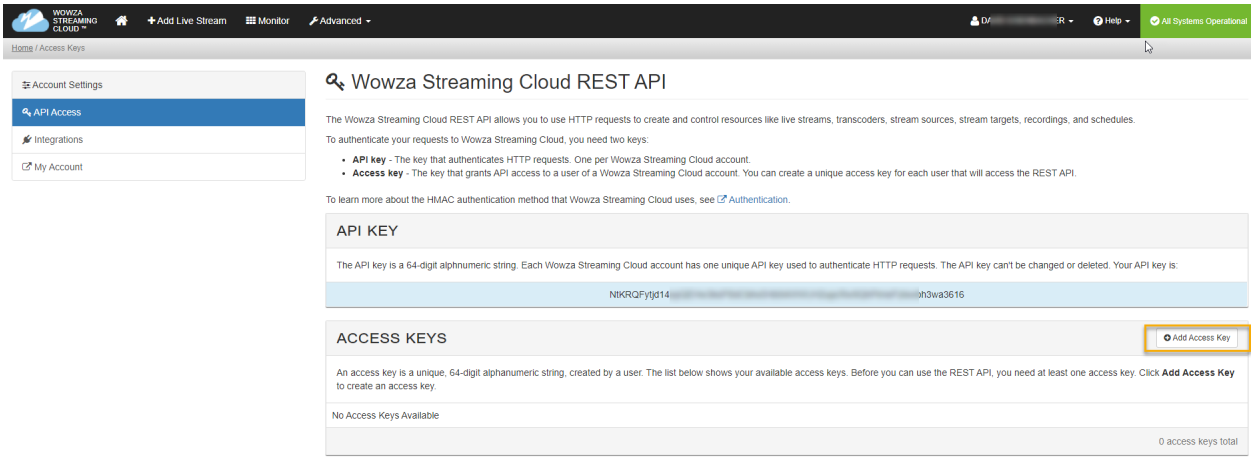
The **API key** is a 64-digit alphanumeric string. Each Wowza Streaming Cloud account has one unique API key used to authenticate HTTP requests. The API key can't be changed or deleted. To find your API Key, under your username menu, select Account Settings.



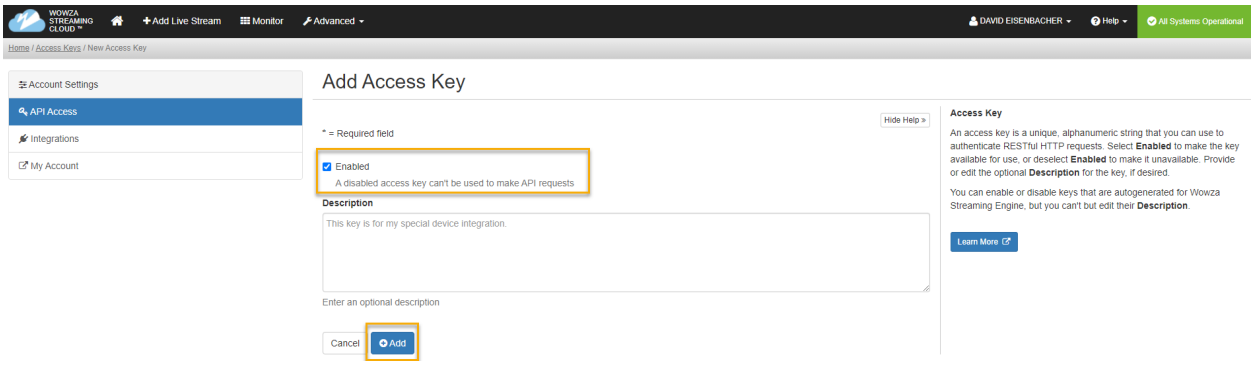
Locate your **API Key** on the API Access Page



