Limelight

Performance at the Edge

Video Delivery

Content API Developers Reference

Table of Contents

Analytics API Developers Reference

Audience and Overview

This documentation is intended for programmers who are writing client or server applications that retrieve viewership data from a Limelight Video Delivery account.

Table of Contents

Analytics API Overview

The Limelight Video Delivery Analytics API gives developers programmatic access to various viewership metrics within an account. Developers can retrieve data normally viewed within the Reports tab of the Media Library console, enabling the ability to externally analyze, build custom applications, or integrate with existing systems. Common uses for the Analytics API include aggregating analytics information into custom tables or graphs or displaying analytics information on a web page without granting access to a Video Delivery account. Specifically, the Analytics API permits the following:

- Query - Request information about viewership metrics in an account

Analytics may not be accurate if accessed within 48 hours of the current time because of processing steps needed to collect, aggregate, and make data available.

REST

All API requests in the Content API are REST-based, and some are versioned. APIs are available through the following base endpoints:

Endpoint	Description
`http://api.video.limelight.com/rest/`	Use this for non-versioned APIs.

`http://api.video.limelight.com/rest/v1/`
`http://api.video.limelight.com/rest/v2/`
`http://api.video.limelight.com/rest/v3/`
	Use these for versioned APIs.

Endpoint	Description
`http://api.video.limelight.com/rest/v4/` `http://api.video.limelight.com/rest/v5/`	

Each end-point in this document includes its URL, which indicates if it is versioned or not.

Allowed values for the Content-Type header

When making API requests, you may omit the `Content-Type` header; however, if you choose to include it, the value MUST be either:

- `multipart/form-data`
- `application/x-www-form-urlencoded`

Specifying values for PUT or POST Requests

You must specify values as key:value pairs.

Pagination

For performance reasons, responses are paginated. The default and maximum page size for a response is 100. A value of `true` for `has_next` indicates that more results are available with subsequent page calls. The following optional parameters can be used to control the page of results:

Paging Parameter	Description
`page_id`	The zero-based identifier of the page to return. **Example:** `page_id=3`
`page_size`	The number of results to return per page. The default and maximum page size are 100. **Example:** `page_size=25`
`sort_by`	The field by which the results should be sorted. By default, results are sorted by popularity (in other words, Plays). However, certain calls may offer alternative sorting methods. See the details of each call below for specifics. **Example:** `sort_by=plays`
`sort_order`	The order in which the results should display. Specify `asc` for ascending or `desc` for descending. Results will display in descending order by default. **Example:** `sort_order=desc`

Filtering Results

By default, API calls will return the analytics for all media. However, for certain calls, you can limit the response to media that only meet specified criteria. One example is if you only wanted to see the analytics for all `published` media with a `football` tag. To accomplish this, pass an extra parameter to the API call, specifically, `and=state:published;tag:football`.

To apply any filter to an API call, specify either of the following parameters:

- `and = <criteria represented as <key>:<value> pairs, separated by semi-colons. All criteria must be true for a video to be returned>`
- `or = <criteria represented as <key>:<value> pairs, separated by semi-colons. At least one of the criteria must be true for a video to be returned>`

Filter criteria for the above parameters can be built using any of the following media properties. All criteria are case insensitive:

Key	Value
`title`	Any word or phrase in the title.
`description`	Any word or phrase in the description.
`original_filename`	Any word or phrase in the original filename.
`tag`	Any word or phrase in the tag.
`state`	An exact match of one of the following: `new, uploading, processing, publishable, published, error`
`media_type`	An exact match of one of the following: `video, audio, livestream`
`channel_id`	The ID of a channel where the search should be limited.
`created_after`	Date, represented in Unix Time. Limits search to media created after a particular date.
`updated_after`	Date, represented in Unix Time. Limits search to media updated after a particular date.
`published_after`	Date, represented in Unix Time. Limits search to media published after a particular date.
`custom_property[<custom_property_id>]` The ID of a custom property can be found by accessing **Settings** then **Custom Properties** in the MediaLibrary console. Right-click on the property name to reveal. Click to copy.	Any word or phrase in the value of the custom property

Authentication

To ensure the security of your information, we require some requests to be signed. Those methods indicated below as requiring authentication must have a signature. See Authentication - Signing Requests below for how to sign requests.

Errors

Standard errors are returned with 4xx codes and a descriptive message. Internal server errors are returned with a status code of 500.

Deprecation Policy

Technology moves fast, and we want to keep our developer API at the innovative forefront of new technologies and techniques. At times, Limelight will deprecate certain API calls. Limelight will announce if it intends to discontinue or make backward-incompatible changes to the API. Following a deprecation notice, we will use commercially reasonable efforts to continue to operate the deprecated API for one year after the announcement.

Server-Side Caching

To ensure a fast, reliable, and responsive site, we recommend that all interaction with the Analytics API be done on the server. For example, don't try to call the API on every page load of your hugely popular website. Instead, call the API to periodically update a local cache and save the response, displaying your cached version on your site. Caching is essential to ensuring the reliability of your web application, ensuring robustness, and preventing overall network load. For information on a suggested technique to implement caching, see File Swap Caching.

Results by Date

To search results by date, you must provide a start date and an end date. Please note that the provided end date will not be included in your search results calculation.

For example, the end date of `2020-01-24` means `2020-01-24 00:00:00` or midnight before 2020-01-24. The analytics for 2020-01-24 will not be included in your results. To get the analytics for 2020-01-24, the end date of the query will need to be 2020-01-25.

The provided dates are interpreted using the timezone of the publisher specified during account creation. The default timezone is America/Los_Angeles.

For example, if you make a query for 2020-01-24 > 2020-02-24 with an account set to America/Los_Angeles, the data returned will be from `2020-01-24 00:00:00 PST` to `2020-02-24 00:00:00 PST` - midnight to midnight, in the account's timezone. The account timezone is used regardless of where the query originates to ensure the data returned is consistent across users in different timezones.

Results Format

All results are returned in `JSON` format by default. You can specify an alternative response format (such as XML or CSV) by applying a **dot extension** on the end of the resource path. For example, to request the media performance metrics in XML, add **.xml**:

```
http://api.video.limelight.com/rest/v2/organizations/<org
id>/analytics/performance/media.xml
```

API Requests - Performance Metrics

The performance metrics provide metrics for media, channels, or traffic sources for a given time.

The following methods are available for requesting performance metrics:

Performance Overview Metrics

Media Performance Metrics

Channel Performance Metrics

Traffic Sources (Domains) Performance Metrics

Performance Overview Metrics Aggregated by Sub-organizations

Performance Overview Metrics

Data for media performance metrics can be retrieved programmatically via the following method:

<table>
<tr>
<td>Get an overview of your analytics data</td>
<td>
<b>Get a snapshot of your analytics data (media, channels, domain).</b>
<table>
<tr><td><b>URL</b></td><td>`http://api.video.limelight.com/rest/v1/organizations/<org id>/analytics/performance/overview`</td></tr>
<tr><td><b>Formats</b></td><td>`XML, JSON, CSV`</td></tr>
<tr><td><b>HTTP Method</b></td><td>`GET`</td></tr>
<tr><td><b>Requires Authentication</b></td><td>Yes (See Authentication - Signing Requests for Authentication Instructions.)</td></tr>
<tr><td><b>Required Parameters</b></td><td>`start:` Date string, represented as yyyy-MM-dd. For example, 2018-03-01. See Results by Date for more details.
`end:` Date string, represented as yyyy-MM-dd. For example, 2018-03-31. See Results by Date for more details.</td></tr>
<tr><td><b>Optional Parameters</b></td><td>None</td></tr>
<tr><td><b>Paging Parameters</b></td><td>`page_id:` The zero-based identifier of the page to return.
`page_size:` The number of results to return per page. (default/max: 100)
`sort_by:` The field by which the results should be sorted {`title`, `plays`, `mediaId`}. (default: `plays`)
`sort_order:` The order in which the results should display {`asc`, `desc`}. (default: `desc`)</td></tr>
<tr><td><b>Filter Parameters</b></td><td>None</td></tr>
<tr><td><b>Response</b></td><td>A list containing the overview of media analytics over the given time.
Data returned by a call to this method:
<table>
<tr><th>Property Name</th><th>Description</th></tr>
<tr><td>`media_overviews`</td><td>A list of media overview objects. Each object has the following properties:
`media_id`: A unique identifier for the media.
`media_type`: Either `audio` or `video`.
`media_title`: The title of the media.
`num_views`: The number of times the media was played in a time frame.
`total_time_viewed_in_milliseconds`: Total amount of time the media has been viewed (in milliseconds).</td></tr>
</table>
</td></tr>
</table>
</td>
</tr>
</table>

Property Name	Description
`channel_overviews`	A list of channel overview objects. Each object has the following properties: `channel_id`: A unique identifier for the channel. `channel_title`: The title of the channel. `num_views`: The number of times the media was played in a time frame.
`referrer_overviews`	A list of referrer overview objects. Each object has the following properties: `referrer_url_host`: The domain name of the referring website where embedded content has been played. `num_views`: The number of times the media has been played on the referring domain.

Errors	Both start and end dates are required, and the start date must come before the end date.
Example Usage	Example code for this method can be found in Example Code.

Media Performance Metrics

Note: The Media Performance Metrics endpoint provides a broad set of media metrics for a given time. Please see Media Engagement Metrics if you are interested in more specific metrics that reflect how viewers have interacted with media.

Data for media performance metrics can be retrieved programmatically via the following method:

Get a listing of the most popular media

Get a list of the most popular (most played) media for a timeframe.

URL	`http://api.video.limelight.com/rest/v4/organizations/<org id>/analytics/performance/media`
Formats	`XML, JSON, CSV`
HTTP Method	GET
Requires Authentication	Yes (See Authentication - Signing Requests for Authentication Instructions.)
Required Parameters	• `start`: Date string, represented as yyyy-MM-dd. For example, 2018-03-01. See Results by Date for more details.

