EZDRM Packaging
Dolby Hybrik

Table of Contents

Version 1.0 / June 17, 2021

Introduction

The following document outlines the steps necessary to successfully encrypt and package a file with EZDRM using Dolby Hybrik.

The following are required for packaging:

1. Dolby Hybrik account
2. Amazon S3 or Google Cloud Account
3. CURL / Postman knowledge (examples utilize Postman)
4. EZDRM account

For more details on the Hybrik API visit
https://docs.hybrik.com/api/v1/HybrikAPI.html?#getting-started

Here's the link to download the sample JSONs:
https://www.ezdrm.com/downloads/hybrik/hybrik-sample-fairplay.zip

Generating Keys - Widevine/PlayReady

Below are the steps to create the DRM Keys for encryption for Widevine and PlayReady.

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://cpix.ezdrm.com/keygenerator/cpix.aspx?**k=kid**&**u=username**&**p=password**&**c=hyb-001**

   The parameters are as follows:

Parameter	Description
k	kid or Key ID value (client generated) in GUID format*
u	EZDRM username
p	EZDRM password
c	Content ID – generic resource name/identifier (client generated) – passed into **id** field

* To generate a GUID for the k value, you can use a GUID generator like the one found here: http://guid-convert.appspot.com.

Key Value Definitions

Here are the descriptions of the key values returned by EZDRM:

- **id** – c value returned, generic resource name/identifier (client generated)
- **kid** – Key ID in GUID format (client generated)*
- **pskc:Secret key**– the Secret Content Encryption Key in Base 64 generated by EZDRM and returned as a plain value
- **ContentProtectionData (PlayReady)** – The modular specific protection system specific header (PSSH) data for the encryption process; Base 64 encoded.

  *NOTE: Decode from Base 64 to Text format to get the **playready_pssh** value*

* To generate a GUID for the k value, you can use a GUID generator like the one found here: http://guid-convert.appspot.com.

Here Is the example XML return:

```
<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc" id="hyb-001">
<cpix:ContentKeyList>
<cpix:ContentKey kid="582aXXXX-XXXX-41ef-XXX-ae533ccaXXXX" explicitIV="WCq8DAXXXXXXE65TPMpe+A==">
<cpix:Data>
<pskc:Secret>
<pskc:PlainValue>sU+XXXXXXqvSN0aXbXXXXX==</pskc:PlainValue>
</pskc:Secret>
</cpix:Data>
```

```
</cpix:ContentKey>
</cpix:ContentKeyList>
<cpix:DRMSystemList>
<cpix:DRMSystem kid="b99ed9e5-c641-49d1-bfa8-43692b686ddb" systemId="edef8ba9-79d6-4ace-a3c8-27dcd51d21ed"
>
<cpix:PSSH>AAAAd3Bzc2gAAAAA7e+XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
idVo3WjVjWkJTZEdcL3FFTnBLMmh0Mnc9PSIsInRyYWNrcyI6WyJTRCJdfSoCU0Q=</cpix:PSSH>
</cpix:DRMSystem>
<cpix:DRMSystem kid="b99XXXXX-c6XX-XXXX-bfa8-XXXX2b686ddb" systemId="9a04XXXX-XX40-XXXX-ab92-XXXXe0885f95"
>
<cpix:PSSH>AAADBnBzc2gAAAAAmgTweZhAQoarkuZb4IhflQAAAubmAgAAAQABANwCPABXAFIATQBIAEUAQQBEAEUAUgAgAHgAbQBsAG4
AcwA9ACIAaAB0AHQAcAA6AC8ALwBzAGMAaABlAG0AYQBzAC4AbQBpAGMAcgBvAHMAbwBmAHQALgBjAG8AbQAvAEQAUgBNAC8AMgAwADAAN
wAvADAAMwAvAFAAbABhAHkAUgBlAGEAZAB5AEgAZQBhAGQAZQByACIAIAB2AGUAcgBzAGkAbwBuAD0AIgA0AC4AMAAuADAALgAwACIAPgA
8AEQAQQBUAEEAPgA8AFAAUgBPAFQARQBDAFQASQBOAEYATwA+ADwASwBFAFkATABFAE4APgAxADYAPAAvAEsARQBZAEwARQBOAD4APABBA
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
ABzADoALwAvAHAAbABhAHkAcgBlAGEAZAB5AC4AZQB6AGQAcgBtAC4AYwBvAG0ALwBjAGUAbgBjAHkALwBwAHIAZQBhAHUAdABoAC4AYQB
zAHAAeAA/AHAAWAA9AEUAMAAxADgAMwBGADwALwBMAEEAXwBVAFIATAA+ADwARABTAF8ASQBEAD4AVgBsAFIANwBJAGQAcwBJAEoARQB1A
FIAZAAwADYATABhAHEAcwAyAGoAdwA9AD0APAAvAEQAUwBfAEkARAA+ADwAQwBIAEUAQwBLAFMAVQBNAD4AdABQADYASwB3ADcAaQB1AFE
AYgA4AD0APAAvAEMASABFAEMASwBTAFUATQA+ADwALwBEAEEAVABBAD4APAAvAFcAUgBNAEgARQBBAEQARQBSAD4A</cpix:PSSH>
<cpix:ContentProtectionData>PG1zcHI6cHJvPjVnSUFBQUVBQVFEY0Fqd0FWd0JTQUUwQVNBQkZBRUVBUkFCRkFGSUFJQUI0QUcwQW
JBQnVBSE1BUFFBaUFHZ0FkQUIwQUhBQU9nQXZBQzhBY3dCakFHZ0FaUUJ0QUdFQWN3QXVBRzBBYVFCakFISUFid0J6QUc4QVpnQjBBQzRB
WXdCdkFHMEFMd0JFQUZJQVRRQXZBRElBTUFBd0FEY0FMd0F3QURNQUx3QlFBR3dBWVFCNUFGSUFaUUJoQUdRQWVRQklBR1VBWVFCa0FHVU
FjZ0FpQUNBQWRnQmxBSElBY3dCcEFHOEFiZ0E5QUNJQU5BQXVBREFBTGdBd0FDNEFNQUFpQUQ0QVBBQkVBRUVBVkFCQkFENEFQQUJRQUZJ
QVR3QlVBRVVBUXdCVUFFa0FUZ0JHQUU4QVBnQXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXQVBBQk1BRUVBWHdCVkFGSUFUQUErQUdnQWRBQjBBSEFBY3dBNkFDOEFMd
0J3QUd3QVlRQjVBSElBWlFCaEFHUUFlUUF1QUdVQWVnQmtBSElBYlFBdUFHTUFid0J0QUM4QVl3QmxBRzRBWXdCNUFDOEFjQUJ5QUdVQVl
RQjFBSFFBYUFBdUFHRUFjd0J3QUhnQVB3QndBRmdBUFFCRkFEQUFNUUE0QURNQVJnQThBQzhBVEFCQkFGOEFWUUJTQUV3QVBnQThBRVFBV
XdCZkFFa0FSQUErQUZZQWJBQlNBRGNBU1FCa0FITUFTUUJLQUVVQWRRQlNBR1FBTUFBMkFFd0FZUUJ4QUhNQU1nQnFBSGNBUFFBOUFEd0F
Md0JFQUZNQVh3QkpBRVFBUGdBOEFFFTUFTQUJGQUVNQVN3QlRBRlVBVFFBK0FIUUFVQUEyQUVzQWR3QTNBR2tBZFFCUkFHSUFPQUE5QUR3Q
Ux3QkRBRWdBUlFCREFFc0FVd0JWQUUwQVBnQThBQzhBUkFCQkFGUUFRUUErQUR3QUx3QlhBRklBVFFCSUFFVUFRUUJFQUVVQVVnOStBQT0
9PC9tc3ByOnBybz4=</cpix:ContentProtectionData>
</cpix:DRMSystem>
<cpix:DRMSystem kid=" b99XXXXX-c6XX-XXXX-bfa8-XXXX2b686ddb" systemId="9a04XXXX-XX40-XXXX-ab92-XXXXe0885f95
”>
<cpix:URIExtXKey>c2tkOi8XXXXXXXXXXXXXXXXXS87Yjk5ZWQ5ZTUtYzY0MS00OWQxLWJmYTgtNDM2OTJiNjg2ZGRi</cpix:URIExt
XKey>
</cpix:DRMSystem>
</cpix:DRMSystemList>
</cpix:CPIX>
```

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.

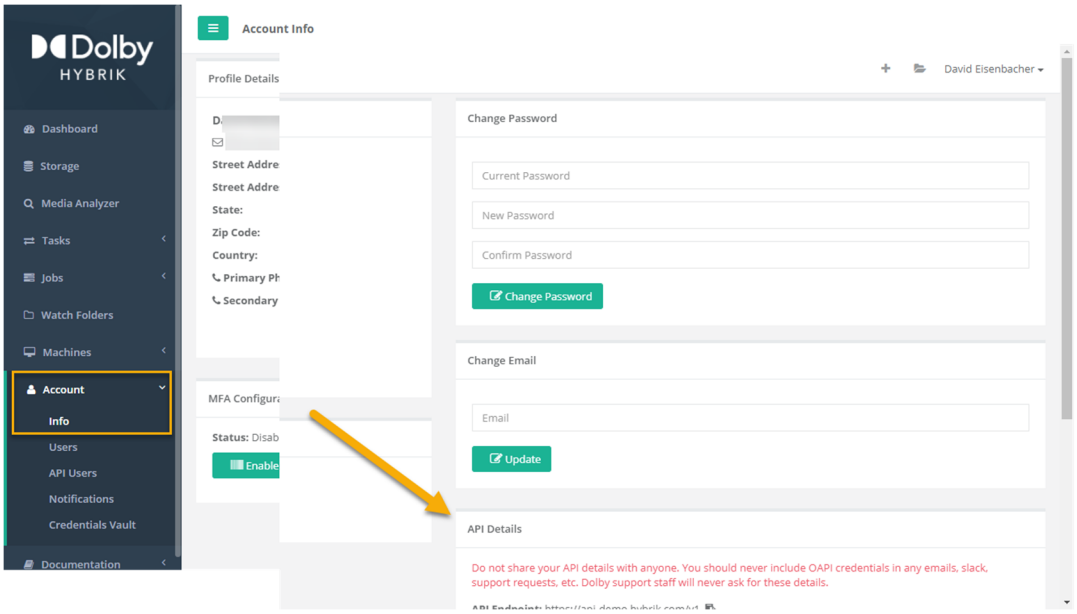
```
curl -v https://cpix.ezdrm.com/keygenerator/cpix.aspx?k=kid&u=username&p=password&c=hyb-001
```

* To generate a GUID for the k value, you can use a GUID generator like the one found here: http://guid-convert.appspot.com.

