

# Enable or disable analytics using the JavaScript API

You can also implement your own solution, such as a custom consent banner as an alternative to the consent dialog that we provide (see the documentation [here](#)).

In these cases, 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

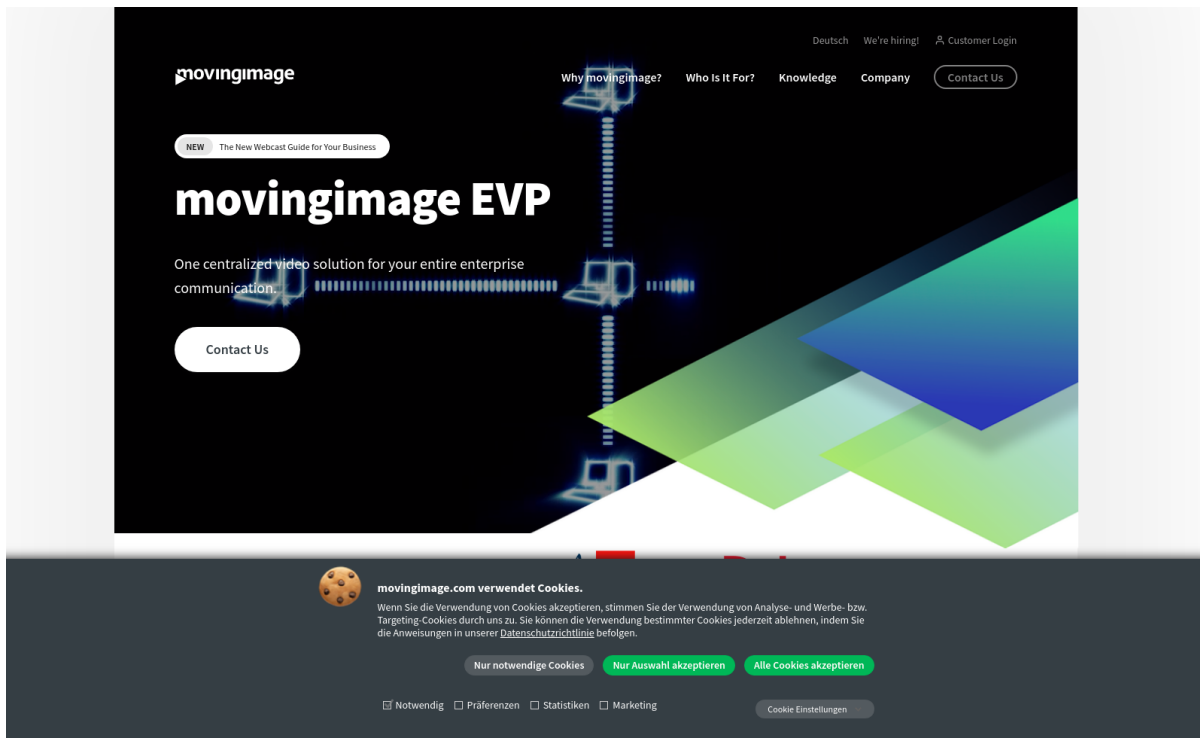
A content banner can look like this in HTML:

```
<!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 (more functions are available [here](#)):

- **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
// We wait until the content of the page was loaded
document.addEventListener("DOMContentLoaded", function () {
  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', '*');
  });
});

```