	- `end`: Date string, represented as yyyy-MM-dd. For example, 2018-03-31. See Results by Date for more details.
Paging Parameters (Optional)	`page_id`: The zero-based identifier of the page to return `page_size`: The number of results to return per page (default/max: 100) `sort_by`: The field by which the results should be sorted `{title, plays, distinctViewers, distinctReferrers, replays, averageDailyDistinctPlays, averageDailyDistinctReferrers, originalFilename, totalTimeViewed, loads, averageTimeViewed, percentPlayStarts, averageTimeFirstFrame, percentBuffering, percentSessionsBuffering, playStarts, percentViewsCompleted`} (default: `plays`) `sort_order`: The order in which the results should display {`asc`, `desc`} (default: `desc`)

Filter Parameters (Optional)	**Note:** By default, this API will return the analytics for all media objects played during the specified time. Media objects which were not viewed during the specified time will NOT be included in the analytics results. If desired, you can limit the response to media that only meets specified criteria. See Filtering Results for more information.
	`and = <filter criteria represented as <key>:<value> pairs, separated by semi-colons.>` All criteria must be true for a video to be returned
	Example: limit response to only published media with a tag of football:
	`and=tag:football;state:published`
	`or = <filter criteria represented as <key>:<value> pairs, separated by semi-colons>`
	`At least one of the criteria must be true for a video to be returned>`
	Example: limit response to only media with a tag of football or a tag of hockey:
	`or=tag:football;tag:hockey`

Response	A list containing the detailed media analytics over the given period Data returned by a call to this method:

Property Name	Description
`media_id`	A unique identifier for the media.
`media_title`	The title of the media.
`media_original_filename`	The original name of the file that was uploaded.
`media_type`	Either `audio` or `video`.
`num_media_distinct_viewers_v2`	The number of unique viewers of the media (**Unique Plays**). deprecated
`num_media_distinct_referrers_v2`	The number of unique domains on which the video has been embedded and played (**Traffic Sources**). deprecated
`average_daily_distinct_plays`	The average number of unique viewers of the media for 24 hours (**Unique Plays**).
`average_daily_distinct_referrers`	The average number of unique domains on which the video has been embedded and played for 24 hours (**Traffic Sources**).
`num_media_plays`	The number of times the media has been played.
`num_media_replays`	The number of times the media has been replayed.
`total_media_time_viewed`	The total amount of time the media has been viewed (in milliseconds).
`ave_media_time_viewed`	The average amount of time the media has been viewed (in milliseconds).
`num_media_loads`	The number of times a media was loaded.
`num_media_`	The number of times playback was begun.

Property Name	Description
`play_starts`	
`percent_media_play_starts`	Percent of sessions in which playback was begun.
`ave_media_time_first_frame`	The average amount of time between play being clicked and the first frame is presented (in milliseconds).
`percent_media_time_buffering`	Percent of time watching a media that was spent buffering.
`percent_media_sessions_buffering`	The percentage of sessions that experienced at least one rebuffer event.
`percent_media_views_completed`	The percentage of views that watched the media to completion.

Errors	• You supplied an invalid value: `page_id` must be `>= 0` • You supplied an invalid value: `page_size` must be in the range `[1..100]` • The sort field specified was invalid. Please check your sort field and try again. • The sort order specified was invalid. Please check your sort order and try again. • Both start and end dates are required, and the start date must come before the end date. • You supplied an invalid search field
Example Usage	Example code for this method can be found in Example Code.

Channel Performance Metrics

Data for channel performance metrics can be retrieved programmatically via the following method:

Get full performance analytics for channels	**Get comprehensive channel performance metrics.**

<table>
<tr><td><b>URL</b></td><td><code>http://api.video.limelight.com/rest/v4/organizations/<org id>/analytics/performance/channels</code></td></tr>
<tr><td><b>Formats</b></td><td><code>XML, JSON, CSV</code></td></tr>
<tr><td><b>HTTP Method</b></td><td><code>GET</code></td></tr>
<tr><td><b>Requires Authentication</b></td><td>Yes (See Authentication - Signing Requests for Authentication Instructions.)</td></tr>
<tr><td><b>Required Parameters</b></td><td><code>start</code>: Date string, represented as yyyy-MM-dd. For example, 2015-03-01. See Results by Date for more details.
<code>end:</code> Date string, represented as yyyy-MM-dd. For example, 2015-03-31. See Results by Date for more details.</td></tr>
<tr><td><b>Paging Parameters</b></td><td><code>page_id:</code> The zero-based identifier of the page to return
<code>page_size:</code> The number of results to return per page (default/max: 100)
<code>sort_by:</code> The field by which the results should be sorted {<code>title</code>, <code>plays</code>, <code>distinctViewers</code>, <code>distinctReferrers</code>, <code>replays</code>, <code>averageDailyDistinctPlays</code>, <code>averageDailyDistinctReferrers</code>, <code>percentViewsCompleted</code>, <code>averageTimeViewed</code>, <code>plays</code>, <code>totalTimeViewed</code>, <code>playStarts</code>, <code>percentBuffering</code>, <code>percentSessionsBuffering</code>, <code>averageTimeFirstFrame</code>, <code>percentPlayStarts</code>, <code>loads</code>} (default: <code>plays</code>)
<code>sort_order:</code>The order in which the results should display {<code>asc</code>, <code>desc</code>} (default: <code>desc</code>)</td></tr>
<tr><td><b>Response</b></td><td>The list containing the detailed channel analytics over the given time.
Data returned by a call to this method:
<table>
<tr><th>Property Name</th><th>Description</th></tr>
<tr><td><code>chan_id</code></td><td>A unique identifier for the channel.</td></tr>
<tr><td><code>chan_title</code></td><td>The title of the channel.</td></tr>
<tr><td><code>num_chan_distinct_viewers_v2</code></td><td>The average number of unique viewers of the channel for 24 hours.
deprecated</td></tr>
<tr><td><code>num_chan_distinct_</code></td><td>The average number of unique domains on which the video has been embedded</td></tr>
</table></td></tr>
</table>

Property Name	Description
`referrers_v2`	and played for 24 hours. deprecated
`num_chan_plays`	The number of times the channel has been played.
`total_chan_time_viewed`	The total amount of time the channel has been viewed (in milliseconds).
`ave_chan_time_viewed`	The average amount of time the channel has been viewed (in milliseconds).
`num_chan_loads`	The number of times a channel was loaded.
`num_chan_play_starts`	The number of times playback was begun.
`percent_chan_play_starts`	Percent of sessions in which playback was begun.
`ave_chan_time_first_frame`	The average amount of time between play being clicked and the first frame is presented (in milliseconds).
`percent_chan_time_buffering`	Percent of time watching a channel that was spent buffering.
`percent_chan_sessions_buffering`	The percentage of sessions that experienced at least one rebuffer event.
`percent_chan_views_completed`	The percentage of views that watched the channel to completion.

Errors

- You supplied an invalid value: `page_id` must be `>= 0`
- You supplied an invalid value: `page_size` must be in the range `[1..100]`
- The sort field specified was invalid. Please check your sort field and try again.
- The sort order specified was invalid. Please check your sort order and try again.
- Both start and end dates are required, and the start date must

		come before the end date.

Traffic Sources (Domains) Performance Metrics

Data for the Top Traffic Sources (Domains) performance metrics can be retrieved programmatically via the following method:

Get a list of traffic sources (domains)	**Get a list of all traffic sources (domains) where any media has been played.**	
	URL	`http://api.video.limelight.com/rest/v3/organizations/<org id>/analytics/performance/domain`
	Formats	`XML, JSON, CSV`
	HTTP Method	`GET`
	Requires Authentication	Yes (See Authentication - Signing Requests for Authentication Instructions.)
	Required Parameters	`start:` Date string, represented as yyyy-MM-dd. For example, 2018-03-01. See Results by Date for more details. `end:` Date string, represented as yyyy-MM-dd. For example, 2018-03-31. See Results by Date for more details.
	Paging Parameters (Optional)	`page_id:` The zero-based identifier of the page to return. (default: `0`) `page_size:` The number of results to return per page. (default/max: `100`) sort_by: The field by which the results should be sorted> {`plays`, `domain`, `totalTimeViewed`, `loads`, `replays`, `distinctViewers`, `averageDailyDistinctPlays`, `averageTimeViewed`, `percentPlayStarts`, `averageTimeFirstFrame`, `percentBuffering`, `percentSessionsBuffering`, `playStarts`, `percentViewsCompleted`} (default: `plays`) `sort_order:`The order in which the results should display {`asc`, `desc`}. (default: `desc`)
	Response	A list of domains representing the traffic sources for the given timeframe. Data returned by a call to this method:

Property	**Name**
`domain`	The domain name of the referring

Property	Name
	website where embedded content has been played.
`total_domain_time_viewed`	The total time viewed (in milliseconds) on the referring domain.
`ave_domain_time_viewed`	The average time viewed (in milliseconds) on the referring domain.
`num_domain_plays`	The number of times the media has been played on the referring domain.
`num_domain_distinct_viewers_v2`	The number of distinct viewers on the referring domain.
`num_domain_loads`	The number of times a media from that domain was loaded.
`num_domain_replays`	The number of times the media has been replayed on the referring domain.
`num_domain_play_starts`	The number of times playback was begun.
`percent_domain_play_starts`	Percent of sessions in which playback was begun.
`ave_domain_time_first_frame`	The average amount of time between play being clicked and the first frame is presented (in milliseconds).
`percent_domain_time_buffering`	Percent of time watching a media that was spent buffering.
`percent_domain_sessions_buffering`	The percentage of sessions that experienced at least one rebuffer event.
`percent_domain_views_completed`	The percentage of views that watched the media to completion.