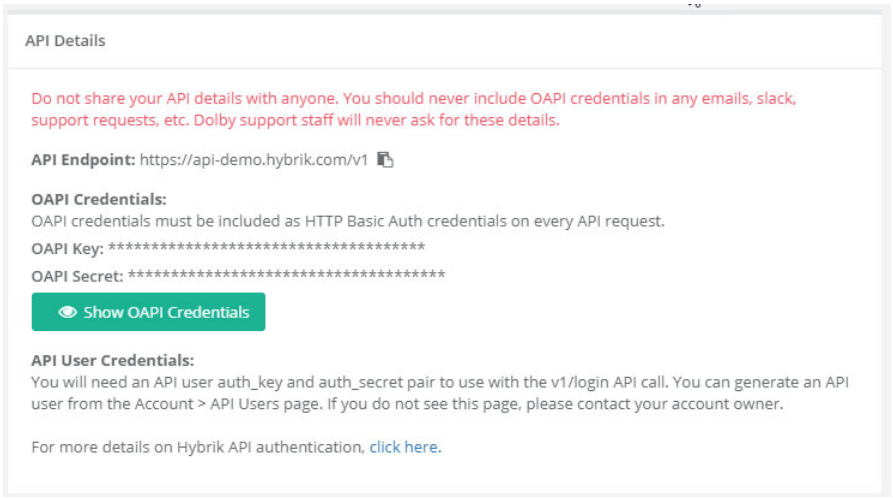
DRM for Widevine/PlayReady CENC

Hybrik API Keys

When logged into Dolby Hybrik, find **API Details** under the **Account/Info** menu.

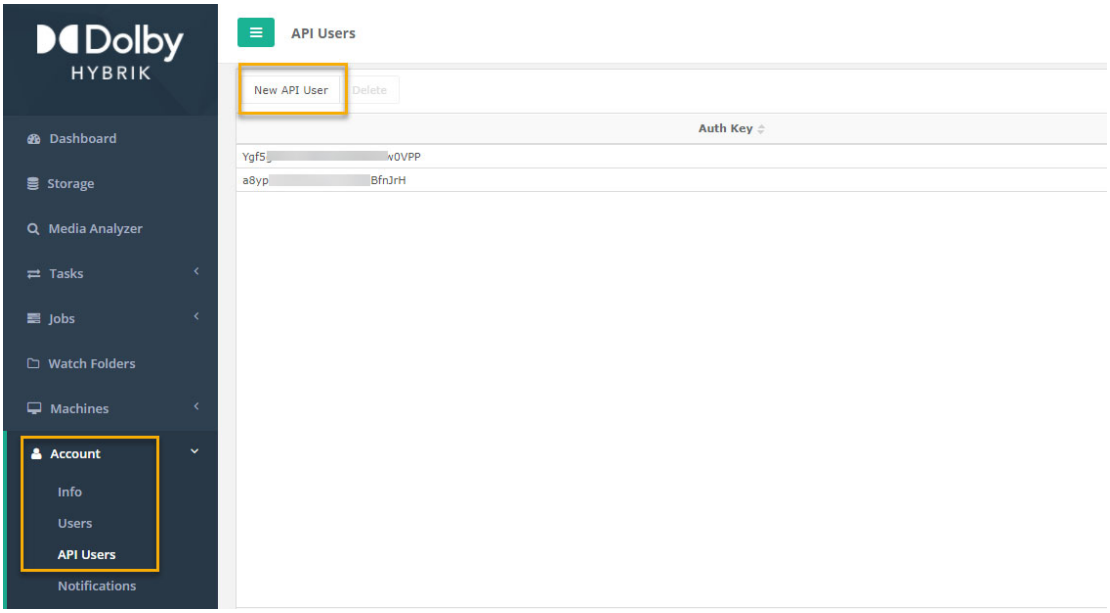


Capture the **OAPI Key** and **OAPI Secret Key**. OAPI credentials must be included as HTTP Basic Auth credentials on every API request.



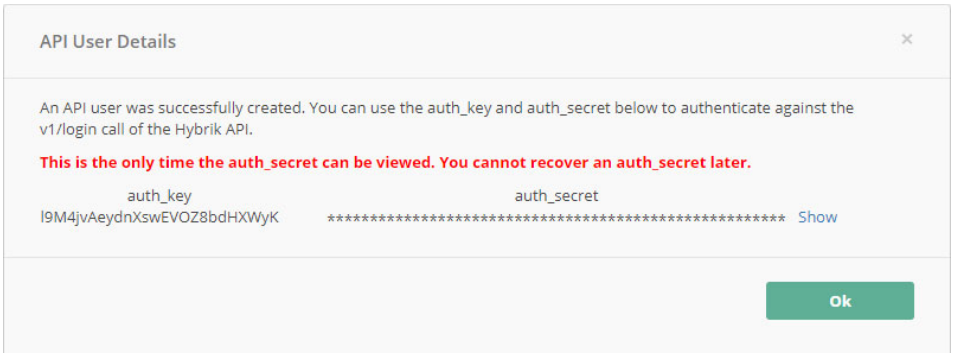
API Users

To create a New API User, under **Account/API Users** click the **New API User** button.



Click **OK** to confirm creation of the new user.

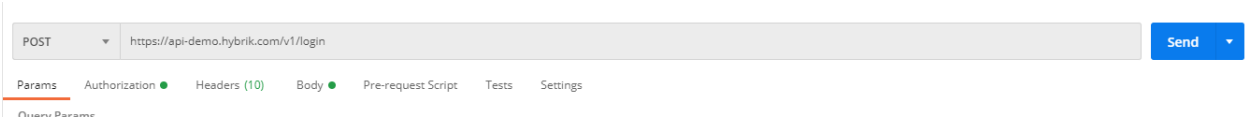
API User Details will be provided. The **auth_key** and **auth_secret** will be used to authenticate against the v1/login call o the Hyrbik API. Save your **auth_secret** key (see note below).



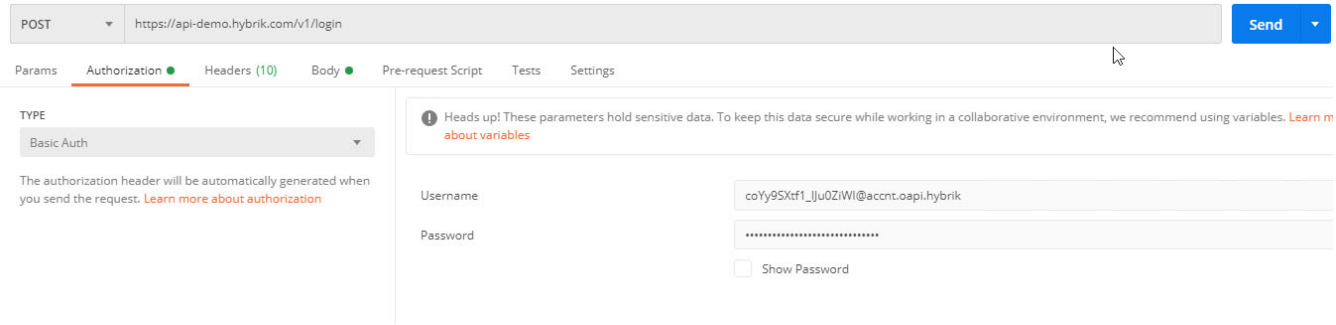
Note: The auth_secret key can only be viewed at this time – be sure to capture and save the auth_secret. You cannot recover an auth_secret later.

POST Request in Postman - Security Token

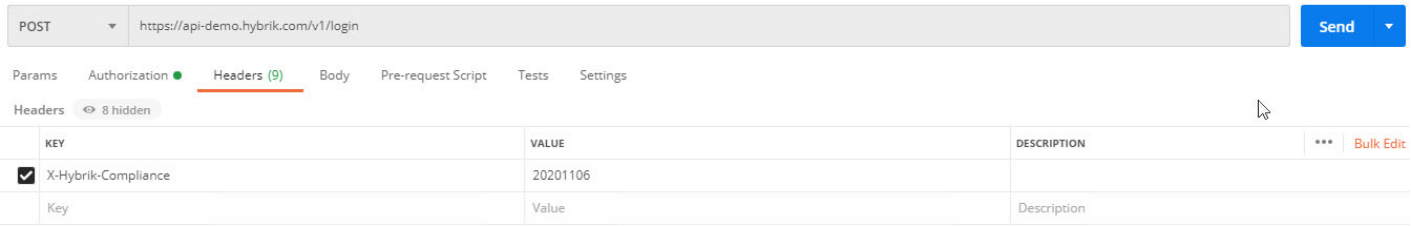
1. In Postman, create **POST** request to the API https://api-demo.hybrik.com/v1/login



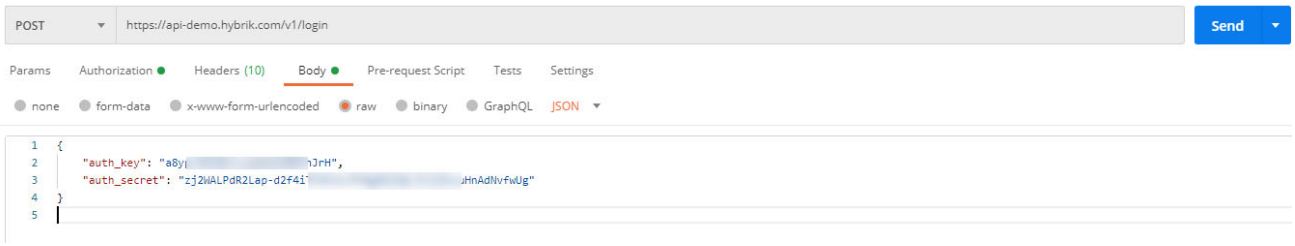
2. Under **Authorization**, select Type: **Basic Auth** and enter your **Hybrik OAPI Key (Username)** and **Hybrik OAPI Secret Key (Password)**.



