



MENTOR

Mobilize Efforts for a harmonized diffusion of New smart and shared Mobility Technologies in the Overall program Regions

ID: 562850

Bike Sharing Bolzano Data Collector Specifiche di integrazione nell'Open Data Hub

v1.1, 04.01.2020

Capofila IT / Lead Partner IT	Comune di Merano / Gemeinde Meran
Capofila CH / Lead Partner CH	Comune di Brig-Glis / Brig-Glis
Partner 1 IT	NOI Techpark
Partner 2 IT	SASA
Partner 1 CH	Postauto



STADTGEMEINDE MERAN
COMUNE DI MERANO



TECHPARK SÜDTIROL / ALTO ADIGE



Indice

Note preliminari.....	4
Metadati stazioni.....	4
Metadati tipi.....	5
Dati.....	7

Note preliminari

Alla luce della struttura dell'API del Data Provider (Ecospazio), si è deciso in fase di avvio dei lavori di scegliere l'approccio di modellare sia le stazioni di ritiro delle biciclette, sia ogni stallo della stazione, sia le singole biciclette come "station", secondo il concetto usato nell'Open Data Hub.

End-point e meccanismo di autenticazione

Gli end-point alla piattaforma Ecospazio sono:

- **Anagrafica completa delle stazioni:** <https://maas.ecospazio.it/api/v1/stations>
- **Stato di una singola stazione:**
https://maas.ecospazio.it/api/v1/stations/{id_stazione}

Per effettuare le chiamate è necessario includere un **Header Token**, fornito separatamente.

Il dettaglio di ogni singola bicicletta è disponibile nel secondo metodo messo a disposizione dall'API, che fornisce anche i dati in tempo reale.

Metadati stazioni

Attraverso il metodo "/stations" sono disponibili i metadati essenziali di ciascuna **stazione** che è parte del servizio di bike sharing. Le informazioni da memorizzare nella tabella station sono quelle indicate in .

Campi file webservice	Colonne tabella stations del database
id	stationcode
name	name
lat, lng	pointprojection

Tabella 1: Mapping tra gli attributi del web-service "stations" della piattaforma Ecospazio e la tabella "station" del database (metadati stazioni bike sharing).

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 **BIKE_SHARING_BOLZANO**.
- il valore della colonna **stationtype** nel nostro database è da settare come **BikesharingStation**;
- 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'unico attributo aggiuntivo da salvare come json nella tabella **metadata**, creando un opportuno riferimento con la stazione di riferimento (colonna station_id), è il campo

“**address**”. Il campo “**state**” viene gestito tra i tipi associati alla stazione, come successivamente trattato. Nella tabella metadata va anche salvato il numero di stalli presenti nella stazione, fornito nel secondo metodo (campo “**totalBays**”).

Attraverso il metodo “/stations/{id_stazione}” sono disponibili i metadati essenziali sia di ogni stallo che caratterizza una stazione (ossia il punto di prelievo), sia di ciascuna bici che è parte del servizio di bike sharing.

Le informazioni da memorizzare nella tabella station sono quelle indicate in Tabella 2 ed in Tabella 3.

Campi file webservice	Colonne tabella stations del database
label	stationcode
label	name
lat, lng	pointprojection

Tabella 2: Mapping tra gli attributi del web-service “stations/{id_stazione}” della piattaforma Ecospazio e la tabella “station” del database (metadati stalli delle stazioni del servizio di bike sharing).

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 stazione di cui lo stallo è parte.
- il valore della colonna **origin** è da settare con **BIKE_SHARING_BOLZANO**.
- il valore della colonna **stationtype** nel nostro database è da settare come **Bicyclestationbay** (è un nuovo tipo nell'Open Data Hub).
- 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;

Come indicato, si propone di usare il contenuto dell'attributo label sia per valorizzare il contenuto della colonna “stationcode” e “name”. Le coordinate sono da valorizzare con quelle della stazione a cui appartiene.

L'API fornisce degli attributi aggiuntivi da salvare come json nella tabella **metadata**, creando un opportuno riferimento con la stazione di riferimento (colonna station_id). Tali attributi sono i campi “**charger**” (se TRUE indica se è possibile caricare una bici elettrica) e “**use**” (indica il tipo di utilizzo del mezzo parcheggiato in questo stallo). Gli altri attributi disponibili sono informazioni in tempo reale e quindi vanno gestite come tipi.

