Tvorba prototypov pre monitorovanie a správu sieťových zariadení využitím nástroja Node-RED

Obsahom tohto dokumentu je ukážka práce s nástrojom Node-RED. Dôraz bude kladený na jeho možnosti tvorby prototypov v kontexte správy monitorovania a riadenia sieťových zariadení.

Inštalácia nástroja Node-RED

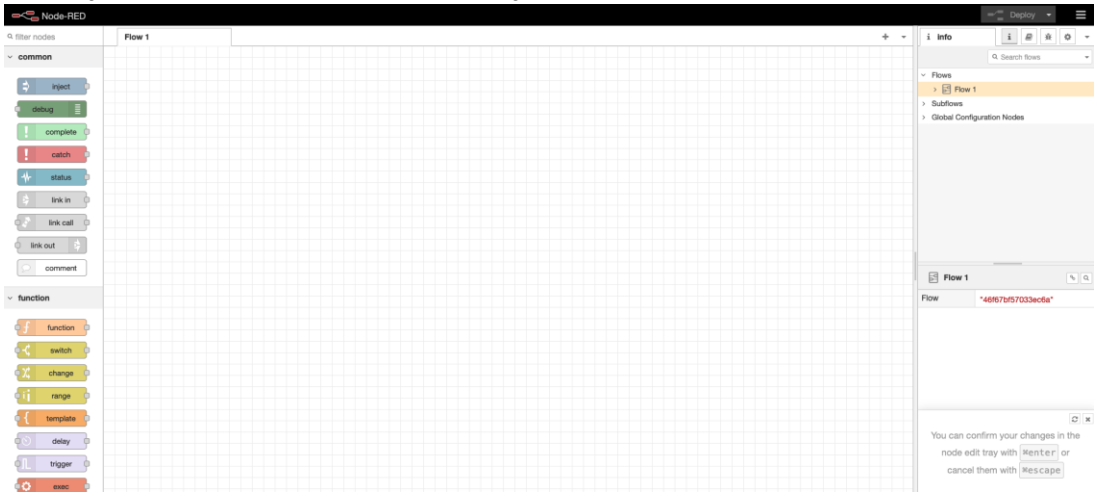
Nástroj Node-RED vyžaduje pre svoje fungovanie balíčkovací inštalačný nástroj **npm** a knižnicu pre tvorbu webových aplikácií **nodejs**. Po nainštalovaní vyššie uvedených nástrojov je možné inštalovať samotnú Node-RED aplikáciu. Pre inštaláciu vykonajte nasledujúce príkazy:

```
sudo apt install npm
sudo apt install nodejs

npm -version
node -version

sudo npm install -g --unsafe-perm node-red
```

V tejto chvíli je možné spustiť nástroj Node-RED príkazom:

```
sudo node-red
```

V tejto chvíli je aplikácia dostupná na webovej stránke **http://localhost:1880**



Tvorba prototypu pre monitorovanie a správu sieťových zariadení

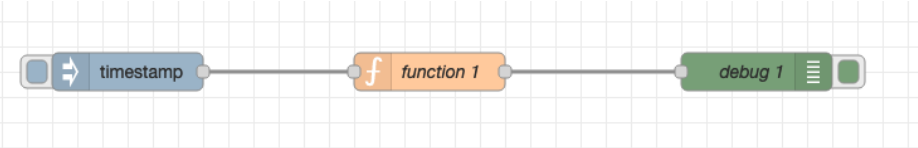
Nástroj Node-RED poskytuje množstvo uzlov, ktoré reprezentujú istý aspekt informačného systému a sú konfigurovateľné využitím používateľského rozhrania, čím umožňujú vytváranie prototypov bez použitia nutnosti programovať a ovládať programovací jazyk.

Základné Objekty

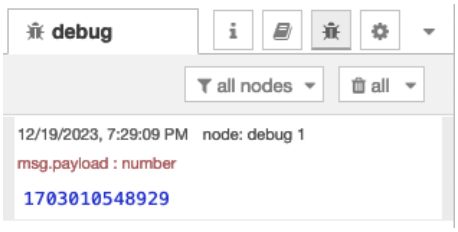
Medzi základné objekty pre prácu s nástrojom Node-RED patrí:

- **Inject**: umožňuje spustiť dátový tok odoslaním istej správy (možné definovať obsah)
- **Function**: umožňuje prijať správu, upraviť jej obsah a odoslať upravenú správu na ďalší uzol
- **Debug**: umožňuje zobraziť prijatú správu

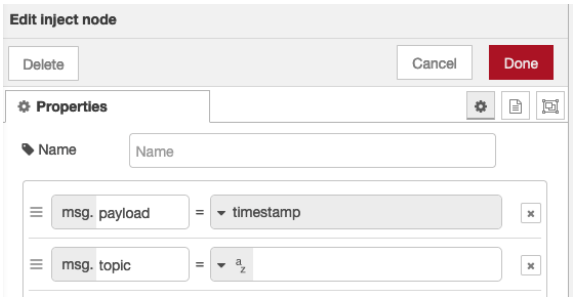
Uvedené uzly je možné vložiť do pracovnej plochy a vzájomne ich prepojiť ťahaním myši:



