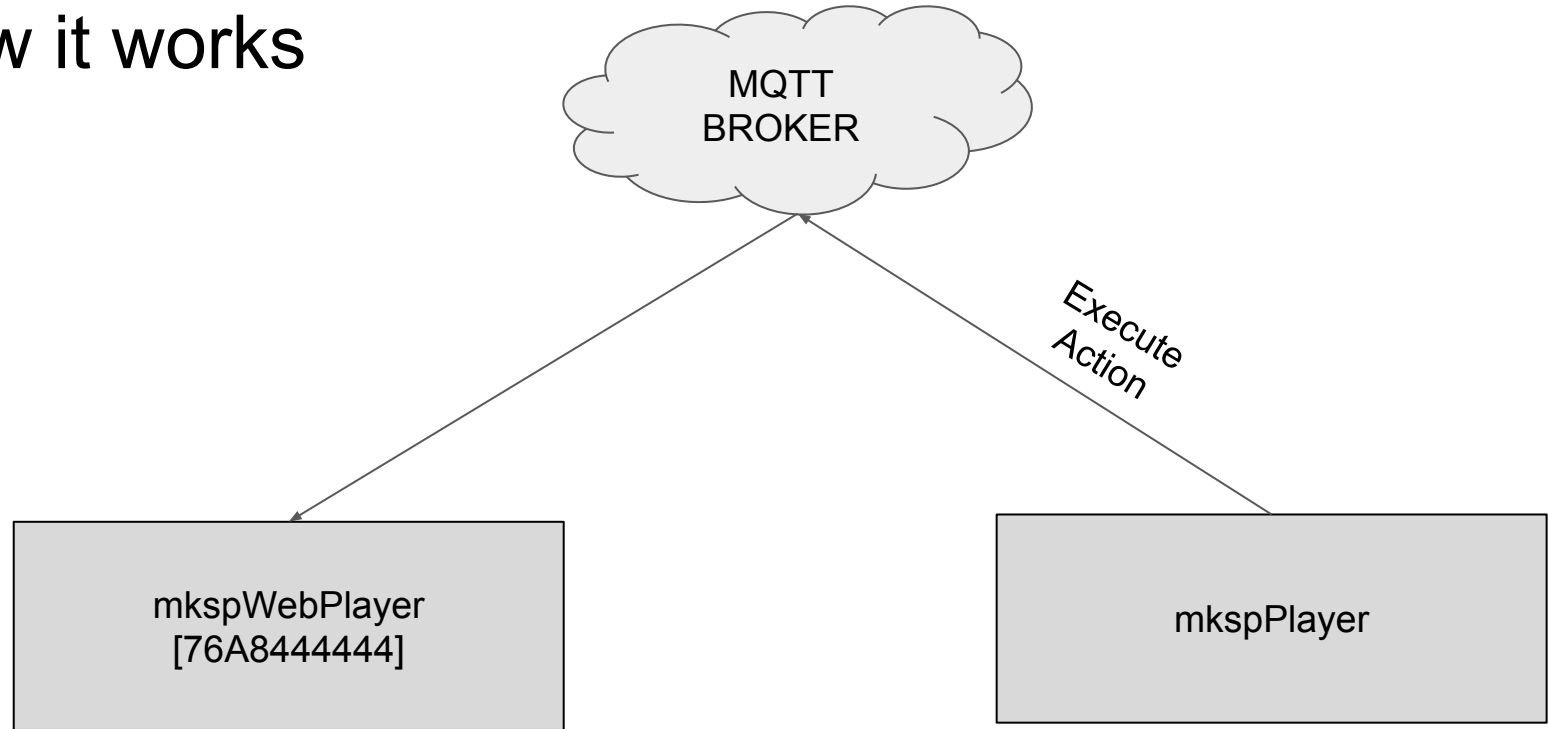# mkspWebPlayer

Web App viewer for mesch.io platform

How it works

URL WebPlayer

Simply open the following URL from a browser

- Use Safari on iPad, iPhone, MAC
- Use Chrome for Android, Windows and Linux

http://cdn.phygo.io/mkspWebPlayer/

The Viewer

meSch Player 1.3 [76A848FF7F11]

Start   Configure   Simulator

Viewer ID

meSch Appliances

viewnode - Tablet - Tablet (CBE47E9967F9)

