**Al Jazeera VOD Application**

API Guide

Ver 1, October 30, 2013 by Brian Franklin

# Overview

The Al Jazeera VOD is a complex system consisting of a front-end UI, a back-end web application, and a number of third party APIs. In order to standardize the development of key features on the front-end, a series of VOD APIs were created as part of the VOD back-end system. This document details the VOD APIs, what they are, and how they work. In the examples in this document, code will reference the API path relative to the server root that the VOD is running on.

**Table of Contents**

[/api/login](#h.bisey0uq8ep5)

[/api/preferences/get](#h.q79hvhlmvlmk)

[/api/preferences/set](#h.g2k4gcxphwwi)

[/api/playlists/get](#h.oy7iylu01lyg)

[/api/playlists/create](#h.dog3gerexoj6)

[/api/playlists/add](#h.3nxuleng8lmt)

[/api/playlists/remove](#h.nyc5pc1w9v22)

[/api/playlists/delete](#h.huxmwb3p8u1z)

[/api/playlists/rename](#h.pkc9xaxkyjhp)

[/api/playlists/reorder](#h.n5aen0p9jrm8)

[/api/subscriptions/get](#h.771drxqns330)

[/api/subscriptions/add](#h.i5dxb4irv3fz)

[/api/subscriptions/remove](#h.86uizvkgwt5c)

[/api/subscriptions/order](#h.vk0n0nbab724)

[/api/positions/get](#h.liq7ccdgsv0j)

[/api/positions/set](#h.j3sxodtqtz7)

[/api/positions/remove](#h.82ajw99v8zcy)

[/api/events/timeline](#h.cb4zewfsvlxj)

[/api/mediaquery](#h.u2j2i392euhl)

[/api/epg](#h.xmtqcy7eiymz)

# API Access

The API methods supported by the VOD exist off of the /api/ URL route. The URL segments directly after the "api" segment represent the API's method name. Additional arguments for each method are expected as GET or POST parameters,

Many of the API methods operate on user-specific data, and as such they require a valid user session to function. If an authenticated API method is called without a valid user session, an error will be returned. Additionally, cross-origin requests are disallowed by browsers per security sandboxing. This means attempts to call a VOD API from a non-VOD domain will result in a browser security error. The domain c.brightcove.com has been explicitly granted permission to access VOD API URLs via CORS headers. This is important in ensuring that Brightcove player plugins and enhancements (including future custom player work) are able to interface with the APIs correctly.

Requests to the API from Javascript from non-VOD domains is not allowed by CORS security rules, but server-to-server requests are permitted so long as those requests include the appropriate session headers to ensure that the API recognizes the authenticated user. Due to the distributed nature of the VOD system, it is also important to be sure to include any headers/cookies set by load balancing infrastructure on the first connection to the API in all subsequent requests. The default session cookie name for the VOD is "JSESSIONID". This will ensure that the sticky sessions created by Al Jazeera's load balancers will route requests to the correct server instance. Without these precautions, API calls may fail as unauthenticated due to the request being made to a server without a valid session in progress.

# API Responses

The API will return JSON representations of responses.

If an error occurs, the following object format will be returned:

{"message":"MESSAGE", "details":"DETAILS", "code":xxx}

The message and details properties help in debugging the issue, while the numeric code gives a reliable way to determine how to react to an error from the user-side of the API (ie; front-end or application code should not perform comparisons on the message and details values to trigger application states).

Many API calls do not have return data. In these cases, a simple success objects are returned:

{"status":200}

### 

**Authentication**

### /api/login

This API method is used to establish a server-side session on a signed-in user's behalf. The following request parameters should be passed to this endpoint:

**UID:**  The UID value from the data returned by a successful Gigya sign-in

**UIDSignature:** The UIDSignature value from the data returned by a successful Gigya sign-in

**signatureTimestamp:**  the signatureTimestamp value from the data returned by a successful Gigya sign-in

An example of using this method from Javascript is below. This message assumes that the onGigyaLogin callback method was executed in response to a successful sign-in using Gigya's social widgets (note that JSESSIONID is not required in this example because the browser automatically appends cookies to requests). Also note that integration with Gigya to perform the initial authentication request to retrieve the UID and signature information is not covered by this documentation, as it related directly to Gigya as an identity provider and does not involve the VOD. See [Gigya’s social login documentation](http://developers.gigya.com/010_Developer_Guide/10_UM360/030_Social_Login) for information on how to integrate with this service.

function onGigyaLogin (data) {

$.ajax({

url: "/api/login",

method: "post",

data: {

UID: data.UID,

UIDSignature: data.UIDSignature,

signatureTimestamp: data.signatureTimestamp

}

}).done(function (result) {

// result contains success or error information

});

}

Or using cURL (note that the pink values should be replaced with actual Gigya values, server session IDs, and host names):

