

API for video publishers

version 1.14

last update 16/12/2011

Concept

The purpose of the **API** for video publishers is to provide web video publishers with a clear and reliable server to server communication interface through which their applications are able to publish video online. This interface provides automated means for nearly all functionality there is in the video panel.

From our perspective an **API publisher** is any user that has a valid **API KEY/API CODE** pair associated with their account. Each publisher may implement any part of this **API** interface in their own application/website depending on their needs.

The **API** is implemented as a set of web services and notification callbacks that receive a standard **HTTP** request(**GET/POST**) and return structured response in the form of **XML** response.

The API services allow for the following:

1. **Listing of publishers videos with filtering**
2. **Deleting a video**
3. **Receiving(changing) properties of a video** (title,aspect ratio,enable/disable,video options)
4. **Receiving video statistics – same as video statistics in site in structured XML form** (detailed & daily builds)
5. **Listing TV or PPV channels**
6. **Listing player templates**
7. **Generating embed codes for video player publishing** (option for using pre-configured templates)
8. **Receiving publisher's streams amount**
9. **Listing(receiving) video images**
10. **Start/stop live broadcast**
11. **Live broadcast feeds**

We provide a client **PHP** library to facilitate integration. The library implements proper call convention and can be used as a solid starting point for application integration.

NOTE: This **API** is intended for a diverse set of requirements. For performance reasons we recommend any heavy load applications/sites to cache all of the needed data with their own service. Then update the local copy of the data when change notifications are received.

Entry point

API can be accessed from <http://api.webvideocore.net/api.php>

Any other parameters can be passed through **HTTP GET/POST** requests. Service name is expected as “a” parameter:

Example of **GET** Request: http://api.webvideocore.net/api.php?a=SERVICE_NAME_HERE

Authentication

All service calls through this **API** require authentication. Authentication is achieved through **API Key / API Code** pair provided by the user panel at *Account > Integration > API Access*.

The **API key/API code** pair is used with the **svp_auth_get_token** service to obtain an authentication **token**.

Error handling

All errors are returned in the form:

```
<response>
  <result>ERROR</result>
  <code>xxxx</code>
  <message>Error message here</message>
</response>
```

Success responses are described per each service bellow.

NOTE: The error messages returned are “error class” messages. For actual detailed error messages look at **appendix 1** in this document.

Notifications

Server to server notifications are available as part of this **API**. These notifications are calls from the platform to publisher's defined callback address(**URL**) that allow publishers to react on changes in your associated data. For further information about notification messages please look at **appendix 2** in this document.

Video services list:

1 . svp_auth_get_token

Description:

Retrieves an authentication token for further use with all other **API** service calls.

Notes:

The authentication token has a meaning similar to session **ID**. The lifetime of the authentication token is currently **48h**. If the service is been accessed with valid token within this time, the validation period will be renewed.

parameter	description
api_key *	Your publisher API Key .
api_code *	Your publisher API Code .
token	Valid API authentication token.

Returned data:

Authentication token code or error code if error occurs.

Example success response:

```
<response>
  <result>OK</result>
  <auth_token>xxxxxxxxxxxxxxxxxxxx</auth_token>
</response>
```

2 . svp_list_videos

Description:

Returns a list of videos based on the search/filtering criteria passed with the parameters.

Notes:

Stream name parameter in video list is present only when video source is live video feed.

parameter	description
token *	Valid API authentication token. See svp_auth_get_token service.
tag_number	Integer user data filtering. You can filter the results based on the tag_number associated with the video. The comparison is for exactness. The tag_number can be set by svp_upload_video and svp_set_video_property services.
tag_string	String user data filtering. You can filter the results based on the tag_string associated with the video. The comparison is for inclusion(Like '%val%'). The tag_string can be set by svp_upload_video and svp_set_video_property services.
title_search	Filter based on video titles. Useful for searching. The comparison is for inclusion(Like '%val%'). The title of the video can be set by svp_upload_video and svp_set_video_property services.
start_ref_no/end_ref_no	Filter returned videos by reference IDs start and end range.
start/limit	Filter returned videos by result set row numbers start and end range. Same as MySQL " Limit start,limit ".
video_source	Each video is linked to a specific video source - on-demand video file or live video feed. You can filter the results based on the video source. Available values: ondemand & live .
count_only	Returns only total count of videos based on search/filtering criteria. <i>NOTE:</i> Video list is not present in response.
channel_ref	TV or PPV channel reference ID . Useful if you need to get only videos in a associated TV or PPV channel playlist. Take a look at svp_list_channels service form information how to get reference IDs dynamically.

