Flow-based programming: NodeRed

Andrea Sterbini – sterbini@di.uniroma1.it

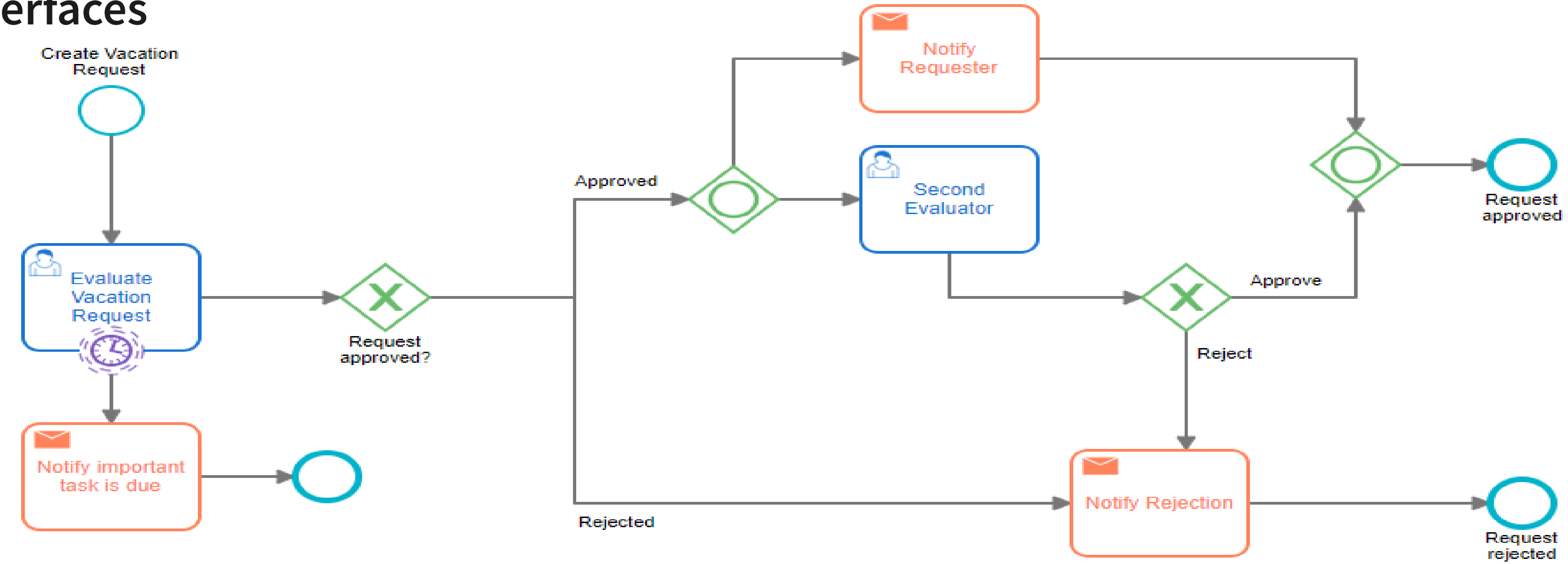
Flow-based programming

No-code programming style:

- Connect functional modules

- Configure their properties

- Build User Interfaces



Node-RED (http://nodered.org)

Flow-based visual programming tool created initially by IBM for IoT

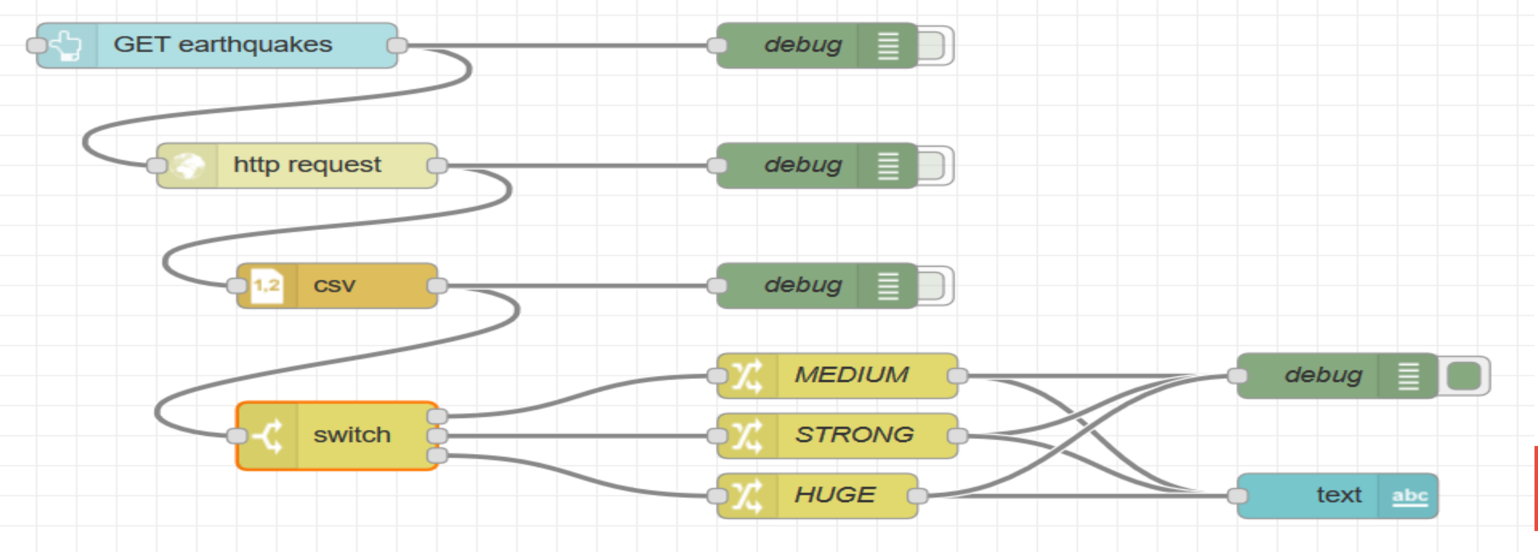
Interconnected JavaScript functional units (installed with NPM)