3. Under **Headers**, add the following Header:
   - **X-Hybrik-Compliance** – enter value: current date (YYYMMDD format)

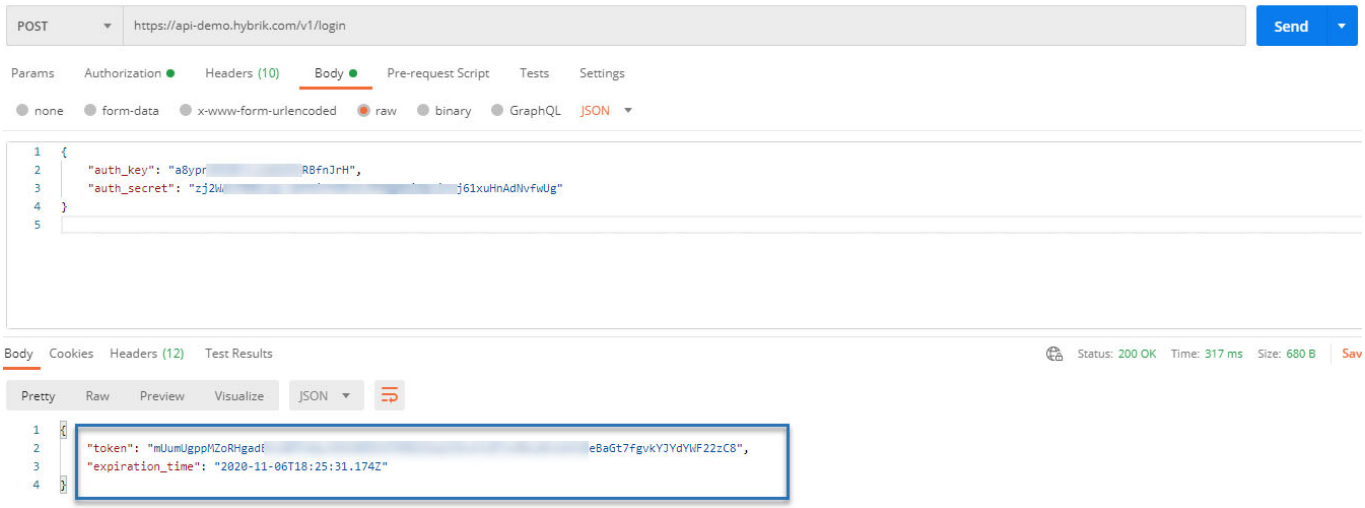


4. Build the Body of the request:
    - Click “Body” Tab and select **raw** and the format is **JSON**.
    - Enter the following parameters:
        - **auth_key**
        - **auth_secret**



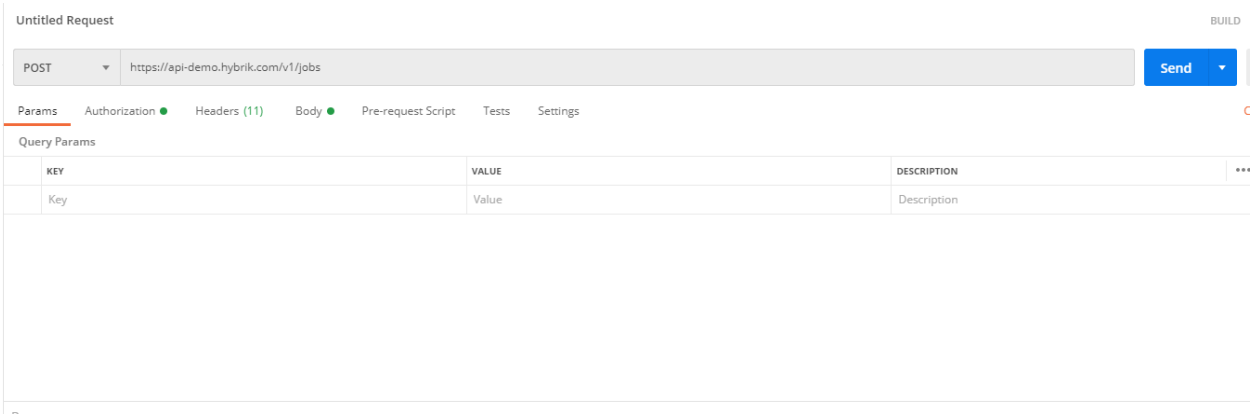

```
{
        “auth_key”: “8yprXXXXXXXXXXX9jRBfXXXX”
        “auth_secret”: “zj2XXXXXXXXX-d2f4iXXXXXXXXXXXXXXXXXXXXj61xuHnAdNXXXXg”
}
```

5. Click **SEND**. The expected response is the security **Token** that confirms login to the Hybrik API. The token will expire by the **expiration_time** if actions are not taken against the API within that timeframe. Every new action will extend the token.

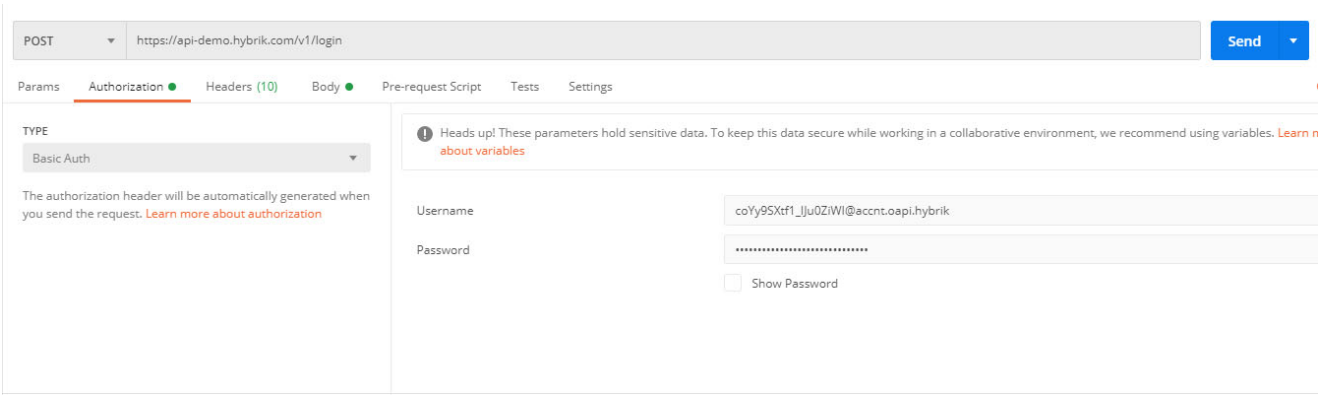


POST Request in Postman - Send Job and Tasks

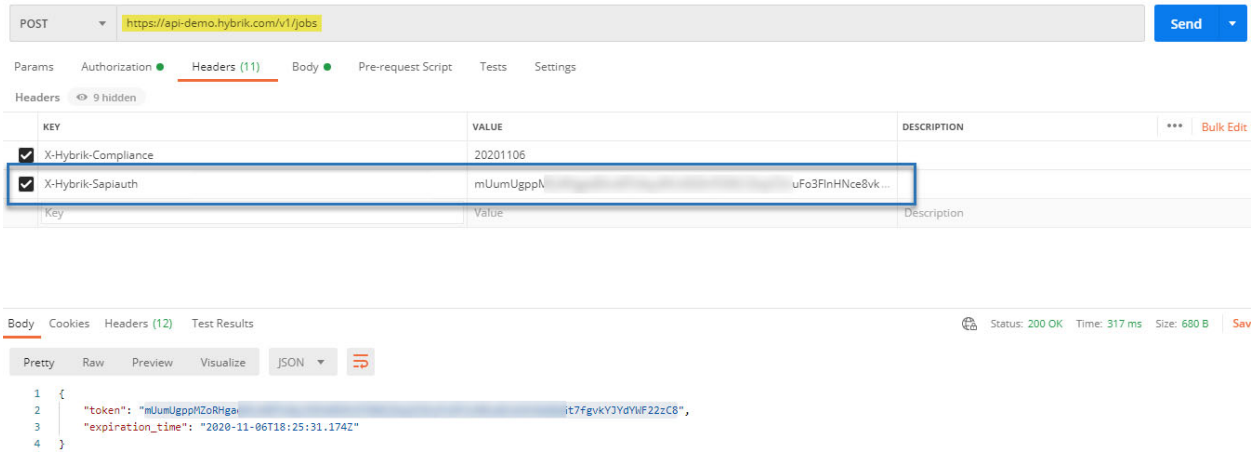
1. In Postman, create **POST** request to the API https://api-demo.hybrik.com/v1/jobs



2. Under **Authorization**, select Type: **Basic Auth** and enter your **Hybrik OAPI Key (Username)** and **Hybrik OAPI Secret Key (Password)**.

