

# **EZDRM Bitmovin** Configuration Encoding



# **Table of Contents**

Prerequisites	3
Preparing for Bitmovin Encoding	4
Step 1: API Key from Bitmovin	4
Step 2. Generating DRM Keys	5
Widevine and PlayReady	5
Apple FairPlay Streaming	7
Step 3. AWS - S3 Bucket Set-up	10
Create a New User	10
Create a New Role	13
Create a New S3 Bucket	15
<b>Bitmovin Encoding</b>	<b>18</b>
Widevine and PlayReady	18
Apple FairPlay Streaming	19
Encoding File Configuration Example - Python	<b>20</b>
Encoding Success	22
<b>Appendix 1 – Call for Existing EZDRM Keys</b>	<b>24</b>
Widevine and PlayReady	24
Apple FairPlay Streaming	27
Appendix 2 – Installing ARC Plug-in	30

# Prerequisites

To use Encoding through the Bitmovin API, you must first get credentials at <u>https://bitmovin.com/bitmovins-video-api/</u>.

To begin encoding through the Bitmovin API, you will pick your API language and go through the setup steps at: <a href="https://dashboard.bitmovin.com/getting-started/encoding">https://dashboard.bitmovin.com/getting-started/encoding</a>

Then go to the corresponding GitHub library for your programming language at: <u>https://github.com/bitmovin/</u>

If using Python; Python 3.6 or higher is required.

To download Python 3.6: <u>https://www.python.org/downloads/</u>

# **Preparing for Bitmovin Encoding**

## **Step 1: API Key from Bitmovin**

An API Key is assigned to your Bitmovin account. To find the ID, under your Account name on the top right of the screen, click on the dropdown menu and select **Settings**.

*	BITMOVIN		Active Trials  🕑								O merchants •
E.	Dashboard		DASHBOARD								Your Organizations
*	Encoding	>	Billing								Show Team
Þ	Player	>	Encoding 0 / 40 Encodings		0 \$ 2018-09-30	Picyer 0 / Sk Impressions		0 \$ 2018-09-30	Analytics 0 / 5k Impressions		Your Subscriptions
al	Analytics	>									<ul> <li>Traing</li> <li>Player</li> </ul>
?	Getting Storted		Encoding			Player			Analytics		Analytics
-	Documentation of		0 Minutes	0 Minutes	0 Minutes	O	O	O	O	O	Manage Subscriptions Billing
4	Demos of		last 7 days	last 14 days	last 30 days	last 7 days	last 14 days	lost 30 days	last 7 days	kost 14 day	Settings
Q	Contact Soles			7 days	14 days 30 days		7 days	14 days 30 days		7 da	14 days 30 days

On the Account tab, under Api Keys, you will find the key you will need to save for the encoding file. Click the **Show** button to reveal the key.

Technical Contact		1
First Name:		
Lost Nome:	Rapin	
Company:	ezdm.com	
Phone:		
Emolt:	⊒ezdm.com	
Organizations		
Nome	Organization Id 😡	
@ezdm.com_Organization	2 = 0 + 0 − 1 - 1 + 23 tz = - 2	
Api Keys	Greate New	Show
c6e42adexxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx		

## **Step 2. Generating DRM Keys**

## Widevine and PlayReady

Below are the steps to create the DRM Keys for CENC-Widevine or CENC-PlayReady encryption for Bitmovin.

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".

LicenseInfo						
Click here for a complete list of operations.						
GenerateKeys						
Test						
To test the	operation using the HTTP POST protocol, click the 'Invoke' button.					
Parameter	Value					
U:						
P:	P:					
C:						
	Invoke					

The parameters are as follows:

Parameter	Description
u	EZDRM username
р	EZDRM password
С	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:



## **Option 2: Request DRM keys with curl**

The second option to request DRM keys from EZDRM is to script the process with curl or another web service call.

Using EZDRM's web service, the curl script below retrieves the DRM values from the web service.

curl -v 'http://wvm.ezdrm.com/ws/LicenseInfo.asmx/GenerateKeys?U=<u>EZDRM\_USERNAME</u>&P=<u>EZDRM\_PASSWORD</u>&C= """

The parameters are as follows:

Parameter	Description
U	EZDRM username
Р	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=""></ezdrm>
<widevine diffgr:haschanges="inserted" diffgr:id="WideVine1" msdata:roworder="0"></widevine>
<contentid>6IxXXx0Z8xXXXXXXXLbg==</contentid>
<key>W5XXXXXXZHxTjhXXXXVw==</key>
<keyhex><u>5bXXXXXXXXX9191fXXe38XXXXX56bf</u></keyhex>
<keyid>WVXXXXXXXliBEXXw+XXXXX==</keyid>
<keyidguid>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXb</keyidguid>
<keyidhex><u>5XXXXXX36d85XXXXXXXXXXXb24XXXb</u></keyidhex>
<pssh><u>EhXXXXXXXXXXXXXXXXXS6skGLGXXXXXX06IebXXXZ8kSYMXXXXXXXXXXXXXXXXXXXXXXX==</u></pssh>
<serverurl>https://widevine-dash.ezdrm.com/proxy?pX=XXXXXX</serverurl>
<serverget>request={"policy": "", "tracks": [ {"type": "SD"}], "content_id": "6IxXXx0Z8xXXXXXXX</serverget>
XXLbg=="}
<responseraw></responseraw>
{"status":"OK","drm":[{"type":"WIDEVINE","system_id":"edef8ba979d64acea3c827dcd51d21ed"}],"tracks":[{"type
":"SD","key_id":" WVXXXXXX1iBEXXw+XXXXX==","key":" W5XXXXXXZHxTjhXXXXXvw==","pssh":[{"drm_type":"WIDEVIN
E","data":"EhXXXXXXXXXXXXXXXX6skGLGXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXj3JXXXX=="}]}]}
<playready diffgr:haschanges="inserted" diffgr:id="PlayReady1" msdata:roworder="0"></playready>
<key>W5XXXXXXZHxTjhXXXXvw==</key>
<keyhex>5bXXXXXXXX9191fXXe38XXXXX56bf</keyhex>
<keyidguid>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXb</keyidguid>
<laurl>https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX</laurl>
<checksum>1Xq+XXXXX0=</checksum>

# 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 <u>Appendix 2</u> for instructions.
- 2. Change the Method dropdown to **POST**.

	Request	
HTTP request	GET	Request URL https://
Socket	POST	
History	PUT	Headers
Today		

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

http://fps.ezdrm.com/api/keys?u=USERNAME&p=PASSWORD

The parameters are as follows:

Parameter	Description
u	EZDRM username
р	EZDRM password



4. The following is an example of the response:

```
<FairPlay>
<AssetID>1XXXXX0-c7ed-4XXX-b15c-XXXXXXX25</AssetID>
<KeyHEX>D230XXXXXXXXXX17300XXXXXXXX4EABXXXXXX4349271XXXXB</KeyHEX>
<KeyID>0jXXXXXXXXXXXXDRjUrJ/XXXXXXXXQ0knFXXX=</KeyID>
<KeyUri>skd://fps.ezdrm.com/;1XXXXX0-c7ed-4XXX-b15c-XXXXXXX225</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 <u>Appendix 1</u>.

#### **Option 2: Request DRM keys with curl**

Using EZDRM's web service, the curl script below retrieves the DRM values from the web service.

curl -X POST 'http://fps.ezdrm.com/api/keys?U=<u>Username</u>&P=<u>Password</u>' -d ' '

*Note:* there is a space between the single quotes at the end of the line.

The parameters are as follows:

```
© EZDRM | www.EZDRM.com
```

Parameter	Description
U	EZDRM username
Р	EZDRM password

The following is returned:

<fairplay></fairplay>	
<assetid>1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXa25</assetid>	
<keyhex><mark>D230XXXXXXXXXXX17300XXXXXXXXXX</mark>4EABXXXXXXXXXXX4349271XXXXXB</keyhex>	
<keyid>0jXXXXXXXXXXXWDRjUrJ/XXXXXXXXXXXkQ0knFXXX=</keyid>	
<keyuri><u>skd://fps.ezdrm.com/:1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXa25</u></keyuri>	
<licensesurl>http://fps.ezdrm.com/api/licenses</licensesurl>	
<supportedfpsversions>1</supportedfpsversions>	

Here are the descriptions of the items returned:

- **KeyHEX:** the DRM Encryption Key. This is given back in two forms: Base64 and HEX. The <u>first 32 characters form **KeyHEX** are used in the key file for "key"</u> and the last 32 characters are used for "iv".
- **KeyUri**: the License URL for encryption.

## Step 3. AWS - S3 Bucket Set-up

## Create a New User

To create a new User in AWS complete the following steps:

1. Launch the AWS IAM console by searching for IAM.

aws	Services 🗸	Resource Groups 🐱	*	
_				
AV	NS services			
IA	AM]			Q
L	AM Manage User Access an	d Encryption Keys		*
				~

2. Go to the Users menu. Click the **Add user** button.

aws s	ervices 🗸 Resource Groups 🗸 🛠		
Search IAM	Add user Delete user		
Dashboard	Q Find users by username or access k	rey	
Groups	liser name	Groups	Access ki
Users		Groups	
Roles	· · · · · · · · · · · · · · · · · · ·		There are no IAM users. Le
Policies			
Identity providers			
Account settings			

3. Enter a **User name**, for this example "bitmovin-demo". Then under the "Select Access Type" section, click the checkbox to enable **Programmatic access**. This enables an access key ID and secret access key .

Add user		0	2 3 4
Set user det	ails		
You can add multip	le users at once wi	the same access type and permissions. Learn more	
	User name*	bitmovin-demo	
		O Add another user	
Select AWS ac	cess type		
Select how these u	sers will access AV	S. Access keys and autogenerated passwords are provided in the last step. Learn more	
	Access type*	<ul> <li>Programmatic access</li> <li>Enables an access key ID and secret access key for the AWS API, CLI, SDK, and other development tools.</li> </ul>	
		AWS Management Console access Enables a password that allows users to sign-in to the AWS Management Console.	
* Required		Cancel	Next: Permissions

- 4. Click the **Next: Permissions** button.
- 5. Select the **Attach existing policies directly** menu, search for "S3" and click the checkbox to select **AmazonS3FullAccess**. This provides full access to all buckets via the AWS Management Console.

ρ		dd user to group	existing user	direc	ch existing policies http://www.commons.com/commons/com http://commons/commons/commons/commons/commons/commons/commons/commons/commons/commons/commons/commons/commons/c
Creat	te p	olicy			0
ilter	poli	icies - Q 53	>		Showing 5 results
		Policy name -	Туре	Used as	Description
17		AmazonDMSRedshi	AWS managed	None	Provides access to manage S3 settings for
/	÷	8 AmazonS3FullAccess	AWS managed	Permissions policy (3)	Provides full access to all buckets via the A
	•	AmazonS3ReadOnI	AWS managed	None	Provides read only access to all buckets via
	×	CrossS3Test	Customer managed	None	
		QuickSightAccessF	AWS managed	None	Policy used by QuickSight team to access c

- 6. Click the **Next: Review** button.
- 7. Review the new User settings on the Review page and click the **Create user** button.

Add user			1	2	3	4
Review						
Review your choices.	After you create th	user, you can view and download the autogenerated password and access k	ey.			
User details						
	User name	bitmovin-demo				
AV	VS access type	Programmatic access - with an access key				
Permis	sions boundary	Permissions boundary is not set				
Permissions sun	nmary					
The following policies	will be attached t	he user shown above.				
Туре	Name					
Managed policy	AmazonS3Ful	ccess				
		Cancel	Previ	ous	Create	user

8. The last step is the Success screen. Here you will have access to the **Console Login Link**, the **Access key ID** and the **Secret access key**. **Download the .csv** file with the key information and save this file where it can be accessed for future reference.

*Important Note:* This is the ONLY TIME you will have access to the Secret access key. It is important to download the .csv with this information or copy and paste the keys into a saved document. Once you leave this screen you will no longer have access to the Secret access key.

#### Add user



# These values will be used for encoding:

C

S3_INPUT_ACCESSKEY = '	AKIAXXXXXXXXXXXXXXAA' (AWS User Access Kev ID)	
S3_INPUT_SECRETKEY = '	PoXXX/qL0xXXXXxxXXXtnn26xXXXXXXXVV0vXXxx' (AWS User Secret Access Key)	

D

AKIAJXXXXXXJZ4MOY2KA POXXXXX/qLGAIhXXXXXPoyS53pOGrC7VulHyaXXX https://888859578094.signin

Console login link

Secret access key

## Create a New Role

A

2 cli-access

B

1 User name Password Access key ID

If you don't already have a Role created in AWS with Administrator or S3 Full Access:

1. From the AWS IAM console, click on the **Roles** menu.

lashboard Iroups	What are IAM roles?
sees	IAM roles are a secure way to grant permissions to entities that you trust. Examples of entities include the following:
oles	IAM user in another account
	<ul> <li>Application code running on an EC2 instance that needs to perform actions on AWS resources</li> </ul>
*olicies	<ul> <li>An AWS service that needs to act on resources in your account to provide its features</li> </ul>
dentity providers	Users from a corporate directory who use identity federation with SAML