Returned data:

A list of videos or error code if error occurs.

Example success response:

```
<response>
  <result>OK</result>
  <count>xx</count>
  <filters>
    <tag_number>xx</tag_number>
    <tag_string>xx</tag_string>
    <title_search>xx</title_search>
    <start_ref_no>xx</start_ref_no>
    <end_ref_no>xx</end_ref_no>
    <start>xx</start>
    <limit>xx</limit>
    <channel_ref>xx</channel_ref>
    <video_source>ondemand/live</video_source>
  </filters>
  <video_list>
    <video>
      <ref_no>1</ref_no>
      <video_key>xxx</video_key>
      <title>My video #1</title>
      <tag_number>1000</tag_number>
      <tag_string></tag_string>
      <video_source>ondemand/live</video_source>
      <stream_name>xxx</stream_name>
    </video>
    ...
  </video_list>
</response>
```

3 . svp_delete_video

Description:

Deletes the specified video from the system.

Notes:

parameter	description
token *	Valid API authentication token. See svp_auth_get_token service.
video_ref *	Video reference ID .

Returned data:

OK or error code if error occurs

Example success response:

```
<response>
  <result>OK</result>
</response>
```

4 . svp_get_video_properties

Description:

Returns a list of all editable properties of the given video.

Notes:

parameter	description
token *	Valid API authentication token. See svp_auth_get_token service.
video_ref *	Video reference ID .

Returned data:

All editable properties of the given video or error code if error occurs. For further information about editable video properties please look at **appendix 4** in this document.

Example success response:

```
<response>
  <result>OK</result>
  <video_properties>
    <property>
      <id>X</id>
      <name>XXX</name>
      <value>XXX</value>
    </property>
    ...
  </video_properties>
</response>
```

5 . svp_set_video_property

Description:

Modifies a property value.

Notes:

parameter	description
token *	Valid API authentication token. See svp_auth_get_token service.
video_ref *	Video reference ID .
property_id *	Property ID received from svp_get_video_properties . For further information about editable video properties please look at appendix 4 in this document.
new_value *	New Value for the property.

Returned data:

OK or error code if error occurs.

Example success response:

```
<response>
  <result>OK</result>
</response>
```

6 . svp_get_video_stats_detail

Description:

Returns a list of video viewer statistics. The data includes all individual views.

Notes:

Detail statistics are available for **3** months. For older data only daily statistics are available.

The data amount can be quite large thus number of nodes returned is limited to **100,000!** Make a second call with a **start_row** parameter set in order to retrieve further data.

parameter	description
token *	Valid API authentication token. See svp_auth_get_token service.
video_ref *	Video reference ID .
start_time *	Start date/time for the requested statistics. Unix time stamp(integer) format. <i>NOTE:</i> Only 3 months ago is supported for the start time.
end_time *	End date/time for the requested statistics. Unix time stamp(integer) format.
start_row	Starting row for the result set. i.e. how many rows are skipped from the start.
limit	Maximum number of rows to return. Can be at most 100,000 . If a higher value is set it is assumed 100,000 .

Returned data:

OK or error code if error occurs.

Example success response:

```
<response>
  <result>OK</result>
  <count>xxxxxx</count>
```

```

<stats>
  <stat>
    <time>xxxxxxxxxxx</time>
    <ip>1.2.3.4</ip>
    <country>GB</country>
    <streams>1</streams>
  </stat>
  ...
</stats>
</response>

```

7 . svp_get_video_stats_daily

Description:

Returns a list of video viewer statistics. The data is aggregated per day.

Notes:

parameter	description
token *	Valid API authentication token. See svp_auth_get_token service.
video_ref *	Video reference ID .
start_date *	Start date for the requested statistics. Unix time stamp(integer) format . <i>NOTE:</i> If there is no records between given start date and date of first video statistics, start date will be made equal to first date.
end_date *	End date for the requested statistics. Unix time stamp(integer) format . <i>NOTE:</i> If there is no records between given end date and date of last video statistics, start date will be made equal to last date.

