BACKLOT API REFERENCE

CONTENTS

COPYRIGHT NOTICE

This content was last updated on 2016-04-04.

OVERVIEW

Welcome to the Ooyala Backlot API Reference.

This guide contains detailed information on each operation or route that can be performed using the Backlot APIs.

HTTP RESPONSE CODES

Depending on request results, the Backlot API returns different HTTP response codes.

Response Code	Short Description	Comments
200	OK	The Backlot API successfully processed the request.
204	No content	The server has fulfilled the request, but does not need to return an entity-body. The response does not have a message body.
400	Bad request	The request is poorly formed or contains invalid data. For example, the Backlot API returns 400 response code if your request contains malformed JSON or invalid settings, such as a time setting of "25:00:00".
401	Not authorized	The request is not properly signed or signed with an invalid API key.
403	Forbidden	The request is not properly authenticated for the requested operation.
404	Not found	The resource does not exist. For example, if you tried to delete a video using the wrong content ID.
429		Insufficient "API credits" to fulfill request. For explanation, see *http://support.ooyala.com/developers/documentation/concepts/api_rate_limiting.html*

ASSETS

The `/v2/assets` route deals with all types of assets.

When you create a new asset, the `asset_type` property in the request body can specify any of the following values.

- `video`
- `ad`: An ad is simply a video, marked as an ad in the Backlot UI.
- `audio`
- `remote_asset`
- `youtube`: A video destined for YouTube
- `live_stream`
- `live_audio`
- `channel`
- `channel_set`

After an asset is created with `[POST] /v2/assets` (as described in), in general you can update it in a variety of ways with `[PATCH] /v2/assets` or change its associations with `[PUT] /v2/assets`. See *Routes and Attributes Common to All Asset Types* on page 7.

ROUTES AND ATTRIBUTES COMMON TO ALL ASSET TYPES

The basic routes common to all assets have the same syntax, regardless of their type.

The common routes for all assets are the create, read, update, and delete (CRUD) operations. Except for creation and updating, which requires certain *properties* or workflow depending on the asset type, most of these common routes require only the identifier of the asset (`asset_id`).

CREATE ASSET

```
[POST]/v2/assets{
   properties
}
```

Certain asset types have considerable variation in how they are created:

- *Video and audio*
- *Remote assets*
- *Live streams*
- *YouTube assets*

LIST ALL ASSETS

```
[GET]/v2/assets
```

LIST SPECIFIC ASSET

```
[GET]/v2/assets/asset_id
```

MODIFY ASSET

```
[PATCH]/v2/assets/asset_id
```

REPLACE ASSET

```
[POST]/v2/assets/asset_id/replacement
```

DELETE ASSET

```
[DELETE]/v2/assets/asset_id
```

ROUTE ATTRIBUTES

The following table describes the attributes common to all variations of the `/v2/assets` routes.

Route Attribute	Description
ad_set_id	The ID of an ad set. Type: String Default: None Example: /assets/IzNnllMjphu2XF3_UgPROoCi9B2BwtSg/ad_set/PROoCi9B2BwPROoCi9B2Bw123
asset_id	The ID of the asset. This value is referred to by different names depending on where you look for it. In the Backlot API, asset_id is the identifier for a specific asset. asset_id has the same value as the content ID found in the Backlot UI that represents a piece of content. In the Backlot API, asset_id is also equivalent to embed_code. Type: String Default: None Example: /assets/IzNnllMjphu2XF3_UgPROoCi9B2BwtSg
channel_id	The content ID or embed code of a channel. Type: String Default: None Example: /assets/IzNnllMjphu2XF3_UgPROoCi9B2BwtSg
channel_set_id	The ID of a channel set. Type: String Default: None

Route Attribute	Description
	Example: /assets/IzNnllMjphu2XF3_UgPROoCi9B2BwtSg
drm_type	For Digital Rights Management (DRM), the type of DRM system. Type: String Default: None Valid Values: widevine | playready Example: /assets/IzNnllMjphu2XF3_UgPROoCi9B2BwtSg/drm_attributes/widevine
embed_code	The content ID from Backlot or embed code of a video within a channel. In the Backlot API, embed_code is equivalent to asset_id and the content ID from the Backlot UI. Please note that in the Backlot UI, the field for embed code refers to the embed snippet that you embed on a web page. Type: String Default: None Example: /assets/IzNnllMjphu2XF3_UgPROoCi9B2BwtSg
external_id	The external ID of a video within a channel, if applicable. Type: String Default: None Example: /assets/IzNnllMjphu2XF3_UgPROoCi9B2BwtSg
label_id	The ID of a label. Type: String Default: None Example: /assets/IzNnllMjphu2XF3_UgPROoCi9B2BwtSg/labels/QtsjklrewsddROoCi9B2B45drs
player_id	The ID of a player. Type: String Default: None Example: /players/b4f1a93b779145b4a9c8ef15f8502345
publishing_rule_id	The ID of a publishing rule. Type: String Default: None

Route Attribute	Description
	Example: /assets/IzNnllMjphu2XF3_UgPROoCi9B2BwtSg/publishing_rule/QtsjklrewsddROoCi9B2B45drs

ASSET PROPERTIES BY ASSET TYPES

Some basic properties are common to all assets, regardless of their type.

The following table describes all properties that can be associated with an asset. Some properties are used only with specific asset types.

Note:

- Unless stated otherwise below for a property, there are no default values.
- Examples of these properties are in the sections on individual types of assets.

Property	Description	For Asset Types	Required?
`all_day`	Recurring only. Specifies whether the content can be viewed during the entire day. If set to false, specify a start and end time. Type: Boolean Valid Values: true | false Default: true Example: true Parent: time_restrictions	all	No
`asset_type`	Type of asset. Type: String Valid Values: `video` | `ad` | `audio` | `remote_asset` | `youtube` | `live_stream` | `live_audio` | `channel` | `channel_set` Example: "video"	all	Yes
`backup_entry_point`	For RTMP streams only, URL of secondary source of live stream. Type: String Example: "rtmp://a.ep12372.i.akamaientrypoint.net/EntryPoint"	live_stream	Yes
`bitrate`	Bit rate of the encoding. Type: Integer Example: 300 Parent: encodings	live_stream	No
`chunk_size`	Size of the chunks to upload, in bytes.	video, audio, ad	Yes

Property	Description	For Asset Types	Required?
	Type: Integer Example: "10000"		
`click_url`	URL to go to when ad is clicked Type: String	ad	No
`closed_captions_url`	The URL for the closed captions file for this asset. Type: String Example: "http://my.closedcaptionssite.com/dxfpfile.dxfp"	all	No
`created_at`	Read only. Date and time the video was processed. Type: DateTime Example: "2011-04-25T15:32:57+00:00"	all	No
`description`	Description of the asset. **Note:** Cannot contain hidden characters, which are those with ASCII values between 0 and 31, except for 10 (newline) and 13 (carriage return). Type: String Example: "Season 22 Opening Game"	all	No
`disable_seek_for_standalone`	Do not allow viewers to manually advance or reverse the ad when it is embedded by itself or in a channel Type: String Valid Values: true | false Default: false	ad	No
`duration`	Length of the video, in milliseconds. Type: Integer Conditions: Required for remote assets unless `is_live_stream` is set `true`, in which case, `duration` is optional. Example: "120"	remote_asset	No
`embed_code`	Read only. Content ID of the asset that was assigned by Backlot when the asset was created. Type: String Example: "IzNnllMjphu2XF3_UgPROoCi9B2Bwrde"	all	No

Property	Description	For Asset Types	Required?
`embeddable`	Specifies whether the player showing the video can be embedded. Type: Boolean Valid Values: true | false Default: true Example: true Parent: youtube_syndication_settings	youtube	No
`encodings`	Specifies available encodings. Type: Container	live_stream	No
`end_date`	End date of the restriction. Type: DateTime (range) or Date (recurring) Default: None (does not end) Example: "2011-08-09T21:21:49-07:00" or 2011-08-09 Parent: time_restrictions	all	No
`end_time`	Recurring only. Specifies the end time when content can play. Type: Time Default: 23:59:59 Example: 21:00:00 Parent: time_restrictions	all	No
`external_id`	A custom identifier you define that you can use instead of the content ID (embed_code). Type: String Example: "my_movie.avi"	all	No
`file_size`	Size of the file, in bytes. Type: Integer Example: "20000"	video, audio, ad	Yes
`flash`	Specifies the URL of the Flash stream. Type: String Example: "http://example.com/my_flash_file.flv" Parent: stream_urls	remote_asset	No
`height`	Height of the encoding in pixels. Type: Integer Example: 600	live_stream	No

Property	Description	For Asset Types	Required?
	Parent: encodings		
`hosted_at`	The permanent URL at which you embed the asset. Type: String Example: "http://example.com"	remote_asset	No
`ipad`	Specifies the URL of the iPad stream. **Note:** Only MP4 or M3U8 formats are valid. Type: String Example: "http://example.com/ipad_compatible_file.mp4" Parent: stream_urls	remote_asset	No
`iphone`	Specifies the URL of the iPhone stream. **Note:** Only MP4 or M3U8 formats are valid. Type: String Example: "http://example.com/iphone_compatible_file.mp4" Parent: stream_urls	remote_asset	No
`is_flash`	Specifies whether the live stream is available in Flash. Type: Boolean Valid Values: true | false Default: `true` Example: `false`	live_stream	No
`is_ios`	Specifies whether the stream live stream is available in iOS. Type: Boolean Valid Values: true | false Default: `false` Example: `true`	live_stream	No
`is_live_stream`	Specifies whether this is a live stream. If the remote asset is a live stream, set this to true. Otherwise, set this to false. Type: Boolean Valid Values: true | false Default: `false` Example: false	remote_asset	No

Property	Description	For Asset Types	Required?
`itunes`	Specifies the URL of the iTunes stream. Type: String Example: "http://example.com/itunes_compatible_file.flv" Parent: stream_urls	remote_asset	No
`keywords`	Specifies keywords for the video. Type: Array Example: ["Hockey", "Ballet", "Trainspotting"] Parent: youtube_syndication_settings	youtube	No
`mobile`	Specifies whether the video can be viewed on mobile devices. Type: Boolean Valid Values: true | false Default: true Example: true Parent: youtube_syndication_settings	youtube	No
`name`	Name of the asset. **Note:** Cannot contain hidden characters, which are those with ASCII values between 0 and 31, except for 10 (newline) and 13 (carriage return). Type: String Example: "My Movie"	all	Yes
`original_file_name`	Name of the file that was specified when uploading the asset. Type: String Example: "my_movie.avi"	video, audio	Yes
`password`	For live_stream asset: Password of the primary encoder. For youtube asset: YouTube account password Type: String Example: "shhhh...itsasecret" Parent for youtube: youtube_syndication_settings	live_stream, youtube	Yes
`primary_encoder_ip`	For all streams other than RTMP, IP address of the primary encoder. Type: IP Address	live_stream	Yes

Property	Description	For Asset Types	Required?
	Example: "8.8.8.8"		
primary_entry_point	For RTMP streams only, URL of primary source of live stream. Type: String Example: "rtmp://a.ep12372.i.akamaientrypoint.net/EntryPoint"	live_stream	Yes
private	Specifies whether the video is private. When set to true, the video is unavailable for viewing. Type: Boolean Valid Values: true | false Default: false Example: false Parent: youtube_syndication_settings	youtube	No
publishing_rule_id	Identifier of publishing rule associated with the asset, if any. Type: String Example: ru1ze4ea4e648fd07d4509123254	all	No
recurring_days	Recurring only. List of days which the content can be viewed. Type: Array Valid Values: "SUN" | "MON" | "TUE" | "WED" | "THU" | "FRI" | "SAT" Default: None (all days) Example: ["SUN", "FRI", "SAT"] Parent: time_restrictions	all	No
source_file	Specifies the file representing the source stream. Type: String Example: "source_file.flv" Parent: stream_urls	remote_asset	No
start_date	Start date of the restriction. Type: DateTime (range) or Date (recurring) Default: None (starts immediately) Example: "2011-05-09T21:21:49-07:00" or 2011-05-09 Parent: time_restrictions	all	No
start_time	Recurring only. Specifies the start time when content can play.	all	No

Property	Description	For Asset Types	Required?
	Type: Time Default: 0:00:00 Example: 20:00:00 Parent: time_restrictions		
`status`	Sets the post-processing (post-transcoding) status of the asset. Type: String Valid Values: `live` | `paused` Default: `live` Example: "http://my.closedcaptionssite.com/dxfpfile.dxfp"	all	No
`status`	Returns the status of the video. Type: String Valid Values: uploading | processing | paused | live Example: "uploading"	all	No
`stream_urls`	Specifies the stream URLs. Type: Container	remote_asset	No
`time_restrictions`	Specifies time restrictions for the asset. Type: Container Default: None. Restrictions inherited from publishing rule, if any	all	No
`tracking_pixel_urls`	Comma-delimited list of URLs for tracking pixels Type: String Example: `tracking_pixel_urls: [ "http://trackingpixelurl.com","http://trackingurl.net" ]`	ad	No
`type`	Specifies whether this is a recurring or one-time range restriction. Type: String Valid Values: "range" | "recurring" Example: "range" Parent: time_restrictions	all	No
`updated_at`	Read only. Date and time the video was last modified. Type: DateTime	all	No

Property	Description	For Asset Types	Required?
	Example: "2011-04-25T15:32:57+00:00"		
`width`	Width of the encoding in pixels. Type: Integer Example: 800 Parent: encodings	live_stream	No
`youtube_id`	Identifier of the YouTube video. To find the YouTube identifier, locate the `v` request parameter within the YouTube URL. For example, for the URL `www.youtube.com?v=dQw4w9WgXcQ`, the YouTube ID is `dQw4w9WgXcQ` . Type: String Example: "dQw4w9WgXcQ"	youtube	Yes
`youtube_syndication_settings`	Contains YouTube syndication settings. Type: Container	video	No

TYPES OF ASSETS AND CORRELATION TO /V2/ASSETS QUALIFIERS

The `/v2/assets` route deals with all types of assets.

When you create a new asset, the `asset_type` property in the request body can specify any of the following values.

- video
- ad: An ad is simply a video, marked as an ad in the Backlot UI.
- audio
- remote_asset
- youtube: A video destined for YouTube
- live_stream
- live_audio
- channel
- channel_set

After an asset is created with `[POST] /v2/assets` (as described in *Creating an Asset: POST*), in general you can update it in a variety of ways with `[PATCH] /v2/assets`. See the generalized process in *Editing Asset Information: PATCH*.

In addition, the `/v2/assets` route has qualifiers for working with assets in general or specific types of assets, such as associating labels, publishing rules, ad sets, and more. Exact use of these qualifiers is detailed in other sections.

Table 1: /v2/assets Qualifiers by Asset Type

	video	ad	audio	remote_ asset	you tube	live_ stream	live_ audio	channel	channel_ set
/ad_set	X	X							
/closed_captions	X	X	X	X	X	X	X	X	X
/drm_attributes	X	X	X	X	X	X	X	X	X
/generated_preview_images	X	X	X						
/labels	X	X	X	X	X	X	X	X	X
/lineup									
/metadata	X	X	X	X	X	X	X	X	X
/player	X	X	X	X	X	X	X	X	X
/preview_image_files	X	X	X	X	X	X	X	X	X
/preview_image_urls	X	X	X	X	X	X	X	X	X
/primary_preview_image	X	X	X	X	X	X	X	X	X
/publishing_rule	X	X	X	X	X	X	X	X	X
/source_file_info	X	X	X						
/streams	X	X	X						
/upload_status	X	X	X		X				
/uploading_urls	X	X	X		X				
/youtube	X			X	X				
/replacement	X	X	X		X				

VIDEO AND AUDIO ASSETS

When you upload a video to Ooyala, Ooyala automatically transcodes it using the settings specified for your account and makes the videos available for playback through your player.

We recommend that you upload videos to Ooyala in H.264, but we also support many commonly used codecs. See *http://support.ooyala.com/users/documentation/concepts/chapter_encoding.html*.

For creating new video or audio assets, the sequence of steps is as as follows:

1. Create the asset with basic properties
2. Retrieve the "uploading URLs" from the system
3. For each "uploading URL" put the appropriate file chunk to the system

   Note: ***When you upload each chunk, do not sign the request***, as you would other requests. Simply PUT the chunk contents to the exact URLs. These URLs have already been signed for you.

COMMON ROUTES, ATTRIBUTES, PROPERTIES AND QUERY STRING PARAMETERS

Refer to the following information:

- *Common API routes* used by this asset type
- *Global definitions of attributes*

- *All properties by asset type*
- *Common query mechanism*
- *Adding metadata to an asset*
- *Adding closed captions*

GET UPLOADING URLS FOR NEWLY CREATED ASSET

```
[GET]/v2/assets/asset_id/uploading_urls
```

Note: ***When you upload each chunk, do not sign the request***, as you would other requests. Simply PUT the chunk contents to the exact URLs. These URLs have already been signed for you.

SET UPLOADING STATUS

```
[PUT] /v2/assets/asset_id/upload_status{
   "status":"uploaded"
}
```

VIEW STREAMS FOR A VIDEO

The term *streams* in this context means the internal-to-Ooyala processes that are started by the system to transcode your videos.

Note: You cannot control or manipulate these streams; they are controlled by the system. You can only view their names, not change them.

```
[GET]/v2/assets/asset_id/streams
```

VIEW SOURCE FILES FOR A VIDEO

```
[GET]/v2/assets/asset_id/source_file_info
```

GET URLS FOR AUTOGENERATED PREVIEW IMAGES

```
[GET] /v2/assets/asset_id/generated_preview_images
```

UPLOAD CUSTOM PREVIEW IMAGE

```
[POST]/v2/assets/asset_id/preview_image_files
<file_contents>
```

SET PRIMARY PREVIEW IMAGE CONFIGURATION

Set the type of the primary preview image of an asset to one of the following:

- `generated`: use the autogenerated preview image
- `uploaded_file`: use the uploaded custom preview image
- `remote_url`: URL for the preview image

```
[PUT]/v2/assets/asset_id/primary_preview_image
```


```
   "type":"generated"   | "uploaded_file" | "remote_url"
}
```

ABOUT METADATA

An asset can have up to 100 name/value pairs of custom metadata. A name is limited to 128 characters, and its value is limited to 2,048 characters.

EXAMPLES

This example creates a video:

```
[POST]/v2/assets{
   "name":"Honey Badger",
   "file_name":"honeybadger.avi",
   "asset_type":"video",
   "file_size":398422,
   "chunk_size":100000
}
```

Backlot returns a response similar to the following:

```
{
   "name":"Honey Badger",
   "preview_image_url":null,
   "asset_type":"video",
   "duration":0,
   "embed_code":embed_code,
   "created_at":"2011-10-26T19:40:32+00:00",
   "time_restrictions":null,
   "updated_at":"2011-10-26T19:40:32+00:00",
   "external_id":null,
   "hosted_at":null,
   "original_file_name":"honeybadger.avi",
   "description":null,
   "status":"uploading"
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example returns the uploading URLs:

```
[GET]/v2/assets/asset_id/uploading_urls
```

Backlot returns a response similar to the following:

```
["https://uploader-v2.ooyala.com/send?
filename=EwZ2RyMjrulwSBrJMpFd7xQ6FoftMnaA/0000000000000-0000000099999&filesize=100000&ex
Q0c5dVu4tDFUDIjIngyDu/LGng+rCG3KNSF4",
   "https://uploader-v2.ooyala.com/send?
filename=EwZ2RyMjrulwSBrJMpFd7xQ6FoftMnaA/0000000100000-0000000199999&filesize=100000&ex
   "https://uploader-v2.ooyala.com/send?
filename=EwZ2RyMjrulwSBrJMpFd7xQ6FoftMnaA/0000000200000-0000000299999&filesize=100000&ex
+bT8Jft9kvVdkYMND87BSs/VWu1r2imCNkl9v2++eTU",
```

```
   "https://uploader-v2.ooyala.com/send?
filename=EwZ2RyMjrulwSBrJMpFd7xQ6FoftMnaA/0000000300000-0000000398421&filesize=98422&exp
wuNCAjYABgxqdlfPaLeATOmsKxmdg" ]
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example marks the video as uploaded, which tells Backlot to start processing the file:

```
[PUT]/v2/assets/asset_id/upload_status{
   "status":"uploaded"
}
```

Backlot returns a response similar to the following:

```
{
   "status": "uploaded"
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

```
[GET]/v2/assets/asset_id/source_file_info
```

Backlot returns a response similar to the following:

```
{
   "original_file_name": lakeside.wmv,
   "source_file_url":source_url"
   "file_size": 25227886
}
```

Note: The URLs to the source files are signed and expire in 72 hours.

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

REMOTE ASSETS

A remote asset is a piece of content that you add to Backlot without uploading, processing, or storing the content with Ooyala.

For an introduction, see *Remote Assets*.

COMMON ROUTES, ATTRIBUTES, PROPERTIES AND QUERY STRING PARAMETERS

Refer to the following information:

- *Common API routes* used by this asset type
- *Global definitions of attributes*
- *All properties by asset type*

- *Common query mechanism*
- *Adding metadata to an asset*
- *Adding closed captions*

EXAMPLE

The following example creates a remote asset.

Note: Do not create a null `external_id`, that is, an `external_id` with no value ("") or a value of "`null`". Such null external IDs cannot be searched for later.

```
[POST]/v2/assets{
   "name":"ApiAutoV2_08",
   "stream_urls":{
      "flash":"http://MyDomain.com/MyRemoteAsset.mp4"
   },
 "asset_type":"remote_asset"
 "duration":120000,
 "external_id":"1234"
}
.
.
.
```

Backlot returns a response similar to the following:

```
{
   "asset_type":"remote_asset",
   "name":"ApiAutoV2_08",
   "preview_image_url":null,
   "duration":120000,
   "created_at":"2011-06-03T22:13:00+00:00",
   "embed_code":"embed_code",
   "stream_urls":{
      "ipad":null,
      "source_file":null,
      "flash":"http://MyDomain.com/MyRemoteAsset.mp4",
      "itunes":null,
      "iphone":null
   },
   "time_restrictions":null,
   "updated_at":"2011-06-03T22:13:00+00:00",
   "external_id":"1234",
   "description":null,
   "status":"live"
}
.
.
.
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

LIVE STREAMS

Live streaming is the delivery of video content in real time.

Live streams consist of content creation and encoding, content delivery through a CDN, and content consumption through Internet-connected devices. Backlot manages the set up, publishing rules, and monetization of your live streams.

To set up a live stream with Ooyala, you only need a video camera, an Internet connection, and an encoder. Popular encoders include hardware encoders such as Inlet Spinnaker HD and software encoders such as Flash Media Live (free).

If live streaming is not enabled on your account, contact Sales, your Customer Success Manager, or Technical Support.

This example creates a live stream with three encodings.

Note: This example uses the property `primary_encoder_ip`, which is for non-RTMP streams. For RTMP streams, the properties are `primary_entry_point` and `backup_entry_point`.

```
[POST]/v2/assets{
   "name":"Extreme Working at My Desk",
   "asset_type":"live_stream",
   "primary_encoder_ip":"8.8.8.8",
   "password":"encoder_password",
   "is_flash":true,
   "is_ios":false,
   "encodings":[
      {
         "width":800,
         "height":600,
         "bitrate":600
      },
      {
         "width":800,
         "height":600,
         "bitrate":300
      },
      {
         "width":400,
         "height":300,
         "bitrate":200
      }
   ]
}
```

Backlot returns a response similar to the following:

```
{
   "asset_type":"live_stream",
   "duration":0,
   "name":"Extreme Working at My Desk",
   "preview_image_url":null,
   "created_at":"2011-06-06T18:15:00+00:00",
   "embed_code":"ZnbWVpMjqToT1W_zYtCep5Ew0bEGmRgx",
   "time_restrictions":null,
   "updated_at":"2011-08-10T00:47:46+00:00",
   "external_id":null,
   "hosted_at":null,
   "original_file_name":null,
```

```
   "description":null,
   "is_flash":true,
   "is_ios":false,
   "status":"live",
   "encodings":[
      {
         "width":800,
         "height":600,
         "bitrate":300
      },
      {
         "width":800,
         "height":600,
         "bitrate":600
      }
   ]
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

YOUTUBE ASSETS

A YouTube video is a video hosted by YouTube and played through the Ooyala player.

This enables you to:

- Maintain a consistent look and feel across your site with the Ooyala player or your custom player.
- Leverage the use of Ooyala's rich APIs for an integrated and customized experience.
- Gain insight and maintain the benefits of Ooyala's leading analytics for all videos in your account.
- Discover supplemental content for your site from Youtube, the Internet's largest video library.
- Apply time, geographic, or domain-based rules to the content.
- Leverage Ooyala's intuitive and comprehensive tools to manage, syndicate, customize and analyze YouTube content along with your own content.

YouTube videos are sourced from YouTube, are not downloaded to your account, are not transcoded by Ooyala, and are not applied to your delivery or managed content allowances. As a result, all videos will contain a YouTube watermark and will be delivered using the best quality YouTube can provide based on the embed size of the player.

Because Backlot does not transcode YouTube videos, it does not generate any preview images and the content cannot be played back on iOS devices.

YouTube content might not always be available to the player. For example, videos might be removed, videos might be restricted based on YouTube publisher rules, or access to YouTube might be blocked by a viewer’s company or institution.

Note: No two YouTube assets in your account can have the same YouTube ID.

If you attempt to add a YouTube video that has been previously added, you receive the following error message, where *asset_id* identifies the previously created asset:

```
"{"duplicate_youtube_id":["asset_id"]}"
```

COMMON ROUTES, ATTRIBUTES, PROPERTIES AND QUERY STRING PARAMETERS

Refer to the following information:

- *Common API routes* used by this asset type
- *Global definitions of attributes*
- *All properties by asset type*
- *Common query mechanism*
- *Adding metadata to an asset*
- *Adding closed captions*

GET YOUTUBE SETTINGS

```
[GET]/v2/assets/asset_id/youtube
```

SET YOUTUBE SYNDICATION SETTINGS

```
[PUT]/v2/assets/asset_id/youtube
{
   "youtube_id":"some_youtube_id",
   "youtube_syndication_settings":{
      "private":true|false,
      "mobile":true|false,
      "embeddable":true|false,
      "keywords":comma-separated-list-of-keywords
   }
}
```

EXAMPLES

This example creates a YouTube asset:

```
[POST]/v2/assets{
   "name":"My YouTube Asset",
   "asset_type":"youtube",
   "youtube_id":"a1Y73sPHKxw"
}
```

Backlot returns a response similar to the following:

```
{
   "asset_type":"youtube",
   "duration":5,
   "name":"My YouTube Asset",
   "preview_image_url":null,
   "created_at":"2011-08-25T21:01:49+00:00",
   "embed_code":"1saGRyMjoOgjkUSjod_1TQINs_HOQnoU",
   "time_restrictions":null,
   "updated_at":"2011-08-25T21:01:50+00:00",
   "youtube_id":"a1Y73sPHKxw",
   "external_id":null,
   "hosted_at":null,
   "original_file_name":null,
   "description":"The best 5 second clip on the Internet.",
   "status":"live"
```


```
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example gets YouTube settings:

```
[GET]/v2/assets/1saGRyMjoOgjkUSjod_1TQINs_HOQnoU/youtube
```

Backlot returns a response similar to the following:

```
{
   "author":"cregets",
   "youtube_id":"a1Y73sPHKxw",
   "youtube_syndication_settings":{
      "private":null,
      "mobile":null,
      "embeddable":null,
      "keywords":null
   }
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example adds keywords to the YouTube syndication settings and makes the player for this video embeddable:

```
[PATCH]/v2/assets/1saGRyMjoOgjkUSjod_1TQINs_HOQnoU/youtube{
   "youtube_syndication_settings":{
      "private":null,
      "mobile":null,
      "embeddable":true,
      "keywords":[
         "old",
         "classic"
      ]
   }
}
```

Backlot returns a response similar to the following:

```
{
   "author":"cregets",
   "youtube_id":"a1Y73sPHKxw",
   "youtube_syndication_settings":{
      "private":null,
      "mobile":null,
      "embeddable":true,
      "keywords":[
         "old",
         "classic"
      ]
   }
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

CHANNEL AND CHANNEL SET ASSETS

A channel is a collection of videos in a single player. A channel set is a collection of channels.

COMMON ROUTES, ATTRIBUTES, PROPERTIES AND QUERY STRING PARAMETERS

Refer to the following information:

- *Common API routes* used by this asset type
- *Global definitions of attributes*
- *All properties by asset type*
- *Common query mechanism*
- *Adding metadata to an asset*
- *Adding closed captions*

LIST VIDEOS IN CHANNEL

```
[GET]/v2/assets/channel_id/lineup
```

SET OR REPLACE VIDEOS

```
[PUT]/v2/assets/channel_id/lineup[
   "embed_code","embed_code", ...
]
```

APPEND VIDEO TO CHANNEL

```
[PUT]/v2/assets/channel_id/lineup/embed_code
[PUT]/v2/assets/channel_id/lineup/external_id
```

DELETE VIDEO FROM A CHANNEL

```
[DELETE]/v2/assets/channel_id/lineup/embed_code
[DELETE]/v2/assets/channel_id/lineup/external_id
```

LIST CHANNELS IN CHANNEL SET

```
[GET]/v2/assets/channel_set_id/lineup
```

SET OR REPLACE CHANNELS

```
[PUT]/v2/assets/channel_set_id/lineup[
   "embed_code","embed_code", ...
]
```


APPEND CHANNEL TO CHANNEL SET

```
[PUT]/v2/assets/channel_set_id/lineup/embed_code
[PUT]/v2/assets/channel_set_id/lineup/external_id
```

DELETE CHANNEL FROM A CHANNEL SET

```
[DELETE]/v2/assets/channel_set_id/lineup/embed_code
[DELETE]/v2/assets/channel_set_id/lineup/external_id
```

EXAMPLES

This example creates a channel:

```
[POST]/v2/assets{
   "name":"My Channel",
   "asset_type":"channel"
}
```

Backlot returns a response similar to the following:

```
{
   "asset_type":"channel",
   "duration":0,
   "name":"My Channel",
   "preview_image_url":null,
   "created_at":"2011-08-10T17:45:24+00:00",
   "embed_code":"dvNTVxMjrRktSlb5v",
   "time_restrictions":null,
   "updated_at":"2011-08-10T17:45:24+00:00",
   "external_id":null,
   "hosted_at":null,
   "original_file_name":null,
   "description":null,
   "status":"live"
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example adds the `IzNnllMjphu2XF3` and `Y1dTdvMjq9QtOMG` video assets to the `dvNTVxMjrRktSlb5v` channel, replacing the current videos:

```
[PUT]/v2/assets/dvNTVxMjrRktSlb5v/lineup[
   "IzNnllMjphu2XF3",
   "Y1dTdvMjq9QtOMG"
]
```

Backlot returns a response similar to the following:

```
[
   "IzNnllMjphu2XF3",
   "Y1dTdvMjq9QtOMG"
]
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example adds the `BrdXVjMjrgtupU3M` video asset to the `dvNTVxMjrRktSlb5v` channel, without replacing the current videos:

```
[PUT]/v2/assets/dvNTVxMjrRktSlb5v/lineup/BrdXVjMjrgtupU3M
```

Backlot returns a response similar to the following:

```
[
   "BrdXVjMjrgtupU3M”,
   "IzNnllMjphu2XF3",
   "Y1dTdvMjq9QtOMG"
]
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example creates a channel set:

```
[POST]/v2/assets{
   "name":"My Channel",
   "asset_type":"channel_set"
}
```

Backlot returns a response similar to the following:

```
{
   "asset_type":"channel_set",
   "duration":0,
   "name":"My Channel",
   "preview_image_url":null,
   "created_at":"2011-08-10T17:45:24+00:00",
   "embed_code":"dvNTVxMjrRktSlb5v",
   "time_restrictions":null,
   "updated_at":"2011-08-10T17:45:24+00:00",
   "external_id":null,
   "hosted_at":null,
   "original_file_name":null,
   "description":null,
   "status":"live"
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example adds the `IzNnllMjphu2XF3` and `Y1dTdvMjq9QtOMG` channels to the `dvNTVxMjrRktSlb5v` channel set, replacing the current channels:

```
[PUT]/v2/assets/dvNTVxMjrRktSlb5v/lineup[
   "IzNnllMjphu2XF3",
   "Y1dTdvMjq9QtOMG"
]
```

Backlot returns a response similar to the following:

```
[
   "IzNnllMjphu2XF3",
   "Y1dTdvMjq9QtOMG"
]
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example adds the `BrdXVjMjrgtupU3M` channel to the `dvNTVxMjrRktSlb5v` channel set, without replacing the current channels:

```
[PUT]/v2/assets/dvNTVxMjrRktSlb5v/lineup/BrdXVjMjrgtupU3M
```

Backlot returns a response similar to the following:

```
[
   "BrdXVjMjrgtupU3M",
   "IzNnllMjphu2XF3",
   "Y1dTdvMjq9QtOMG"
]
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

CUSTOM METADATA

You can add up to 100 name-value pairs to any type of asset.

The metadata name must follow these restrictions:

- Maximum length: 128 characters
- Digits or letters (a to z, A to Z, 0 to 9) and any of the following special characters: - _ ! @ # $ % & ^ * ( ) . ?
- The following characters must be escaped, as shown:

Character	Escape Sequence
!	%21
#	%23
$	%24
^	%5E

The value is limited to 2,048 characters.