ccount settings	IAM roles issue keys that are valid for short durations, making them a more secure way to grant access.
veoenciai report	Additional resources:
	IAM Roles FAQ
Encryption keys	IAM Roles Documentation
	Tutorial: Setting Up Cross Account Access
	Common Scenarios for Roles

2. Select AWS Service, the **S3** service, and click **Next: Permissions**.

AWS service	Another A	WS account	b identity	SAML 2.0 federation
EC2, Lambda and off	hers Belonging to	you or 3rd party	vider	Your corporate directory
llows AWS services to perf	orm actions on your behalf. Le	am more		
Choose the servic	e that will use this r	ole		
EC2				
Allows EC2 instances to ca	II AWS services on your behalf			
Lambda				
	h call AWS services on your he	18/f.		
Allows Lambda functions to	o can rate out they on your de			
Allows Lambda functions to API Gateway	Config	EMR	IoT	Rekognition
Allows Lambda functions to API Gateway AWS Support	Config DMS	EMR	loT Kinesis	Rekognition S3
Allows Lambda functions to API Gateway AWS Support AppSync	Config DMS Data Lifecycle Manager	EMR ElastiCache Elastic Beanstalk	loT Kinesis Lambda	Rekognition 53 SMS
Allows Lambda functions to API Gateway AWS Support AppSync Application Auto Scaling	Config DMS Data Lifecycle Manager Data Pipeline	EMR ElastiCache Elastic Beanstalk Elastic Container Service	loT Kinesis Lambda Lex	Rekognition S3 SMS SNS

3. Either give the Role AdministratorAccess or select AmazonS3FullAccess. Click Next: Review.

Create role		1 2 3		
Attach permissions policies Choose one or more policies to attach to your new role. Create policy		0		
Fitter policies ~ Q. Search Policy name ~	Used as	Create role		1 2 3
<ul> <li>AdministratorAccess</li> </ul>	Permissions policy (4)	<ul> <li>Attach permissions policies</li> </ul>		
AlexaForBusinessDeviceSetup	None	Choose one or more policies to attach to your new role.		
AlexaForBusinessFullAccess	None	Create policy		0
AlexaForBusinessGatewayExecution	None			
AlexaForBusinessReadOnlyAccess	None	Filter policies ~ Q 30		Showing 5 results
AmazonAPIGatewayAdministrator	None	Policy name -	Used as	Description
AmazonAPIGatewayInvokeFullAccess	Permissions policy (4)	Amaran Di / Charlebite 20a/a	Alana	Drouides societe to manage S2 settings f
AmazonAPiGatewayPushToCloudWatchLog     Beouled	s None	AmazonS3FullAccess	Permissions policy (4)	Provides full access to all buckets via th.
nequires		AmazonS3ReadOnlyAccess	None	Provides read only access to all buckets
		CrossS3Test	None	
		QuickSightAccessForS3StorageManagement.	None	Policy used by QuickSight team to acce
		* Required		Cancel Previous Next: Review

4. Enter the **Role Name** and click **Create role**.

oreate role		1 2 3
Review		
Provide the required information below and review	this role before you create it.	
Role name*	EZDRM-bitmovin	
	Use alphanumeric and '++, 0-,,' characters. Maximum 64 characters.	
Role description	Allows S3 to call AWS services on your behalf.	
	Maximum 1000 characters. Use alphanumeric and '++, @' characters.	
Trusted entities	AWS service: s3.amazonaws.com	
Policies	👔 AmazonS3FullAccess 🕼	
Permissions boundary	Permissions boundary is not set	

## Create a New S3 Bucket

1. From the AWS Console, search for **S3 Scalable Storage in the Cloud** and open.

aws	Services 🗸 Resource Groups 🗸 🛠
	AWS services
	\$3
	S3 Scalable Storage in the Cloud
	Athena Query Data in S3 using SQL
$\sim$	10W <sup>4</sup>

#### 2. Click the Create Bucket button.



3. Enter the following parameters and click **Next**:

• Bucket name: Bucket name can be any format you prefer. We recommend that you use a naming convention that is unique and reusable.

Note: The bucket name must be unique across all existing bucket names in Amazon S3.

• Region: enter the region closest to you.

	Create	bucket		×
1 Name and region	2 Set properties	3 Set permissions	(4) Review	
Name and region				
Bucket name 🕚				
ezdrm-aws-12345	56-001			
Region				
US East (N. Virgir	nia)		~	
Copy settings fro	m an existing bucket			
Select bucket (op	tional)	3	2 Buckets 🗸 🗸	
Create			Cancel	lext

4. Click the **Next** button through the next three screens keeping all the default settings, then click the **Create Bucket** button.

ame and region (2)		Create buck		Create bucket	
Versioni	Manage users	properties (3	Name and region	Set properties 🕜 Set permissions	(4) Review
Keep multiple versions o the same buck			Name and region		
Disabled	okornienko(Owner)	✓ Read ✓ Write	Bucket name ezdrm-aws- Properties	123456-001 Region US East (N. Virginia)	
Tags	Access for other AWS account	+ Add account	Versioning	Disabled	
lago		Objects 🚺 O	Tagging	0 Tags	
Use tags to track your projects or other o	Manage public permissions		Object-level logging Default encryption	Disabled None	
	Do not grant public read access to	o this bucket (Recomm	Permissions		
	Mine and the state				

5. Once the bucket is created, create two folders named "input" and "output" by clicking the **Create folder** button.

Overview	Properties	Permissions	Management		
Type a prefix and press Ente	r to search. Press ESC to clear.				
Upload + Create folder	More ~				US East (N. Virginia)
					Viewing 1 to 3
Name 15		Last	modified 11	Size 1=	Storage class 1=
🕞 😂 input					

6. Upload your files for encoding in the **input** folder.

These values will be used for encoding:

S3\_INPUT\_BUCKETNAME = 'ezdrm-aws-123456-001'(S3 Bucket Name)
S3\_INPUT\_PATH = 'input/BigBuckBunny\_320x180.mp4' (S3 Input Path and Filename)
S3\_OUTPUT\_BUCKETNAME = 'ezdrm-aws-123456-001'(S3 Bucket Name)

# **Bitmovin Encoding**

# Widevine and PlayReady

To create Multi-DRM protected content, the following EZDRM DRM Key values are required for Widevine and PlayReady:

```
API_Key = 'YOUR BITMOVIN ACCOUNT API_KEY'

S3_INPUT_ACCESSKEY = 'AWS_USER_ACCESS_KEY'

S3_INPUT_SECRETKEY = 'AWS_USER_SECRET_KEY'

S3_INPUT_BUCKETNAME = 'S3_BUCKET_NAME'

S3_OUTPUT_ACCESSKEY = 'AWS_USER_ACCESS_KEY'

S3_OUTPUT_SECRETKEY = 'AWS_USER_ACCESS_KEY'

S3_OUTPUT_SECRETKEY = 'AWS_USER_SECRET_KEY'

S3_OUTPUT_BUCKETNAME = 'S3_BUCKET_NAME'

CENC_KEY = 'EZDRM_KevHEX'

CENC_KID = 'EZDRM_KevHEX'

CENC_WIDEVINE_PSSH = 'Widevine_PSSH'

CENC_PLAYREADY_LA_URL = 'PlayReady_LAURL'
```

The parameters of the configuration have the following meaning:

- **API Key:** This is the **API Key** provided on your Bitmovin Account (see <u>Step 1</u>).
- S3 Input/Output Access Key: This is the Access Key provided when the AWS User is created (see <u>Step 3</u>).
- S3 Input/Output Secret Key: This is the Secret Key provided when the AWS User is created (see <u>Step 3</u>).
- S3 Input Bucket Name: the name of the S3 Bucket (see Step 3).
- **S3 Input Path:** This is the path to the file you want to encode, located in the Input folder of the S3 bucket.

- **Key:** This is the common content encryption key in hex format, the EZDRM **KeyHEX** value.
- **Kid:** This is the common unique identifier for your content key in hex format, the EZDRM **KeyIDHEX** value.
- **widevinePssh:** This is the value for the Widevine PSSH box, the EZDRM **PSSH** value.
- **playreadyLaUrl:** This is the URL to the PlayReady license server, the EZDRM PlayReady LAURL value.

More information on creating Multi-DRM protected content please reference: <u>https://developer.bitmovin.com/hc/en-us/articles/115001080113-How-to-create-combined-Multi-DRM-protected-content</u>

# Apple FairPlay Streaming

To create Multi-DRM protected content, the following EZDRM DRM Key values are required for Apple FairPlay Streaming:

```
FAIRPLAY_KEY = "D230XXXXXXXXXXXXXXX17300XXXXXXXXXX"; (The first 32 characters of EZDRM KeyHEX)
FAIRPLAY_IV = "X4EABXXXXXXXXXX4349271XXXXXB"; (The last 32 characters of EZDRM KeyHEX)
FAIRPLAY_URI = "_skd://fps.ezdrm.com/:1XXXXX0-c7ed-4XXX-b15c-XXXXXXXX25" (EZDRM KeyUri)
```

Here are the descriptions of the items returned:

- **Key:** the 128-bit content encryption key. This is the first 32 characters from the EZDRM **KeyHEX** value.
- **IV:** the initialization vector (IV) used for encryption. This is the last 32 characters in the EZDRM **KeyHEX** value.
- **URI**: the License URL used for the license key call, this is the EZDRM **KeyUri** value.

More information on creating DRM protected content for Apple FairPlay please reference:

https://developer.bitmovin.com/hc/en-us/articles/115001080133-How-to-create-Fairplay-DRM-protected-content

# **Encoding File Configuration Example - Python**

1. To find your proper encoding file script, go to the GitHub library for your programming language at: <u>https://github.com/bitmovin/</u>.

For this example, we will use the Python encoding files at: <u>https://github.com/bitmovin/bitmovin-python</u>

- Then select the proper encoding script, for example the encoding for Dash CENC HLS FairPlay is located here: <u>https://github.com/bitmovin/bitmovin-</u> <u>python/blob/master/examples/encoding/create encoding dash cenc hls fair</u> <u>play.py</u>
- 3. In the "create\_encoding\_dash\_cence\_hls\_fairplay.py" file you will need to update these configuration items with your details to encode a test video as described in the previous section:

```
448 lines (382 sloc) 26.6 KB
                                                                                                                                                                                                                      Raw Blame History 🖵 🧨 🗊
                 import datetime
                from bitmovin import Bitmovin, Encoding, S3Output, H264CodecConfiguration, \
                        AACCodecConfiguration, H264Profile, StreamInput, SelectionMode, Stream, EncodingOutput, ACLEntry, ACLPermission, \
                        {\tt FMP4Muxing, MuxingStream, DashManifest, DRMFMP4Representation, FMP4RepresentationType, Period, {\tt Variable} and {\tt FMP4Muxing, MuxingStream, DashManifest, DRMFMP4Representation, FMP4RepresentationType, Period, {\tt Variable} and {\tt FMP4Muxing, MuxingStream} and {\tt FMP4Muxing} and {\tt FMP4Muxing, MuxingStream} and {\tt FMP4Muxing} and {\tt FMP4Muxing, MuxingStream} and {\tt FMP4Muxing, MuxingStream} and {\tt FMP4Muxing, MuxingStream} and {\tt FMP4Muxing, MuxingStream} and {\tt FMP4Muxing} and {\tt F
     4
                        VideoAdaptationSet, AudioAdaptationSet, ContentProtection, S3Input, HlsManifest, VariantStream, \
                       AudioMedia, FairPlayDRM, TSMuxing
     6
            from bitmovin import CENCDRM as CENCDRMResource
     8 from bitmovin.resources.models import CENCPlayReadyEntry, CENCWidevineEntry
              from bitmovin.errors import BitmovinError
     9
               from bitmovin.resources.models.encodings.drms.cenc_marlin_entry import CENCMarlinEntry
    10
    12 API_KEY = '<YOUR_API_KEY>'
             S3_INPUT_ACCESSKEY = '<YOUR_S3_OUTPUT_ACCESSKEY>'
    14
    15 S3_INPUT_SECRETKEY = '<YOUR_S3_OUTPUT_SECRETKEY>'
    16 S3_INPUT_BUCKETNAME = '<YOUR_S3_OUTPUT_BUCKETNAME>'
    17 S3_INPUT_PATH = '<YOUR_S3_INPUT_PATH>'
    18
    19 S3_OUTPUT_ACCESSKEY = '<YOUR_S3_OUTPUT_ACCESSKEY>'
   20 S3_OUTPUT_SECRETKEY = '<YOUR_S3_OUTPUT_SECRETKEY>'
    21 S3_OUTPUT_BUCKETNAME = '<YOUR_S3_OUTPUT_BUCKETNAME>'
              CENC_KEY = '<YOUR_CENC_KEY>'
    24 CENC_KID = '<YOUR_CENC_KID>'
    25 CENC_WIDEVINE_PSSH = '<YOUR_CENC_WIDEVINE_PSSH>'
   26 CENC_PLAYREADY_LA_URL = '<YOUR_PLAYREADY_LA_URL>'
    27
    28 FAIRPLAY_KEY = '<YOUR_FAIRPLAY_KEY>'
   29 FAIRPLAY IV = '<YOUR FAIRPLAY IV>'
    30 FAIRPLAY_URI = '<YOUR_FAIRPLAY_LICENSING_URL>'
   31
               date_component = str(datetime.datetime.now()).replace(' ', '_').replace(':', '-').split('.')[0].replace('_', '__')
    32
    33
               OUTPUT_BASE_PATH = 'your/output/base/path/{}/'.format(date_component)
    34
```

#### Sample values:



Once the encoding file has been updated, it can be run using the chosen script.

## **Encoding Success**

Once the file encoding script is completed, the Encoding will be created in the Bitmovin dashboard.

*	BITMOVIN		Active Trials 😥 🕟 📹				1	• •
	Dashboard		ENCODING / ENCODINGS					
*	Encoding	~	Encodings					T Filter C Refresh
HI.	Encodings		□ Name ≎	Type ‡	Region \$	Created At +	Enqueued	Overall Status \$
B	Live Encodings		005-example encoding	VOD	a	2018-09-05 14:25	1m 10s	2m 28s 100%
0	Per-Title Demo 👫							1 ())
0*	Codec Configs							
2	Inputs							
4	Outputs							
Ø	Filters							
攌	Manifests							
8	Infrostructure							
ы	Statistics							
C	Notifications							

The output folder in your S3 bucket will store the encoded files.

Overview				
Type a prefix and press	Enter to search. Press ESC to clear.			
Upload + Create fe	older More ~			US East (N. Virginia)
				Mauria a 1 to 2
				viewing 1 to 3
Name †=		Last modified 1=	Size 1=	Storage class 1=
Name ↑≞.		Last modified 1=.	Size 11	Storage class 1=
Name 1=		Last modified 1=	Size ↑≞ 	Storage class 1°=.

# 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".

L	LicenseInfo			
Cli	ck <u>here</u> for a	a complete list of operations.		
G	enerate	Keys		
Te	st			
	To test the o	operation using the HTTP POST protocol, click the 'Invoke' button.		
	Parameter	Value		
	U:	username		
	P:	password		
	C: 6IxXXx0Z8xXXXXXXXLbg==			
		Invoke		

The parameters are as follows:

Parameter	Description
u	EZDRM username
р	EZDRM password
С	Content ID

c. Your return results will always return the existing DRM keys that are tied to the Content\_ID.

