

Elpriser

Denne QuickApp bruges til at visualisere elprisen, med oplysninger om nuværende pris, sammenlignet med i dag og i morgen

Den åbner for følgende muligheder

- Se prisen for de 48 timer (i dag og i morgen).
- Den regner min, max og avg for de 48 timer og for sidste uge.
- Den regner en procent for aktuelle pris, sammenlignet med de 48 timer
- Den regner en procent for aktuelle pris, sammenlignet med sidste uge
- Man kan bruge disse oplysninger i sine scener
- Lægger man Lua scenen ind, kan man få en liste af taster, til at lyse afhængig af prisen sammenlignet med de 48 timer
kraftig grøn til lys grøn = 0% til 30%, lys gul til kraftig gul = 31% til 70% og lys rød til kraftig rød = 71% til 100%
- Man kan få en liste af taster, til at lyse afhængig af prisen sammenlignet med sidste uge.
- Man kan få 3 taster til at lyse afhængig af prisen om 1, 2 og 3 time, sammenlignet med de 48 timer

Installation

Indlæs QuickApp, Elpriser.

Opret en Lua-scene og tilføj indhold fra denne manual

Variable: Konfiguration

!! Hvis ikke QuickAppen viser de rigtige oplysninger, kan du prøve følgende

Klik på getData.

Klik Rediger/Edit, klik i koden i højre side, klik CTRL+S, klik på Luk/Close

EnergDataService giver elspot pris for dagen



QuickApp:
sw:1.0.7
Netværk:
Status: **Forbundet**

Sidst opdateret: **den 18. September 2022 01:02:02**

Samlet pris: **1.83 kr/KW**

El pris: **0.2 kr/KW**
Afgift: **0.97 kr/KW**
Transport: **0.3 kr/KW**
Moms: **25%** **0.36 kr/KW**

Dags type: **4.0%**
Uge type: **1.0%**

Energipriser (per Mega Watt)

PA	Min 48t	Max 48t	Gen 48t	Min Uge	Max Uge
DK2-e1 +11 pr1s/MW	1793.77	3583.11	2317.0	1796.6	6121.91
DK2-e1 DK1/MW	165.02	1596.49	583.6	162.48	3627.53
DK2-e1 EUR/MW	22.19	214.68	78.47	21.85	487.82

Get data

Lua-Scene Elpriser

Du kan lave en scene, som gør det let, at knytte Matrix taster til visning af elpriser.

Den venstre del af scenen er kriteriet for, hvornår scenen skal kaldes.

Denne scene tjekker prisen hver 5 minut

```
{
  conditions = { {
    isTrigger = true,
    operator = "matchInterval",
    property = "cron",
    type = "date",
    value = {
      date = { "00", "12", "22", "5", "*", "2022" },
      interval = 300
    }
  } },
  operator = "all"
}
```

Den højre del af scenen er den der vælger hvilke taster, som skal styre og hvilken enhed der skal styres.

I følgende eksempel er de to Matrix'er lagt ind, til styring af 3 Hue enheder

```
-- Angiv en eller flere taster, som skal vise farven for prisen
local dayIds = {} -- taster, som viser prisen sammenlignet med i dag
local dayPIds = {{},{},{}} -- taster, som viser prisen +1, +2 og +3 timer
local weekIds = {} -- taster, som viser prisen sammenlignet med sidste uge
local priceIds = {} -- taster, som viser prisen i forhold til Low og High

-- skal ikke ændres
local qa = (api.get("/devices?name=Elpriser") or {})[1];
local qaId = qa.id or 0;
if (qaId == nil or qaId == 0) then return; end

function fibaro.getQAVariable(id,name)
  local props = (api.get("/devices/"..(id or MID)) or {}).properties or {}
  for _, v in ipairs(props.quickAppVariables or {}) do
    if v.name==name then return v.value end
  end
end
```

```

    end
end

function fibaro.setQAVariable(id,name,value)
    return fibaro.call(id,"setVariable",name,value)
end

local cDkk = fibaro.getQAVariable(qaId,"currentPrice");
local cDkkPct = fibaro.getQAVariable(qaId,"currentPricePct");
local dType = fibaro.getQAVariable(qaId,"currentTypeDay");
local cDkkP1 = fibaro.getQAVariable(qaId,"currentPriceP1");
local dTypeP1 = fibaro.getQAVariable(qaId,"currentTypeP1Day");
local cDkkP2 = fibaro.getQAVariable(qaId,"currentPriceP2");
local dTypeP2 = fibaro.getQAVariable(qaId,"currentTypeP2Day");
local cDkkP3 = fibaro.getQAVariable(qaId,"currentPriceP3");
local dTypeP3 = fibaro.getQAVariable(qaId,"currentTypeP3Day");
local wType = fibaro.getQAVariable(qaId,"currentTypeWeek");

function calcRandG(_type)
    local r = 0;
    local g = 0;
    local colType = 64;
    if (_type >= 70) then
        g = 0;
        r = (192 + (63 * ((_type - 70) / 30)));
    elseif (_type < 0) then
        colType = 138;
        g = 64;
        r = 64;
    elseif (_type <= 30) then
        g = (192 + (63 * ((30 - _type) / 30)));
        r = 0;
    else
        g = (192 - (192 * ((_type - 30) / 40)));
        r = (192 - (192 * ((40 - (_type - 30)) / 40)));
    end
    return math.floor(r),math.floor(g);
end

function showColOnButton(_dkk,_Ids,_Type)
    print("Elpriser: _ids:" .. json.encode(_Ids) .. ", _Type:" .. (_Type or 'nil') ..
        ", _dkk:" .. (_dkk or 'nil'));
    if (type(_Ids) == "number") then _Ids = {_Ids}; end
    if (_Ids ~= nil and #_Ids > 0 and _Type ~= nil) then

```

```

    local r,g = calcRandG(_Type);
    print("_Type r,g:" .. r .. ", " .. g)
    hub.call(_Ids, 'setColor', r, g, 0, 64);
end
end

print("Elpriser: dkk:" .. cDkk .. ", dtype:" .. dtype .. ", wType:" .. wType);
showColOnButton(cDkk,dayIds,dType);
showColOnButton(cDkkP1,dayPIds[1],dTypeP1);
showColOnButton(cDkkP2,dayPIds[2],dTypeP2);
showColOnButton(cDkkP3,dayPIds[3],dTypeP3);
showColOnButton(cDkk,weekIds,wType);
showColOnButton(cDkkPct,priceIds,dType);

```

Den første 3 linje skal rettes, så id'erne for tasterne er angivet

```

local dayIds = {}           -- taster, som viser prisen sammenlignet med i dag
local dayPIds = {{},{},{}} -- taster, som viser prisen +1, +2 og +3 timer
local weekIds = {}        -- taster, som viser prisen sammenlignet med sidste uge

```

```
dayIds = {65}
```

Dette kunne være en opsætning, hvor 65 er id på en tast, der ønskes at lyse med aktuelle pris, sammenlignet med de 48 timer.

```
dayPIds = { 67, 69, 71}
```

Dette ville få tast med id 67 til at lyse med prisen om en time, sammenlignet med de 48 timer.
Tast med id 69 lyser med prisen om to time, sammenlignet med de 48 timer.
Tast med id 71 lyser med prisen om tre time, sammenlignet med de 48 timer.

```
weekIds = {77,81}
```

Dette ville være en opsætning, hvor 77 og 81 er id'er på taster, der ønskes at lyse med aktuelle pris, sammenlignet med sidste uge.