* 1. *Modifiche Interfaccia Node-RED di Snap4City*

Le modifiche all’interfaccia di Node-RED consentono di importare nelle iotapp i nodi creati all’interno del Resource Manager (normali e di Data Analytics). Consentono anche di importare ed esportare nel Resource Manager i flussi di un utente in modo da pubblicarlo e renderlo disponibile per altri utenti. Tali modifiche sono presenti per la versione 1.1.3 di Node-RED.

Immagine che contiene schermata, testo, software

Descrizione generata automaticamente

Immagine che contiene testo, schermata, numero, software

Descrizione generata automaticamente

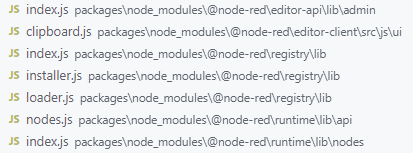
* 1. *Porting alle versioni 2.2.3 e 3.0.2*

Le operazioni necessarie per aggiornare Snap4City alla versione Node-RED 3.0.2 si limitano a trasferire tutte le sezioni di codice comprese tra i commenti //S4C nella versione ufficale di Node-RED 3.0.2, di fatti, questo si limita a coinvolgere esclusivamente la cartella **packages/node-modules/@node-red**. In particolare, i seguenti file modificati nella versione 1.1.3:

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

Risultano nelle seguenti modifiche:



Si può subito notare che le modifiche al file **red.js** devono essere riportate all’interno del file **clipboard.js** dato il cambio di gestione del modulo **@editor-client.** I file **S4Cnodes.js** e **S4CAuth.js** non compaiono in quanto non contengono i commenti //S4C ma sono comunque presenti nelle versioni aggiornate.

* 1. *Integrazione Progetto CAULDRON*

Tutte le modifiche apportate dal progetto CAULDRON, anche loro limitate alla cartella **@node-red**, sono comprese tra commenti //CAULDRON. In particolare, le modifiche riguardano i seguenti file:

A screenshot of a computer

Description automatically generated

La logica principale del progetto è contenuta nei file all’interno della cartella **packages\node\_modules\@node-red\editor-client\src\js\monitoring**. Per una documentazione più completa del progetto, leggere la documentazione ufficiale al seguente link <https://repositorio-aberto.up.pt/bitstream/10216/128964/2/414763.pdf>.

In sintesi, l’integrazione del progetto CAULDRON offre la possibilità di monitorare il passaggio dei messaggi entranti nei nodi di Node-RED, fornendo anche la possibilità di utilizzare breakpoint per bloccare il flusso attraverso un nodo per poterlo analizzare, farlo procedere step-by-step, rilasciare la coda di messaggi trattenuti dal breakpoint o cancellarla.

Una particolare aggiunta alla l’attuale versione di Cauldron permette di selezionare, attraverso una apposita textbox presente in ogni nodo, l’attributo da tracciare. La sintasi ammessa è quella usuale del Javascript quindi per un oggetto del tipo:

{

A:[

{

B:’0b’

},

{

B:”1b”

}

]

}

Digitare nella textbox l’espessione A[0].B mostra il valore ‘0b’ mentre l’espressione

A[1].B mostra il valore ‘1b’. Se l’espressione non corrisponde a nessun attributo dell’oggetto non vien mostrato alcun risultato.

Tale funzionalità può essere usata in modalità statica(‘static’) o dinamica(‘dynamic’): per modalità statica si intende che il cambio della textbox ha effetto dal prossimo messaggio che attraversa il nodo, mentre per modalità dinamica si intende che l’aggiornamento della textbox comporta un aggiornamento istantaneo del valore visualizzato a schermo.

Se la textbox rimane vuota, il valore visualizzato sarà quello di default, solitamente timestamp di arrivo del messaggio per i nodi generici.

Le feature appena discusse sono accessibili tramite una UI dropdown propria di ogni nodo, attivabile tramite il rispettivo pulsante di debug presente sul nodo.

Ai fini dell’estensione del grafico, si fa presente che esso memorizza e tiene di conto solo gli ultimi 10 messaggi ricevuti, e NON lo storico completo dall’attivazione del flusso.

* 1. *Dockerfiles*

Nella sotto-cartella **dockerfiles** sono presenti 6 Dockerfile per comporre le immagini di riferimento ovvero l’aggiornamento alla versione 3.0.2 di Node-RED con le modifiche di snap4city, l’aggiornamento alla versione 3.0.2 di Node-RED con snap4city e cauldron e l’ aggiornamento alla versione 2.2.3 di Node-RED con snap4city e cauldron. Per ogni scenario sono presenti due versioni: basic o user e advanced o developer.

I dockerfile per le versioni advanced utilizzano come base l’immagine **disitlab/snap4city-nodered-v2.2.2-adv:v10** mentre le versioni basic utilizzano l’immagine **disitlab/snap4city-nodered-v2.2.2-basic:v7**.

Tutti i dockerfile hanno la stessa struttura: viene eseguito un clone del tag riferito alla versione desiderata dalla repository **lorenzo-dominici/cauldron** su github dove è hostato il progetto. Vengono installate le dipendenze di cauldron e viene effettuato il build del progetto. In seguito viene copiata la directory **packages/node\_modules/@node-red** all’interno della cartella che contiene l’eseguibile (nella versione 3 è necessario copiare anche il modulo node-watch) e viene lanciato l’eseguibile con l’istruzione **npm start –userDir /data**.

Le immagini, inoltre, sono presenti su docker hub per un utilizzo immediato con i seguenti nomi:

- ipsdisit/snap4city-cauldron-v3.0.2-adv:v1

- ipsdisit /snap4city-cauldron-v3.0.2-usr:v1

- ipsdisit /snap4city-v3.0.2-adv:v1

- ipsdisit /snap4city-v3.0.2-usr:v1

- ipsdisit /snap4city-cauldron-v2.2.3-adv:v1

- ipsdisit /snap4city-cauldron-v2.2.3-usr:v1