```
v <EZDRM xmlns="">
  v<WideVine diffgr:id="WideVine1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
     <ContentID>6IxXXx028xXXXXXXLbg==</ContentID>
     <Key>W5XXXXXXZHxTjhXXXXvw==</Key>
     <KeyHEX>5bXXXXXXXX9191fXXe38XXXXX56bf </KeyHEX>
     <KeyID>WVXXXXXXX1iBEXXw+XXXXX==</KeyID>
<KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXb</KeyIDGUID>
     <KeyIDHEX>5XXXXX36d85XXXXXXXXXXX2b24XXXb</KeyIDHEX>
    V <PSSH>
       EhXXXXXXXXXXXXXXX6skGLGXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXj3JXXXX==
     </PSSH>
     <ServerURL>https://widevine-dash.ezdrm.com/proxy?pX=XXXXXX</ServerURL>
    ▼<ServerGet>
       request={"policy": "", "tracks": [ {"type": "SD"}], "content_id": "6IxXXx0Z8xXXXXXXLbg=="}
     </ServerGet>
    V<ResponseRaw>
       ("status":"OK","drm":[{"type":"WIDEVINE","system_id":"edef8ba979d64acea3c827dcd5ld2led"}],"tracks":
[{"type":"SD","key_id":"WVXXXXXX1iBEXXw+XXXXX=","key":"WVXXXXXX1iBEXXw+XXXXX=,"pssh":
[{"drm_type":"WIDEVINE","data":"EhXXXXXXXXXXXXXXS6skGLGXXXXXXXQ6lebXXX28kSYMXXXXXXXXXXXXXXXXXXXj3JXXXX="}]}]
     </ResponseRaw>
   </WideVine>
  v<PlayReady diffgr:id="PlayReady1" msdata:rowOrder="0" diffgr:hasChanges="inserted">
     <Key>W5XXXXXXZHxTjhXXXXVw==</Key>
     <KeyHEX>5bXXXXXXX9191fXXe38XXXXX56bf </KeyHEX>
     <KeyIDGUID>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXb</KeyIDGUID>
    ▼ <LAURL>
       https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX
     </LAURL>
     <Checksum> 1Xg+XXXXX0=</</Checksum>
   </PlayReadv>
 </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.

The parameters are as follows:

Parameter	Description
U	EZDRM username
Р	EZDRM password
С	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 'http://wvm.ezdrm.com/ws/LicenseInfo.asmx/GenerateKeys?U=EZDRM USERNAME&P=EZDRM PASSWORD&C=6IxXXx0 Z8xXXXXXXXLbg==""

## Existing DRM Keys returned:

<ezdrm xmlns=""></ezdrm>
<widevine diffgr:haschanges="inserted" diffgr:id="WideVine1" msdata:roworder="0"></widevine>
<contentid><u>6IxXXx0Z8xXXXXXXXLbg==</u></contentid>
<key>W5XXXXXXZHxTjhXXXXvw==</key>
<keyhex>5bXXXXXXXXX19191fXXe38XXXXX56bf </keyhex>
<keyid>WVXXXXXXliBEXXw+XXXXX==</keyid>
<keyidguid>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXb</keyidguid>
<keyidhex>5XXXXXX36d85XXXXXXXXXXb24XXXb</keyidhex>
<pssh>EhXXXXXXXXXXXXXXXXXX6skGLGXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXXj3JXXXX==</pssh>
<serverurl>https://widevine-dash.ezdrm.com/proxy?pX=XXXXXX</serverurl>
<serverget>request={"policy": "", "tracks": [ {"type": "SD"}], "content_id": "6IxXXx0Z8xXXXXXX</serverget>
XXLbg=="}
<responseraw></responseraw>
{"status":"OK","drm":[{"type":"WIDEVINE","system_id":"edef8ba979d64acea3c827dcd51d21ed"}],"tracks":[{"type
":"SD","key_id":" WVXXXXXXliBEXXw+XXXXX==","key":" W5XXXXXXZHxTjhXXXXXvw==","pssh":[{"drm_type":"WIDEVIN
E","data":"EhXXXXXXXXXXXXXXXXX6skGLGXXXXXXXQ6IebXXXZ8kSYMXXXXXXXXXXXXXXXXj3JXXXX=="}]}]}
<playready diffgr:haschanges="inserted" diffgr:id="PlayReady1" msdata:roworder="0"></playready>
<key>W5XXXXXXZHxTjhXXXXvw==</key>
<keyhex>5bXXXXXXXXXX9191fXXe38XXXXX56bf</keyhex>
<keyidguid>5XXXXXX3-36XX-5XX8-8XX1-10XXXXXXXXb</keyidguid>
<laurl>https://playready.ezdrm.com/cency/preauth.aspx?pX=XXXXXX</laurl>
<checksum>1Xq+XXXXX0=</checksum>

# 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 <u>Appendix 2</u> for instructions.
- b. Change the Method dropdown to GET.