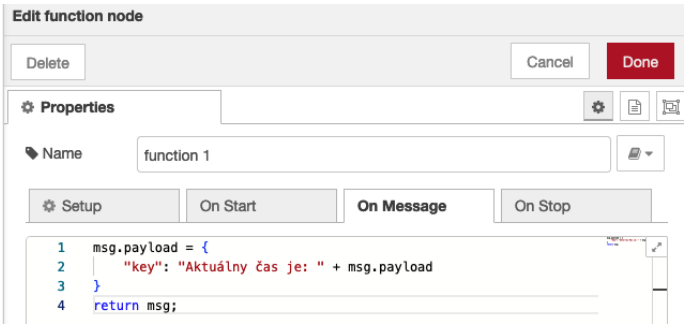
Po vložení objektov je potrebné stlačiť tlačidlo Deploy. Teraz je možné kliknúť na tlačidlo vstupu pre inject objekt, čo spôsobí odoslanie časovej značky, ktorá po príchode do debug objektu bude zobrazená v debug okne:

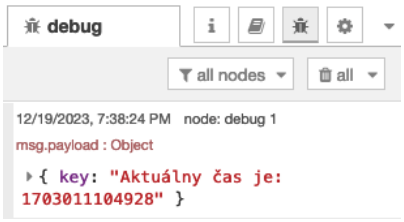


Dvojklikom na daný objekt je možné zobraziť jeho konfiguračné možnosti a pri objekte input je možné definovať obsah správy ale aj ako často sa bude správa odosielať. Možné napr. nastaviť, že po vykonaní operácie Deploy sa automaticky odošle správa, čím sa spustí dátový tok, alebo v prípade potreby generovania správy v istých časových intervaloch. Tento prípad je možné použiť, ak potrebujeme vykonávať istú funkciu pravidelne, napr. čítať štatistické dáta každú 1 sekundu.



Pre zmenu obsahu prenášanej správy je možné použiť uzol funkcia, ktorý umožní prijatú správu, ktorá je uložená v objekte msg.payload upraviť a preposlať ďalej. Funkcie je potrebné písať v jazyku JavaScript. Vzorové riešenie je znázornené na nasledujúcom obrázku:

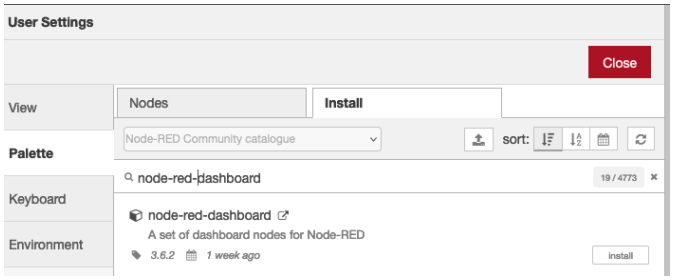




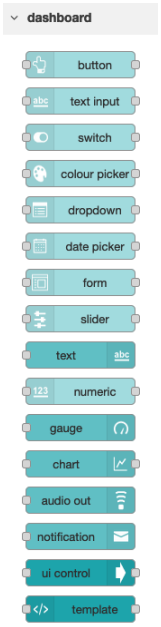
Nasledujúce úlohy sa budú venovať objektom potrebným pre správu a monitorovanie sieťových zariadení protokolmi RESTCONF, SNMP, ICMP, Syslog a SSH.

Objekt Dashboard

Aplikácie pre správu a monitorovanie sieťových zariadení často obsahujú používateľské rozhranie, najčastejšie webové. Pre tento účel nástroj Node-RED obsahuje modul **node-red-dashboard**, ktorý je možné nainštalovať využitím záložky **Manage Palette**:



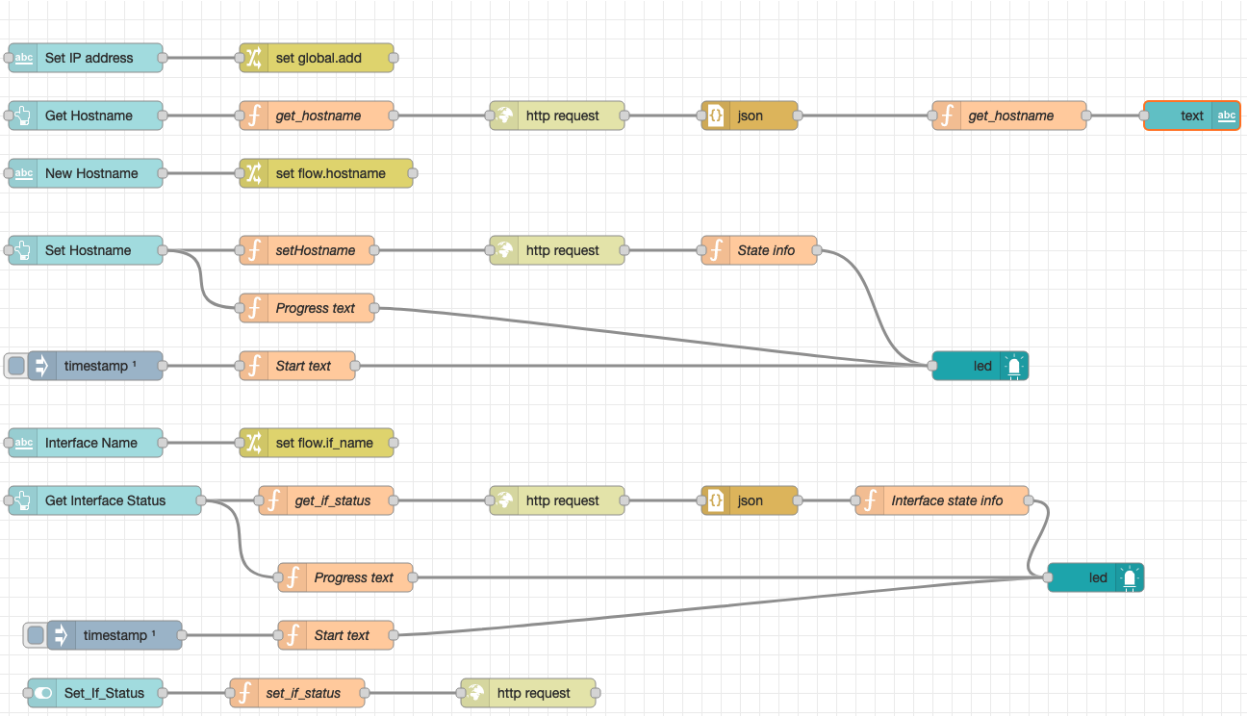
Po inštalovaní pribudne medzi zoznamom dostupných uzlov množina nových:



Vytvorené používateľské rozhranie je dostupné na adrese: **http://localhost:1880**

Čitanie a zmena konfigurácie využitím protokolu RESTCONF

Vytváranie všetkých tokov by trvalo dlho a preto pre zjednodušenie práce je k dispozícii už predvytvorený Flow, ktorý je možné jednoducho importovať do projektu. Aktuálny tok vyžaduje ešte inštaláciu modulu **node-red-contrib-ui-led**. Výsledný Flow s všetkými potrebnými uzlamy vyzerá nasledovne:

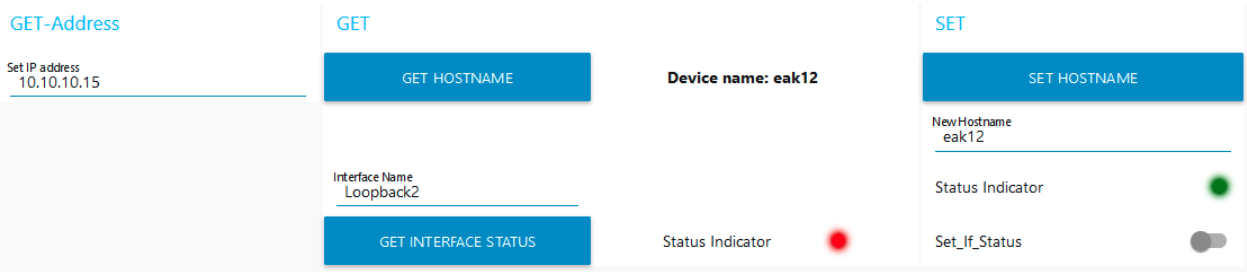


Node-red vyžaduje prácu s validnými certifikátmi. Vzhľadom k tomu, že zariadenie CSR1kv, ku ktorému sa pripájame má self-signed certifikát, tak je potrebné vypnúť kontrolu validity certifikátov (nebezpečné na reálnom serveri):

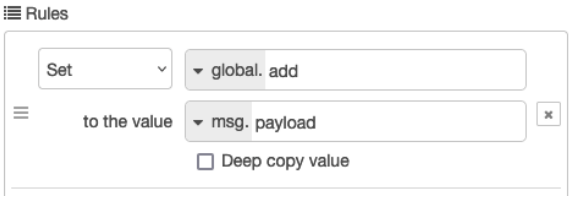
```
export NODE_TLS_REJECT_UNAUTHORIZED=0  - linux
set NODE_TLS_REJECT_UNAUTHORIZED=0  - Windows
```

V prípade pretrvávajúcich problémov je potrebné pregenerovať certifikát alebo použiť iný operačný systém (Windows nezvykne mať s touto úlohou problém).

Po spustení používateľského rozhrania bude viditeľné nasledujúce rozhranie:



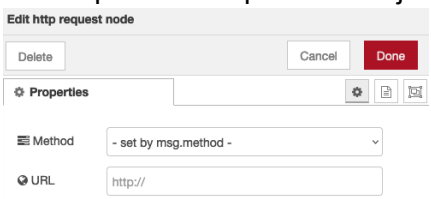
Elementy úplne naľavo a napravo predstavujú elementy používateľského rozhrania na získanie a zobrazenie dát. Ich nastavenie si viete pozrieť ich rozkliknutím. Elementy **change** slúžia na uloženie načítaných dát do premenných, ktoré je možné používať neskôr v iných uzloch. Pre pochopenie je ukázané uloženie IP adresy do globálnej premennej s názvom **add** (k tejto premennej sa je možné neskôr dostať použitím zápisu global.add):



Zdrojový kód funkcie, ktorá vygeneruje HTTP požiadavku je nasledovný:

```
msg.url ="https://"+ global.get("add") +"/restconf/data/native/hostname";
msg.method="GET";
msg.payload = {};
msg.headers = {
   "content-type": "application/yang-data+json",
   "accept": "application/yang-data+json",
   'Authorization': 'Basic Y2lzY286Y2lzY28xMjMh'
};
return msg;
```

Samotný HTTP request node je potrebné nastaviť nasledovne (nepoužijeme ho na definovanie požiadavky, ale celá požiadavka bude prevziatá z predchádzajúcej funkcie):



Prijatú odpoveď je potrebné deserializovať a previesť prijatý reťazec do JSON podoby, na čo je využitý uzol JSON:

Posledná funkcia pred vypísaním názvu zariadenia do používateľského rozhranie už len prijatý JSON rozparsuje a získa z neho konkrétny názov zariadenia:

```
msg.payload = msg.payload["Cisco-IOS-XE-native:hostname"];
return msg;
```

