Getting Started

Before using the JavaScript API, consult the following article for prerequisites. It also covers concepts that will help you understand its use.

Prerequisites

In order for the JavaScript API to function, ensure the following details have been set up in advance:

VideoManager Pro:

- Check your lazy loading setting (listed under the Player's "Additional Settings") - if lazy loading is enabled, the Player (including the API) will not load until a user clicks play

Embed Codes

Finding video embed codes in your VideoManager is an important first step in preparing to use the JavaScript API.

VideoManager Pro

In the front end of VideoManager Pro, embed codes are found in a video's **Publish** view:

Use the "HTML-CODE" (the other embed types cannot be used with this API) - an example code is shown below:

```
<div mi24-video-player video-id="4Zo1QNH5FKxQDLsgKBpvD8"
player-id="2X32X_p31xVFCpkY7KoDhj" config-type="vmpro"
flash-path="//e.video-cdn.net/v2/" api-url="//d.video-cdn.net/play">
</div><script src="//e.video-cdn.net/v2/embed.js"></script>
```

You can also get a video's embed code using the "embed-codes" endpoint of the movingimage REST API.

Each embed code includes the JavaScript source; this is the "embed.js" file, which is needed to make the embed code work.

However, it only needs to be added to the page once, regardless of how many players will be added to the page.

The embed code also includes a <div> that contains our "mi24-video-player" tag and the following attributes:

Attribute	Example Value	Description
video-id	"4Zo1QNH5FKxQDLsgK BpvD8"	a unique string identifying the video that will be embedded (the value listed here is just an example)
player-id	"2X32X_p31xVFCpkY7K oDhj"	a unique string identifying the Player that the embedded video will use (the value listed here is just an example)
config-type	"vmpro"	indicates the VideoManager type the embed code is for ("vmpro" or "vm")
flash-path	"//e.video-cdn.net/v2/"	the flash path of the Player (this is used to supply a backup Flash player in the event that the HTML5 player does not load)
api-url	"//d.video-cdn.net/play"	the URL of the API

To use the JavaScript API, you will need to make use of this embed code in one of two ways:

1. you will either need to add an "id" attribute to the embed code so that you can reference it in JavaScript (as in the Adding or Removing a Player tutorial), or
2. you will use these attributes to create a JavaScript object that you can use to add a Player (as in the Onload Callback Function tutorial).

> ✅ We highly recommend the second option, which auto-loads the Player as soon as it is available. The first option should only be used when you wish to add the Player manually.

Video and Player IDs

The video and player IDs are required embed code attributes, so it's important to know what they are. All of the videos and Players from a VideoManager have their own unique IDs. The embed code for every video will include a "video-id" and a "player-id". In the tutorial examples that follow this chapter, you will need to insert your own video-id and player-id whenever they are used.

Video ID - A unique string used to identify and differentiate each video. The embed code for one video will always have a different video-id than the embed code for another video. This ID is also listed in the metadata accordion menu of the video in VideoManager Pro. You can also query a list of videos along with their IDs by using the "videos" endpoint in the movingimage REST API.

Player ID - A unique string used to identify and differentiate each Player. Different Players are created in the Player Generator area of VideoManager Pro. Several Player customizations can be made there, including color scheme, responsive settings, etc. The Player ID can be found in the metadata accordion menu of the Player. It will also appear in the embed code of a released video's publish accordion menu. There you can select a Player to get an embed code with the corresponding Player ID. Since many videos can be set up to use the same Player, the embed codes for two different videos can have the same player-id.