

IoT Challenge 2018- Node-RED Intro

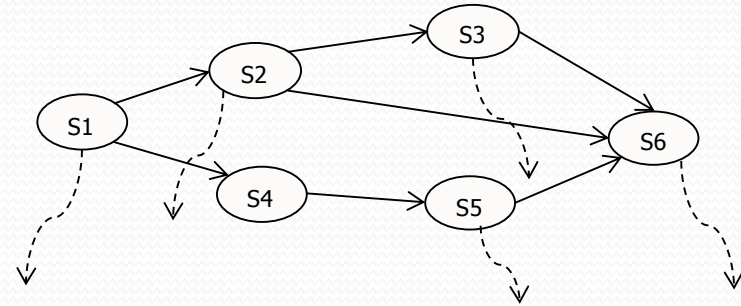
16-17 Μαρτίου 2018

Πηγές δεδομένων

- Πληροφορίες MMM
- Υπηρεσίες έξυπνων πόλεων
 - Θέσεις παρκινγκ
 - Πληροφορίες γεγονότων
 - Κίνησης
 - Συγκεντρώσεων
 - Πολιτιστικών γεγονότων
- Κοινωνικά Δίκτυα
- Δημόσιες υπηρεσίες
 - Πληροφορίες ελέγχων
 - Καιρού
 - Κατάστασης δρόμων, ταχύτητας κλπ
- Όλες διαθέσιμες στον χρήστη
 - Αλλά μη συνδεδεμένες μεταξύ τους
 - Προσφερόμενες μέσω πληθώρας πρωτοκόλλων
 - RSS, REST Services, Messaging systems κλπ

Ροές εργασίας

- Ορισμός: Υπηρεσίες που προσφέρουν λειτουργικότητα βάσει άλλων υπάρχουσών υπηρεσιών
- Η παροχή προϋποθέτει
 - δημιουργία (μέσω γλώσσας)
 - εκτέλεση (μέσω «ενορχηστρωτών»)
- Οι υπηρεσίες προκύπτουν μέσω σύνθεσης διεπαφών (mashups) και δημιουργίας ροών εργασίας (workflows)
- Επιτρέπουν την αυτοματοποίηση εργασιών, τη δημιουργία ακολουθίας εργασιών και την υλοποίηση πολλαπλών διασυνδεδεμένων βημάτων



Βασικά προβλήματα

- Πολλαπλά πρωτόκολλα
 - διαφορετικά για κάθε πηγή
- Ασύγχρονη Λογική
- Χρειάζεται λογική συμπίεσης της πληροφορίας στη μορφή γεγονότων και έλεγχος αν αυτά αφορούν το χρήστη
- Πώς μπορούν να λυθούν?

Layer		Protocol	Implementation
User System Bridge	Level Feed	DDP to MQTT	Meteor and Node-RED
Analytics Job submission		REST	Spark and Node-RED
Analytics Link to Storage		SQL syntax	Spark SQL driver
Coordination between Spark and NodeRED		MQTT	Node-RED
Twitter Counter Services per box		REST	Node-RED
CEP and Node-RED		MQTT	Node-RED
Notification to User Level App (Reactive Box)		AMQP	Node-RED node with underlying script

Υλοποιήσεις ροών εργασίας

- Για να χρησιμοποιήσουμε με αυτή τη λογική μια ροή εργασίας που περιλαμβάνει
 - Πόρους
 - Πλαίσια
 - Δεδομένα
 - Κλπ
- Χρειάζεται μια λογική διασύνδεσης, υλοποίησης και έκφρασης της ακολουθίας πράξεων
 - Ασύγχρονης λογικής (node.js)
- Χρειάζεται μια λογική ύπαρξης και άλλων πελατών για επί μέρους συστήματα που μπορεί να μην είναι σε REST
 - Ειδικά για τα δεδομένα
 - Αποθετήριο node.js
- Είναι καλό να υπάρχει μια διεπαφή φιλική προς το χρήστη ώστε η σχεδίαση της ροής να είναι γραφική και παραμετροποιήσιμη
 - Node-RED

Drag and Drop

The screenshot displays the Node-RED web interface. On the left, the 'input' node palette is visible, containing nodes like 'inject', 'catch', 'status', 'link', 'http', 'websocket', 'tcp', and 'udp'. A red box highlights the 'inject' node being dragged from this palette into the workspace. In the workspace, a 'timestamp' node is already present, also highlighted with a red box. On the right, the 'info' sidebar is open, showing the properties of the selected 'inject' node. A red box highlights this sidebar, which includes a 'Deploy' button at the top and a detailed description of the 'inject' node's functionality and options.

