

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

E-Charging Station – static data collection (Bayern, Tyrol)

v1.2, 27.07.2023

Preliminary information

The regions of Bayern and Tyrol share static information about their available e-charging stations. This data is made available through the organization DZT, <https://www.germany.travel/de/ueber-uns/die-dzt.html>

Reference API calls

Request reference of all available e-charging stations

```
GET https://proxy-dzt-pre-production.onlim.com/api/ts/v2/kg/things
?kw=&
filterDs=https://semantify.it/ds/E85TgOxMg
Content-Type: application/json
x-api-key: *****
```

Request data of a single e-charging station

```
GET https://proxy-dzt-pre-production.onlim.com/api/ts/v1/kg/things/50861002
?ns=http%3A%2F%2Fonlim.com%2Fentity%2FLadestationen-Api-Bund%2FECarChargingStation%2F
Content-Type: application/json
x-api-key: *****
```

(Please note that the provided API Key is not reported in this document for security purposes).

The first call just contains the reference to the charging stations that are made available, so that the second call is possible to be carried out. The following specifications refer to the data of a single e-charging station retrieved through this second request.

Pagination

Since the list of available data is quite large, a pagination mechanism is available. In the first above described request it is possible to ask a specific page and a specific page size, considering that the API can provide maximum 50 stations per request. Such request details can be provided as follows:

```
curl --location 'https://proxy.opendatagermany.io/api/ts/v2/kg/things?filterDsList=https%3A%2F%2Fsemantify.it%2Fds%2FE85TgOxMg' \
--header 'page: 3' \
--header 'page-size: 5' \
--header 'Content-Type: application/ld+json' \
--header 'x-api-key: ****' \
--data-raw '{
  "@context": {
    "ometa": "http://onlim.com/meta/",
    "sq": "http://www.onlim.com/shapequery/",
    "@vocab": "http://www.onlim.com/shapequery/"
  },
  "sq:query":[
    {
      "ometa:dateModified":{
        "sq:value":"2023-02-23T19:00:00",
        "sq:op": ">",
        "sq:datatype": "dateTime"
      }
    }
  ]
}'
```

What is important to point out, through this call it is possible to get only the stations that have had a modification in their data from a certain timestamp. This can be useful in particular in case of data update, since it is possible to use this feature to just ask and then update only the stations that have had some variations. Since the data available is not real-time data, it is very likely that the such new data retrieval will update some few stations, without the necessity to request and update, again and again, all the whole set of data.

For this request, the response looks like:

```
{
  "metaData": {
    "dsCount": {
      "https://semantify.it/ds/E85TgOxMg": {
        "count": "32730",
        "name": {
          "en": "e-chargingstation"
        }
      }
    },
    "page-size": 5,
    "sortSeed": "13404025-38dc-4d11-947a-f5245a29bd95",
    "total": 32730,
    "current-page": 3
  },
  "data": [
```

```
{
  "@type": [
    "https://odta.io/voc/EChargingStation"
  ],
  "https://vocab.sti2.at/ds/compliesWith": {
    "@id": "https://semantify.it/ds/E85TgOxMg"
  },
  "@id": "http://onlim.com/entity/Ladestationen-API-Bund/ECarChargingStation/65102546"
},
{
  "@type": [
    "https://odta.io/voc/EChargingStation"
  ],
  "https://vocab.sti2.at/ds/compliesWith": {
    "@id": "https://semantify.it/ds/E85TgOxMg"
  },
  "@id": "http://onlim.com/entity/Ladestationen-API-Bund/ECarChargingStation/88321013"
},
{
  "@type": [
    "https://odta.io/voc/EChargingStation"
  ],
  "https://vocab.sti2.at/ds/compliesWith": {
    "@id": "https://semantify.it/ds/E85TgOxMg"
  },
  "@id": "http://onlim.com/entity/Ladestationen-API-Bund/ECarChargingStation/67851203"
},
{
  "@type": [
    "https://odta.io/voc/EChargingStation"
  ],
  "https://vocab.sti2.at/ds/compliesWith": {
    "@id": "https://semantify.it/ds/E85TgOxMg"
  },
  "@id": "http://onlim.com/entity/Ladestationen-API-Bund/ECarChargingStation/67851451"
},
{
  "@type": [
    "https://odta.io/voc/EChargingStation"
  ],
  "https://vocab.sti2.at/ds/compliesWith": {
    "@id": "https://semantify.it/ds/E85TgOxMg"
  },
  "@id": "http://onlim.com/entity/Ladestationen-API-Bund/ECarChargingStation/73021005"
}
]
```

The field 'total' has to be used to consider the amount of pages that are available, according to the page size set (recommendation: page size = 50), and implement all the necessary calls to retrieve all available charging stations.

Integration in the Open Data Hub

The mapping between the fields supported by the web-service and the structured data model used in the Open Data Hub is summarized in the following table. This data is saved both as EChargingStation as well as EChargingPlug in the **station** table of the Open Data Hub database. More in detail:

EChargingStation:

Substructure	Description
address/addressCountry	METADATA
address/addressLocality	METADATA
address/postalCode	METADATA
address/ streetAddress	METADATA
geo/latitude geo/longitude	pointprojection
identifier/value/value	stationcode
name	name

Additional notes:

- 'METADATA' indicates that the correspondent fields have to be stored in the metadata table
- Only the attribute name needs to be stored (e.g. addressCountry), i.e. it is not necessary to keep the reference to the reference structure as defined in schema.org
- **origin:** set as 'DZT'
- **stationtype:** set as 'EChargingStation'

EChargingPlug:

- As for the other available static charging station, the value for the field '**name**' in the Open Data Hub database can be an increasing integer, depending on the number of available power structures in the API
- The field '**stationcode**' could be set as this: stationcode of the parentStation with the value name appended (e.g. 'Officina Elettrica San Vigilio di Marebbe SPA:E-Werk St. Vigil/Officina Elettrica San Vigilio:1').
- The position of the EChargingPlug is to be set as the one of the parent station.
- Each EChargingPlug Station should have a linked metadata record, in which the outlet information is stored. In that record it is expected to have the information provided through the API, in particular the field socket in the structure power (to be associated to "outletTypeCode", if we want to keep the current values of the existing static charging stations).
- **origin:** set as 'DZT'
- **stationtype:** set as 'EChargingPlug'