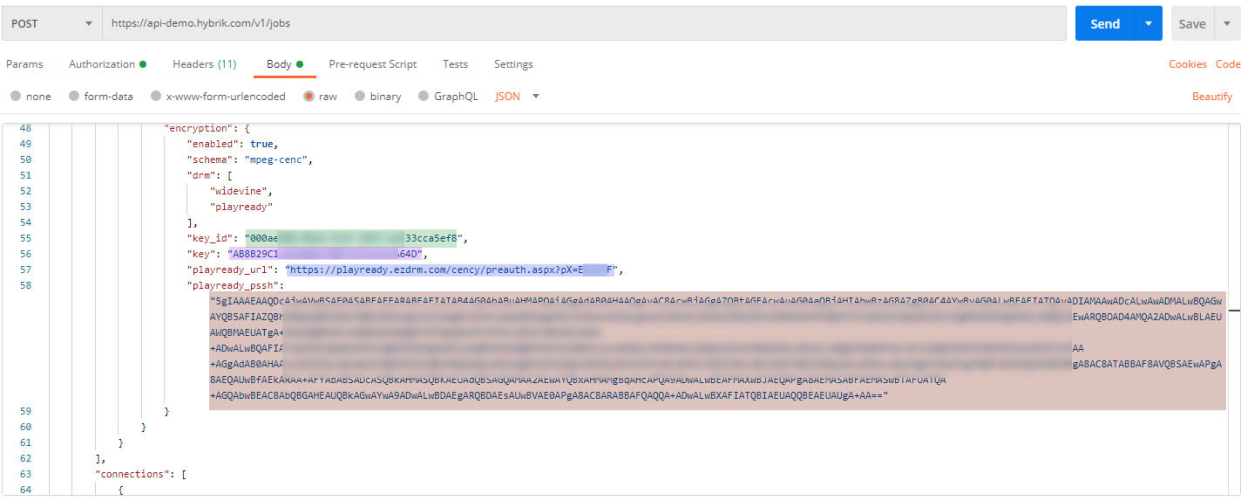


3. Under **Headers**, add the following Header:
   - **X-Hybrik-Compliance** – enter value: current date (YYYMMDD format)
   - **X-Hybrik-Sapiauth** – enter the security **Token** from **Step 5** in the last section



4. Build the Body of the request:
    - Click “Body” Tab and select **raw** and the format is **JSON**.

5. Enter the following parameters:



- **input**: location of the file to be encoded (S3 bucket, etc. where the .mp4 can be downloaded) * if utilzing S3, locations should be set to Public

- **output**: location of the encoded output file to be encoded (S3 bucket, etc.) * if utilzing S3, locations should be set to Public, and CORS Access-Control-Allow-Origin

- **key_id**: Content Key **kid** in GUID format (client generated)

- **key**: use the **pskc:Secret key** value and decode from Plain Value Base 64 to HEX format in lowercase. An example decoder can be found at: https://tomeko.net/online_tools/base64.php?lang=en

pskc:Secret key (Base 64) = 2rEsXXXXXXXXXXAh9PHCLpQ==

key (HEX) = DAB1XXXXF06FXXXXED4XXXXD3C708BA5

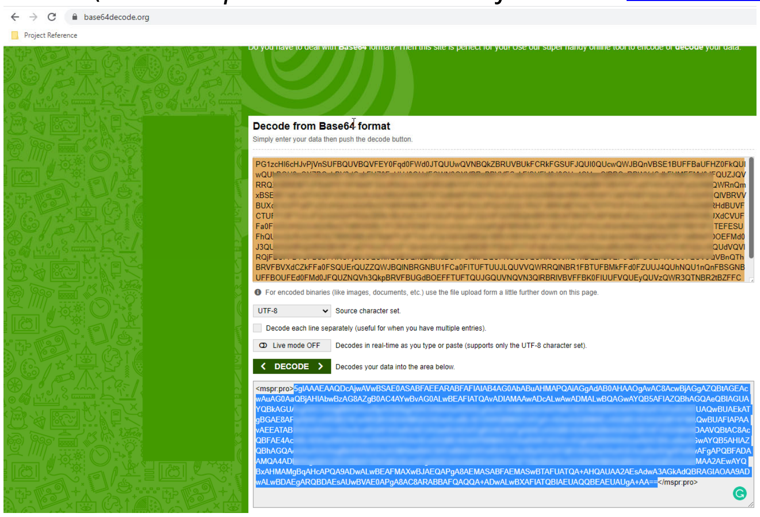
- **playready_url**: provided by EZDRM including your **PX Value**.

  https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXX

  Note: Your PlayReady **PX value** is the last six characters of your PlayReady Profile ID. The appropriate one is required for all packagers you use. For more details on how to find your PX value visit: https://www.ezdrm.com/Documentation/EZDRM_Testing_Playback_v2.pdf

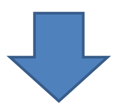
- **ContentProtectionData (PlayReady)** – The modular specific protection system specific header (PSSH) data for the encryption process; Base 64 encoded.

***NOTE: Decode from Base 64 to Text format to get the playready_pssh value** (An example decoder can be found at www.base64decode.org):*



ContentProtectionData (Base 64) =

PG1zcHI6cHJvPjVnSUFBQUVBQVFEY0Fqd0FWd0JTQUUwQVNBQkZBRUVBUkFCRkFGSUFJQUI0QUcwQWJBQnVBSE1BUFFBaUFHZ0FkQUIwQUhBQU9nQXZBQzhBY3dCakFHZ0FaUUJ0QUdFQWN3QXVBRzBB YVFCakFISUFid0J6QUc4QVpnQjBBQzRBWXdCdkFHMEFMd0JFQUZJQVRRQXZBREIBTUFBd0FEY0FMd0F 3QURNQUx3QlFBR3dBWVFCNUFGSUFaUUJoQUdRQWVRQklBR1VBWVFCa0FHVUFjZ0FpQUNBQWRnQ mxBSElBY3dCcEFHOEFiZ0E5QUNJQU5BQXVBREFBTGdBd0FDNEFNQUFpQUQ0QVBBQkVBRUVBVkFCQ kFENEFQQUJRQUZJQVR3QlVBRVVBUXdCUUFFa0FUZ0JHQUU4QVBnQXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXQVBBQk1BRUVBWHdCVkFG SUFUQUErQUdnQWRBQjBBSEFBY3dBNkFDOEFMd0J3QUd3QVlRQjVBSElBWlFCaEFHUUFlUUF1QUdVQ WVnQmtBSElBYlFBdUFHTUFid0J0QUM4QVl3QmxBRzRBWXdCNUFDOEFjQUJ5QUdVQVlRQjFBSFFBYU FBdUFHRUFjd0J3QUhnQVB3QndBRmdBUFFCRkFEQUFNUUE0QURNQVJnQThBQzhBVEFCQkFGOEFWU UJTQUV3QVBnQThBRVFBVXdCZkFFa0FFSQUErQUZZQWJBQlNBRGNBU1FCa0FITUFUUJLQUVVQWRRQ lNBR1FBTUFBMkFFd0FZUUJ4QUhNQU1nQnFBSGNBUUFBBOUFEd0FMd0JFQUZNQVh3QkpBR1FBUGdBdG OEFFTUFTQUJGQUVNQVN3QlRBRlVBVFFBK0FIUUFUR0E1QUdVQUVEQVZBQUlJBVFFBK0FIUUFUR0 QUR3QUx3QkRBRWdBUlFCREFEc0FVd0JVAE0FVd0JWQUUwQVBnQThBQzhBRUFBQUFQQUlBRUFDBUFWBQzhBRQAUEVFBQUFQKkRGFEQUUEUIUFVFRUEJFQUVVQVVnQStBQT09PC9tc3ByOnBybz4=



playready_pssh (Text) =

5gIAAAEAAQDcAjwAVwBSAE0ASABFAEEARABFAFIAIAB4AG0AbABuAHMAPQAiAGgAdAB0AHAAOgAv AC8AcwBjAGgAZQBtAGEAcwAuAG0AaQBjAHIAbwBzAG8AZgB0AC4AYwBvAG0ALwBEAFIATQAvADIA MAAwADcALwAwADMALwBQAGwAYQB5AFIAZQBhAGQAeQBIAGUAYQBkAGUAcgAiACAAdgBlAHIAc wBpAG8AbgA9ACIANAAuADAALgAwAC4AMAAiAD4APABEAEEAVABBAD4APABQAFIATwBUAEUAQwB UAEkATgBGAE8APgA8AEsARQBZAEwARQBOAD4AMQA2ADwALwBLAEUAWQBMAEUATgA+ADwAQQB MAEcASQBEAD4AQQBFAFMAQwBUAFIAPAAvAEEATABHAEkARAA+ADwALwBQAFIATwBUAEUAQwBU AEkATgBGAE8APgA8AEsASQBEAD4ANQBkAG0AZQB1AFUASABHADAAVQBtAC8AcQBFAE4AcABLADIA aAB0ADIAdwA9AD0APAAvAEsASQBEAD4APAXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

AEwAPgA8AEQAUwBfAEkARAA+AFYAbABSADcASQBkAHMASQBKAEUAdQBSAGQAMAA2AEwAYQBxAHMAMgBqAHcAPQA9ADwALwBEAFMAXwBJAEQAPgA8AEMASABFAEMASwBTAFUATQA+AHQAUAA2AEsAdwA3AGkAdQBRAGIAOAA9ADwALwBDAEgARQBDAEsAUwBVAE0APgA8AC8ARABBBAFQAQQA+ADwALwBXAFIATQBIAEUAQQBEAEUAUgA+AA==

Note: *Widevine PSSH is not necessary*

```
{
    "name": "pr/wv-003",
    "payload": {
        "elements": [
            {
                "uid": "source_file",
                "kind": "source",
                "payload": {
                    "kind": "asset_url",
                    "payload": {
                        "storage_provider": "s3",
                        "url": "s3://ezdrm-sample-content/BigBuckBunny_320x180.mp4" // S3  source location
                    }
                }
            },
            {
                "uid": "dash_encrypted_packaging",
                "kind": "package",
                "payload": {
                    "location": {
                        "storage_provider": "s3",
                        "path": "s3://ezdrm-demos/hybrik/output/both003", //output path
                        "attributes": [
                            {
                                "name": "ContentType",
                                "value": "application/dash+xml"
                            }
                        ]
                    },
                    "file_pattern": "manifest.mpd", //output type
                    "kind": "dash", //document type
                    "uid": "main_manifest",
                    "force_original_media": false,
                    "media_location": {
                        "storage_provider": "s3",
                        "path": "s3://ezdrm-demos/hybrik/output/both003", //output path
```