SET/REPLACE CUSTOM METADATA

```
[PUT]/v2/assets/asset_id/metadata{
   "name1" : "value1",
   "name2" : "value2",
   "name3" : "value3",
```

```
    .
    .
    .
   "name100" : "value100"
}
```

MODIFY CUSTOM METADATA

```
[PATCH]/v2/assets/asset_id/metadata{
   "name1" : "value1",
   "name2" : "value2",
   "name3" : "value3",
    .
    .
    .
   "name100" : "value100"
}
```

GET CUSTOM METADATA

```
[GET]/v2/assets/asset_id/metadata
```

DELETE A SINGLE CUSTOM METADATUM FOR PLAYER

```
[PATCH]/v2/assets/asset_id/metadata{
      "name" : "null"
}
```

DELETE ALL CUSTOM METADATA FOR ASSET

```
[DELETE]/v2/assets/asset_id/metadata
```

EXAMPLES

This example creates custom metadata for the `IzNnllMjphu2XF3_UgPROoCi9B2rtWs` asset:

```
[PUT]/v2/assets/IzNnllMjphu2XF3_UgPROoCi9B2rtWs/metadata{
   "director":"Michel Gondry",
   "year":"2004",
   "rating":"R"
}
.
.
.
```

Backlot returns a response similar to the following:

```
{
   "rating":"R",
   "director":"Michel Gondry",
   "year":"2004"
}
.
.
```


```
.
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

CLOSED CAPTIONS

You can add closed captions to videos, ads, remote assets, or YouTube videos.

Note: The default behavior of closed captions in the V3 Player is "Off". You can enable closed captions with the JavaScript function `setClosedCaptionsLanguage()`, described *here*.

UPLOAD CLOSED CAPTIONS

```
[PUT]/v2/assets/asset_id/closed_captions<dfxp_file_content>
```

GET CLOSED CAPTIONS

```
[GET]/v2/assets/asset_id/closed_captions
```

DELETE CLOSED CAPTIONS

```
[DELETE]/v2/assets/asset_id/closed_captions
```

EXAMPLES

This example adds closed captions to the `IzNnllMjphu2XF3_UgPROoCi9B2rtWs` asset:

```
[PUT]/v2/assets/IzNnllMjphu2XF3_UgPROoCi9B2rtWs/closed_captions
<dfxp_file_contents>
```

Backlot returns a 200 response.

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

ASSOCIATE AD SET WITH ASSET

You can associate ad sets with videos, live streams, and remote assets. For information about creating ad sets, refer to the Backlot User Guide.

ASSOCIATE AD SET

```
[PUT]/v2/assets/asset_id/ad_set/ad_set_id
```


GET AD SET FOR AN ASSET

```
[GET]/v2/assets/asset_id/ad_set
```

REMOVE AD SET FROM ASSET

```
[DELETE]/v2/assets/asset_id/ad_set
```

EXAMPLES

This example associates the `PROoCi9B2BwPROoCi9B2Bw123` ad set with the `IzNnllMjphu2XF3_UgPROoCi9B2rtWs` asset:

```
[PUT]/v2/assets/IzNnllMjphu2XF3_UgPROoCi9B2rtWs/ad_set/
PROoCi9B2BwPROoCi9B2Bw123
```

Backlot returns a 200 response.

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

ASSOCIATE PUBLISHING RULE WITH ASSET

You can associate publishing rules with videos, live streams, remote assets, YouTube videos, channels, and channel sets.

ASSOCIATE PUBLISHING RULE

```
[PUT]/v2/assets/asset_id/publishing_rule/publishing_rule_id
```

GET PUBLISHING RULE FOR AN ASSET

```
[GET]/v2/assets/asset_id/publishing_rule_id
```

REMOVE PUBLISHING RULE FROM AN ASSET

```
[DELETE]/v2/assets/asset_id/publishing_rule/publishing_rule_id
```

EXAMPLES

This example associates the `QtsjklrewsddROoCi9B2B45drs` publishing rule with the `IzNnllMjphu2XF3_UgPROoCi9B2rtWs` asset:

```
[PUT]/v2/assets/asset_id/publishing_rule/publishing_rule_id
```

Backlot returns a 200 response.

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

ASSOCIATE LABELS WITH ASSET

You can associate labels with videos, ads, live streams, remote assets, and YouTube videos.

ADD SINGLE LABEL TO ASSET

```
[PUT]/v2/assets/asset_id/labels/label_id
```

ADD MULTIPLE LABELS TO ASSET

The`POST` method adds labels to the asset, leaving untouched any labels previously added, whereas the `PUT` method replaces all labels already added to the asset; see the next section.

```
[POST] /v2/assets/asset_id/labels/[
  "label_id",
  "label_id",
  ...
]
```

REPLACE ALL LABELS FOR AN ASSET WITH NEW LABELS

```
[PUT] /v2/assets/asset_id/labels/[
  "label_id",
  "label_id",
  ...
]
```

VIEW LABELS FOR AN ASSET

```
[GET]/v2/assets/asset_id/labels
```

REMOVE SINGLE LABEL FROM AN ASSET

```
[DELETE]/v2/assets/asset_id/labels/label_id
```

REMOVE ALL LABELS FROM AN ASSET

```
[DELETE]/v2/assets/asset_id/labels
```


EXAMPLES

This example associates the `Qtsjklrewsdd ROoCi9B2B45drs` label with the `IzNnllMjphu2XF3_UgPROoCi9B2rtWs` asset:

```
[PUT]/v2/assets/IzNnllMjphu2XF3_UgPROoCi9B2rtWs/
labels/Qtsjklrewsdd ROoCi9B2B45drs ?api_key=<your API
 key>&expires=1930294539&signature=<signature>
```

Backlot returns a 200 response.

This example associates the `638aed2a18734375b3b332cb85999dd2` and `814efb109416490a98ee3f4fcd6784cf` labels with the `IzNnllMjphu2XF3_UgPROoCi9B2rtWs` asset:

```
[PUT]/v2/assets/IzNnllMjphu2XF3_UgPROoCi9B2rtWs/labels[
   "638aed2a18734375b3b332cb85999dd2",
   "814efb109416490a98ee3f4fcd6784cf"
]
```

Backlot returns a response similar to the following:

```
{
   "items":[
      {
         "name":"Sports",
         "id":"814efb109416490a98ee3f4fcd6784cf",
         "parent_id":null,
         "full_name":"/Sports"
      },
      {
         "name":"Riveting",
         "id":"638aed2a18734375b3b332cb85999dd2",
         "parent_id":null,
         "full_name":"/Riveting"
      }
   ]
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

ASSOCIATE PLAYER WITH ASSET

You can associate a player with videos, ads, live streams, remote players, and YouTube videos.

ADD PLAYER TO ASSET

```
[PUT]/v2/assets/asset_id/player/player_id
```

VIEW PLAYER FOR AN ASSET

```
[GET]/v2/assets/asset_id/player
```

EXAMPLES

This example associates the `b4f1a93b779145b4a9c8ef15f8502345` player with asset `IzNnllMjphu2XF3_UgPROoCi9B2rtWs`:

```
[PUT]/v2/assets/IzNnllMjphu2XF3_UgPROoCi9B2rtWs/player/
b4f1a93b779145b4a9c8ef15f8502345
```

This example retrieves all the players defined for the account making the request.

```
[GET]/v2/players
```

Backlot returns a response similar to the following:


```
{
   "items":[
      {
         "provider_homepage_url":"",
         "default_closed_caption_language":"",
         "scrubber":{
            "always_show":"false",
            "image_url":""
         },
         "related_videos":{
            "labels":[

            ],
            "click_behavior":"new_page",
            "order":"desc",
            "source":"shared_labels",
            "sort":"upload_time"
         },
         "id":"e18ab1da1813483499554ea2d8e67fbd",
         "watermark":{
            "alpha":"1.0",
            "image_url":"",
            "click_url":""
         },
         "playback":{
            "buffer_on_pause":"false"
         },
         "ooyala_branding":{
            "url_sharing":"true",
            "show_info_screen_homepage_link":"true",
            "show_end_screen_info_button":"true",
            "digg_sharing":"false",
            "show_embed_button":"true",
            "enable_error_screen":"true",
            "show_channel_button":"true",
            "show_info_screen_description":"false",
            "show_ad_countdown":"true",
            "show_info_button":"true",
            "show_share_button":"true",
            "show_bitrate_button":"false",
```

```
            "chromeless":"false",
            "twitter_sharing":"false",
            "email_sharing":"true",
            "show_volume_button":"true",
            "show_info_screen_title":"true",
            "accent_color":"#ffbb00",
            "show_end_screen_share_button":"true",
            "show_end_screen_more_videos":"true",
            "stumble_upon_sharing":"false",
            "facebook_sharing":"false",
            "show_info_screen_presented_by":"true",
            "show_end_screen_replay_button":"true",
            "show_end_screen_embed_button":"true"
         },
         "is_default":"true",
         "name":"Default Player",
         "up_next_preview":{
            "seconds_before_end":"",
            "timing":"disabled"
         },
         "provider_name":"",
         "audio":{
            "show_download":"false"
         }
      },
      {
         "provider_homepage_url":"",
         "default_closed_caption_language":"en",
         "scrubber":{
            "always_show":"false",
            "image_url":""
         },
         "related_videos":{
            "labels":[

            ],
            "click_behavior":"new_page",
            "order":"desc",
            "source":"shared_labels",
            "sort":"upload_time"
         },
         "id":"1b80572b5bd64648b17db6c515f551ab",
         "watermark":{
            "alpha":"0.44",
            "image_url":"image_url",
            "click_url":"",
            "position":"bottom-right"
         },
         "playback":{
            "buffer_on_pause":"false"
         },
         "ooyala_branding":{
            "url_sharing":"false",
            "show_info_screen_homepage_link":"true",
            "show_end_screen_info_button":"false",
            "digg_sharing":"false",
            "show_embed_button":"true",
            "enable_error_screen":"true",
            "show_channel_button":"false",
            "show_info_screen_description":"false",
            "show_ad_countdown":"false",
            "show_info_button":"true",
            "show_share_button":"true",
            "show_bitrate_button":"false",
```


```
            "chromeless":"false",
            "twitter_sharing":"false",
            "email_sharing":"true",
            "show_volume_button":"true",
            "show_info_screen_title":"true",
            "accent_color":"#ff00",
            "show_end_screen_share_button":"false",
            "show_end_screen_more_videos":"false",
            "stumble_upon_sharing":"false",
            "facebook_sharing":"false",
            "show_info_screen_presented_by":"true",
            "show_end_screen_replay_button":"true",
            "show_end_screen_embed_button":"false"
         },
         "is_default":"false",
         "name":"alex player",
         "up_next_preview":{
            "seconds_before_end":"",
            "timing":"disabled"
         },
         "provider_name":"",
         "audio":{
            "show_download":"false"
         }
      }
   ]
}
```


QUERY CONSTRUCTION

To limit the amount of data returned in a GET call, you can specify `where` queries.

Note:

Flight times (date/time restrictions with the `start_date`, `end_date`, `start_time`, and `end_time` properties) might also be restricted by publishing rules. However, the Query API looks at flight time restrictions only of the asset itself.

Because assets with no date restrictions are playable during a specified period of time, they are also returned in the results.

You cannot query for null values.

Note: Make sure non-ASCII letters and **'** characters are escaped.

QUERY STRING PARAMETERS

The following table describes all parameters that can be used on the query string.

Parameter	Description	Required?
`and`	For querying for intersections of labels. **Note:** Make sure non-ASCII letters and **'** characters are escaped. Type: String Default: none	No

Parameter	Description	Required?
	Example: [GET] /v2/assets?where=labels INCLUDES 'Sports' AND labels INCLUDES 'Case Study'	
`include`	Returns additional information about the asset. Type: String Default: name Valid Values: metadata | labels Example: [GET] /v2/assets?include=metadata,labels	No
`limit`	The maximum number of results to return. If you receive more results than the value you specify, you can get the next page of results with the paging token. Type: String Default: 100 Valid Values: 1-500 Example: [GET] /v2/assets?limit=5	No
`next_page`	A URL returned by the previous API request that enables you to page through results. **Note:** If the number of results for a given page is divisible by the value of `limit`, the service returns a 404 response. Type: String Default: None Example: [GET] /v2/assets?limit=2&page_token=1332345124%3B0wZTQ4NDpW27FQgXF6zry0b42OI76GZN	No
`orderby`	The field to sort on. Type: String Default: created_at descending order Valid Values: name | duration | asset_type | created_at Examples: [GET] /v2/assets?orderby=name, [GET] /v2/assets?orderby=name+DESCENDING	No
`where`	The where clause returns results that meet the specified criteria. **Note:** You cannot search for properties with null values. **Note:** Make sure non-ASCII letters and **'** characters are escaped. Type: String Default: name Example: [GET] /v2/assets?where=description='cat video'	No

ABOUT PATTERN MATCHING AND RESULTS

The query returns a result for each item that contains all terms. Queries are case-insensitive, return singular and plural versions of the terms, and exclude common words such as "a" and "the" from the search. For example, `where=description='Cat videos'` returns:

- "Videos about cats"
- "Funny video of a cat"
- "Cats watching videos"

Note: Deleted assets are never returned in query results.

COMPARISONS

All comparison functions are supported for integer and string fields on assets. (<, <=, =, >=, >, !=). For example, `duration>600` returns all assets over 10 minutes (600 seconds).

INTERSECTION OF LABELS

To find assets that have two or more labels in common, use the `and` operator. For example, to find assets that have both the label `Sports` and the label `Baseball`:

```
[GET]/v2/assets?where=labels INCLUDES 'Sports' AND labels INCLUDES
 'Baseball'
```

Note: We recommend that you use no more than 3 'labels INCLUDES' in the call.

EXAMPLES

This example returns the first two assets that have "cat" and "funny" in the description, and are more than 10 minutes long:

```
[GET]/v2/assets?where=description='cat funny'+AND
+duration>600&orderby=name&limit=2
```

Backlot returns a response similar to the following:

```
{
   "items":[
      {
         "asset_type":"video",
         "duration":19000,
         "name":"Funny Cats",
         "preview_image_url":"http://ak.c.ooyala.com/
IzNnllMjphu2XF3_UgPROoCi9B2BwtSg/Ut_HKthATH4eww8X5hMDoxOjBrOw-uIx",
         "created_at":"2011-04-25T15:32:57+00:00",
         "embed_code":"IzNnllMjphu2XF3_UgPROoCi9B2Bwqwt",
         "time_restrictions":null,
         "updated_at":"2011-08-29T23:12:09+00:00",
         "external_id":null,
         "hosted_at":null,
         "original_file_name":"Movie on 2011-04-25 at 08.31.mov",
         "description":"A video of funny cats.",
         "status":"live"
      },
      {
         "asset_type":"video",
```

```
         "duration":25959,
         "name":"More Funny Cats",
         "preview_image_url":"http://ak.c.ooyala.com/9hcDdkMjq4LUKvPP-
vJcgDds_TPIsP0u/Ut_HKthATH4eww8X5hMDoxOjBrOw-uIx",
         "created_at":"2011-04-05T21:49:46+00:00",
         "embed_code":"9hcDdkMjq4LUKvPP-vJcgDds_TPIsrte",
         "time_restrictions":null,
         "updated_at":"2011-06-01T23:13:40+00:00",
         "external_id":null,
         "hosted_at":null,
         "original_file_name":"Movie on 2011-04-05 at 14.44.mov",
         "description":"Can there really be more funny cat videos? Yes.",
         "status":"live"
      }
   ],
   "next_page":"/v2/assets?
limit=2&page_token=1314306109%3B1saGRyMjoOgjkUSjod_1TQINs_HOQnoU"
}
```

This example gets a list of assets that match the specified embed codes:

```
/v2/assets?where=embed_code+IN
+('IzNnllMjphu2XF3_UgPROoCi9B2Bwqwt','9hcDdkMjq4LUKvPP-vJcgDds_TPIsrte')
```

Backlot returns a response similar to the following:

```
{
   "items":[
      {
         "asset_type":"video",
         "duration":19000,
         "name":"Funny Cats",
         "preview_image_url":"http://ak.c.ooyala.com/
IzNnllMjphu2XF3_UgPROoCi9B2BwtSg/Ut_HKthATH4eww8X5hMDoxOjBrOw-uIx",
         "created_at":"2011-04-25T15:32:57+00:00",
         "embed_code":"IzNnllMjphu2XF3_UgPROoCi9B2Bwqwt",
         "time_restrictions":null,
         "updated_at":"2011-08-29T23:12:09+00:00",
         "external_id":null,
         "hosted_at":null,
         "original_file_name":"Movie on 2011-04-25 at 08.31.mov",
         "description":"A video of funny cats.",
         "status":"live"
      },
      {
         "asset_type":"video",
         "duration":25959,
         "name":"More Funny Cats",
         "preview_image_url":"http://ak.c.ooyala.com/9hcDdkMjq4LUKvPP-
vJcgDds_TPIsP0u/Ut_HKthATH4eww8X5hMDoxOjBrOw-uIx",
         "created_at":"2011-04-05T21:49:46+00:00",
         "embed_code":"9hcDdkMjq4LUKvPP-vJcgDds_TPIsrte",
         "time_restrictions":null,
         "updated_at":"2011-06-01T23:13:40+00:00",
         "external_id":null,
         "hosted_at":null,
         "original_file_name":"Movie on 2011-04-05 at 14.44.mov",
         "description":"Can there really be more funny cat videos? Yes.",
         "status":"live"
      }
   ]
}
```


This example gets a list of assets that have a value of the `category` metadata key equal to "Cat Videos" :

```
[GET]/v2/assets?where=metadata.category='Cat Videos'&include=metadata
```

Backlot generates a feed similar to the following:

```
{
   "items":[
      {
         "asset_type":"video",
         "duration":19000,
         "name":"Funny Cats",
         "preview_image_url":"http://ak.c.ooyala.com/
IzNnllMjphu2XF3_UgPROoCi9B2BwtSg/Ut_HKthATH4eww8X5hMDoxOjBrOw-uIx",
         "created_at":"2011-04-25T15:32:57+00:00",
         "embed_code":"IzNnllMjphu2XF3_UgPROoCi9B2Bwqwt",
         "time_restrictions":null,
         "updated_at":"2011-08-29T23:12:09+00:00",
         "external_id":null,
         "metadata":{
            "category":"Cat Videos"            "rating":"G"
         },
         "hosted_at":null,
         "original_file_name":"Movie on 2011-04-25 at 08.31.mov",
         "description":"A video of funny cats.",
         "status":"live"
      },
      {
         "asset_type":"video",
         "duration":25959,
         "name":"More Funny Cats",
         "preview_image_url":"http://ak.c.ooyala.com/9hcDdkMjq4LUKvPP-
vJcgDds_TPIsP0u/Ut_HKthATH4eww8X5hMDoxOjBrOw-uIx",
         "created_at":"2011-04-05T21:49:46+00:00",
         "embed_code":"9hcDdkMjq4LUKvPP-vJcgDds_TPIsrte",
         "time_restrictions":null,
         "updated_at":"2011-06-01T23:13:40+00:00",
         "external_id":null,
         "metadata":{
            "category":"Cat Videos"            "rating":"PG"
         },
         "hosted_at":null,
         "original_file_name":"Movie on 2011-04-05 at 14.44.mov",
         "description":"Can there really be more funny cat videos? Yes.",
         "status":"live"
      }
   ]
}
```

This example returns all videos with the "funny" label:

```
[GET]/v2/assets?where=labels+INCLUDES+'funny'
```

Backlot generates a feed similar to the following:

```
{
   "items":[
      {
         "asset_type":"video",
         "duration":19000,
         "name":"Funny Cats",
```


```
         "preview_image_url":"http://ak.c.ooyala.com/
IzNnllMjphu2XF3_UgPROoCi9B2BwtSg/Ut_HKthATH4eww8X5hMDoxOjBrOw-uIx",
         "created_at":"2011-04-25T15:32:57+00:00",
         "embed_code":"IzNnllMjphu2XF3_UgPROoCi9B2Bwqwt",
         "time_restrictions":null,
         "updated_at":"2011-08-29T23:12:09+00:00",
         "external_id":null,
         "hosted_at":null,
         "original_file_name":"Movie on 2011-04-25 at 08.31.mov",
         "description":"A video of funny cats.",
         "status":"live"
      },
      {
         "asset_type":"video",
         "duration":25959,
         "name":"More Funny Cats",
         "preview_image_url":"http://ak.c.ooyala.com/9hcDdkMjq4LUKvPP-
vJcgDds_TPIsP0u/Ut_HKthATH4eww8X5hMDoxOjBrOw-uIx",
         "created_at":"2011-04-05T21:49:46+00:00",
         "embed_code":"9hcDdkMjq4LUKvPP-vJcgDds_TPIsrte",
         "time_restrictions":null,
         "updated_at":"2011-06-01T23:13:40+00:00",
         "external_id":null,
         "hosted_at":null,
         "original_file_name":"Movie on 2011-04-05 at 14.44.mov",
         "description":"Can there really be more funny cat videos? Yes.",
         "status":"live"
      }
   ]
}
```

Flight times restrict when content can be played. This example returns assets that were playable for the entirety of June 2011:

```
[GET]/v2/assets?
where=time_restrictions.start_date<='2011-06-01T00:00:00Z'+AND
+time_restrictions.end_date>='2011-07-01T00:00:00Z'
```

Backlot returns a response similar to the following:

```
{
   "items":[
      {
         "asset_type":"video",
         "duration":3366,
         "name":"My movie",
         "preview_image_url":"http://ak.c.ooyala.com/
B5bzAzMzrrnbUrxt_vBj0hLcjgdQSkSH/Ut_HKthATH4eww8X5hMDoxOjBrOw-uIx",
         "created_at":"2011-01-21T23:37:28+00:00",
         "embed_code":"B5bzAzMzrrnbUrxt_vBj0hLcjgdQSkSH",
         "time_restrictions":{
            "type":"range",
            "start_date":"2011-06-01T00:00:00Z",
            "end_date":"2011-06-15T00:00:00Z"
         }         "updated_at":"2011-11-29T00:40:37+00:00",
         "external_id":null,
         "original_file_name":"My movie.mov",
         "hosted_at":null,
         "description":null,
         "status":"live"
      },
      {
```

```
         "asset_type":"video",
         "duration":0,
         "name":"My other movie",
         "preview_image_url":null,
         "created_at":"2011-01-01T19:40:32+00:00",
         "embed_code":"QycmV4Mjp_iICcP6fhpiHsY81jDgmOUY",
         "time_restrictions":{
            "type":"range",
            "start_date":"2011-06-015T00:00:00Z",
            "end_date":null
         }         "updated_at":"2011-10-26T19:40:32+00:00",
         "external_id":null,
         "original_file_name":"My other movie.avi",
         "hosted_at":null,
         "description":null,
         "status":"uploading"
      }
   ]
}
```

DELETED ASSETS

Use this request to get information about assets that have been deleted.

In the response, assets are listed in order of deletion date (`deleted_at` property), so that most recently deleted assets are first.

Results are grouped 500 assets per page. Use the `next_page` value at the end of each response to retrieve the next set of 500 assets. See example below.

LIST ALL DELETED ASSETS

```
[GET]/v2/deleted_assets
```

EXAMPLES

This example returns information about all deleted assets, ordered by most recently deleted:

```
[GET]/v2/deleted_assets
```

Backlot returns a response similar to the following:

```
{
   "items":[
      {
         "description":null,
         "name":"Bogus",
         "embed_code":"FpY3Q0NalexlangerulescClLUSeTVkeSnfS",
         "provider_id":"47xxx4",
         "created_at":"2012-11-21T16:53:42Z",
         "deleted_at":"2012-12-12T21:41:47Z",
         "duration":"0",
         "asset_type":"ad"
      },
      {
         "description":"",
         "name":"bleebleblobble 2.mov",
```

```
         "embed_code":"xkdG43NjrlM92YJ1HredezaqiWQ3SZpO5da7",
         "provider_id":"47xxx4",
         "created_at":"2012-10-16T16:15:17Z",
         "deleted_at":"2012-11-16T20:00:48Z",
         "duration":"5038",
         "asset_type":"ad"
      },
      {
         "description":"description",
         "name":"bleebleblobble.mov",
         "embed_code":"tyZnc0NjrAq_ZsvJEqesedsIQxAAlvccE35m",
         "provider_id":"47xxx4",
         "created_at":"2012-10-10T15:33:01Z",
         "deleted_at":"2012-10-10T15:33:42Z",
         "duration":"0",
         "asset_type":"video"
      }
   ]   "next_page":"/v2/assets?
limit=2&page_token=1314306109%3B1saGRyMjoOgjkUSjod_1TQINs_HOQnoU"
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

DRM ATTRIBUTES FOR REMOTE ASSETS (INCLUDING LIVE STREAMS)

For remote assets protected by Digital Rights Management (DRM) systems, you need to associate information about that system.

Note: These steps are applicable to live linear (assets not packaged by Ooyala).

Assets that are transcoded by and stored on the Ooyala system have the DRM attributes applied automatically, but remote assets do not pass through Ooyala transcoding and thus must be updated by you. For example, for live encoders this has to be updated by someone configuring the encoder (such as you or your vendor's technical support team).

For every remote asset that is protected by licenses issued from DRM systems operated by Ooyala, you need to set some attributes pertinent to the type of DRM system in use (FairPlay Streaming, Widevine or PlayReady).

- **FairPlay Streaming (FPS)** requires a content key and iv. You can optionally set a lease or rental duration.
- **Widevine** needs an *id* (explained below). Widevine should provide your service information, which contains all the necessary parameters needed for packaging your media (id, IV, Key).
- **PlayReady** needs the license acquisition URL, `content_key`, and `key_id` (explained below).
- To configure DRM with Adobe Access, contact your CSM to set up Adobe Access certificates for your live encoders.

Note: For PlayReady-protected remote assets (when `drm_type` is `playready` or `playready_hls` on the request), you need to set these attributes *before* acquiring a license from PlayReady; otherwise, the remote asset under the protection of the DRM cannot be played.

Note: The DRM attributes used for PlayReady are key_id and content_key. The key id is an identifier associated with the content. It can be any 16-byte value expressed in base64 or hex format. The content key should be a random 16-byte value that is used as the AES key to encrypt the content. This may also be stored in base64 or hex format. You must check if your encoder requires the key id and content key to

be in the base64 format or hex format. When you create a key for the first time, you don't need to use a version. In this case your content key field will be content_key and the license URL would be:

http://player.ooyala.com/sas/drm2/jqbkH9m/UPHGNsr/playready/ooyala.

Ooyala allows you to associate multiple content keys with an asset. You can do this by naming the content key field as content_key_version where version is a number that needs to match the version in the license URL. For example, if the content key field is content_key_2, then the license URL would need to end with a 2: .../playready/ooyala/version/2. You may associate multiple keys with an asset by using a different version number each time. See example below.

SET/REPLACE FPS ENCRYPTION KEY

Create a new FPS encryption key:

```
[POST] /v2/assets/asset_id/drm_attributes/fps
```

Response:

```
{
  "drm_type": "fps",
  "fps_content_key": "base64 encoded content key",
  "fps_content_key_iv": "base64 encoded iv"
}
```

The `fps_content_key` field contains a base64-encoded, 128-bit key that can be used for performing AES encryption on the video. The `fps_content_key_iv` field is a base64-encoded, 128-bit value to use as an initialization vector for encrypting the content.

Sample response:

```
{
  "drm_type": "fps",
  "fps_content_key": "5HdVooYGEROm+LX2NJBDZg==",
  "fps_content_key_iv": "rOw7gQauk7RCGVi1aeP9QQ=="
}
```

The first time you create a key, its value will be returned unmasked. Later on, if you create another key, the value of any existing key is returned masked.

To associate an existing FPS encryption key with an asset:

```
[PATCH] /v2/assets/asset_id/drm_attributes/fps
```

If you have multiple keys associated with the asset, include the version for this key in the body. For example, if this is the third key used for this asset, you will send the following JSON body with a key of 2QCEebexS0G8+3jP/pM7TA== and a version number of 3:

```
{ "fps_content_key_3" : "2QCEebexS0G8+3jP/pM7TA==" }
```

The response indicates that key 3 was stored successfully and with the key value masked:

```
{
  "fps_content_key_3": "*****",
  "drm_type": "fps"
}
```

To create versioned keys, use the POST route.

```
[POST] /v2/assets/:asset_id/drm_attributes/fps?version=1
```

Sample response:

```
{
  "fps_content_key" : "****",
  "fps_content_key_1" : "12s213" ,
  "fps_content_key_iv_1" : "daiosd"
}
```

To change a duration for a rental from the default (3,600 seconds) or to set a lease duration:

```
[PATCH] /v2/assets/asset_id/drm_attributes/fps
```

The body is:

```
{
  "fps_lease_duration" : lease duration in number of seconds,
  "fps_rental_duration" : rental duration in number of seconds
}
```

For example, to set a lease duration to 18,0000 seconds (5 hours) and the rental duration to 6,000 seconds:

```
{
   "fps_lease_duration": 18000,
   "fps_rental_duration" : 6000
}
```

SET/REPLACE WIDEVINE ATTRIBUTE FOR DRM PROTECTION OF REMOTE ASSET

`id` is generated by the encoder and needs to be passed to Ooyala through the API shown below. Ooyala stores the id in Ooyala systems. Work with Widevine to retrieve your `id`. If you are unable to retrieve the `id` from Widevine, your CSM can work with you and the encoder to get these values configured in the encoder.

```
[PATCH]/v2/assets/asset_id/drm_attributes/widevine{
   "id":"value1"
}
```

SET/REPLACE PLAYREADY ATTRIBUTES FOR DRM PROTECTION OF REMOTE ASSET

If you need to use a non-Ooyala system to generate the content key, you may use the following route for PlayReady Smooth to associate the key and key id with the Ooyala asset.

Note: The API route you need to use is different when managing attributes for PlayReady HLS and PlayReady Smooth. For PlayReady Smooth use the endpoint /v2/assets/asset_id/drm_attributes/playready, as shown below. For PlayReady HLS use the endpoint /v2/assets/asset_id/drm_attributes/playready_hls.

```
[PATCH]/v2/assets/asset_id/drm_attributes/playready{
   "key_id":"value1"
   "content_key":"value2"
}
```

If you have multiple keys associated with the asset, include the version for this key in the body. For example, if this is the second key used for this asset, you will send the following body:

```
{
   "key_id":"value1"
   "content_key_1":"value2"
}
```

GENERATING A PLAYREADY KEY_ID AND CONTENT_KEY

This route generates a random key id and content key for the asset and associates them with the video. You do not need to call the PATCH route described above to store these attributes in Ooyala's datastore if you get your key id and content key using this route. The JSON response contains the key id in base64 as well as guid/hex formats. The response also contains the content_key in base64 encoded format. drm_type can be specified as playready or playready_hls.

Note: This route will only work over https. API calls made over plain http will be rejected because the encryption key is sensitive information.

```
[POST] /v2/assets/asset_id/drm_attributes/drm_type
```

This will return:

```
{
   "key_id":"base64 encoded key id",
   "key_id_guid":"hex version of the key id",
   "drm_type":"playready",
   "content_key":"base64 encoded content key"
}
```

To create another key for the same asset, you can call the same route with the version set to any number.

```
[POST] /v2/assets/asset_id/drm_attributes/drm_type?version=2
```

This will return:

```
{
   "key_id":"base64 encoded key id",
   "key_id_guid":"hex version of the key id",
   "drm_type":"playready",
   "content_key_2":"base64 encoded content key",
   "content_key":"***"
}
```

Notice the old `content_key` is not returned. Its value is masked with ***. This is for security reasons.

GET DRM ATTRIBUTES

```
[GET] /v2/assets/asset_id/drm_attributes/drm_type
```

LICENSE ACQUISITION URL

You need to put the license acquisition URL into the encoder. If you are unable to do so, your CSM can work with you to get this value configured in the encoder. Ooyala stores the license acquisition URL in Ooyala systems.

```
https://player.ooyala.com/sas/drm2/pcode/asset_id/playready/ooyala
https://player.ooyala.com/sas/drm2/pcode/asset_id/playready_hls/ooyala
```

EXAMPLE

This example creates DRM attributes for the `IzNnoCi9B2rtWs` asset for a live encoder. The license acquisition URL for this asset is as follows:

```
https://player.ooyala.com/sas/drm2/IzNnoCi9B2rtWs/asset_id/playready/ooyala
```

Backlot returns a response similar to the following:

```
{
   "content_key":"*****",
   "drm_type":"playready",
   "key_id":"1234"
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

PROPERTIES

The following table describes the properties you need to set for each DRM type.

drm_type	Property	Description
fps	`fps_content_key_`*version*	The FairPlay Streaming identifier for the asset.
playready	`key_id`	The key_id is an identifier associated with the content. It can be any 16-byte value expressed in base64 or hex format. Type: String Default: None Example in base64 - V/YqH723UV48kjRlUzyqww== Example in hex - 1f2af657b7bd5e513c923465533caac3
playready	`content_key_version`	The PlayReady content key. The *version* of the parameter name is an integer that matches the version in the license acquisition URL for the specific asset.

drm_type	Property	Description
		For example, if the license acquisition URL is: `https://player.ooyala.com/sas/drm2/jqbkH9m/UPHGNsr/playready/ooyala/version/3` Then the corresponding parameter name is `content_key_3`. Type: String Default: None Example in base64 - V/YqH723UV48kjRlUzyqww== Example in hex - 1f2af657b7bd5e513c923465533caac3
widevine	`id`	The Widevine identifier for the asset Type: String Default: None Example: "1234"

PARAMETERS

The following table describes the parameters of the routes.