Node-RED

filter nodes

Flow 1

input

inject

catch

status

link

http

websocket

tcp

udp

output

debug

link

http response

timestamp

Deploy

info

debug

Node

Type	inject
ID	b6abb2c7.f3da28

Properties

Pressing the button on the left side of the node allows a message on a topic to be injected into the flow.

The payload defaults to the current time in millisecs since 1970, but can also be set to various other javascript types.

The repeat function allows the payload to be sent on the required schedule.

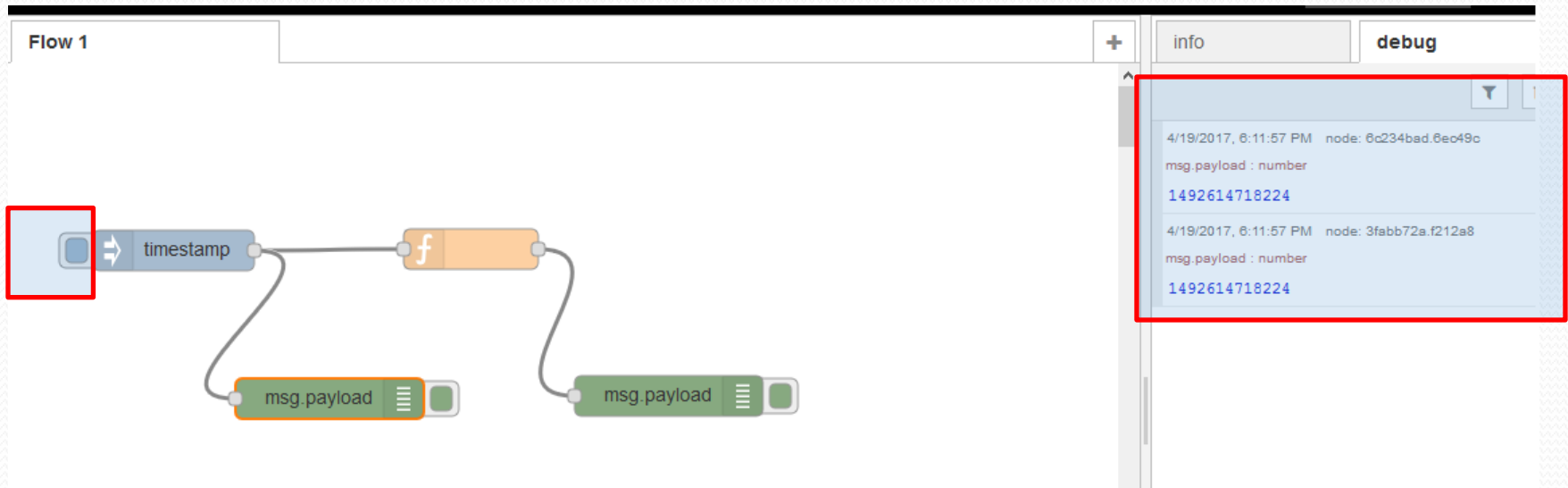
The *Inject once at start* option actually waits a short interval before firing to give other nodes a chance to instantiate properly.

The *Flow* and *Global* options allow one to inject a flow or global context value.

Note: "Interval between times" and "at a specific time" uses cron. This means that 20 minutes will be at the next hour, 20 minutes

Εκτέλεση ροής

- Όταν έρθει κάποιο msg σαν είσοδος από κάποιο connection/endpoint
- Αν δεν έχετε τέτοιο στη ροή μπορείτε να κάνετε trigger μέσω inject node



- Με mouse over στο εκτυπωμένο βλέπετε από ποιον κόμβο εκτυπώθηκε

Κόμβος function

- Ο κόμβος function είναι πολύ χρήσιμος, μπορείτε να βάλετε ένα δικό σας κομμάτι κώδικα που να πραγματοποιεί οποιαδήποτε λειτουργία στο εισερχόμενο μήνυμα
 - Ή και γενικότερη
 - Εγγραφή σε μια υπηρεσία που να λαμβάνει ασύγχρονα μηνύματα
 - Π.χ. Εγγραφή σε κοινωνικά δίκτυα
- Στην προηγούμενη ροή δεν είχαμε βάλει τίποτα οπότε απλά μεταφέρει το μήνυμα χωρίς αλλαγές στην έξοδο

