JavaScript library for Domoticz and MQTT

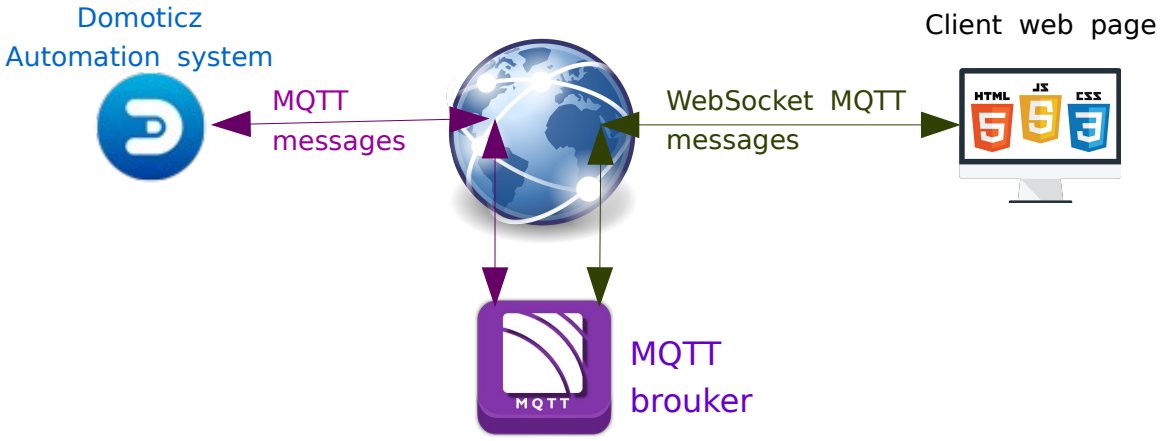
Description

This library is used for create professional live visualization smart home or regulation scheme with quick response to control entities. This library take care for can use vector graphics ,vector animations and live color gradients. For vector images and animates is used SVG images, who can be control by javascript.

Request:

Domoticz automation system
MQTT server with websocket (best is mossquito)
Web browser with support svg animation (firefox or chrome, edge not now)

Principle:



- This principle enable control your automation behind NAT if you use public MQTT server.
- And enable quick response into user action (more quick than original domoticz GUI).
- This is utility for create more nice dashboards for your projects.
- Use only html, XML, and javascript code, web page can be stored localy on client, no need web server (only MQTT server as mediator beatween domoticz and javascript on client web browser.

For clearly source code is split into more files. Every file care of one task. Therefore in header of html file must be those files includet. E.t.:

```
<script src="config.js"></script>
<script src="js/lib_auxliary_functions.js"></script>
<script src="js/mqttws31.js"></script>
<script src="js/lib_base_mqtt.js"></script>
<script src="js/lib_load_svg.js"></script>
<script src="js/lib_set_points.js"></script>
<script src="js/lib_switchs.js"></script>
<script src="js/lib_temperatures.js"></script>
<script src="js/lib_get_position.js"></script>
<script src="js/domMQTTsvg/lib_clocks.js"></script>
```

auxiliary function for easy position graphic element into web page

additional function for easy draw analogue clocks

Some files depends of other, therefore orders is important.

Config file must create every own. This file contain connection information and entities (devices) setting. For relation between html element and domoticz device must exist record in this config file. Html elemetns is marked by his *id attribute* and domoticz device by his *idx number*.

Example config .js :

```
// Create a client instance
var connection = {
        //host: "10.10.100.166",
        host: "localhost",
        port: 8083,
        options: {
  useSSL: false,
        }
}

// define switches devices
var switchs=[
  {
    "id": "cerpadlo1",
    "idx": 1,
    "status": 0,
    "image_on": "img/pump_wilo_ON.svg",
    "image_off": "img/pump_wilo_OFF.svg",
  }];

// define temperature devices
var temperatures=[];

//define setPoints or Svalue devices
var setPoints=[];

// for visualise clcok
var clocks=[];

// define on/off line image
var linestatus=
  {
        "name": "off-on-line",
    "image_on_line": "img/online.png",
    "image_off_line": "img/offline.png",
  };
```

Note: if etc. “js/lib_switchs.js” is included then var switchs=[] array may by empty, but must be exist.

Item “id” - contain value of id attribute used in html for this device.
Item “idx” - contain id used in domoticz system for this device
ether items will be described bellow.