Parameter	Description	Required?
`asset_id`	The unique identifier for an asset. Type: String	Yes
`pcode`	Your provider code.	Yes
`version`	The number identifying the version to use if multiple versions exist. Type: Int	No
`fps_rental_duration`	Number of seconds for which the content key in the CKC is valid. If this is not set, a default value of 3600 (1 hour) is used. The player will not start the decryption if the content key has expired. However, the player continues the user experience if the content key expires during the playback. When started again with an	No

Parameter	Description	Required?
	expired key, the player declines the playback.	
`fps_lease_duration`	Similar to `fps_rental_duration` except that if the key expires during playback, the playback will stop immediately. There is no default value for this, so if not set, no lease duration is enforced. The player may keep sending CKC requests to the server during playback to renew the lease periodically.	No

LABELS

When added to external publishing targets, labels specify which videos, channels, and channel sets to publish externally. Additionally, labels are also useful for organizing your video library, searching for videos, and retrieving targeted analytics.

CREATE A LABEL

```
[POST]/v2/labels{
   properties
}
```

CREATE MULTIPLE LABELS USING FULL LABEL NAME

The full paths of labels must be CGI-escaped (URL-encoded) and must begin with a leading slash. All components of the path are created.

Note:

If a label specified in a request already exists, the system returns the identifier and other information about that pre-existing label. For example, if your request contains two new labels and two that already exist, the two new labels are created. The identifiers and other information for all four labels are returned in the response.

If there are multiple labels with the same full path name (for example, two labels with the exactly same full path name "/a/b/c"), and you try to create a label with the full path name "/a/b/c/d", the system responds with error 400 because it does not know which "/a/b/c" to use.

```
[POST]/v2/labels/by_full_path//full/path/label1,/full/path/label2,...
```

LIST LABELS

```
[GET]/v2/labels
```

LIST ALL TOP-LEVEL LABELS

```
[GET]/v2/labels?is_root=true
```

GET ASSETS ASSIGNED TO A LABEL

```
[GET]/v2/labels/label_id/assets
```

VIEW LABEL INFORMATION

```
[GET]/v2/labels/label_id
```


VIEW ALL SUB-LABELS ASSOCIATED WITH A LABEL

```
[GET]/v2/labels/label_id/children
```

VIEW LABEL INFORMATION BY FULL PATH NAMES

The full paths of labels must be CGI-escaped (URL-encoded). The services returns 400 if any paths do not exist. The response body states which components of the paths already exist or are missing.

```
[GET]/v2/labels/by_full_path//full/path/label1,/full/path/label2
```

DELETE A LABEL

```
[DELETE]/v2/labels/label_id
```

ROUTE ATTRIBUTES

The following table describes all attributes that can be expressed through the route.

Route Attribute	Description
label_id	The ID of the label. To get a list of labels, perform a get against the `/v2/labels` route. Type: String Default: None Example: /labels/ r28ertfe44ea4e648fd07d4509123254 b

QUERY STRING PARAMETERS

The following table describes all parameters that can be expressed through the route.

Parameter	Description
is_root	Retreives all top-level labels. Type: Boolean Default: If is_root is not specified, default value is false. Example: /v2/labels?is_root=true

PROPERTIES

The following table describes all properties that can be associated with a label.

Property	Description	Required?
`asset_count`	Read-only. Specifies the number of assets using the label. Returned when you make a get request against a specific label ID. Type: Integer Example: 42	No
`full_name`	Full path to the label, including any parents. Type: String Example: "/Sports/Motorcycle Racing"	Yes
`id`	Read only. ID of a label. Type: String Example: "814efb109416490a98ee3f4fcd673244"	Yes
`items`	Array containing labels. Returned by a get request to `/v2/labels`. Type: Array Example: n/a	No
`name`	Name of a label. To use non-Latin characters, make sure they are UTF-8 and URI-encoded. Type: String Example: "Motorcycle Racing"	Yes
`parent_id`	ID of the parent label. Type: String Condition: Required when adding a child label to a parent label. Example: "Sports"	Conditional

BEST PRACTICES FOR LABELS

The following best practices will ensure that you have optimal performance and speed in Theme Builder.

- Limit the use of excess labels in your Backlot account as a low this will keep load times low in Theme Builder.
- Create broad and categorical labels as labels that apply to only 1 or 2 movies will result in many labels that are of limited usefulness.

EXAMPLES

This example creates a top-level label:

```
[POST]/v2/labels{
   "name":"Hobbies"
}
```

Backlot returns a response similar to the following:

```
{
   "name":"Hobbies",
   "id":"d5751b77a0c24972888bf906734d4522",
   "full_name":"/Hobbies",
   "parent_id":null
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example adds a child label to a parent label:

```
[POST]/v2/labels{
   "name":"Hockey",
   "parent_id":"d5751b77a0c24972888bf906734d4522"
}
```

Backlot returns a response similar to the following:

```
{
   "name":"Hockey",
   "id":"85042f300fc143c093e8f4ee01894355",
   "full_name":"/Hobbies/Hockey",
   "parent_id":"d5751b77a0c24972888bf906734d4522"
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example creates multiple labels, specified by their full path names, in a single call.

```
[POST]/v2/labels/by_full_path//sports/football/NFL,/animal videos/buffaloes
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example lists all labels:

```
[GET]/v2/labels
```

Backlot returns a response similar to the following:

```
{
   "items":[
      {
         "name":"Sports",
         "id":"814efb109416490a98ee3f4fcd6784cf",
         "parent_id":null,
         "full_name":"/Sports"
      },
      {
         "name":"Motorcycle Racing",
         "id":"bace921fdea44cc18a5a273155514522",
         "parent_id":"814efb109416490a98ee3f4fcd673244",
```

```
            "full_name":"/Sports/Motorcycle Racing"
         },
         {
            "name":"Hockey",
            "id":"85042f300fc143c093e8f4ee01894355",
            "parent_id":"d5751b77a0c24972888bf906734d8c34",
            "full_name":"/Hobbies/Hockey"
         },
         {
            "name":"Hobbies",
            "id":"d5751b77a0c24972888bf906734d4522",
            "parent_id":null,
            "full_name":"/Hobbies"
         }
      ]
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example deletes a label:

```
[DELETE]/v2/labels/85042f300fc143c093e8f4ee01894355
```

Backlot returns a 200 response.

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

PLAYERS (BACKLOT API)

PLAYERS

When you create a player and assign it to one or more assets in your account, it defines the playback options and the look and feel of your content.

Note: To change the default player (see `is_default`), simply specify the new player you want as default. Do not try to "unset" the earlier default.

CREATE A PLAYER

```
[POST]/v2/players{
   properties
}
```

LIST PLAYERS

```
[GET]/v2/players
```

VIEW PLAYER INFORMATION

```
[GET]/v2/players/player_id
```

DELETE A PLAYER

```
[DELETE]/v2/players/player_id
```

ROUTE ATTRIBUTES

The following table describes all attributes that can be expressed through the route.

Route Attribute	Description
player_id	The ID of the player. To get a list of players, perform a get against the `/players` route. Type: String Default: None Example: /players/ b4f1a93b779145b4a9c8ef15f8502345

PROPERTIES

The following table describes all properties that can be associated with a player. For an introduction to player visual components from an end user perspective, see *Player V3 Components*.

Property	Description	Required?
`accent_color`	Specifies the accent color in the web color format. Type: String Default: ffbb00 Example: "66FF33" Parent: ooyala_branding	No
`alpha`	Specifies the transparency of the watermark. Type: String Default: None Example: 1.0 Valid Values: 0.0 to 1.0 Parent: watermark	No
`always_show`	Shows the scrubber bar during video playback. Type: Boolean Valid Values: true | false Default: false Example: "false" Parent: scrubber	No
`buffer_on_pause`	Causes the video to buffer when paused. This is useful for users with slow network connections. Type: Boolean Valid Values: true | false Default: false Example: "false" Parent: playback	No
`click_behavior`	Specifies how the related video opens when its thumbnail is clicked. Type: String Valid Values: same_player | same_page | new_page Default: same_player Conditions: Required if related_videos is specified.	Conditional

Property	Description	Required?
	Example: "same_player" Parent: related_videos	
`click_url`	The URL to which the user is directed when the watermark is clicked. Type: String Default: None Example: "www.mywebsite.com" Parent: watermark	No
`default_closed_caption_language`	Specifies the default language for closed captions. Type: String Default: None Example: "true"	No
`digg_sharing`	Enables Digg sharing. If enabled, show_share_button must be set to true. Type: Boolean Valid Values: true | false Default: false Example: "true" Parent: ooyala_branding	No
`email_sharing`	Enables users to share the video through email. If enabled, show_share_button must be set to true. Type: Boolean Valid Values: true | false Default: true Example: "true" Parent: ooyala_branding	No
`enable_error_screen`	Displays the Ooyala error screen if there is a playback error. Type: Boolean Valid Values: true | false Default: true Example: "true" Parent: ooyala_branding	No
`facebook_sharing`	Enables Facebook sharing. If enabled, show_share_button must be set to true.	No

Property	Description	Required?
	Type: Boolean Valid Values: true | false Default: false Example: "true" Parent: ooyala_branding	
`image_url`	Read only. The URL to the placeholder thumbnail that is shown while scrubbing through a video. To upload a new image, see [Scrubber](/v2/docs/players_scrubber). Type: String Default: None Example: "http://ak.c.ooyala.com/ZnMmM6AO3C-iBubra3R2Xd6pXHgK/shelf123467931" Parent: scrubber	No
`image_url`	Read only. The watermark shown during a video. We accept .PNG, .JPG, .GIF and .SWF watermarks, and we recommend using .PNG files with transparent backgrounds. To upload a new image, see [Watermark](/v2/docs/players_watermark). Type: String Default: None Example: "http://ak.c.ooyala.com/ZnMmM6AO3C-iBubra3R2Xd6pXHgK/watermark123467943" Parent: watermark	No
`is_default`	Sets this player as the default player template to use when creating new players. Type: Boolean Valid Values: true | false Default: false Example: "true"	No
`labels`	Specifies a label to use for shared videos. Type: Array of strings Default: None Conditions: Required if source is specified. Example: ["videos_of_kittens","video_of_dogs"] Parent: related_videos	Conditional

Property	Description	Required?
`name`	Name of the player. Type: String Example: "My Player with Watermark"	Yes
`ooyala_branding`	Specifies branding options. Type: Container	No
`order`	Specifies whether related videos are sorted in ascending or descending order. Type: String Valid Values: "asc" | "desc" Default: "desc" Example: "desc" Parent: related_videos	No
`playback`	Specifies playback options. Type: Container	No
`position`	Specifies the display position of the watermark. Type: String Default: bottom-right Example: bottom-left Valid Values: bottom | bottom-left | bottom-right | left | right | top | top-left | top-right Parent: watermark	No
`provider_homepage_url`	The URL to your website home page. Type: String Default: None Conditions: Required if show_info_screen_homepage_link is set to true. Example: "http://myhomepage.com"	Conditional
`related_videos`	Specifies related video settings. Type: Container	No
`scrubber`	Specifies scrubber options. Type: Container	No
`show_ad_countdown`	Displays a countdown timer during the advertisement that lets the user when the video will start. Type: Boolean Valid Values: true | false Default: true Example: "true" Parent: ooyala_branding	No

Property	Description	Required?
show_bitrate_button	Displays a button that allows users to choose from available bitrates. Type: Boolean Valid Values: true | false Default: false Example: "true" Parent: ooyala_branding	No
show_channel_button	Displays the Channel button which users can click to view other content on the channel. Type: Boolean Valid Values: true | false Default: true Example: "true" Parent: ooyala_branding	No
show_embed_button	Displays an embed button which contains a code snippet for the video that users can embed on their websites. Type: Boolean Valid Values: true | false Default: true Example: "true" Parent: ooyala_branding	No
show_end_screen_replay_button	When the video completes, enables the user to click a button to replay the video. Type: Boolean Valid Values: true | false Default: true Example: "true" Parent: ooyala_branding	No
show_info_button	Enables the Info button which the user can use to view information about the video. Type: Boolean Valid Values: true | false Default: false Example: "true" Parent: ooyala_branding	No

Property	Description	Required?
show_info_screen_description	Displays the description of the video when the user clicks the Info button. If enabled, show_info_button must be set to true. Type: Boolean Valid Values: true | false Default: false Example: "true" Parent: ooyala_branding	No
show_info_screen_homepage_link	Displays a link to the video provider. If enabled, show_info_screen_homepage_link and show_info_screen_homepage_name must be specified. Type: Boolean Valid Values: true | false Default: false Example: "true" Parent: ooyala_branding	Conditional
show_info_screen_title	Displays the title of the video when the user clicks the Info button. If enabled, show_info_button must be set to true. Type: Boolean Valid Values: true | false Default: false Example: "true" Parent: ooyala_branding	No
show_share_button	Enables the share button, which enables users to share the content with social networks. If you enable this button without enabling other sharing options (e.g., twitter_sharing), the button will appear with no options. Type: Boolean Valid Values: true | false Default: true Conditions: Required if one or more sharing options are specified. Example: "true" Parent: ooyala_branding	Conditional
show_volume_button	Displays the volume button. Type: Boolean	No

Property	Description	Required?
	Valid Values: true | false Default: true Example: "true" Parent: ooyala_branding	
`sort`	Specifies how related videos are sorted. Type: String Valid Values: "upload_time" Default: "upload_time" Example: "upload_time" Parent: related_videos	No
`source`	Specifies which types of videos are displayed as related videos. Type: String Valid Values: shared_labels | specified_label Default: None Conditions: Required if related_videos is specified. Example: "same_player" Parent: related_videos	Conditional
`stumbleupon_sharing`	Enables StumbleUpon sharing. If enabled, show_share_button must be set to true. Type: Boolean Valid Values: true | false Default: false Example: "true" Parent: ooyala_branding	No
`twitter_sharing`	Enables Twitter sharing. If enabled, show_share_button must be set to true. Type: Boolean Valid Values: true | false Default: false Example: "true" Parent: ooyala_branding	No
`url_sharing`	Displays a Copy URL button. If enabled, show_share_button must be set to true. Type: Boolean Valid Values: true | false	No

Property	Description	Required?
	Default: true Example: "true" Parent: ooyala_branding	
watermark	Specifies watermark options. Type: Container	No

EXAMPLES

Example: Create a Basic Player with Default Settings

This example creates a basic player that uses all default settings:

```
[POST] /v2/players{
   "name": "Basic Player"
}
```

Backlot returns a response similar to the following:

```
{
   "ooyala_branding":{
      "show_share_button":true,
      "twitter_sharing":false,
      "show_info_screen_homepage_link":true,
      "show_ad_countdown":true,
      "show_info_screen_title":true,
      "facebook_sharing":false,
      "show_info_button":true,
      "url_sharing":false,
      "email_sharing":true,
      "digg_sharing":false,
      "show_embed_button":true,
      "show_info_screen_description":false,
      "show_bitrate_button":false,
      "show_channel_button":true,
      "enable_error_screen":true,
      "show_end_screen_replay_button":true,
      "accent_color":"#ffbb00",
      "show_volume_button":true
   },
   "name":"Basic Player",
   "provider_homepage_url":null,
   "related_videos":{
      "click_behavior":"new_page",
      "order":"desc",
      "sort":"upload_time",
      "labels":[

      ],
      "source":"shared_labels"
   },
   "is_default":false,
   "scrubber":{
      "image_url":null,
      "always_show":false
   },
   "playback":{
      "buffer_on_pause":false
   },
```

```
   "id":"83fc33534581448eab446935baa6b1f0",
   "default_closed_caption_language":"",
   "watermark":{
      "image_url":null,
      "click_url":null,
      "alpha":1.0,
      "position":"bottom-right"
   }
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

Note: You can get this information at any time by generating a get request against the player ID (e.g., `[GET] /v2/players/83fc33534581448eab446935baa6b1f0`).

Example: Change Accent Color and Enable Twitter and Facebook

This example changes the accent color to pink and turns on Twitter and Facebook sharing:

```
[PATCH]/v2/players/83fc33534581448eab446935baa6b1f0{
   "ooyala_branding":{
      "twitter_sharing":true,
      "facebook_sharing":true,
      "accent_color":"#FF1493"
   }
}
```

Backlot returns a response similar to the following:

```
{
   "ooyala_branding":{
      "show_share_button":true,
      "twitter_sharing":true,
      "show_info_screen_homepage_link":true,
      "show_ad_countdown":true,
      "show_info_screen_title":true,
      "facebook_sharing":true,
      "show_info_button":true,
      "url_sharing":false,
      "email_sharing":true,
      "digg_sharing":false,
      "show_embed_button":true,
      "show_info_screen_description":false,
      "show_bitrate_button":false,
      "show_channel_button":true,
      "enable_error_screen":true,
      "show_end_screen_replay_button":true,
      "accent_color":"#FF1493",
      "show_volume_button":true
   },
   "name":"Basic Player",
   "provider_homepage_url":null,
   "related_videos":{
      "click_behavior":"new_page",
      "order":"desc",
      "sort":"upload_time",
      "labels":[

      ],
      "source":"shared_labels"
```

```
   },
   "is_default":false,
   "scrubber":{
      "image_url":null,
      "always_show":false
   },
   "playback":{
      "buffer_on_pause":false
   },
   "id":"83fc33534581448eab446935baa6b1f0",
   "default_closed_caption_language":"",
   "watermark":{
      "image_url":null,
      "click_url":null,
      "alpha":1.0,
      "position":"bottom-right"
   }
}
```

Example: List All Players

This example lists all players:

```
[GET]/v2/players
```

Backlot returns a list of players and their configuration settings.

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

Example: Delete a Player

This example deletes a player:

```
[DELETE]/v2/players/83fc33534581448eab446935baa6b1f0
```

Backlot returns a 200 response.

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

UPLOAD WATERMARK IMAGE FOR PLAYER V3

You can upload a partially transparent image that displays during content playback for Player V3.

Note: Player customizations with the Backlot API are only valid for Ooyala Flash Player V3. Customize the Ooyala HTML5 Player V3 with the Player JavaScript API. Customize Player V4 by modifying skin.json or the player CSS.

UPLOAD WATERMARK

```
[PUT]/v2/players/player_id/watermark
 <file_contents>
```

REMOVE WATERMARK FROM A PLAYER

```
[DELETE]/v2/players/player_id/watermark
```

ROUTE ATTRIBUTES

The following table describes all attributes that can be expressed through the route.

Route Attribute	Description
player_id	The ID of the player. Type: String Default: None Example: /players/ IzNnllMjphu2XF3_UgPROoCi9B2BwtSg

EXAMPLES

This example uploads a watermark image for a player:

```
[PUT]/v2/assets/IzNnllMjphu2XF3_UgPROoCi9B2rtWs/watermark
```

Backlot returns a 200 response.

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

UPLOAD PLAYER SCRUBBER IMAGE FOR PLAYER V3

You can upload an image to show as viewers scrub through a video for Player V3.

Note: Player customizations with the Backlot API are only valid for Ooyala Flash Player V3. Customize the Ooyala HTML5 Player V3 with the Player JavaScript API. Customize Player V4 by modifying skin.json or the player CSS.

UPLOAD SCRUBBER IMAGE

```
[PUT]/v2/players/player_id/scrubber_image
   <file_contents>
```

GET SCRUBBER IMAGE FOR A PLAYER

```
[GET]/v2/players/player_id/scrubber_image
```

REMOVE SCRUBBER IMAGE FROM A PLAYER

```
[DELETE]/v2/players/player_id/scrubber_image
```

ROUTE ATTRIBUTES

The following table describes all attributes that can be expressed through the route.

Route Attribute	Description
player_id	The ID of the player. Type: String Default: None Example: /players/IzNnllMjphu2XF3_UgPROoCi9B2BwtSg

EXAMPLES

This example uploads a scrubber image for a player:

```
[PUT]/v2/assets/IzNnllMjphu2XF3_UgPROoCi9B2rtWs/scrubber_image
```

Backlot returns a 200 response.

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

PLAYER CUSTOM METADATA

You can add your own name-value pairs to any player.

SET/REPLACE PLAYER CUSTOM METADATA

```
[PUT]/v2/players/player_id/metadata{
   "name1":"value1",
   "name2":"value2",
   "name3":"value3",
   .
  .
  .
 "name100":"value100"
}
```

MODIFY CUSTOM METADATA

```
[PATCH]/v2/players/player_id/metadata{
   "name1":"value1",
```

```
   "name2":"value2",
   "name3":"value3",
   .
 .
 .
 "name100":"value100"
}
```

GET CUSTOM METADATA

```
[GET]/v2/players/player_id/metadata
```

DELETE ALL CUSTOM METADATA FOR PLAYER

```
[DELETE]/v2/players/player_id/metadata
```

DELETE A SINGLE CUSTOM METADATUM FOR PLAYER

```
[PATCH]/v2/players/player_id/metadata
{
   "name":"null"
}
```

ROUTE ATTRIBUTES

The following table describes all attributes that can be expressed through the route.

Route Attribute	Description
player_id	The ID of the player. Type: String Default: None Example: /players/ IzNnllMjphu2XF3_UgPROoCi9B2BwtSg

EXAMPLES

This example creates custom metadata for the `IzNnllMjphu2XF3_UgPROoCi9B2rtWs` player:

```
[PUT]/v2/players/IzNnllMjphu2XF3_UgPROoCi9B2rtWs/metadata
{
   "skin": "branded"
}
```

Backlot returns a response similar to the following:

```
{

}
```


Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

PLAYER V3 THIRD-PARTY MODULES

With third-party modules, you can associate additional features with V3 players.

Third-party modules currently available are as follows:

- Playlists/Pods, module ID d42a1c8d4f91437c88db1c25cb8d6d04

To see descriptions of all available third-party modules, use the `[GET] /v2/third_party_modules` request.

LIST AVAILABLE THIRD-PARTY MODULES

```
[GET]/v2/third_party_modules
```

The system responds with an `items` hash describing all available third-party modules. For example:

```
{
   "items":[
      {
         "description":"Playlists",
         "is_flash":"false",
         "is_v3":"true",
         "id":"d42a1c8d4f91437c88db1c25cb8d6d04",
         "type":"v3-playlists"
      }
   ]
}
```

ASSOCIATE THIRD-PARTY MODULE WITH A PLAYER

When you associate a third-party module with a player, you can optionally include properties that set certain characteristics.

```
[PUT] /v2/players/player_id/third_party_modules/third_party_module_id{
   {
      "parameters":{
         properties
      }
   }
}
```

VIEW THIRD-PARTY MODULES ASSOCIATED WITH A PLAYER

To view all third-party modules associated with a player:

```
[GET]/v2/players/player_id/third_party_modules
```


To view details about a single third-party module associated with a player:

```
[GET]/v2/players/player_id/third_party_modules/third_party_module_id
```

UPDATE THIRD-PARTY MODULE INFORMATION

New properties are added, and existing properties are updated. Parameters not included in the `parameters` hash are not affected or deleted.

Note: If you need to delete a parameter (rather than simply update its value), you can first DELETE the original association and then recreate it with the original PUT request but excluding the undesired parameter.

```
[PATCH]/v2/players/player_id/third_party_modules/third_party_module_id{
   {
      "parameters":{
         properties
      }
   }
}
```

DELETE ASSOCIATION OF THIRD-PARTY MODULE WITH A PLAYER

```
[DELETE]/v2/players/player_id/third_party_modules/third_party_module_id
```

ROUTE ATTRIBUTES

The following table describes all attributes that can be expressed through the route.

Route Attribute	Description
player_id	The ID of the player. Type: String Default: None Example: b4f1a93b779145b4a9c8ef15f8502345
third_party_module_id	The ID of the third-party module. Type: String Default: None Example: d42a1c8d4f91437c88db1c25cb8d6d04

PROPERTIES

The following table describes all properties that can be associated with a player.

Property	Description	Third-Party Module
`activeMenuColor`	Background color of selected tab in pod Type: Hexadecimal color value Default: #ffbb00	Playlists/Pods

Property	Description	Third-Party Module
	Example: "#ffbb00"	
`caption`	Format of title, description, and duration displayed in a pod. Values can be combined in any order, with a dash as delimiter, like the following examples: • "title" - to display only title • "title-description" - to display title and description • "title-description-duration" - to display all info • "title-duration" - to display title and duration only Type: String Valid Values: title | description | duration Default: title Example: "title-description-duration"	Playlists/Pods
`captionPosition`	Position of caption relative to thumbnails Type: String Valid Values: inside | outside Default: inside Example: "outside"	Playlists/Pods
`parameters`	Parent element for all properties for a specific third-party module	
`podType`	Format of pod. Type: Boolean Valid Values: scrolling | paging Default: scrolling Example: "scrolling"	Playlists/Pods
`position`	Position of PODs relative to a player,. Note that in playerless PODs, in which we have no player, top and bottom are rendered the same way, as are "right" and "left". Type: String Valid Values: top | bottom | left | right | none Default: bottom Example: "top"	Playlists/Pods
`rowsNumber`	Set number of rows, only when `podType` is “paging”. Value is not limited. Type: Integer	Playlists/Pods

Property	Description	Third-Party Module
	Default: 1 Example: "10"	
`theme`	Color scheme of pod Type: String Valid Values: dark | light Default: dark Example: "light"	Playlists/Pods
`thumbnailsSize`	Width of thumbnails in pixels; heigth is calculated from this. Type: Integer Default: 150 Example: "175"	Playlists/Pods
`thumbnailsSpacing`	Spacing between thumbnails in pixels Type: Integer Default: 3px Example: "4px"	Playlists/Pods
`wrapperFontSize`	Baseline font size in pixels. All other text sizing is relative to this. Type: Integer Default: 14px Example: 12px	Playlists/Pods

EXAMPLES

This example displays information about all available third-party modules:

```
[GET]/v2/third_party_modules
```

Backlot returns a response similar to the following:

```
{
   "items":[
      {
         "description":"Playlists",
         "is_flash":"false",
         "is_v3":"true",
         "id":"d42a1c8d4f91437c88db1c25cb8d6d04",
         "type":"v3-playlists"
      }
   ]
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example associates the Playlists/Pods third-party module (identified by d42a1c8d4f91437c88db1c25cb8d6d04) with a player:

```
[PUT]/v2/players/e18ab1da1813483499554ea2d8e67fbd/third_party_modules/
d42a1c8d4f91437c88db1c25cb8d6d04{
   "parameters":{
      "menuFontSize":"10",
      "thumbnailsSpacing":"0",
      "thumbnailsSize":"150",
      "activeMenuColor":"#f50505",
      "caption":"",
      "wrapperFontSize":"",
      "podType":"scrolling",
      "tabsFontSize":"",
      "position":"bottom",
      "captionPosition":"inside",
      "theme":"dark"
   }
}
```

Backlot returns a response similar to the following:

```
{
   "description":"Playlists",
   "is_flash":"false",
   "parameters":{
      "menuFontSize":"10",
      "thumbnailsSize":"150",
      "position":"bottom",
      "podType":"scrolling",
      "activeMenuColor":"#f50505",
      "tabsFontSize":"",
      "caption":"",
      "wrapperFontSize":"",
      "captionPosition":"inside",
      "thumbnailsSpacing":"0",
      "theme":"dark"
   },
   "is_v3":"true",
   "id":"d42a1c8d4f91437c88db1c25cb8d6d04",
   "type":"v3-playlists"
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example lists all third-party modules associated with player e18ab1da1813483499554ea2d8e67fbd:

```
[GET]/v2/players/e18ab1da1813483499554ea2d8e67fbd/third_party_modules
```

Backlot returns a response similar to the following:

```
{
   "items":[
      {
         "parameters":{
            "activeMenuColor":"#f50505",
```

```
            "position":"bottom",
            "caption":"",
            "podType":"scrolling",
            "menuFontSize":"10",
            "thumbnailsSize":"150",
            "wrapperFontSize":"",
            "thumbnailsSpacing":"0",
            "tabsFontSize":"",
            "theme":"dark",
            "captionPosition":"inside"
         },
         "type":"v3-playlists",
         "id":"d42a1c8d4f91437c88db1c25cb8d6d04",
         "is_flash":"false",
         "description":"Playlists",
         "is_v3":"true"
      }
   ]
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

CROSS-DEVICE RESUME: PLAYBACK POSITION

This request is only one part of implementing cross-device resume (XDR).

A full discussion is in the *Player Developer Guide*.

There are two ways to get a playback position:

- For a specific asset for a given user
- For all assets of a given user

Ooyala stores the playhead positions only as far back as 90 days.

Note:

User account ids are distinct per provider. Two different providers can have a user with the same name, but these are treated as distinct users. The `account_id` you use with Ooyala APIs must be unique in your own systems. For privacy, Ooyala encourages that the value of `account_id` be some sort of GUID (global unique identifier), rather than a plain-text username or email address. For example, an acceptable `account_id` is to use a base64, URl-encoded, Secure-hash-algorithm-(SHA)-256 digest of the username or email address, salted with some secret string that only you know. This salt must not be reused for any of your vendors other than Ooyala. This ensures that neither Ooyala nor a "man-in-the-middle" hacker sniffing network traffic can translate back from your GUIDs to real usernames or passwords. The `account_id` must be less than 255 characters and must not contain reserved URL characters such as [/, &, |, or].
In most cases, you do *not* need to explicitly create an account with Ooyala APIs; you simply refer to an account wherever an API request requires it.

PLAYBACK POSITION FOR A SINGLE ASSET FOR A GIVEN ACCOUNT

To get the playback position for a specific asset for a given user:

```
[GET]/v2/cross_device_resume/accounts/account_id/viewed_assets/embed_code/
playhead_info
```

Response:

```
{
   "playhead_seconds":200.0,
   "timestamp":1369438194
}
```

PLAYBACK POSITION FOR ALL ASSETS FOR A GIVEN ACCOUNT

To get the playback positions for all assets viewed for a specific user. Items are grouped by asset ID (`embed_code`). The response includes the account ID and the provider ID.

```
[GET]/v2/cross_device_resume/accounts/account_id/playhead_info?
limit=max_number_results&start_date=yyyy-mm-dd
```

Example Response

```
{
   "items":[
      {
         "playhead_seconds":200.0,
         "timestamp":1369438194,
         "embed_code":"g1cTd3NDpdGQC6iaqlLnP7Fb"
      },
      {
         "playhead_seconds":100.0,
         "timestamp":1369438193,
         "embed_code":"gzcTd3ND3W-K2sSjxS1Joehi9"
      }
   ],
   "account":"097325097354-9y3-8y7",
   "provider_id":"x4Z3g6cupBeswLiTso3nxE-BqW"
}
```

ROUTE ATTRIBUTES

The following table describes all attributes that can be expressed through the route.

Route Attribute	Description
account_id	The identifier of the user account. Type: String Default: None

QUERY STRING PARAMETERS

The following table describes all parameters that can be used on the query string.

Parameter	Description	Required?
`limit`	The maximum number of results to return. Type: Integer Example: ?limit=5	No

Parameter	Description	Required?
`start_date`	Starting date for search, in the format YYYY-MM-DD Type: String Example: ?start_date=2013-09-11	No

PROPERTIES

The following table describes all properties that can be returned in the response3.

Property	Description
`account`	The account ID you passed on the request. Type: String
`playhead_seconds`	The position expressed in seconds of the last playhead update received by Ooyala. **Note:** This is not necessarily the same as the farthest position the viewer may have reached. Type: Float Example: "201.0"
`provider_id`	The identifier of the provider (owner of the asset). Type: String Example: x4Z3g6cupBeswLiTso3nxE-BqW
`timestamp`	Epoch time (seconds since January 1, 1970) of the most recent playhead update for the asset. Type: String Example: "1369438193"

ANALYTICS API REFERENCES

Please note that all our platform services are currently dependent on Ooyala players being the generator of analytics data.

V3 ANALYTICS (OOYALA IQ) API

Parameter Reference

Reporting Query String Parameters

Here are the complete reference details for all query string parameters for the v3 Analytics *Reporting API*.

All parameter names are case-sensitive and lowercase.

The entire query string must be URL-encoded.

General Syntax of Reporting GET

The base syntax of the route and query string is as follows. For ease of reading, the single-line request has been split across several lines.

```
[GET] /v3/analytics/reports/?
            report_type=type
            &dimensions=dimensions
            &metrics=metrics
            &filters=filter_type=='filter_value'
            &start_date=date
            &end_date=date
            &other_parms
            &api_key=your_api_key
```

- The **required** query string parameters (shown in bold) are *report_type*, `start_date`, and api_key.
- If no dimensions are specified, total values across all dimension are returned.
- You can specify up to 3 dimensions at a time.
- If no metrics are specified, all metrics are returned.

Note: At this time, the only valid value for *report_type* is `performance`.

Note: You can only use 1 dimension (`url`) for `url` queries.