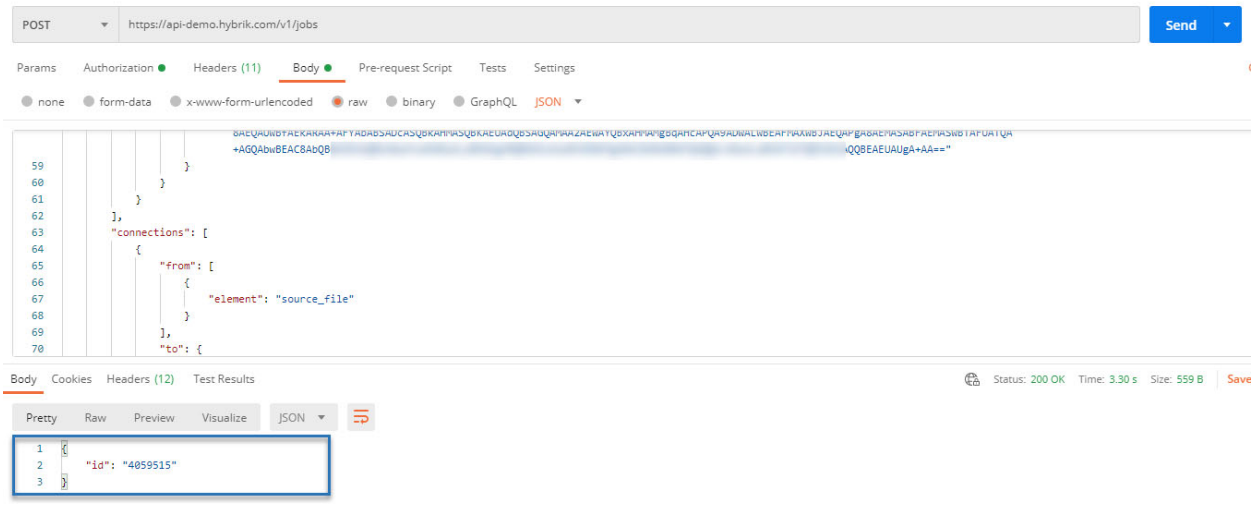
```
                    "attributes": [
                        {
                            "name": "ContentType",
                            "value": "video/mp4"
                        }
                    ]
                },
                "dash": {
                    "title": "000-both003",
                    "info_url": "www.ezdrm.com"
                },
                "encryption": {
                    "enabled": true,
                    "schema": "mpeg-cenc",
                    "drm": [ //drm type
                        "widevine",
                        "playready"
                    ],
                      "key_id": "b99edXXXX6414XXXXfa8XXXX2b686ddb",
                       "key": "65fecXXXX9eccXXXXeaXXXXX308f6549",
                       "playready_url": "https://playready.ezdrm.com/cency/preauth.aspx?pX=E0
XXXF",
                       "playready_pssh": "5gIAAAEAAQDcAjwAVwBSAE0ASABFAEEARABFAFIAIAB4AG0AbAB
uAHMAPQAiAGgAdAB0AHAAOgAvAC8AcwBjAGgAZQBtAGEAcwAuAG0AaQBjAHIAbwBzAG8AZgB0AC4AYwBvAG0ALwBEAFIAT
QAvADIAMAAwADcALwAwADMALwBQAGwAYQB5AFIAZQBhAGQAeQBIAGUAYQBkAGUAcgAiACAAdgBlAHIAcwBpAG8AbgA9ACI
ANAAuADAALgAwAC4AMAAiAD4APABXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXAEsARQBZAEwARQBOAD
4AMQA2ADwALwBLAEUAWQBMAEUATgA+ADwAQQBMAEcASQBEAD4AQQBFAFMAQwBUAFIAPAAvAEEATABHAEkARAA+ADwALwBQ
AFIATwBUAEUAQwBUAEkATgBGAE8APgA8AEsASQBEAD4ANQBkAG0AZQB1AFUASABHADAAVQBtAC8AcQBFAE4AcABLADIAaA
B0ADIAdwA9AD0APAAvAEsASQBEAD4APABMAEEAXwBVAFIATAA+AGgAdAB0AHAAcwA6AC8ALwBwAGwAYQB5AHIAZQBhAGQA
eQAuAGUAegBkAHIAbQAuAGMAbwBtAC8AYwBlAG4AYwB5AC8AcAByAGUAYQB1AHQAaAAuAGEAcwBwAHgAPwBwAFgAPQBFAD
AAMQA4ADMARgA8AC8ATABBAF8AVQBSAEwAPgA8AEQAUwBfAEkARAA+AFYAbABSADcASQBkAHMASQBKAEUAdQBSAGQAMAA2
AEwAYQBxAHMAMgBqAHcAPQA9ADwALwBEAFMAXwBJAEQAPgA8AEMASABFAEMASwBTAFUATQA+AHQAUAA2AEsAdwA3AGkAdQ
BRAGIAOAA9ADwALwBDAEgARQBDAEsAUwBVAE0APgA8AC8ARABBAFQAQQA+ADwALwBXAFIATQBIAEUAQQBEAEUAUgA+AA==
",
                    }
                }
            }
        ],
        "connections": [
            {
                "from": [
                    {
                        "element": "source_file"
```

```
                    }
                ],
                "to": {
                    "success": [
                        {
                            "element": "dash_encrypted_packaging"
                        }
                    ]
                }
            }
        ]
    }
}
```

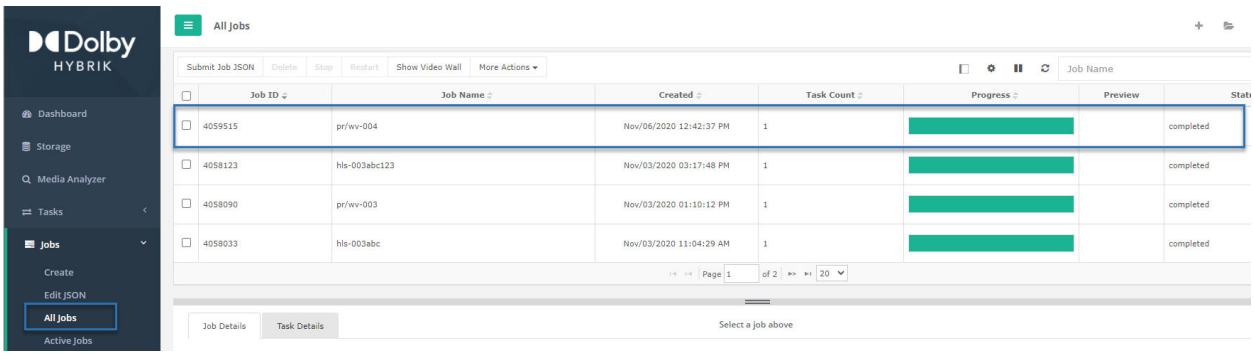
Send the Post.

Send Job

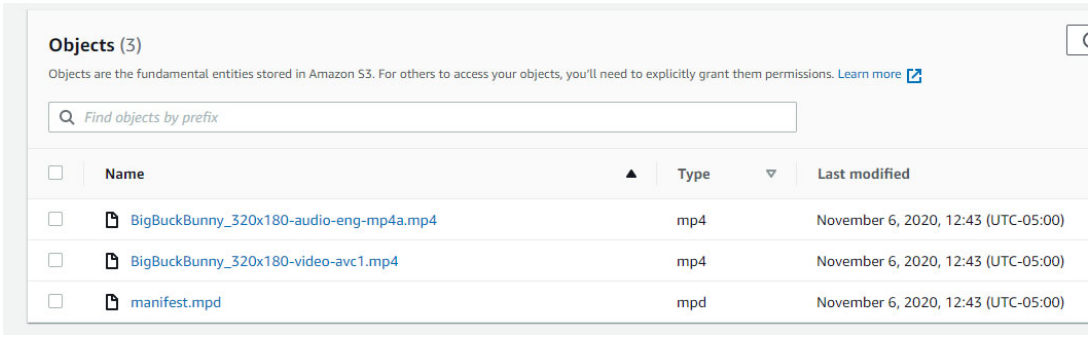
The return will include the job **id** – use this value to verify the status of the job in Hyrbik.



To verify the job in Hybrik, click on **Jobs** menu, then **All Jobs**. The **id** return will show the status of the job.



The output files will be created in the specified file location.



Testing Playback

When testing playback, use the **Widevine License Acquisition URL** that includes:

1. The base URL with your account **PX value**
2. **Key ID**

The base URL is: https://widevine-dash.ezdrm.com/widevine-php/widevine-foreignkey.php?**pX=XXXXXX**&kid=5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXXb

For more details testing playback, visit:
https://www.ezdrm.com/Documentation/EZDRM_Testing_Playback_v2.pdf

Generating Keys - Apple FairPlay HLS

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://cpix.ezdrm.com/keygenerator/cpix.aspx?**k=kid**&**u=username**&**p=password**&**c=may2020-**

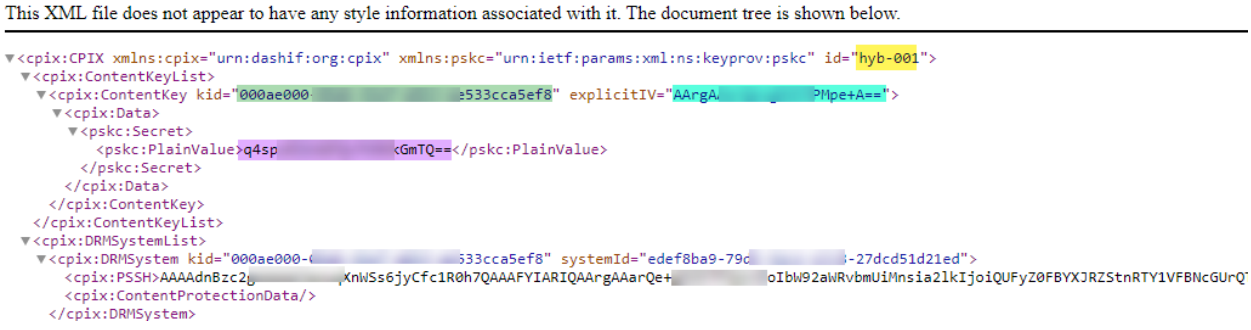
   The parameters are as follows:

Parameter	Description
k	**kid** or Key ID value (client generated) in GUID format*
u	EZDRM username
p	EZDRM password
c	Content ID – generic resource name/identifier (client generated) – passed into **id** field

* To generate a GUID for the k value, you can use a GUID generator like the one found here: http://guid-convert.appspot.com.

Key Value Definitions

Here are the descriptions of the key values returned by EZDRM:



- **id** – c value returned, generic resource name/identifier (client generated)
- **kid** – Key ID in GUID format (client generated)*
- **pskc: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

* To generate a GUID for the k value, you can use a GUID generator like the one found here: http://guid-convert.appspot.com.

Here Is the example XML return:

```
<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc" id="hyb-001">
<cpix:ContentKeyList>
<cpix:ContentKey kid="000aXXXX-06ab-XXXX-a013-ae533ccXXXX8" explicitIV="AArgXXXrQe+gEXXXXXpe+A==">
<cpix:Data>
<pskc:Secret>
<pskc:PlainValue>g4sXXXXXAGFQvFKRKXXXTQ==</pskc:PlainValue>
</pskc:Secret>
</cpix:Data>
</cpix:ContentKey>
</cpix:ContentKeyList>
```

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.

