Web API example

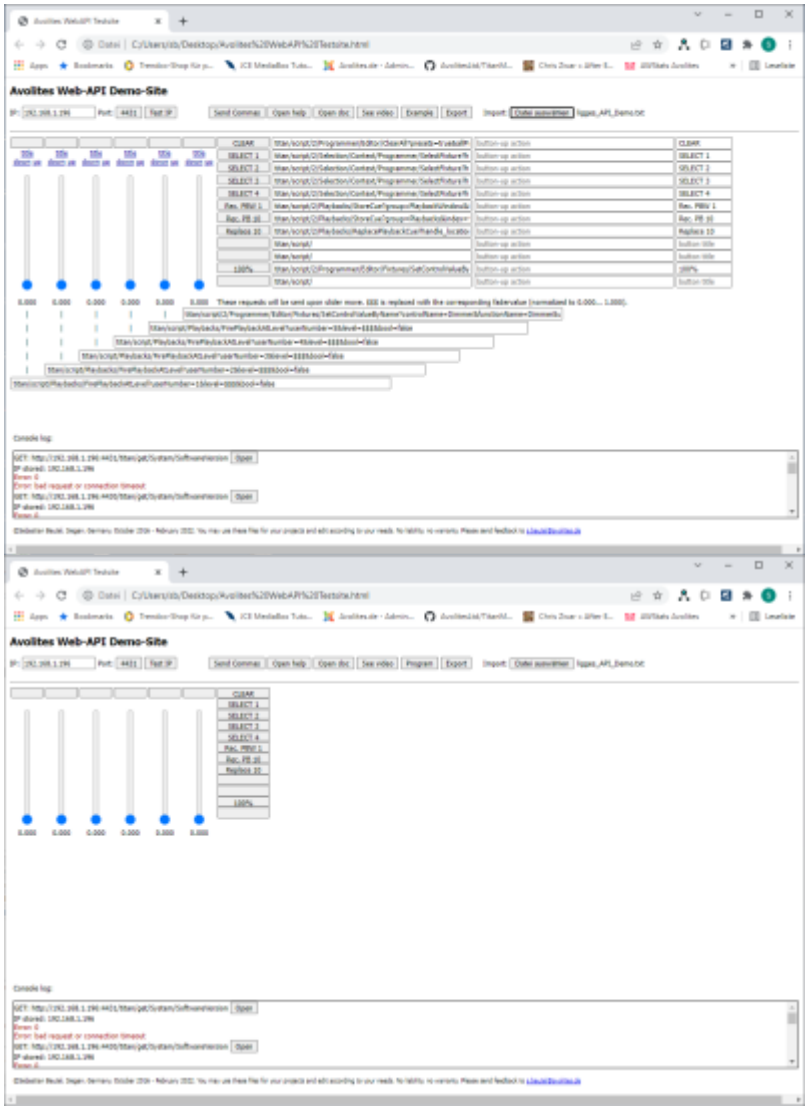
Programming playbacks

Starting point to programm playbacks using the Web API. Inspired by a project where the customer shall be able to program simple cues. Also this might be the starting point for more sophisticated applications.

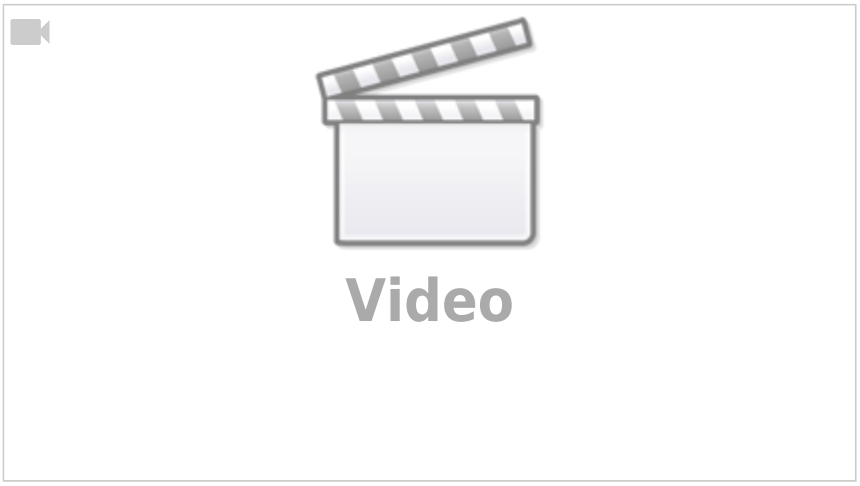
Also this solves the mystery about the 'separate programmer'.