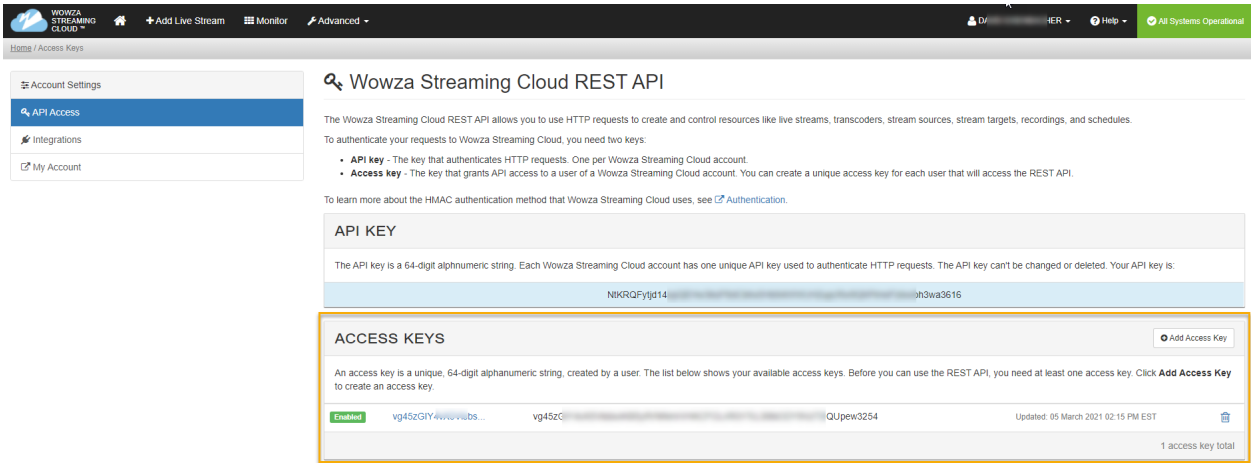
An **Access Key** is a unique, 64-digit alphanumeric string, created by a user. Before you can use the REST API, you need at least one access key. Click **Add Access Key** to create an access key.



Be sure the **Enabled** checkbox is checked (Enabled makes the key available for use). Then click **Add** to create a new Access Key.



The Access Key will now be available under API Access:

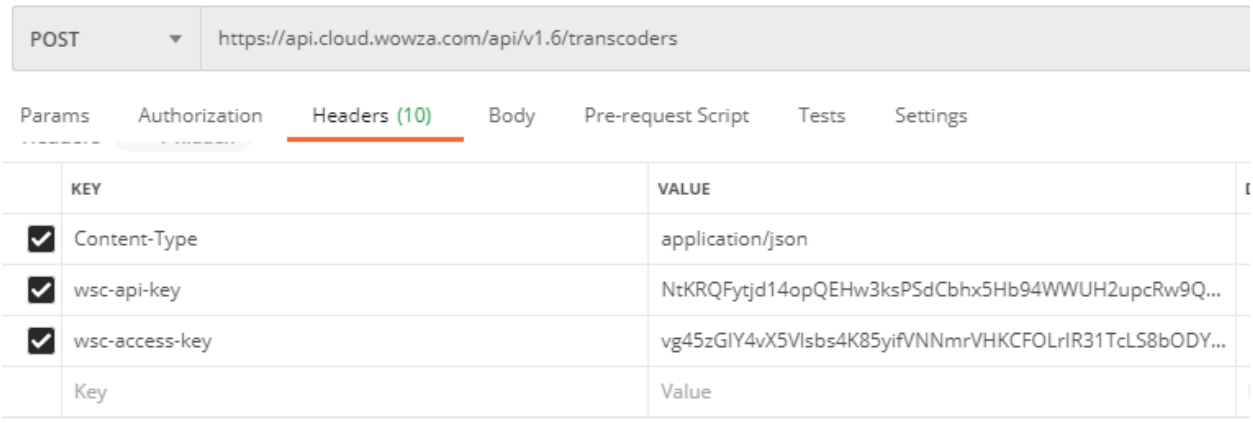


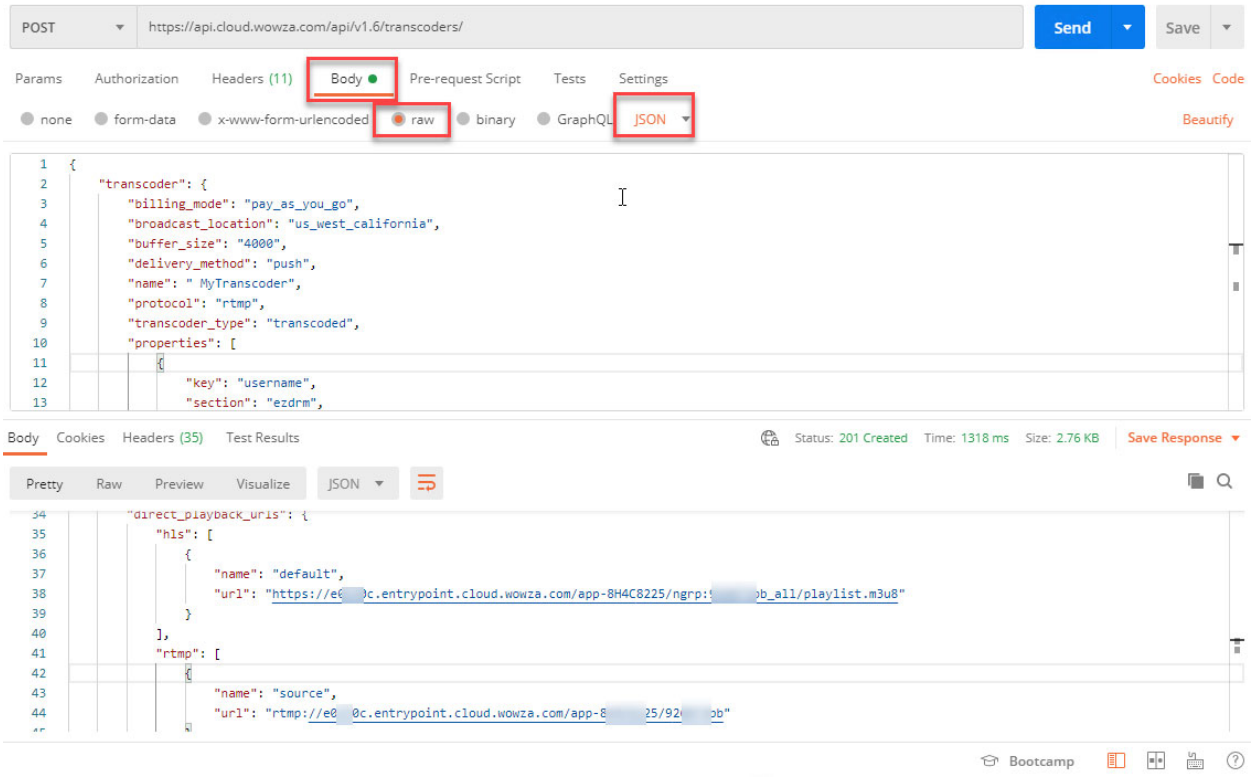
Configure FairPlay DRM when creating a new transcoder

To configure DRM when creating a new transcoder, utilize curl or emulate in Postman as shown (POST to https://api.cloud.wowza.com/api/v1.6/transcoders) and update the values as shown in the following example:

```
curl -X POST \
-H "Content-Type: application/json" \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
-d '{
   "transcoder": {
     "billing_mode": "pay_as_you_go",
     "broadcast_location": "us_west_california",
     "buffer_size": "4000",
     "delivery_method": "push",
     "name": " MyTranscoder",
     "protocol": "rtmp",
     "transcoder_type": "transcoded",
```

```
    "properties": [
    {
       "key": "username",
       "section": "ezdrm",
       "value": "your_ezdrm_username"
     },
     {
       "key": "password",
       "section": "ezdrm",
       "value": "your_ezdrm_password"
     },
     {
       "key": "fairPlayAssetId",
       "section": "ezdrm",
       "value": "[asset_id_from_ezdrm]"
     }
    ]
  }
}' "${WSC_HOST}/api/${WSC_VERSION}/transcoders"
```





Sample response

```
{
    "transcoder": {
        "id": "vcfXXXzr",
        "name": " MyTranscoder",
        "transcoder_type": "transcoded",
        "billing_mode": "pay_as_you_go",
        "broadcast_location": "us_west_california",
        "closed_caption_type": "none",
        "protocol": "rtmp",
        "delivery_method": "push",
        "source_port": 1935,
        "domain_name": "eXXX0c.entrypoint.cloud.wowza.com",
        "application_name": "app-8HXXXX25",
        "stream_name": "9dXXXXb5",
        "playback_stream_name": "9XXXX3bb",
        "delivery_protocols": [
            "rtmp",
            "rtsp",
            "wowz",
            "hls",
            "webrtc"
```