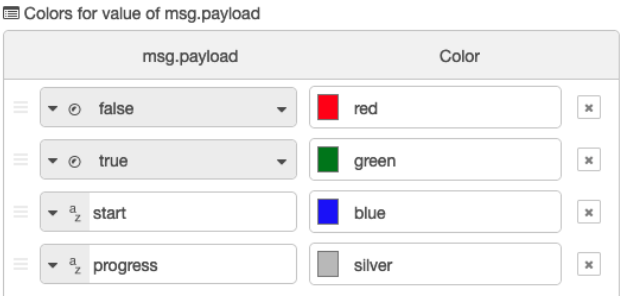
Podobným spôsobom je definovaná funkcia pre nastavenie názvu zariadenia:

```
msg.url = "https://"+ global.get("add") +"/restconf/data/native/hostname";
msg.method="PUT";
msg.payload = {
   "Cisco-IOS-XE-native:hostname":flow.get("hostname")
};
msg.headers = {
   "content-type": "application/yang-data+json",
   "accept": "application/yang-data+json",
   'Authorization': 'Basic Y2lzY286Y2lzY28xMjMh'
};
return msg;
```

V tomto prípade je súčasťou toku aj funkcia pre získanie návratovej hodnoty indikujúcej úspešnosť vykonania zmeny názvu zariadenia, ktorá vyzerá nasledovne:

```
var trueMsg = { payload: true };
var falseMsg = { payload: false };

if (msg.payload === "") {
   return trueMsg;
}
else {
   return falseMsg;
}
```

Funkcia vráti hodnotu true/false. Na tieto hodnoty reaguje uzol LED, ktorý sa rozsvieti na príslušnú farbu. Nastavenie uzla LED je nasledovné:



Poslednou dvojicou funkcii sú funkcie na zistenie stavu rozhrania a na zmenu stavu rozhrania, ktorých zdrojový kód vyzerá nasledovne:

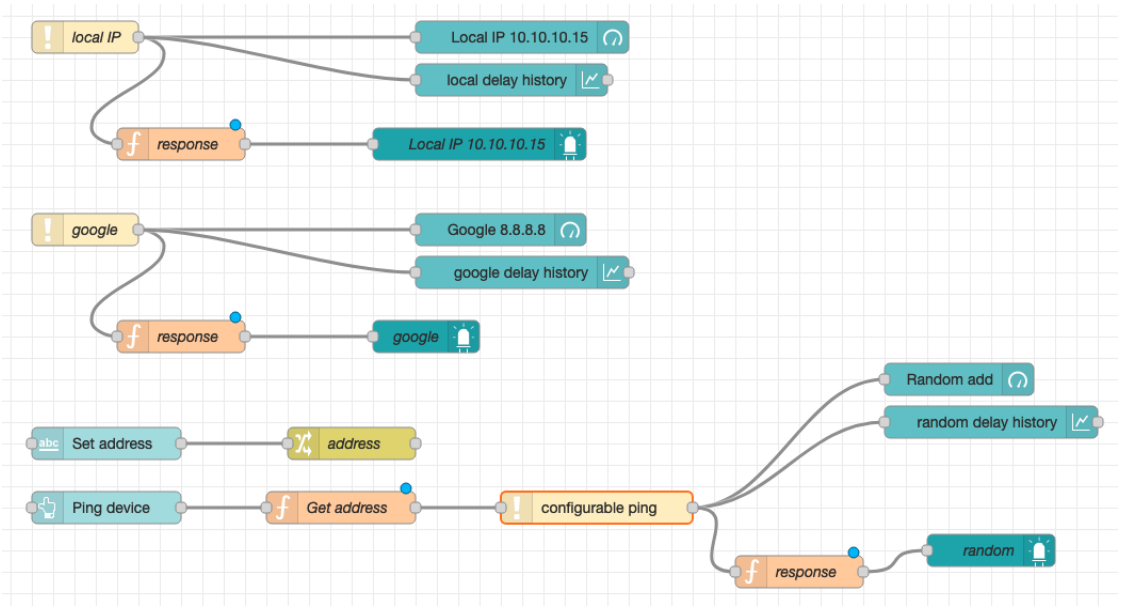
```
msg.url = "https://" + global.get("add") + "/restconf/data/ietf-
interfaces:interfaces/interface="+flow.get("if_name")+"/enabled";
msg.method="GET";
msg.payload = {};
msg.headers = {
```

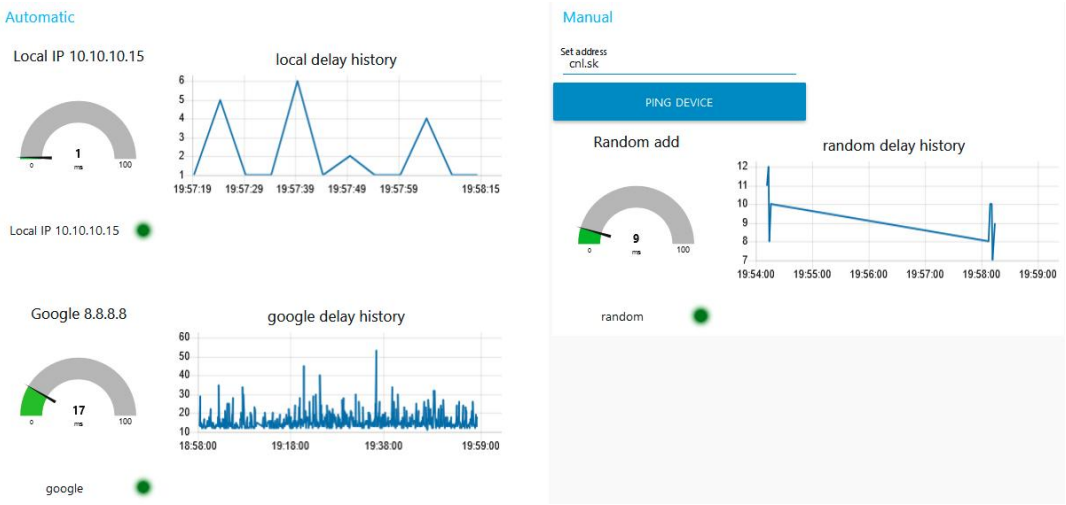
```
    "content-type": "application/yang-data+json",
    "accept": "application/yang-data+json",
    'Authorization': 'Basic Y2lzY286Y2lzY28xMjMh'
};

return msg;
```

```
msg.url = "https://" + global.get("add") + "/restconf/data/ietf-
interfaces:interfaces/interface=" + flow.get("if_name") + "/enabled";
msg.method="PUT";

if(msg.payload === true){
    msg.payload = {"ietf-interfaces:enabled": true};
}else{
    msg.payload = {"ietf-interfaces:enabled": false};
}

msg.headers = {
    "content-type": "application/yang-data+json",
    "accept": "application/yang-data+json",
    'Authorization': 'Basic Y2lzY286Y2lzY28xMjMh'
};

return msg;
```