author, date	Sebastian Beutel, February 2022
published	here, Youtube
prog. lang./ framework	simple text editor (I used the Web API Demo to play with the requests)
description	collection of http requests used to program simple cues.
other requirements	This is essentially just a collection of requests. You can send them with a web browser or with other tools.

Screenshots



Video



Separate Programmer

By default the Web API is available on network port 4430 (hence the requests start with something like 'http://localhost:4430/...'. Also, actions on the user interface, e.g. changing pages or selecting fixtures, do not reflect in the Web API. In turn anything done in the API doesn't reflect in the normal user interface, e.g. if you select fixtures or set a level you won't see it in Titan. This is pretty much like the blind mode or another user. Unfortunately this makes it really hard to debug any API requests as you see the outcome only when everything is correct and you did succeed programming something.

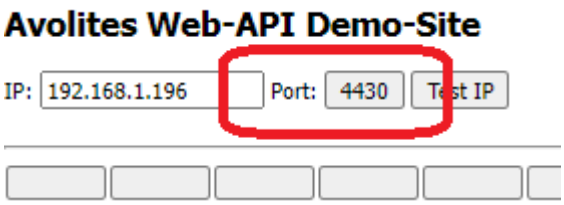
However there is a solution: you can tell Titan to also listen on another port, with the user and interface you see directly in Titan. In order to do this you need to set specific parameters:

- open the .exe.config file of the Titan application you are using (e.g. `C:\Program Files\Avolites\Titan Mobile\Mobile.exe.config`)
- find the `<appSettings>` part
- add these lines:

```
  <add key="webapi.enabled" value="true" />
  <add key="webapi.port" value="4431" />
```

- maybe these lines already exist but you need to change `false` to `true`
- restart the Titan software. Now, the Web API is available on both, port 4430 with the traditional separate user, and port 4431 with the same user and interface you see in Titan.

In order to make it easier to work with these settings the WebAPI test page now also features a port toggle button where you can switch between port 4430 and 4431:



API Requests

The intention of this example is to make it possible to program simple playbacks using the API. Hence the actions and requests are as follows:

- **Clear** the programmer
- **Select** one or more fixtures
- **Set a dimmer level**, or more generally, set a certain level of an attribute of the selected fixtures
- **Record** the contents of the programmer to a certain playback handle e.g. on a playback fader or a buttom in the playbacks window
- **Clear** the programmer again

Of course this can be extended to some degree, e.g. set another attribute (if the fixtures have more than just a dimmer channel), record-merge or record-replace if the handle is already in use, delete the playback etc. Please also note the different ways of spedifying a certain item (i.e. the location): some commands refer to the current page of playbacks or of another window while others can work on any page, some commands use the userNumber while others use the location, some indexes are 0-based while others are 1-based… I strongly suggest reading the macro article on identifiers as this is closely related.

- Programmer/Editor/ClearAll
- Selection_Context_Programmer_SelectFixture
- Programmer_Editor_Fixtures_SetControlValueByName
- Playbacks_StoreCue
- Playbacks_ReplacePlaybackCue

From:
https://avosupport.de/wiki/ - **AVOSUPPORT**

Permanent link:
https://avosupport.de/wiki/webapi/examples/programming_playbacks?rev=1645111918

Last update: **2022/02/17 15:31**