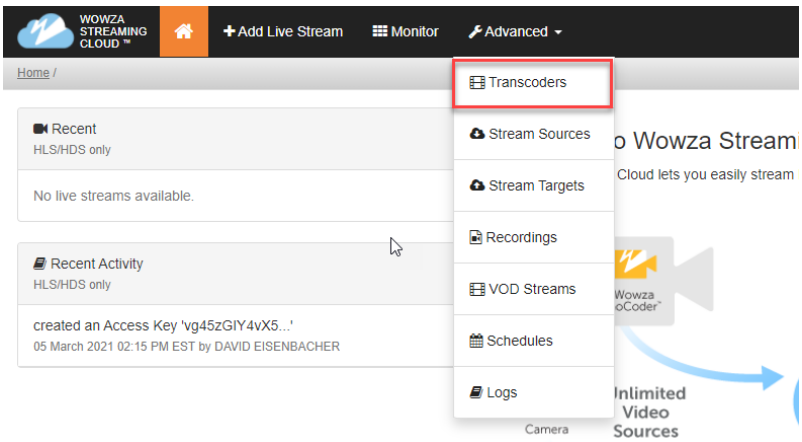
```
        ],
        "buffer_size": 4000,
        "low_latency": false,
        "stream_smoother": false,
        "idle_timeout": 1200,
        "play_maximum_connections": 10,
        "disable_authentication": false,
        "username": "clientXXXX2",
        "password": "0XXX182a",
        "watermark": false,
        "created_at": "2021-03-08T14:58:43.000Z",
        "updated_at": "2021-03-08T14:58:43.000Z",
        "direct_playback_urls": {
            "hls": [
                {
                    "name": "default",
                    "url": "https://eXXX0c.entrypoint.cloud.wowza.com/app-8HXXXX25/ngrp:9XXXX3bb_all/playl
ist.m3u8"
                }
            ],
            "rtmp": [
                {
                    "name": "source",
                    "url": "rtmp://e0230c.entrypoint.cloud.wowza.com/app-8HXXXX25/9XXXX3bb"
                }
            ],
            "rtsp": [
                {
                    "name": "source",
                    "url": "rtsp://e0230c.entrypoint.cloud.wowza.com:1935/app-8HXXXX25/9XXXX3bb"
                }
            ],
            "wowz": [
                {
                    "name": "source",
                    "url": "wowz://e0230c.entrypoint.cloud.wowza.com:1935/app-8HXXXX25/9XXXX3bb"
                }
            ],
            "webrtc": [
                {
                    "name": "source",
                    "url": "wss://e0230c.entrypoint.cloud.wowza.com/webrtc-session.json",
                    "application_name": "app-8HXXXX25",
                    "stream_name": "9XXXX3bb"
```

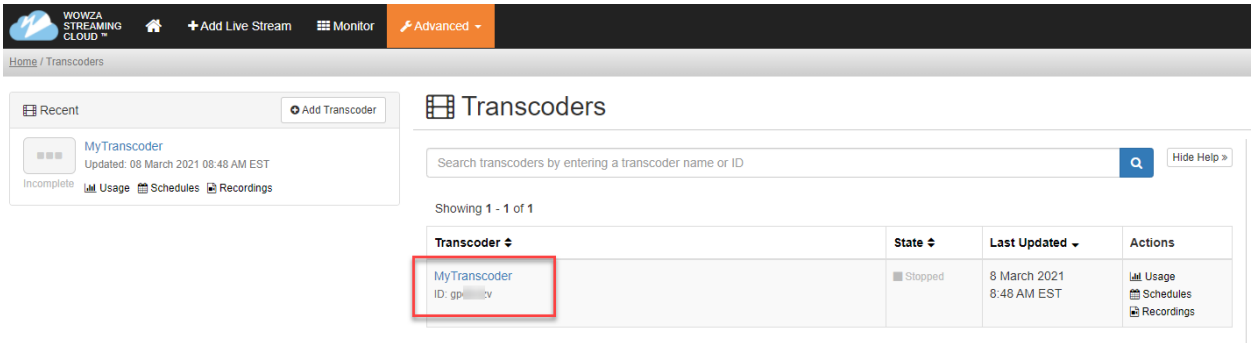
```
                }
            ]
        },
        "outputs": []
    }
}
```

Configure DRM on an existing transcoder

To configure DRM on an existing transcoder, utilize curl or emulate in Postman as shown (PATCH to https://api.cloud.wowza.com/api/v1.6/transcoders/*[transcoder_id]*) and update the values as shown in the following example.