Testovanie dostupnosti nástrojom PING

Táto úloha vyžaduje nainštalovanie modulov **node-red-configurable-ping** a **node-red-node-ping**.
Celkový Flow spolu s výsledným používateľským rozhraním vyzerajú nasledovne:

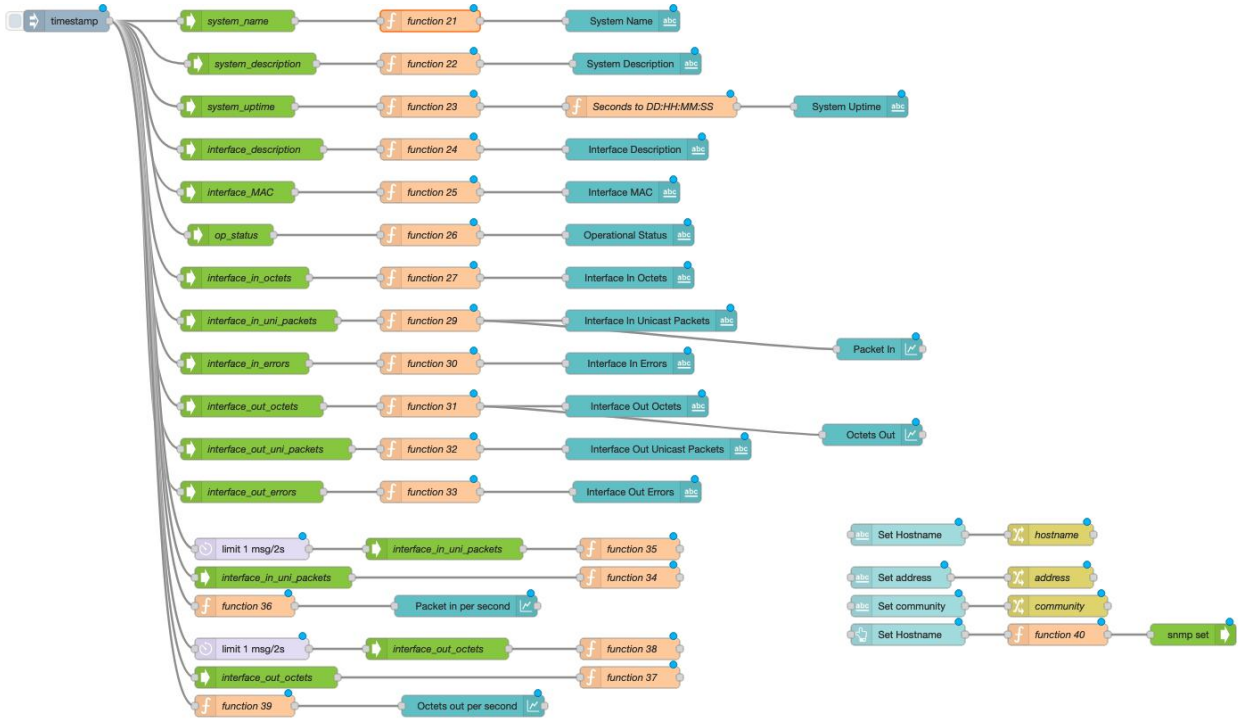




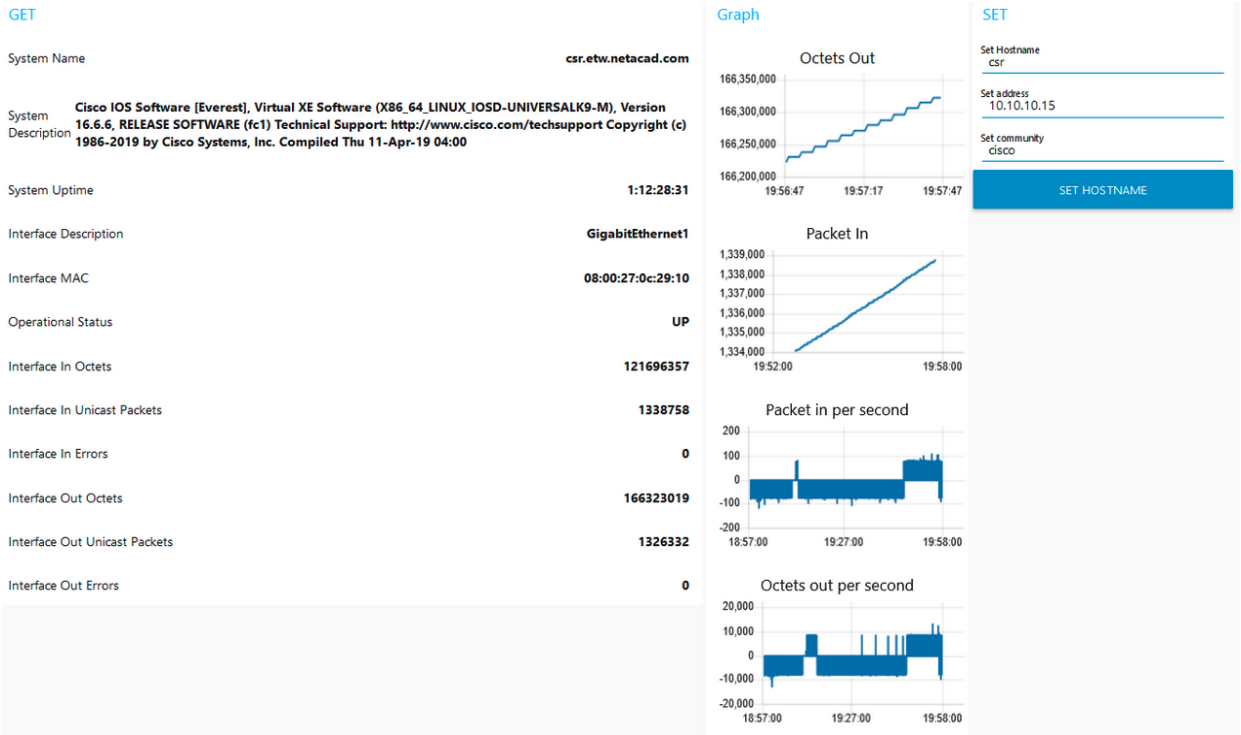
Nastavenie ICMP objektov vyzerá nasledovne (druhý ICMP typ prijme Host adresu z predchádzajúcej funkcie):

Získavanie štatistických dát a zmena konfigurácie protokolom SNMP

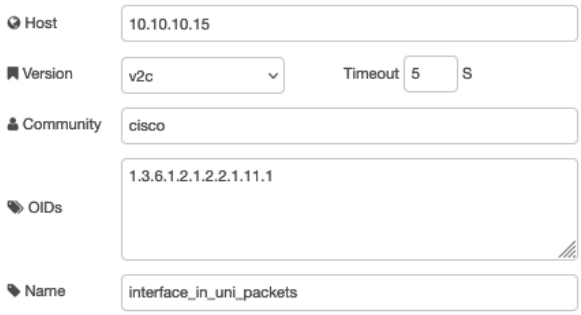
