

NOI A.G. / S.p.A.
 Roberto Cavaliere
 r.cavaliere@noi.bz.it
 T +39 0471 066 676

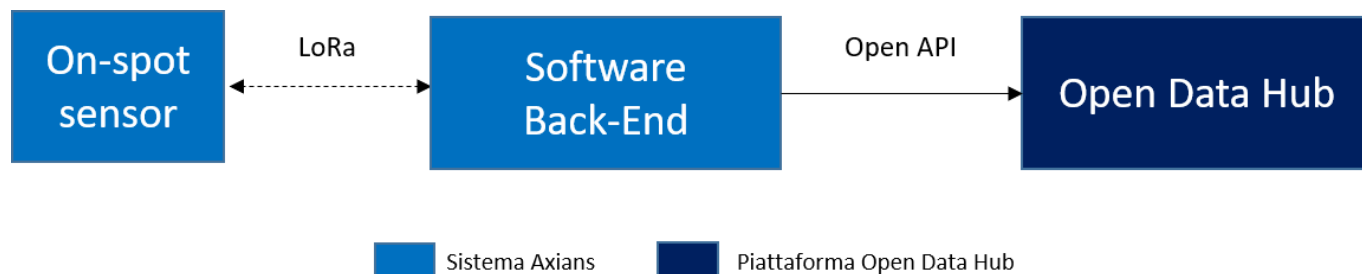
On-Street Parking BZ Data Collector **v1.0, 18.02.2020**

Note preliminari	1
Dettagli del flusso dati messo a disposizione dal data provider	1
End-point e meccanismo di autenticazione	3
Metadati stazioni	3
Metadati tipi	4
Dati	5

Note preliminari

Il Comune di Bolzano ha avviato una sperimentazione con il fornitore Axians (<https://www.axians.it/it/>) di un sistema di rilevamento degli stalli di occupazione dei parcheggi a bordo strada (di seguito “on-street parking”, per differenziarlo dai dati di “off-street parking” relativi alle aree di parcheggio, già integrate nell’Open Data Hub.

Il sistema di Axians prevede l’installazione di un sensore per stallo di parcheggio, ciascuno connesso via tecnologia LoRa ad un sistema di back-end.



Dettagli del flusso dati messo a disposizione dal data provider

I dati sono messi a disposizione dell’Open Data Hub via API con messaggi HTTP POST in logica push, formattati come di seguito riportato.

```
{
  "devEUI": "001bc50670100539",
  "appEUI": "70b3d59b60000003",
  "fPort": 1,
  "gatewayCount": 1,
  "rssi": -51,
  "data": "gAAAAyIQ8/kBAAAD1FY6ADUAABWZAMk6",
  "parsedData": {
    "FW-version": 58,
    "Magnitude": 980,
    "Payload": "KEEPALIVE",
    "Progtx": 5529,
    "Retries": 0,
    "Rssi": 53,
    "Sensor-address": "00000322",
    "Status": "free"
  },
  "loRaSNR": 10.8,
  "frequency": 868100000,
  "dataRate": {
    "modulation": "LORA",
    "spreadFactor": 12,
    "bandwidth": 125
  },
  "devAddr": "e8a0b67e",
  "fCntUp": 1,
  "time": "2019-10-08T06:16:14.385035347Z",
  "encrypted": false,
  "format": "base64",
  "gateways": [
    {
      "mac": "00800000a0004693",
      "time": "0001-01-01T00:00:00Z",
      "timestamp": 3625854332,
      "frequency": 868100000,
      "channel": 0,
      "rfChain": 1,
      "crcStatus": 1,
      "codeRate": "4/5",
      "rssi": -51,
      "loRaSNR": 10.8,
      "size": 37,
      "dataRate": {
        "modulation": "LORA",
        "spreadFactor": 12,
        "bandwidth": 125
      }
    }
  ]
}
```

Il messaggio contiene molti dati specifici della trasmissione LoRa, ma ai fini dell'integrazione dell'Open Data Hub solo i seguenti campi sono rilevanti (evidenziati in **giallo** nel record dati):

- **Sensor-address**, che indica in maniera univoca il sensore analizzato
- **Status**, che indica la presenza di un veicolo in sosta o meno.

- **time**, che indica l'istante di interrogazione del sensore.

L'API non espone alcuna informazione di anagrafica come la posizione esatta in cui il sensore è installato. Per gestire quest'aspetto, essenziale per qualsiasi utilizzo del dato, l'applicativo dovrà leggere questi metadati da un file di supporto, strutturato come indicato qui:

<https://docs.google.com/spreadsheets/d/1gPWuvmYntWj3IXfZ0uSHm5xNU23IIgVCcPIBb9kK0A0/edit?usp=sharing>

Il data provider è responsabile della correttezza e del continuo aggiornamento di queste informazioni.

End-point e meccanismo di autenticazione

Attualmente i dati vengono trasmessi in chiaro all'end-point <http://logger.testingmachine.eu>

Specificare come i dati in ingresso sono gestiti e come il Data Collector può leggerli in maniera corretta.