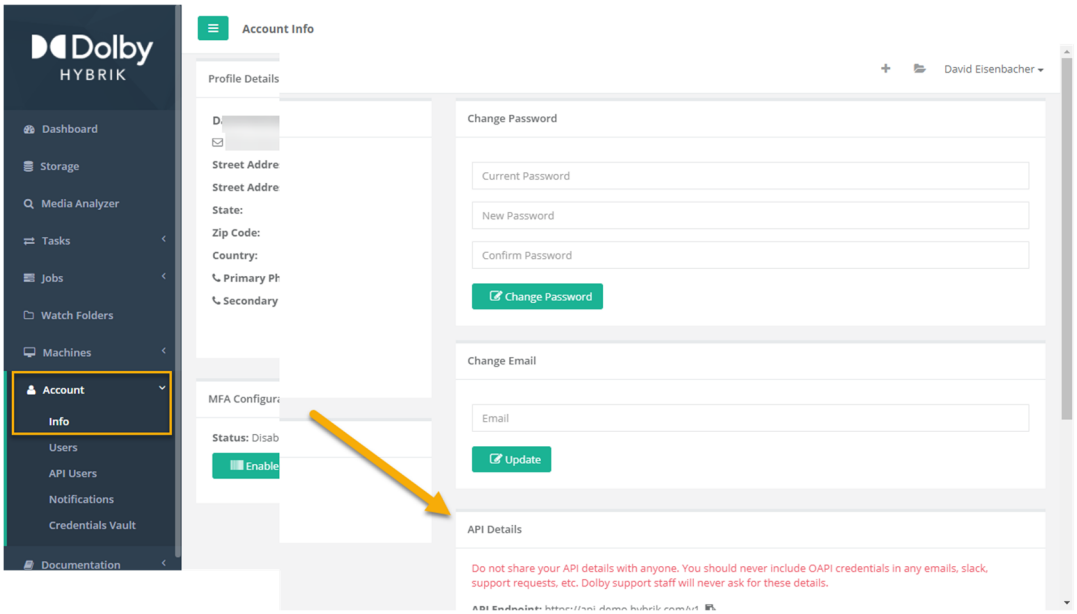
```
curl -v https://cpix.ezdrm.com/keygenerator/cpix.aspx?k=kid&u=username&p=password&c=may2020-
```

* To generate a GUID for the k value, you can use a GUID generator like the one found here: http://guid-convert.appspot.com.

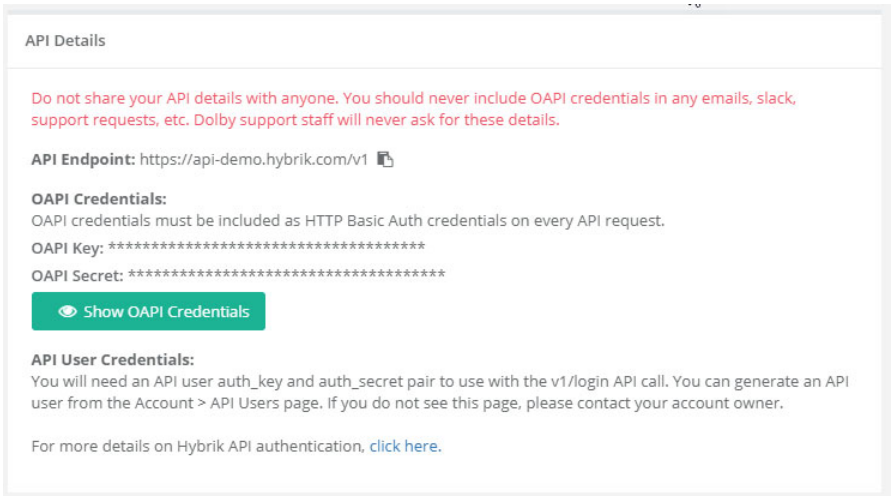
Apple FairPlay DRM for HLS

Hybrik API Keys

When logged into Dolby Hybrik, find **API Details** under the **Account/Info** menu.

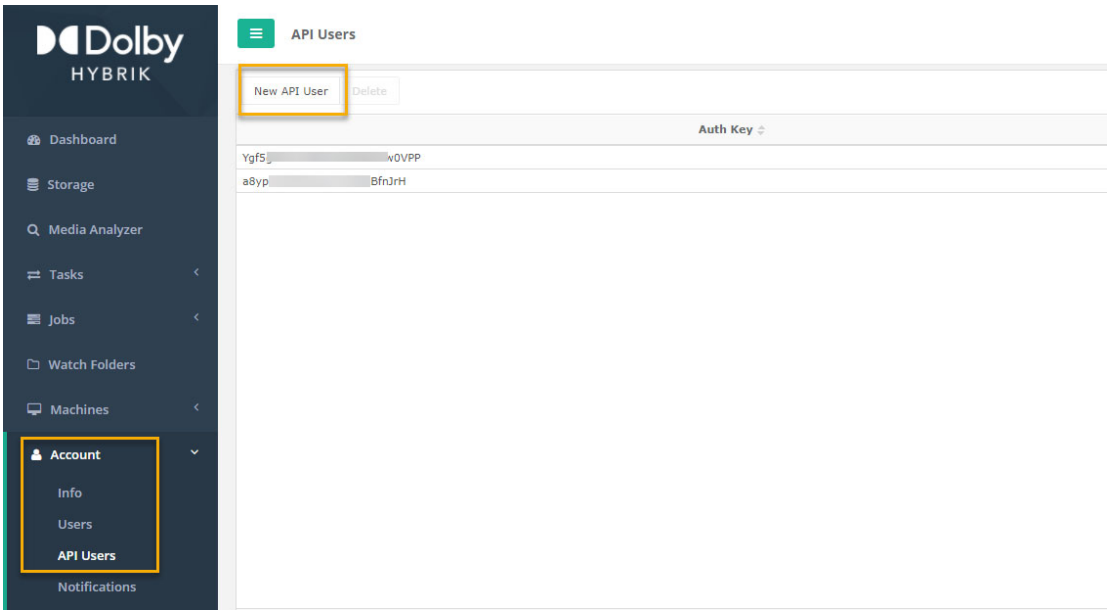


Capture the **OAPI Key** and **OAPI Secret Key**. OAPI credentials must be included as HTTP Basic Auth credentials on every API request.



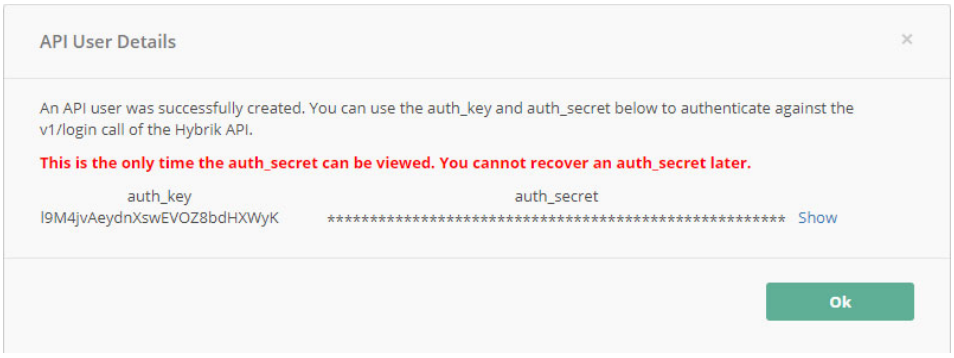
API Users

To create a New API User, under **Account/API Users** click the **New API User** button.



Click **OK** to confirm creation of the new user.

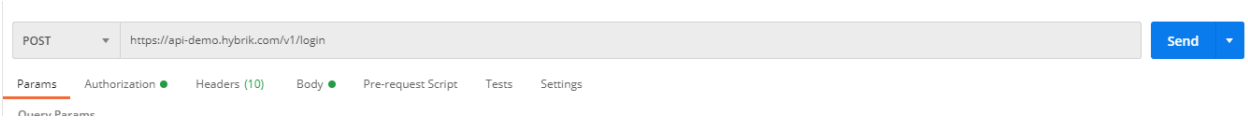
API User Details will be provided. The **auth_key** and **auth_secret** will be used to authenticate against the v1/login call o the Hyrbik API. Save your **auth_secret** key (see note below).



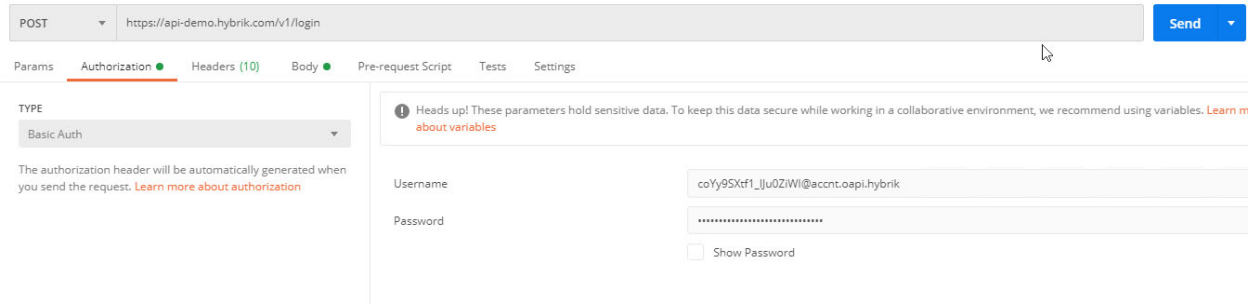
Note: The auth_secret key can only be viewed at this time – be sure to capture and save the auth_secret. You cannot recover an auth_secret later.