Αποθήκευση/αντιγραφή ροής

The screenshot displays the Node-RED web interface. On the left, a flow diagram for 'Flow 1' is visible, featuring a 'timestamp' node connected to a function node 'f', which in turn connects to two 'msg' nodes. On the right, a sidebar contains a menu with the following items: View, Import, Export, Search flows, Configuration nodes, Flows, Subflows, Manage palette, Keyboard shortcuts, Show tips (checked), and Node-RED website. A red rectangular box highlights the 'Clipboard' and 'Library' options within the sidebar menu.

Flow 1

timestamp

f

msg

msg

info

debug

Clipboard

Library

4/19/2017, 6:11:57 PM node: 3fab72a.f212a8
msg.payload : number
1492614718224

4/19/2017, 6:26:50 PM node: 6c234bad.6ec49c
msg : Object
▶ { _msgid: "ead12466.152ed8", topic: "myfield" }

4/19/2017, 6:26:50 PM node: 3fab72a.f212a8
msg : Object
▶ { _msgid: "ead12466.152ed8", topic: "myfield" }

Deploy

View

Import

Export

Search flows

Configuration nodes

Flows

Subflows

Manage palette

Keyboard shortcuts

☒ Show tips

Node-RED website

Αποθετήριο ροών

- Με την ίδια λογική μπορείτε να εισάγετε ροές (δικές σας ή από αποθετήρια)
- <https://flows.nodered.org/>
- Επιλογή
- Αντιγραφή περιεχομένου
- Import από τις επιλογές
 - Clipboard
- Paste

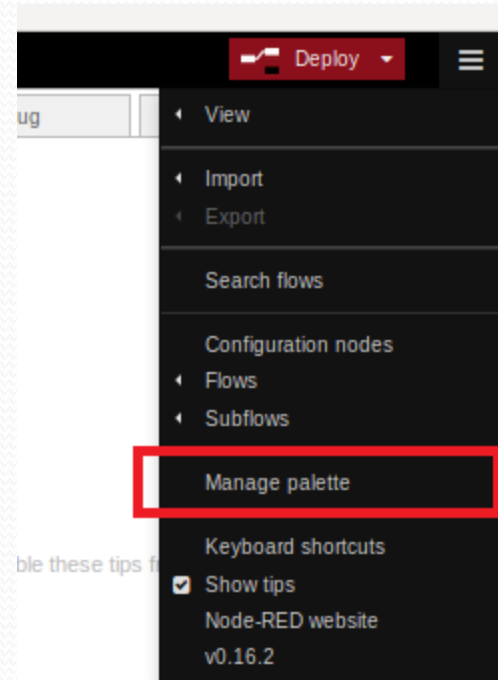
[illegible]

Προσοχή!

- Κατά την εισαγωγή μιας ροής μπορεί να σας βγάλει μήνυμα ότι έχει η συγκεκριμένη ροή ένα κόμβο που δεν μπορεί να τον αναγνωρίσει το Node-RED
- Αν το αγνοήσετε και πατήσετε Deploy, θα σταματήσει ο Node-RED server και δεν θα μπορεί να γίνει start μέχρι να σβήσετε το επίμαχο κομμάτι από το flows.json file του περιβάλλοντος!
- Για αυτό πρέπει να σβήσετε τη ροή από το workspace, κάνετε install τον Node-RED node που λείπει και την αντιγράφετε μετά

Εισαγωγή κόμβου που είναι Node-RED packaged

Select
install tab



Εισαγωγή κόμβου που δεν είναι Node-RED packaged

- Κανονικά μέσα στον κώδικα του function θα γράφατε
 - `require('somelib');`
 - Αντίστοιχο του `import` σε άλλες γλώσσες
- Επειδή το Node-RED τρέχει σε sandbox και δεν έχει πρόσβαση σε όλο το Node.js περιβάλλον, χρειάζεται μια συγκεκριμένη διαδικασία