Pre prácu s protokolom SNMP je potrebné inštalovať modul **node-red-node-snmp**. Výsledný Flow vyzerá nasledovne:



Výsledné používateľské rozhranie vyzerá nasledovne:



Práca s protokolom SNMP je jednoduchá, pretože pri ňom stačí definovať SNMP parametre pre pripojenie sa k cieľovému zariadeniu a následne obslužnú funkciu, ktorá rozparsuje prijaté dáta a získa požadovanú hodnotu:

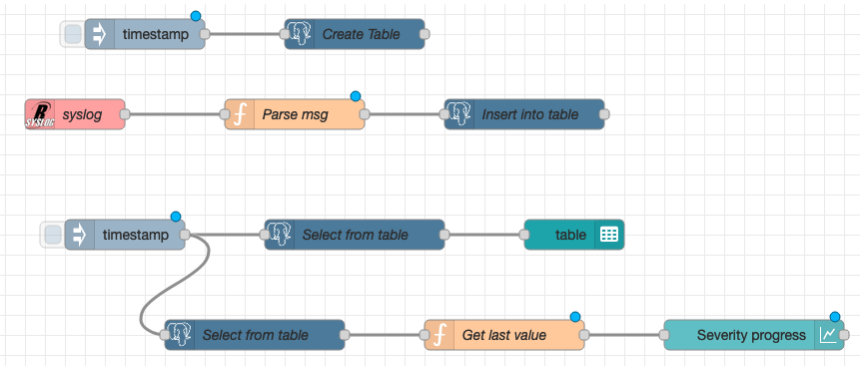


Zmena konfigurácie protokolom SNMP ale vyžaduje aj telo, ktoré je možné vygenerovať využitím nasledujúcej funkcie:

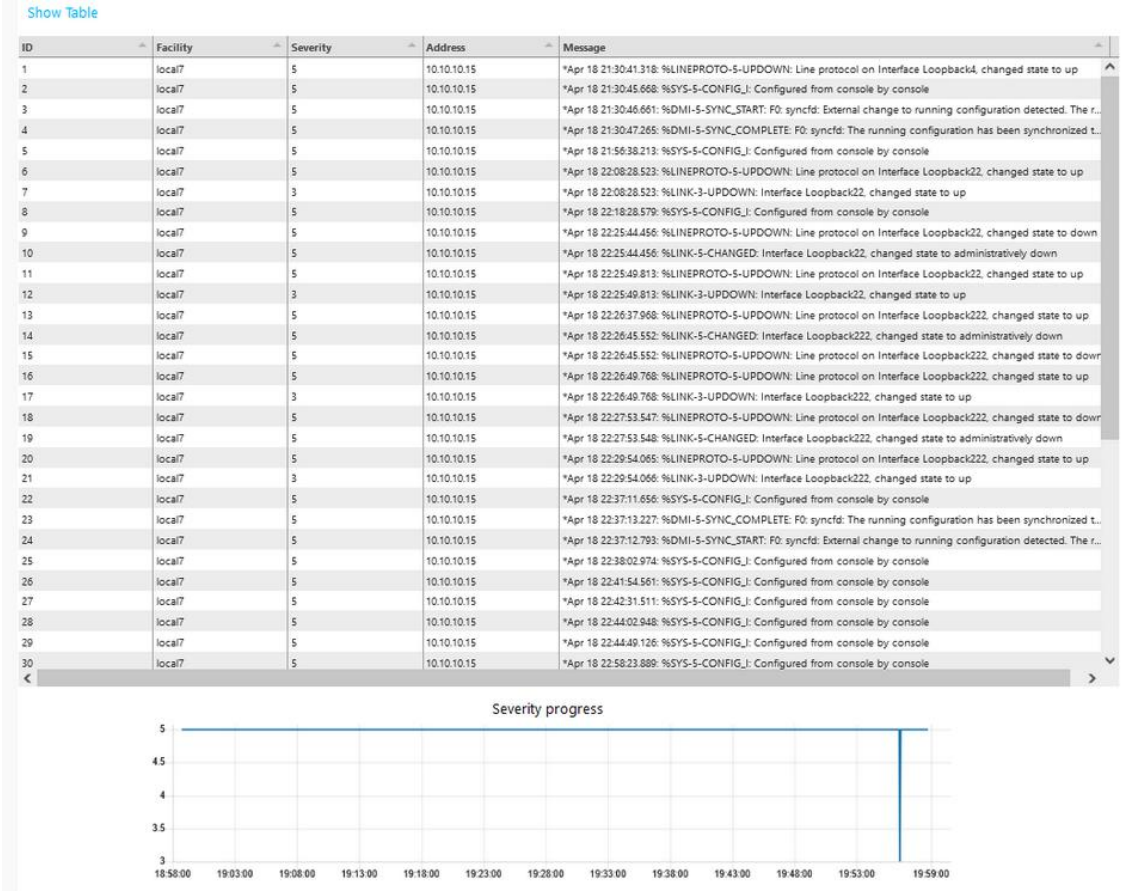
```
msg.varbinds = [
    {
        "oid": "1.3.6.1.2.1.1.5.0",
        "type": "OctetString",
        "value": flow.get("hostname")
    }
];
msg.host = flow.get("address");
msg.community = flow.get("comm");
return msg;
```

Získavanie a ukladanie syslog dát do databázy

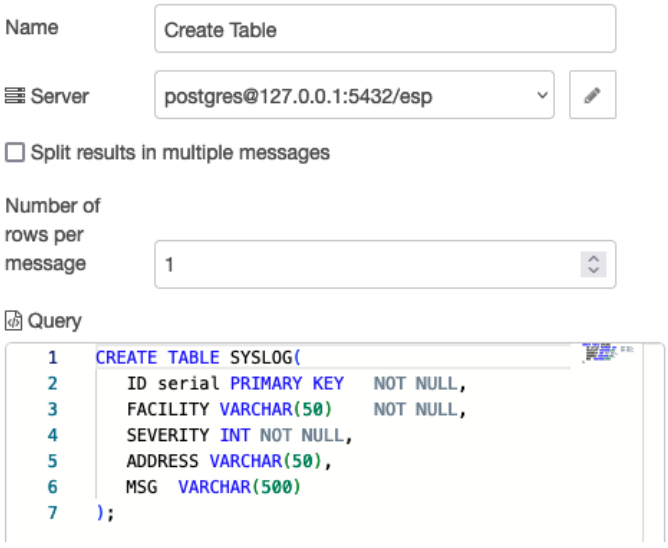
Pre riešenie tejto úlohy je potrebné inštalovať nasledujúce moduly: **node-red-node-ui-table**, **node-red-contrib-postgresql**, **node-red-contrib-syslog-input2**. Výsledná Flow vyzerá nasledovne:



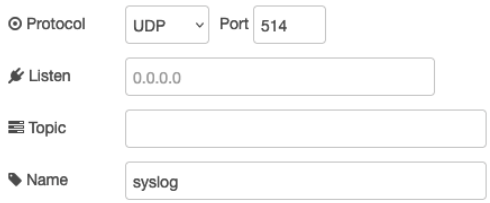
Používateľské rozhranie prototypu je zobrazené na nasledujúcom obrázku:



V rámci tohto cvičenia je potrebné mať predinštalovanú PostgreSQL databázu. Interakcia s databázou využitím node-RED komponentov je priamočiara, čo znázorňuje nasledujúci obrázok:



Interakcia s protokolom Syslog je jednoduchá, pričom stačí vložiť objekt reprezentujúci syslog server a na monitorovanom zariadení ho nastaviť ako cieľ komunikácie.