Errors	• You supplied an invalid value: `page_id` must be `>= 0` • You supplied an invalid value: `page_size` must be in the range `[1..100]` • The sort field specified was invalid. Please check your sort field and try again. • The sort order specified was invalid. Please check your sort order and try again. • Both start and end dates are required, and the start date must come before the end date.
Example Usage	Example code for this method can be found in Example Code.

Performance Overview Metrics Aggregated by Sub-organizations

An overview of your analytics data for your organization and your child organizations can be retrieved programmatically via the following method:

Get an overview of your analytics data for your organization and your child organization

Get a snapshot of your and your child organization's analytics data (media, channels, domain).

URL	`http://api.video.limelight.com/rest/v1/organizations/<org id>/analytics/performance/overview/suborgs`
Formats	`None`
HTTP Method	`GET`
Requires Authentication	Yes (See Authentication - Signing Requests for Authentication Instructions.)
Required Parameters	`start:` Date string, represented as yyyy-MM-dd. For example, 2018-03-01. See Results by Date for more details. `end:` Date string, represented as yyyy-MM-dd. For example, 2018-03-31. See Results by Date for more details.
Response	A list of video play metrics aggregated by sub-organization. Each organization's list contains the following sub-lists: • `media_overviews` • `media_overviews_by_num_shares` • `channel_overviews` • `referrer_overviews`
Errors	• Media not owned by the organization. • Invalid value. • Channel does not exist. • Missing signature.

API Requests - Engagement Metrics

> **Note:** The Media Engagement Metrics report is only calculated once each day based on the publisher's timezone (typically PST). You must wait until the day has been completed in your timezone for the data to appear in the daily reports.

The following method is available for requesting media **Engagement** information for an account:

Media Engagement Metrics

Media Engagement Metrics

The media engagement metrics reflect viewer engagement with media, such as the number of plays and total time viewed.

Data for these metrics can be retrieved programmatically via the following method:

Get engagement metrics for media	**Get a comprehensive listing of media engagement metrics.**	
	URL	`http://api.video.limelight.com/rest/v4/organizations/<org id>/analytics/engagement/media`
	Formats	`XML, JSON, CSV`
	HTTP Method	`GET`
	Requires Authentication	Yes (See Authentication - Signing Requests for Authentication Instructions.)
	Required Parameters	`start:` Date string, represented as yyyy-MM-dd. For example, 2018-03-01. See Results by Date for more details. `end:` Date string, represented as yyyy-MM-dd. For example, 2018-03-31. See Results by Date for more details.
	Paging Parameters (Optional)	`page_id:` The zero-based identifier of the page to return. `page_size:` The number of results to return per page. (default/max: `100`) `sort_by`: The field by which the results should be sorted `{title, plays, averageTimePlayed, totalTimePlayed, milestone25, milestone50, milestone75, milestone100}`. (default: `plays`) `sort_order:` The order in which the results should display {`asc`, `desc`}. (default: `desc`)
	Filter Parameters (Optional)	**Note:** By default, this API will return the analytics for all media objects played during the specified time. Media

objects which were not viewed during the specified time will NOT be included in the analytics results. If desired, you can limit the response to media that only meets specified criteria. See Filtering Results for more information.

`and: <filter criteria represented as <key>:<value> pairs, separated by semi-colons.>` All criteria must be true for a video to be returned.

Example: limit response to only published media with a tag of football:

`and=tag:football;state:published`

`or = <filter criteria represented as <key>:<value> pairs, separated by semi-colons.>` At least one of the criteria must be true for a video to be returned.

Example: limit response to only media with a tag of football or a tag of hockey:

` or=tag:football;tag:hockey`

Response

The list containing the detailed engagement analytics over the given time.

Data returned by a call to this method:

Property Name	Description
`media_id`	A unique identifier for the media.
`media_title`	The title of the media.
`media_duration`	The duration of the media.
`num_media_plays`	The number of times the media has been played.
`total_media_time_viewed`	The total number of played milliseconds.
`ave_media_time_viewed`	The average time the media was played.
`media_milestones`	The number of plays as a relative amount

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>
<table>
<tr><th>Property Name</th><th>Description</th></tr>
<tr><td></td><td>of the total video duration. The percentage of the video watched by a viewer (ignoring portions of the video that are 're-watched' or scrubbed back to). The following are returned: <code>25%</code>, <code>50%</code>, <code>75%</code>, <code>100%</code></td></tr>
<tr><td><code>media_segments</code></td><td>The number of viewers that watched a particular segment within a video. A video is broken into 40 segments, where each segment represents 2.5% of the total video duration. The 40 segments are used to plot the dropoff graph.</td></tr>
</table>
</td></tr>
<tr><td><b>Errors</b></td><td>
<ul>
<li>You supplied an invalid value: <code>page_id</code> must be <code>&gt;= 0</code></li>
<li>You supplied an invalid value: <code>page_size</code> must be in the range <code>[1..100]</code></li>
<li>The sort field specified was invalid. Please check your sort field and try again.</li>
<li>The sort order specified was invalid. Please check your sort order and try again.</li>
<li>Both start and end dates are required, and the start date must come before the end date.</li>
</ul>
</td></tr>
</table>
</td></tr>
</table>

API Requests - Geographic Metrics

Note: The Geographic Metrics reports are only calculated once each day based on the publisher's timezone (typically PST). You must wait until the day has been completed in your timezone for the data to appear in the daily reports.

The following methods are available for requesting **Geographic** information for an account:

Country Metrics

- Metrics by Country
- Metrics by Country Sorted by Media ID

Region Metrics

- Metrics by Region
- Metrics by Region Sorted by Media ID

City Metrics

- Metrics by City
- Metrics by Region Sorted by Media ID

DMA (Designated Marketing Area) Metrics

Metrics by DMA

Metrics by DMA Sorted by Media ID

Country Metrics

Metrics by Country

Data for metrics by country can be retrieved programmatically via the following method:

Get metrics by geography (country)	**Get a comprehensive listing of geographic metrics by country.**	
	URL	`http://api.video.limelight.com/rest/v3/organizations/<org id>/analytics/performance/country`
	Formats	`XML, JSON, CSV`
	HTTP Method	`GET`
	Requires Authentication	Yes (See Authentication - Signing Requests for Authentication Instructions.)
	Required Parameters	`start:` Date string, represented as yyyy-MM-dd. For example, 2018-03-01. See Results by Date for more details. `end:` Date string, represented as yyyy-MM-dd. For example, 2018-03-31. See Results by Date for more details.
	Paging Parameters	`page_id:` The zero-based identifier of the page to return. `page_size:` The number of results to return per page. (default/max: `100`) `sort_by:` The field by which the results should be sorted. {`plays`, `country`} (default: `plays`) `sort_order:` The order in which the results should display. {`asc`, `desc`} (default: `desc`)

	Filter Parameters	**Note:** By default, this API will return the analytics for all media. If desired, you can limit the response to media that only meets specified criteria. See Filtering Results for more information.
		`and = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. All criteria must be true for a video to be returned>`
		Example: limit response to only published media with a tag of football:
		`and=tag:football;state:published`

<table>
<tr><td></td><td>
<table>
<tr><td></td><td><code>or = &lt;filter criteria represented as <key>:<value> pairs, separated by semi-colons. At least one of the criteria must be true for a video to be returned&gt;</code>
<b>Example</b>: limit response to only media with a tag of football or a tag of hockey:
<code>or=tag:football;tag:hockey</code></td></tr>
<tr><td><b>Response</b></td><td>List of countries and performance metrics.
Data returned by a call to this method:
<table>
<tr><th>Property Name</th><th>Description</th></tr>
<tr><td><code>country_id</code></td><td>The two-digit ISO 3166-1 country code</td></tr>
<tr><td><code>country</code></td><td>The ISO 3166-1 country name</td></tr>
<tr><td><code>num_media_ country_plays</code></td><td>The number of times the media has been played in the country</td></tr>
</table></td></tr>
<tr><td><b>Errors</b></td><td>
<ul>
<li>You supplied an invalid value: <code>page_id</code> must be <code>&gt;= 0</code></li>
<li>You supplied an invalid value: <code>page_size</code> must be in the range <code>[1..100]</code></li>
<li>The sort field specified was invalid. Please check your sort field and try again.</li>
<li>The sort order specified was invalid. Please check your sort order and try again.</li>
<li>Both start and end dates are required, and the start date must come before the end date.</li>
<li>You supplied an invalid search field.</li>
</ul></td></tr>
<tr><td><b>Example Usage</b></td><td>Example code for this method can be found in Example Code.</td></tr>
</table>
</td></tr>
</table>

Metrics by Country Sorted by Media ID

You can also retrieve metrics by country sorted by media ID.

<table>
<tr><td>Get metrics by geography (country) sorted by media id</td><td><b>Get metrics sorted by media, where each media is followed by a list of countries where the media was played and the number of plays in each country.</b>
<table>
<tr><td><b>URL</b></td><td><code>http://api.video.limelight.com/rest/v1/organizations/{organizationId}/analytics/performance/country/media</code></td></tr>
<tr><td><b>Formats</b></td><td><code>XML, JSON, CSV</code></td></tr>
<tr><td><b>HTTP Method</b></td><td><code>GET</code></td></tr>
</table></td></tr>
</table>

Requires Authentication	Yes (See Authentication - Signing Requests for Authentication Instructions.)
Required Parameters	`start:` Date string, represented as yyyy-MM-dd. For example, 2018-03-01. See Results by Date for more details. `end:` Date string, represented as yyyy-MM-dd. For example, 2018-03-31. See Results by Date for more details.
Paging Parameters	`page_id:` The zero-based identifier of the page to return. `page_size:` The number of results to return per page. (default/max: `100`) `sort_by:` The field by which the results should be sorted {`title`, `plays`, `mediaId`}. (default: `plays`) `sort_order:` The order in which the results should display {`asc`, `desc`} (default: `desc`).