General Syntax of Reporting Long Queries POST

For queries with query parameters that would exceed the HTTP GET specification limit of 230 characters , please use a POST request. Some browsers and http clients may support more than 230 characters, but we will not provide official support for queries that violate the HTTP GET specification. For POST requests, pass a JSON object in the request body instead of the query string parameters.

```
[POST] /v3/analytics/reports

{
    "report_type":"type",
    "dimensions":"dimensions",
```

```
    "metrics":"metrics",
    "filters":"(filter_type==\"filter_value\")",
    "start_date":"date",
    "end_date":"date",
    "other_parms":"other_param_value",
    "api_key":"your_api_key"
}
```

Parameter	Description	Required
`report_type`	Specifies the type of report. Valid values: `performance` Default: None Example: `report_type=performance`	Yes
`start_date`	The start date is specified as *YYYY-MM-DD*. The value of `start_date` is the provider's timezone. Default: None Limitations: You can query a range of up to 1 year (366 days) for up to 2 filters and up to 1 month (31 days) for 3 filters. Example: `start_date=2014-10-28`	Yes
`end_date`	The end date is specified as *YYYY-MM-DD*. **Note:** The end date is not inclusive. That is, data in the response is up to but not including the end date. Default: Tomorrow's date in the provider's timezone. Limitations: You can query a range of up to 1 year (366 days) for up to 2 filters and up to 1 month (31 days) for 3 filters. Example: To get data through the end of 2014-10-29: `end_date=2014-10-29`	No
`metrics`	List of comma-separated metric names. Default: * (all metrics). Valid values: See *Metrics* on page 86 section for details.	No

Parameter	Description	Required
	Example: `metrics=plays_requested,displays`	
`dimensions`	List of comma-separated dimension names. Results are grouped by the specified dimensions. If no dimensions are specified, total values across all dimension are returned. Valid values: See *Dimensions* on page 83 section for details. You can specify up to 3 dimensions at a time. **Note:** You can only use 1 dimension (`url`) for `url` queries. Default: None Example: `dimensions=country,region`	No

`filters`	Restricts the set of results by specified filter values.
	
	Note: For up to 2 filters you can set a date range of up to 1 year (366 days). For 3 filters you can set a date range of up to 1 month (31 days).
	
	Valid values: See *Filters* on page 95 for valid filter names and boolean operations.
	
	• The value of `filter_by` must be URL-encoded.
	• The value of the actual filter must be enclosed in single quotation marks.
	
	Default: None
	
	Examples:
	
	• Filter by the country Australia:
	`filters=country=='AU'`
	• Filter by mobile devices in country Colombia
	`filters=country=='CO',device_type=='mobile'`
		No

`time_segment`	Specify the time-based segment for dimension data. See the discussion of behavior in *About time_segment and Data Persistence* on page 83. This will sort blocks of data.	No

Parameter	Description	Required
	Note: A week is defined as Monday - Sunday. Valid values: • `month` | `week` | `day` Default: None Example: `time_segment=day`	
`sort`	List of comma-separated metrics to sort by. For multiple metrics, sorts by the metrics in the order the metrics are placed in the query. You can explicitly use as many sort metrics as you want (given that you have the metric), but default sorting has a limit of two metrics. Default: Sort by first two metrics (if present) in query order. Default value ordering is descending order. For ascending order, prefix a given metric with the + character. Examples: • Plays requested, displays, and video starts in descending order: `sort=plays_requested,displays,video_starts` • Video starts in ascending order: `sort=+video_starts` **Note:** We recommend that you do not use segment_watched or percentage_watched for sorting. Please use other metrics, like plays_requested, instead. There are different sort semantics for different metrics. For example, segment_watched and percentage_watched use the string (of the entire array) for sorting and do not use a numeric sort. This means that if you sort by segment_watched, your results may not appear in numeric order. Your results may appear with the first result having a lower count of segment_watched than the second result (e.g. [99,…] , [999,...]).	No

Parameter	Description	Required
	In contrast, plays_requested uses a numeric sort, which will provide you with clear results.	
`limit`	Limit the records returned in the response. Maximum limit: 1000. You can use the `page` parameter to paginate through all of your data. Default: 1000 Example: `limit=100`	No
`page`	Integer for pagination. Starts with 0. Default: 0 Example: For the second page: `page=1`	No

ABOUT TIME_SEGMENT AND DATA PERSISTENCE

The time range automatically expands to the minimal time range to fulfill the request. For example, specifying a start and end date for only today but with a `time_segment` of `week` causes the specified the start/end date to expand to cover the entire current week (in the provider's timezone). When there is no specified `time_segment`, the aggregate total for the specified dimensions is returned. When there is neither `time_segment` nor any specified dimension, the grand total for the time range is returned.

Dimensions

Here are the reference details for all predefined dimensions for the v3 Analytics *Reporting API*.

Dimensions are specified with the *`dimensions=`* query string parameter. Some considerations:

- You can filter by up to 3 dimensions at a time.

  Note: You can only use 1 dimension for `url` queries.
- Multiple values must be comma-separated with no spaces.
- The order of multiple is not important:
  - `dimensions=device_type,dma` is the same as `dimensions=dma,device_type`

General Syntax of Reporting GET

The base syntax of the route and query string is as follows. For ease of reading, the single-line request has been split across several lines.

```
[GET] /v3/analytics/reports/?
            report_type=type
            &dimensions=dimensions
            &metrics=metrics
            &filters=filter_type=='filter_value'
            &start_date=date
            &end_date=date
            &other_parms
            &api_key=your_api_key
```

- The **required** query string parameters (shown in bold) are *report_type*, start_date, and api_key.
- If no dimensions are specified, total values across all dimension are returned.
- You can specify up to 3 dimensions at a time.
- If no metrics are specified, all metrics are returned.

Note: At this time, the only valid value for *report_type* is performance.

Note: You can only use 1 dimension (url) for url queries.

General Syntax of Reporting Long Queries POST

For queries with query parameters that would exceed the HTTP GET specification limit of 230 characters , please use a POST request. Some browsers and http clients may support more than 230 characters, but we will not provide official support for queries that violate the HTTP GET specification. For POST requests, pass a JSON object in the request body instead of the query string parameters.

```
[POST] /v3/analytics/reports

{
    "report_type":"type",
    "dimensions":"dimensions",
    "metrics":"metrics",
    "filters":"(filter_type==\"filter_value\")",
    "start_date":"date",
    "end_date":"date",
    "other_parms":"other_param_value",
    "api_key":"your_api_key"
}
```

Name	Meaning	Additional Info
asset	Video or other assets, including both Ooyala asset IDs (embed codes or content IDs) and external identifiers.	To query unlimited asset data with the API, in your query you must: 1. Set a date range of less than or equal to 1 month (<32 days). 2. Use *dimensions=asset* (with no additional dimensions). 3. Set the *page* parameter (where *page*=X and X<= the last page). 4. Set the *limit* parameter (where *limit*<=1000). To filter the asset dimension by label, include the filter: filters=label=='yourLabelId' Response also includes: { *embed_code,name,status,type* } • *name* is the title of the video. • *embed_code* is the video identifier (content ID or asset ID).

Name	Meaning	Additional Info
		• `status` is the video's status in Ooyala Backlot: live or paused.
`country`	Country code. See *Country and Location Codes* on page 100.	Response includes full country name. In the Analytics UI, part of ***Geography***.
`region`	Geographic region. Our geographic reporting uses the definition of region provided by *Quova*. You cannot specify `region` alone; you must also specify its `country`.	In the Analytics UI, part of ***Geography***.
`dma`	Defined marketing area (a US-centric concept not prominent in other parts of the world), specified by *DMA ID*.	Response includes DMA ID and a descriptive name. In the Analytics UI, part of ***Geography***.
`state`	State name. For example, if you drill down by Geo on the United States, you will see California, Arizona, etc. Our geographic reporting uses the definition of state provided by *Quova* (state is "the first-level administrative division" of a country). You cannot specify `state` alone; you must also specify its `country`.	In the Analytics UI, part of ***Geography***.
`device_type`	Device type.	See possible values in *Codes for Platforms, Devices, and Operating Systems* on page 98.
`domain`	Internet domain (fully qualified domain name).	In the Analytics UI, part of ***Traffic Source***.
`url`	Uniform resource locator (URL). **Note:** You can only use 1 dimension (`url`) for `url` queries. You can only apply `url` or `domain` filters for `url` queries.	In the Analytics UI, part of ***Traffic Source***.
`os`	Operating system of the user's device. Must be exact string; partial matches not supported.	See possible values in *Codes for Platforms, Devices, and Operating Systems* on page 98.
`browser`	User's web browser. **Note:** The `browser` dimension applies only when `device_type` is `desktop`.	See possible values in *Codes for Platforms, Devices, and Operating Systems* on page 98.
`pcode`	Ooyala-supplied provider ID.	
`player_id`	The identifiers of Ooyala or other named video players used during playback.	**Note:** The Ooyala mobile SDKs (Android and iOS) currently do not send a player_id, so the player_id

Name	Meaning	Additional Info
		for the mobile SDKs is attributed as "unknown".

Metrics

Here are the complete reference details for all metrics for the v3 Analytics *Reporting API*.

Note: All metrics, except for player_loads, are associated with the asset pcode. The player_loads metric is associated with the player pcode.

Metrics are specified with the *`metrics=`* query string parameter. Multiple values must be comma-separated with no spaces.

Note: You can retrieve the entire set of metrics by not including `metrics` in your query.

Warning: If you don't specify metrics for the `url` dimension, an error message will be returned. Supported metrics for the `url` dimension include:

- `displays`
- `plays_requested`
- `video_starts`
- `time_watched`
- `autoplays`
- `player_loads`
- `replays`
- `uniq_plays_requested`
- `uniq_displays`
- `uniq_video_starts`
- `playthrough_25`
- `playthrough_50`
- `playthrough_75`
- `playthrough_100`

To retrieve a specific metric, specify it as in the following examples:

- `metrics=plays_requested`
- `metrics=uniq_plays_requested`
- `metrics=displays,uniq_displays`

General Syntax of Reporting GET

The base syntax of the route and query string is as follows. For ease of reading, the single-line request has been split across several lines.

```
[GET] /v3/analytics/reports/?
            report_type=type
            &dimensions=dimensions
            &metrics=metrics
            &filters=filter_type=='filter_value'
            &start_date=date
            &end_date=date
            &other_parms
            &api_key=your_api_key
```

- The **required** query string parameters (shown in bold) are *`report_type`*, `start_date`, and api_key.
- If no dimensions are specified, total values across all dimension are returned.

- You can specify up to 3 dimensions at a time.
- If no metrics are specified, all metrics are returned.

Note: At this time, the only valid value for *report_type* is `performance`.

Note: You can only use 1 dimension (`url`) for `url` queries.

General Syntax of Reporting Long Queries POST

For queries with query parameters that would exceed the HTTP GET specification limit of 230 characters , please use a POST request. Some browsers and http clients may support more than 230 characters, but we will not provide official support for queries that violate the HTTP GET specification. For POST requests, pass a JSON object in the request body instead of the query string parameters.

```
[POST] /v3/analytics/reports

{
    "report_type":"type",
    "dimensions":"dimensions",
    "metrics":"metrics",
    "filters":"(filter_type==\"filter_value\")",
    "start_date":"date",
    "end_date":"date",
    "other_parms":"other_param_value",
    "api_key":"your_api_key"
}
```

UNIQUE USERS

You may want to measure the number of unique users. To measure unique users, use the `uniq_plays_requested` metric described below. The Ooyala mobile SDKs generate and store a random unique ID which is application-specific. In other environments, a unique user is identified by local storage or cookie. If a user clears their browser cache, that user/device's ID will get regenerated next time they watch video. The generated IDs are completely random and don't include any user-identifiable information. When such information is not available for a user, a new unique identifier will be created for that user. We de-duplicate when calculating the number of unique users over time. For example, day 1 has users A, B, C; day 2 has users B, E., then when you pick date range = day 1 and day 2, then total unique users = 4 (A, B, C, E), Daily Avg. Unique Users = (3+2) / 2 = 3 (2.5 is converted to the closed integer).

VIDEO AND ENGAGEMENT-RELATED METRICS

Metric Name	Meaning	Related Event, UI Label
`autoplays`	Number of automatic plays (without user intervention).	
`plays_requested`	The number of times that the "Play" button is triggered either manually or automatically for any content (including video and ad content). This metric does not include the replay event. **Note:** We have improved our calculation of this metric. You may see the number of plays requested from your HTML5 and SDK players	***Plays Requested*** on Business Intelligence page

Metric Name	Meaning	Related Event, UI Label
	increase after 11/11/2014 due to this implementation.	
`displays`	The number of times that the particular video content is loaded and displayed within the player before it gets played (displays are related to each individual asset). Each time an embed code is changed this event gets triggered.	***Displays*** on Business Intelligence page
`replays`	Number of times the replay button is clicked.	
`video_starts`	Number of times the actual non-ad video content begins playback. Thus, not fired until the completion of pre-roll ads and actual asset playback has begun. If a user clicks on the “Play” button, a pre-roll ad starts playing, and the user drops off before the actual video content starts playing, this event won’t be triggered. **Note:** We have improved our calculation of this metric. You may see the number of video starts from your HTML5 and SDK players decrease after 11/11/2014 due to this implementation.	***Video Starts*** on Business Intelligence page
`time_watched`	Amount of time the user spent playing back the video asset, in HH:MM:SS format. We have added a smart format for values >1000h (For example, "1234:56:57" gets converted to "1.2k").	
`playthrough_25`	Number of video plays for the selected assets that reached the state of 25% completion of the video asset. No matter how many times the user rewinds within the same view/play session, once the "state" is reached it won't be marked/counted again. For example, within the same video viewing session (i.e. a user didn't refresh the page or reload the player), if a user rewinds and watches through the 25% quartile multiple times, Playthrough 25% will only count once. Applies only to video-on-demand (VOD), not live streams.	
`playthrough_50`	Number of video plays for the selected assets that reached the state of 50% completion of the video asset. No matter how many times the user	

Metric Name	Meaning	Related Event, UI Label
	rewinds within the same view/play session, once the "state" is reached it won't be marked/counted again. For example, within the same video viewing session (i.e. a user didn't refresh the page or reload the player), if a user rewinds and watches through the 50% quartile multiple times, Playthrough 50% will only count once. Applies only to video-on-demand (VOD), not live streams.	
`playthrough_75`	Number of video plays for the selected assets that reached the state of 75% completion of the video asset. No matter how many times the user rewinds within the same view/play session, once the "state" is reached it won't be marked/counted again. For example, within the same video viewing session (i.e. a user didn't refresh the page or reload the player), if a user rewinds and watches through the 75% quartile multiple times, Playthrough 75% will only count once Applies only to video-on-demand (VOD), not live streams.	
`playthrough_100`	Number of video plays for the selected assets that reached the state of 100% completion of the video asset. No matter how many times the user rewinds within the same view/play session, once the "state" is reached it won't be marked/counted again. For example, within the same video viewing session (i.e. a user didn't refresh the page or reload the player), if a user rewinds and watches through the 100% quartile multiple times, Playthrough 100% will only count once. Applies only to video-on-demand (VOD), not live streams.	
`avg_time_watched_per_video`	Average video viewing time, calculated as `sum(time watched)/video_starts`, converted to HH:MM:SS format. **Note:** We have improved our calculation of video starts. You may see avg_time_watched_per_video	

Metric Name	Meaning	Related Event, UI Label
	increase after 11/11/2014 due to this implementation. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	
`play_conversion_rate`	Ratio of plays requested events to displays events. **Note:** We have improved our calculation of plays requested. You may see the play_conversion_rate metric from your HTML5 and SDK players increase after 11/11/2014 due to this implementation. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	
`video_conversion_rate`	Ratio of video starts events to plays requested events. **Note:** We have improved our calculation of video starts and plays requested. You may see the number of video starts from your HTML5 and SDK players change after 11/11/2014 due to this implementation. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	
`player_loads`	Triggered each time the player loads on the page. The `player_loads` metric is associated with the pcode for the player.	

DISCOVERY INSIGHTS METRICS

Metric Name	Meaning	Used in Report Type
`end_screen_ctr`	The CTR of the Discovery end screen, including auto-plays. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance
`end_screen_ctr_excluding_autoplays`	The CTR of the Discovery end screen, excluding auto-plays.	Performance

Metric Name	Meaning	Used in Report Type
	Note: This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	
`end_screen_impressions`	The number of impressions of the Discovery end screen. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance
`pause_screen_ctr`	The CTR of the Discovery pause screen. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance
`pause_screen_impressions`	The number of impressions of the Discovery pause screen. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance
`aggregate_ctr`	The CTR covering impressions and clicks of both the Discovery end screen and pause screen. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance
`lift_percent_end_screen`	The lift from the Discovery end screen alone, including auto-plays. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance
`lift_percent_end_screen_excluding_autoplays`	The lift from the Discovery end screen alone, excluding auto-plays. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance
`lift_percent_pause_screen`	The lift from the Discovery pause screen. **Note:** This metric is not available for the `url` dimension. If you specify this metric	Performance

Metric Name	Meaning	Used in Report Type
	with a `url` query, an error message will be returned.	
`aggregate_lift_percent`	Total lift covering plays from the Discovery end screen and pause screen. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance
`added_view_time_hours`	The total added view time in hours attributed to Discovery plays. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance
`time_lift_percent_end_screen`	The time lift from the Discovery end screen alone, including auto-plays. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance
`time_lift_percent_end_screen_excluding_autoplays`	The time lift from the Discovery end screen alone, excluding auto-plays. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance
`time_lift_percent_pause_screen`	The time lift from the Discovery pause screen. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance
`aggregate_time_lift_percent`	Total time lift covering plays from the Discovery end screen and pause screen. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance
`discovery_plays`	**Discovery Plays** measure how many times a video was played after a recommendation by Discovery. **Note:** This metric is not available for the `url` dimension. If you specify this metric	Performance

Metric Name	Meaning	Used in Report Type
	with a `url` query, an error message will be returned.	
`discovery_leads`	**Discovery Leads** measure the number of times different Discovery-recommended videos got played as a result of watching the original video. For example, the video asset "The Big Game" has a Discovery end screen at the end of the video. Discovery Leads measures the number of times a video shown on the Discovery end screen for "The Big Game" get played. For additional details on what value you can derive from Discovery Plays and Leads data, see *Deriving Value From Discovery Insights*. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance

OTHER METRICS AND THEIR RELATED REPORTS

Metric Name	Meaning	Used in Report Type
`uniq_plays_requested`	Number of unique users that trigger the "play" button (manually or automatically) for any content (see ***plays_requested***). **Note:** uniq_plays_requested is a measure of Unique Users (uniq_users). **Note:** We have improved our calculation of this metric. You may see the number of plays requested from your HTML5 and SDK players increase after 11/11/2014 due to this implementation.	Performance
`uniq_displays`	Number of unique users that see the video loaded and displayed within the player before it gets played (displays are related to each individual asset). Each time an embed code is changed the display event gets triggered (see ***displays***).	Performance
`uniq_video_starts`	Number of unique users that see playback of actual non-ad video content (see ***video_starts***). **Note:** We have improved our calculation of this metric. You may see the number of video starts from your HTML5 and SDK players decrease after 11/11/2014 due to this implementation.	Performance
`embeds_copied`	Number of times the embed code has been copied.	Performance

Metric Name	Meaning	Used in Report Type
	Note: This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	
`digg`	Number of times the "Digg" button has been pressed. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance
`emails_sent`	Number of times the "share via email" button is pressed. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance
`facebook`	Number of times the Facebook "Like" button has been pressed. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance
`twitter`	Number times the Twitter "tweet" button is pressed. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance
`urls_copied`	Number of times the "Copy URL" button has been pressed. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance
`segment_watched`	The number of times each segment of a piece of video content is watched. 1 segment is defined as 2.5% of video length. Please note if a user rewinds and watches the same segment N times, Segments Watched for that segment will count as N times. Applies only to video-on-demand (VOD), not live streams. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance
`percentage_watched`	Measures the percentage of a video (in its entirety) that ever got watched in increments of 2.5%. This metric will not increase if a user rewinds and re-watches a segment in the same video session. **Note:** This metric is not available for the `url` dimension. If you specify this metric with a `url` query, an error message will be returned.	Performance

Filters

Here are the complete reference details for filters for the v3 Analytics *Reporting API*.

The filter names are essentially the same as dimension names. Filters are specified with the *filters=* query string parameter. You can further refine the results with boolean operators. Surround filter values with single quotations (e.g. filters=asset=='asset_id'). Filter values are case-sensitive.

For examples of how to use filters, see *What Filters Can I Use?* on page 111.

Date Range Limitation

Note: For up to 2 filters you can set a date range of up to 1 year (366 days). For 3 filters you can set a date range of up to 1 month (31 days).

For queries with query parameters that would exceed the HTTP GET specification limit of 230 characters , please use a *POST request*.

General Syntax of Reporting GET

The base syntax of the route and query string is as follows. For ease of reading, the single-line request has been split across several lines.

```
[GET] /v3/analytics/reports/?
            report_type=type
            &dimensions=dimensions
            &metrics=metrics
            &filters=filter_type=='filter_value'
            &start_date=date
            &end_date=date
            &other_parms
            &api_key=your_api_key
```

- The **required** query string parameters (shown in bold) are *report_type*, `start_date`, and api_key.
- If no dimensions are specified, total values across all dimension are returned.
- You can specify up to 3 dimensions at a time.
- If no metrics are specified, all metrics are returned.

Note: At this time, the only valid value for *report_type* is `performance`.

Note: You can only use 1 dimension (`url`) for `url` queries.

General Syntax of Reporting Long Queries POST

For queries with query parameters that would exceed the HTTP GET specification limit of 230 characters , please use a POST request. Some browsers and http clients may support more than 230 characters, but we will not provide official support for queries that violate the HTTP GET specification. For POST requests, pass a JSON object in the request body instead of the query string parameters.

```
[POST] /v3/analytics/reports

{
    "report_type":"type",
    "dimensions":"dimensions",
    "metrics":"metrics",
    "filters":"(filter_type==\"filter_value\")",
    "start_date":"date",
    "end_date":"date",
    "other_parms":"other_param_value",
    "api_key":"your_api_key"
```

```
}
```

Filter Type	Meaning	Valid Values
`asset`	Filter data by asset. **Note:** This filter is not valid with the `url` dimension. You can only use `domain` and `url` filters with the `url` dimension.	Any asset id associated with the provider or subproviders. The asset id can be `embed_code` or `external_ids` that you have entered in Backlot.
`country`	Filter data by country. **Note:** This filter is not valid with the `url` dimension. You can only use `domain` and `url` filters with the `url` dimension.	Country code. See *Country and Location Codes* on page 100.
`region`	Filter data by geographic region. Our geographic reporting uses the definition of region provided by *Quova*. **Note:** This filter is not valid with the `url` dimension. You can only use `domain` and `url` filters with the `url` dimension.	Any region string. You cannot specify `region` alone; you must also specify its `country`.
`dma`	Filter by DMA (defined marketing area, a US-centric concept not prominent in other parts of the world). **Note:** This filter is not valid with the `url` dimension. You can only use `domain` and `url` filters with the `url` dimension.	Any provider-defined DMA.
`state`	Filter data by state (i.e. the states within a country). In our geographic reporting, state is "the first-level administrative division" of a country, as defined by *Quova*. For example, you can filter so you only see data for plays in California. **Note:** This filter is not valid with the `url` dimension. You can only use `domain` and `url` filters with the `url` dimension.	Any state string. You cannot specify `state` alone; you must also specify its `country`.
`device_type`	Filter by device type. **Note:** This filter is not valid with the `url` dimension. You can only use `domain` and `url` filters with the `url` dimension.	See possible values in *Codes for Platforms, Devices, and Operating Systems* on page 98.
`domain`	Filter by internet domain.	Any domain string.
`url`	Filter data by a specific URL. **Note:** You can only use the `url`filter with the `url` dimension. You cannot use this filter with any other dimensions.	Any URL string. If you use a URL that is not tracked by Ooyala analytics, the API will return 0 data results.

Filter Type	Meaning	Valid Values
`os`	Filter data by operating system. **Note:** This filter is not valid with the `url` dimension. You can only use `domain` and `url` filters with the `url` dimension.	See possible values in *Codes for Platforms, Devices, and Operating Systems* on page 98.
`browser`	Filter data by browser. **Note:** The `browser` dimension applies only when `device_type` is `desktop`. **Note:** This filter is not valid with the `url` dimension. You can only use `domain` and `url` filters with the `url` dimension.	See possible values in *Codes for Platforms, Devices, and Operating Systems* on page 98.
`pcode`	Filter data by the provider's pcode (Ooyala-supplied provider ID). For details, see *Your API Credentials*. **Note:** This filter is not valid with the `url` dimension. You can only use `domain` and `url` filters with the `url` dimension.	Any pcode that is the current account (the provider executing the API call) or one of its sub-accounts.
`player_id`	Filter data by player id (of an Ooyala player). **Note:** This filter is not valid with the `url` dimension. You can only use `domain` and `url` filters with the `url` dimension.	
`label`	Filter data by label using the label id. Keep in mind the following points about working with labels in Ooyala IQ: • Correlations among labels and associated videos are not pre-aggregated or pre-summarized (rolled up) but are resolved at the time of an API request. • The Ooyala system aggregates based on a label's parent hierarchy. For example, imagine two labels, the parent "Cycle World" and its child label "Sport Bikes". A video labeled with the child "Sport Bikes" is rolled up into the parent label "Cycle World". If you request data for a label, the metrics returned are for that label and its child labels. **Note:** This filter is not valid with the `url` dimension. You can only use `domain` and `url` filters with the `url` dimension.	You can use any defined label in Backlot. You can filter by multiple labels at a time.

BOOLEAN OPERATORS; (AND) AND | (OR)

The formal syntax of a filter is shown below.

```
filters= filter [boolOp filter]*
```


Symbol	Expansion
filter	*dimension-filter* or *composite-filter*
dimension-filter	*dimension op value* Where: • *dimension* is a dimension name listed in *Dimensions* on page 83 or a custom dimension name. • *op* is == (equal to) or != (not equal to). • *value* is a single-quoted dimension value, like os!='android' (data with operating system Android).
composite-filter	(*filter* [*boolOp filter*]*) Where: • *filter* is defined above. • *boolOp* is ; (AND) or | (OR). The boolean operator and second filter are optional and can be repeated without limit.

Grouping with Parentheses

Use parentheses to logically group components of the filter value. For an example, see *What Filters Can I Use?* on page 111.

Codes for Platforms, Devices, and Operating Systems

A device type is a category of the physical device on which a video was accessed. Platforms are a concatenation of a device type, an OS, and a browser. These tables shows the input you will use to query device or OS type.

Note: The Ooyala Player does not track AirPlay mode in Ooyala IQ.

DEVICE TYPES

deviceType
desktop
mobile
settop
tablet
unknown.devicetype

OPERATING SYSTEMS

Note: You can monitor Chromecast analytics by looking at data for browser=chrome, deviceOs=chromecast, and deviceType=settop.

osType
android
bada os

osType
chromecast
firefox os
ios
linux
mac
meego
rim os
rim tablet os
sunos
unknown.os
webos
windows
windows rt
xrossmediabar

BROWSER TYPES

browserType
android_sdk
android webkit
aol explorer
blackberry
chrome
chromium
eventmachine
fennec
firefox
ie
ios_sdk
msie
netfront
opera
safari
seamonkey
ucweb
unknown.browser
webkit/webos

Country and Location Codes

These country codes are used in a variety of Ooyala features.

The following table lists supported location codes:

Alpha-2 Code	English Short Name	Country Name Returned by Analytics API (If Different Than English Short Name)
AD	Andorra	
AE	United Arab Emirates	
AF	Afghanistan	
AG	Antigua and Barbuda	
AI	Anguilla	
AL	Albania	
AM	Armenia	
AN	Netherlands Antilles	
AO	Angola	
AQ	Antarctica	
AR	Argentina	
AS	American Samoa	
AT	Austria	
AU	Australia	
AW	Aruba	
AZ	Azerbaijan	
BA	Bosnia and Herzegovina	
BB	Barbados	
BD	Bangladesh	
BE	Belgium	
BF	Burkina Faso	
BG	Bulgaria	
BH	Bahrain	
BI	Burundi	
BJ	Benin	
BM	Bermuda	
BN	Brunei Darussalam	
BO	Bolivia	
BR	Brazil	Brasil
BS	Bahamas	
BT	Bhutan	
BV	Bouvet Island	

Alpha-2 Code	English Short Name	Country Name Returned by Analytics API (If Different Than English Short Name)
BW	Botswana	
BY	Belarus	
BZ	Belize	
CA	Canada	
CC	Cocos (Keeling) Islands	
CD	Congo, The Democratic Republic of the	Democratic Republic of the Congo
CF	Central African Republic	
CG	Congo	
CH	Switzerland	
CI	Cote D'Ivoire	Cote d'Ivoire
CK	Cook Islands	
CL	Chile	
CM	Cameroon	
CN	China	People's Republic of China
CO	Colombia	
CR	Costa Rica	
CU	Cuba	
CV	Cape Verde	
CX	Christmas Island	
CY	Cyprus	
CZ	Czech Republic	
DE	Germany	
DJ	Djibouti	
DK	Denmark	
DM	Dominica	
DO	Dominican Republic	
DZ	Algeria	
EC	Ecuador	
EE	Estonia	
EG	Egypt	
EH	Western Sahara	
ER	Eritrea	
ES	Spain	
ET	Ethiopia	

Alpha-2 Code	English Short Name	Country Name Returned by Analytics API (If Different Than English Short Name)
FI	Finland	
FJ	Fiji	
FK	Falkland Islands (Malvinas)	
FM	Micronesia, Federated States of	Micronesia
FO	Faroe Islands	
FR	France	
FX	France, Metropolitan	
GA	Gabon	
GB	United Kingdom	
GD	Grenada	
GE	Georgia	
GF	French Guiana	
GG	Gurnesy	
GH	Ghana	
GI	Gibraltar	
GL	Greenland	
GM	Gambia	
GN	Guinea	
GP	Guadeloupe	
GQ	Equatorial Guinea	
GR	Greece	
GS	South Georgia and the South Sandwich Islands	
GT	Guatemala	
GU	Guam	
GW	Guinea-Bissau	
GY	Guyana	
HK	Hong Kong	
HM	Heard Island and McDonald Islands	
HN	Honduras	
HR	Croatia	
HT	Haiti	
HU	Hungary	
ID	Indonesia	
IE	Ireland	Republic of Ireland

Alpha-2 Code	English Short Name	Country Name Returned by Analytics API (If Different Than English Short Name)
IL	Israel	
IM	Isle of Man	
IN	India	
IO	British Indian Ocean Territory	
IQ	Iraq	
IR	Iran, Islamic Republic of	Iran
IS	Iceland	
IT	Italy	
JE	Jersey	
JM	Jamaica	
JO	Jordan	
JP	Japan	
KE	Kenya	
KG	Kyrgyzstan	
KH	Cambodia	
KI	Kiribati	
KM	Comoros	
KN	Saint Kitts and Nevis	
KP	Korea, Democratic People's Republic of	North Korea, Korea (North)
KR	Korea, Republic of	South Korea, Korea (South)
KW	Kuwait	
KY	Cayman Islands	
KZ	Kazakhstan	
LA	Lao People's Democratic Republic	Laos
LB	Lebanon	
LC	Saint Lucia	
LI	Liechtenstein	
LK	Sri Lanka	
LR	Liberia	
LS	Lesotho	
LT	Lithuania	
LU	Luxembourg	
LV	Latvia	
LY	Libyan Arab Jamahiriya	Libya

Alpha-2 Code	English Short Name	Country Name Returned by Analytics API (If Different Than English Short Name)
MA	Morocco	
MC	Monaco	
MD	Moldova, Republic of	Moldova
MG	Madagascar	
MH	Marshall Islands	
MK	Macedonia	
ML	Mali	
MM	Myanmar	
MN	Mongolia	
MO	Macau	
MP	Northern Mariana Islands	
MQ	Martinique	
MR	Mauritania	
MS	Montserrat	
MT	Malta	
MU	Mauritius	
MV	Maldives	
MW	Malawi	
MX	Mexico	
MY	Malaysia	
MZ	Mozambique	
NA	Namibia	
NC	New Caledonia	
NE	Niger	
NF	Norfolk Island	
NG	Nigeria	
NI	Nicaragua	
NL	Netherlands	
NO	Norway	
NP	Nepal	
NR	Nauru	
NU	Niue	
NZ	New Zealand	
OM	Oman	
PA	Panama	

Alpha-2 Code	English Short Name	Country Name Returned by Analytics API (If Different Than English Short Name)
PE	Peru	
PF	French Polynesia	
PG	Papua New Guinea	
PH	Philippines	
PK	Pakistan	
PL	Poland	
PM	Saint Pierre and Miquelon	
PN	Pitcairn Islands	
PR	Puerto Rico	
PS	Palestinian Territory, Occupied	Occupied Palestinian Territory
PT	Portugal	
PW	Palau	
PY	Paraguay	
QA	Qatar	
RE	Reunion	
RO	Romania	
RU	Russian Federation	Russia
RW	Rwanda	
SA	Saudi Arabia	
SB	Solomon Islands	
SC	Seychelles	
SD	Sudan	
SE	Sweden	
SG	Singapore	
SH	Saint Helena	
SI	Slovenia	
SJ	Svalbard and Jan Mayen	
SK	Slovakia	Slovak Republic
SL	Sierra Leone	
SM	San Marino	
SN	Senegal	
SO	Somalia	
SR	Suriname	
ST	Sao Tome and Principe	
SV	El Salvador	

Alpha-2 Code	English Short Name	Country Name Returned by Analytics API (If Different Than English Short Name)
SY	Syrian Arab Republic	Syria
SZ	Swaziland	
TC	Turks and Caicos Islands	
TD	Chad	
TF	French Southern Territories	
TG	Togo	
TH	Thailand	
TJ	Tajikistan	
TK	Tokelau	
TM	Turkmenistan	
TN	Tunisia	
TO	Tonga	
TP	East Timor	
TR	Turkey	
TT	Trinidad and Tobago	
TV	Tuvalu	
TW	Taiwan	Taiwan, Republic of China Taiwan
TZ	Tanzania, United Republic of	Tanzania
UA	Ukraine	
UG	Uganda	
UM	United States Minor Outlying Islands	
US	United States	
UY	Uruguay	
UZ	Uzbekistan	
VA	Holy See (Vatican City State)	
VC	Saint Vincent and the Grenadines	
VE	Venezuela	
VG	Virgin Islands, British	Virgin Islands (British)
VI	Virgin Islands, U.S.	Virgin Islands (U.S.)
VN	Vietnam **Note:** The country name is Viet Nam in Backlot.	
VU	Vanuatu	
WF	Wallis and Futuna	Wallis and Futuna Islands

Alpha-2 Code	English Short Name	Country Name Returned by Analytics API (If Different Than English Short Name)
WS	Samoa	
YE	Yemen	
YT	Mayotte	
YU	Yugoslavia	
ZA	South Africa	
ZM	Zambia	
ZR	Zaire	
ZW	Zimbabwe	
A1	Anonymous Proxy	
A2	Satellite Provider	
O1	Other	

HTTP Response Codes and Messages

The top of all responses contain the following message format. Possible values for *status message* and *code* are described in the table below. Unless otherwise indicated, these codes pertain to the Ooyala IQ Reporting API. In the case of an error, the response includes the errors key, with an explanatory message.

```
{
  “status”: “status message”,
  “statusCode”: code
  "errors" : "explanatory message"
}
```

Example:

```
 {
    “status”: “Invalid parameter”,
    “statusCode”: “400”,
    “errors”: [“filter has invalid syntax: invalid character . at position
 4”]
}
```

HTTP Response code	General Meaning	Detailed Description
200	Success	The request was successful.
204 No Content	Successfully received	Returned by the Event Ping API. No response body is returned.
400	Bad request	The request is incorrect. For example, it does not contain required query string parameters.
401	Not authorized	Either the credentials (Ooyala-provided API key and secret) are missing or their values are not valid.