Campi file webservice	Colonne tabella stations del database
code	stationcode
vehicleType->name	name
lat, lng	pointprojection

Tabella 3: Mapping tra gli attributi del web-service “stations/{id_stazione}” della piattaforma Ecospazio e la tabella “station” del database (metadati biciclette del servizio di bike sharing).

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 dello stallo in cui la bici è parcheggiata.
- il valore della colonna **origin** è da settare con **BIKE_SHARING_BOLZANO**.
- il valore della colonna **stationtype** nel nostro database è da settare come **Bicycle**.
- 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'API fornisce un attributo aggiuntivo da salvare come json nella tabella **metadata**, creando un opportuno riferimento con la stazione di riferimento (colonna **station_id**). Si tratta nello specifico dell'attributo **“electric”** (se TRUE indica che si tratta di una bici elettrica) Gli altri attributi disponibili sono informazioni in tempo reale e quindi vanno gestite come tipi. Da notare che il servizio di bike sharing in questione è one-way, per cui l'associazione bici-stazione può essere estremamente variabile. Per storicizzare le informazioni su tali associazioni per future analisi statistiche, si propone di salvare i parametri **lat**, **lng** anche nella tabella metadata.

Metadati tipi

I metadati dei tipi gestiti dal sistema vanno salvati nella tabella *type* del database. I tipi da salvare sono i seguenti:

Stazioni di tipo Bicyclestation:

- **availability** (indica se la stazione è operativa). Va riutilizzato il tipo “availability” già gestito nell'Open Data Hub. Va associata l'informazione presente nell'attributo **“state”**.
- **free_bays** (indica quanti stalli sono liberi. E' un'informazione importante per il pubblico per sapere dove poter lasciare una bici in utilizzo). **Va creato un nuovo tipo “spots_availability”**, complementare ad “number-available” (quest'ultimo si riferisce ad elementi presenti, questo nuovo a spazi liberi presenti, un'informazione che è in qualche modo duale). Va associata l'informazione presente nell'attributo **“freeBay”**.
- **available_vehicles** (indica quante bici sono parcheggiate presso la stazione). Va riutilizzato il tipo “number-available” già gestito nell'Open Data Hub. Va associata l'informazione presente nell'attributo **“availableVehicles”**.

Stazioni di tipo Bicyclestationbay:

- **availability** (indica se lo stallo è effettivamente utilizzabile). Va riutilizzato il tipo “availability” già gestito nell’Open Data Hub. Va associata l’informazione presente nell’attributo “**state**” all’interno della struttura “**bays**”.
- **usageState** (indica lo stato dello stallo). E’ una stringa che può assumere valori “PRELIEVO” (se c’è una bici parcheggiata ed utilizzabile dall’utenza), “RICONSEGNA” (se non c’è una bici e quindi è disponibile per ospitare la riconsegna di una bici in uso), “” (se lo stallo non è disponibile). Va associata l’informazione presente nell’attributo “**usageState**”.

Stazioni di tipo Bicycle:

- **availability** (indica se la bici è effettivamente disponibile). Non c’è un attributo specifico associato, ma l’informazione è da associare a TRUE se il veicolo viene rilevato presso uno degli stalli di una stazione.
- **batteryState** (indica lo stato della batteria). E’ una stringa che può assumere ad es. il valore “CHARGED” (se la bici è carica). Va associata l’informazione presente nell’attributo “**batteryState**”.

Dati

Lo stato delle diverse tipologie di stazione è da aggiornare in maniera periodica tramite chiamate GET (meccanismo pull) Si propone di configurare inizialmente il sistema con chiamate all’API Ecospazio ogni 5 minuti.

I dati sono da salvare nelle tabelle measurement (in cui vengono salvati solo i valori più recenti) e measurementhistory. Da notare che alcuni valori sarebbero in realtà delle stringhe, ma essendo logicamente degli oggetti booleani si propone di associare agli stati forniti i valori “0” e “1”. Alcune osservazioni di dettaglio:

- Il valore della colonna **period** nel nostro database va settata con **300** (vedi considerazioni sopra).
- 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). Dal momento che l’API non fornisce alcuna indicazione sul timestamp in quanto i dati rappresentano lo stato attuale delle bici, tale valore è da memorizzare anche per la colonna **timestamp**.
- Il valore della colonna **double_value** è da valorizzare con il valore restituito dall’API Ecospazio.
- 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.