Filter Parameters	**Note:** By default, this API will return the analytics for all media. If desired, you can limit the response to media that only meets specified criteria. See Filtering Results for more information.
	`and = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. All criteria must be true for a video to be returned>`
	Example: limit response to only published media with a tag of football:
	`and=tag:football;state:published`
	`or = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. At least one of the criteria must be true for a video to be returned>`
	Example: limit response to only media with a tag of football or a tag of hockey:
	`or=tag:football;tag:hockey`

Response	List of countries and performance metrics for each media. Data returned by a call to this method:
Errors	• You supplied an invalid value: `page_id` must be `>= 0` • You supplied an invalid value: `page_size` must be in the range `[1..100]` • The sort field specified was invalid. Please check your sort field and try again. • The sort order specified was invalid. Please check your sort order and try again. • Both start and end dates are required, and the start date must come before the end date.

		- You supplied an invalid search field.
	Example Usage	Example code for this method can be found in Example Code.

Region Metrics

Metrics by Region

Data for metrics by region can be retrieved programmatically via the following method:

Get metrics by geography (region)	**Get a comprehensive listing of geographic metrics by region (for example, by US state)**	
	URL	`http://api.video.limelight.com/rest/v3/organizations/<org id>/analytics/performance/region`
	Formats	`XML, JSON, CSV`
	HTTP Method	`GET`
	Requires Authentication	Yes (See Authentication - Signing Requests for Authentication Instructions.)
	Required Parameters	`start`: Date string, represented as yyyy-MM-dd. For example, 2018-03-01. See Results by Date for more details. `end`: Date string, represented as yyyy-MM-dd. For example, 2018-08-31. See Results by Date for more details. `country_id`: The two-digit ISO 3166-1 country code.
	Paging Parameters (Optional)	`page_id`: The zero-based identifier of the page to return. `page_size`: The number of results to return per page. (default/max: `100`) `sort_by`: The field by which the results should be sorted {`plays`, `region`}. (default: `plays`) `sort_order`: The order in which the results should display {`asc`, `desc`}. (default: `desc`) ARC Light Custom rules executed by the Limelight Content Delivery Network (CDN) server handle user requests, make real-time modifications to speed connections to content, and offload complex data manipulations from application infrastructure.
	Filter Parameters (Optional)	**Note:** By default, this API will return the analytics for all media. If desired, you can limit the response to media that only meets specified criteria. See Filtering Results for

<table>
<tr>
<td></td>
<td>
<p>more information.</p>
<p><code>and = &lt;filter criteria represented as <key>:<value> pairs, separated by semi-colons. All criteria must be true for a video to be returned&gt;</code></p>
<p><b>Example</b>: limit response to only published media with a tag of football:</p>
<p><code>and=tag:football;state:published</code></p>
<p><code>or = &lt;filter criteria represented as <key>:<value> pairs, separated by semi-colons. At least one of the criteria must be true for a video to be returned&gt;</code></p>
<p><b>Example</b>: limit response to only media with a tag of football or a tag of hockey:</p>
<p><code>or=tag:football;tag:hockey</code></p>
</td>
</tr>
<tr>
<td><b>Response</b></td>
<td>
<p>A list of regions and performance metrics for the given country</p>
<p>Data returned by a call to this method:</p>
<table>
<tr><th>Property Name</th><th>Description</th></tr>
<tr><td><code>region_id</code></td><td>The unique ID for the region.</td></tr>
<tr><td><code>region</code></td><td>The name of the region.</td></tr>
<tr><td><code>num_media_region_plays</code></td><td>The number of times the media has been played in the region.</td></tr>
</table>
</td>
</tr>
<tr>
<td><b>Errors</b></td>
<td>
<ul>
<li>You supplied an invalid value: <code>page_id</code> must be <code>&gt;= 0</code>.</li>
<li>You supplied an invalid value: <code>page_size</code> must be in the range <code>[1..100]</code>.</li>
<li>The sort field specified was invalid. Please check your sort field and try again.</li>
<li>The sort order specified was invalid. Please check your sort order and try again.</li>
<li>Both start and end dates are required, and the start date must come before the end date.</li>
<li>You supplied an invalid search field.</li>
<li>You supplied an invalid value: <code>country_id</code> is required and must be a two-character value.</li>
</ul>
</td>
</tr>
<tr>
<td><b>Example Usage</b></td>
<td>Example code for this method can be found in Example Code.</td>
</tr>
</table>

Metrics by Region Sorted by Media ID

You can also retrieve metrics by region sorted by media ID.

Get metrics by geography region sorted by media id	**Get metrics sorted by media, where each media is followed by a list of regions (for example, US state) where the media is played and the number of plays in each region.**	
	URL	`http://api.video.limelight.com/rest/v1/organizations/<org id>/analytics/performance/region/media`
	Formats	`XML, JSON, CSV`
	HTTP Method	`GET`
	Requires Authentication	Yes (See Authentication - Signing Requests for Authentication Instructions.)
	Required Parameters	`start:` Date string, represented as yyyy-MM-dd. For example, 2018-03-01. See Results by Date for more details. `end:` Date string, represented as yyyy-MM-dd. For example, 2018-03-31. See Results by Date for more details.
	Paging Parameters	`page_id:` The zero-based identifier of the page to return. `page_size:` The number of results to return per page. (default/max: `100`) `sort_by:` The field by which the results should be sorted {`title`, `plays`, `mediaId`}. (default: `plays`) `sort_order:` The order in which the results should display {`asc`, `desc`} (default: `desc`).

	Filter Parameters	**Note**: By default, this API will return the analytics for all media. If desired, you can limit the response to media that only meets specified criteria. See Filtering Results for more information.
		`and = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. All criteria must be true for a video to be returned>`
		Example: limit response to only published media with a tag of football:
		`and=tag:football;state:published`
		`or = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. At least one of the criteria must be true for a video to be returned>`
		Example: limit response to only media with a tag of football or a tag of hockey:
		`or=tag:football;tag:hockey`

	Response	A list of media ids and titles, then a list of regions and performance metrics

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>for the given country
Data returned by a call to this method:
<table>
<tr><th>Property Name</th><th>Description</th></tr>
<tr><td><code>media_id</code></td><td>A unique identifier for the media.</td></tr>
<tr><td><code>media_title</code></td><td>The title of the media.</td></tr>
<tr><td><code>regions</code></td><td>A list of regions in which the media was played. Each region object in the list has these properties:
<code>region</code>: The name of the region.
<code>region_id</code>: A unique identifier for the region.
<code>plays</code>: The number of times the media has been played in the region.</td></tr>
</table>
</td></tr>
<tr><td><b>Errors</b></td><td>
<ul>
<li>You supplied an invalid value: <code>page_id</code> must be <code>>= 0</code></li>
<li>You supplied an invalid value: <code>page_size</code> must be in the range <code>[1..100]</code></li>
<li>The sort field specified was invalid. Please check your sort field and try again.</li>
<li>The sort order specified was invalid. Please check your sort order and try again.</li>
<li>Both start and end dates are required, and the start date must come before the end date.</li>
<li>You supplied an invalid search field.</li>
</ul>
</td></tr>
<tr><td><b>Example Usage</b></td><td>Example code for this method can be found in Example Code.</td></tr>
</table>
</td></tr>
</table>

City Metrics

Metrics by City

Data for metrics by city can be retrieved programmatically via the following method:

<table>
<tr><td>Get metrics by geography (city)</td><td><b>Get a comprehensive listing of geographic metrics by city.</b>
<table>
<tr><td><b>URL</b></td><td><code>http://api.video.limelight.com/rest/v3/organizations/<org id>/analytics/performance/city</code></td></tr>
<tr><td><b>Formats</b></td><td><code>XML, JSON, CSV</code></td></tr>
<tr><td><b>HTTP Method</b></td><td><code>GET</code></td></tr>
<tr><td><b>Requires</b></td><td>Yes (See Authentication - Signing Requests for Authentic-</td></tr>
</table>
</td></tr>
</table>

Authentication	ation Instructions.)
Required Parameters	`start`: Date string, represented as yyyy-MM-dd. For example, 2018-03-01. See Results by Date for more details. `end`: Date string, represented as yyyy-MM-dd. For example, 2018-03-31. See Results by Date for more details. `country_id`: The two-digit ISO 3166-1 country code. `region_id`: The two-digit region code (for example, **CA** for the state of California).
Paging Parameters (Optional)	`page_id`: The zero-based identifier of the page to return. `page_size`: The number of results to return per page. (default/max: `100`) `sort_by`: The field by which the results should be sorted {`plays`, `city`}. (default: `plays`) `sort_order`: The order in which the results should display {`asc`, `desc`}. (default: `desc`)

Filter Parameters (Optional)	**Note:** By default, this API will return the analytics for all media. If desired, you can limit the response to media that only meets specified criteria. See Filtering Results for more information.
	`and = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. All criteria must be true for a video to be returned>`
	Example: limit response to only published media with a tag of football:
	`and=tag:football;state:published`
	`or = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. At least one of the criteria must be true for a video to be returned>`
	Example: limit response to only media with a tag of football or a tag of hockey:
	`or=tag:football;tag:hockey`

Response	A list of cities/regions and performance metrics Data returned by a call to this method:

Property Name	Description
`city`	The name of the city.
`region`	The name of the region.
`region_id`	The unique ID for the region.
`num_media_city_plays`	The number of times the media has been played in the city.

Errors	• You supplied an invalid value: `page_id` must be `>= 0` • You supplied an invalid value: `page_size` must be in the range `[1..100]` • The sort field specified was invalid. Please check your sort field and try again. • The sort order specified was invalid. Please check your sort order and try again. • Both start and end dates are required, and the start date must come before the end date. • You supplied an invalid search field. • You supplied an invalid value: `country_id` is required and must be a two-character value. • You supplied an invalid value: `region_id` is required, must be a two-character value.
Example Usage	Example code for this method can be found in Example Code.