Wires communicate messages (Javascript objects/dictionaries) aggregating all data

- functional units get/add data to the messages and/or remember info
- messages can be split/joined into message sequences

Inherently parallel

A node could contain full programs



Concepts

Flow: a graph of interconnected nodes

Node: a functional unit (JavaScript program) triggered by a message or event

Context: a blackboard to share/store data at node/flow/global level

Wire: a bus interconnecting two nodes, transmitting messages

Message: a Javascript object/dictionary of informations passed from node to nodes

Subflow: a node containing a flow (==> hierarchical flows)

Group: a grouped set of nodes with label (documentation)

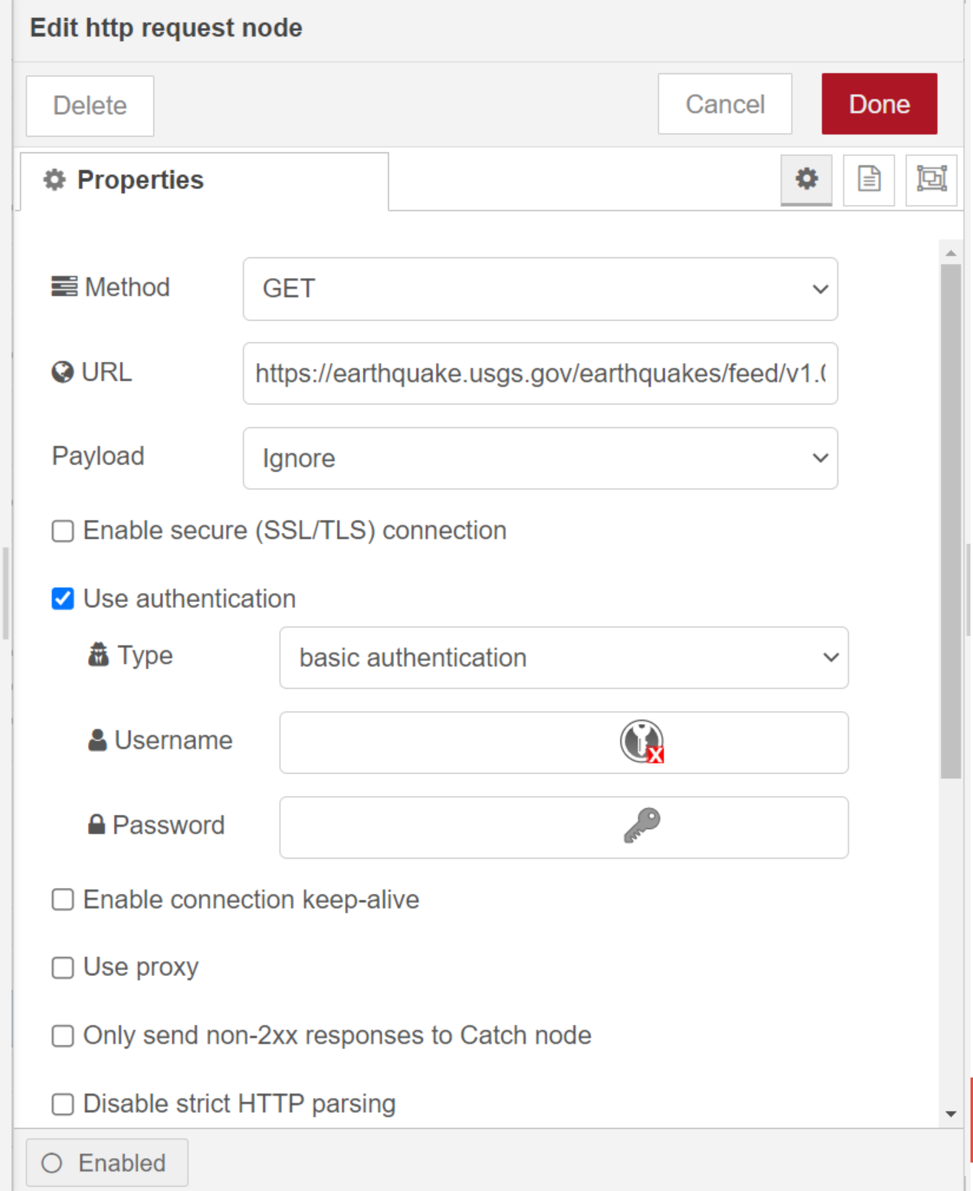
Function node: a programmable node updating the message (or blocking it)