DESCRIOTION OF CONFIG SECTIONS:

CONNECTION:

no need more description only connect information into MQTT server.

```
var connection = {
        host: "localhost", // or IP address
        port: 8083,
        options: {
   useSSL: false,
        }
 }
```

SWITCHS:

this section create active switch by define image of off and on status of switch or only pasive view status svitch. Now is supported tree types of switch confiurations:
image type:
create active switch by ON/OFF images.
Example in config file:

```
  {
    "id": "pump1",       // value of id attribute used in html for this device.
    "idx": 1,            // identification number of device in  domoticz system  (first column in device table                                                                    on domotic device list)
    "type": "image",     // specify type switch
    "status": 0,         // in this switch store last status
    "image_on": "img/pump_wilo_ON.svg", // image of ON status
    "image_off": "img/pump_wilo_OFF.svg", // image of OFF status
  },
```

Example in inside html file:

```
<img src=”default_image_for_switch.svg” id=”pump1”>
```

style type:
only change visage of html element by change CSS atribute.
Example in config file:

```
  {
    "id": "path8903", // value of id attribute used in html for this device.
    "idx": 1,              // identification number of device in  domoticz system  (first column in device table                                                                    on domotic device list)
    "type": "style",       // specify type switch
    "style": "stroke-dasharray", // CSS attribute who will be changed
    "value_on": 20, // value of CSS atribute for ON status
    "value_off": 0, //  value of CSS atribute for OFF status
    "status": null, // in this switch store last status
  },
```

Example in inside svg file (only id is important):

```
<path
     style="stroke:#ff0018;stroke-width:11.5;stroke-dasharray:0;"
     d="m 1187.1429,1982.3622 0,-412.8572 -268.57147,0 0,144.2858 -48.57143,0 0,-1.4286"
     id="path8903"
     inkscape:connector-curvature="0">
</path>
```

animate stroke type
only for animate switch status. Nice for animate water flow in tube.
Example in config file:

```
  {
    "id": "path8922",          // value of id attribute used in html for this device.
    "idx": 1,                  // identification number of device in  domoticz system  (first column in                                                                device table on domotic device list)
    "type": "animate_stroke",  // specify type switch
    "from": 80,                // animate from position
    "to": 0,                   // animato to position
    "dur": "1s",               // duration for animation
    "status": null,            // in this switch store last status
  },
```

Example in inside svg file (only id is important):

```
<path
     style="stroke:#ff0018;stroke-width:11.5;stroke-dasharray:0;"
     d="m 1187.1429,1982.3622 0,-412.8572 -268.57147,0 0,144.2858 -48.57143,0 0,-1.4286"
     id="path8903"
     inkscape:connector-curvature="0">
```

```
</path>
```

TEMERATURES:

this section is definitions of representaion temperature. Curently is support four types representions of temerature. One domoticz temperature can have more configuration options (one chage color, next text value,..)

color type

generate RGB color from temperature and change CSS attribute defined in “style” config parameter on html tag. This also can be used for change color inside gradient.

Example in config file:

```
  {
    "id": "stop4107", // value of id attribute used in html tag who representate temperature
    "idx": 4,                 // identification number of device in  domoticz system  (first column in
                                                                    device table on domotic device list)
    "type": "color",          // specify type represention of temperature
    "status": null,           //  in this temperature store last status
    "style": "stop-color",  // css atributeName for change color
  },
```

Example in inside html or svg file:

```
<linearGradient id="linearGradient4105">
    <stop style="stop-color: rgb(160, 0, 95); stop-opacity: 1;" offset="0" id="stop4107"></stop>
    <stop style="stop-color:#ff816e;stop-opacity:0.99770093;" offset="1" id="stop4109"></stop>
</linearGradient>
```

text type

This type of temperature representaion chage text to actual temperature in html tag.

Example in config file:

```
  {
    "id": "temp1",          // value of id attribute used in html tag who representate temperature
    "idx": 4,               // identification number of device in  domoticz system
    "type": "text",         // specify type presentation of temperature
    "status": null,         //  in this temperature store last status
  },
```

Example inside html file:

```
<div>
        <span id="temp1">--</span> C
</div>
```

“value %” or “value 100-%” type

change percentile value CSS attribute defined in “style” config parameter on html tag. Usefully for change width or height CSS attribute.

Example in config file:

```
{
  "id": "temp2",
  "idx": 4,
  "type": "value 100-%",
  "status": null,
  "style": "width",
  "min": -20,
  "max": 150,
},
```

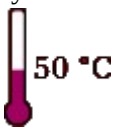
Example inside html file:

```
<div style="background-color: #F5FA0A;position: relative;width: 100px;">
        <div id="temp2" style="background-color: #F50A0A;position: relative;width: 100%;"> </div>
</div>
```

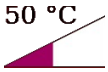
create type

Currently library has predefined two own styles of temperature images. Every by temperature change text, color, and size of temperature column. In html source create temperature only by tag .

style1:



style2:



Example in config file:

```
{
  "id": "temp2",              // value of id attribute used in html tag who representate temperature
  "idx": 4,                   // identification number of device in  domoticz system
  "type": "create",           // specify type presentation of temperature
  "status": null,             //  in this temperature store last status
  "styleX": "style2",         // style of temperature image (see above), default is “style1”
  "min": -20,                 // down range (default -20)
  "max": 125,                       // up range (default +90)
},
```

Example inside html file:

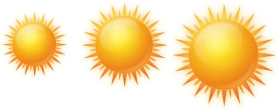
```
        <div id="temp2" ></div>
```

SET POINTS:

this section is definitions of set points (sliders) for thermostat,(or every device who in domoticz store value into sValue, etc. temperature, counters, barometers,..) . Currently is support six types representations of set_points. Frist for types is some as temperature types only show current value (by text, color, or percent by CSS). Only six type “created” create slider who can change set_point value.