Metrics by City Sorted by Media ID

You can also retrieve metrics by region sorted by media ID.

Get metrics by geography (city) sorted by media id	**Get metrics sorted by media, where each media is followed by a list of cities where the media is played and the number of plays in each city.**	
	URL	`http://api.video.limelight.com/rest/v1/organizations/<org id>/analytics/performance/city/media`
	Formats	`XML, JSON, CSV`
	HTTP Method	`GET`
	Requires Authentication	Yes (See Authentication - Signing Requests for Authentication Instructions.)
	Required Parameters	`start:` Date string, represented as yyyy-MM-dd. For example, 2018-03-01. See Results by Date for more details. `end:` Date string, represented as yyyy-MM-dd. For example, 2018-03-31.

<table>
<tr><td></td><td>See Results by Date for more details.</td></tr>
<tr><td><b>Paging Parameters</b></td><td>

`page_id:` The zero-based identifier of the page to return.

`page_size:` The number of results to return per page. (default/max: 100)

`sort_by:` The field by which the results should be sorted {`title`, `plays`, `mediaId`}. (default: `plays`)

`sort_order:` The order in which the results should display {`asc`, `desc`} (default: `desc`).

</td></tr>
<tr><td><b>Filter Parameters</b></td><td>

> **Note:** By default, this API will return the analytics for all media. If desired, you can limit the response to media that only meets specified criteria. See Filtering Results for more information.

`and = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. All criteria must be true for a video to be returned>`

Example: limit response to only published media with a tag of football:

`and=tag:football;state:published`

`or = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. At least one of the criteria must be true for a video to be returned>`

Example: limit response to only media with a tag of football or a tag of hockey:

`or=tag:football;tag:hockey`

</td></tr>
<tr><td><b>Response</b></td><td>

A list of cities/regions and performance metrics.

Data returned by a call to this method:

<table>
<tr><th>Property Name</th><th>Description</th></tr>
<tr><td>`media_id`</td><td>A unique identifier for the media</td></tr>
<tr><td>`media_title`</td><td>The title of the media</td></tr>
<tr><td>`cities`</td><td>A list of cities in which the media was played. Each city object in the list has these properties:
`city`: The name of the city.
`region`: The name of the region.
`region_id`: The unique ID for the region.
`num_media_city_plays`: The number of times the media</td></tr>
</table>

</td></tr>
</table>

Property Name	Description
	has been played in the city.

Errors	• You supplied an invalid value: `page_id` must be `>= 0` • You supplied an invalid value: `page_size` must be in the range `[1..100]` • The sort field specified was invalid. Please check your sort field and try again. • The sort order specified was invalid. Please check your sort order and try again. • Both start and end dates are required, and the start date must come before the end date. • You supplied an invalid search field.
Example Usage	Example code for this method can be found in Example Code.

DMA (Designated Marketing Area) Metrics

Metrics by DMA

Data for metrics by DMA can be retrieved programmatically via the following method:

Get metrics by geography (DMA)	**Get a comprehensive listing of geographic metrics by DMA.**	
	URL	`http://api.video.limelight.com/rest/v3/organizations/<org id>/analytics/performance/dma`
	Formats	`XML, JSON, CSV`
	HTTP Method	`GET`
	Requires Authentication	Yes (See Authentication - Signing Requests for Authentication Instructions.)
	Required Parameters	`start`: Date string, represented as yyyy-MM-dd. For example, 2018-03-01. See Results by Date for more details. `end`: Date string, represented as yyyy-MM-dd. For example, 2018-03-31. See Results by Date for more details.
	Paging Parameters (Optional)	`page_id`: The zero-based identifier of the page to return `page_size`: The number of results to return per page (default/max: `100`) `sort_by`: The field by which the results should be sorted {`plays`, `DMA`} (default: `plays`) `sort_order`: The order in which the results should display

<table>
<tr><td></td><td>{asc, desc} (default: desc)</td></tr>
<tr><td><b>Filter Parameters (Optional)</b></td><td>
<b>Note:</b> By default, this API will return the analytics for all media. If desired, you can limit the response to media that only meets specified criteria. See Filtering Results for more information.
<code>and = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. All criteria must be true for a video to be returned></code>
<b>Example</b>: limit response to only published media with a tag of football:
<code>and=tag:football;state:published</code>
<code>or = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. At least one of the criteria must be true for a video to be returned></code>
<b>Example</b>: limit response to only media with a tag of football or a tag of hockey:
<code>or=tag:football;tag:hockey</code>
</td></tr>
<tr><td><b>Response</b></td><td>
A list of DMAs and performance metrics
Data returned by a call to this method:
<table>
<tr><th>Property Name</th><th>Description</th></tr>
<tr><td><code>dma_id</code></td><td>A unique identifier for the DMA.</td></tr>
<tr><td><code>dma</code></td><td>The title of the DMA.</td></tr>
<tr><td><code>num_media_dma_plays</code></td><td>The number of times the media has been in the DMA.</td></tr>
</table>
</td></tr>
<tr><td><b>Errors</b></td><td>
<ul>
<li>You supplied an invalid value: <code>page_id</code> must be <code>&gt;= 0</code></li>
<li>You supplied an invalid value: <code>page_size</code> must be in the range <code>[1..100]</code></li>
<li>The sort field specified was invalid. Please check your sort field and try again.</li>
<li>The sort order specified was invalid. Please check your sort order and try again.</li>
<li>Both start and end dates are required, and the start date must come before the end date.</li>
<li>You supplied an invalid search field.</li>
</ul>
</td></tr>
</table>

<table>
<tr><td></td><td><table><tr><td><b>Example Usage</b></td><td>Example code for this method can be found in Example Code.</td></tr></table></td></tr>
</table>

Metrics by DMA Sorted by Media ID

You can also retrieve metrics by DMA sorted by media id:

<table>
<tr>
<td>Get metrics by geography (DMA) sorted by media id</td>
<td><b>Get metrics sorted by media, where each media is followed by a list of DMAs where the media is played and the number of plays in each DMA.</b>
<table>
<tr><td><b>URL</b></td><td><code>http://api.video.limelight.com/rest/v1/organizations/<org id>/analytics/performance/dma/media</code></td></tr>
<tr><td><b>Formats</b></td><td><code>XML, JSON, CSV</code></td></tr>
<tr><td><b>HTTP Method</b></td><td><code>GET</code></td></tr>
<tr><td><b>Requires Authentication</b></td><td>Yes (See Authentication - Signing Requests for Authentication Instructions.)</td></tr>
<tr><td><b>Required Parameters</b></td><td><code>start:</code> Date string, represented as yyyy-MM-dd. For example, 2018-03-01. See Results by Date for more details.
<code>end:</code> Date string, represented as yyyy-MM-dd. For example, 2018-03-31. See Results by Date for more details.</td></tr>
<tr><td><b>Paging Parameters</b></td><td><code>page_id:</code> The zero-based identifier of the page to return.
<code>page_size:</code> The number of results to return per page. (default/max: <code>100</code>)
<code>sort_by:</code> The field by which the results should be sorted {<code>title</code>, <code>plays</code>, <code>mediaId</code>}. (default: <code>plays</code>)
<code>sort_order:</code> The order in which the results should display {<code>asc</code>, <code>desc</code>}. (default: <code>desc</code>)</td></tr>
<tr><td><b>Filter Parameters</b></td><td><b>Note:</b> By default, this API will return the analytics for all media. If desired, you can limit the response to media that only meets specified criteria. See Filtering Results for more information.
<code>and = &lt;filter criteria represented as <key>:<value> pairs, separated by semi-colons. All criteria must be true for a video to be returned&gt;</code>
<b>Example</b>: limit response to only published media with a tag of football:
<code>and=tag:football;state:published</code>
<code>or = &lt;filter criteria represented as <key>:<value> pairs, separated by semi-colons. At least one of the criteria must be true for a video to be returned&gt;</code></td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><td></td><td><strong>Example</strong>: limit response to only media with a tag of football or a tag of hockey:
<code>or=tag:football;tag:hockey</code></td></tr>
<tr><td><strong>Response</strong></td><td>A list of DMAs and performance metrics
Data returned by a call to this method:
<table>
<tr><th>Property Name</th><th>Description</th></tr>
<tr><td><code>media_id</code></td><td>A unique identifier for the media.</td></tr>
<tr><td><code>media_title</code></td><td>The title of the media.</td></tr>
<tr><td><code>dmas</code></td><td>A list of DMAs in which the media was played. Each DMA object in the list has these properties:
<code>dma_id</code>: A unique identifier for the DMA.
<code>dma</code>: The title of the DMA.
<code>num_media_dma_plays</code>: The number of times the media has been played in the DMA.</td></tr>
</table></td></tr>
<tr><td><strong>Errors</strong></td><td>• You supplied an invalid value: <code>page_id</code> must be <code>&gt;= 0</code>
• You supplied an invalid value: <code>page_size</code> must be in the range <code>[1..100]</code>
• The sort field specified was invalid. Please check your sort field and try again.
• The sort order specified was invalid. Please check your sort order and try again.
• Both start and end dates are required, and the start date must come before the end date.
• You supplied an invalid search field.</td></tr>
<tr><td><strong>Example Usage</strong></td><td>Example code for this method can be found in Example Code.</td></tr>
</table>

API Requests - Platform Metrics

> **Note:** The Platform Metrics reports are only calculated once each day based on the publisher's timezone (typically PST). You must wait until the day has been completed in your timezone for the data to appear in the daily reports.