Kmeans node παράδειγμα

- <https://github.com/NathanEpstein/clusters>
- `sudo npm install clusters --save`
- Από το `/home/user/`
- `cd .node-red`
- `sudo nano settings.js`
- Με `ctrl-W` ψάξτε για το «global»
- Εισάγετε τη γραμμή
 - Επανεκκίνηση Node-red
- Το 'clusters' είναι το όνομα του πακέτου που θέλετε
- Το clusters είναι ένα όνομα που δίνετε εσείς για reference μέσα στον κώδικα

```
// in front of all http in nodes. This allows custom authS
// applied to all http in nodes, or any other sort of comS
//httpNodeMiddleware: function(req,res,next) {
//    // Handle/reject the request, or pass it on to the S
//    // Optionally skip our rawBodyParser by setting thiS
//    //req.skipRawBodyParser = true;
//    next();
//},

// Anything in this hash is globally available to all funS
// It is accessed as context.global.
// eg:
//    functionGlobalContext: { os:require('os') }
// can be accessed in a function block as:
//    context.global.os

functionGlobalContext: {
    clusters:require('clusters')
    // os:require('os'),
```

Documentation & Example

```
var clusterMaker = require('clusters');
```

```
//number of clusters, defaults to undefined  
clusterMaker.k(3);
```

```
//number of iterations (higher number gives more time to converge), defaults to 1000  
clusterMaker.iterations(750);
```

```
//data from which to identify clusters, defaults to []  
clusterMaker.data([[1, 0], [0, 1], [0, 0], [-10, 10], [-9, 11], [10, 10], [11, 12]]);
```

```
console.log(clusterMaker.clusters());
```

```
// { centroid: [10.5 , 11], points: [[10, 10], [11, 12]] },  
// { centroid: [-9.5, 10.5], points: [[-10, 10], [-9, 11]] },  
// { centroid: [0.3333333333333333, 0.3333333333333333], points: [[1, 0], [0, 1], [0, 0]] }
```

Αλλαγή τρόπου
δήλωσης

Function

```
1 var clusterMaker = context.global.clusters;  
2
```

```
3 //number of clusters, defaults to undefined  
4 clusterMaker.k(3);  
5
```

```
6 //number of iterations (higher number gives more time to converge), defaults to 1000  
7 clusterMaker.iterations(750);  
8
```

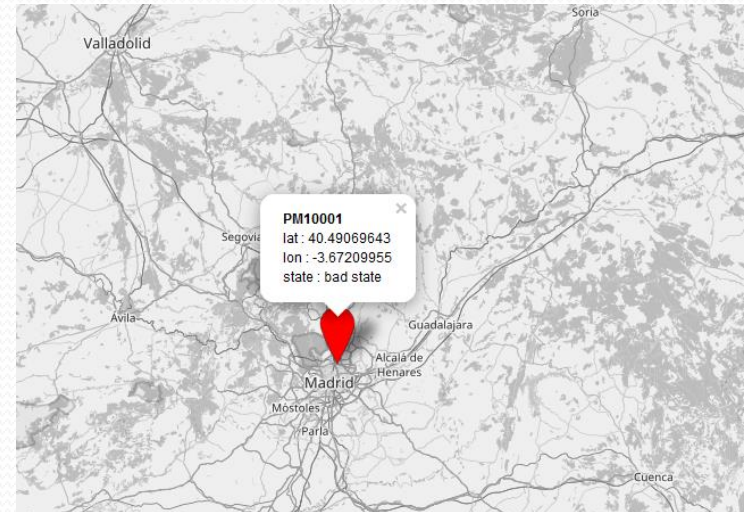
```
9 //data from which to identify clusters, defaults to []  
10 clusterMaker.data([[1, 0], [0, 1], [0, 0], [-10, 10], [-9, 11], [10, 10], [11, 12]]);  
11
```

```
12 msg.payload=clusterMaker.clusters()  
13 return msg;
```

Αλλαγή τρόπου
χειρισμού
αποτελέσματος
(εισαγωγή σε
μήνυμα και
προώθηση στον
επόμενο κόμβο
της ροής

Οπτικοποίηση μέσω Worldmap node-red node extension

- Δημιουργία JSON εισόδου
- Από το doc του κόμβου βλέπουμε ότι θέλουμε μέσα στο msg.payload:
 - .name: PM10001
 - .lon: -3.67209955
 - .lat: 40.49069643
 - .iconColor: Green / Red
 - https://www.w3schools.com/colors/colors_names.asp
- Μπορούμε να το φτιάξουμε στο προηγούμενο function node
- Δεξί κλικ- Ανοιγμα σε νέο tab
 - <http://IP:PORT/worldmap>



Dashboard node-red node extension

- Κόμβοι για web forms, graphs, gauges για μια μορφή analytics