Metadati stazioni

I metadati delle stazioni sono da salvare all'interno della tabella "station" del database. Si propone di introdurre una nuova tipologia di stazione, chiamata "**ParkingSensor**", da differenziare rispetto alla tipologia "ParkingStation" che è associata ad un'area di stazione. Ogni stazione associata a questa nuova tipologia deve essere mappata come segue:

Campi file di supporto	Colonne tabella stations del database
sensor_address	stationcode
id	name
Longitude	pointprojection
Latitude	pointprojection

Tabella 1: Mapping tra gli attributi del file di supporto e la tabella "station" del database per la gestione della tipologia di stazione "ParkingSensor".

Alcune osservazioni di dettaglio:

- i valori delle colonne **active** e **available** sono da settare a **true**.
- il valore della colonna **parent_id** è da valorizzare con l'ID della ParkingStation a cui una stazione di tipo ParkingStationSpot appartiene (vedi sotto);
- il valore della colonna **origin** è da settare con **AXENTIA**.
- il valore della colonna **stationtype** nel nostro database è da settare come **ParkingSensor**;
- i valori delle colonne **id** e **meta_data_id** (id corrispondente nella tabella metadata) vanno gestiti automaticamente in fase di scrittura del nuovo record nella tabella.

L'attributo **description** del file di supporto è da salvare come json nella tabella **metadata**, creando un opportuno riferimento con la stazione di riferimento (colonna **station_id**). All'interno di questa tabella va anche inserito l'attributo "**capacity**", che deve essere settato a 1 per ogni stazione.

Per facilitare la visualizzazione a livello utente, gruppi di sensori di occupazione di parcheggio sono raggruppati tra loro. In particolare, il campo "**group**" serve a raggruppare una serie di sensori installati a breve vicinanza tra loro. Un gruppo di sensori è quindi da gestire come una "ParkingStation" tradizionale, simile a quelle già a sistema, secondo questa logica:

Campi file di supporto	Colonne tabella stations del database
group	stationcode
group	name
Longitude*	pointprojection
Latitude*	pointprojection

Tabella 2: Mapping tra gli attributi del file di supporto e la tabella "station" del database per la gestione della tipologia di stazione "ParkingStation".

* Per quello che riguarda la longitudine e latitudine, essa non viene resa disponibile a livello di gruppo. Si propone di associare a questa stazione "fittizia" la posizione media di tutte le stazioni di tipo "ParkingSensor" associate.

Alcune osservazioni di dettaglio:

- i valori delle colonne **active** e **available** sono da settare a **true**.
- il valore della colonna **parent_id** non è da valorizzare;
- il valore della colonna **origin** è da settare con **AXENTIA**.
- il valore della colonna **stationtype** nel nostro database è da settare come **ParkingStation**;
- i valori delle colonne **id** e **meta_data_id** (id corrispondente nella tabella metadata) vanno gestiti automaticamente in fase di scrittura del nuovo record nella tabella.

In questo caso l'unico record da salvare nella tabella **metadata** è l'attributo "**capacity**", che deve essere calcolato come numero delle stazioni di tipo ParkingSensor associate.

Metadati tipi

L'unica informazione disponibile riguarda lo stato di occupazione dello stallone. Si propone quindi di utilizzare per entrambe le tipologie di stazioni il tipo già esistente "**occupied**", che indica il numero di posti occupati. Da notare come poi saranno associati anche tutti i tipi generati dalla catena di previsione di occupazione di parcheggio (che dovrebbero essere generate in automatico una volta che le nuove stazioni di tipo ParkingStation sono a sistema; per le stazioni di tipo ParkingSensor va effettuata una piccola modifica della catena di elaborazione).

Dati

A causa dell'attesa variabilità temporale dei dati, si propone di generare un nuovo record dati ad ogni messaggio POST inviato dal sistema di back-end di Axentia. I dati sono da salvare nelle tabelle **measurement** (in cui vengono salvati solo i valori più recenti) e **measurementhistory**. Alcune osservazioni di dettaglio:

Alcune osservazioni di dettaglio:

- Il valore della colonna **period** nel nostro database va settata con la **frequenza media di trasmissione dei messaggi POST [da verificare]**
- Il valore della colonna **id** va gestito automaticamente in fase di scrittura del nuovo record nella tabella.
- Il valore della colonna **created_on** va gestito automaticamente in fase di scrittura del nuovo record nella tabella (è il timestamp relativo all'operazione di scrittura).
- Il valore della colonna **timestamp** corrisponde al valore dell'attributo "**time**" fornito dal web-service Axentia;
- Il valore della colonna **double_value** è da valorizzare con il valore restituito dal web-service Axentia. Da notare che va effettuato un valore di mapping rispetto a quanto esposto (tipicamente una stringa del tipo "FREE" / "OCCUPIED"). Per quello che riguarda i dati della stazione di "gruppo" (tipo ParkingStation), vanno sommati i valori di occupazione di tutte le ParkingSensor associate.
- Il valore della colonna **provenance** è da valorizzare rispetto all'ID del record corrispondente nella tabella provenance.
- I valori delle colonne **station_id** e **type_id** sono da valorizzare in maniera appropriata in fase di scrittura dei record.

VERIFICARE USER EXPERIENCE NELLA FRUIZIONE DEI DATI SUGLI APPLICATIVI PARKING.BZ.IT E MOBILITY.MERAN.EU