$ curl --data "UID=UID&UIDSignature=SIGNATURE&signatureTimestamp=00000000" --cookie "JSESSIONID=SESSIONID" http://SERVER/api/login

On a successful sign-in, the API will return the following object:

{"status":200}

**User Preferences**

### /api/preferences/get

This API method retrieves a user's saved preferences as well as the data required for rendering preference selection UI. If a valid user session following Gigya sign-in has not occurred, then the preferences API will return default VOD preferences. No parameters are required and the return value is a JSON object with the following format:

{"languages":[], "timezones":[], "countries":[], "preferences":{

"timezone":"TIMEZONE",

"lang":"LANGUAGE",

"country":"COUNTRY"

}, "status": 200}

The returned "languages" array will have objects with a two-character language "code" and a language name. Only languages for which the VOD has localization files for will be present in this array:

{"name":"English", "code":"en"}

The timezone array will be populated with a full list of global timezones, ordered alphabetically. These have a timezone ID (to be used programmatically) and a timezone name (to be displayed to a user):

{"id":"America/Los\_Angeles", "name":"America/Log\_Angeles (Pacific Daylight Time)"}

The countries array will be populated with a full list of possible countries, localized and ordered alphabetically:

{"name": "Bermuda", "code":"BM"}

The preferences object in the return value consists of the user-selected (or system-default if the user has not set preferences) values for the timezone, country, and language (using a timezone ID, language code, and country code from their respective arrays).

### /api/preferences/set

This API method saves a user's preferences to their VOD account. It expects a parameter for each of the three preferences:

**lang:** A valid two-letter language code (as returned by the preferences/get method). Note that only languages that the VOD has been localized in are usable, so take care only to set values that are returned from the preferences/get method.

**country:** A two-letter country code (as returned by the preferences/get method)

**timezone:** A user's selected timezone ID (as returned by the preferences/get method)

**Playlists**

### /api/playlists/get

Used to retrieve a list of a user's playlists. The playlists/get method has no parameters. The return JSON on success will include a status and a list of playlists.

{"status":200, "playlists":[{"id":"xxx", "name":"NAME"}]}

Note that this method does not retrieve a list of videos, rather it retrieves the queries that can be made against the mediaquery API to populate a list of videos as necessary. The "url" parameter of each subscription is the page on which the user subscribed to the given query (for linking back to search results, show pages, etc).

### /api/playlists/create

Used to create a new playlists, optionally inserting a video during the creation process. The playlists/create method expects at least one parameter:

name: Required, the name of the playlist

videoID: Optional, a video to insert into the playlist as it is created

On success, the method will return a JSON object with a "status" attribute and an "id" given to the playlist (for use with other playlist methods):

{"id":"xxx", "status":200}

### /api/playlists/add

Used to add a video to a playlist. This method expects the following parameters:

**videoID:** The Video Cloud media ID of the video to add

**playlistID:** The VOD playlist ID (as returned by the playlists/get or playlists/create methods) to add the selected video to

**playlistName:** The playlist name (as returned by the playlists/get or playlists/create methods) to add the selected video to

**videoName:** The Video Cloud media name of the video to add

The returned JSON object will contain a "status" attribute and an "added" attribute with a value of true or false depending on whether the API was able to add the video to the chosen playlist:

{"status":200, "added":true}

### /api/playlists/remove

Used to remove a selected video from a selected playlist to which it belongs. The following parameters are expected:

**videoId:** The Video Cloud media ID to remove

**videoName:** The name of the video

**playlistId:** The playlist ID (as returned by the playlists/get or playlists/create methods) from which to remove the selected video

**playlistName:**  The name of the playlist from which to remove the selected video

The returned JSON object will contain a "status" attribute and a "removed" attribute with a value of true or false depending on whether the API was able to remove the video:

{"status":200, "removed":true}

### /api/playlists/delete

Used to delete a playlist from the user's account. A single parameter is required:

**playlistId:**  The playlist ID (as returned by the playlists/get or playlists/create methods) to delete

The returned JSON object will contain a "status" attribute and an "deleted" attribute with a value of true or false depending on whether the API was able to delete the chosen playlist:

{"status":200, "deleted":true}

### /api/playlists/rename

Used to change the name of an existing playlist. This method expects two parameters:

**playlistId:**  The ID of the playlist (as returned by the playlists/get or playlists/create methods)

name: The new name to give to the playlist

On success, the return value will be a simple success object:

{"status":200}

### /api/playlists/reorder

Used to change the order in which the playlists are displayed to the user in the VOD. A single parameter is required:

**playlistIds:** A comma-separated list of playlist IDs (as returned by the playlists/get or playlists/create methods) in the desired order

On success, the return value will be a simple success object:

{"status":200}

**Subscriptions**

### /api/subscriptions/get

Used to retrieve a list of a user's subscriptions. The subscriptions/get method has no parameters. The return JSON on success will include a status and a list of subscriptions. Subscription items have the following format:

{"status":200, "subscriptions":[{"id":"xxx", "query":"{...}", "name":"NAME", "url":"http://SERVER/PATH"}]}

Note that this method does not retrieve a list of videos, rather it retrieves the queries that can be made against the mediaquery API to populate a list of videos as necessary. The "url" parameter of each subscription is the page on which the user subscribed to the given query (for linking back to search results, show pages, etc).