Returned data:

OK or error code if error occurs.

Example success response:

```

<response>
  <result>OK</result>
  <count>xxxxx</count>
  <stats_daily>
    <stat_day>
      <date>xxx</date>
      <views>xxxxxx</views>
      <streams>yyyy</streams>
    </stat_day>
    ...
  </stats_daily>
</response>

```

8 . svp_list_channels

Description:

Returns a list of **TV/PPV** channels associated with publisher's account.

Notes:

parameter	description
token *	Valid API authentication token. See svp_auth_get_token service.
channel_type	Filter channel playlists by type. Available values: tv & ppv .

Returned data:

List of publisher's channels or error code if error occurs..

Example success response:

```

<response>
  <result>OK</result>
  <count>xx</count>
  <filters>

```

```

    <channel_type>tv/ppv</channel_type>
  </filters>
  <channel_list>
    <channel>
      <ref_no>1</ref_no>
      <title>My TV/PPV channel #1</title>
      <channel_type>tv/ppv</channel_type>
    </channel>
    ...
  </channel_list>
</response>

```

9 . svp_list_player_templates

Description:

Returns a list of player embed code templates that the publisher has saved for further use.

Notes:

parameter	description
token *	Valid API authentication token. See svp_auth_get_token service.

Returned data:

List of publisher's present embed code templates or error code if error occurs..

Example success response:

```

<response>
  <result>OK</result>
  <count>xx</count>
  <embed_code_list>
    <embed_code>
      <ref_no>1</ref_no>
      <title>My template #1</title>
      <embed_type>Embed type name</embed_type>
    </embed_code>
    ...
  </embed_code_list>
</response>

```

10 . svp_get_player_code

Description:

Returns an embed code for publishing. This service can be used for automated code generation in publisher's own website/application.

Notes:

parameter	description
token *	Valid API authentication token. See svp_auth_get_token service.
video_key	Video key. See svp_list_videos service.
video_ref *	Video reference ID . See svp_list_videos service.
param_overrides	Name-Value-Pair list of override template parameters. The NVP data must be URL encoded (each value of NVP string must be URL encoded too). See appendix 3 for more information.
embed_type *	The type of code to be generated. Available values: js_embed , popup , popin , player_link , object & transparent .
template_ref *	Reference code of embed code template that you have saved from platform. <i>NOTE</i> : If you pass a valid template reference code the embed_type* parameter is not necessary (It will be replaced with embed type associated with given template reference code).

You need to pass either `embed_type` or a valid template reference **ID**! You may pass a `video_key` or `video_ref` to get embed code ready for.

Returned data:

HTML code for embedding or error code if error occurs.. The code is returned as base64 encoded string value in order to preserve formatting.

NOTE!!! If you do not pass a **video key** or a **video reference code** the returned code will have the string **%clip_key%** as a placeholder for a valid video key (also **%title%** for pupup embed type). In that case you need to replace that placeholder with a valid **video key** to get a working embed code! This may be used to publish arbitrary videos automatically with the same video code.

Example success response:

```
<response>
  <result>OK</result>
  <embed_code_html>
    K0F483ABX==
  </embed_code_html>
</response>
```

11 . svp_get_streams_amount

Description:

Returns the current streams (**VP**) credits amount with the platform. May be used for custom tracking or notifications.

Notes:

parameter	description
token *	Valid API authentication token. See svp_auth_get_token service.

Returned data:

VP account information or error code if error occurs..

Example success response:

```
<response>
  <result>OK</result>
  <streams_account>
    <account_status>Active/Locked</account_status>
    <total_streams>xxxx</total_streams>
    <monthly_streams>xxxx</monthly_streams>
    <renewal_streams>xxxx</renewal_streams>
    <renewal_date>dd/mm/yyyy</renewal_date>
    <additional_streams>xxxx</additional_streams>
  </streams_account>
</response>
```

12 . svp_list_video_images

Description:

Returns a list of images associated with publisher's videos.

