

Primis Video Discovery Player API

V1.0

29.05.2019



Overview

This article explains the basics of how to use the JavaScript API component of Primis Player.

The player object allows registering custom callbacks to events, that are triggered inside the video player, such as when video content starts playing or when user clicks on an ad.

Initializing The Player API

Embedding the player tag with an ID:

In order to activate the Primis API feature, it is necessary to add a unique player ID into the player embed tag code. This identifier also let you to differentiate between players, in case you have more than one on the same page.

Example:

```
<script type="text/javascript" language="javascript"
src="http://live.sekindo.com/live/liveView.php?s=87297&cbuster=[CACHE_BUSTER]&pubUrl=[PAGE_URL_ENCODED]&x=[WIDTH]&y=[HEIGHT]&playerApild=[MY_PLAYER_ID]"></script>
```

** The Primis video player API is not supported when embedding the tag inside a safe frame or non-friendly iframe.*

Get the Primis Player object when ready:

When the video player finishes loading, a javascript event `'primisPlayerInit'` is triggered by the `'dispatchEvent'` method on the `'window.top'` object.

This event has the Player object attached to it, that can be uniquely identified by its `'playerApild'` variable (as was defined in the embed tag).

This Player object can be referenced and stored in a local variable as shown below for later usage.

Example:

```
window.addEventListener('primisPlayerInit', function (e) {
  if (e.detail.playerApild === [MY_PLAYER_ID])
  {
    var primisPlayer = e.detail;
  }
});
```

Events

After the Primis Player was loaded on page, using the Player object - it is possible to register to events in order to get notifications when those events occur. It is also possible to unregister from events.

Register to Events

Without cbParams:

Using the player object, add a callback function that will be executed when the desired event is triggered. Notice that these types of events are triggered without any extra information. Therefore, the callback function doesn't need any parameters.

Example:

```
primisPlayer.addEventListener('videoStart', function () { //do something; });
```

With cbParams

Some events are triggered along with extra information that is sent with the event object. It is possible to reference the extra information through the callback function by adding a parameter to the function declaration.

Example:

The *'volumeChange'* event is triggered when the volume in the player is set to a new value. The new value is sent as a parameter to the registered callback function:

```
primisPlayer.addEventListener('volumeChange', function (newVal) { console.log(newVal); });
```

Unregister from Event

Unregister from event is possible by using the returned object from the `addEventListener` function as an argument to the `remove` function.

Example:


```
var resObj = primisPlayer.addEventListener('videoStart', function () { //do something; });
primisPlayer.removeEventListener(resObj);
```

Event Types:

The following event types can be registered by using the player object:

I. Ad Events

eventType	cbParams	Description
adStarted	N/A	Ad impression - a new ad has begun playing



adCompleted	N/A	Ad completed playing fully
adFirstQuartile	N/A	Ad reached its first duration quartile
adMidQuartile	N/A	Ad reached second quartile
adThirdQuartile	N/A	Ad reached third quartile
adClickthrough	N/A	User clicked through the ad
adPause	N/A	Ad was manually paused by user
adPlay	N/A	Ad was manually resumed by user
adSkip	N/A	Ad was manually skipped by user

II. Playlist Events

eventType	cbParams	Description
videoStart	N/A	A new content video was loaded and began playing
videoEnd	N/A	Content Video completed playing fully
videoSkip	N/A	Content Video skipped by user
videoClickthrough	N/A	User clicked through the video

III. Player Notifications Events

eventType	cbParams	Description
VolumeChange	<i>value</i> : float between 0.0 - 1.0	Volume changed to <i>value</i>



The Player Object Reference

This is the interface that is supported through the player object:

Functions:

(object) callbackId **addEventListener** (eventType, callback)

Register a callback to an event

- eventType (string): The type of the event to register to
- Callback (function): The callback function to invoke upon the occurrence of the eventType event
- Return value: Object, specifying the unique identifier of the callback in the event-callback system. It allows the user to remove that specific callback if desired, or **null** if failed

(void) **removeEventListener** (object)

Remove a registered callback with the given callback id

- (object) object: The object that was returned from the addEventListener method upon registering the callback to the event type
- Return value: N/A

Public Variables:

- (*int*) playerApild: The unique id of the player as was set in the embed tag by the user and uniquely identifies this player object
- (*String array*) allowedEvents: The names of the events that are supported by this player object

Full Code Example

A full test page example, which registers to 'videoStart' and 'volumeChange' events, and prints them to the test page when they arrive:

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>PLAYER API</title>
</head>
<body>
<script type="text/javascript">
  window.top.addEventListener('primisPlayerInit', function (e) {
    if (e.detail.playerApild === 123)
    {
      var player = e.detail;
      player.addEventListener('videoStart', function () {logEvent('videoStart');});
      player.addEventListener('volumeChange', function (val) {logEvent('volumeChange', val);});
    }
  });

  function logEvent(e, val)
  {
    document.getElementById('event_prints').innerHTML += 'Event: '+e;
    document.getElementById('event_prints').innerHTML += val !== undefined?'['+val+']':'';
    document.getElementById('event_prints').innerHTML += '<br />';
  }

</script>
<!-- code from primis -->
<script type="text/javascript" language="javascript"
src="http://live.sekindo.com/live/liveView.php?s=99445&cbuster=98765&pubUrl=example.com&x=300&y=250&vp_template=5579&playerApild=123"></script>
<!-- code from primis -->

<p id="event_prints"></p>
</body>
</html>
```