Viewer ID

3D model   Film   Audio   Image   Text

Edit appliance: ⓘ

Label
Tablet

Title:
Tablet

Type:
view node ▾

Viewnode Id
CBE47E9967F9

Change the viewer
ID with yours and
save

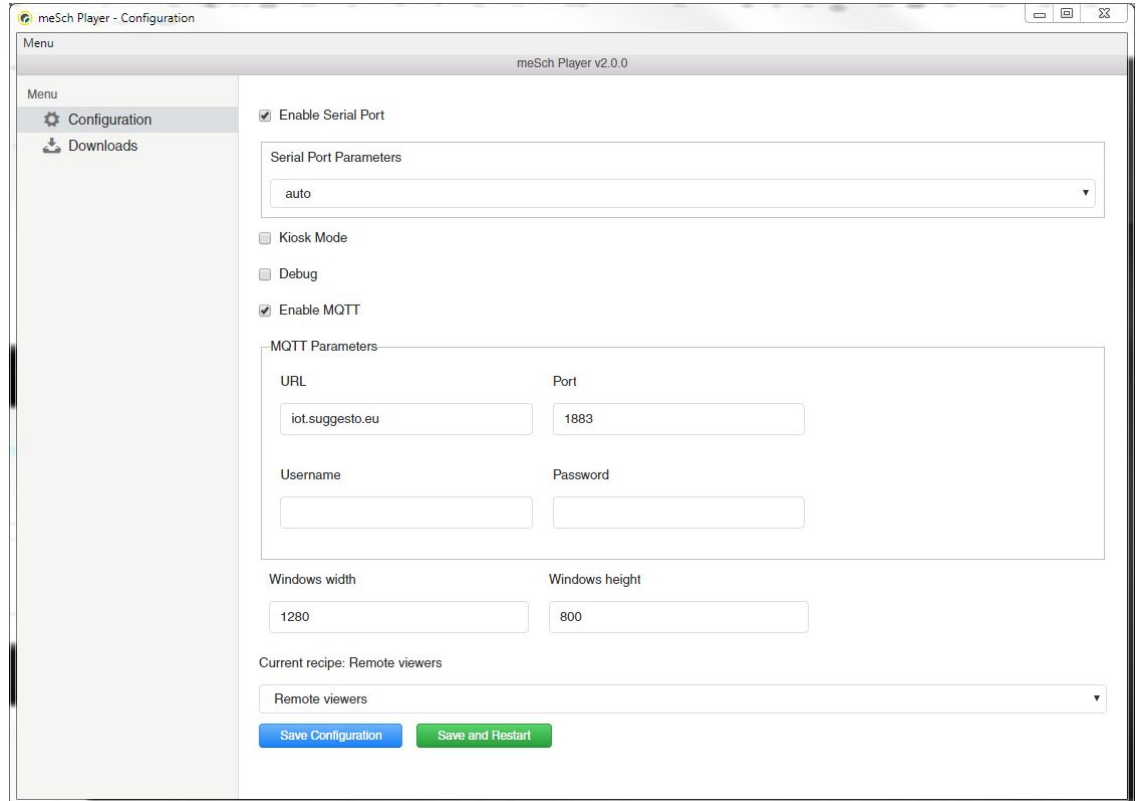Parent appliance
None ▾

💾 SAVE   💾 CLOSE

Enable MQTT on the mkspPlayer

Go to **Configuration** page

Select
Enable MQTT

URL: iot.suggesto.eu
Port: 1883

with empty username and
password

The Viewer - Start

meSch Player 1.3 [76A848FF7F11]

Start   Configure   Simulator
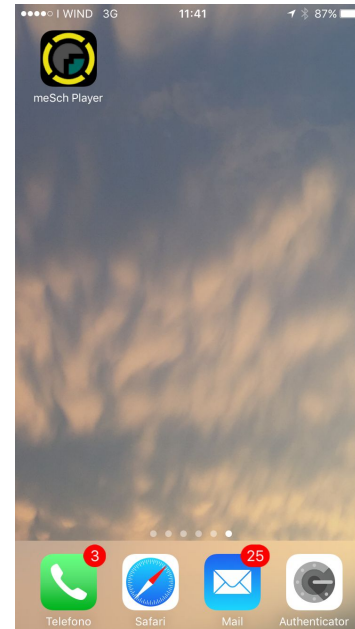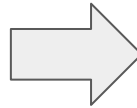
Press Start to activate
the viewer

iPad/iPhone

With iPad/iPhone you can add the webapp to your home screen, from Safari Web Browser use **Add to Home.**
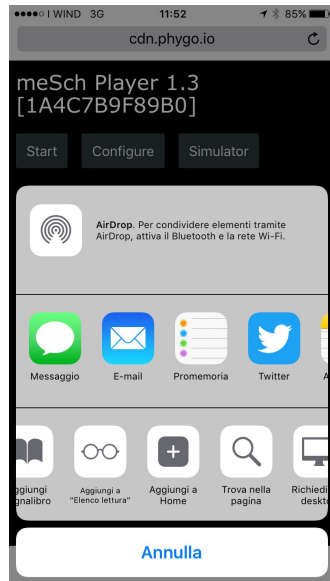
Configuration

You can change the default configuration selecting the **configure** button.

The **Viewer ID** must be unique among all viewers.