Notes:

The lifetime of the image name is currently **48h**. You can obtain video image content within this time (See service **svp_get_video_image**).

parameter	description
token *	Valid API authentication token. See svp_auth_get_token service.
video_ref *	Video reference ID . See svp_list_videos service.

Returned data:

List of video images or error code if error occurs. Parameter **name** is autogenerated and it is intended for using with service **svp_get_video_image**. Parameter **extension** is image file extension (**jpg, gif, png, etc.**) – you can use it for properly saving of image files on your local environment. Parameter **primary** shows if image in list is marked as startup video image (available values – **yes/no**).

Example success response:

```
<response>
```

```

<result>OK</result>
<count>xx</count>
<image_list>
  <image>
    <number>1...4</number>
    <name>xxx</name>
    <extension>jpg,gif,png...</extension>
    <primary>yes/no</primary>
  </image>
  ...
</image_list>
</response>

```

13 . svp_get_video_image

Description:

Displays video image associated with some of publisher's videos.

Notes:

You have to store video image content in your local environment. The lifetime of generated image names is currently **48h** (See service **svp_list_video_images**), so after that time this service won't be able to display the image.

parameter	description
token *	Valid API authentication token. See svp_auth_get_token service.
image_file *	Image file name. See svp_list_video_images service.
type	Type of displayed image. With this parameter you can obtain thumb or large video image. If this parameter is not passed, the service displays large video image. Available values: thumb, large

Returned data:

Video image content or error code if error occurs. .

14 . svp_set_video_image

Description:

Set video image associated with some of publisher's videos.

Notes:

If video image already exists it will be replaced with new image.

parameter	description
token *	Valid API authentication token. See svp_auth_get_token service.
video_ref *	Video reference ID . See svp_list_videos service.
number *	Image number (must be between 1 and 4). See svp_list_video_images service.
url *	URL of image file. URL string must be URL encoded and maximum 255 chars long.
type *	Type of setting video image. Image can be established directly by image URL or can be stored on Video platform system. Available values: direct, store <i>NOTE:</i> When type is store , actual file pull will start asynchronously from a separate server and may take some time. Do not delete the image file immediately, wait for a notification or do so with a clean up scheduled job on your server, in an hour for example! When type is direct , image thumbnails in Panel video list will be created asynchronously and stored on Video platform server. If you change content of image URL , you have to call this service again in order to generate new image thumbnails. Notifications are sending as notice event type. See APPENDIX 2 – Section 5 .

Returned data:

OK or error code if error occurs.

Example success response:

```

<response>
  <result>OK</result>
</response>

```

15 . svp_set_primary_video_image

Description:

Marks video image as startup image associated with some of publisher's videos.

Notes:

parameter	description
token *	Valid API authentication token. See svp_auth_get_token service.
video_ref *	Video reference ID . See svp_list_videos service.
number *	Image number (must be between 1 and 4). See svp_list_video_images service.

Returned data:

OK or error code if error occurs.

Example success response:

```
<response>
  <result>OK</result>
</response>
```

16 . svp_start_broadcast

Description:

This service can be used to start broadcasting of live videos.

Notes:

parameter	description
token *	Valid API authentication token. See svp_auth_get_token service.
video_ref *	Video reference ID . See svp_list_videos service.

Returned data:

OK or error code if error occurs

Example success response:

```
<response>
  <result>OK</result>
</response>
```

17 . svp_stop_broadcast

Description:

This service can be used to stop broadcasting of live videos.

Notes:

parameter	description
token *	Valid API authentication token. See svp_auth_get_token service.
video_ref *	Video reference ID . See svp_list_videos service.

Returned data:

OK or error code if error occurs

Example success response:

```
<response>
  <result>OK</result>
</response>
```

18 . svp_broadcast_feed

Description:

Broadcast live feed to your audience.

Notes:

parameter	description
token *	Valid API authentication token. See svp_auth_get_token service.
whitelabel	WhiteLabel SVP video destination. Available values: yes & no (default). If value is yes you can re-brand this kind of player and insert your own logo or text (watermark). <i>NOTE:</i> If live video is registered under channel playlist and whitelabel parameter is not present, video destination will be established from default branding option in the channel.
video_title	Title to be associated with this live video in the video list.
tag_number	Integer user data. May be used at publisher's discretion. For example category or internal user ID .
tag_string	String(100) user data. May be used at publisher's discretion. For example tagging information.
channel_ref	TV or PPV channel under which the live video is to be associated with. Useful if you need to publish videos in a TV or PPV channel automatically. Take a look at svp_list_channels service for information how to get reference IDs dynamically.