HTTP Response code	General Meaning	Detailed Description
403	Insufficient permission	The credentials in the request do not have sufficient permission for data requested.
429	Quota exceeded	The requester's quota of API calls has been exceeded. Quotas are described at *API Rate Limiting*.
500	System error	Possible problem in the Ooyala system.
503	Service Unavailable	There is a temporary service outage.

v3 Analytics Reporting API Request Examples

The business use case examples here illustrate a variety of API requests to return data suitable for analysis to gain business insights.

Be sure to *sign your requests* and URL-encode your parameter values. The examples here do not show signed or URL-encoded requests.

For additional examples, see *API Requests: v2 Analytics and v3 Analytics Comparison* on page 112.

How Are My Different Brands Performing?

Context: Your company has many different brands, so you use a different player for each brand. You want to see how each brand is performing.

HOW TO IN V3 ANALYTICS

Query against the `asset` and player_id dimensions, filtering to select only player 1 and player 2. The results include the `plays_requested` metric.

```
[GET] /v3/analytics/reports/?
report_type=performance
&dimensions=asset,player_id
&filters=player_id=='player1',player_id=='player2'
&metrics=plays_requested
&start_date=2014-10-10
&end_date=2014-10-20
&api_key=yourApiKey
```

```
https://api.ooyala.com/v3/analytics/reports/?
report_type=performance&dimensions=asset,player_id&filters=player_id==’player1’
player_id==’player2’&metrics=plays_requested&start_date=2014-10-10&end_date=2014-10-20
&api_key=yourApiKey&expires=yourExpiration& signature=yourSignature
```

Response:

```
{
   "status": "OK",
   "results":[
      {
         "data":[
            {
              "metrics":{
                  "plays_requested":2500
```

```
                },
                "group":{
                   "asset":"asset_id_1",
                   "player_id":"player1",
                   "name":"unknown",
                   "status":"unknown",
                   "type":"unknown"
                }
             },
             {
                "metrics":{
                   "plays_requested":2200
                },
                "group":{
                   "asset":"asset_id_2",
                   "player_id":"player2",
                   "name":"unknown",
                   "status":"unknown",
                   "type":"unknown"
                }
             }
          ],
          "start_date":"2014-10-10T00:00Z",
          "end_date":"2014-10-20T00:00Z"
       }
    ],
    "status_code": "200",
    "total_count": "100",
    "result_count": "2"
}
```


What Geographies Are Driving My Content Traffic?

Context: Your content reaches viewers in multiple countries. You want to see what portion of your traffic is driven by each geography your content reaches.

HOW TO IN V3 ANALYTICS

Query against the *country* dimension and filter by domain='exampledomain.com' and device_type='tablet'. Sort by the *plays_requested* metric, as well, limiting the results to the month of February, 2015.

```
[GET] /v3/analytics/reports/?
report_type=performance
&dimensions=country
&filters=((device_type=='tablet')) AND ((domain=='exampledomain.com'))
&metrics=plays_requested
&start_date=2014-10-10
&end_date=2014-10-20
&sort=plays_requested
&api_key=yourApiKey
```

```
https://api.ooyala.com/v3/analytics/reports/?
report_type=performance&dimensions=country&amp;filters=%28%28device_type%3D
%3D%27tablet%27%29%29+AND+%28%28domain%3D%3D%27exampledomain.com%27%29%29
&metrics=plays_requested&start_date=2014-10-10&end_date=2014-10-20&sort=plays_requested&a
expires=yourExpiration&signature=yourSignature
```

Response:


```
{
   "status":"OK",
   "results":[
      {
         "data":[
            {
               "metrics":{
                  "plays_requested":11000
               },
               "group":{
                  "country":"us",
                  "countryName":"United States"
               }
            },
            {
               "metrics":{
                  "plays_requested":5100
               },
               "group":{
                  "country":"fr",
                  "countryName":"France"
               }
            },
            {
               "metrics":{
                  "plays_requested":2000
               },
               "group":{
                  "country":"gb",
                  "countryName":"United Kingdom"
               }
            },
            {
               "metrics":{
                  "plays_requested":2000
               },
               "group":{
                  "country":"be",
                  "countryName":"Belgium"
               }
            },
            {
               "metrics":{
                  "plays_requested":1600
               },
               "group":{
                  "country":"ca",
                  "countryName":"Canada"
               }
            },
            {
               "metrics":{
                  "plays_requested":1400
               },
               "group":{
                  "country":"de",
                  "countryName":"Germany"
               }
            },
            {
```

```
                "metrics":{
                   "plays_requested":1100
                },
                "group":{
                   "country":"ua",
                   "countryName":"Ukraine"
                }
             },
             {
                "metrics":{
                   "plays_requested":1000
                },
                "group":{
                   "country":"se",
                   "countryName":"Sweden"
                }
             },
             {
                "metrics":{
                   "plays_requested":800
                },
                "group":{
                   "country":"nl",
                   "countryName":"Netherlands"
                }
             },
             {

                "metrics":{
                   "plays_requested":600
                },
                "group":{
                   "country":"it",
                   "countryName":"Italy"
                }
             }
          ],
          "start_date":"2014-10-10T00:00Z",
          "end_date":"2014-10-20T00:00Z"
       }
    ],

    "status_code":"200",
    "total_count":100,
    "result_count":10
}
```


What Filters Can I Use?

Some example filters are shown below. These are not fully formed requests (see *v3 Analytics Reporting API Syntax* for complete syntax), only examples of the `filters` query parameter.

For queries with query parameters that would exceed the HTTP GET specification limit of 230 characters , please use a *POST request*.

Note: For up to 2 filters you can set a date range of up to 1 year (366 days). For 3 filters you can set a date range of up to 1 month (31 days).

GROUPING FILTERS

- Return data with the country Australia (AU; see *Country and Location Codes* on page 100) and the `mobile` device type:

```
...&filters=((country=='AU')) AND ((device_type=='mobile'))
```

- Mobile and tablet devices in Colombia:

```
...&filters=country==’CO’,device_type==’mobile’,device_type=='tablet'
```

- Only mobile in Columbia and the US:

```
...&filters=((device_type==’mobile’) AND (country==’CO’,country=='US'))
```

FILTERING BY LABEL

Ooyala labels are a mechanism for grouping videos. In Ooyala Analytics, labels can be used to filter. Use the label id to identify the label filter.

Keep in mind the following points about working with labels in Ooyala IQ:

- Correlations among labels and associated videos are not pre-aggregated or pre-summarized (rolled up) but are resolved at the time of an API request.
- The Ooyala system aggregates based on a label's parent hierarchy. For example, imagine two labels, the parent "Cycle World" and its child label "Sport Bikes". A video labeled with the child "Sport Bikes" is rolled up into the parent label "Cycle World". If you request data for a label, the metrics returned are for that label and its child labels.

This request returns totals for *all* metrics, grouped by a label with the label id 123abc456yui123xyz.

```
...&filters=label=='123abc456yui123xyz'
```

API Requests: v2 Analytics and v3 Analytics Comparison

These examples show some commonly used queries in v2 Analytics and their equivalents in v3 Analytics (Ooyala IQ).

For ease of reading:

- The v3 query parameter values are not URL-encoded.
- Variable values are prefixed with a $ sign.
- Some requests are broken across lines.
- "&metrics=metrics" is left out of the syntax because by default all metrics are returned. Define specific metrics to return in your reports with "&metrics=yourDesiredMetrics".

Examples here include the following:

- *Total Dimensions*
- *Total with Breakdown by Day*

- *Geo Dimensions* on page 116
- *Device and Platform Dimensions* on page 119

TOTAL DIMENSIONS

v2 URL Syntax	v3 Equivalent
`/v2/analytics/reports/account/performance/total/$date_range`	`/v3/analytics/reports/?report_type=performance&start_date=$start&end_date=$end&api_key=yourApiKey`

```
https://api.ooyala.com/v3/analytics/reports/?
report_type=performance&start_date=YYYY-MM-DD&end_date=YYYY-MM-
DD&api_key=yourApiKey&expires=yourExpiration&signature=yourSignature
```

Sample Ooyala IQ Response:

```
{
     “status_code”:“200”,
     "result_count":1,
     "results":[
      {
          “start_date”:“YYYY-MM-DD”,
          “end_date”:“YYYY-MM-DD”,
          “data”:[
            {
                 “group”:{
                     },
                 "metrics":{
                     "playthrough_50": "40000",
                     "uniq_displays": "3000000",
                     "video_starts": "4000000",
                     "displays": "5000000",
                     “percentage_watched”:[
                        18,
                        14,
                        13,
                        13,
                        15,
                        20,
                        12,
                        14,
                        12,
                        11,
                        19,
                        18,
                        8,
                        13,
                        11,
                        18,
                        16,
                        10,
                        15,
                        17,
                        15,
                        16,
                        8,
                        10,
```

```
                        12,
                        13,
                        8,
                        9,
                        8,
                        14,
                        8,
                        10,
                        11,
                        11,
                        12,
                        10,
                        11,
                        5,
                        0,
                        14
                    ],
                    "player_loads":5000000,
                    "segment_watched":[
                        18,
                        14,
                        13,
                        13,
                        15,
                        20,
                        12,
                        14,
                        12,
                        11,
                        19,
                        18,
                        8,
                        13,
                        11,
                        18,
                        16,
                        10,
                        15,
                        17,
                        15,
                        16,
                        8,
                        10,
                        12,
                        13,
                        8,
                        9,
                        8,
                        14,
                        8,
                        10,
                        11,
                        11,
                        12,
                        10,
                        11,
                        5,
                        0,
                        14
                    ],
                    "urls_copied": "3",
                    "plays_requested": "6500",
                    "embeds_copied": "2",
                    "facebook": "3",
```


```
                    "replays": "145",
                    "uniq_video_starts": "2000000",
                    "autoplays": "0",
                    "emails_sent": "0",
                    "digg": "0",
                    "playthrough_25": "50000",
                    "playthrough_75": "40000",
                    "twitter": "0",
                    "uniq_plays_requested": "60000",
                    "playthrough_100": "30000",
                    "time_watched": "8.3k:33:00"

                }
            }
        ]
    }
  ],
    “total_count”:1,
    “status”:“OK”
}
```

Note: There is no data in the group because there’s no dimension specified, and the plays requested, displays is grand total for all the metrics across the time frame

TOTAL WITH BREAKDOWN BY DAY

v2 URL Syntax	v3 Equivalent
`/v2/analytics/reports/account/performance/total/$date_range?breakdown_by=day`	`/v3/analytics/reports/?report_type=performance&time_segment=day&start_date=$start&end_date=$end&api_key=yourApiKey`

```
https://api.ooyala.com/v3/analytics/reports/?report_type=performance&time_segment=day&start_date=YYYY-MM-DD&end_date=YYYY-MM-DD&api_key=yourApiKey&expires=yourExpiration&signature=yourSignature
```

Sample Ooyala IQ Response:

```
{
   “status_code”:“200”,
   "result_count":3,
   "results":[
      {
         “start_date”:“YYYY-MM-DD”,
         “end_date”:“YYYY-MM-DD”,
         “data”:[
            {
               “group”:{

               },
               "metrics":{
                 "playthrough_50":100,
                 "uniq_displays":2500,
                 "video_starts":3000,
                 ...
                 "time_watched":24:00:00
         }
```

```
            }
         ]
      },
   {
      “start_date”:“YYYY-MM-DD”,
      “end_date”:“YYYY-MM-DD”,
      “data”:[
         {
            “group”:{

            },
            "metrics":{
              "playthrough_50":100,
              "uniq_displays":2500,
              "video_starts":3000,
              ...
              "time_watched":24:00:00
         }
         }
      ]
   },
{
   “start_date”:“YYYY-MM-DD”,
   “end_date”:“YYYY-MM-DD”,
   “data”:[
      {
         “group”:{

         },
         "metrics":{
            "playthrough_50":100,
            "uniq_displays":2500,
            "video_starts":3000,
            ...
            "time_watched":24:00:00
         }
      }
   ]
}
],
“status”:“OK”,
“total_count”:3
}
```

This returns metric totals broken down by day.

GEO DIMENSIONS

v2 URL Syntax	v3 Equivalent
`/v2/analytics/reports/asset/$asset_id/performance/countries/$country/regions/$region/$date_range`	`/v3/analytics/reports/?report_type=performance&dimensions=asset,country,region&filters=asset=='asset_id',country=='country',re&start_date=$start&end_date=$end&api_key=yourApiKey`

```
https://api.ooyala.com/v3/analytics/reports/?
report_type=performance&dimensions=asset,country,region&filters=asset==’asset_id’,
```

country==’country’,region==’region’&start_date=YYYY-MM-DD&end_date=YYYY-MM-DD
&api_key=yourApiKey&expires=yourExpiration&signature=yourSignature

Sample Ooyala IQ Response:


```
{
     “status_code”:“200”,
     "result_count":1,
     "results":[
      {
          “start_date”:“YYYY-MM-DD”,
          “end_date”:“YYYY-MM-DD”,
          “data”:[
            {
                 “group”:{
                     "asset”:”asset_id”,
                     "country":"Country Code",
                     "countryName":"Country Name",
                     "region":"Region",
                     "name”:”Asset Name”,
                     "status":"live”,
                     "type":”video"
                     },
                 "metrics":{
                     "playthrough_50": "40000",
                     "uniq_displays": "3000000",
                     "video_starts": "4000000",
                     "displays": "5000000",
                     “percentage_watched":[
                        18,
                        14,
                        13,
                        13,
                        15,
                        20,
                        12,
                        14,
                        12,
                        11,
                        19,
                        18,
                        8,
                        13,
                        11,
                        18,
                        16,
                        10,
                        15,
                        17,
                        15,
                        16,
                        8,
                        10,
                        12,
                        13,
                        8,
                        9,
                        8,
                        14,
                        8,
                        10,
                        11,
                        11,
```

```
            12,
            10,
            11,
            5,
            0,
            14
        ],
        "player_loads":5000000,
        "segment_watched":[
            18,
            14,
            13,
            13,
            15,
            20,
            12,
            14,
            12,
            11,
            19,
            18,
            8,
            13,
            11,
            18,
            16,
            10,
            15,
            17,
            15,
            16,
            8,
            10,
            12,
            13,
            8,
            9,
            8,
            14,
            8,
            10,
            11,
            11,
            12,
            10,
            11,
            5,
            0,
            14
        ],
        "urls_copied": "3",
        "plays_requested": "6500",
        "embeds_copied": "2",
        "facebook": "3",
        "replays": "145",
        "uniq_video_starts": "2000000",
        "autoplays": "0",
        "emails_sent": "0",
        "digg": "0",
        "playthrough_25": "50000",
        "playthrough_75": "40000",
        "twitter": "0",
        "uniq_plays_requested": "60000",
        "playthrough_100": "30000",
```

```
                    "time_watched": "75:33:00"

                }
            }
        ]
    }
  ],
    “total_count”:1,
    “status”:“OK”
}
```

DEVICE AND PLATFORM DIMENSIONS

v2 URL Syntax	v3 Equivalent
`/v2/analytics/reports/asset/$asset_id/performance/device_types/$device_type/platforms/$platform/$date_range`	`/v3/analytics/reports/?report_type=performance&dimensions=asset,device_type&filters=((device_type=='$device_type')) AND ((asset_id=='$asset_id'))&start_date=$start&end_date=$end&api_key=yourApiKey`

```
https://api.ooyala.com/v3/analytics/reports/?
report_type=performance&dimensions=asset,device_type&filters=
%28%28device_type%3D%3D%27device_type%27%29%29+AND+%28%28asset_id%3D
%3D%27$asset_id%27%29%29 &start_date=YYYY-MM-DD&end_date=YYYY-MM-DD
&api_key=yourApiKey&expires=yourExpiration&signature=yourSignature
```

Sample Ooyala IQ Response:

```
{
     “status_code”:“200”,
     "result_count":1,
     "results":[
      {
          “start_date”:“YYYY-MM-DD”,
          “end_date”:“YYYY-MM-DD”,
          “data”:[
            {
                 “group”:{
                     "asset”:”asset_id”,
                     "name”:”Asset Name”,
                     "status":"live”,
                     "type":”video",
                     "device_type":"device_type”
                     },
                 "metrics":{
                     "playthrough_50": "40000",
                     "uniq_displays": "3000000",
                     "video_starts": "4000000",
                     "displays": "5000000",
                     “percentage_watched":[
                        18,
                        14,
                        13,
                        13,
```

```
                        15,
                        20,
                        12,
                        14,
                        12,
                        11,
                        19,
                        18,
                        8,
                        13,
                        11,
                        18,
                        16,
                        10,
                        15,
                        17,
                        15,
                        16,
                        8,
                        10,
                        12,
                        13,
                        8,
                        9,
                        8,
                        14,
                        8,
                        10,
                        11,
                        11,
                        12,
                        10,
                        11,
                        5,
                        0,
                        14
                    ],
                    "player_loads":5000000,
                    "segment_watched":[
                        18,
                        14,
                        13,
                        13,
                        15,
                        20,
                        12,
                        14,
                        12,
                        11,
                        19,
                        18,
                        8,
                        13,
                        11,
                        18,
                        16,
                        10,
                        15,
                        17,
                        15,
                        16,
                        8,
                        10,
                        12,
```


```
                        13,
                        8,
                        9,
                        8,
                        14,
                        8,
                        10,
                        11,
                        11,
                        12,
                        10,
                        11,
                        5,
                        0,
                        14
                    ],
                    "urls_copied": "3",
                    "plays_requested": "6500",
                    "embeds_copied": "2",
                    "facebook": "3",
                    "replays": "145",
                    "uniq_video_starts": "2000000",
                    "autoplays": "0",
                    "emails_sent": "0",
                    "digg": "0",
                    "playthrough_25": "50000",
                    "playthrough_75": "40000",
                    "twitter": "0",
                    "uniq_plays_requested": "60000",
                    "playthrough_100": "30000",
                    "time_watched": "75:33:00"

                }
            }
        ]
    }
  ],
    “total_count”:1,
    “status”:“OK”
}
```

V2 ANALYTICS API

Based on your account, different analytics packages are available.

The following are the currently offered Ooyala account packages:

- Custom Analytics Package—Custom Analytics
- Default Analytics Package—Totals, Videos, Engagement, Sharing
- Device Types Analytics Package—Device types, Platforms
- Domain Analytics Package—Domains, URLs
- Geo Analytics Package—Countries, Regions, Cities, DMAs

All packages are discussed here.

Please note that all our platform services are currently dependent on Ooyala players being the generator of analytics data.

GENERALIZED SYNTAX

With only one exception, all API requests for Analytics use the `GET` method, because you are only reading data, not writing it. (The one exception is Custom Analytics to create your own custom tags for tracking; see *Custom Analytics: Tags*.)

Thus, directly on the `GET` request, you specify various levels of detail, various types of data, or various constraints. On the route there are either *attributes* or *query string parameters* or both:

```
[GET]/v2/analytics/
reports/attributes_suchas_levels_geo_andsoforth[/
optional_attributes]?query_string_parms
```

As a result, the routes themselves can appear long or verbose but because of their verbosity, the meaning of the routes is clear.

In the syntax for the analytics routes, in the notation `[/optional_qualifiers]` the square brackets indicate that the attrubte is optional; they are not part of the real syntax.

Considerations about the Analytics API

Be aware of some notable points about the Analytics API, including differences between the analytics in the Backlot UI and the API.

Some considerations about the data available from the analytics API are noted below, some in comparison with analytics in the Backlot UI.

- **Time**

  Because they are based on different clock start times, date/time ranges in Analytics UI and Analytics API differ by one day. When specifying date ranges in Analytics UI, you specify the date range you are interested in (for example, April 1 through April 25).

  In the Analytics API, however, you need to specify the date range + 1 day. So, for the same example, to retrieve analytics for April 1 through 25, in the API you need to specify April 1 through April 26. (For details, see *the date_range attribute*.)

- **Units in Hours**

  Likewise, data involving date/time ranges is returned in increments of an hour, not by minute. That is, even if you specify a date/time range that includes a certain minute, the data returned are rounded down to the nearest hour.

- **Unidimensional Data**

  In the Analytics UI, for many of the reports, you are shown data in multiple dimensions, such as devices by geographical distribution by time (three dimensions).

  With the Analytics API, however, you might need to make multiple requests to acheve the same result. For instance, continuing the example above, you need to make one call to retrieve data related to devices and a second call to retrieve data related to geographic distribution.

- **Deriving Data**

  For other values, in many cases, the desired values can be derived from the data returned by the Analytics API. For example, with the Analytics API you can retrieve the total number of unique users for a given time period, but to derive the *average* number of unique users *per day*, you need to divide the totals by number of days involved.

Displays, Plays, and Play Starts

Ooyala Player usage is subdivided by events for displays, plays, and plays starts.

As recorded for Ooyala Analytics, the Ooyala Player categorizes video play events by three sub-events: displays, plays, and play starts. These data are gathered from discrete events in the Ooyala Player, as described below. In the Analytics reports from the Backlot UI, these events are selectable from the **Plays** menu:

Plays = Play Events

Initial Plays = Play Start Events

Player Loads = Display Events

DISPLAY EVENTS, OR "PLAYER LOADED"

Displays measure the number of times that a piece of video content is loaded and displayed within the player before it gets played (displays are related to each individual asset). Each time an embed code is changed, this event gets triggered by the Ooyala Player event `CONTENT_TREE_FETCHED`, as noted in *Player Message Bus Events*.

Loading the player does not necessarily mean a video will be played, only that player has loaded. Some scenarios in which display events are generated but there is no play include the following:

- No video autoplay, and the consumer does not click “play.”
- Autoplay enabled, but consumer abandons page before player has time to start playing.

Similarly, a single display event (or player load) can result in multiple video plays, for example if multiple videos in a playlist are viewed consecutively within the same player instance.

PLAY EVENTS

A play event measures the number of times that the "Play" button is triggered either manually or automatically. The requested content could be ad content or the actual video content (a play is counted regardless of which type of content is requested to play). Plays currently don’t include the Replay event.

This is triggered by the Ooyala Player event `WILL_PLAY_FROM_BEGINNING`, as noted in *Player Message Bus Events*.

No distinction is made between pre-roll ads and the actual video asset itself. Either constitutes a "play."

PLAY START EVENTS

A "play start" event measures the number of times that actual non-ad video content starts playing. If the user initiates the playback experience and only watches a pre-roll ad without continuing on to the actual video content, a Play count is reported, but not a Start. Play Starts is only recorded if the user waits until the actual video starts playing back.

This is triggered by the Ooyala Player event `PLAYHEAD_TIME_CHANGED`, as noted in *Player Message Bus Events*.

EXAMPLES

In this example, a player configured to autoplay plays four videos back-to-back without interruption. An assumption here is that either there are no pre-roll ads or the viewer lets all pre-roll ads play and finally views the videos.

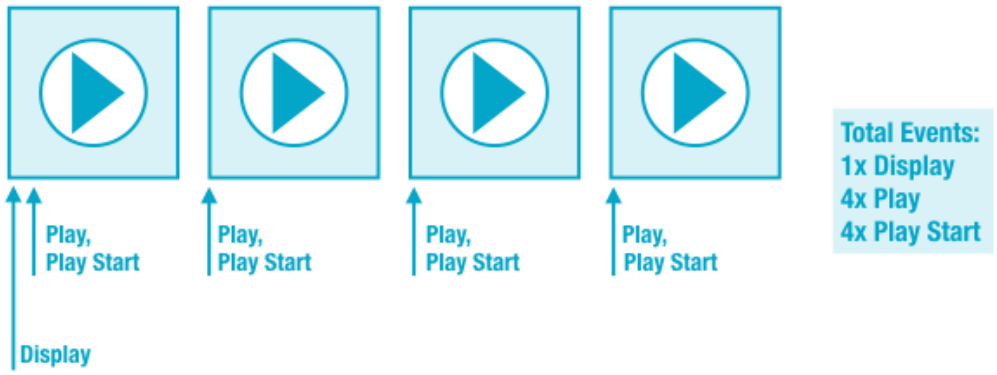


Figure 1: Autoplay Four Consecutive Videos

Below, the key difference from the first example is that the viewer abandons viewing during the pre-roll ad before the third video starts. In this example, for the third video, only the "play" event is recorded, not "play start" because the viewer abandons during the ad before the "play start" event can be recorded.

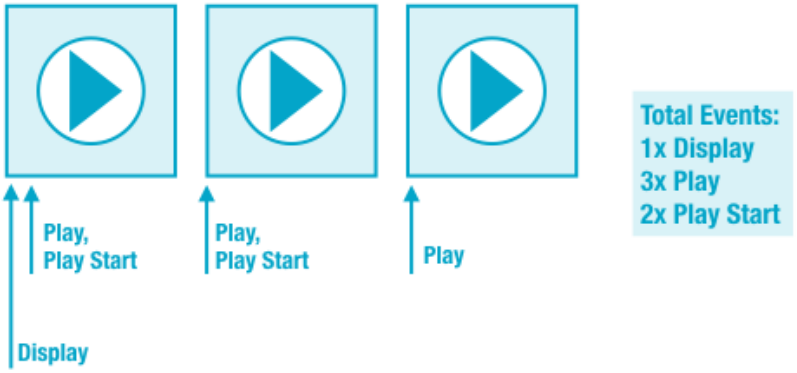


Figure 2: Autoplayer with Pre-roll Ads, Viewer Abandons during Pre-roll Ad of Third Consecutive Video

How to Export Your Data With the v2 Analytics API

INTRODUCTION

The following information will guide you on retrieving and saving your analytics v2 data.

As we migrate customers from v2 Analytics to Ooyala IQ (v3 Analytics), you need to know the following:

- All customers will have access to reprocessed data starting from January 1, 2014.
- You will continue to have access to the old v2 Analytics API until March 31, 2016.

This means that if you need access to more than 1 year of historical analytics data, you need to export the data using the v2 APIs while they are still active.

OVERVIEW: THE OOYALA V2 ANALYTICS API

With the Ooyala v2 Analytics API you can easily create a report that will provide you with your analytics data. The results will be in JSON format. (JSON is a lightweight data-interchange format that is easy to read and write.)

To retrieve analytics results from a specific date range you simply need to define the type of result you need with the api call /v2/analytics/reports/, either in a terminal emulator such as Terminal on a Mac or in the Ooyala Scratchpad.

What is an API?

In computer programming. **API (Application Programming Interface)** is the name of a set of routines and protocols for software applications. An **API** expresses a software component in terms of its operations and results.

Where can I locate my API Credentials?

You can locate your API Key and Secret in the Backlot UI. Please use your API v2 credentials located in Backlot under the ACCOUNT>Developers tab.

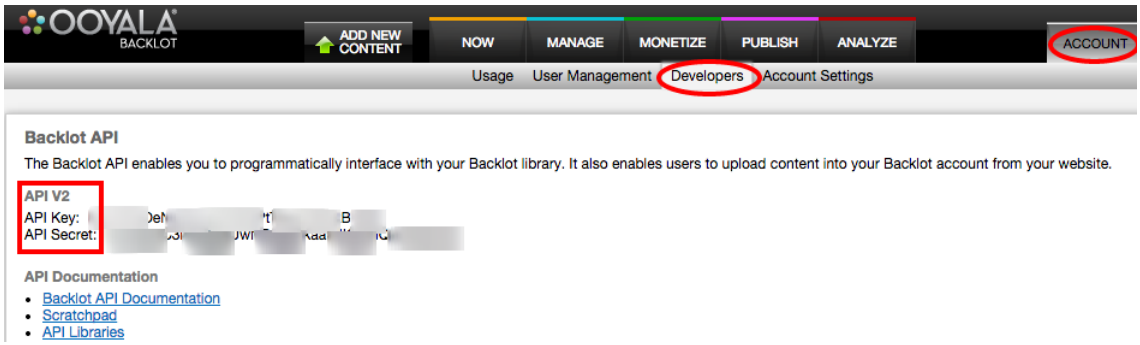


Figure 3: Finding Your API Keys

How Does the API Work?

An Ooyala API call requires 4 basic elements: API Key, API Secret, a Signature and an Expiration time.

API calls are made via *HTTP* methods. The GET API call is used to retrieve data without directly modifying it and allows you to get a typed JSON document response based on the id of the object.

What are the Available Results I can Retrieve with the Ooyala v2 Analytics API?

On the v2 Analytics API there are 4 key qualifiers that you need to identify in order to retrieve your results:

asset_id: This value is referred to by different names depending on where you look for it. In the Backlot API, asset_id is the identifier for a specific asset. asset_id has the same value as the content ID found in the Backlot UI that represents a piece of content. The value is the same for all scenarios. asset_id can be used if you want to retrieve results for a specific asset. Alternatively, you can retrieve results for your account, which would include all of your assets.

date_range: Defines the date range for this report. Analytics is based on dates. You can define the date following the format (YYYY-MM-DD) or you can define a date range with (YYYY-MM-DD...YYYY-MM-DD).

Report Request: Defines the type of report you want to retrieve. Valid values include *performance*, *sharing*, *engagement*, and *delivery*.

Dimension: Dimensions are common criteria that are used to aggregate data, such as the date when the user activity occurred or the country where the users were located. Every Dimension also has "**Drilldowns**", which allow you to filter your results by specific values for each dimension.

Common query string parameters and attributes can be found at *Common Attributes and Query String Parameters* on page 142.

HOW TO RETRIEVE YOUR V2 ANALYTICS DATA

What is the API Call Format?

The v2 Analytics API follows a specific order on the body of the API.

For example, the API call: `/v2/analytics/reports/account/performance/[:dimension/:drilldown]/:date_range`

Should appear like the following if you want to get a performance report from 2011-01-01 to 2014-01-01:

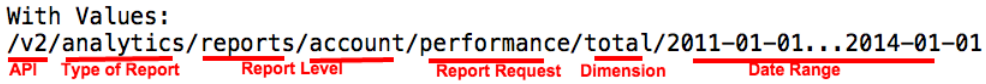


Figure 4: API Call For Performance Report

How Can I Use the Scratchpad to Save Reports?

The Scratchpad is a tool created by Ooyala that allows you to make API queries in your browser.

To retrieve an analytics report with the Scratchpad:

1. Go to *https://api.ooyala.com/docs/api_scratchpad?url=*.
2. Select "Your Account" in the Credentials section in the upper right corner of the page.
3. Enter your v2 API credentials (API Key and Secret) in the Credentials section in the upper right corner of the page.
4. In the Query field located on the left side of the page, enter your Analytics query. For example, if you would like to get the performance report from 2011 to 2014, copy and paste this query: `/v2/analytics/reports/account/performance/total/2011-01-01...2014-01-01`
5. Select GET.
6. Click Submit.

   Note: Your response appears in the response field.
7. If you prefer to see your results in a larger browser window, copy the API URL shown in green above the Submit button into your browser. In this case, the URL would be *https://api.ooyala.com/v2/analytics/reports/account/performance/total/2011-01-01...2014-01-01?api_key=yourApiKey&signature=yourSignature&expires=1418771221*.
8. Save the JSON by selecting File > Save As… in your browser. For information on how to convert JSON to CSV, see *Converting Analytics JSON to CSV*

Additional Query Examples

For more details on how to form queries in Scratchpad and for specific analytics report types, see:

- *Performance* query examples
- *Sharing* query examples
- *Engagement* query examples
- *Delivery* query examples

Note: To retrieve all data for a report type for your account, use the "total" query string parameter. You can find examples using "total" in each of the query example links mentioned above. "total" is used to retrieve all data for that particular report type for your account.