POST Request in Postman - Security Token

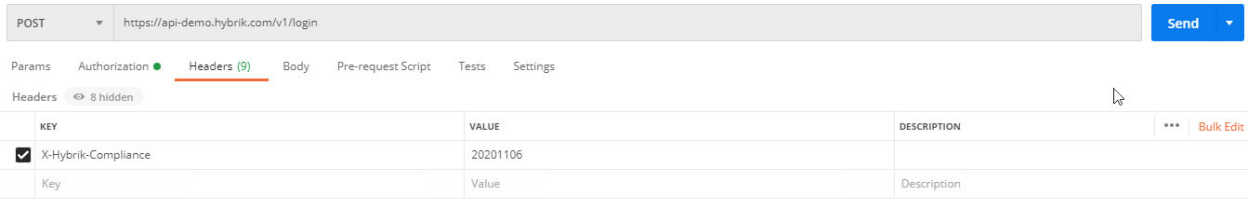
1. In Postman, create **POST** request to the API https://api-demo.hybrik.com/v1/login



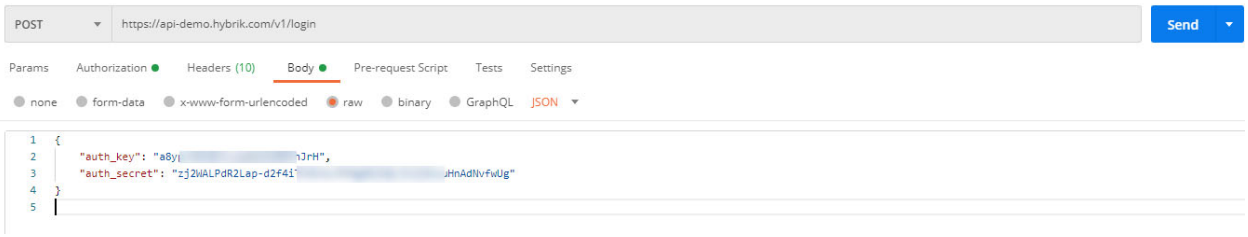
2. Under **Authorization**, select Type: **Basic Auth** and enter your **Hybrik OAPI Key (Username)** and **Hybrik OAPI Secret Key** (**Password**).



3. Under **Headers**, add the following Header:
   - **X-Hybrik-Compliance** – enter value: current date (YYYMMDD format)

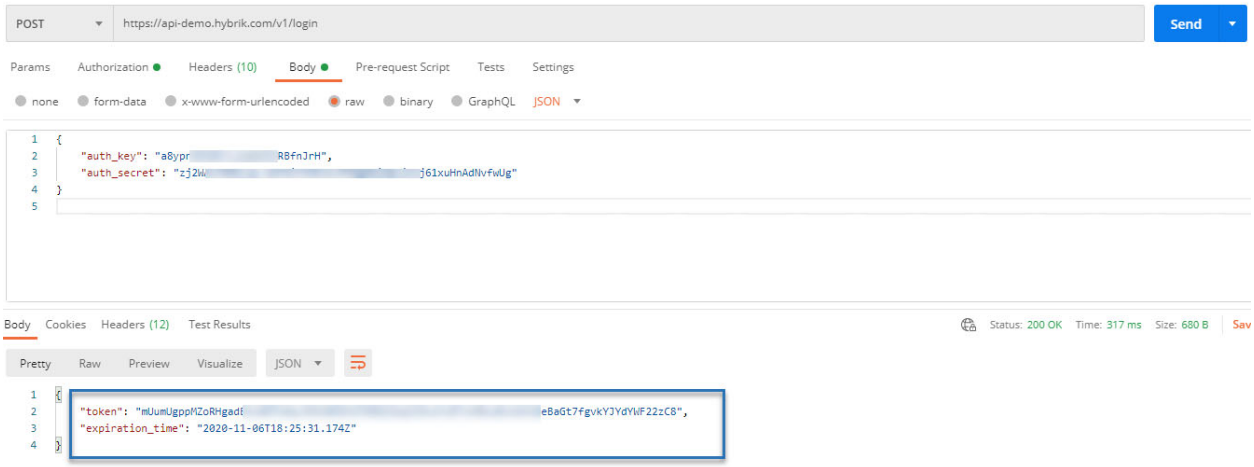


4. Build the Body of the request:
   a. Click "Body" Tab and select **raw** and the format is **JSON**.
   b. Enter the following parameters:
      – **auth_key**
      – **auth_secret**




```
{
        "auth_key": "8yprXXXXXXXXXXXX9jRBfXXXX"
        "auth_secret": "zj2XXXXXXXXX-d2f4iXXXXXXXXXXXXXXXXXXXXj61xuHnAdNXXXXg"
}
```

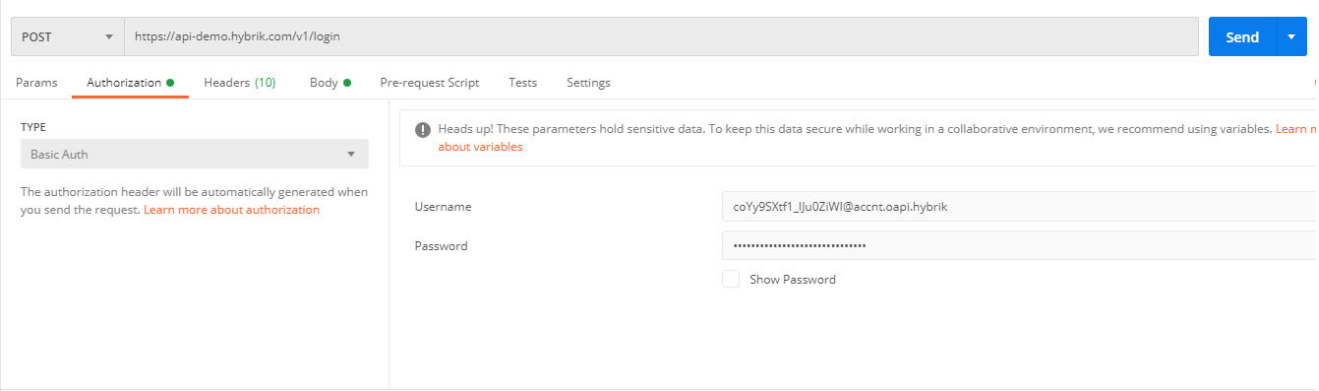
5. Click **SEND**. The expected response is the security **Token** that confirms login to the Hybrik API. The token will expire by the **expiration_time** if actions are not taken against the API within that timeframe. Every new action will extend the token.



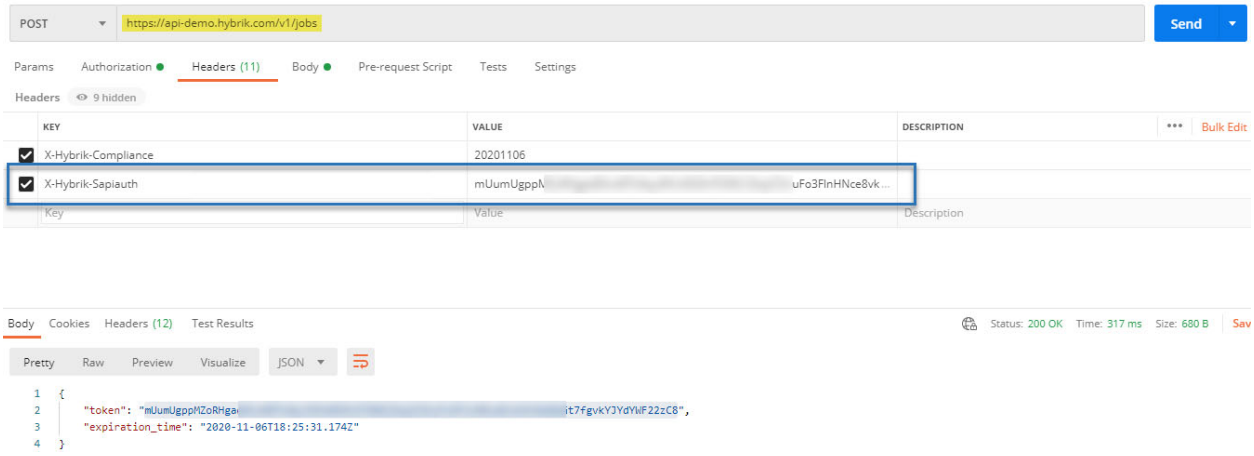
POST Request in Postman - Send Job and Tasks

1. In Postman, create **POST** request to the API https://api-demo.hybrik.com/v1/jobs

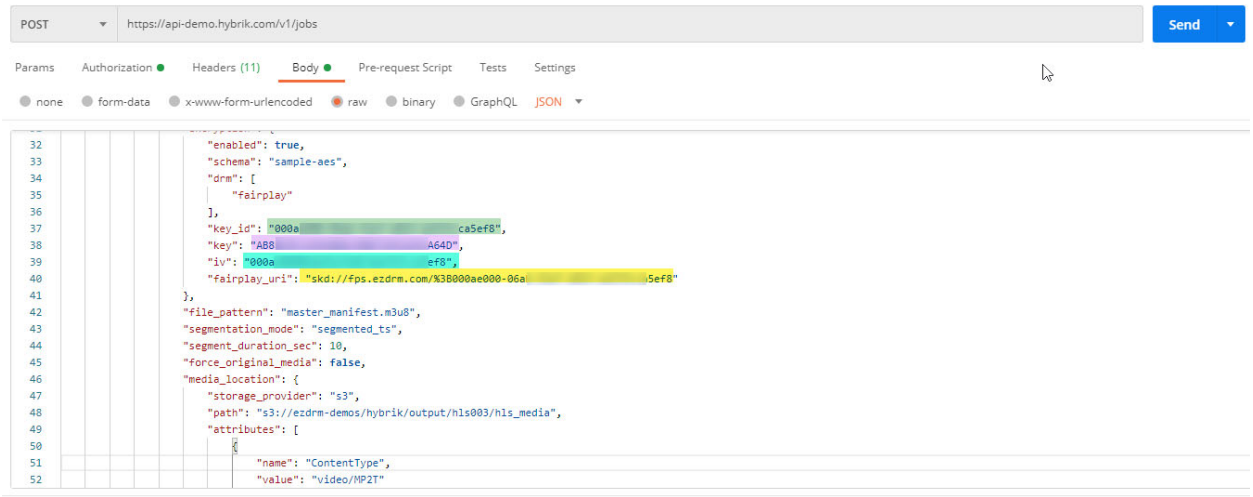
2. Under **Authorization**, select Type: **Basic Auth** and enter your **Hybrik OAPI Key (Username)** and **Hybrik OAPI Secret Key (Password)**.