The following methods are available for requesting **Platform** information for an account:

Operating System Metrics

Metrics by Platform (OS)

Metrics by Platform (OS) Sorted by Media ID

Operating System Metrics

Metrics by Platform (OS)

Data for metrics by platform (OS) can be retrieved programmatically via the following method:

<table>
<tr>
<td>Get metrics by platform (OS)</td>
<td><b>Get a listing of operating systems that have been used to play the media.</b>
<table>
<tr><td><b>URL</b></td><td><code>http://api.video.limelight.com/rest/v3/organizations/<org id>/analytics/performance/os</code></td></tr>
<tr><td><b>Formats</b></td><td><code>XML, JSON, CSV</code></td></tr>
<tr><td><b>HTTP Method</b></td><td><code>GET</code></td></tr>
<tr><td><b>Requires Authentication</b></td><td>Yes (See Authentication - Signing Requests for Authentication Instructions.)</td></tr>
<tr><td><b>Required Parameters</b></td><td><code>start</code>: Date string, represented as yyyy-MM-dd. For example, 2018-03-01. See Results by Date for more details.
<code>end</code>: Date string, represented as yyyy-MM-dd. For example, 2018-03-31. See Results by Date for more details.</td></tr>
<tr><td><b>Paging Parameters (Optional)</b></td><td><code>page_id</code>: The zero-based identifier of the page to return.
<code>page_size</code>: The number of results to return per page. (default/max: <code>100</code>)
<code>sort_by</code>: The field by which the results should be sorted {<code>plays</code>, <code>operatingSystem</code>}. (default: <code>plays</code>)
<code>sort_order</code>: The order in which the results should display {<code>asc</code>, <code>desc</code>} (default: <code>desc</code>).</td></tr>
<tr><td><b>Filter Parameters (Optional)</b></td><td><b>Note:</b> By default, this API will return the analytics for all media. If desired, you can limit the response to media that only meets specified criteria. See Filtering Results for more information.
<code>and = &lt;filter criteria represented as <key>:<value> pairs, separated by semi-colons.</code></td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>
<table>
<tr>
<td></td>
<td>
`All criteria must be true for a video to be returned>`
Example: limit response to only published media with a tag of football:
`and=tag:football;state:published`
`or = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. At least one of the criteria must be true for a video to be returned>`
Example: limit response to only media with a tag of football or a tag of hockey:
`or=tag:football;tag:hockey`
</td>
</tr>
<tr>
<td>**Response**</td>
<td>
A list of operating systems and performance metrics.
Data returned by a call to this method:
<table>
<tr><th>Property Name</th><th>Description</th></tr>
<tr><td>`media_os`</td><td>The name of the OS.</td></tr>
<tr><td>`num_media_os_plays`</td><td>The number of times the media has been played via the OS.</td></tr>
</table>
</td>
</tr>
<tr>
<td>**Errors**</td>
<td>
- You supplied an invalid value: `page_id` must be `>= 0`
- You supplied an invalid value: `page_size` must be in the range `[1..100]`
- The sort field specified was invalid. Please check your sort field and try again.
- The sort order specified was invalid. Please check your sort order and try again.
- Both start and end dates are required, and the start date must come before the end date.
- You supplied an invalid search field.
</td>
</tr>
</table>
</td>
</tr>
</table>

Metrics by Platform (OS) Sorted by Media ID

You can also retrieve metrics by platform (OS) sorted by media ID.

<table>
<tr>
<td>Get metrics by platform (OS) sorted by media id</td>
<td>
Get a listing of operating systems that have been used to play the media and the media to which they are associated.
<table>
<tr><td>**URL**</td><td>`http://api.video.limelight.com/rest/v1/organizations/<org id>/analytics/performance/os/media`</td></tr>
<tr><td>**Formats**</td><td>`XML, JSON, CSV`</td></tr>
</table>
</td>
</tr>
</table>

HTTP Method	`GET`
Requires Authentication	Yes (See Authentication - Signing Requests for Authentication Instructions.)
Required Parameters	`start:` Date string, represented as yyyy-MM-dd. For example, 2018-03-01. See Results by Date for more details. `end:` Date string, represented as yyyy-MM-dd. For example, 2018-03-31. See Results by Date for more details.
Paging Parameters	`page_id:` The zero-based identifier of the page to return. `page_size:` The number of results to return per page. (default/max: `100`) `sort_by:` The field by which the results should be sorted {`plays`, `title`, `mediaId`}. (default: `plays`) `sort_order:` The order in which the results should display {`asc`, `desc`}. (default: `desc`)

Filter Parameters	**Note:** By default, this API will return the analytics for all media. If desired, you can limit the response to media that only meets specified criteria. See Filtering Results for more information.
	`and = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. All criteria must be true for a video to be returned>`
	Example: limit response to only published media with a tag of football:
	`and=tag:football;state:published`
	`or = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. At least one of the criteria must be true for a video to be returned>`
	Example: limit response to only media with a tag of football or a tag of hockey:
	`or=tag:football;tag:hockey`

Response	A list of operating systems and performance metrics for each media. Data returned by a call to this method:

Property Name	Description
`media_id`	A unique identifier for the media.
`media_title`	The title of the media.

Property Name	Description
`operating_systems`	A list of operating systems on which the media was played. Each operating system object in the list has these properties: `media_os`: The name of the OS. `num_media_os_plays`: The number of times the media has been played via the browser.

Errors	- You supplied an invalid value: `page_id` must be `>= 0` - You supplied an invalid value: `page_size` must be in the range `[1..100]` - The sort field specified was invalid. Please check your sort field and try again. - The sort order specified was invalid. Please check your sort order and try again. - Both start and end dates are required, and the start date must come before the end date. - You supplied an invalid search field.
Example Usage	Example code for this method can be found in Example Code.

Browser Metrics

Metrics by Platform (Browser)

Data for metrics by platform (browser) can be retrieved programmatically via the following method:

Get metrics by platform (Browser)	**Get a listing of browsers that have been used to play the media.**
URL	`http://api.video.limelight.com/rest/v3/organizations/<org id>/analytics/performance/browser`
Formats	`XML, JSON, CSV`
HTTP Method	`GET`
Requires Authentication	Yes (See Authentication - Signing Requests for Authentication Instructions.)
Required Parameters	`start`: Date string, represented as yyyy-MM-dd. For example, 2018-03-01. See Results by Date for more details. `end`: Date string, represented as yyyy-MM-dd. For example, 2018-03-31. See Results by Date for more details.
Paging Parameters (Optional)	`page_id`: The zero-based identifier of the page to return. `page_size`: The number of results to return per page. (default/max: `100`)

<table>
<tr><td></td><td>

`sort_by`: The field by which the results should be sorted {`plays`, `browser`}. (default: `plays`)

`sort_order`: The order in which the results should display {`asc`, `desc`}. (default: `desc`)

</td></tr>
<tr><td>

Filter Parameters (Optional)

</td><td>

Note: By default, this API will return the analytics for all media. If desired, you can limit the response to media that only meets specified criteria. See Filtering Results for more information.

`and = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. All criteria must be true for a video to be returned>`

Example: limit response to only published media with a tag of football:

`and=tag:football;state:published`

`or = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. At least one of the criteria must be true for a video to be returned>`

Example: limit response to only media with a tag of football or a tag of hockey:

`or=tag:football;tag:hockey`

</td></tr>
<tr><td>

Response

</td><td>

A list of browsers and performance metrics.

Data returned by a call to this method:

<table>
<tr><th>Property Name</th><th>Description</th></tr>
<tr><td><code>media_browser</code></td><td>The name of the browser.</td></tr>
<tr><td><code>num_media_browser_plays</code></td><td>The number of times the media has been played via the browser.</td></tr>
</table>

</td></tr>
<tr><td>

Errors

</td><td>

- You supplied an invalid value: `page_id` must be `>= 0`
- You supplied an invalid value: `page_size` must be in the range `[1..100]`
- The sort field specified was invalid. Please check your sort field and try again.
- The sort order specified was invalid. Please check your sort order and try again.
- Both start and end dates are required, and the start date must come before the end date.

</td></tr>
</table>

		• You supplied an invalid search field.
	Example Usage	Example code for this method can be found in Example Code.

Metrics by Platform (Browser) Sorted by Media ID

You can also retrieve metrics by platform (browser) sorted by media ID.

Get metrics by platform (Browser) sorted by media id	**Get a listing of browsers that have been used to play the media for the media they're associated with.**	
	URL	`http://api.video.limelight.com/rest/v1/organizations/<org id>/analytics/performance/browser/media`
	Formats	`XML, JSON, CSV`
	HTTP Method	`GET`
	Requires Authentication	Yes (See Authentication - Signing Requests for Authentication Instructions.)
	Required Parameters	`start:` Date string, represented as yyyy-MM-dd. For example, 2018-03-01. See Results by Date for more details. `end:` Date string, represented as yyyy-MM-dd. For example, 2018-03-31. See Results by Date for more details.
	Paging Parameters	`page_id:` The zero-based identifier of the page to return. `page_size:` The number of results to return per page. (default/max: `100`) `sort_by:` The field by which the results should be sorted {`plays`, `title`, `mediaId`}. (default: `plays`) `sort_order:` The order in which the results should display {`asc`, `desc`}. (default: `desc`)

	Filter Parameters	**Note:** By default, this API will return the analytics for all media. If desired, you can limit the response to media that only meets specified criteria. See Filtering Results for more information.
		`and = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. All criteria must be true for a video to be returned>`
		Example: limit response to only published media with a tag of football:
		`and=tag:football;state:published`
		`or = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. At least one of the criteria must`