### /api/subscriptions/add

Used for adding subscriptions to a user's account from the VOD front-end. The subscriptions/add method expects a request body containing a JSON object with details about the subscription:

{"name":"NAME", "query":"{...}", "url:"http://SERVER/PATH"}

On success, the method will return a JSON object with a "status" attribute and an "id" given to the subscription (for use with the remove and order methods):

{"id":"xxx", "status":200}

### /api/subscriptions/remove

The remove API method expects a single parameter:

**id:**  The subscription ID as returned in a subscription object from the subscriptions/get request

The returned JSON object will contain a "status" attribute and a "removed" attribute with a value of true or false depending on whether the API was able to remove the subscription:

{"status":200, "removed":true}

### /api/subscriptions/order

The subscriptions/order method is used to set the order in which a user's subscriptions are listed in the VOD UI. It expects a request body containing a JSON object with an "ids" attribute. This attribute should be an array of subscription IDs in the order in which they should be displayed:

{"ids":["xxx", "xxx", "xxx"]}

The subscriptions/order method will return with a {"status":200} response on success.

**Auto-resume Positions**

### /api/positions/get

Used to retrieve the last playback index for a given video ID. With this index, a player can seek forward to where the user left off. This method expects a video ID parameter:

**id:** The Video Cloud media ID to get the position for

The return object will contain a status and a position (in seconds). If the video has not previously been started, a position of 0 will be returned:

{"id":"xxx", "status":200, "pos":0}

### /api/positions/set

Used to set the position at which a user left the video. Two parameters are expected:

**id:** The Video Cloud media ID to set the position for

**pos:** The number of seconds into the video the user stopped watching

On success, the return value will be a simple success object:

{"status":200}

### /api/positions/remove

Used when a video has been fully watched to notify the API that it is no longer incomplete. The API removes the video from the unfinished videos list for the user. A single parameter is expected:

**id:** The Video Cloud media ID to set the position for

The returned JSON object will contain a "status" attribute and a "removed" attribute with a value of true or false depending on whether the API was able to remove the playback position:

{"status":200, "removed":true}

**Timeline Events**

### /api/events/timeline

Used to retrieve a list of events for display in a timeline. The method expects a single parameter:

**videoId:** The Video Cloud media ID to set the position for.

The event custom field values on the given video will be used to retrieve a list of other videos that are part of the same event. On success, the following information will be returned:

{"status":200, "videos":[...], "eventName":"NAME"}

Each video in the returned list will contain:

**showUrl:** An absolute path to a show page (to be appended to the VOD domain to create a show page link)

**id:** The Video Cloud media ID of the video

**displayDate:** The localized date to display to the user. This date applies to the video's date within the timeline for the event

**duration:**  The video's length from Video Cloud

**description:** The video's description value from Video Cloud

**thumbnailUrl:** The video's thumbnail image URL from Video Cloud

**name:** The video's name from Video Cloud

**publishDate:** If applicable, the Video Cloud publish date

**showText:** The name of the show that the video is part of (if any)

**Media Query Proxy**

### /api/mediaquery

The mediaquery API is used to proxy Video Cloud API requests to ensure that the Al Jazeera Video Cloud token isn't discoverable in front-end source code. The following parameters are expected:

**query:** A JSON representation of a [Video Cloud API request](http://docs.brightcove.com/en/video-cloud/media/reference.html)

**encrypted:** The VOD performs encryption over subscription list queries when rendering to the front-end to prevent user tampering. This value should be false when performing calls from outside of the VOD.

The return format of a mediaquery API call will be a JSON representation of a Video Cloud Media API response.

**Electronic Programming Guide**

### /api/epg

Used to retrieve EPG information from the VOD for a given channel. Two parameters are expected:

**cid:**  The three-letter channel ID to fetch programming for (such as "eng" or "ara")

**offset:** The number of minutes to offset time values in results (for localizing to timezone)

The result of an EPG API call is a JSON object with:

{"days":[...], "times":[...], "status":200}

Each object in the times array is simply a localized representation of the time to display to the user:

{"time":"1:00 AM"}

Each day is an object:

{"date":"Wed, Oct 30", "Shows":[...]}

Each show contains information as defined in the EPG feeds as fetched from Al Jazeera's feed service ([Arabic](http://feeds-custom.aljazeera.net/ar/vod/epg.php) and [English](http://feeds-custom.aljazeera.net/en/vod/epg.php)).