

# VideoPlayer.Collection Object

The JavaScript API is accessed by initializing a "VideoPlayer.Collection" object. There are two functions which can be used to do this (both take the id of the HTML element containing your Player as an argument):

1. This function lets you add the Player dynamically:

```
VideoPlayer.Collection.addPlayerById("player_container");
```



The "addPlayerById" function can take an optional second argument, an object which will initialize your Player. The object can contain parameters to initialize the Player (the attributes from the embed code) and/or a success callback function, which will be called when the Player is finished loading successfully. These options are covered in detail in the [Onload Callback Function](#) tutorial.

2. And this function lets you access an existing Player:

```
var player = VideoPlayer.Collection.getPlayerById("player_container");
```



Please note that you must wait for the player to load before you can connect to it. If lazy loading is enabled, the player will not load until a user clicks play. Find more information on lazy loading [here](#).

You can also remove a Player dynamically with the VideoPlayer.Collection object by calling this function:

```
VideoPlayer.Collection.removePlayerById("player_container");
```

After connecting the VideoPlayer.Collection object to an element containing a Player, you can then access any of the API functions, register events, and change the appearance of Player elements. The following chapter contains several tutorials that will demonstrate these features in greater detail.

## Enable or disable analytics tracking

To implement custom solutions, such as a custom consent banner as an alternative to the consent dialog that we provide (see the documentation [here](#)), you can use the JavaScript API to tell the player to accept or reject analytics.



Note that the steps below require a custom consent banner or element to be displayed.

## HTML Example

A content banner can look like this in HTML:

```

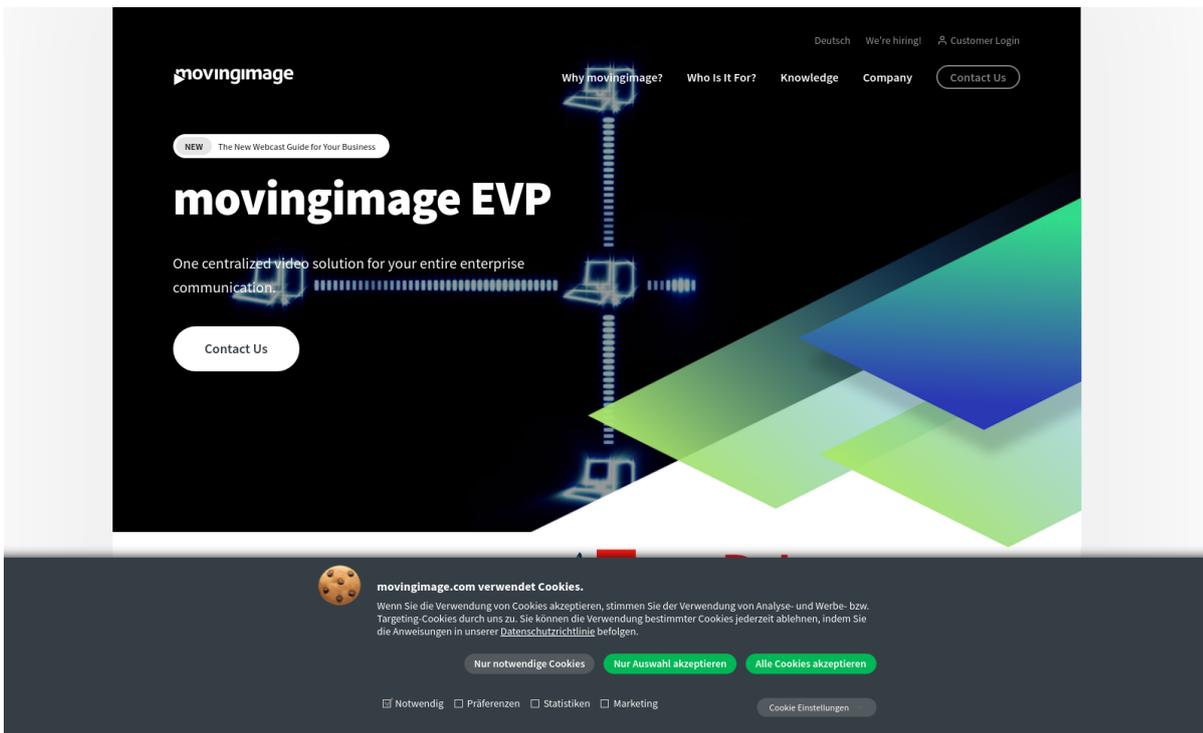
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <title>Your website</title>
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <script src="main.js"></script>
  </head>

  <body>
    <!-- Embedded video -->
    <div
      mi24-video-player
      video-id="<VideoID>"
      player-id="<PlayerID>"
      channel-id="6"
      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>

    <!-- Consent Banner-->
    <div class="consent-banner">
      <button id="accept-button">Accept</button>
      <button id="reject-button">Reject</button>
    </div>
  </body>
</html>

```

The viewer will see a consent banner like this on your website:



## Functions

To enable or disable analytics, add the following functions to your script:

- **Enable analytics**

```
VideoPlayer.Collection.acceptAnalytics()
```

- **Disable analytics**

```
VideoPlayer.Collection.rejectAnalytics()
```

The following example shows you how to enable or disable analytics in the script.

Note that the accept and reject buttons are connected to the functions.

```
// main.js
document.getElementById("accept-button").addEventListener("click", function () {
  VideoPlayer.Collection.acceptAnalytics();
});
document.getElementById("reject-button").addEventListener("click", function () {
  VideoPlayer.Collection.rejectAnalytics();
});
```

## iframe Example



If the video player is within an iframe, the parent domain calls the postMessage API to set consent.

If your video is embedded within an iframe, your HTML code would look like this:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <title>Your website</title>
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <script src="main.js"></script>
  </head>

  <body>
    <!-- Embedded video -->
    <iframe
      id="player_iframe"
      width="800"
      height="450"
      src="<SOURCE_URL>"
      allowfullscreen
      frameborder="0"
    ></iframe>

    <!-- Consent Banner-->
    <div class="consent-banner">
      <button id="accept-button">Accept</button>
      <button id="reject-button">Reject</button>
    </div>
  </body>
</html>
```

In turn, the script for an embedded video within an iframe could look like this:

```
// main.js
document.getElementById('accept-button').addEventListener('click', function(){
  var iframeWindow = document.getElementById("player_iframe").contentWindow;
  iframeWindow.postMessage('CONSENT_GIVEN', '*');
});
document.getElementById('reject-button').addEventListener('click', function(){
  var iframeWindow = document.getElementById("player_iframe").contentWindow;
  iframeWindow.postMessage('CONSENT_DECLINED', '*');
});
```

Please be aware that iframes are completely isolated from each other, so if multiple videos are embedded into the page with iframe codes, you will need to send the "CONSENT\_GIVEN" post message to all the relevant iframes in the current page.

The same is valid for giving and declining consent with the "Consent Dialog", all the dialogs will need to be "Accepted" / "Declined" individually.