<table>
<tr><td></td><td>
<table>
<tr><td></td><td><code>be true for a video to be returned></code>
<b>Example</b>: limit response to only media with a tag of football or a tag of hockey:
<code>or=tag:football;tag:hockey</code></td></tr>
<tr><td><b>Response</b></td><td>A list of the browsers and performance metrics
Data returned by a call to this method:
<table>
<tr><th>Property Name</th><th>Description</th></tr>
<tr><td><code>media_id</code></td><td>A unique identifier for the media.</td></tr>
<tr><td><code>media_title</code></td><td>The title of the media.</td></tr>
<tr><td><code>browsers</code></td><td>A list of browsers the media was played with. Each browser object in the list has these properties:
<code>media_browser</code>: The name of the browser.
<code>num_media_browser_plays</code>: The number of times the media has been played via the browser.</td></tr>
</table>
</td></tr>
<tr><td><b>Errors</b></td><td>
<ul>
<li>You supplied an invalid value: <code>page_id</code> must be <code>>= 0</code></li>
<li>You supplied an invalid value: <code>page_size</code> must be in the range <code>[1..100]</code></li>
<li>The sort field specified was invalid. Please check your sort field and try again.</li>
<li>The sort order specified was invalid. Please check your sort order and try again.</li>
<li>Both start and end dates are required, and the start date must come before the end date.</li>
<li>You supplied an invalid search field.</li>
</ul>
</td></tr>
<tr><td><b>Example Usage</b></td><td>Example code for this method can be found in Example Code.</td></tr>
</table>
</td></tr>
</table>

Platform Metrics

Metrics by Platform

Data for metrics by platform can be retrieved programmatically via the following method:

<table>
<tr><td>Get metrics by platform</td><td><b>Get a listing of platforms that have been used to play the media (for example, HTML5, and so forth).</b>
<table>
<tr><td><b>URL</b></td><td><code>http://api.video.limelight.com/rest/v3/organizations/<org id>/analytics/performance/platform</code></td></tr>
<tr><td><b>Formats</b></td><td><code>XML, JSON, CSV</code></td></tr>
</table>
</td></tr>
</table>

HTTP Method	`GET`
Requires Authentication	Yes (See Authentication - Signing Requests for Authentication Instructions.)
Required Parameters	`start`: Date string, represented as yyyy-MM-dd. For example, 2018-03-01. See Results by Date for more details. `end`: Date string, represented as yyyy-MM-dd. For example, 2018-03-31. See Results by Date for more details.
Paging Parameters (Optional)	`page_id`: The zero-based identifier of the page to return. `page_size`: The number of results to return per page. (default/max: `100`) `sort_by`: The field by which the results should be sorted {`plays`, `platform`}. (default: `plays`) `sort_order`: The order in which the results should display {`asc`, `desc`}. (default: `desc`)

Filter Parameters (Optional)	**Note:** By default, this API will return the analytics for all media. If desired, you can limit the response to media that only meets specified criteria. See Filtering Results for more information.
	`and = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. All criteria must be true for a video to be returned>`
	Example: limit response to only published media with a tag of football:
	`and=tag:football;state:published`
	`or = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. At least one of the criteria must be true for a video to be returned>`
	Example: limit response to only media with a tag of football or a tag of hockey:
	`or=tag:football;tag:hockey`

Response	A list of the platforms (for example, HTML) and performance metrics Data returned by a call to this method:

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>
<table>
<tr><th>Property Name</th><th>Description</th></tr>
<tr><td><code>media_platform</code></td><td>The name of the platform.
<strong>Note:</strong> <code>HTML</code> and <code>Video</code> represent media played via <code>HTML5</code>, while <code>Link</code> represents media played via <code>3GP</code>.</td></tr>
<tr><td><code>num_media_platform_plays</code></td><td>The number of times the media has been played via the platform.</td></tr>
</table>
</td></tr>
<tr><td><strong>Errors</strong></td><td>• You supplied an invalid value: <code>page_id</code> must be <code>>= 0</code>
• You supplied an invalid value: <code>page_size</code> must be in the range <code>[1..100]</code>
• The sort field specified was invalid. Please check your sort field and try again.
• The sort order specified was invalid. Please check your sort order and try again.
• Both start and end dates are required, and the start date must come before the end date.
• You supplied an invalid search field.</td></tr>
<tr><td><strong>Example Usage</strong></td><td>Example code for this method can be found in Example Code.</td></tr>
</table>
</td></tr>
</table>

Metrics by Platform Sorted by Media ID

You can also retrieve metrics by platform sorted by media ID.

<table>
<tr><td>Get metrics by platform sorted by media id</td><td>
<strong>Get a listing of platforms that have been used to play the media (for example, HTML5, and so forth) for the media to which they are associated.</strong>
<table>
<tr><td><strong>URL</strong></td><td><code>http://api.video.limelight.com/rest/v1/organizations/<org id>/analytics/performance/platform/media</code></td></tr>
<tr><td><strong>Formats</strong></td><td><code>XML, JSON, CSV</code></td></tr>
<tr><td><strong>HTTP Method</strong></td><td><code>GET</code></td></tr>
<tr><td><strong>Requires Authentication</strong></td><td>Yes (See Authentication - Signing Requests for Authentication Instructions.)</td></tr>
<tr><td><strong>Required Parameters</strong></td><td><code>start:</code> Date string, represented as yyyy-MM-dd. For example, 2018-03-01. See Results by Date for more details.
<code>end:</code> Date string, represented as yyyy-MM-dd. For example, 2018-03-31. See Results by Date for more details.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td><b>Paging Parameters</b></td><td>page_id: The zero-based identifier of the page to return.
page_size: The number of results to return per page. (default/max: 100)
sort_by: The field by which the results should be sorted {plays, title, mediaId}. (default: plays)
sort_order: The order in which the results should display {asc, desc}. (default: desc)</td></tr>
<tr><td><b>Filter Parameters</b></td><td><b>Note:</b> By default, this API will return the analytics for all media. If desired, you can limit the response to media that only meets specified criteria. See Filtering Results for more information.
and = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. All criteria must be true for a video to be returned>
<b>Example</b>: limit response to only published media with a tag of football:
and=tag:football;state:published
or = <filter criteria represented as <key>:<value> pairs, separated by semi-colons. At least one of the criteria must be true for a video to be returned>
<b>Example</b>: limit response to only media with a tag of football or a tag of hockey:
or=tag:football;tag:hockey</td></tr>
<tr><td><b>Response</b></td><td>A list of the platforms (for example, HTML) and performance metrics.
Data returned by a call to this method:
<table>
<tr><th>Property Name</th><th>Description</th></tr>
<tr><td>media_id</td><td>A unique identifier for the media.</td></tr>
<tr><td>media_title</td><td>The title of the media.</td></tr>
<tr><td>platforms</td><td>A list of platforms on which the media was played. Each platform object in the list has these properties:
media_platform: The name of the platform: HTML and Video represent media played via HTML5; Link represents media played via 3GP.
num_media_platform_plays: The number of times the media has been played via the platform.</td></tr>
</table></td></tr>
<tr><td><b>Errors</b></td><td>• You supplied an invalid value: page_id must be >= 0</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>- You supplied an invalid value: page_size must be in the range [1..100]
- The sort field specified was invalid. Please check your sort field and try again.
- The sort order specified was invalid. Please check your sort order and try again.
- Both start and end dates are required, and the start date must come before the end date.
- You supplied an invalid search field.</td></tr>
<tr><td><b>Example Usage</b></td><td>Example code for this method can be found in Example Code.</td></tr>
</table>
</td></tr>
</table>

API Requests - Usage Metrics

The following method is available for requesting **Usage** information (bandwidth and storage) for an account:

Usage Metrics (Bandwidth and Storage)

Usage Metrics (Bandwidth and Storage)

Data for usage metrics (bandwidth and storage) can be retrieved programmatically via the following method:

<table>
<tr><td>Get usage information for an account</td><td><b>Get bandwidth and storage usage data.</b>
<table>
<tr><td><b>URL</b></td><td>http://api.video.limelight.com/rest/v2/organizations/<org id>/analytics/usage</td></tr>
<tr><td><b>Formats</b></td><td>XML, JSON, CSV</td></tr>
<tr><td><b>HTTP Method</b></td><td>GET</td></tr>
<tr><td><b>Requires Authentication</b></td><td>Yes (See Authentication - Signing Requests for Authentication Instructions.)</td></tr>
<tr><td><b>Required Parameters</b></td><td>start: Start date represented as epoch time. Example: 1521741120.
end: End date represented epoch time. Example: 1527011520.</td></tr>
<tr><td><b>Optional Parameters</b></td><td>Paging Parameters
page_id: The zero-based identifier of the page to return.
page_size: The number of results to return per page. (default/max: 100)
Return Data Granularity Parameters
interval: The granularity of the data to be returned.
{Daily, Monthly, Yearly}
default: Yearly</td></tr>
<tr><td><b>Response</b></td><td>Storage and bandwidth statistics for the specified time,</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>aggregated over the interval.
Data returned by a call to this method:
<table>
<tr><th>Property Name</th><th>Description</th></tr>
<tr><td>`start`</td><td>Unix time, represented as the number of seconds since the Unix epoch at which the time starts.</td></tr>
<tr><td>`end`</td><td>Unix time, represented as the number of seconds since the Unix epoch at which the time ends.
Note:
The time zone for both `start` and `end` is UTC.</td></tr>
<tr><td>`storage_in_megabytes`</td><td>Storage used (in megabytes).</td></tr>
<tr><td>`bandwidth_in_megabytes`</td><td>Bandwidth used (in megabytes).</td></tr>
</table>
</td></tr>
<tr><td>**Errors**</td><td>
- You supplied an invalid value: `page_id` must be `>= 0`
- You supplied an invalid value: `page_size` must be in the range `[1..100]`
- The sort field specified was invalid. Please check your sort field and try again.
- The sort order specified was invalid. Please check your sort order and try again.
- Both start and end dates are required, and the start date must come before the end date.
- Invalid interval.
</td></tr>
<tr><td>**Example Usage**</td><td>Example code for this method can be found in Example Code.</td></tr>
</table>
</td></tr>
</table>

Publisher Endpoints

Publisher Timezone

Data for retrieving a publisher's timezone can be retrieved programmatically via the following method:

Fetch publisher's timezone	**Fetch the time zone of a publisher represented by the organization ID that is passed.** The publisher

URL	`http://api.video.limelight.com/rest/organizations/<org id>/publisher/timezone`
Formats	`None`
HTTP Method	`GET`
Requires Authentication	Yes (See Authentication - Signing Requests for Authentication Instructions.)
Required Parameters	None
Response	A string representation of the organization's timezone. Example: `America/Los_Angeles`
Errors	- Missing signature.

