

# Searching videos on YouTube with jQuery and the YouTube Google APIs

Running searches on YouTube is easy with jQuery and the YouTube APIs.

We start from the following markup:

```
<body>

<form action="" method="get" id="search-youtube">
    <div>
        <input type="text" name="q" id="q" />
        <input type="submit" value="Search"
    />
    </div>
</form>

<div id="response"></div>

<script type="text/javascript" src="jquery.js">
</script>
<script type="text/javascript" src="app.js"></script>
<script src="https://apis.google.com/js/client.js?
onload=init"></script>
</body>
```

The `init()` method must be global. We can build our code as follows:

```
(function( $ ) {

    var App = {
        init: function() {
            gapi.client.setApiKey( "API key" );
            gapi.client.load( "youtube", "v3",
function() {
                App.getVideos();
            });
        },
        getVideos: function( query ) {
            query = query || "jquery";

            var request =
gapi.client.youtube.search.list({
                part: "snippet",
                type: "video",
                q: query,
                maxResults: 6,
                order: "viewCount"
            });

            request.execute(function( response ) {
                var results = response.result;
                var html = "";
                $.each( results.items, function(
index, item ) {
                    var title = item.snippet.title;
                    var id = item.id.videoId;
                    html += "<div class='video'>";
                    html += '<iframe width="560"
```

```

height="315" src="https://www.youtube.com/embed/' +
id + '?controls=0" frameborder="0" allowfullscreen>
</iframe>';

        html += "<h3>" + title + "</h3>";
        html += "</div>";
    });
    $( "#response" ).html( html );
});
}
};

$(function() {
    $( "#search-youtube" ).submit(function( e ) {
        e.preventDefault();
        var queryStr = encodeURIComponent( $( "#q"
).val() ).replace( /%20/g, "+" );
        App.getVideos( queryStr );
    });
});

window.init = App.init;

})( jQuery );

```

## Caveats

- Our code won't work locally.
- After creating our app in the Google console, we must limit the access to one or more specific domain(s). Otherwise, our API key can be used globally.