

PLAYDECK API

The PLAYDECK API is a construct to allow anyone to interface with PLAYDECK. This article will show how to get started developing with the API:

In this article:

- Introduction
 - Examples
 - Implement the API
-

Introduction

The PLAYDECK API is bi-directional: You receive the Playout Status, but you can also send Commands to PLAYDECK.

The API is based on WebSockets: They are a part of JavaScript, so you can develop in JavaScript, or NodeJS, or TypeScript. The WebSocket Port is fixed to 11411 and is always enabled in PLAYDECK.

We decided to go with WebSockets, because:

- It supports both directions: From and To PLAYDECK
- No Installation required: Use any Text-Editor
- No specific Network Security required
- Easy to build own User Interface or Animation with HTML

What can be send to PLAYDECK?

- CUE or play Blocks or Clips, Overlays or Actions
- Start/Stop video assets like Streams

What can be received from PLAYDECK?

- Events (e.g. a Block or Clip has started)
 - Playout Status (e.g. What is playing and the remaining Block Time)
 - Content (e.g. complete Playlist with Blocks and Clips)
-

Examples

1) Companion

PLAYDECK API is being used by bitfocus Companion in the PLAYDECK module. This gives you a quick touch+feel about what the API is capable of:

<https://playdeck.tv/howto/companion/>

2) Director View

The API is also used by the Director View, which is part of PLAYDECK. You can find the HTML Templates here:

c:\Users\Public\Documents\JoyEventMedia\Playdeck\HTML-Templates\Director-View

3) HTML Overlays

The API is also used by some HTML Overview Examples, which you can find here:

c:\Users\Public\Documents\JoyEventMedia\Playdeck\HTML-Templates\Overlay-Templates

Implement the API

We prepared some examples for you to quickly review a working script. Please check them in order, as they build up on each other.

Tip 1: You can double click any HTML File to open in the Browser to start the Script/API

Tip 2: You can edit the File "WebSocketSDK\Version.js" to change the IP Address

1) Native Connection (Without the API)

Let's start with something very minimalistic:

c:\Users\Public\Documents\JoyEventMedia\Playdeck\HTML-Templates\Automation-And-UI\SampleMinimalistic.html

This will open the WebSocket to PLAYDECK with native JavaScript Code and send a Command to PLAY the first Clip in the first Block.

Another Example is this, where you can click Buttons in HTML to start Playout in PLAYDECK:

c:\Users\Public\Documents\JoyEventMedia\Playdeck\HTML-Templates\Automation-And-UI\SampleUserInterface.html

2) Minimalistic Connection (WITH API)

This example will introduce the API:

c:\Users\Public\Documents\JoyEventMedia\Playdeck\HTML-Templates\Automation-And-UI\SampleMinimalisticSDK.html

It uses the Subfolder "WebSocketAPI" and includes its MAIN SCRIPT

“AppInterface.js”, which is kind of the “Loader” for the whole API.

The API will now take care of your WebSocket Connection to PLAYDECK, so you don’t have to code this manually. The API also exposes a Template Framework to JavaScript. The Template name is derived from the Filename, so it needs to match. There are several pre-defined functions available. For this example we only use “Start”: This function will be called by the API, once the WebSocket Connection to PLAYDECK has been established successfully. In this example it will PLAY the first Clip of Channel 1.

3) More Framework Functions

Besides “Start” there are many more Framework Functions available:

c:\Users\Public\Documents\JoyEventMedia\Playdeck\HTML-Templates\Automation-And-UI\SamplePlayRandomClip.html

Please see the comments in the JS File for all available Functions.

If you open the Browser Console (typically with F12) you can see the Feedback of the “console.log” instructions.

4) Parse Project Data

To access all Clips the function “DataProject” is available and this example show how to enumerate the Project Data:

c:\Users\Public\Documents\JoyEventMedia\Playdeck\HTML-Templates\Automation-And-UI\SampleShowPlaylist.html

5) Custom Schedule Sample

The last example shows a full working and complete Script for a Custom Scheduling outside of PLAYDECK:

c:\Users\Public\Documents\JoyEventMedia\Playdeck\HTML-Templates\Automation-And-UI\SampleWeeklySchedule.html