Node properties

All nodes have properties that define their work

E.g. the HTTP request

- Type of method (GET/POST/PUT/DELETE/from parameter)
- URL
- what to do with the msg payload activating this
- authentication / proxy … etc
- headers
- what to return



Edit http request node

Method: GET

URL: https://earthquake.usgs.gov/earthquakes/feed/v1.(

Payload: Ignore

☐ Enable secure (SSL/TLS) connection

☑ Use authentication

Type: basic authentication

Username:

Password:

☐ Enable connection keep-alive

☐ Use proxy

☐ Only send non-2xx responses to Catch node

☐ Disable strict HTTP parsing

Features

Typed wires?	NO	(messages/dictionaries)
Functions?	YES	(in JavaScript, inside a node)
Functional programming?	NO?	(in Javascript, inside a node)
Recursion?	YES	(in Javascript, inside a node)
Loops?	YES	(in Javascript or with a node)
External languages?	NO?	(possibly an app in a node server?)
File I/O	YES?	(in Javascript, inside a node)
Modularization?	YES	(Subflows)
Concurrency?	YES	

Programming style

Low level programming is done in JavaScript in function nodes (if required)

Flows: higher level programming (organization, data exchange)

Good to blend many already available nodes + some personalized

Data exchange style: blackboard messages (JavaScript dictionaries)

- every node can read/write/modify messages from the stream
- there could be many "blackboards" if there are many paths

Some synchronization pattern is slightly cumbersome?

- wait for K data on the stream (count and keep memory)
- wait for K data from different streams (check for all present)

A huge palette of available Flows/Nodes
https://flows.nodered.org

Microcontrollers:

- Raspberry Pi
- Arduino

Applications:

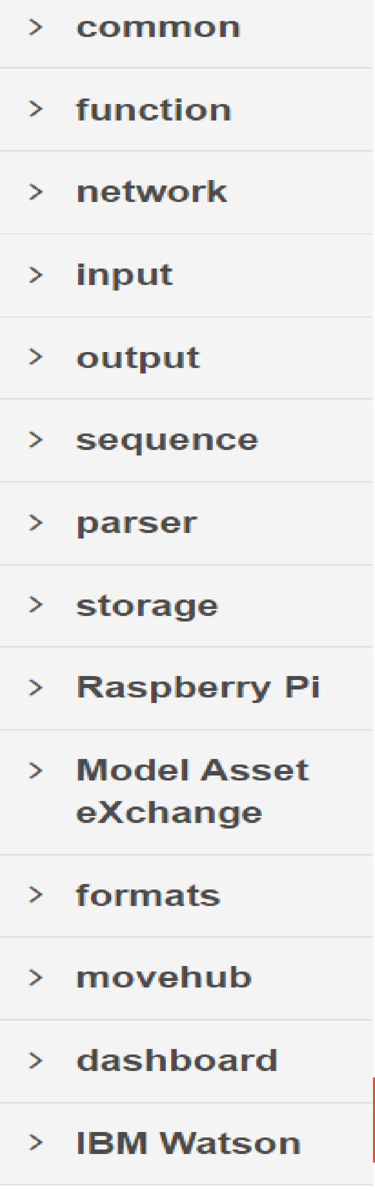
- Mysql/… databases
- Excel/… spreadsheets

Application fields

- IoT
- AI
- Crypto
- …

Home automation:

- Alexa
- Google Home
- Tuya smart
- Home appliances
- Fritz!Box routers
- Printers
- Vacuum (Roomba …)
- Washers (Miele …)
- …



Node Red to teach Computational Thinking?

PRO

- reuse and connection of functional units
- communication between different servers
- problem mapped to flow of information
- a flow is like a pipeline of transformations
- …

CON

- good programming Javascript required
- not obvious synchronization
- no data typing visualized
- …

Demo

EXAMPLES