Find your Transcoder IDs under the Advanced menu / Transcoders:





```
curl -X PATCH \
-H "Content-Type: application/json" \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
-d '{
    "transcoder": {
      "properties": [
        {
          "key": "username",
          "section": "ezdrm",
          "value": "your_ezdrm_username"         },
        {
          "key": "password",
          "section": "ezdrm",
          "value": "your_ezdrm_password"
        },
        {
          "key": "fairPlayAssetId",
          "section": "ezdrm",
          "value": "[asset_id_from_ezdrm]"
        }
     ]
   }
}' "${WSC_HOST}/api/${WSC_VERSION}/transcoders/[transcoder_id]"
```

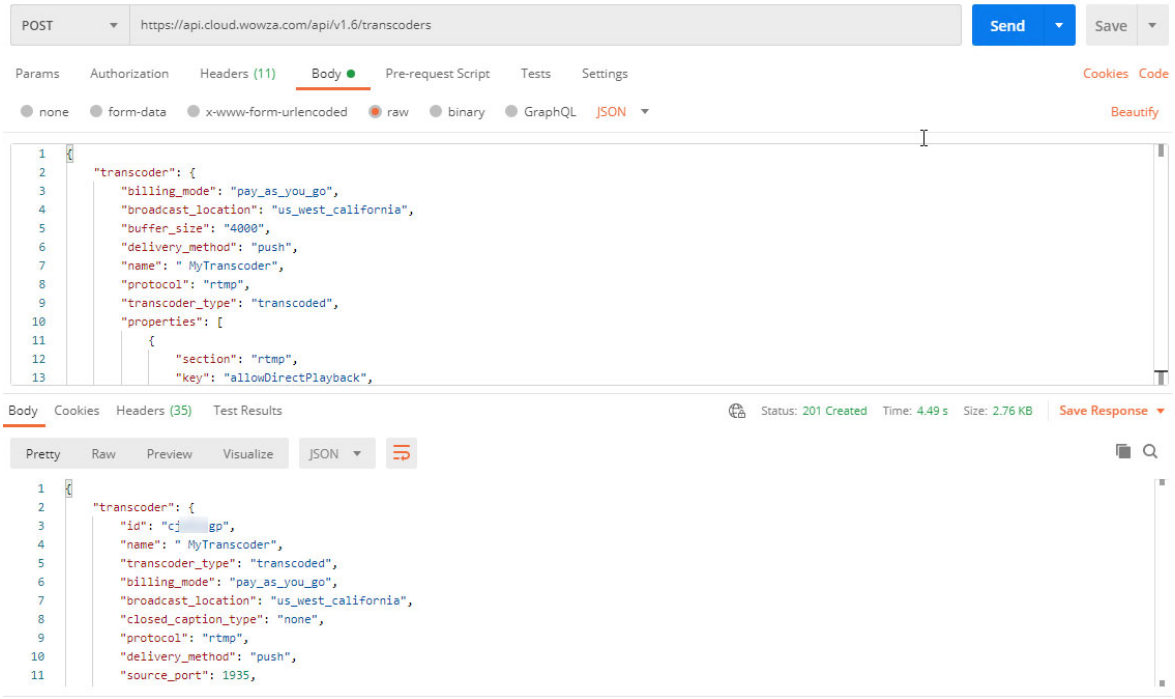
(Optional) Block RTMP direct playback for enhanced security

Direct playback through RTMP is enabled by default, but you might want to block RTMP direct playback to ensure only devices and platforms that can decrypt your stream can access it.

Configure RTMP playback when creating a new transcoder

```
curl -X POST \
-H "Content-Type: application/json" \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
-d '{
   "transcoder": {
     "billing_mode": "pay_as_you_go",
     "broadcast_location": "us_west_california",
```