Authentication - Signing Requests

To ensure the security of your information, we require some requests to be authenticated. All methods indicated above as requiring authentication must have a signature. A signature is generated by doing an HMAC 264 hash on the request URL and its parameters. To do this, you must obtain your unique **access_key** and **secret**. The `access_key` is passed as a normal parameter to the URL. The secret is used to generate a signature by performing a hash. Both the `access_key` and secret can be found in your Limelight Video Delivery account under the **Developer Tools** tab in **Settings**. Lastly, you need to determine a value for the **expires** parameter. This value, represented in Unix Time, indicates when the URL should no longer return results. A handy Unix Time converter can be found here: http://www.epochconverter.com To illustrate the steps involved in generating a signature, we provide an example below. In this example, we authenticate the URL used to **Get performance metrics for media**.

We will use the following variables for this example:

Org ID:	`bfb3caa8e6204fea9a23ce855768fc93`
Access Key:	`esU4jGTelX8UAsfaiWUcjWsn1mY=`
Secret:	`+IEbSWOctQhnMi7DGop9KdPysqw=`
Expires:	`1352355380`
Start:	`1348102704`
End:	`1348189104`

1. Start with the base URL for your request. The base URL for **Get performance metrics for media** looks like this:

   ```
   http://api.video.limelight.com/rest/v2/organizations/bfb3caa8e6204fea9a23ce855768fc93/
   analytics/performance/media.json
   ```

2. Drop the `http://` prefix and append the http method (`get`, `post`, `put`, `delete`) to the front of the base URL and surround the host of the URL and the path of the URL with the `|` symbol. Make sure everything is lower case. Our URL string now looks like this (notice there are three places where the `|` symbol has been inserted):

   ```
   get|api.video.limelight.com|/rest/v2/organizations/bfb3caa8e6204fea9a23ce855768fc93/
   analytics/performance/media.json|
   ```

3. Next, sort the list of parameters alphabetically and append them as <key>=<value> pairs to the end of the URL string. Separate all parameters with the `&` symbol. Our URL string now looks like this:

   ```
   get|api.video.limelight.com|/rest/v2/organizations/bfb3caa8e6204fea9a23ce855768fc93/
   analytics/performance/media.json|access_key=esU4jGTelX8UAsfaiWUcjWsn1mY=&end=
   1348189104&expires=1352355380&start=1348102704
   ```

4. Now, using your secret, generate a sha256 HMAC hash in base64 format on the URL string. The result of this hash will be your signature. In this example, our signature looks like this:

   ```
   2jH0F/o94qiqtb6Cg0k9s9eJxcrvOCcQNYtI8zqeK3I=
   ```

5. Lastly, create the final executable URL by beginning with the base URL. Append all the parameters in alphabetic order, starting with `?` and separated by `&`. URL encode all key and values on your parameters. Finally, URL encode the signature you generated above and append it at the end as signature=<signature>. The final executable URL looks like this (reformatted for readability):

   ```
   http://api.video.limelight.com/rest/organizations/bfb3caa8e6204fea9a23ce855768fc93/
   analytics/performance/media.json?
     access_key=esU4jGTelX8UAsfaiWUcjWsn1mY%3D&end=1348189104&
     expires=1352355380&start=1348102704&
     signature=2jH0F%2Fo94qiqtb6Cg0k9s9eJxcrvOCcQNYtI8zqeK3I%3D
   ```

Code to Perform Authentication

Limelight offers the following code to aid in authenticating requests. Simply drop this into your project and call the authentication function with the right parameters. The result will be an authenticated request:

PHP	**LvpAuthUtil.php**
Ruby	**LvpAuthUtil.rb**
Java	**LvpAuthUtil.java**
C#	**LvpAuthUtil.cs**
VB	**LvpAuthUtil.vb**
Python	**LvpAuthUtil.py**

Example Code

The examples below illustrate the common uses of the Analytics API. All examples make use of the following code to authenticate requests: **LvpAuthUtil.php**

(Example 1) Get the Top Ten Most Played Media

Description	A list of the top ten most-played media can simply be obtained by getting the analytics for all media, limiting the page size to 10, and sorting by popularity (in other words, Plays).
Code Sample	**PHP** api_get_top_ten.txt

(Example 2) Get the Analytics for all Media

Description	Similar to the example above, simply get the analytics for all media without specifying any optional paging parameters.
Code Sample	**PHP** api_get_analytics_all.txt

(Example 3) Evaluate the Performance of a Particular Sub-set of Media

Description	We want to see only the analytics for `published` media that has a tag of `football` or `hockey`. To do this, we limit the result set by applying the optional filtering parameters `and` & `or`, and building criteria using the elements outlined in Filter Parameters. For this specific example, we pass the following when making the API call (see params below). `and=state:published` `or=tag:football;tag:hockey`
Code Sample	**PHP** api_get_analytics_w_filter.txt

(Example 4) Find Out Which Media in a Particular Channel is the Most Popular

Description	We want to find out which media is the most popular in a particular channel. Similar to the example above, we simply use the filtering parameters to limit the result set. Specifically, we get a list of media by popularity (in other words, order by `Plays` in descending order) and

	limit the result set to a channel by passing in the channel ID as an `and` parameter. Like this: `and=channel_id:f608e642dc824303b023e6519c15afc6`
Code Sample	**PHP** api_get_analytics_limit_channel.txt

(Example 5) Find out the Top Destinations (Domains) Where Media is Played

Description	We want to find out the sites where media is being played. To do this, we simply list the domains by making a call to the domain API.
Code Sample	**PHP** api_get_analytics_domain.txt

(Example 6) List Usage Information (Bandwidth/Storage)

Description	Bandwidth and Storage information for an account can be obtained by making a call to the Usage API.
Code Sample	**PHP** api_get_analytics_usage.txt

Example `GET` Request Response

API request for `Get full performance analytics for media` (prior to authentication):

```
http://api.video.limelight.com/rest/organizations/{org_id}/analytics/performance/
media.xml
```

Returned Results:

```
<media-list has_next="true" data_as_of="1352268880" page_id="0" type="array">
  <media>
  <id>8aa8cb15c3ba459fb0900088a457d5c0</id>
  <media_type>Video</media_type>
    <title>Demo of the Video Platform</title>
    <copy_embeds>113</copy_embeds>
    <emails>6</emails>
    <distinct_viewers>7527</distinct_viewers>
    <distinct_referrers>208</distinct_referrers>
    <plays>8466</plays>
    <replays>231</replays>
    <total_time_viewed_in_milliseconds>873805010</total_time_viewed_in_milliseconds>
  </media>
    <media>
    <id>bdcc8e9987cf4167bdc3aec20a1bbdac</id>
    <media_type>Video</media_type>
    <title>How to Upload Media</title>
    <copy_embeds>14</copy_embeds>
    <emails>0</emails>
    <distinct_viewers>1641</distinct_viewers>
    <distinct_referrers>322</distinct_referrers>
    <plays>2367</plays>
    <replays>73</replays>
    <total_time_viewed_in_milliseconds>145450700</total_time_viewed_in_milliseconds>
  </media>
</media-list>
```

File Swap Caching

Information courtesy of Yahoo, http://developer.yahoo.com/php/howto-cacheRestPhp.html

File swap caching is a simple flat-file caching technique. In essence, it does nothing more than store your Web service requests in local files. When a file becomes too old (when the cache becomes "stale"), the file is deleted and replaced by a new Web service request. To make a Web services request using the cache, you call the caching function instead of making a direct request.

```
$response = request_cache($request, $cache_fullpath,$cache_timeout);
```

The three parameters are the actual request URL, the path to the cached version of the request, and the number of seconds before the cache becomes stale:

```
$request =
'http://api.video.limelight.com/rest/organizations/7fd6def47cde
4d5694f9b16bfa04c521/media.xml';
$cache_fullpath = '/mydir/ImageMad1';
$cache_timeout = 7200;
```

Here, we arbitrarily chose the name ImageMad1 as the cache file name corresponding to this particular Web services request. Typically, you would have a unique file name corresponding to each variation of Web services request. You would need to develop a mapping function to map requests into file names if you want to use the caching with different Web services requests. The function `request_cache` does all the real work. First, it checks whether the request has already been cached or if it has become stale.

```
function request_cache($url, $dest_file, $timeout)
{
  if (!file_exists($dest_file) || filemtime($dest_file) < (time()-$timeout))
  {
```

If the request has not been cached or it is stale, then we make the request and cache it:

```

      $data = file_get_contents($url);
      $tmpf = tempnam('/tmp','YWS');
      $fp = fopen($tmpf,"w");
      fwrite($fp, $data); fclose($fp);
      rename($tmpf, $dest_file);
```

Otherwise, we return the contents of the cached request:

```
  } else {
    return file_get_contents($dest_file);
    }
```

A potential problem arises if more than one user of your Web application accesses a cached file just as it is being deleted and refreshed. The statement rename(`$tmpf, $dest_file`) appears to destroy the currently cached version of the file (if any). However, the Unix filesystem implementation guarantees that any process currently reading the previous cache file will continue to read it until it closes the file. So, no process will be cut off or otherwise damaged by the rename. See the following code sample:
`http://developer.yahoo.com/php/samples/cache/cacheSWAP.txt`