



## YouTube API Integration documentation

Created by [Yougapi Technology](http://codecanyon.net/user/yougapi/portfolio) - July 2011

Find out our other products on <http://codecanyon.net/user/yougapi/portfolio>

---

Thank you for purchasing our product. This is the documentation to help you getting the product working. You can contact us if you need assistance and we'll do our best to help you.

# Introduction

This application enables you to literally get all YouTube on your own website. You can integrate the YouTube search so your visitors can search among millions of videos and watch them directly on your website. You can also integrate videos from YouTube channels, featured videos, requests videos by categories and use more than 25 available filters.

You also have **2 types of display**. One that can display the videos right under the title (just like on Facebook) and another display a wall of thumbnails than opens a video inside a fancybox on click. This is the most advanced YouTube integration ever ! And it's flexible enough to be extended the way you wish.

**This application can be setup in less than 2 minutes.** Just follow the instructions given in the next page.

# Installation & configuration

Once the app package is copied on your FTP server, open the file “config.php” situated in the folder “vbox/include” and update the variables with your custom data.

You need to define **only one mandatory value**. The “path\_vbox” that is the path from the root of your server to the vbox folder (**with** a slash at the end).

**Example:**

```
$GLOBALS['app_url'] = '/products/youtube/vbox/';
```

That's all for the configuration !

## Some possible customizations

**`$GLOBALS['social_sharing']`**

Whether or not to integrate the display of social icons for the list type of display (when a video id is displayed). Possible values: 0 or 1.

# How it works

**Please also note that you can refer to the 2 working demo scripts included to better understand how all is working together and see it in action !**

Most of the display is done through AJAX, passing the parameters through a Javascript object and calling a function. All the parameters (including the type of display) are passed to that object.

First you need to load the “vbox” library (that manage all the API calls and YouTube videos):  
`include_once('./vbox/include/webzone.php');`

Then you need to instantiate the Vbox class and call a function that will include all the Javascript and CSS files needed by the API to work:

```
include_once('./vbox/include/webzone.php');
$v1 = new Vbox_class();
$v1->display_vbox_js_css();
```

You will notice a Javascript object is declared with all the parameters:

```
$videoCriteria
= "{dom:$domVideoResults, 'q':$q, 'username':$username, 'feed':$feed, 'time':$time, 'type':$t, 'category':$cat, 'pageNumber':$pageNumber, 'nb_display':$nb_display, 'display_type':$display_type}";
```

## Where to display the videos

**\$domVideoResults:** contains the value dom where you want to have your videos displayed.

You will notice on the same page this: `echo '<div id="'. $domVideoResults. "'></div>';`

It simply means that this DIV will contains all the videos loaded through AJAX.

Once you have saved the previous declared Javascript object through this declaration:

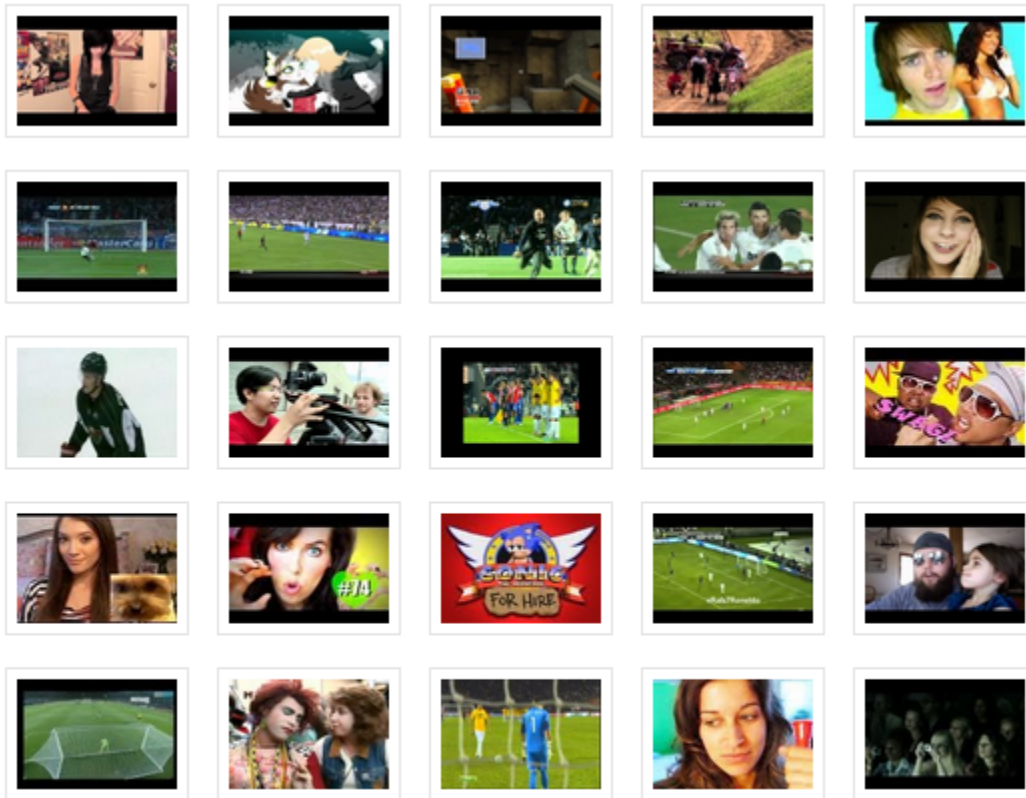
```
$('body').data('videoCriteria', $videoCriteria);
```