Prijaté syslog dáta sú štruktúrované a preto pre ich zápis do databázy ich potrebujeme najskôr rozparsovať a vložiť do databázy, čo je možné dosiahnuť nasledujúcou funkciou:

```
var v1 = msg.payload.facility;
var v2 = msg.payload.severityCode;
var v3 = msg.payload.address;
var v4 = msg.payload.msg;

msg.query = `INSERT INTO SYSLOG(FACILITY, SEVERITY, ADDRESS, MSG) VALUES('${v1}',
${v2}, '${v3}', '${v4}');`
return msg;
```

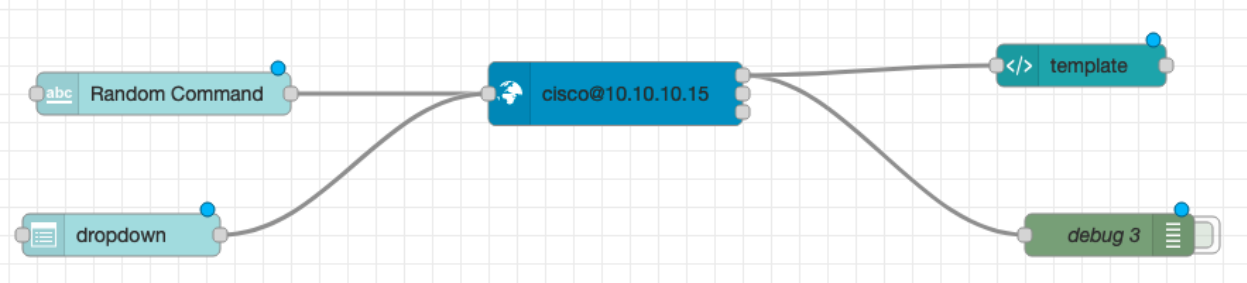
Nasledujúci PostgreSQL objekt už obsahuje len informácie pre pripojenie k databáze a neobsahuje samotnú požiadavku. Pre získanie dát z databázy stačí vytvoriť jednoduché select poziadavky, ktoré vyzerajú nasledovne:

```
SELECT * FROM syslog;
SELECT severity FROM syslog;
```

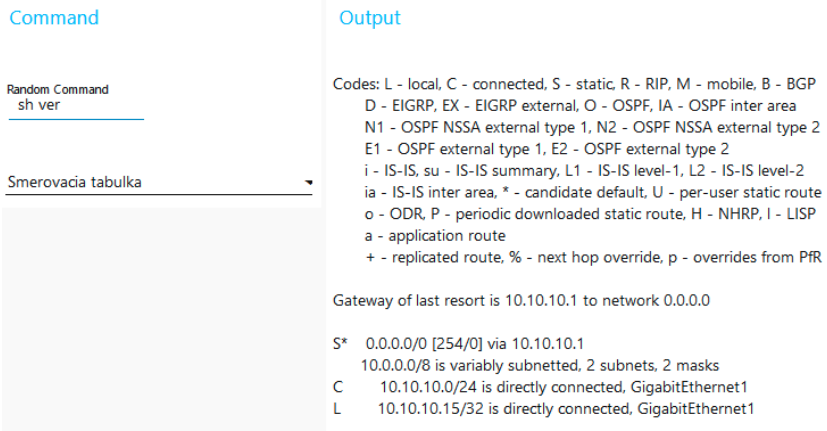
Pre zobrazenie dát do tabuľovej podoby je možné definovať atribúty tabuľky na základe získaných hodnôt z databázy nasledovne:

Konfigurácia sieťového zariadenia protokolom SSH

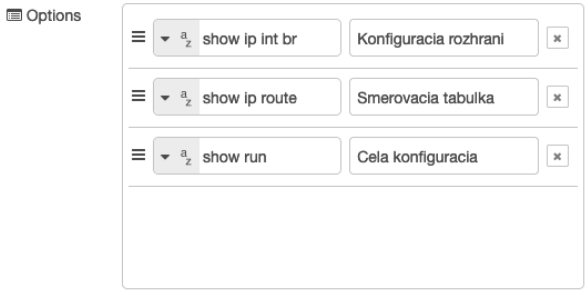
Pre prácu s protokolom SSH je potrebné inštalovať modul **node-red-contrib-bigssh**. Výsledný Flow vyzerá nasledovne:



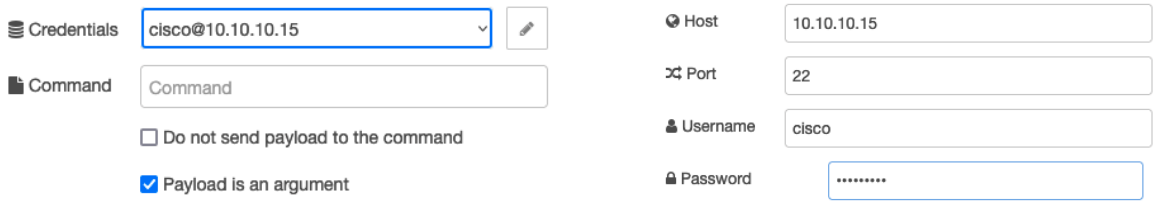
Použitím vyššie uvedeného Flow-u dôjde k vytvoreniu nasledujúceho rozhrania:



Objekt SSH prijme riadiaci príkaz, ktorý následne odošle na cieľové zariadenie. Prvý vstup umožňuje používateľovi vložiť akýkoľvek príkaz pre privilegovaný režim. Druhý zoznam je jednoduché mapovanie používateľsky prívetivých názvov, za ktorými sa ale skrýva reálny IOS príkaz:



Definovanie SSH pozostáva z vyplnenia nasledujúcich atribútov:



Posledným komponentom tejto úlohy je objekt šablóny, ktorý je potrebný pre zobrazenie prijatých dát do čitateľnej podoby (dôležité správne interpretovať biele znaky). Za týmto účelom stačí definovať CSS štýl nasledovne:

```
<style>
   #mylog {
       min-height: 300px;
       padding: 0;
       background-color: white;
       white-space: pre;
   }
</style>

<div id="mylog" ng-bind="msg.payload" contenteditable="true">
   <!-- payload will go here -->
</div>
```