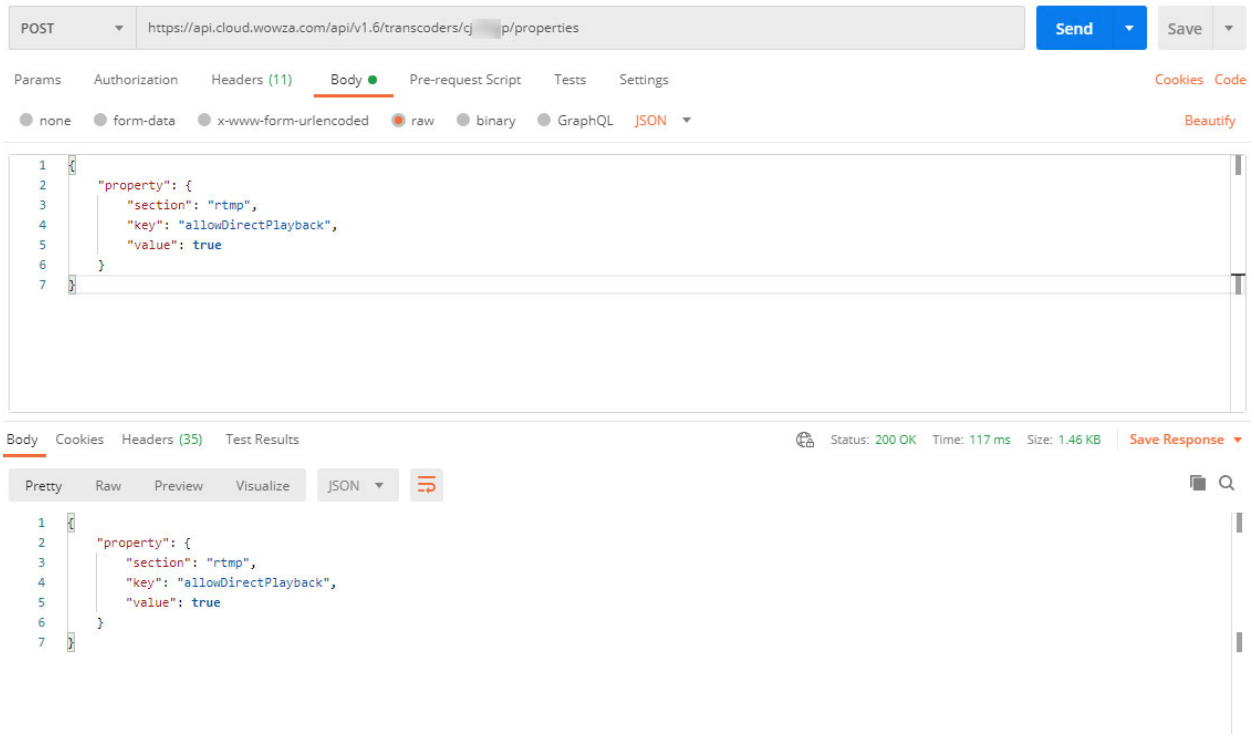
```
    "buffer_size": "4000",
    "delivery_method": "push",
    "name": " MyTranscoder",
    "protocol": "rtmp",
    "transcoder_type": "transcoded",
    "properties": [
     {
       "section": "rtmp",
       "key": "allowDirectPlayback",
       "value": false
     }
    ]
  }
}' "${WSC_HOST}/api/${WSC_VERSION}/transcoders"
```



Configure RTMP playback on an existing transcoder

```
curl -X POST \
-H "Content-Type: application/json" \
-H "wsc-api-key: ${WSC_API_KEY}" \
-H "wsc-access-key: ${WSC_ACCESS_KEY}" \
```

```
-d '{
     "property": {
        "section": "rtmp",
        "key": "allowDirectPlayback",
        "value": true
      }
}' "${WSC_HOST}/api/${WSC_VERSION}/transcoders/[transcoder_id]/properties"
```



Test playback with encryption

1. Start your transcoder and your stream.
2. Verify that the stream encryption works as you expect on an Apple device or service. To test your playback, you'll need a test player and some other information. The tips below are based on the user interface for https://developer-tools.jwplayer.com/stream-tester/ in Safari set to **Fairplay**:
    - **File URL** – The URL for your protected stream.
    - **Certificate URL** – Part of onboarding for your EZDRM account with Fairplay DRM. You'll need to generate a Fairplay certificate through an Apple Developer account and then post the resulting certificate on a

publically accessible endpoint. The URL to this endpoint is the value for this field.

- **Process SPC URL** – Build this URL from the EZDRM response when you generated the asset ID. The format is `https://[LicensesUrl]/[AssetID]`.

Refer to the ***EZDRM Apple FairPlay DRM Setup*** and ***EZDRM Testing Playback*** guides at www.ezdrm.com under **Resources > Documentation > EZDRM Implementation** for information about how to deliver the Fairplay license and approve viewers, proxy URLs you'll need for playback, and sample players.

3. Stop your transcoder when your testing is complete.

More resources

- *EZDRM KeyZ API* – Refer to the ***EZDRM KeyZ API*** guide at www.ezdrm.com under **Resources > Documentation > EZDRM Implementation** for information about generating DRM keys and detailed information about responses returned in the key generation process.
- *EZDRM Testing Playback* – Refer to the ***EZDRM Testing Playback*** guide at www.ezdrm.com under **Resources > Documentation > EZDRM Implementation** for information about sample players and proxy URLs.
- For more documentation about digital rights management in Wowza Streaming Cloud please visit https://wowza.com/docs/wowza-streaming-cloud

Additional Information

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