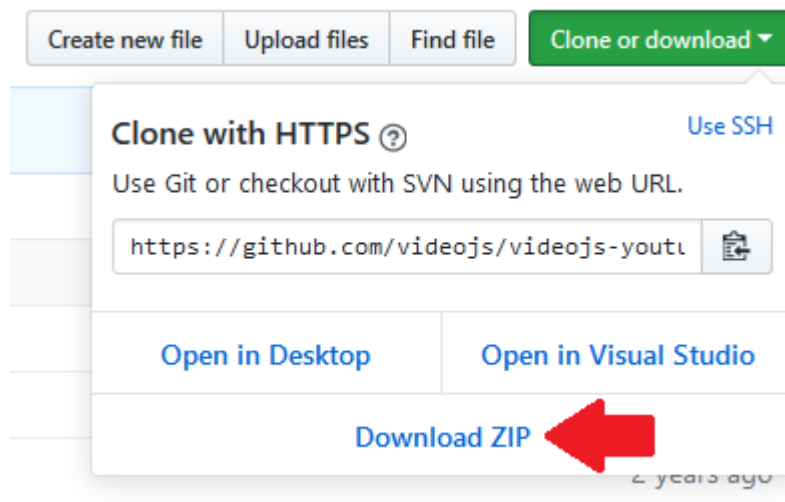


Adding VideoJS

This tutorial shows how you can install and use VideoJS as a free alternative to JWPlayer.

Adding the CDNs:

1. Browse to: <https://github.com/videojs/video.js/blob/stable/docs/guides/setup.md>
2. Locate the section named CDN Version and copy the two HTML tags listed there.
`<link href="//vjs.zencdn.net/5.4.6/video-js.min.css" rel="stylesheet">`
`<script src="//vjs.zencdn.net/5.4.6/video.min.js"></script>`
3. Go to the `_Layout.cshtml` file in the `Views/Shared` folder in your solution.
4. Paste in the two code-lines above the `</head>` element.
5. Browse to: <https://github.com/videojs/videojs-youtube>
6. Click on the Green download button and select **Download ZIP**.



7. Unpack the Zip file to a folder.
8. Go to your solution and expand the `Scripts` folder.
9. Drag the two files `Youtube.js` and `Youtube.min.js` from the folder where you unzipped the ZIP file and drop them in the `Scripts` folder in your solution.
10. Drag the `Youtube.min.js` file from the `Scripts` folder below the two code-lines you just added to the `_Layout.cshtml` file. Make sure the `<script>` tag you drop end up above the `</head>` element.
`<script src="~/Scripts/Youtube.min.js"></script>`
11. Add a new JavaScript file called `videoJS.js` to the `Scripts` folder.
12. Add a function called `videoJS` that takes one parameter called `video` and call the video player function.

```
function videoJS(video) {  
    var container = document.getElementById("video");  
    videojs(container, {  
        controls: true,  
        techOrder: ["youtube"],  
        sources: [{ type: "video/youtube", src: video }]  
    }, function () {  
    });  
}
```
13. Save the file.

14. Open the *BundleConfig.cs* file in the *App_Start* folder and add the *videoJS.js* file to the **ui** script bundle, then save the file.

```
bundles.Add(new ScriptBundle("~/bundles/ui").Include(
    "~/Scripts/carret.js",
    "~/Scripts/JWPlayer.js",
    "~/Scripts/videoJS.js", /*Alternate video player*/
    "~/Scripts/code.js",
    "~/Scripts/RegisterUser.js",
    "~/Scripts/login.js",
    "~/Scripts/forgot-password.js"
));
```

15. Open the *Content.cshtml* view and add a script section at the bottom of the view with a function that loads the player by calling the **videoJS** function you added in the *videoJS.js* file. If you have implemented JWPlayer, you only have to add the highlighted code.

```
@section scripts
{
    <script type="text/javascript">
        $(function () {
            //jwVideo($("#hiddenUrl").text());
            videoJS($("#hiddenUrl").text());
        });
    </script>
}
```

16. Add a video element to the if-block that checks that a valid video URL exist in the model. It's the same block you added the <div> for the JWPlayer player in.

```
@if (Model.VideoURL != null)
{
    @*<div id="video" class="video-margin">&nbsp;</div>*@
    <div id="hiddenUrl" hidden="hidden">@Model.VideoURL</div>
    <video id="video" class="video-js vjs-default-skin vjs-big-play-centered vjs-16-9" controls preload="auto" > </video>
}
```