Node-RED

A Visual Tool for Wiring the Internet of Things
@NodeRED
nodered.org

dave conway-jones • @ceejay
Why Node-RED

We have
  Processors for editing Words
  Spreadsheets for working with Numbers
  Powerpoint for arranging Pictures and Ideas
Why Node-RED

We don't have a simple tool for co-ordinating Events
  Business events – status of processes, alerts from machines
  Social events – tweets, alerts
  Internet of Things events – temperatures, weather, lights, doors
Something that anyone can use to build situational applications
... Node-RED can fill that gap
**Node-RED** is

an application composition tool.
a lightweight proof of concept runtime.
easy to use for simple tasks.
simple to extend to add new capabilities and types of integration.
capable of creating the back-end glue between social applications.
a great way to try ...

"can I just get this data from here to here ?"
"...and change it slightly on the way"
Node-RED is NOT

an enterprise strength application runtime.
a dashboard with widgets.
a mobile application builder.
the answer to life the universe and everything...

... not yet anyway.
**Node-RED** is already capable of connecting to many things, including:
Inject node
- Allow manual triggering of flow
- Can be scheduled to automatically inject at fixed intervals

Debug node
- Shows message content, either just payload or entire object in the debug sidebar

Function node
- Runs user-defined js against the messages flowing past

Template node
- Renders messages into a {{mustache}} template

TCP/UDP nodes
- Connects out, or listens for incoming connections

Logic nodes
- Comparisons, re-scaling, re-mapping

HTTP nodes
- Define http endpoints for incoming requests, or trigger gets of urls in the middle of a flow

MQTT Nodes
- Publish and Subscribe nodes

Hardware nodes
- Serial devices
- node.js, v8 engine - fast
- event driven, asynchronous io - it's all about events
- single threaded, event queue - great for fairness
- javascript front and back - only the one language
msg object

{
  "payload": "Hello World!",
  "topic": "foo",
  ...
  ...
}
Function Node

Uses `vm.createScript` under the covers to sandbox execution

`console`, `util`, `Buffer` included for convenience,
easy to add others via settings
Function Node
Returning messages

return msg;

return [msg1, msg2, msg3];

return [msg1, [msgA, msgB]];
Function Context

Each function node has its own `context` object to save state between calls.

```javascript
context.count = context.count || 0;
context.count += 1;
var countMessage = { payload: context.count };
return [msg, countMessage];
```
Function Global Context

in settings.js

... functionGlobalContext: { arDrone: require('ar-drone') }, ...

makes a Node-RED-copter function simple...

context.client = context.client || context.global.arDrone.createClient();

if (msg.payload === "takeoff") { context.client.takeoff(); }
else if (msg.payload === "land") { context.client.land(); }
else ...
Easy to wrap any npm module into a palette node*

Each node defined in a pair of files:

- `js`: server-side behaviour
- `html`: appearance in editor & help

* Once we get the API documented
For example, `sentiment*:

```javascript
var sentiment = require('sentiment');
sentiment('Cats are stupid.',
    function (err, result) {
        console.dir(result); // Score: -2
    }
);

sentiment('Cats are totally amazing!',
    function (err, result) {
        console.dir(result); // Score: 4
    }
);
```

* https://github.com/thisandagain/sentiment
var RED = require(process.env.NODE_RED_HOME+"/red/red");
var sentiment = require('sentiment');

function SentimentNode(n) {
    RED.nodes.createNode(this,n);

    this.on("input", function(msg) {
        var node = this;
        sentiment(msg.payload, function (err, result) {
            msg.sentiment = result;
            node.send(msg);
        });
    });
}

RED.nodes.registerType("sentiment",SentimentNode);
var RED = require(process.env.NODE_RED_HOME+"/red/red");
var sentiment = require('sentiment');

function SentimentNode(n) {
    RED.nodes.createNode(this,n);

    this.on("input", function(msg) {
        var node = this;
        sentiment(msg.payload, function (err, result) {
            msg.sentiment = result;
            node.send(msg);
        });
    });
}

RED.nodes.registerType("sentiment",SentimentNode);
<script type="text/javascript">
RED.nodes.registerType('sentiment',{ 
    category: 'analysis-function',
    color: '#E8E6F8',
    defaults: {
        name: {value:""},
    },
    inputs:1,
    outputs:1,
    icon: "arrow-in.png",
    label: function() {
        return this.name||"sentiment";
    },
    labelStyle: function() {
        return this.name?"node_label_italic":"";
    }
});
</script>
<script type="text/x-red" data-template-name="sentiment">
  <div class="form-row">
    <label for="node-input-name"> Name </label>
    <input type="text" id="node-input-name">
  </div>
</script>
Wiring the internet of things

New developers and education
- Short learning curve
- Easy to use
- Low barrier to entry

App Developers
- Rapid prototyping
- Easy to integrate with existing tools and applications
- Easy to extend with richer/bespoke functionality

Community Developers
- Open standards
- Flexibility
- Ability to share

Hardware Hackers
- Runs on Raspberry Pi, Beaglebone, other low power devices.
- Works with Arduino, etc
- Easy to add devices
**Node-RED**

The Visual Tool for Wiring the Internet of Things

http://nodered.org

@nodered

chat.freenode.net/##nodered

dave conway-jones • @ceejay

IBM Emerging Technology
Node-RED
A visual tool for wiring the Internet of Things

Node.js based
Built on top of Node.js, taking full advantage of its event-driven, non-blocking IO model.

Social Development
Any part of a flow can be imported or exported using JSON for easy sharing and collaborative development.

Updates
Version 0.6.0 released - 21
Feb 2014