“color”, “text” ,“value %” or “value 100-%” type
is some as temperature setting

value type



can change any html or CSS attribute (etc. hight, width, color, left, top,..)

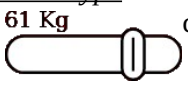
Example in config file:

```
{
  "id": "sun",              // value of id attribute used in html tag who representate set point
  "idx": 8,                 // identification number of device in  domoticz system
  "status": null,           //  in place for store last value
  "type": "value",          // specify “value” type presentation
  "attr": "height",         // specify HTML attribute will be change on base actual value
                            // for CSS attribute use "style" instead “attr”
                            // e.g.  "style": "stroke-dasharray"
  "min": 0,                 // min., max. define range of valid values from domoticz
  "max": 1100,
  "from": 0,                // from, to define correspond range of values used in html entry
  "to": 300,
  "unit": "px"
 },
```

Example inside html file:

```
<img id="sun" src="img/Sun_animation.svg"  height="200" width="210">
```

create type



create slider who can change set point value.

Example in config file:

```
{
  "id": "setPoint1",        // value of id attribute used in html tag who representate set point
  "idx": 7,                 // identification number of device in  domoticz system
  "type": "create",         // specify type presentation
  "status": null,           //  place for store last value
  "styleX": "style1",       // style of set point , default is “style1” and currently is available only
```

```
                                        “style1”
  "min": -30,                   // down range (default -20)
  "max": 90,                    // up range (default +90)
  "unit": "Kg",                 // unit
 },
```

Example inside html file:

```
        <div id="setPoint1" ></div>
```

ADITIONAL FEATURES:

CLOCKS:

this section is definite javascript clocks. I use this for easy visualise time in my project with silumate home heating with accelerated time. You can create clock with virtual time getting from domoticz text device or create clock with current time.

Example in config file:

```
var clocks=
 [{
                "id": "clock",              // value of id attribute used in html tag
                "idx": 7,           // identification number of device in  domoticz system
                                    // if you use "realTime": "yes" no need
                                    // accessible format  from domotic is:
                                    // hh:mm:ss or hh:mm or hh e.g.: 23:15:00
                "styleX": "style2",         // currently is viable only two style - style1 and style2
                "type": "create",// type must be always “cereate”
                "realTime": "no",           // “no” - use value from domoticz (from device by idx)
                                    // “yes” - show current client time
 }];
```

Example inside html file:

```
<div id="clock"></div>
```

AUTO LOAD SVG IMAGES INTO HTML CODE:

This enable easy use javascript for manipulation graphics elements inside SVG image and keep clearly html source code. Only add type=”svg” and src=”pat/to/svg/file” into html tag “ ”.
Example of usage in HTML source code:

```
<div id="svg1" type="svg" src="img/image.svg" > </div>
```

GET EASY COORDINATION FOR PUTS GRAPHICS ELEMENT INTO MAIN SCHEME IMAGE:

Into html source insert graphics elemts (pumps, valves, temperatures,...) into own html tag “<div id="Ac" setLocation='yes' style='position: absolute;”> graphics element ”.

setLocation='yes' - enable mouse move graphics element in page with view actual location coordination, who can be next copy into html source.

For easy fix location of graphics elements into main scheme image use next gist:

```
<div style='position: relative;'>
        <div id="main_svg" type="svg" src="img/main_image_scheme.svg" ></div>

        <!-- next put gpraphics element, after you get right location delete setLocation='yes' -->

        <!-- insert image -->
        <div id="1" setLocation='yes' style='position: absolute; left: 468px; top:478'>
                <img src="img/pump1.svg" id="pump1" height="95" width="65">
        </div>
        <!-- insert temperature (type=create) -->
        <div id="2" setLocation='yes' style='position: absolute; left: 1005px; top: 974px;'>
                <div id="temp1" > </div>
        </div>
        <!-- insert set point (type=create) -->
        <div id="3" setLocation='yes' style='position: absolute; left: 41px; top: 1175px;'>
                <div id="setPoint1" > </div>
        </div>
</div>
```

Více informací:

- mosquitto (MQTT server): https://mosquitto.org/
- domoticz: https://domoticz.com/
- web seahu: http://www.seahu.cz

Write by:

Ing. Ondřej Lyčka 5-2018

Document version: 1.01 (drawt)