You can call the function that will call the display of the videos:

```
displayVideosList();
```

## The types of display

This application comes with 2 possible displays. The **grid type of display** that opens each video inside a Fancybox.




1 2 3 >>


To get this display you need to set the “display\_type” Javascript object parameter to “2”.

And the **list type of display** which display the videos just like a Facebook wall and opens the videos right bellow the clicked video thumbnail or title.

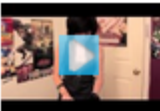
1 2 3 4 5 >>




**Big Blue Ball Machine**  
01:19 - Views: 440,426



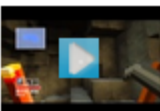
**Minecraft - "Shadow of Israphel" Part 28: Fire and Ice**  
19:03 - Views: 365,250



**Me Singing "Skyscraper" by Demi Lovato (Cover)**  
04:41 - Views: 405,122



**[TEASER] 2NE1 - HATE YOU**  
00:35 - Views: 425,919



**Simon and Hannah play Total Miner: Forge**  
19:03 - Views: 248,174

To get this display you need to set the “display\_type” Javascript object parameter to “1”.

## The filters available

We have seen some filters related to which type of display to enable. But the Javascript object takes other filters like the number of videos to display, which kind of videos feeds to load, and more.

The “**type**” is the most important filter. It tells which videos feeds to requests from YouTube:  
**type=1**: searched videos. With that filter the “q” parameters takes the keywords to search for.  
**type=2**: correspond to the video channel. Works with the “username” parameter.  
**type=3**: correspond to the featured videos. Works with the “feed” and “time” parameter.  
**type=4**: correspond to the videos favorited by a user. Works with the “username” parameter.  
**type=5**: correspond to the videos by categories. Need to set the “category”

You will also notice that most of these parameters are passed dynamically to the Javascript object through the URL get variables. All that is coming from the forms on the page (you can check the 2 demo script for a more practical example).

**Search:**

**Username:**

**Featured:**

Feeds:  ▼

Time:  ▼

**Categories:**

▼

- Film & Animation
- Autos & Vehicles
- Music
- Pets & Animals
- Sports
- Travel & Events
- Gaming
- Comedy
- People & Blogs
- News & Politics
- Entertainment
- Education
- Howto & Style
- Nonprofits & Activism
- Science & Technology

# Files used by the application

You don't need to read this section. It's more to give some information about what files and/or libraries are used by the application.

## 1/ Javascript files

The jQuery library need to be included.

It's done through these declarations:

```
<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.5.2/jquery.min.js"></script>
```

## 2/ CSS files

We have to include 2 CSS files. One directly coming from the jQuery Mobile framework, and one custom file related to our app.

```
<link rel="stylesheet" href="include/css/style.css" type="text/css" media="screen, projection">
<link rel="stylesheet" href="include/css/grid.css" type="text/css" media="screen, projection">
```

## 3/ Fancybox files

```
<script type="text/javascript" src="include/js/fancybox/jquery.fancybox-1.3.4.pack.js"></script>
<script type="text/javascript" src="include/js/fancybox/swfobject.js"></script>
<link rel="stylesheet" href="include/js/fancybox/jquery.fancybox-1.3.4.css" type="text/css"
media="screen" />
```

# Credits

Fancybox library: <http://fancybox.net/>

YouTube API v2: [Data API Protocol](#).

# Contact us

We always want to make sure that you are satisfied with your purchase. For support you can contact us using the following ways. Please don't forget to rate us on codecanyon if you like our work.

=> <http://codecanyon.net/user/yougapi>

=> <http://www.yougapi.com>

=> [contact@yougapi.com](mailto:contact@yougapi.com)