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

Bike Boxen STA

~~v1.4~~v2.1, ~~0830.0306~~.2023

Preliminary notes

STA is implementing a network of bike parking systems in South Tyrol, to be placed mainly just outside train stations. The idea is to foster intermodal sustainable trips by combining individual bikes to be parked in a secured way in these new parking facilities with trips carried out by train.

The technological system for the management of this infrastructure was assigned to Bicincittà (https://www.bi-cincitta.com), one of the most known companies at national level in relation to bike sharing systems and services, which has started to enter also the bike parking market.

Within the public tender STA has requested to Bicincittà the provide an API to the Open Data Hub for the retrieval of real-time information related to the bike parking availability of the different parking areas. Bicincittà has put at disposal an API which is used to share not only real-time information about all bike sharing and parking implementations, but also to carry out more complex functionalities like booking or payment.

Bicincittà end-point description

The end-point is available at the following URL: https://stapro.dnsalias.com~~http://stabic.dnsalias.com~~

The access credentials have been made available separately by Bicincittà. The requests to the API need to have a valid token.

The API implemented by Bicincittà supports different use cases, including booking. For the Open Data Hub, only the method /resources is relevant. More specifically, the following services have to be considered:

- **/resources/locations**: this API call allows to retrieve the cities where the parking service is present (not only South Tyrol)
- **/resources/stations**: this API call allows to retrieve the list of stations present in the specified city and its overall real-time state.
- **/resources/station**: this API call allows to retrieve the list of parking slots associated to a station and its real-time state.

Specification of the modalities of integration in the Open Data Hub

METADATA – BIKE PARKING STATION

The proposal is to insert all this static data in the "station" table. All the data of Bicincittà should be imported (i.e. for all available cities), not only for South Tyrolean cities. The Data Collector should therefore first make a request to the method /resources/locations, get the codes of all available stations, and then make a separate call for each location to the method /resources/stations to get the details of all associated stations. Please note that the web-service provides names in different languages; for the locations web-service, it is sufficient to use one of the supported languages and make the API request only once, while for the stations web-service it is necessary to call the method for each different language so to have the station names in all different languages.

The following proposed mapping takes as reference the fields provided the method /resources/stations. All fields marked as "METADATA" indicate the necessity to have a linked record in the metadata table in which these values have to be stored. Please note that some fields provided by the method stations are measurements and are not to be considered among the metadata.

Web-service		Web-service fields		Open Data Hub parameters	
locations		idStationstationId idStation		stationcode	stationcode
locations		locationID name		METADATA	name
stations		Nname (languageID = it) address		name	METADATA
stations		name (languageID = itldel, longitude		METADATA	pointprojection
stations		name (languageID = itlld) type		METADATA	METADATA. Store directly the associated mapping (0 = Sharing with real stations; 1=Sharing with virtual stations, 2=Parking)
stations		name (languageID = enit)		METADATA	
stations		address	urlGuide	METADATA	METADATA
stations		latitude, longitude totalPlaces		pointprojection	METADATA
locations		type stationPlaces		METADATA. Store directly the associated mapping (0 = Sharing with real stations, 1=Sharing with virtual stations, 2=Parking1 = veloHub, 3 = bikeBoxGroup)	METADATA. The field "state" should not be stored since it is a measurement of the associated ba
		urlGuide maxDistanceRent		METADATA	METADATA
stations		totalPlaces		METADATA	

Web-service	Web-service fields	Open Data Hub parameters
	stationPlaces	~~METADATA. The field „state" should not be stored since it is a measurement of the associated bay~~
	maxDistanceRent	~~METADATA~~

Table 1: Mapping between main web-service and Open Data Hub fields (reference: "station" table- bike parking station).

The following specifications have to be also considered:

- the Open Data Hub field **origin** is to set as **BICINCITTA**.
- the Open Data Hub field **stationtype** is to set as **BikeParking**

METADATA – BIKE PARKING BAY

Important note: the method /resources/station provides the information for each bay, similarly to the Data Collector "Bike Chargers". For this reason, it is proposed to save each bay a station in the "station" table, and the use the field "parent_id" to store the hierarchy of the stations. The reference mapping has to be considered. Please note that certain attributes made available through this station refer to the bike parking station and not the bay, so they are stored only for the above type of station.

Web-service fields	Open Data Hub parameters
idStation_stationPlaces/position	stationcode
name_stationPlaces/position	name
latitude, longitude	pointprojection
stationPlaces/type	METADATA. Store directly the associated mapping (1=~~Normal bay~~withoutRefill, 2=~~Bike box~~withRefill). ~~To not be saved in case the associated bike parking station is of type = 0 (Sharing with real stations) or type = 1 (Sharing with virtual stations)~~
stationPlaces/~~isAssisted~~level	METADATA

Table 2:   *Mapping between main web-service and Open Data Hub fields (reference: "station" table- bike parking bay).*

The following specifications have to be also considered:

- the Open Data Hub field **stationtype** is to set as **BikeParkingBay**

DATA

The associated real-time state for the two types of stations is provided through certain fields in the above mentioned methods. Where possible, existing types already available in the Open Data Hub should be reused. The following measurements have to be stored, with reference the type of data and the associated station.

Web-service fields (/stations)	Measurement type
state	Existing type ‚**state**'. Reference mapping to be stored (1=in service, 2= out of service 1=FREE,2=OCCUPIED,3=OUT OF SERVICE) Table: **measurementstring**
countFreePlacesAvailable count_MuscularBikesAvailable	New type ‚**availableMuscularBikes**'. Table: **measurement**
countFreePlacesAvailable _countAssistedBikesAvailable	New type ‚**availableAssistedBikes**'. Table: **measurement**
countFreePlacesAvailablecountFreePlacesAvailable	Existing type ‚ **availableVehicles** '. Table: **measurement**

Table 3: Measurements to be stored for a bike parking station.

Web-service fields (/station)	Measurement type
state	Existing type ‚**usageState**'. Reference mapping to be stored (1=in service, 2= occupied – in service; 3 = out of service 1=FREE,2=OCCUPIED,3=OUT OF SERVICE) Table: **measurementstring**

Table 4: Measurements to be stored for a bike parking bay.

Formatierte Tabelle