For example, the following query retrieves all performance data for the account over the date range 2011-08-01...2011-08-02.

```
[GET]/v2/analytics/reports/account/performance/total/2011-08-01...2011-08-02
```

How can I Create my own API Report?

You should only create your own API script if you are comfortable with the Ooyala API and have created scripts before or if you have the technical resources available who can modify the pre-made query to retrieve the data for you.

If you check the following snippet from our *sample code* that shows a terminal, you will be able to identify that we send the request of the API call using *cURL*. cURL makes http request where you can modify the parameters and the headers.

```
$apiSecret = "xxxxxx";
$apiKey = "xxxxxx";                                   Your API V2 Credentials

$restMethod = 'GET';                                  HTTP Method

$requestPath = '/v2/analytics/reports/account/performance/total/2011-01-01...2014-01-01';   API Call

$expDate = time() + 1209600;                          Expiration time

$parameters = array( "expires" => $expDate, "api_key" => $apiKey, );

$content = "";

$signature = createSignature($apiSecret, $restMethod, $requestPath,  $parameters, $content);   Signature

$url = 'https://api.ooyala.com'.$requestPath.'?&api_key=txxxxxxxxxxxx&expires='.$expDate.'&signature='.$signature;

echo $url;

$ch = curl_init($url);                                Curl request to the api url
curl_setopt ($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt ($ch, CURLOPT_HTTPHEADER, Array("Content-Type: application/json"));
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "GET");
curl_setopt($ch, CURLOPT_POSTFIELDS, $content);
```

Figure 5: Sample API Report

Note: For more script examples, please check our support site documentation at *Sample Code for Signing Requests*.

Performance

The asset performance report shows comprehensive performance at the account or asset level.

GET PERFORMANCE ANALYTICS (VIDEOS)

```
[GET]/v2/analytics/reports/level/performance/total/date_range
[GET]/v2/analytics/reports/account/performance/videos/date_range
```

GET PERFORMANCE ANALYTICS (GEO)

```
[GET]/v2/analytics/reports/level/performance/countries[/country]/date_range
[GET]/v2/analytics/reports/level/performance/regions/date_range
[GET]/v2/analytics/reports/level/performance/cities/date_range
[GET]/v2/analytics/reports/level/performance/countries/country/
regions[/region]/date_range
[GET]/v2/analytics/reports/level/performance/countries/country/
regions/region/cities[/city]/date_range
[GET]/v2/analytics/reports/level/performance/dmas[/dma]/date_range
```

GET PERFORMANCE ANALYTICS (DOMAINS AND URLS)

```
[GET]/v2/analytics/reports/level/performance/domains[/domain]/date_range
[GET]/v2/analytics/reports/level/performance/domains/domain/
urls[/url]/date_range
[GET]/v2/analytics/reports/level/performance/urls[/url]/date_range
```

GET PERFORMANCE ANALYTICS (PLATFORMS AND DEVICES)

```
[GET]/v2/analytics/reports/level/performance/
device_types[/devicetype]/date_range
[GET]/v2/analytics/reports/level/performance/device_types/devicetype/
platforms[/platform]/date_range
[GET]/v2/analytics/reports/level/performance/platforms[/platform]/date_range
```

EXAMPLES

This example returns video performance results:

```
[GET]/v2/analytics/reports/account/performance/
videos/2012-01-01...2012-01-24
```

which returns a response similar to the following:

```
{
   "results":[
      {
         "name":"Lady GAGA London Live Special",
         "movie_data":{
            "embed_code":"dkY3YzMzoyy7arcaUiJIoQ-9N6nz-KeG",
            "type":"Video",
            "status":"live"
         },
         "metrics":{
            "video":{
               "plays":123766,
               "video_starts":120665,
               "playthrough_50":1430,
               "uniq_plays":{
                  "weekly_uniqs":104137,
                  "monthly_uniqs":102876,
                  "daily_uniqs":107163
               },
               "playthrough_100":788,
               "playthrough_75":1171,
               "initial_plays":123467,
               "displays":136751,
               "uniq_video_starts":{
                  "weekly_uniqs":102905,
                  "monthly_uniqs":101686,
                  "daily_uniqs":105786
               },
               "uniq_displays":{
                  "weekly_uniqs":115246,
                  "monthly_uniqs":113813,
                  "daily_uniqs":118562
               },
```

```
                "time_watched":2965310785,
                "video_bytes_downloaded":623902999875,
                "playthrough_25":1952
             }
          }
       },
       {
          "name":"NISSAN Presents Premium Live Collection",
          "movie_data":{
             "embed_code":"NnbmExMzoexZjgi3lYb90VORV46sxvQV",
             "type":"Channel",
             "status":"live"
          },
          "metrics":{
             "video":{
                "plays":123486,
                "video_starts":120396,
                "playthrough_50":342,
                "replays":83,
                "uniq_replays":{
                   "weekly_uniqs":72,
                   "monthly_uniqs":71,
                   "daily_uniqs":73
                },
                "uniq_plays":{
                   "weekly_uniqs":104135,
                   "monthly_uniqs":102874,
                   "daily_uniqs":107153
                },
                "playthrough_100":76,
                "playthrough_75":178,
                "initial_plays":123469,
                "displays":136447,
                "uniq_video_starts":{
                   "weekly_uniqs":102907,
                   "monthly_uniqs":101688,
                   "daily_uniqs":105780
                },
                "uniq_displays":{
                   "weekly_uniqs":115246,
                   "monthly_uniqs":113810,
                   "daily_uniqs":118553
                },
                "time_watched":6305296573,
                "playthrough_25":969
             }
          }
       }
    }
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example returns the performance results for the account:

```
[GET]/v2/analytics/reports/account/performance/total/2011-08-01...2011-08-02
```

which returns a response similar to the following:

```
{
```

```
   "results":[
      {
         "metrics":{
            "video":{
               "plays":969,
               "playthrough_50":350,
               "uniq_replays":{
                  "weekly_uniqs":13,
                  "monthly_uniqs":13,
                  "daily_uniqs":14
               },
               "replays":15,
               "uniq_player_loads":{
                  "weekly_uniqs":681,
                  "monthly_uniqs":681,
                  "daily_uniqs":778
               },
               "uniq_plays":{
                  "weekly_uniqs":367,
                  "monthly_uniqs":367,
                  "daily_uniqs":403
               },
               "playthrough_100":229,
               "playthrough_75":293,
               "displays":2707,
               "initial_plays":693,
               "uniq_displays":{
                  "weekly_uniqs":742,
                  "monthly_uniqs":742,
                  "daily_uniqs":865
               },
               "time_watched":45596418,
               "player_loads":1971,
               "video_bytes_downloaded":4864577326,
               "playthrough_25":451
            }
         },
         "id":"total"
      }
   ]
}
```


This example returns the top two cities, ordered by plays:

```
[GET]/v2/analytics/reports/account/performance/
cities/2011-08-01...2011-08-02?limit=2&order_by=plays
```

which returns a response similar to the following:


```
{
   "results":[
      {
         "name":"Sydney",
         "geo_data":"{\"location\":{\"lat\":\"-33.883333\",\"long\":
\"151.216667\"}}",
         "id":"AU:7256:2448",
         "metrics":{
            "video":{
               "plays":99,
               "playthrough_50":30,
               "uniq_replays":{
                  "weekly_uniqs":1,
```

```
                  "monthly_uniqs":1,
                  "daily_uniqs":1
               },
               "replays":1,
               "uniq_player_loads":{
                  "weekly_uniqs":26,
                  "monthly_uniqs":26,
                  "daily_uniqs":33
               },
               "uniq_plays":{
                  "weekly_uniqs":22,
                  "monthly_uniqs":22,
                  "daily_uniqs":29
               },
               "playthrough_100":13,
               "playthrough_75":20,
               "displays":270,
               "initial_plays":77,
               "uniq_displays":{
                  "weekly_uniqs":28,
                  "monthly_uniqs":28,
                  "daily_uniqs":37
               },
               "time_watched":2711089,
               "player_loads":196,
               "video_bytes_downloaded":505785218,
               "playthrough_25":40
            }
         }
      },
      {
         "name":"Mountain View",
         "geo_data":"{\"location\":{\"lat\":\"37.3861111\",\"long\":
\"-122.0827778\"}}",
         "id":"US:9280:20142",
         "metrics":{
            "video":{
               "plays":90,
               "playthrough_50":31,
               "uniq_replays":{
                  "weekly_uniqs":2,
                  "monthly_uniqs":2,
                  "daily_uniqs":2
               },
               "replays":3,
               "uniq_player_loads":{
                  "weekly_uniqs":59,
                  "monthly_uniqs":59,
                  "daily_uniqs":82
               },
               "uniq_plays":{
                  "weekly_uniqs":34,
                  "monthly_uniqs":34,
                  "daily_uniqs":41
               },
               "playthrough_100":20,
               "playthrough_75":28,
               "displays":355,
               "initial_plays":59,
               "uniq_displays":{
                  "weekly_uniqs":67,
                  "monthly_uniqs":67,
                  "daily_uniqs":98
               },
```


```
                "time_watched":4069756,
                "player_loads":263,
                "video_bytes_downloaded":374955894,
                "playthrough_25":33
             }
          }
       }
   ],

 "next_page_token":"W1siY29sdW1uIiwiQ086MTAwNjU6NDcwMiJdLFsiZGlyZWN0aW9uIiwwXSxbImluZGV4
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example returns the top two cities in California, ordered by plays:

```
[GET]/v2/analytics/reports/account/performance/countries/US/regions/
California/cities/2011-08-01...2011-08-02?limit=2&order_by=plays
```

which returns a response similar to the following:

```
{
   "results":[
      {
         "name":"San Bruno",
         "geo_data":"{\"location\":{\"lat\":\"37.6305556\",\"long\":
\"-122.4100000\"}}",
         "id":"US:9280:20850",
         "metrics":{
            "video":{
               "plays":15,
               "uniq_player_loads":{
                  "weekly_uniqs":1,
                  "monthly_uniqs":1,
                  "daily_uniqs":1
               },
               "uniq_plays":{
                  "weekly_uniqs":1,
                  "monthly_uniqs":1,
                  "daily_uniqs":1
               },
               "displays":36,
               "initial_plays":14,
               "uniq_displays":{
                  "weekly_uniqs":1,
                  "monthly_uniqs":1,
                  "daily_uniqs":1
               },
               "time_watched":81511,
               "player_loads":29,
               "video_bytes_downloaded":41992404,
               "playthrough_25":2
            }
         }
      },
      {
         "name":"Mountain View",
         "geo_data":"{\"location\":{\"lat\":\"37.3861111\",\"long\":
\"-122.0827778\"}}",
         "id":"US:9280:20142",
```

```
         "metrics":{
            "video":{
               "plays":12,
               "playthrough_50":2,
               "uniq_replays":{
                  "weekly_uniqs":1,
                  "monthly_uniqs":1,
                  "daily_uniqs":1
               },
               "replays":2,
               "uniq_player_loads":{
                  "weekly_uniqs":18,
                  "monthly_uniqs":18,
                  "daily_uniqs":18
               },
               "uniq_plays":{
                  "weekly_uniqs":7,
                  "monthly_uniqs":7,
                  "daily_uniqs":7
               },
               "playthrough_100":1,
               "playthrough_75":2,
               "displays":58,
               "initial_plays":5,
               "uniq_displays":{
                  "weekly_uniqs":19,
                  "monthly_uniqs":19,
                  "daily_uniqs":19
               },
               "time_watched":788648,
               "player_loads":38,
               "video_bytes_downloaded":45327745,
               "playthrough_25":3
            }
         }
      }
   ],

 "next_page_token":"W1siY29sdW1uIiwiVVM6OTI4MDoyMDI4MCJdLFsiZGlyZWN0aW9uIiwwXSxbImluZGV4
}
```


Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example returns performance results for asset `4234fds34ude`:

```
[GET]/v2/analytics/reports/asset/4234fds34ude/performance/total/2011-08-01
```

which returns a response with no results:


```
{
   "response_code":404,
   "message":"No data found."
}
```


Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

Sharing

The sharing report shows the number of share actions (e.g., Facebook, Twitter) for all content within your account or for an individual asset.

GET VIDEO ANALYTICS

```
[GET]/v2/analytics/reports/account/sharing/videos/date_range
[GET]/v2/analytics/reports/account/sharing/videos/asset_id/date_range
[GET]/v2/analytics/reports/asset/asset_id/sharing/total/date_range
```

GET GEO ANALYTICS

```
[GET]/v2/analytics/reports/level/sharing/countries/[country]/date_range
[GET]/v2/analytics/reports/level/sharing/regions/date_range
[GET]/v2/analytics/reports/level/sharing/cities/date_range
[GET]/v2/analytics/reports/level/sharing/countries/country/regions/
[region]/date_range
[GET]/v2/analytics/reports/level/sharing/countries/country/regions/region/
cities/[city]/date_range
[GET]/v2/analytics/reports/level/sharing/dmas/[dma]/date_range
```

GET DOMAINS AND URLS

```
[GET]/v2/analytics/reports/level/sharing/domains/[domain]/date_range
[GET]/v2/analytics/reports/level/sharing/domains/domain/
urls[/url]/date_range
[GET]/v2/analytics/reports/level/sharing/urls[/url]/date_range
```

GET PLATFORM AND DEVICE ANALYTICS

```
[GET]/v2/analytics/reports/level/sharing/device_types/
[devicetype]/date_range
[GET]/v2/analytics/reports/level/sharing/device_types/devicetype/platforms/
[platform]/date_range
[GET]/v2/analytics/reports/level/sharing/platforms/[platform]/date_range
```

EXAMPLES

This example returns the sharing results for a single day:

```
[GET]/v2/analytics/reports/account/sharing/total/2011-08-01...2011-08-02
```

which returns a response similar to the following:

```
{
   "results":[
      {
         "id":"total",
         "metrics":{
            "syndication":{
               "embeds_copied":1,
```

```
                "digg":1,
                "facebook":2,
                "urls_copied":1,
                "twitter":2
             }
          }
       }
    ]
}
```

This example returns the top 3 cities:

```
[GET]/v2/analytics/reports/account/sharing/cities/2011-08-01...2011-08-02?
limit=3
```

which returns a response similar to the following:

```
{
   "results":[
      {
         "name":"Sydney",
         "geo_data":"{\"location\":{\"lat\":\"-33.883333\",\"long\":
\"151.216667\"}}",
         "id":"AU:7256:2448",
         "metrics":{
            "syndication":{
               "digg":1
            }
         }
      },
      {
         "name":"Singapore",
         "geo_data":"{\"location\":{\"lat\":\"1.2930556\",\"long\":
\"103.8558333\"}}",
         "id":"SG:7064:19148",
         "metrics":{
            "syndication":{
               "embeds_copied":1
            }
         }
      },
      {
         "name":"San Francisco",
         "geo_data":"{\"location\":{\"lat\":\"37.7750000\",\"long\":
\"-122.4183333\"}}",
         "id":"US:9280:20272",
         "metrics":{
            "syndication":{
               "facebook":2,
               "twitter":2
            }
         }
      }
   ]
}
```

This example returns the top 3 "sharing" cities in California:

```
[GET]/v2/analytics/reports/account/sharing/countries/US/regions/California/
cities/2011-08-01...2011-08-02?limit=3
```


which returns a response similar to the following:


```
{
   "results":[
      {
         "name":"Sydney",
         "geo_data":"{\"location\":{\"lat\":\"-33.883333\",\"long\":
\"151.216667\"}}",
         "id":"AU:7256:2448",
         "metrics":{
            "syndication":{
               "digg":1
            }
         }
      },
      {
         "name":"Singapore",
         "geo_data":"{\"location\":{\"lat\":\"1.2930556\",\"long\":
\"103.8558333\"}}",
         "id":"SG:7064:19148",
         "metrics":{
            "syndication":{
               "embeds_copied":1
            }
         }
      },
      {
         "name":"San Francisco",
         "geo_data":"{\"location\":{\"lat\":\"37.7750000\",\"long\":
\"-122.4183333\"}}",
         "id":"US:9280:20272",
         "metrics":{
            "syndication":{
               "facebook":2,
               "twitter":2
            }
         }
      }
   ]
}
```


Engagement

The engagement report displays the number of viewers as a video plays, which enables you to locate points where viewers lose interest, discover the effects of advertising placement, and assess its overall effectiveness.

GET ENGAGEMENT ANALYTICS (VIDEOS)

```
[GET]/v2/analytics/reports/asset/asset_id/engagement/total/date_range
```

GET ENGAGEMENT ANALYTICS (GEO)

```
[GET]/v2/analytics/reports/asset/asset_id/engagement/
countries[/country]/date_range
[GET]/v2/analytics/reports/asset/asset_id/engagement/regions/date_range
[GET]/v2/analytics/reports/asset/asset_id/engagement/cities/date_range
[GET]/v2/analytics/reports/asset/asset_id/engagement/countries/country/
regions[/region]/date_range
```

```
[GET]/v2/analytics/reports/asset/asset_id/engagement/countries/country/
regions/region/cities[/city]/date_range
[GET]/v2/analytics/reports/asset/asset_id/engagement/dmas[/dma]/date_range
```

GET ENGAGEMENT ANALYTICS (DOMAINS AND URLS)

```
[GET]/v2/analytics/reports/asset/asset_id/engagement/
domains[/domain]/date_range
[GET]/v2/analytics/reports/asset/asset_id/engagement/domains/domain/
urls[/url]/date_range
[GET]/v2/analytics/reports/asset/asset_id/engagement/urls[/url]/date_range
```

GET ENGAGEMENT ANALYTICS (PLATFORMS AND DEVICES)

```
[GET]/v2/analytics/reports/asset/asset_id/engagement/
device_types[/devicetype]/date_range
[GET]/v2/analytics/reports/asset/asset_id/engagement/
device_types/devicetype/platforms[/platform]/date_range
[GET]/v2/analytics/reports/asset/asset_id/engagement/
platforms[/platform]/date_range
```

EXAMPLES

This example returns the engagement data for the specified asset:

```
[GET]/v2/analytics/reports/asset/dmbHViOgvgZMqXtY40DWmh4DuW6Ie6I6/
engagement/total/2011-09-01...2011-09-05
```

which returns a response similar to the following:


```
{
  "results":[
     {
        "id":"total",
        "metrics":{
           "engagement":{
              "segments_watched":[
                 292,
                 272,
                 250,
                 230,
                 230,
                 211,
                 185,
                 165,
                 166,
                 155,
                 149,
                 139,
                 139,
                 134,
                 127,
                 120,
                 120,
                 118,
                 113,
                 110,
```


```
                110,
                108,
                105,
                95,
                96,
                91,
                89,
                85,
                85,
                86,
                85,
                80,
                81,
                81,
                98,
                77,
                78,
                79,
                77,
                105
            ],
            "percentage_watched":[
                305,
                274,
                251,
                236,
                232,
                217,
                191,
                170,
                169,
                162,
                153,
                144,
                143,
                139,
                131,
                124,
                123,
                121,
                116,
                111,
                111,
                109,
                106,
                96,
                96,
                93,
                91,
                86,
                85,
                83,
                80,
                76,
                76,
                74,
                72,
                71,
                70,
                69,
                67,
                67
            ]
        }
```


```
            }
         }
      ]
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

The `segments_watched` array describes how many viewers were watching the video at each 2 1/2 % segment. Each segment bucket is for 2 1/2 % of the content.

The `percentage_watched` array describes how many viewers watched the specified percentage of the video. In this example, 305 watched 0 - 2 1/2 % of the video and 67 viewers watched 97 1/2 - 100%. The `percentage_watched` result is not linear. Users who watched 50% of the video could have watched the first half, the second half, the 25% point to the 75% point, and so on.

Delivery

The delivery report returns the number of VOD bytes delivered for all content within your account or for an individual asset.

GET VIDEO ANALYTICS

```
[GET]/v2/analytics/reports/level/delivery/total/date_range
```

EXAMPLES

This example returns the bytes delivered for one day:

```
[GET]/v2/analytics/reports/account/delivery/total/2011-09-01
```

which returns a response similar to the following:

```
{
   "results":[
      {
         "id":"total",
         "metrics":{
            "bytes_delivered":91625071328
         }
      }
   ]
}
```

This example returns the bytes delivered for one asset:

```
[GET]/v2/analytics/reports/asset/dmbHViOgvgZMqXtY40DWmh4DuW6Ie6I6/delivery/
total/2011-09-01
```

which returns a response similar to the following:

```
{
   "results":[
      {
         "id":"total",
         "metrics":{
```

```
            "bytes_delivered":625071328
         }
      }
   ]
}
```

Custom Analytics: Tags

Tags enable you to inject custom information into Ooyala analytics that you can use to create custom reports.

CREATE A TAG

To create a new tag:

```
[PUT]/v2/analytics/tags/tag_id{
  properties
}
```

LIST TAGS

To list all the tags in your account:

```
[GET]/v2/analytics/tags
```

GET TAG

To get information about a specific tag:

```
[GET]/v2/analytics/tags/tag_id
```

DELETE TAG

To permanently delete a tag:

```
[DELETE]/v2/analytics/tags/tag_id
```

UPDATE TAG

To update a tag:

```
[PATCH]/v2/analytics/tags/tag_id{
  properties
}
```

PROPERTIES

The following table describes all properties that can be associated with a syndication.

Property	Description	Required?
display_name	The name that will appear in reports.	Conditional

Property	Description	Required?
	Type: String. Default: None. Constraints: Less than 255 characters. Conditions: Required for create operations. Example: "18 to 24 Year Olds".	
tag	ID of the tag. Type: String. Default: None. Constraints: Less than 8 alphanumeric characters, cannot contain spaces. Conditions: Required for create and delete operations. Example: "18to24".	Conditional

EXAMPLES

This example creates a tag with the name `premium_customer`:

```
[PUT]/v2/analytics/tags/premium_customer{
"display_name" : "Premium Customers"
}
```

and returns this response:

```
{
   "display_name":"Premium Customers"
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example modifies a tag:

```
[PATCH]/v2/analytics/tags/premium_customer{
   "tag":"super_premium_customer"
   "display_name":"Super Premium Customers"
}
```

and returns this response:

```
{
   "tag" : "super_premium_customer"
   "display_name" : "Super Premium Customers"
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example deletes a tag:

```
[DELETE]/v2/analytics/tags/free_customer
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

Custom Analytics: Reporting

The custom analytics report shows performance information for tags that you configured.

GET CUSTOM ANALYTICS

```
[GET]/v2/analytics/reports/account/performance/tags/{
   date_range
}
[GET]/v2/analytics/reports/asset/asset_id/performance/tags[
   date_range
]
[GET]/v2/analytics/reports/asset/asset_id/performance/tags/tag_id[
   date_range
]
```

EXAMPLES

This example returns the custom analytics for the account for a single day, sorted by displays:

```
[GET]/v2/analytics/reports/account/performance/tags/2011-08-01...2011-08-02?
order_by=displays
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

Common Attributes and Query String Parameters

These attributes and query string parameters are used by the Analytics routes.

ROUTE ATTRIBUTES

The following table describes all attributes of the route.

Route Attribute	Description
asset_id	Specifies the ID of the asset. Type: String Default: None Examples: /v2/analytics/reports/asset/fjkl3s23fdsf2/engagement
city	Returns information cities.

Route Attribute	Description
	Type: String Default: Returns top ten cities. Examples: [GET] /v2/analytics/reports/account/performance/cities/2011-08-01...2011-08-08 (all cities) [GET] /v2/analytics/reports/account/performance/countries/us/regions/california/cities/2011-08-01...2011-08-08 (all cities in California) [GET] /v2/analytics/reports/account/performance/countries/us/regions/california/cities/san+francisco/2011-08-01...2011-08-08 (San Francisco only) [GET] /v2/analytics/reports/asset/*asset_id*/performance/cities/2011-08-01...2011-08-08 (specified asset, all cities) [GET] /v2/analytics/reports/asset/*asset_id*/performance/countries/us/regions/california/cities/2011-08-01...2011-08-08 (specified asset, all cities in California) [GET] /v2/analytics/reports/asset/*asset_id*/performance/countries/us/regions/california/cities/san+francisco/2011-08-01...2011-08-08 (specified asset, San Francisco only)
country	Returns information about countries (not case-sensitive). Type: String Default: Returns top ten countries. Valid Values: For a list of valid country codes, refer to the two-letter *ISO 3166-1* country codes. For unknown countries, Backlot uses `u0`. Examples: [GET] /v2/analytics/reports/account/performance/countries/2011-08-01...2011-08-08 (all countries) [GET] /v2/analytics/reports/account/performance/countries/us/2011-08-01...2011-08-08 (US only) [GET] /v2/analytics/reports/asset/*asset_id*/performance/countries/2011-08-01...2011-08-08 (specified asset in all countries) [GET] /v2/analytics/reports/asset/*asset_id*/performance/countries/us/2011-08-01...2011-08-08 (specified asset in US only)
date_range	Specifies the date range. Type: String Default: None Examples: [GET] /v2/analytics/reports/account/performance/regions/2012-01-01 (one day) [GET] /v2/analytics/reports/account/performance/regions/2012-01-01...2012-01-02 (one day) [GET] /v2/analytics/reports/account/performance/regions/2012-01-01...2012-01-08 (weekly) [GET] /v2/analytics/reports/account/performance/regions/2012-01-01...2012-02-01 (monthly)

Route Attribute	Description
	[GET] /v2/analytics/reports/account/performance/regions/2012-01-01...2012-04-01 (quarterly) **Note:** The last date in the range is not included in the results.
devicetype	Returns information about device types. Type: String Valid Values: mobile | tablet | desktop | settop | unknown.devicetype Default: None Example: /analytics/reports/account/performance/device_types/tablet/2011-08-01...2011-08-08
dma	Returns information about designated marketing areas (DMAs). Default: Returns top ten DMAs. Examples: [GET] /v2/analytics/reports/account/performance/dmas/2011-08-01...2011-08-08 (all DMAs) [GET] /v2/analytics/reports/account/performance/dmas/new+york/2011-08-01...2011-08-08 (New York DMA) [GET] /v2/analytics/reports/asset/*asset_id*/performance/dmas/2011-08-01...2011-08-08 (specified asset, all DMAs) [GET] /v2/analytics/reports/asset/*asset_id*/performance/dmas/new+york/2011-08-01...2011-08-08 (specified asset, New York DMA)
domain	Returns information about domains. Type: String Default: Returns top ten domains. Examples: [GET] /v2/analytics/reports/account/performance/domains/2011-08-01...2011-08-08 (all domains) [GET] /v2/analytics/reports/account/performance/domains/ooyala.com/2011-08-01...2011-08-08 (ooyala.com domain) [GET] /v2/analytics/reports/asset/*asset_id*/performance/domains/2011-08-01...2011-08-08 (specified asset, all domains) [GET] /v2/analytics/reports/asset/*asset_id*/performance/domains/ooyala.com/2011-08-01...2011-08-08 (specified asset, ooyala.com domain)
level	Specifies whether the report runs at the account or asset level. When you specify asset, you must provide an asset ID. Type: String Default: None Valid Values: account | asset/*asset_id* Examples: [GET] /v2/analytics/reports/account/performance/total/2011-08-01...2011-08-08

Route Attribute	Description
	[GET] /v2/analytics/reports/asset/fjkl3s23fdsf2/performance/total/2011-08-01...2011-08-08
platform	Returns information about platforms (device types, operating systems, and browsers). Type: String Default: None Example: [GET] /v2/analytics/reports/account/performance/platforms/2011-08-01...2011-08-08 (all platforms) [GET] /v2/analytics/reports/account/performance/platforms/desktop:mac-chrome/2011-08-01...2011-08-08 (Macs running Chrome) [GET] /v2/analytics/reports/asset/*asset_id*/performance/platforms/2011-08-01...2011-08-08 (specified asset, all platforms) [GET] /v2/analytics/reports/asset/*asset_id*/performance/platforms/tablet:android-unknown.browser/2011-08-01...2011-08-08 (specified asset, iOS tablets) [GET] /v2/analytics/reports/account/performance/device_types/desktop/platforms/2011-08-01...2011-08-08 (All desktop platforms)
region	Returns information about regions (e.g., states in the United States). Type: String Default: Returns top ten regions. Examples: [GET] /v2/analytics/reports/account/performance/regions/2011-08-01...2011-08-08 (all regions) [GET] /v2/analytics/reports/account/performance/countries/us/regions/2011-08-01...2011-08-08 (all regions in the US) [GET] /v2/analytics/reports/account/performance/countries/us/regions/california/2011-08-01...2011-08-08 (California only) [GET] /v2/analytics/reports/asset/*asset_id*/performance/regions/2011-08-01...2011-08-08 (specified asset, all regions) [GET] /v2/analytics/reports/asset/*asset_id*/performance/countries/us/regions/2011-08-01...2011-08-08 (specified asset, all regions in the US) [GET] /v2/analytics/reports/asset/*asset_id*/performance/countries/us/regions/california/2011-08-01...2011-08-08 (specified asset, California only)
tag_id	The ID of the tag. Type: String Default: None Example: /analytics/tags/extreme_couch_potato
url	Returns information about the specified URL. Type: String Default: Returns top ten URLs.

Route Attribute	Description
	Example: /analytics/reports/account/performance/domains/ooyala.com/support
videos	Returns top videos. Type: String Default: 10 Example: [GET] /v2/analytics/reports/account/performance/videos/2011-08-01...2011-08-08&limit=20

QUERY STRING PARAMETERS

The following table describes the query string parameters of the routes.

Note: For clarity, the examples here are not URL-encoded; however, in actual practice, all query string parameters and values must be URL-encoded.

Parameter	Description	Required?
`breakdown_by`	Returns results by day, week, and or month instead of returning the metrics as a single value. This parameter is only supported when the dimension is **total** and there are no drilldowns. Type: String Valid Values: day | week | month Default: day Example: breakdown_by=month	No
`limit`	The maximum number of results to return. If you receive more results than the value you specify, you can get the next page of results with the paging token. Type: Integer Default: 10 Valid Values: 1-500 Example: limit=20	No
`order_by`	The field to sort on, in descending order. Any query sorted by plays, displays, or time watched will return up to 1000 results. Type: String Default: plays Valid Values: plays | displays | time_watched | video_starts | ad_requests | ad_impressions | ad_clicks Example: order_by=displays	No
`paging_token` or	URL to the next page of results.	No

Parameter	Description	Required?
`page_token`	**Note:** If the number of results for a given page is divisible by the value of `limit`, the service returns a 404 response. Type: String Default: None Example: [GET] /v2/analytics/reports/account/performance/videos/2012-01-01...2012-01-24?page_token=W1siY29sdW1uIiwiMzRTYUun1yJdLFsiaW5kZXgiLDEwXSxbInZhbHVlIiwxXV0=	

DISCOVERY API REFERENCE

ABOUT PAGING THE RESULTS FROM OOYALA DISCOVERY

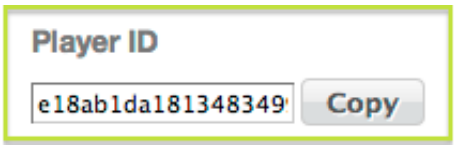
Ooyala Discovery does not support paging of results above 1,000 records.

Unlike other Ooyala APIs, Ooyala Discovery does not support the `page_token` or `next_page_token` properties.

In the context of personalization, paging of results is a function best left to client application programs. It is they who can best determine how results from the Discovery API requests must be processed. In addition, different client programs have different needs, which vary from program to program.

DISCOVERY PROFILES

The Discovery Profiles API provides routes to configure Discovery recommendation settings. When using this API, the `player_id` and the `profile_id` are the same. You can find the identifier on the lower left of the player Discovery tab in the Backlot UI:



The preferred syntax, using a slash, is shown below:

```
[PUT] /v2/players/player_id/discovery/profile
```

The response from this older form of the Discovery Profile syntax returns a less-readable response:

```
[PUT] /v2/players/player_id/discovery_profile
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

TRENDING AND POPULAR CONTENT

Trending enables viewers to discover videos that are popular or have gained significant momentum during the time period you specify.

Note: Rather than construct ad hoc requests (with individual parameters for each request), use Discovery Profiles to centralize your settings and recommendations behavior. Discovery Profiles are defined configurations of settings created by you, which you then reuse by reference on the `GET` routes. For more information, see *Managing Discovery Profiles with the Discovery API*.

GET VIDEOS WITH THE MOST MOMENTUM

```
[GET] /v2/discover/trending/momentum
```

GET THE MOST POPULAR VIDEOS

```
[GET] /v2/discover/trending/top
```

QUERY STRING PARAMETERS

The following table describes the query string parameters specific to this route.