HTTP request	(	GET	Request URL https://
Socket		POST	_
History	^	PUT	Headers

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

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

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

HTTP request

Method GET Request URL

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

HTTP request	Method Request URL POST - http://fos.ez/rm.com/ani/keys/Ass	etID2u=IISERNAME&n=PASSWORD	, I v	SEND
Socket	Parameters ^			
History ^	Headers	Body	Variables	
Today	C <> Toogle source mode + Insert headers set			
POST http://fps.ezdrm.com/api/keys/AssetID?u_				
Monday, February 26, 2018	Header name Header value			×
GET http://fps.ezdrm.com/api/keys/1fdfc660-c7	ADD HEADER			
POST http://fps.ezdrm.com/api/keys?u=cpix@e Friday, February 23, 2018	Ay Headers are valid			Headers size: bytes
POST http://fps.ezdrm.com/api/keys?u=cpix@e	200 OK 160.20 ms			DETAILS 🗸
Saved				
Save a request and recall it from here	There was an error parsing XML d	ata		
Use meta+s to save a request. It will appear in this place.	<pre><assetid>1XXXXXX0-c7ed-4XXX <keyhex>D230XXXXXXXXXXXXXXI73 <keyid>01XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</keyid></keyhex></assetid></pre>	-bl5c-XXXXXXXXA25 00XXXXXXXXXXX4EABXXXXXXXXXXXXX J/XXXXXXXXXXXXXQ0knFXXX=/;1XXXXXX0-c7ed-4XXX-bl5c-XXX rm.com/api/licenses/Licenses! pportedFPSVersions>	<pre>K4349271XXXXXBD&gt; XXXXXXa25 Url&gt;</pre>	HEX>
Proiects 🗸				

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 '<u>http://fps.ezdrm.com/api/keys/AssetID</u>?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
Р	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 'http://fps.ezdrm.com/api/keys/1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXa25?U=Username&P=Password' -d
____'
```

#### Existing DRM Keys returned:

<fairplay></fairplay>
<assetid>1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXa25</assetid>
<keyhex><mark>D230XXXXXXXXXXX17300XXXXXXXXXX</mark>X4EABXXXXXXXXXXX4349271XXXXXB</keyhex>
<keyid>0jXXXXXXXXXXXWDRjUrJ/XXXXXXXXXXXkQ0knFXXX=</keyid>
<keyuri><u>skd://fps.ezdrm.com/;1XXXXXX0-c7ed-4XXX-b15c-XXXXXXXXA25</u></keyuri>
<licensesurl>http://fps.ezdrm.com/api/licenses</licensesurl>
<supportedfpsversions>1</supportedfpsversions>

# 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.

Google	advanced rest client				ļ	.↓ Q		
	All Video	is News	Images	Shopping	More	Settings	Tools	<
	About 2,200,000 results (0.31 seconds)							<
(	Advanced	REST clier	nt - Chrom	e Web Stor	e	6		<
	https://chrome.google.com//advanced-rest-client/hgmlootodiftdnphfgcellkdfbfbjeloo •							1
	https://www.toll	advancedresto	lient.com A b	etter 12 restin	a tool! Save your t	an desktop cheminor		
	testing tool of	it there. No cor	nplicated for	ms and scripts.	Easy to use yet ve	ime with the easiest A ery powerful. The only	REST	1

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

	40.0000 T	1			
OVERVIEW	REVIEWS	SUPPORT	RELATED		
		Compatible with your device			
Request	Request		8	-	
Socket	> Request URL			The web developers helper program	
Projects				to create and test custom HTTP requests.	
Saved	⑥ GET ○ POST ○ PUT ○	OELETE Other methods *			
History	Raw headers	Headers form	Predefined sets	GOOGLE.	
moott / avoot				Please, install desktop client from	

d. Click the **Add app** button to confirm installation.

Ad	dd "Advanced REST client"?
lt d	can:
•	Exchange data with any device on the local network or internet
•	Store data in your Google Drive account

e. Open the plug-in by clicking the ARC app button.

Web Store	Docs	YouTube	Gmail	Google Drive	Slides
Sheets					

Return to <u>Apple FairPlay Streaming</u> section to continue instructions for calling the EZDRM Key servers API.