

# **jQuery: full screen videos: a work in progress with WordPress and Videopress**

I'm currently working on a video-based WordPress site which is basically a container for full screen videos. The entire video streaming hinges on the Videopress service. The main requirement of this site is that videos must be handled as a continuous carousel by also providing a base set of video controls (play, pause, progress, forward, backward, full screen mode). Here is what I've learned so far.

## **How the Videopress plugin works**

Videopress delivers videos through a simple plugin which is made up of two components:

1. The shortcode generator for the Media Library
2. The JavaScript plugin

The JavaScript plugin inserts videos into your pages. This plugin detects browser's capabilities so that videos will be always served in the most appropriate format. For example, HTML5-capable browsers will get OGG or MP4 files (depending on the browser) while obsolete browsers will simply receive a Flash version of the video.

## **jQuery vs Videopress**

The major problem with Videopress is that it simply creates a shortcode to be inserted within your posts. In turn, the JavaScript plugin generates the video element once the DOM is fully loaded.

If you take a look at the video's URL provided in your Media Library, you'll be surely disappointed: there's only one single URL, though we all know that Videopress generates several versions of the same video.

In Chrome you will see only the OGG/OGV (Theora) version of your file:

```
https://videos.files.wordpress.com/videoID/video_fmt1.ogv
```

But if you open the same page with Safari, you'll see something different:

```
https://videos.files.wordpress.com/videoID/video_dvd.mp4
```

The difference is in the suffix of the video file: `_fmt1` for Theora videos and `_dvd` for MP4 files. Now you need only to extract the video URL from the markup generated by the Videopress shortcode.

Before doing so, however, we need an utility object to detect browsers. As we'll see later, most of our problems will come from IE and Safari so we actually need to target these browsers:

```
var Browsers = {  
  isSafari: function() {  
    var ua = navigator.userAgent;  
    var u = ua.toLowerCase();  
    if (u.indexOf('safari') != -1 &&  
u.indexOf('chrome') == -1) {  
      return true;  
    }  
  }  
};
```

```

        } else {
            return false;
        }
    },
    isIE: function() {
        var ua = navigator.userAgent;
        var u = ua.toLowerCase();
        if (u.indexOf('msie') != -1) {
            return true;
        } else {
            return false;
        }
    }
};

```

Now we need to properly collect video's URL in the appropriate format and remove the DOM structure created by Videopress:

```

var posters = $('img.videopress-poster');
var compilation = [];

posters.each(function() {
    var $poster = $(this);
    var src = $poster.attr('src');
    var videoURL = '';

```

```
        if(Browsers.isSafari() || Browsers.isIE()) {
            videoURL =
src.replace('_dvd.original.jpg', '_dvd.mp4');
        } else {
            videoURL =
src.replace('_dvd.original.jpg', '_fmt1.ogv');
        }
        compilation.push(videoURL);
        $poster.parent().remove();
    });
```

At last we have our compilation! It's time to display it.

## The BigVideo plugin

BigVideo creates stunning full screen background videos with ease. To create our compilation we simply need to add the following code:

```
var BV = new $.BigVideo();
BV.init();
BV.show(compilation);
```

Take a look at the public methods provided in the plugin's description page for further information.

## Going full screen

screenfull.js is an excellent wrapper for cross-browser usage of the JavaScript Fullscreen API. This plugin mitigates the existing differences between those browsers that already have an implementation of this API.

Unfortunately, IE has no support for the Fullscreen API. Trying to mess with Active X objects just to mimic the F11 key makes no sense to me, so the best way to deal with IE's users is to inform them.

You can use this plugin as follows:

```
$('#fullscreen').on('click', function(e) {  
    e.preventDefault();  
    screenfull.request(document); // Enter full  
screen  
});
```

You can later use the `exit()` method to exit from the full screen mode, for example after a user has pressed the Esc key. You can see the live demo in the video below.