Parameter	Description	Required?
Editorial functions	See the descriptions of editorial functions, such as `discovery_profile_id`, in *Query String Parameters*	No
`countries`	Specifies the country or countries to evaluate. Type: String Valid Values: all | valid ISO-639-1 two-digit country codes Default all Example: countries=us,uk,ca	No
`time`	Specifies the end date and time of the data to return. For example, if you specify December 31st with a monthly window, it returns results from the month of December. If you specify December 31st with a weekly window, it returns data from December 31st and the previous six days. **Note:** You can specify a date up to 30 days old. Type: String Valid Values: now | YYYY-MM-DD | YYYY-MM-DD-HH Default now Example: time=2013-12-31	No
`window`	Specifies the window of time to evaluate. For example, you might want to return the most popular videos for the day for frequent visitors and the most popular videos for the week for less frequent visitors. Type: String Valid Values: hour | day | week | month Default day Example: window=week	No

EXAMPLES

The following example returns the five most popular videos for the day:

```
[GET] /v2/discover/trending/top?countries=all&time=now&window=day&limit=5
```

The return response is similar to the following:

```
{
  "results": [
    {
      "name": "Cat Video #145678434",
      "description": "More cats!",
      "embed_code": "embed_code",
      "external_id": "external_id",
      "preview_image_url": "image_url",
      "duration": "91000",
      "hostedAtURL": null,
      "bucket_info": "bucket_info_id",
      "reason": "top"
    },
    {
      "name": "Cat Video #145678435",
      "description": "Even more cats!",
      "embed_code": "embed_code",
      "external_id": "external_id",
      "preview_image_url": "image_url",
      "duration": "91000",
      "hostedAtURL": null,
      "bucket_info": "bucket_info_id",
      "reason": "top"
    },
    {
      "name": "Cat Video #145678436",
      "description": "Yet even more cats!",
      "embed_code": "embed_code",
      "external_id": "external_id",
      "preview_image_url": "image_url",
      "duration": "91000",
      "hostedAtURL": null,
      "bucket_info": "bucket_info_id",
      "reason": "top"
    },
    {
      "name": "Cat Video #145678437",
      "description": "Can you believe it?",
      "embed_code": "embed_code",
      "external_id": "external_id",
      "preview_image_url": "image_url",
      "duration": "91000",
      "hostedAtURL": null,
      "bucket_info": "bucket_info_id",
      "reason": "top"
    },
    {
      "name": "Dog Video #0000001",
      "description": "Dogs",
      "embed_code": "embed_code",
      "external_id": "external_id",
      "preview_image_url": "image_url",
      "duration": "91000",
```

```
        "hostedAtURL": null,
        "bucket_info": "bucket_info_id",
        "reason": "top"
      }
    ]}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

SIMILAR VIDEOS

The Ooyala Discovery similar videos API returns videos that are similar to the video that you specify. For example, if a viewer is watching skydiving videos, the Ooyala Discovery algorithms might return other skydiving or extreme sports videos.

Note: Rather than construct ad hoc requests (with individual parameters for each request), use Discovery Profiles to centralize your settings and recommendations behavior. Discovery Profiles are defined configurations of settings created by you, which you then reuse by reference on the `GET` routes. For more information, see *Managing Discovery Profiles with the Discovery API*.

GET SIMILAR VIDEOS

```
[GET] /v2/discover/similar/assets/embed_code?score_type=default?
```

GET SIMILAR REMOTE ASSETS

```
[GET] /v2/discover/similar/externalassets/embed_code?score_type=default?
```

GET VIDEOS SIMILAR TO THE LIVE STREAM CONTENT

```
[GET] /v2/discover/similar/live_assets/virtual_asset_id?score_type=default?
```

EDITORIAL QUERY STRING PARAMETERS

See the description of the editorial query string parameters, such as `discovery_profile_id` in *Query String Parameters*.

Note: Rather than construct ad hoc requests (with individual parameters for each request), use Discovery Profiles to centralize your settings and recommendations behavior. Discovery Profiles are defined configurations of settings created by you, which you then reuse by reference on the `GET` routes. For more information, see *Managing Discovery Profiles with the Discovery API*.

EXAMPLES

This example returns two videos that are related to an asset ID:

```
[GET] /v2/discover/similar/assets/{asset_id}?score_type=default?
```

which returns a response similar to the following:

```
    { "results": [   {   "preview_images":
      "<previewImage>\n<url>image_url</url>\n<width>400</width>
\n<height>250</height>\n</previewImage>",
      "publishing_rule_id": "rule_id",
      "description": "Video compilation of pictures from Arthur's second
 birthday.",
      "bucket_info":
      "1{\"encoded\":\"encode_id\",\"position\":0}",
      "created_at": "2014-09-11 21:15:16 +0000",
      "content_type": "Video",
      "reason": "label:2898278",
      "name": "arthurbday.mp4",
      "hostedAtURL": null,
      "preview_image_url":
      "https://asseturl.com",
      "external_id": null,   "duration": "37033",
      "embed_code": "embed_code"
     },
     {   "preview_images":
      "<previewImage>\n<url>image_url</url>\n<width>320</width>
\n<height>180</height>\n</previewImage>",
      "publishing_rule_id": "rule_id",
      "description": "Winning score was 64.5%",
      "bucket_info":
      ...
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

DEVICE BASED PERSONALIZED RECOMMENDATIONS

The personalized recommendations API returns videos that the user of the device might like based on his or her viewing history.

Device based recommendations provide recommendations based on the type of content a user consumes on a specific device. For example, if a viewer uses his Android tablet to watch investment videos, Ooyala Discovery might return other investment and business-related videos. By default, if there is no history for a device ID, the Ooyala Discovery algorithms return content related to what the viewer is currently watching. If no content is being watched, the Ooyala Discovery algorithms return currently popular content.

With Discovery Recommendations in conjunction with Ooyala's video players, the 10 videos most recently viewed on the device are excluded from current recommendations. Custom video players with custom recommendations may also use *exclude_videos*.

Note: Rather than construct ad hoc requests (with individual parameters for each request), use Discovery Profiles to centralize your settings and recommendations behavior. Discovery Profiles are defined configurations of settings created by you, which you then reuse by reference on the GET routes. For more information, see *Managing Discovery Profiles with the Discovery API*.

GET PERSONALIZED VIDEOS

```
[GET]/v2/discover/personal/device_id
```

ROUTE ATTRIBUTES

The following table describes all attributes of the route.

Route Attribute	Description
device_id	ID of the device or user for which you want to return recommendations. Type: String Default: None Examples: pmd2xzMjrCg4TpQd6cwz9HP4U-pcFSHH

EDITORIAL QUERY STRING PARAMETERS

See the description of the editorial query string parameters in *Query String Parameters*.

Note: Rather than construct ad hoc requests (with individual parameters for each request), use Discovery Profiles to centralize your settings and recommendations behavior. Discovery Profiles are defined configurations of settings created by you, which you then reuse by reference on the `GET` routes. For more information, see *Managing Discovery Profiles with the Discovery API*.

EXAMPLES

The following example relies on a Discovery Profile that had previously been defined. For details, see *Discovery Profiles* on page 148.

```
[GET]/v2/discover/personal/th53THYsdRds?discovery_profile_id=profile_id
```

The following example returns two personalized recommendations for device `th53THYsdRds`:

```
[GET]/v2/discover/personal/th53THYsdRds?limit=2
```

which returns a response similar to the following:

```
{
   "name":"Cat Video #145678434",
   "description":"More cats!",
   "embed_code":"embed_code",
   "external_id":"external_id",
   "preview_image_url":"image_url",
   "duration":"91000",
   "hostedAtURL":null,
   "bucket_info":"bucket_info_id",
   "reason":"top"
},
{
   "name":"Cat Video #145678435",
   "description":"Even more cats!",
   "embed_code":"embed_code",
   "external_id":"external_id",
   "preview_image_url":"image_url",
   "duration":"91000",
   "hostedAtURL":null,
   "bucket_info":"bucket_info_id",
   "reason":"top"
}
```

PROVIDING FEEDBACK

To help tune recommendations, make sure to record displays and plays.

If recommendations appear in the player endscreen, you do not need to do anything. If you display the recommendations outside the player, make sure to record each display and play event.

RECORD A DISPLAY

```
[POST]/v2/discover/feedback/impression{
   bucket_info,
   device_id
}
```

RECORD A PLAY

```
[POST]/v2/discover/feedback/play{
    bucket_info,
    device_id
}
```

PROPERTIES

The following table describes the query string parameters of the routes.

Property	Description	Required?
`bucket_info`	Container. The `bucket_info` structure returned by Ooyala Discovery for trending, related, or other recommendations. **Note:** The contents of `bucket_info` need to be escaped in the JSON data.	Yes
`device_id`	ID of the device or user for which you want to return recommendations. Type: String Default: None Example: pmd2xzMjrCg4TpQd6cwz9HP4U-pcFSHH	Yes

EXAMPLES

Note: The 200 response that Ooyala returns for this route has no body text.

The following example records a display:

```
[POST]/v2/discover/feedback/impression{
   "bucket_info":"1{
 \"encoded\":\"eNpNj1EOgyAQRO+y36ZBQRAuY4hsqIkoWYiJsd69tNbGz9mZeZndIVrCOfdr
\\n7EcHBqTWnAvxalTNOFSA68fNW0QwdQURaSgH64tkj7atIGWyGf1Wup4Q3VZK
```

```
\\ndvILjfkZ+kK3IYHZIQ0L4Q8EZGe3BDju0dmGr4VT4blCIRz6aUz5XCW0kucq\
\nI3UjZCcu1THONLuUYoLxf1IJ1bDupri6fjve7Q5Lhg==\", \"position\":0
 }",
   "device_id":"device_id"
}
```

which returns a 200 response.

The following example records a play:

```
[POST]/v2/discover/feedback/play{
   "device_id":"device_id",
   "bucket_info":"1{

 \"encoded\":\"eNpVT9sOgjAM/Zc+EzNQkfEzSwPNaMItW11C0H+3eEF96+npuXSFGQON4tLs
\\nuIUaSltZcylvxSU3R8iA0sbKMhPUeQYzhUYX6BWagzlngL2fAks3OHXCIUK9\
\ngg84XnvU9aKWhdoID6oAnWIzBXobQkuBEwonUkb1wsLTGLfL+6/1iE+5aNU2\
\n6mmgxvUc5VPYmlfhWlFlq+KLrDH/3OmX+0N5edrf1lzv9nyN6dh3rqUGF7g/\\nAEHSXq0=\",
 \"position\":0}"
}
```

which returns a 200 response.

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

OOYALA SCHEDULE

With Ooyala Schedule, you combine tracks of virtual assets to play on top of live streams.

To work with schedules, there are three API routes.

- `/v2/tracks`
- `/v2/track_segments`
- `/v2/virtual_assets`

TRACKS

A track is the basis of Ooyala Schedule.

A track is composed of one or more linearly scheduled track segments.

Tracks have the following structure.

Note: The `live_stream_asset_id` property is optional.

```
{
   "id": "8ffbfcda77fb40b0af6425ae24752254",
   "name": "P12N",
   "live_stream_asset_id": "0yMTVoMzpqSB_nCyAJwvaAXSePizm6I3"
}
```

CREATE A TRACK

```
[POST]/v2/tracks{
   "name": "new track"
}
```

VIEW ALL TRACKS

```
[GET]/v2/tracks
```

VIEW ONE TRACK

```
[GET]/v2/track/track_id
```

DELETE A TRACK

```
[DELETE]/v2/track/track_id
```

MODIFY A TRACK

```
[PATCH]/v2/track/track_id{
   "name": "new track"
```


```
}
```

ROUTE ATTRIBUTES

The following table describes all attributes that can be expressed through the route.

Route Attribute	Description
track_id	The ID of the track. Type: String Default: None Example: /v2/tracks/8ffbfcda77fb40b0af6425ae24752254

PROPERTIES

The following table describes all properties that can be associated with a track.

Property	Description	Required?
live_stream_asset_id	Identifier of a live stream or remote live stream. Type: String Example: "0yMTVoMzpqSB_nCyAJwvaAXSePizm6I3"	No
name	A descriptive name for this track. Type: String Valid Values: Limit of 256 characters. Must not contain control characters, except for newline (char 10) or carriage return (char 13). Example: "my fave track"	Yes

EXAMPLES

This example creates a track:

```
[POST]/v2/tracks{
   "name": "my fave track",
}
```

Backlot returns a response similar to the following:

```
{
   "id": "2162e39c331d407d9176895f34e36af8",
   "name": "my fave track"
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

TRACK SEGMENTS

A track segment is defined as an instance of a Virtual Asset at a scheduled start/end time within a track.

Each track segment is assigned a unique identifier across all track segments in the schedule (not just within a track). Track segments in the same track cannot overlap in time, but track segments in different tracks can. Track segments have the following structure.

Note: The `external_id` and `override_stream_asset_id` properties are optional.

```
{
   "id": "4fc59badbf9a4f26be1df179082c5459",
   "track_id": "4fcb0f981d70459a9693472d6d05d7b7",
   "start_time": "2013-08-13T11:42:14Z",
   "end_time": "2013-05-13T13:49:14Z",
   "status": "online",
   "virtual_asset_id": "9a6519637be3446e920228c36da18315",
   "external_id": "PAC12_447355812",
   "override_stream_asset_id": "0yMTVoMzpqSB_nCyAJwvaAXSePizm6I3"
}
```

CREATE A TRACK SEGMENT

```
[POST]/v2/track_segments{
   properties
}
```

VIEW ALL TRACK SEGMENTS

```
[GET]/v2/track_segments
```

VIEW ONE TRACK SEGMENT

Users can use either the public ID generated by Ooyala or use the external ID to query for track segments.

```
[GET]/v2/track_segments/track_segment_id
```

INCLUDE VIRTUAL ASSET INFORMATION

You can also retrieve virtual asset information, instead of just the virtual asset id, by specifying the query string parameter `include=virtual_asset`.

```
[GET]/v2/track_segments?include=virtual_asset
```

BATCH TIME-WINDOW ACCESS TO TRACK SEGMENTS

To enable the rendering views of "What's on today" in the style of a schedule, track segments can be accessed in batch for a given time window (e.g. "what are all the track segements that start on Wednesday"). The following query returns all track segments for the 15th of August, 2013.

- You can specify one or both of `from_time` and `to_time`.
- Values of these parameters must be URI-encoded.

- The query returns all track segments between `from_time` and `to_time`.

```
[GET]/v2/track_segments?
from_time=2013-08-13-00:00:00Z&to_time=2013-08-14-00:00:00Z
```

DELETE A TRACK SEGMENT

```
[DELETE]/v2/track_segments/track_segment_id
```

MODIFY A TRACK SEGMENT

```
[PATCH]/v2/track_segments/track_segment_id
 {
 "start_time": "2012-08-13T11:30:00Z"
 }
```

ROUTE ATTRIBUTES

The following table describes all attributes that can be expressed through the route.

Route Attribute	Description
track_segment_id	The ID of the track. To get a list of tracks, perform a get against the `/tracks` route. Type: String Default: None Example: 4fc59badbf9a4f26be1df179082c5459

QUERY STRING PARAMETERS

The following table describes all parameters that can be expressed through the query string.

Parameter	Description	Required?
`from_time`	Start of time period Type: DateTime Example: [GET] /v2/track_segments? from_time=2013-08-13-00:00:00Z	No
`include=object`	To include any of the following: • virtual_asset • virtual_asset.metadata • virtual_asset.track_segment_instances • virtual_asset.asset_instances Type: String Example: [GET] /v2/track_segments? include=virtual_asset	No
`to_time`	End of time period	No

Parameter	Description	Required?
	Type: String Example: [GET] /v2/track_segments? to_time=2013-08-14-00:00:00Z	

PROPERTIES

The following table describes all properties that can be associated with a track.

Property	Description	Required?
`end_time`	Ending time of track segment Type: date/time in ISO-8601 Coordinated Universal Time (UTC) format Example: "2013-08-13T13:30:00Z"	Yes
`external_id`	Identifier for use with external systems Type: String Example: "PAC12_447355812"	No
`override_stream_asset_id`	Identifier of an asset that overrides any asset associated with the track of which this segment is a part. Type: String Example: "hvZGZmMzqaWtUHDm4RGuY9O8xsYzGV3x"	No
`start_time`	Starting time of track segment Type: date/time in ISO-8601 Coordinated Universal Time (UTC) format Example: "2013-08-13T11:00:00Z"	Yes
`status`	Status of the segment Valid Values: online | offline Type: String Example: "online"	Yes
`track_segment_id`	Identifier of track segment Type: String Example: "8ffbfcda77fb40b0af6425ae24752254"	Yes
`virtual_asset_id`	Identifier of the virtual assets associated with the track segment. Type: String Example: "9a6519637be3446e920228c36da18315"	Yes

EXAMPLES

This example creates a track segment. The `override_stream_asset_id` property is optional.

```
[POST]/v2/track_segments{
   "track_id":"8ffbfcda77fb40b0af6425ae24752254",
   "start_time":"2013-08-13T11:00:00Z",
   "end_time":"2013-08-13T13:30:00Z",
   "status":"online",
   "virtual_asset_id":"9a6519637be3446e920228c36da18315",
   "override_stream_asset_id”:"hvZGZmMzqaWtUHDm4RGuY9O8xsYzGV3x"
}
```

Backlot returns a response similar to the following:

```
{
   "end_time": "2013-08-13T13:30:00Z",
   "broadcast_status": "",
   "created_at": "2013-01-28T21:01:57Z",
   "track_id": "8ffbfcda77fb40b0af6425ae24752254",
   "id": "ecb02d1014bb46fdbcb55cfa87d2c860",
   "updated_at": "2013-01-28T21:01:57Z",
   "virtual_asset_id": "9a6519637be3446e920228c36da18315",
   "status": "online",
   "external_id": "",
   "start_time": "2013-08-13T11:00:00Z"
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

VIRTUAL ASSETS

A virtual asset represents an individual program or event.

Custom key-value pair metadata can be associated with virtual assets, following the same conventions as custom metadata for regular assets. See *Custom Metadata* on page 30.

Virtual assets in the API have the following structure.

Note: The `external_id` is an optional field for mapping the ID of virtual assets to external systems.

```
{
   "name": "Utah Football",
   "external_id": "PAC_CLF00011",
   "id": "999fffdc884841aeb05fc9e5c6fcf958"
}
```

VIEW ALL VIRTUAL ASSETS

```
[GET]/v2/virtual_assets
```

VIEW A SINGLE VIRTUAL ASSET

```
[GET]/v2/virtual_asset/virtual_asset_id
```

VIEW TRACK SEGMENTS FOR A VIRTUAL ASSET

```
[GET]/v2/virtual_assets/virtual_asset_id/track_segment_instances
```

VIEW INSTANCES OF A VIRTUAL ASSET

```
[GET]/v2/virtual_assets/virtual_asset_id/asset_instances
```

CREATE A VIRTUAL ASSET

```
[POST]/v2/virtual_assets{
   properties
}
```

MODIFY A VIRTUAL ASSET

```
[PATCH]/v2/virtual_assets/virtual_asset_id{
   "name": "Utah Football 2012"
}
```

SET/REPLACE VIRTUAL ASSET METADATA

```
[PUT]/v2/virtual_assets/virtual_asset_id/metadata{
   "name1":"value1",
   "name2":"value2",
   "name3":"value3",
   .
   .
   .
   "name100":"value100"
}
```

MODIFY CUSTOM METADATA

```
[PATCH]/v2/virtual_assets/virtual_asset_id/metadata{
   "name1":"value1",
   "name2":"value2",
   "name3":"value3",
   .
   .
   .
  "name100":"value100"
}
```


GET VIRTUAL ASSET METADATA

```
[GET]/v2/virtual_assets/virtual_asset_id/metadata
```

DELETE A VIRTUAL ASSET

```
[DELETE]/v2/virtual_assets/virtual_asset_id
```

ROUTE ATTRIBUTES

The following table describes all attributes that can be expressed through the route.

Route Attribute	Description
virtual_asset_id	The ID of the virtual asset. Type: String Default: None Example: "999fffdc884841aeb05fc9e5c6fcf958"

QUERY STRING PARAMETERS

The following table describes all parameters that can be expressed through the query string.

Parameter	Description	Required?
`include=object`	To include a virtual asset Type: CGI-escaped, comma-separated list of values below Valid Values: virtual_asset|virtual_asset.metadata| virtual_asset.track_segment_instances| virtual_asset.asset_instances Example: [GET] /v2/virtual_assets? include=virtual_asset.metadata	No

PROPERTIES

The following table describes all properties that can be associated with a track.

Property	Description	Required?
`external_id`	Identifier for use with external systems Type: String Example: "PAC_CLF00011"	No
`name`	Descriptive name for virtual asset Type: String	Yes

Property	Description	Required?
	Example: "Utah Football"	

EXAMPLES

This example creates a virtual asset:

```
[POST]/v2/virtual_assets{
   "name": "Stanford  at  Cal"
}
```

Backlot returns a response similar to the following:

```
{
   "created_at":"2013-01-28T17:35:53Z",
   "id":"53eed5ed1b734d30b364cf545d2c78c6",
   "external_id":"",
   "updated_at":"2013-01-28T17:35:53Z",
   "name":"Life with the Arts"
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

AD SETS

Ad sets are ad configurations that control the ad experience of an asset during playback.

CREATE AD SET

```
[POST]/v2/ad_sets{
   properties
}
```

ADD AN AD TO AD SET

```
[POST]/v2/ad_sets/ad_set_id/ads{
   properties
}
```

LIST AD SETS

```
[GET]/v2/ad_sets
```

GET AD SET

```
[GET]/v2/ad_sets/ad_set_id
```

MODIFY AD SET

```
[PATCH]/v2/ad_sets/ad_set_id{
   properties
}
```

DELETE AD SET

```
[DELETE]/v2/ad_sets/ad_set_id
```

REMOVE AD FROM AD SET

```
[DELETE]/v2/ad_sets/ad_set_id/ads/ad_id
```

ROUTE ATTRIBUTES

The following table describes all attributes that can be expressed through the route.

Route Attribute	Description
ad_id	The ID of the ad.

Route Attribute	Description
	Type: String Default: None Example: da1c1a4bd90c48c39d7ce3e8347bd758
ad_set_id	The ID of the ad set. To get a list of ad sets, perform a get against the `/v2/ad_sets` route. Type: String Default: None Example: /ad_sets/IzNnllMjphu2XF3_UgPROoCi9B2BwtSg

PROPERTIES

The following table describes all properties that can be associated with an ad set.

Property	Description	Required?
`default`	If 'true' then this ad set is automatically applied to videos added to your content library. Only one ad set can have this field set to 'true'. Type: String Valid Values: true | false Example: "true"	Yes
`id`	Ad set's unique identifier. Example: "Erk3j4k98dh34kshdjf892kjvmF7Diew"	Yes
`name`	Name of your Ad Set. Type: String Example: "asset"	Yes

The following table describes all properties that can be associated with an ad.

Property	Description	Required?
`ad_source`	Source of ad. Type: String Valid Values: assets | *ad_network_id*. See "Supported Ad Providers" below for a list of the supported ad providers. Default: video Example: "video"	Yes
`ad_type`	Type of ad. Type: String Valid Values: preroll | postroll | midroll | cuepoint	

Property	Description	Required?
	Note: `cuepoint` is not a valid value for the Google IMA V3 ad source. To associate your DFP ad rules with your Google IMA V3 ad, use "`ad_type`": "`rules`". Default: asset Example: "preroll"	
`click_url`	URL where to redirect the user when they click the ad. Type: String Default: None Example: "http://www.myclickurl.com"	Yes
`duration`	Indicates the length of your ad in milliseconds. Type: String Default: None Conditions: ad_source set to “assets” Example: "My Movie"	Conditional
`embed_code`	Identifier of asset used as your ad. Type: Integer Default: None Conditions: ad_source set to “assets” Example: "01M2F0MTr3SWBB9dfAPT5xEsg8Gpy-jP"	Conditional
`id`	String that uniquely associates an ad with an ad set. Type: String Valid Values: assets | *ad_network_id* Default: video Example: "video"	Yes
`name`	Identifier of asset used as your ad. Type: Integer Default: None Conditions: Required for videos. Example: "10000"	Conditional
`plays_before`	Ad is first shown after the first *x* plays. Type: Integer Default: None Example: "3"	Yes
`plays_between`	Ad is first shown every *x* plays. Type: DateTime	No

Property	Description	Required?
	Default: None Example: "3"	
position	Indicates when the ad will be played during your movie. Type: String Default: None Conditions: ad_source set to “midroll” Example: "3000"	Conditional
position_type	Indicates whether the position is measured in milliseconds or a percentage of movie duration. Type: String Valid Values: milliseconds | percent Default: None Conditions: ad_source set to “midroll” Example:	Conditional
tag_url	URL to the XML file containing your ad tag. Type: String Default: None Conditions: applies to third-party ads. Example: "http://cdn.mycdn.com/mytag2.xml"	Conditional
tracking_pixel_url	This URL is pinged whenever the ad is played. Type: String Default: None Example: "http://www.mytrackingpixel.com"	No

This example creates an ad set:

```
[POST]/v2/ad_sets{
   "name":"My new Ad Set",
   "default":true
}
```

Backlot returns a response similar to the following:

```
{
   "default":"true",
   "id":"ed4a4c6abdfc4ab8aa3e7b8a05f31ec6",
   "ads":[

   ],
   "name":"My new Ad Set"
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example adds an ad to an ad set:

```
[POST]/v2/ad_sets/ed4a4c6abdfc4ab8aa3e7b8a05f31ec6/ads{
   "ad_source":"assets",
   "ad_type":"preroll",
   "embed_code":"01M2F0MTr3SWBB9dfAPT5xEsg8Gpy-jP",
   "plays_before":0,
   "plays_between":1,
   "click_url":"http://www.myclickurl.com",
   "tracking_pixel_urls":[
      "http://www.mytrackingpixel.com"
   ]
}
```

Backlot returns a response similar to the following:

```
{
   "click_url":"http://www.myclickurl.com",
   "ad_type":"preroll",
   "id":"da1c1a4bd90c48c39d7ce3e8347bd758",
   "tracking_pixel_urls":[
      "http://www.mytrackingpixel.com"
   ],
   "ad_source":"assets",
   "embed_code":"xkdG43NjrlM92YJ1HzaqiWQ3SZpO5da7",
   "name":"bleebleblobble 2.mov",
   "duration":"5038"
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

ACTIVELY SUPPORTED AD PROVIDERS

Ooyala actively supports the following ad providers. These ad integrations are developed and maintained by the Ooyala engineering team.

- FreeWheel
- Google IMA V3
- LiveRail
- Ooyala Pulse (formerly known as Videoplaza)
- VAST 2.0
- VAST 3.0 (Player V4 only)
- VPAID 1.0 (Player V3 and Player V4 with Google IMA only)
- VPAID 2.0 (Player V4 only)

SUPPORTED AD PROVIDERS

Ooyala supports the following ad providers. These ad integrations are developed and maintained by a team at Ooyala or third party developers.

- Adapt.tv
- Tremor

- Yahoo

LEGACY AD PROVIDERS

The following ad providers were previously supported but are no longer actively updated. These modules still exist and the functionality is still there, but Ooyala does not actively update and test these integrations.

- Adify (In-Stream)
- AdTech (In-Stream, Overlay, Companion)
- Atlas (In-Stream, Companion)
- Brightroll (In-Stream, Companion)
- CBSi (In-Stream)
- DART Enterprise
- DoubleClick (via Google IMA V3 ad source)
- Google AdSense (via Google IMA V3 ad source)
- Lightningcast (In-Stream, Companion)
- Open AdStream (In-Stream, Overlay, Companion)
- OpenX (In-Stream, Overlay)
- ScanScout (In-Stream)
- SpotXchange (In-Stream, Companion)
- TV2N (In-Stream)
- YuMe*

*YuMe is now a third party provider of ads that will use Backlot VPAID modules. YuMe now uses the VPAID OPF module for Flash players. For a guide on how to handle the migration, see *Migrating from YuMe to VPAID Ad Source*.

For more information on VPAID see:

- *VAST and VPAID Ad Plugin for Player V4*
- *Integrating VPAID Ads with Player V3*

PUBLISHING RULES

Publishing rules specify when and where your content can be played. With publishing rules, you can restrict your content to specific geographic regions, ensure that it is only embedded on specific domains, and only allow it to be viewed at specific times.

When your asset is loaded, it checks its publishing rule before playing. Each rule that you create can contain any combination of these restrictions and can be assigned to multiple videos or channels.

Note: Within the Backlot UI, publishing rules are called syndication groups.

ADD NEW PUBLISHING RULE

```
[POST]/v2/publishing_rules{
   properties
}
```

LIST PUBLISHING RULES

```
[GET]/v2/publishing_rules
```

GET PUBLISHING RULE

```
[GET]/v2/publishing_rules/publishing_rule_id
```

MODIFY PUBLISHING RULE

```
[PATCH]/v2/publishing_rules/publishing_rule_id{
   properties
}
```

DELETE PUBLISHING RULE

```
[DELETE]/v2/publishing_rules/publishing_rule_id
```

ROUTE ATTRIBUTES

The following table describes all attributes that can be expressed through the route.

Route Attribute	Description
publishing_rule_id	The ID of the publishing rule. To get a list of publishing rules, perform a get against the `/publishing_rules` route. Type: String Default: None

Route Attribute	Description
	Example: /publishing_rules/ru1ze4ea4e648fd07d4509123254

PROPERTIES

The following table describes all properties that can be associated with a publishing rule.

Property	Description	Required?
`all_day`	Specifies whether the content can be played all day. Type: Boolean Valid Values: true | false Conditions: Required if type is set to "recurring" and start_time and end_time are not specified. Example: true Parent: time_restrictions	Conditional
`allowed_devices`	Specifies allowed devices. Type: Array Valid Values: iphone | ipad | android | blackberry Default: All devices are allowed Example: ["iphone", "ipad"]	No
`anonymizer_statuses`	Specifies what anonymizer statuses to block. Type: String Valid Values: private | active | suspect | inactive Conditions: Required if proxy_restrictions are specified. Example: "private", "active" Parent: proxy_restrictions	Conditional
`anonymizer_statuses`	Specifies the anonymizer statuses to block. Type: Array of strings Valid Values: private | active | suspect | inactive Ooyala uses Neustar's IP Intelligence service to identify anonymizing proxies. Neustar assigns the following statuses to IP addresses that have been detected as a proxy: • **private**: IP addresses with this designation allegedly contain anonymous proxies that are not publicly accessible. As such, they cannot be routinely tested with automated tools. These addresses usually belong to commercial ventures that sell anonymity services to the public. Addresses with this designation are derived from ownership information or obtained from trusted, high-confidence sources.	Conditional

Property	Description	Required?
	• **active**: The anonymizer tested positive within the last six months. • **suspect**: The anonymizer tested positive within the last two years, but not the last six months. • **inactive**: The anonymizer gave no positive test results within the last two years. Conditions: Required if proxy_restrictions are specified. Example: ["private","suspect"] Parent: proxy_restrictions	
`dmas`	Array that specifies one or more DMA location codes to whitelist or blacklist. Type: String Valid Values: See *DMA Codes*. Conditions: This property is optional and can only be specified if geographic_restrictions are specified. Example: `["807", "501"]` Parent: geographic_restrictions	No
`domain_restrictions`	Specifies domain restrictions. Type: Container **Note:** Only Internet domains (the so-called "hostname" portion of a URL) are matched, not full URLs. For example, the domain of the URL `http://www.mysite.com/landing_page/campaign2` is `http://www.mysite.com`. You can enter the full URL, but any portion after the domain name proper is ignored in matching.	
`domains`	Specifies one or more domains to whitelist or blacklist. Type: Array of Strings Conditions: Required if domain_restrictions are specified. Example: ["example.com", "myotherdomain.com"] Parent: domain_restrictions	Conditional

`enabled`	Enables or disbles the requirement for the Ooyala Playback Token
	
	Type: String
	
	Valid Values true | false
	
	Example:
	
	`"secure_playback_token":{`
	`"enabled":"true"`
	`}`
		No

Property	Description	Required?
	Parent: secure_playback_token	
`end_date`	Specifies the end of the time restriction. Type: DateTime Valid Values: DateTime | null Conditions: Required if time_restrictions are specified. Example: "2010-01-01T00:00:00Z" Parent: time_restrictions	Conditional
`end_time`	Specifies the end time when the content can be played. Type: Time Conditions: Required if type is set to "recurring" and all_day is not specified or set to false. Example: "23:59:59" Parent: time_restrictions	Conditional
`exception`	Specifies an exception to a geographic restriction. Exceptions for a whitelist or a blacklist differ. • An exception to a whitelist can only be used to subtract from the whitelist. • An exception to a blacklist can only be used to add more authorized locations. Type: Container Example: `"geographic_restrictions":{ "type":"blacklist", "locations":[ "US"], "dmas": ["501"], "ip_addresses": ["162.2.2.201", "162.2.2.0-162.2.2.255"], "exception":{ "ips": ["162.2.2.2", "162.2.2.5-162.2.2.10"], "locations": ["TW"]} }` Parent: geographic_restrictions	
`expiration`	Time in seconds before the Ooyala Playback Token expires Type: Integer Parent: secure_playback_token	
`geographic_restrictions`	Specifies geographic restrictions. Type: Container	

Property	Description	Required?
`ips`	Array that specifies one or more IP address/ranges to whitelist or blacklist. Type: String Valid Values: IPv4 addresses or ranges (see example) Conditions: This property is optional and can only be specified if geographic_restrictions are specified. Example: `["162.2.2.201", "162.2.2.0-162.2.2.255"]` Parent: geographic_restrictions	No
`locations`	Array that specifies one or more two-letter location codes to whitelist or blacklist. Type: String Valid Values: See *Country and Location Codes* on page 100. Conditions: Required if geographic_restrictions are specified. Example: ["US", "EU"] Parent: geographic_restrictions	Conditional
`max_concurrent_streams`	Sets maximum number of concurrent streams. Enables player Authorization API. Type: Integer Conditions: Requires restrict_concurrent_streams set to true. Example: `"secure_playback_token":{ "restrict_concurrent_streams": "true", "max_concurrent_streams": "1" }` Parent: secure_playback_token	No
`name`	Specifies the name of the publishing rule. Type: String Example: "My Rule"	Yes
`proxy_restrictions`	Specifies proxy control restrictions. Type: Container	
`proxy_restrictions`	Specifies the type of anonymizing proxies to block. Type: Container	
`recurring_days`	Specifies valid days. Type: Array of Strings	Conditional

Property	Description	Required?
	Valid Values: "MON" | "TUE" | "WED" | "THU" | "FRI" | "SAT" | "SUN" Conditions: Required if type is set to "recurring". Example: ["MON", "WED", "FRI"] Parent: time_restrictions	
`require_user_entitlement`	Require user to have been entitled to access assets, or not. Enables Rights Locker. Type: String Valid Values: true | false Example: `"secure_playback_token":{ "require_user_entitlement": "true" }` Parent: secure_playback_token	No
`restrict_concurrent_streams`	Enforce a limit on number of concurrent streams, or not. Enables player Authorization API. Type: String Valid Values: true | false Example: `"secure_playback_token":{ "restrict_concurrent_streams": "true" }` Parent: secure_playback_token	No
`secure_playback_token`	Properties related to requiring or not requiring user authorization for playback via the Ooyala Player Token Type: Container	No
`start_date`	Specifies the start of the time restriction. Type: DateTime Valid Values: DateTime | null Conditions: Required if time_restrictions are specified. Example: "2010-01-01T00:00:00Z" Parent: time_restrictions	Conditional
`start_time`	Specifies the start time when the content can be played. Type: Time Conditions: Required if type is set to "recurring" and all_day is not specified or set to false. Example: "20:00:00"	Conditional

Property	Description	Required?
	Parent: time_restrictions	
`time_restrictions`	Specifies time restrictions. Type: Container	
`type`	Specifies whether the locations are whitelisted or blacklisted. Type: String Valid Values: whitelist | blacklist Conditions: Required if geographic_restrictions are specified. Example: "whitelist" Parent: geographic_restrictions	Conditional
`type`	Specifies whether the domains are whitelisted or blacklisted. Type: String Valid Values: whitelist | blacklist Conditions: Required if domain_restrictions are specified. Example: "whitelist" Parent: domain_restrictions	Conditional
`type`	Specifies whether the time restriction is a recurring or one-time restriction. Type: String Valid Values: range | recurring Conditions: Required if time_restrictions are specified. Example: "range" Parent: time_restrictions	Conditional

EXAMPLES

This example creates a basic publishing rule with no restrictions:

```
[POST]/v2/publishing_rules{
   "name":"My Basic Publishing Rule"
}
```

Backlot returns a response similar to the following:

```
{
   "name":"My Basic Publishing Rule",
   "allowed_devices":[
      "iphone",
      "ipad",
      "android",
      "blackberry"
   ],
   "time_restrictions":{
```

```
      "start_date":null,
      "type":"range",
      "end_date":null
   },
   "domain_restrictions":{
      "domains":[

      ],
      "type":"blacklist"
   },
   "geographic_restrictions":{
      "type":"blacklist",
      "locations":[

      ]
   },
   "id":"9b70a34a678346b185f9b81c7a291d8b"
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example modifies a publishing rule to only allow the content to be embedded on mywebsite.com and only be viewed in the US or EU:

```
[POST]/v2/publishing_rules/9b70a34a678346b185f9b81c7a291d8b{
   "domain_restrictions":{
      "domains":[
         "mywebsite.com"
      ],
      "type":"whitelist"
   },
   "geographic_restrictions":{
      "type":"whitelist",
      "locations":[
         "US",
         "EU"
      ]
   }
}
```

Backlot returns a response similar to the following:

```
{
   "name":"My Publishing Rule",
   "allowed_devices":[

   ],
   "time_restrictions":{
      "start_date":"2007-04-05T00:00:00+00:00",
      "type":"range",
      "end_date":null
   },
   "domain_restrictions":{
      "domains":[
         "mywebsite.com"
      ],
      "type":"whitelist"
   },
   "geographic_restrictions":{
      "type":"whitelist",
```

```
      "locations":[
         "US",
         "EU"
      ]
   },
   "id":"79feefbe24d9424786d53b5edffb4b94"
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example lists all publishing rules:

```
[GET]/v2/publishing_rules
```

Backlot returns configuration settings for all publishing rules.

This example information about a specific publishing rule:

```
[GET]/v2/publishing_rules/79feefbe24d9424786d53b5edffb4b94
```

Backlot returns a response similar to the following:

```
{
   "name":"My Publishing Rule",
   "allowed_devices":[

   ],
   "time_restrictions":{
      "start_date":"2007-04-05T00:00:00+00:00",
      "type":"range",
      "end_date":null
   },
   "domain_restrictions":{
      "domains":[
         "mywebsite.com"
      ],
      "type":"whitelist"
   },
   "geographic_restrictions":{
      "type":"whitelist",
      "locations":[
         "US",
         "EU"
      ]
   },
   "id":"79feefbe24d9424786d53b5edffb4b94"
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example deletes a publishing rule:

```
[DELETE]/v2/publishing_rules/79feefbe24d9424786d53b5edffb4b94
```

Backlot returns a 200 response.