Returned data:

Live video reference code and key or error code if error occurs.

Example success response:

```
<response>
  <result>OK</result>
  <ref_no>xxxx</ref_no>
  <video_key>xxxx</video_key>
</response>
```

19 . svp_start_recording

Description:

This service can be used to start recording of live videos.

Notes:

parameter	description
token *	Valid API authentication token. See svp_auth_get_token service.
video_ref *	Video reference ID . See svp_list_videos service.

Returned data:

OK or error code if error occurs

Example success response:

```
<response>
  <result>OK</result>
</response>
```

20 . svp_stop_recording

Description:

This service can be used to stop recording of live videos.

Notes:

parameter	description
token *	Valid API authentication token. See svp_auth_get_token service.
video_ref *	Video reference ID . See svp_list_videos service.

Returned data:

OK or error code if error occurs

Example success response:

```
<response>
  <result>OK</result>
</response>
```

APPENDIX 1: API Error Messages

Actual list of error codes and the associated error messages:

1000	Input/output error!
1001	Failed to create API Authorization token!
1002	Failed to delete video!
1003	Failed to set video property!
2000	Invalid input parameters!
2001	Invalid/Missing API key in parameters list!
2002	Invalid/Missing API code in parameters list!
2003	Invalid API key / API code pair!
2004	Invalid API authorization token!
2005	Token validation period has been expired!
2006	Invalid/Missing API authorization token in parameters list!
2007	Invalid/Missing clip reference code in parameters list!
2008	Invalid video reference code!
2009	Invalid start time in video statistics service!
2010	Invalid end time in video statistics service!
2011	Invalid channel reference code!
2014	Invalid property reference ID!
2015	Invalid/Missing new value in parameters list!
2016	Invalid aspect ratio value!
2018	Invalid template reference code!
2019	Invalid/Missing embed type!
2020	Invalid video key!
2021	No available video storage!
2022	Invalid video file extension! <i>(NOTE: Available video file extensions: avi, mpg, mpeg, flv, vob, mpeg2, divx, mp4, mov, wmv, m4v).</i>
2023	Invalid/Missing image file name!
2024	Unable to find video image resource!
2025	Invalid/Missing video image number!
2026	Invalid/Missing video image URL!

2027	Invalid/Missing video image type!
2028	Invalid video type!
2029	Invalid live video file!
2030	Broadcast is started already!
2031	Broadcast is stopped already!
2033	Not available live streaming access!
2034	Recording is started already!
2035	Recording is stopped already!
2036	Recording service temporary unavailable!

APPENDIX 2: API Callback Notifications

NOTE: In order to receive notifications you must configure your callback **URL** in the Panel at **Account > Integration > API Access**.

All notifications initiated by the publisher **API** are implemented as asynchronous calls. They are initiated with a slight delay from the actual operations performed. The delay would usually be under **10seconds**.

The API notifications are implemented as **HTTP GET** calls to an user specified **URL** address. Parameters are passed with the call depending on the notification event type.

1 . video_transcode

Description:

This notification is sent when there is a status update related to the video transcode system.

Notes:

You will receive this notification if any video associated with your account has had a change in encoding status. This notification can be used for automated publishing of videos whenever they become available in the system.

parameter	description
operation_name *	=video_transcode
result *	OK/ERROR
clip_ref *	The clip reference ID.
clip_key *	The clip key.
err_msg	Error message if result is ERROR .

2 . delete_video

Description:

This notification is sent when a video associated with your account has been deleted through the **API** or the panel.

Notes:

This is an **AFTER DELETE** type of notification! Any **API** calls related to the related video reference **ID** will fail! This notification can be used to update any cached video lists.

parameter	description
operation_name *	=delete_video
result *	OK/ERROR
clip_ref *	The clip reference ID.

clip_key *	The clip key.
err_msg	Error message if result is ERROR .

3 . edit_video

Description:

This notification is sent when some of the properties of the video have been changed.