3. Under **Headers**, add the following Header:
   - **X-Hybrik-Compliance** – enter value: current date (YYYMMDD format)
   - **X-Hybrik-Sapiauth** – enter the security **Token** from **Step 5** in the last section



4. Build the Body of the request:
    a. Click “Body” Tab and select **raw** and the format is **JSON**.
    b. Enter the following parameters:



- **input**: location of the file to be encoded (S3 bucket, etc. where the .mp4 can be downloaded) * if utilzing S3, locations should be set to Public

- **output**: location of the encoded output file to be encoded (S3 bucket, etc.) * if utilzing S3, locations should be set to Public, and CORS Access-Control-Allow-Origin

- **key_id**: Content Key **kid** in GUID format (client generated)

- **key**: use the **pskc:Secret key** value and decode from Plain Value Base 64 to HEX format in lowercase. An example decoder can be found at: https://tomeko.net/online_tools/base64.php?lang=en

pskc:Secret key (Base 64) = q4sXXXXXAGFQvFKRKXXXTQ==

key (HEX) = AB8XXXXXXXXXX6150BC52912A41A64D

- **IV** – use the **explicitIV** and decode from Plain Value Base 64 to HEX format.

  explicitIV (Base 64) = AArgXXXrQe+gEXXXXXpe+A==

  encryption_iv (HEX) = 000AXXXX06ABXXXXA013AE533CCXXXX8

- **fairplay_uri**: the base license URL for FairPlay on Hybrik:

  skd://fps.ezdrm.com/%3B<<kid>>

  ***Note:* %3B is used in the uri format for Hybrik, as opposed to the usual semicolon format (*skd://fps.ezdrm/;<<kid>>*)**

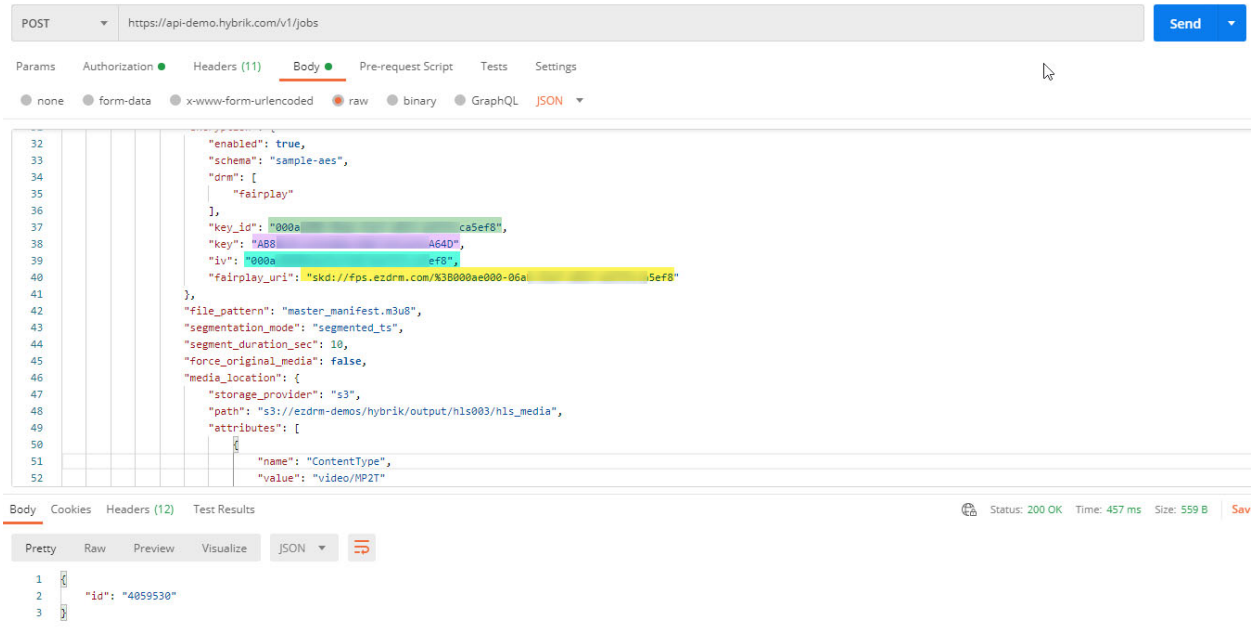
```
{
    "name": "hls-003abc123",
    "payload": {
        "elements": [
            {
                "uid": "source_file",
                "kind": "source",
                "payload": {
                    "kind": "asset_url",
                    "payload": {
                        "storage_provider": "s3",
                        "url": "s3://ezdrm-sample-content/BigBuckBunny_320x180.mp4" //S3  source location
                    }
                }
            },
            {
                "uid": "package_hls",
                "kind": "package",
                "payload": {
                    "kind": "hls", //document type
                    "location": {
                        "storage_provider": "s3",
                        "path": "s3://ezdrm-demos/hybrik/output/hls002/hls_manifests", //output location
                        "attributes": [
                            {
                                "name": "ContentType",
                                "value": "application/x-mpegURL"
                            }
                        ]
                    },
                    "encryption": {
                        "enabled": true,
                        "schema": "sample-aes",
                        "drm": [
                            "fairplay"
                        ],
                        "key_id": "000aXXXX-06ab-XXXX-a013-ae533ccXXXX8",
                        "key": "AB8XXXXXXXXXXX6150BC52912A41A64D",
                        "iv": "000aXXXX06abXXXXa013ae533ccXXXX8",
                        "fairplay_uri": "skd://fps.ezdrm.com/%3B000aXXXX-06ab-XXXX-a013-ae533ccXXXX8"
                    },
                    "file_pattern": "master_manifest.m3u8", // m3u8 manifest URL
```

```
                    "segmentation_mode": "segmented_ts",
                    "segment_duration_sec": 10,
                    "force_original_media": false,
                    "media_location": {
                        "storage_provider": "s3",
                        "path": "s3://ezdrm-demos/hybrik/output/hls002/hls_media", //output location
                        "attributes": [
                            {
                                "name": "ContentType",
                                "value": "video/MP2T"
                            }
                        ]
                    },
                    "media_file_pattern": "bbb.ts",
                    "hls": {
                        "media_playlist_location": {
                            "storage_provider": "s3",
                            "path": "s3://ezdrm-demos/hybrik/output/hls002/hls_manifests" // output
                        },
                        "include_iframe_manifests": true
                    }
                }
            }
        ],
        "connections": [
            {
                "from": [
                    {
                        "element": "source_file"
                    }
                ],
                "to": {
                    "success": [
                        {
                            "element": "package_hls"
                        }
                    ]
                }
            }
        ]
    }
}
```

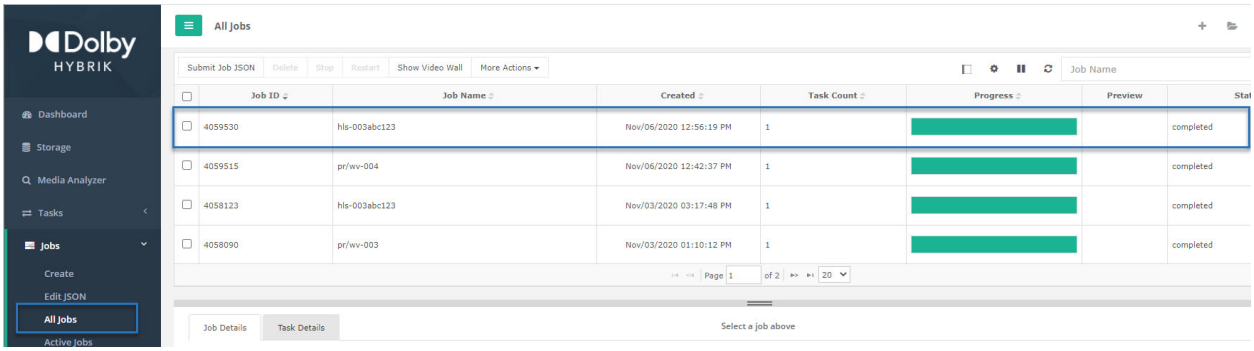
Send the Post.

Send Job

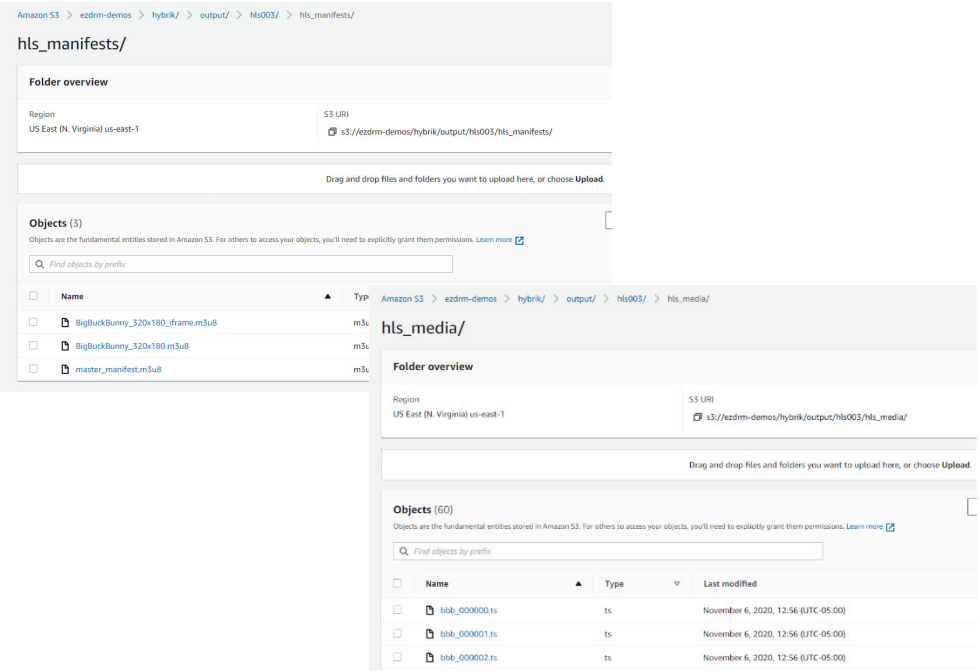
The return will include the job **id** – use this value to verify the status of the job in Hyrbik.



To verify the job in Hybrik, click on **Jobs** menu, then **All Jobs**. The **id** return will show the status of the job.



The output files will be created in the specified file location.



Testing Playback

For details on testing playback, look at **EZDRM Testing Playback** document under EZDRM Implementation on our documentation page: [EZDRM: Documentation and Setup Guides]

Additional Information

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