If you change the MQTT broker **URL**, you have to change also the **Main Player** one.
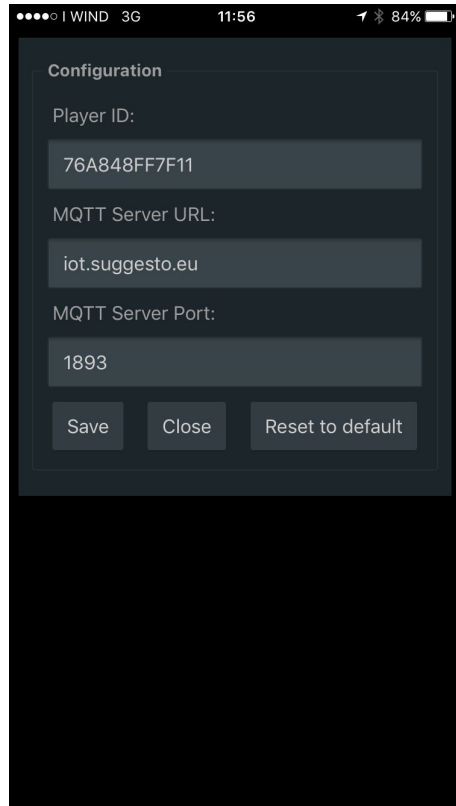
Simulator (alfa)

If you select **Simulator**, a page with a list of buttons, and **RFIDs** comes up.

If you click on one **button (1..5)** or **ENTER/LEAVE** buttons of an RFID, a message is sent via MQTT.

The Main Player can receive those messages.
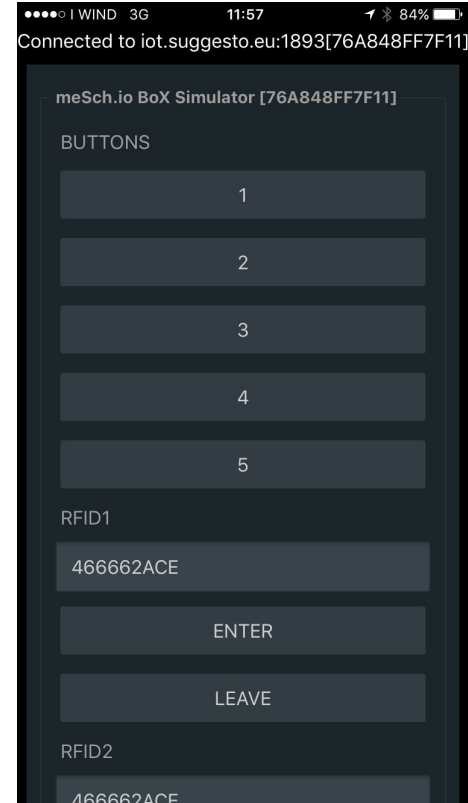
Script commands

To send or receive Events over MQTT, first you
need to enable the use of MQTT inside the
mkspPlayer2.

```
// IMAGE
api.sendEvent(JSON.stringify({ "command": "showImage", "url": content.url }), VIEWER_ID);


// VIDEO
api.sendEvent(JSON.stringify({ "command": "playVideo", "url": content.url }), VIEWER_ID);


// SOUND OR GENERIC URL
api.sendEvent(JSON.stringify({ "command": "openUrl", "url": content.url }), VIEWER_ID);


// HTML TEXT
api.sendEvent(JSON.stringify({ "command": "showText", "text": content.text }), VIEWER_ID);
```

Local MQTT Broker (mosca)

In order to install and use mosca mqtt broker npm and node components must be already installed.

```
npm install mosca bunyan -g
mosca -v --http-port 1893 --http-bundle --http-static ./ | bunyan
```

In this example you have the MQTT server running with this configuration:

```
mqtt: 1883  (Use this with the mkspPlayer)
http: 1893  (Use this with the viewer)
```

more info:

https://github.com/mcollina/mosca/wiki/MQTT-over-Websockets

Local MQTT Broker (mosquitto) https://mosquitto.org/

Installing Mosquitto MQTT on a MAC

```
brew install mosquitto
```

In order to use MQTT Over websocket you have to change the configuration (in my mac is like):

```
/usr/local/Cellar/mosquitto/1.4.14_2/etc/mosquitto/mosquitto.conf
```

Insert at the end of the `mosquitto.conf` file the following lines:

```
listener 1883
protocol mqtt
listener 1893
protocol websockets
```

and restart the mqtt brocker with:

```
brew services restart mosquitto
```

Local MQTT Broker (mosquitto) - 2

Now you can test the installation and ensure the server is running successfully.

Open a new command window and start a listener.

```
mosquitto_sub -t topic/state
```

In another window, send a message to the listener.

```
mosquitto_pub -t topic/state -m "Hello World"
```

In a first window you should see the "Hello World" message coming.

more info: https://simplifiedthinking.co.uk/2015/10/03/install-mqtt-server/