Notes:

This notification can be used to update any cached video lists.

parameter	description
operation_name *	=edit_video
result *	OK/ERROR
clip_ref *	The clip reference ID.
clip_key *	The clip key.
err_msg	Error message if result is ERROR .

4. upload_video

Description:

This notification is sent when procedure of uploading files has been successfully finished or failed.

Notes:

This notification can be used to delete the uploaded file from your server or some corrections if uploading is failed.

parameter	description
operation_name *	=upload_video
result *	OK/ERROR
clip_ref *	The clip reference ID.
clip_key *	The clip key.
err_msg	Error message if result is ERROR .

5. notice

Description:

This notification is sent with various operations in system. It can be used like debugging information.

Notes:

parameter	description
operation_name *	=notice
subject *	Subject of notice. See table below for more information.
type *	SUCCESS/ERROR/WARNING
ref_no *	Reference ID associated with subject of notice.
recorded_from_stream	This parameter is available only with video.publish notice type. It indicates the source stream name of recorded video.
message	Message depending on subject of notice.

Notice subjects:

type	description
video.publish	This notification is sent when video is ready for publishing.
video_image.create_thumbs	Notifications about result of creating video thumbs in direct type of setting video images. See svp_set_video_image service for more information.

video_image.upload_image	Notifications about result of uploading video images in store type of setting video images. See svp_set_video_image service for more information. <i>NOTE:</i> There is no separate notification for creating video thumbs.
--------------------------	--

APPENDIX 3: Embed code generator parameter overrides

When you pass parameter overrides to the **svp_get_player_code** service these override parameters replace the ones saved in the template used thus allowing certain customizations to the code used in the publisher's application.

parameter	description
player_skin	Available values: thin_controls (default), thin_controls_old . Player skin to be used with this player. Further skins will become available when such are added to the system.
player_color1	First color of player skin coloring. Default color is #c8c8c8. <i>NOTE:</i> The value should be hexadecimal representation of color. Example: #FF00FF
player_color2	Second color of player skin coloring. Default color is #a6a6a6. <i>NOTE:</i> The value should be hexadecimal representation of color.
width	Player width in pixels. Default width is 640px.
height	Player height in pixels. Default height is 480px.
video_width	Video width in pixels. <i>NOTE:</i> Player width is calculated automatically based on selected player skin.(Calculated player width overrides width parameter if it passed).
video_height	Video height in pixels. <i>NOTE:</i> Player height is calculated automatically based on selected player skin. (Calculated player height overrides height parameter if it passed).
stretch_video	Available values: yes & no (default). If the value is yes the video will be stretched to fit the player area. This option will override the original aspect ratio of the video.
ratio	Available values: 16:9 & 4:3 . If this parameter is passed player dimension is calculated proportionally by video_width or video_height parameter (only one of these parameters have to be passed).
brand_link	Available values: new_window (default), same_window . Select the open mode for the branding link (if any) in this player.
player_transparency	Available values: yes & no (default). Some player skins use effects and alpha channels to blend with the background. This option allows to control that behaviour. This mode can be used to create transparent video players.
pause_on_start	Available values: yes (default) & no . If value is yes the visitor must click the play button to start video playback.. <i>NOTE:</i> This is available only if value of parameter hide_controls is not set to yes .
show_image	Available values: yes (default) & no . If value is yes the selected image in clip properties is shown as a startup screen. <i>NOTE:</i> This is available only if value of parameter pause_on_start is not set to no .
start_volume	Available values: Number between 0 and 100 (Default volume is 50%).
video_prebuffer	Available values: yes & no (default). If value is yes the video will start prebuffering immediately. Note that this setting will impose a load on visitor's connection even if the visitor has not decided to watch the video! Also you will be charged for video delivery as the video is delivered even without being watched! <i>NOTE:</i> This is available only if value of parameter pause_on_start is not set to no .
video_loop	Available values: yes & no (default). If value is yes the video will start from the beginning when finished.
hide_controls	Available values: yes & no (default). If value is yes only the video is shown without controls. Only auto-play setting applies!
auto_hide_controls	Available values: yes & no (default). If value is yes player controls are automatically hidden if no