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

CONFIGURABLE DRM

Digital Rights Management (DRM) enables you to control how viewers consume and access your high value content through DRM policies. DRM policies can be set at the account or asset level.

Ooyala supports three types of DRM technologies: Adobe Access, Widevine, and Playready. Adobe Access is used with Flash. Widevine is used for mobile devices. Playready is used when specifically set up for your account.

The following is an example of a DRM policy:

```
{
   {
      "id":"86ff97ae7c81495eacbd9a01feff0e10",
      "name":"HD Policy",
      "playready_policy":{
         "analog_video_extension_guid":"guid_#",
         "expiration_date":"60",
         "compressed_digital_video":"500",
         "analog_video":"150",
         "uncompressed_digital_video":"300"
      },
      "flashaccess_policy":{
         "analog_output_protection":"use_if_available_all",
         "digital_output_protection":"required",
         "minimum_security_level":10000
      },
      "widevine_policy":{
         "name":"sample_hd_policy",
         "license_duration":300,
         "digital_copy_protection":"stop_video",
         "encryption_mode_indicator":"copying_prohibited",
         "analog_protection_system":"agc_on, 4_line_split_burst_on",
         "constrained_image_trigger":"none",
         "hdcp":"on",

 "analog_component_output_control":"analog_component_signal_output_allowed",

 "digital_component_output_control":"only_compressed_digital_signal_output",
         "region":"Australia"
      }
   }
}
```

Note: You can have multiple DRM policies that you can assign to your account or to individual assets, but Ooyala must configure the policies. For more information, contact your Customer Success Manager or Ooyala Support.

DRM POLICIES

You can view specific policies or all policies associated with your account.

LIST DRM POLICIES

```
[GET]/v2/drm_policies
```


VIEW A DRM POLICY

```
[GET]/v2/drm_policies/drm_policy_id
```

PROPERTIES

The following table describes all properties that can be associated with a label.

Property	Description	Required?
`id`	The ID of the policy. Type: String Example: "86ff97ae7c81495eacbd9a01feff0f22"	Yes
`name`	Name of a DRM policy. Non-Latin characters must be UTF-8 and URI-encoded. Type: String Example: "HD Policy"	Yes
`playready_policy`	Specifies properties for the PlayReady policy. Type: Container Default: none	Yes
`analog_video_extension_guid`	Specifies an extended technology that is allowed to play protected analog content, such as Macrovision. The GUID specifying the extended technology and configuration data value are comma separated. Type: String Example: C3FD11C6-F8B7-4d20-B008-1DB17D61F2DA,3 Default: none Parent: playready_policy	No
`expiration_date`	The license duration, in seconds. Type: Integer Example: 60 Default: none Parent: playready_policy	No
`compressed_digital_video`	Specifies the minimum bit rate at which compressed digital video is protected. Type: Integer Example: 500 Default: none	No

Property	Description	Required?
	Parent: playready_policy	
`analog_video`	Specifies the minimum bit rate at which analog video is protected. Type: Integer Example: 150 Default: none Parent: playready_policy	No
`uncompressed_digital_video`	Specifies the minimum bit rate at which uncompressed digital video is protected. Type: Integer Example: 300 Default: none Parent: playready_policy	No
`flashaccess_policy`	Specifies properties for the Flash Access policy. Type: Container Default: none	Yes
`minimum_security_level`	Specifies the security level required to access content. (Usually security level in DRM systems specifying the difference between development and production environments.) **Note:** This property should always either be set to 10000 or be left blank. Type: Integer Example: 10000 Default: none Parent: flashaccess_policy	No
`caching_duration`	Specifies how long a license will be cached on the client's device, in seconds. If the client has a cached license that hasn't expired, it will be used instead of obtaining a new license from Flash Access. Type: Integer Example: 3600 Default: none Parent: flashaccess_policy	No
`analog_output_protection`	Specifies the type of analog output protection to use. Type: String	No

Property	Description	Required?
	Valid Values: no_playback | no_protection | required_acp | required_all | required_cgmsa | use_if_available_acp | use_if_available_all | use_if_available_cgmsa Example: no_playback Default: none Parent: flashaccess_policy	
`digital_output_protection`	Specifies the type of digital output protection to use. Type: String Valid Values: no_playback | no_protection | required | use_if_available Example: no_protection Default: none Parent: flashaccess_policy	No
`widevine_policy`	Specifies properties for the Widevine policy. Type: Container Default: none	Yes
`widevine_modular_policy`	Specifies properties for the Widevine modular policy. Type: Container Default: none	No
`name`	Name of the Widevine policy. This must exactly match the name of the policy stored within Ooyala's systems. Type: String Example: hd_policy Default: none Parent: widevine_policy	No
`name_live`	Name of the Widevine policy to use for remote assets. This must exactly match the name of the policy within the system that is doing the Widevine transcoding. Type: String Example: hd_policy Default: none Parent: widevine_policy	No
`license_duration`	Specifies the duration of the license, in seconds. If the license expires in the middle of a video, playback should stop. Type: Integer Example: 600	No

Property	Description	Required?
	Default: none Parent: widevine_policy	
`digital_copy_protection`	Specifies how the client responds to a digital copy protection (DCP) alert. Applicable for clients that support DCP, such as PC and Mac plugins. Type: String Example: stop_video Valid Values: stop_video | do_not_stop_video Default: none Parent: widevine_policy	No
`encryption_mode_indicator`	Sets the digital copy bits restrictions, similar to the Copy Generation Management System - Analog (CGMS-A) standard. Type: String Example: copying_prohibited Valid Values: copying_not_restricted | no_further_copying_permitted | one_generation_copying_permitted | copying_prohibited Default: none Parent: widevine_policy	No
`analog_protection_system`	Specifies the type of analog protection, which is similar to the Macrovision standard. The Macrovision standard uses the Automatic Gain Control (AGC) analog signaling mechanism to let clients know that a work is copyrighted. Type: String Example: agc_on , 4_line_split_burst_on Valid Values: copy_protection_encoding_off | agc_on , split_burst_off | agc_on , 2_line_split_burst_on | agc_on , 4_line_split_burst_on Default: none Parent: widevine_policy	No
`constrained_image_trigger`	Specifies the constraint image trigger, which downsamples the quality of the video if any components in the display chain do not support HDCP. Type: String Example: none Valid Values: required | none Default: none Parent: widevine_policy	No

Property	Description	Required?
hdcp	When enabled, all components within the display chain to support High-bandwidth Digital Content Protection (HDCP). Type: Boolean Valid Values: true | false Example: true Default: none Parent: widevine_policy	No
analog_component_output_control	Specifies the analog component output control, as part of the Extended Copy Control Information (ExCCI) standards. Type: String Example: analog_component_signal_output_allowed Valid Values: analog_component_signal_output_allowed | analog_component_signal_output_not_allowed Default: none Parent: widevine_policy	No
digital_component_output_control	Specifies the digital component output control, as part of the ExCCI standards. Type: String Example: only_compressed_digital_signal_output Valid Values: only_compressed_digital_signal_output | no_digital_signal_output_allowed | only_uncompressed_digital_signal_output | both_compressed_and_uncompressed_digital_output Default: none Parent: widevine_policy	No

EXAMPLE

This example returns settings for the 86ff97ae7c81495eacbd9a01feff0e10 policy:

```
[GET]/v2/drm_policies/drm_policy_#
```

Backlot returns a response similar to the following:

```
{
   "id":"86ff97ae7c81495eacbd9a01feff0e10",
   "name":"HD Policy",
   "playready_policy":{
      "analog_video_extension_guid":"C3FD11C6-F8B7-4d20-
B008-1DB17D61F2DA,3",
      "expiration_date":"60",
      "compressed_digital_video":"500",
```

```
      "analog_video":"150",
      "uncompressed_digital_video":"300"
   },
   "flashaccess_policy":{
      "analog_output_protection":"use_if_available_all",
      "digital_output_protection":"required",
      "minimum_security_level":10000
   },
   "widevine_policy":{
      "name":"sample_hd_policy",
      "license_duration":300,
      "digital_copy_protection":"stop_video",
      "encryption_mode_indicator":"copying_prohibited",
      "analog_protection_system":"agc_on, 4_line_split_burst_on",
      "constrained_image_trigger":"none",
      "hdcp":"on",

 "analog_component_output_control":"analog_component_signal_output_allowed",

 "digital_component_output_control":"only_compressed_digital_signal_output"
   },
      "widevine_modular_policy":{
         "can_play":true,
         "can persist":true,
         "can_renew":true,
         "rental_duration_seconds":0,
         "playback_duration_seconds":0,
         "license_duration_seconds":0,
         "renewal_recovery_duration_seconds":0,
         "renewal_server_url":"",
         "renewal_delay_seconds":0,
         "renewal_retry_interval_seconds":0,
         "renew_with_usage":true
    }
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

ASSOCIATE DRM POLICY WITH AN ACCOUNT

You can associate a default DRM policy with your account. The default policy is assigned to all assets that aren't associated with a specific policy.

GET THE DEFAULT POLICY

```
[GET]/provider_drm_policy
```

ASSOCIATE POLICY

```
[PUT]/provider_drm_policy/policy_id
```

REMOVE THE DEFAULT POLICY

```
[DELETE]/v2/provider_drm_policy
```

EXAMPLE

This example associates the `86ff97ae7c81495eacbd9a01feff0f21` policy with the current account:

```
[PUT]/v2/drm_policy_#
```

Backlot returns a 200 response.

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

ASSOCIATE DRM POLICY WITH AN ASSET

You can associate a DRM policy with one or more specific assets. If a DRM policy is not associated with an asset, the default policy is used.

GET THE POLICY FOR AN ASSET

```
[GET]/assets/asset_id/drm_policy
```

ASSOCIATE POLICY WITH AN ASSET

```
[PUT]/assets/asset_id/drm_policy/drm_policy_id
```

REMOVE A POLICY FROM AN ASSET

```
[DELETE]/assets/asset_id/drm_policy
```

EXAMPLE

This example associates the `86ff97ae7c81495eacbd9a01feff0f21` policy with the IzNnllMjphu2XF3_UgPROoCi9B2rtWs asset:

```
[PUT]/v2/assets/asset_id/drm_policy/drm_policy_#
```

Backlot returns a 200 response.

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

SYNDICATIONS

A syndication consists of a list of content and a destination to which you want to syndicate the content.

DESCRIPTION

The feed type determines which streams will be used in the syndication (e.g. an mp4 stream or a source file stream) and the output format (e.g. plain MRSS, or sending the files to YouTube). Ooyala currently supports the following content destinations:

Syndication	Type	Description
3GP feed	three_gp	MRSS feed containing URLs to 3GP container files
Advanced feed	advanced	TubeMogul's extension to MRSS
Boxee feed	boxee	Feed formatted according to Boxee's extension to MRSS
Google feed	google	Google Sitemap feed
MP4 feed	mp4	MRSS feed containing URLs to the MP4 container file
Player feed	feed	MRSS feed containing player embed snippets
Podcast (iTunes)	itunes	Feed formatted according to iTunes extension to MRSS
Roku feed	roku	Roku compatible MRSS feed
Source	source	MRSS feed containing URLs to the source file.
Universal	universal	Feed using the Universal Syndication Template Language
YouTube	youtube	Syndication that pushes assets and their metadata to YouTube
iPad Adaptive Bitrate feed	ipad_abr	iPad (Adaptive Bitrate) compatible Feed
iPad Feed	ipad	iPad compatible MRSS feed
iPhone Adaptive Bitrate feed	iphone_abr	iPhone (Adaptive Bitrate) compatible MRSS feed
iPhone feed	iphone	iPhone compatible MRSS feed

SYNDICATABLE ASSET TYPES

Any type of asset except audio assets can be syndicated via universal syndication and the API requests described here.

ADD NEW SYNDICATION

To add a new syndication, you specify the name and type parameters:

```
[POST]/v2/syndications{
   properties
}
```

LIST SYNDICATIONS

To list all the syndications in your account:

```
[GET]/v2/syndications
```

GET SYNDICATION

To get a syndication:

```
[GET]/v2/syndications/syndication_id
```

UPLOAD A USTL TEMPLATE

To upload a USTL template to an advanced syndication, you include the template in the body of the request:

```
[PUT]/v2/syndications/syndication_id/template
   <template>
```

TURN ON ACCESS KEYS

To turn on access keys for a syndication, use a PATCH request to set "require_access_key" to "true" for the syndication.

```
[PATCH]/v2/syndications/syndication_id{
   "require_access_key": true
}
```

Note:

If "require_access_key" is set to true, navigating to the feed's syndication_url will no longer work. To access the feed, an "access_key=..." query string parameter must be added to the syndication_url, e.g.:

```
http://api.ooyala.com/v2/syndications/syndication_id/feed?
pcode=yourPcode&access_key=accessKey
```

CREATE ACCESS KEY FOR A SYNDICATION

To create an access key for a syndication:

```
[POST]/v2/syndications/syndication_id/access_keys{
   "name": "yourAccessKeyName"
}
```

LIST ACCESS KEYS

To list the access key associated with a syndication:

```
[GET]/v2/syndications/syndication_id/access_keys
```

REMOVE AN ACCESS KEY

To remove an access key from a syndication:

```
[DELETE]/v2/syndications/syndication_id/access_keysaccess_key
```

ROUTE ATTRIBUTES

The following table describes all attributes that can be expressed through the route.

Route Attribute	Description
access_key	The ID of the access key. To get a list of access keys associated with a syndication, perform a get against the `/syndications/syndication_id/access_keys` route. Type: String Default: None Example: /syndications/*syndication_id*/access_keys/12345678-abcd-abcd-12ab-12345678
syndication_id	The ID of the syndication. To get a list of syndications, perform a get against the `/syndications` route. Type: String Default: None Example: /syndications/d4fe34fe44ea4e648fd07d4509123254

SYNDICATION URL PARAMETERS

The following table describes all parameters that can be expressed when accessing the URL of the syndication.

Note: These cannot be specified in the route used to create syndications and modify their settings.

Parameter	Description	Required?
`limit`	The maximum number of results to return. If there are more results than the value you specify, you can get the next page of results by specifying a page_token. Type: String Default: 20	No

Parameter	Description	Required?
	Valid Values: 1-500 Example: http://api.ooyala.com/v2/syndications/6b6e72dd71bd4fa5/feed?pcode=Y4cG06BlqYMLIj&limit=50	

`paging_token`
or
`page_token`
	URL to the next page of results.
	Note: If the number of results for a given page is divisible by the value of `limit`, the service returns a 404 response.
	Type: String
	Default: None
	Example: [GET] /v2/syndications?limit=5&page_token=W1siY29sdW1uIiwiMzRTYUun1yJdLFsiaW5kZXgiLDEwXSxbInZhbHVlIiw
		No

`redirect_content`	By default, the syndications contain redirect URLs. To get the direct URL, specify false. Type: String Default: true Example: http://api.ooyala.com/v2/syndications/6b6e72dd71bd4fa5/feed?pcode=Y4cG06BlqYMLIj&redirect_content=false	No
`updated_after`	Type: DateTime Default: Returns all results Example: http://api.ooyala.com/v2/syndications/6b6e72dd71bd4fa5/feed?pcode=Y4cG06BlqYMLIj&updated_after=2011-01-01T00:00:00Z	No

PROPERTIES

The following table describes all properties that can be associated with a syndication.

Property	Description	Required?
`album_art_url`	URL to album art for the content. Apple recommends a 300x300 JPG for best results. Type: String Default: None Example: "http://mywebsite.com/myartwork.jpg" Syndication Types: Podcast (iTunes)	No
`asset_defaults`	Container element for other settings.	No
`asset_type`	The type of assets to syndicate. Type: String	No

Property	Description	Required?
	Valid Values: `video` | `ad` | `remote_asset` | `live_stream` Default: None Example: "live_stream" Syndication Types: All	
`author`	The author for content within the syndication. Type: String Default: None Example: "Various" Syndication Types: Podcast (iTunes)	No
`category`	The category to which the syndication belongs. Type: String Default: None Example: "comedy" Syndication Types: Advanced, Podcast (iTunes)	No
`description`	Description of the syndication. Type: String Default: None Example: "Contains viewer-submitted demo reels." Syndication Types: Advanced, Boxee, iPhone, iPhone (Adaptive Bitrate), iPad, MP4, Roku, Source	No
`destination_url`	Read only. The URL to the syndication. Type: String Default: None Example: "http://backlot.ooyala.com/syndication/advanced?id=fd25caa0-977e-4dbd-bbaf-fakefake" Syndication Types: Advanced, Boxee, iPhone, iPhone (Adaptive Bitrate), iPad, MP4, Roku, Source	No
`include_all_content`	Specifies whether to syndicate all content associated with the account. When set to false, Ooyala syndicates content specified by include_labels. Type: Boolean Valid Values: true | false Default: false	Conditional

Property	Description	Required?
	Conditions: Required if no labels are specified in include_labels. Example: "true" Syndication Types: All	
include_labels	Specifies labels to associate with the syndication. Any asset that is assigned to one of the specified labels is automatically included with this syndication. Type: Array of Strings Default: None Conditions: Required when include_all_content is set to false. Example: ["/label1", "/label2", "/label3"] Syndication Types: All	Conditional
keywords	Keywords for the syndication. Type: Array of Strings Default: None Example: ["exciting", "compelling", "insomnia"] Syndication Types: Podcast (iTunes)	No
name	Name of the syndication. Type: String Example: "YouTube Advertising Content" Syndication Types: All Parent: syndications	Yes
name	Name of the access key. This property is only needed when you are creating a new access key. Type: String Example: "iPhone Access Key" Parent: access_keys	Yes (if you are creating an access key)
password	Username for the account. Type: String Default: None Conditions: Required for YouTube syndications. Example: "myyoutubepassword" Syndication Types: YouTube	Conditional

Property	Description	Required?
`private`	Indicates privacy for syndication with YouTube and other services. When set to true, the asset is private. Type: String Valid Values: true | false Default: false Example: true	No
`require_access_key`	Specifies if a syndication requires an access key. When set to true, the syndication requires an access key. Type: Boolean Default: false Example: "require_access_key": true Parent: access_keys	No
`should_create_youtube_videos`	Automatically creates YouTube videos added to the syndication. Type: Boolean Valid Values: true | false Default: true Example: true Syndication Types: YouTube	No
`should_delete_youtube_videos`	Automatically deletes YouTube videos when removed from the syndication. Type: String Valid Values: true | false Default: None Example: true Syndication Types: YouTube	No
`subtitle`	The subtitle for the syndication. Type: String Default: None Example: "Full length and unedited" Syndication Types: Podcast (iTunes)	No
`title`	Title of the syndication. Type: String Default: None Example: "Best of Homemade Reality TV"	No

Property	Description	Required?
	Syndication Types: Advanced, Boxee, iPhone, iPhone (Adaptive Bitrate), iPad, MP4, Roku, Source	
`type`	The type of syndication. Type: String Valid Values: Refer to the table at the start of this section. Default: None Example: "boxee" Syndication Types: All	Yes
`unrestricted_youtube_account`	When set to true, enables you to upload videos longer than 15 minutes. Type: String Valid Values: true | false Default: false Example: false Syndication Types: YouTube **Note:** Make sure your YouTube account is enabled to allow longer videos. Otherwise, the syndication might fail. For more information, contact YouTube.	No
`username`	Username for the account. Type: String Default: None Conditions: Required for YouTube syndications. Example: "myyoutubeaccount" Syndication Types: YouTube	Conditional

EXAMPLES

This example API gets the syndication with embed code ODUzYWNiZGJiZ?:

```
[GET]/v2/syndications/ODUzYWNiZGJiZ
```

and returns this response:

```
{
   "author":null,
   "category":null,
   "description":"",
   "destination_url":"",
   "id":"ODUzYWNiZGJiZ",
   "include_all_content":true,
```

```
   "include_labels":[

   ],
   "keywords":null,
   "name":"Source MRSS",
   "should_create_youtube_videos":null,
   "should_delete_youtube_videos":null,
   "subtitle":null,
   "syndication_url":"http://api.ooyala.com/syndications/ODUzYWNiZ/feed",
   "title":"",
   "type":"source",
   "yt_username":null
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example modifies a syndication:

```
[PATCH]/v2/syndications/ODUzYWNiZGJiZ{
   title: "MY BOLD NEW TITLE",
   description: "Isn't the new title great?"
}
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example deletes a syndication:

```
[DELETE]/v2/syndications/ODUzYWNiZGJiZ
```

Note:

Try out the code samples using your account credentials in the Ooyala Scratchpad. For information about using the Scratchpad, see *The Scratchpad*.

This example displays results 11-20 of the `ODUzYWNiZGJiZ` syndication:

```
http://api.ooyala.com/v2/syndications/ODUzYWNiZGJiZ/feed?
pcode=Y4cG06BlqYMLIjPGy&limit=10#sthash.qqHfzH59.dpuf
```

This example makes the assets of the `982147856bouiguhwfoihw` syndication public:

```
[PATCH/v2/syndications/982147856bouiguhwfoihw{
   "assets_default": { "private" : false
}
```

COUNTRY AND LOCATION CODES

These country codes are used in a variety of Ooyala features.

The following table lists supported location codes:

Alpha-2 Code	English Short Name	Country Name Returned by Analytics API (If Different Than English Short Name)
AD	Andorra	
AE	United Arab Emirates	
AF	Afghanistan	
AG	Antigua and Barbuda	
AI	Anguilla	
AL	Albania	
AM	Armenia	
AN	Netherlands Antilles	
AO	Angola	
AQ	Antarctica	
AR	Argentina	
AS	American Samoa	
AT	Austria	
AU	Australia	
AW	Aruba	
AZ	Azerbaijan	
BA	Bosnia and Herzegovina	
BB	Barbados	
BD	Bangladesh	
BE	Belgium	
BF	Burkina Faso	
BG	Bulgaria	
BH	Bahrain	
BI	Burundi	
BJ	Benin	
BM	Bermuda	
BN	Brunei Darussalam	
BO	Bolivia	
BR	Brazil	Brasil
BS	Bahamas	

Alpha-2 Code	English Short Name	Country Name Returned by Analytics API (If Different Than English Short Name)
BT	Bhutan	
BV	Bouvet Island	
BW	Botswana	
BY	Belarus	
BZ	Belize	
CA	Canada	
CC	Cocos (Keeling) Islands	
CD	Congo, The Democratic Republic of the	Democratic Republic of the Congo
CF	Central African Republic	
CG	Congo	
CH	Switzerland	
CI	Cote D'Ivoire	Cote d'Ivoire
CK	Cook Islands	
CL	Chile	
CM	Cameroon	
CN	China	People's Republic of China
CO	Colombia	
CR	Costa Rica	
CU	Cuba	
CV	Cape Verde	
CX	Christmas Island	
CY	Cyprus	
CZ	Czech Republic	
DE	Germany	
DJ	Djibouti	
DK	Denmark	
DM	Dominica	
DO	Dominican Republic	
DZ	Algeria	
EC	Ecuador	
EE	Estonia	
EG	Egypt	
EH	Western Sahara	
ER	Eritrea	

Alpha-2 Code	English Short Name	Country Name Returned by Analytics API (If Different Than English Short Name)
ES	Spain	
ET	Ethiopia	
FI	Finland	
FJ	Fiji	
FK	Falkland Islands (Malvinas)	
FM	Micronesia, Federated States of	Micronesia
FO	Faroe Islands	
FR	France	
FX	France, Metropolitan	
GA	Gabon	
GB	United Kingdom	
GD	Grenada	
GE	Georgia	
GF	French Guiana	
GG	Gurnesy	
GH	Ghana	
GI	Gibraltar	
GL	Greenland	
GM	Gambia	
GN	Guinea	
GP	Guadeloupe	
GQ	Equatorial Guinea	
GR	Greece	
GS	South Georgia and the South Sandwich Islands	
GT	Guatemala	
GU	Guam	
GW	Guinea-Bissau	
GY	Guyana	
HK	Hong Kong	
HM	Heard Island and McDonald Islands	
HN	Honduras	
HR	Croatia	
HT	Haiti	
HU	Hungary	

Alpha-2 Code	English Short Name	Country Name Returned by Analytics API (If Different Than English Short Name)
ID	Indonesia	
IE	Ireland	Republic of Ireland
IL	Israel	
IM	Isle of Man	
IN	India	
IO	British Indian Ocean Territory	
IQ	Iraq	
IR	Iran, Islamic Republic of	Iran
IS	Iceland	
IT	Italy	
JE	Jersey	
JM	Jamaica	
JO	Jordan	
JP	Japan	
KE	Kenya	
KG	Kyrgyzstan	
KH	Cambodia	
KI	Kiribati	
KM	Comoros	
KN	Saint Kitts and Nevis	
KP	Korea, Democratic People's Republic of	North Korea, Korea (North)
KR	Korea, Republic of	South Korea, Korea (South)
KW	Kuwait	
KY	Cayman Islands	
KZ	Kazakhstan	
LA	Lao People's Democratic Republic	Laos
LB	Lebanon	
LC	Saint Lucia	
LI	Liechtenstein	
LK	Sri Lanka	
LR	Liberia	
LS	Lesotho	
LT	Lithuania	
LU	Luxembourg	

Alpha-2 Code	English Short Name	Country Name Returned by Analytics API (If Different Than English Short Name)
LV	Latvia	
LY	Libyan Arab Jamahiriya	Libya
MA	Morocco	
MC	Monaco	
MD	Moldova, Republic of	Moldova
MG	Madagascar	
MH	Marshall Islands	
MK	Macedonia	
ML	Mali	
MM	Myanmar	
MN	Mongolia	
MO	Macau	
MP	Northern Mariana Islands	
MQ	Martinique	
MR	Mauritania	
MS	Montserrat	
MT	Malta	
MU	Mauritius	
MV	Maldives	
MW	Malawi	
MX	Mexico	
MY	Malaysia	
MZ	Mozambique	
NA	Namibia	
NC	New Caledonia	
NE	Niger	
NF	Norfolk Island	
NG	Nigeria	
NI	Nicaragua	
NL	Netherlands	
NO	Norway	
NP	Nepal	
NR	Nauru	
NU	Niue	
NZ	New Zealand	

Alpha-2 Code	English Short Name	Country Name Returned by Analytics API (If Different Than English Short Name)
OM	Oman	
PA	Panama	
PE	Peru	
PF	French Polynesia	
PG	Papua New Guinea	
PH	Philippines	
PK	Pakistan	
PL	Poland	
PM	Saint Pierre and Miquelon	
PN	Pitcairn Islands	
PR	Puerto Rico	
PS	Palestinian Territory, Occupied	Occupied Palestinian Territory
PT	Portugal	
PW	Palau	
PY	Paraguay	
QA	Qatar	
RE	Reunion	
RO	Romania	
RU	Russian Federation	Russia
RW	Rwanda	
SA	Saudi Arabia	
SB	Solomon Islands	
SC	Seychelles	
SD	Sudan	
SE	Sweden	
SG	Singapore	
SH	Saint Helena	
SI	Slovenia	
SJ	Svalbard and Jan Mayen	
SK	Slovakia	Slovak Republic
SL	Sierra Leone	
SM	San Marino	
SN	Senegal	
SO	Somalia	
SR	Suriname	

Alpha-2 Code	English Short Name	Country Name Returned by Analytics API (If Different Than English Short Name)
ST	Sao Tome and Principe	
SV	El Salvador	
SY	Syrian Arab Republic	Syria
SZ	Swaziland	
TC	Turks and Caicos Islands	
TD	Chad	
TF	French Southern Territories	
TG	Togo	
TH	Thailand	
TJ	Tajikistan	
TK	Tokelau	
TM	Turkmenistan	
TN	Tunisia	
TO	Tonga	
TP	East Timor	
TR	Turkey	
TT	Trinidad and Tobago	
TV	Tuvalu	
TW	Taiwan	Taiwan, Republic of China Taiwan
TZ	Tanzania, United Republic of	Tanzania
UA	Ukraine	
UG	Uganda	
UM	United States Minor Outlying Islands	
US	United States	
UY	Uruguay	
UZ	Uzbekistan	
VA	Holy See (Vatican City State)	
VC	Saint Vincent and the Grenadines	
VE	Venezuela	
VG	Virgin Islands, British	Virgin Islands (British)
VI	Virgin Islands, U.S.	Virgin Islands (U.S.)
VN	Vietnam **Note:** The country name is Viet Nam in Backlot.	

Alpha-2 Code	English Short Name	Country Name Returned by Analytics API (If Different Than English Short Name)
VU	Vanuatu	
WF	Wallis and Futuna	Wallis and Futuna Islands
WS	Samoa	
YE	Yemen	
YT	Mayotte	
YU	Yugoslavia	
ZA	South Africa	
ZM	Zambia	
ZR	Zaire	
ZW	Zimbabwe	
A1	Anonymous Proxy	
A2	Satellite Provider	
O1	Other