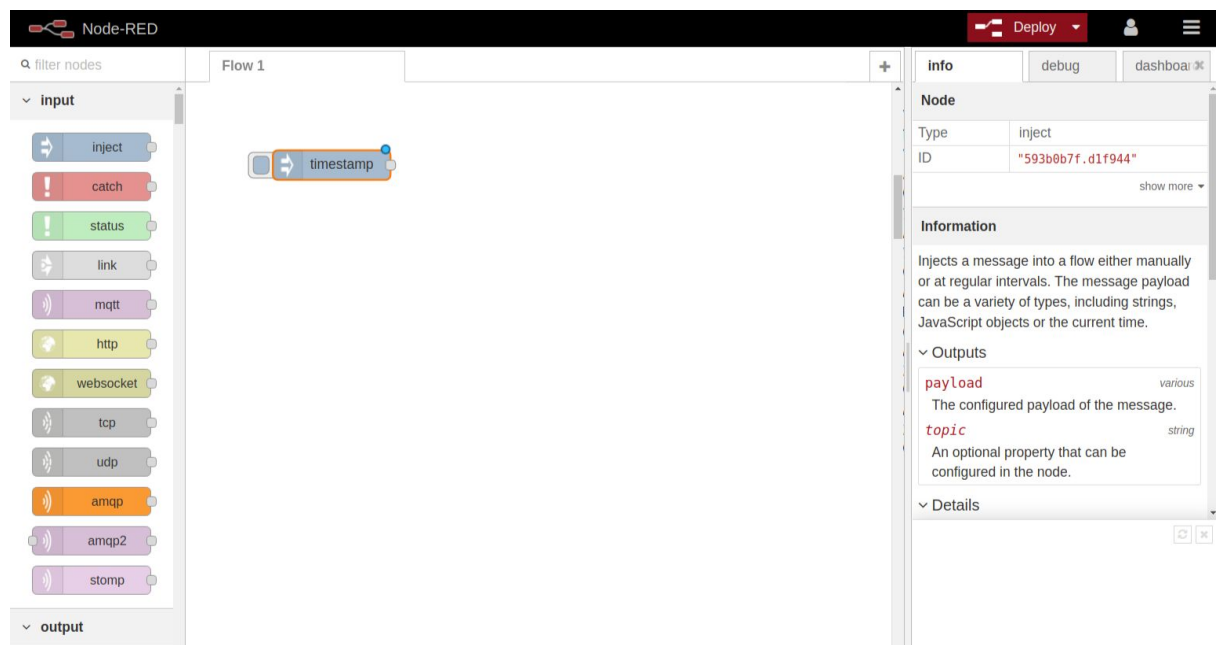
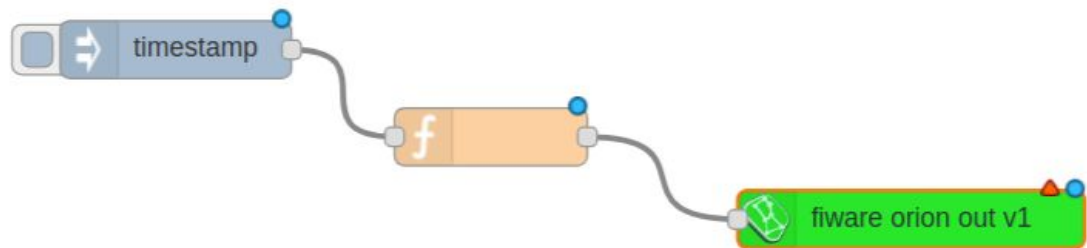


Inserire un dato IoT con una IoTApp

- andare su IOT Applications e creare una nuova IoTApp; l'application Type deve essere basic. E' possibile anche ri-usare una IoT Application che avete già creato; si consiglia di inserire in 1 IoTApp più di 1 procedura per poterle gestire in maniera più semplici e per non occupare troppe risorse del sistema
- l'IOTAPP ha dei blocchi nella colonna di sinistra; selezionare il block "inject" e trascinarlo nella board; Il blocco serve per poter avere un comando per far iniziare una serie di operazioni a cascata; dopo averlo trascinato cliccare su "deploy" per "pubblicare" l'app. A questo punto se cliccate sul "pulsante" alla sinistra del blocco il sistema "inietta" (inserisce) una variabile nel sistema, in questo caso inserisce la data corrente (timestamp) (compare in messaggio "Successfully injected: timestamp")



- per poter inserire un dato servono almeno tre blocchi; inserire i blocchi e collegarli tra di loro nel modo che si vede nell'immagine successiva
 - inject: già visto per dare il via alla procedura
 - function: è un blocco che permette di aggiungere codice javascript
 - fiware_orion_out_v1: questo blocco permette di inserire i dati in un contextBroker di tipo "Orion Context Broker" alla versione v1; questo è il device attualmente più diffuso, nella libreria ci sono dei blocchi dedicati a contextBroker di altre versioni



- cliccare sul blocco funzione (quello al centro) e scrivere la seguente funzione di inserimento dei dati; nell'esempio che mostriamo inseriamo ogni volta che si clicca su inject un dato con la data corrente e con dei valori fissi
 - **id**: il codice esatto del device che avete creato
 - **type**: il tipo che avete definito nel `iotModel` e dell'`iotDevice`
 - **name**: i nomi esatti degli attributi definiti nell'`iotModel`
 - **value**: i valori che vogliamo inserire; `new Date()` inserisce la data corrente

```
var data={
  "id": "test_diego_pisa",
  "type": "misura",
  "attributes":[
    {"name": "dateObserved", "value": new Date().toISOString(),
  "type": "time"},
    {"name": "temperature", "value": 32.1 , "type": "float"},
    {"name": "humidity", "value": 44.1 , "type": "float"}
  ]
}
var msg={payload: data}
return msg;
```

Attenzione il campo temporale (`dateObserved`) deve essere in formato testuale ISO: es: "2020-08-07T12:51:50.548Z"; IN JS da un campo `data` si ottiene la stringa ISO con la funzione `toISOString()`;

- Configurare il **Fiware orion out v1**
 - fare doppio click
 - inserire `orionUNIFI`
 - inserire `k1` e `k2` (se non li avete li trovate nella scheda del device)

Edit fiware orion out v1 node

▼ **node properties**

Service	orionUNIFI	
Certificates	Add new tls-config...	
Device type	Weather	
Device NameID	TestMichele_TOS11000069	
key 1	baf33d30-4592-4f7e-ad15-31a946b30b9d	
key 2	a2574cb0-683b-4158-a24d-a1d7b5a27df5	
apikey		
auth		
Name	node-red-contrib-snap4city-user/fiware-orion:com	

Dove il service va modificato con la matita inserendo **orionUNIFI** come nella seguente figura

Broker URL	broker1.snap4city.org
port	8080
Name	orionUNIFI

Importante: E' possibile passare l'autenticazione del sensore nel messaggio che viene inviato al Fiware orion out v1; per farlo è sufficiente preparare l'oggetto nel seguente modo:

```
var k1 = "f6efd060-375b-4f40-af09-41e0814f7039";
var k2 = "108bfe91-8140-41fd-92d8-c2b317c3cef8";
...
var auth={"k1": k1,"k2": k2,"apikey":"apikey","basicAuth": "basicAuthKey"};
return {"auth": auth, "payload":ret}
```