	action(mouse movement/click) is taken for a short amount of time in the player. The controls reappear if mouse is pointed at the player. <i>NOTE:</i> This is available only if value of parameter hide_controls is not set to yes .
player_align	Available values: none (default), top_left , top_center , top_right , middle_left , middle_center , middle_right , bottom_left , bottom_center , bottom_right . Align of the player in the screen.
offset_x	Player offset (in pixels) in x coordinate (Default offset X is 0). <i>NOTE:</i> This parameter is available only if player_align is different than none
offset_y	Player offset (in pixels) in y coordinate (Default offset Y is 0). <i>NOTE:</i> This parameter is available only if player_align is different than none
allow_fullscreen	Available values: yes (default) & no . If value is yes the [full] button on the player will be available and the viewer can switch to a bigger or fullscreen player. <i>NOTE:</i> This is available only if value of parameter hide_controls is not set to yes .
fullscreen_mode	Available values: big_screen , full_window & flash_native . big_screen fullscreen mode is a larger player shown in the middle of the browser window over your webpage. full_window is when the player fills the entire browser window. flash_native uses native flash plugin capability and fills the entire screen. This mode will only work when Flash9 is installed on the client computer. <i>NOTE:</i> Larger player dimensions may result in poor playback performance on low end computers and will make obvious video quality problems at close sighting range. This is available only if value of parameter allow_fullscreen is not set to no . flash_native is not available for popin embed mode.
background_color	Player background color. Default background color is #FFFFFF <i>NOTE:</i> The value should be hexadecimal representation of color. Example: #FF00FF. This parameter is available only if fullscreen_mode is set to big_screen .
background_transparency	Available values: Number between 0 and 100 (Default 0). This is the transparency of the layer which is put above your page and behind the player in big_screen fullscreen mode. <i>NOTE:</i> This parameter is available only if fullscreen_mode is set to big_screen .
link_type	Available values: text_link (default) & image_link . Choose how the link on your site pointing to the popup player will look like.
link_title	This text will appear as a textual link or set as title for an image link depending on the link_type setting above.
image_address	For an image link fill in the address of the image you would like to use for the link.
close_button	Available values: yes & no (default). Close button of player in the top right corner.

APPENDIX 4: Video editable properties list

property ID	name	description
1	Title	Video title which will help you to differentiate this video from the others
2	Enabled	Video status. Available values: 2 - video is being encoded. 1 - video is active and can be watched. 0 - video is disabled - the player still will be visible in your website, but will not play the movie.
3	Tag Number	Video Tag Number is number (Integer) which will help you to differentiate this video from the others
4	Tag String	Video Tag String is string (max 100 characters) which will help you to differentiate this video from the others
5	Aspect Ratio	Use this property to correct the way of how the image is displayed in the video player. If there are black stripes around the player, simply try each option and save the changes until you find the best result.
6	Channel Name	Channel Playlist name – this property is available only if video is part of TV or PPV channel playlist

7	Channel Ref	Channel Reference Code - this property is available only if video is part of TV or PPV channel playlist
8	Whitelabel	<p>Whitelabel branding status of video. Available values:</p> <p>yes - whitelabel (rebrandable) video - ideal for web designers / developers. You can rebrand this kind of players and insert your own logo or text (watermark). The embed code is also whitelabeled, so you can use them for reselling purposes. During each playback billing point, the system will deducts 1.5 Streams from your account.</p> <p>no – standard video - there is a watermark "powered by our company". Branding cannot be removed/modify by the user. Use whitelabel or TV/PPV channel publishing solutions if you wish to insert your brand. During each playback billing point, the system will deducts 1 Stream from your account.</p>
9	Allowed Domains	<p>Our policy is to keep your content private. It is not essential to apply domain name restriction unless you have some concerns that your online viewers can copy the embed code you have used to publish the video in the first place and use that code for other websites. In that case if you set up domain name restrictions, the embed code will not be valid for other websites. Add the allowed domains separated by comma(",").</p> <p>DO NOT enter www. or http://</p> <p>Leave empty to allow all domains. Subdomains and directories if used will further limit the access.</p> <p><i>EXAMPLE:</i> domain1.com , mydomain2.com , mydomain3.com</p> <p><i>NOTE:</i> This property is available only for single videos.</p>