|  |  |
| --- | --- |
| NOI A.G. / S.p.A.Roberto Cavalierer.cavaliere@noi.bz.itT +39 0471 066 676 |  |

**A22 web-site Data Collectors**

**v1.1, 12.09.2023**

[**Preliminary notes** 1](#_Toc145429453)

[**Authentication** 2](#_Toc145429454)

[**Web-cams Data Collector** 2](#_Toc145429455)

[End-point description 2](#_Toc145429456)

[Integration specification 2](#_Toc145429457)

[**Rest Areas Data Collector** 2](#_Toc145429458)

[End-point description 2](#_Toc145429459)

[Integration specification 2](#_Toc145429460)

[**Toll gates Data Collector** 2](#_Toc145429461)

[End-point description 2](#_Toc145429462)

[Integration specification 2](#_Toc145429463)

[**Traffic Forecasts Data Collector** 3](#_Toc145429464)

[End-point description 3](#_Toc145429465)

[Integration specification 3](#_Toc145429466)

[**Traffic Events Data Collector** 4](#_Toc145429467)

[**Roadworks Events Data Collector** 4](#_Toc145429468)

## **Preliminary notes**

Autostrade del Brennero has decided to share additional data with the Open Data Hub, and it is case to let it be available as open data for 3rd parties. These additional datasets refer to data which are publicly accessible on the A22 web-site (<https://www.autobrennero.it>), i.e.:

* **Web-cams**: placed along the A22 highway, can provide a visual feedback of the current traffic conditions
* **Rest areas**: placed along the A22 highway, can be useful information for travelers along the highway. For each rest areas the available services are provided.
* **Toll gates**: information about the details of the toll gates available.
* **Traffic forecasts**: available traffic bulletins computed and published by Autostrade del Brennero
* **Traffic events**: real-time traffic events published on the A22 web-site
* **Roadworks events**: real-time traffic events published on the A22 web-site

## **Authentication**

The access to the different datasets is protected by a simple authentication mechanism (HTTP BASIC AUTH). The credentials (username + password) are available and are shared internally in a secure way.

## **Web-cams Data Collector**

### End-point description

The end-point that provides this data is the following (method “**GetWebCam**”): <https://www.autobrennero.it/webservice/WSOpenData.asmx/GetWebCam>

### Integration specification

This data will be integrated in the Tourism Data Space, similarly as other web-cams.

## **Rest Areas Data Collector**

### End-point description

The end-point that provides this data is the following (method “**GetAreeDiServizio**”): <https://www.autobrennero.it/webservice/WSOpenData.asmx/GetAreeDiServizio>

### Integration specification

This data will be integrated in the Tourism Data Space, as POIs.

## **Toll gates Data Collector**

### End-point description

The end-point that provides this data is the following (method “**GetCaselli**”): <https://www.autobrennero.it/webservice/WSOpenData.asmx/GetCaselli>

### Integration specification

This data will be integrated in the Tourism Data Space, as POIs.

## **Traffic Forecasts Data Collector**

### End-point description

The end-point that provides this data is the following (method “**GetPrevisioniTrafficoComplete**”): <https://www.autobrennero.it/webservice/WSOpenData.asmx/GetPrevisioniTrafficoComplete>

The data which is provided through this API is actually what is shown on the A22 web-site:



Additionally, to this, the API provides the traffic forecasts also with reference the main locations crossed by the highway, which are actually Brennero, Bolzano and Verona. The API is structured in a way so that for each day all forecasts for each section (location + travel direction, i.e., North + South, the forecast is retrieved). The forecast is provided as a number, following this convention:

* 0 = regular (green)
* 1 = heavy (yellow)
* 2 = severe (red)
* 3 = critical (black)

### Integration specification

In order to properly this integrate this data according to the Open Data Hub data model, the following strategy is considered. A “station” (as in the Open Data Hub concept) is defined as a specific location + travel direction. For example, we should have the following stations (parameter ‘**name’** in the table “station” of the Open Data Hub core):

* Brennero\_Nord
* Brennero\_Sud
* Bolzano\_Nord
* …..

The name of the stations shall be automatically computed by considering the fields “Nord” and “Sud” and the locations provided in the associated structures.

For the other metadata foreseen in the table “station” of the Open Data Hub core, the following approach should be used:

* **origin**: to be set as “a22-web-site”
* **pointprojection**: the method “GetCoordinate” provides the coordinates for each point of the highway, with a resolution of 100 [m]. The following automatic logic should be implemented, so that in case of new locations computed for the forecasts everything can run smoothly without manual interventions:
	+ make a matching between the location name provided in this web-service and the toll gates names provided in the web-service “**GetCaselli**”. If the matching provides more suitable results (e.g. ‘Bolzano’ -> ‘Bolzano Sud’ and ‘Bolzano Nord’), let’s always consider one of them (e.g. the toll gate “south”).
	+ consider the value provided in the field “KM”
	+ use this value to retrieve the position from the web-service “**GetCoordinate**”, which provide the coordinates for each point of the highway, with a resolution of 100 [m]
* **stationcode**: as name
* **stationtype**: to be set as “TrafficForecast”

No additional metadata is foreseen in this case.

These stations should have just one (new) **type**, that we could simply call “**forecast**”. The measurements shall be stored as string in the table **measurementstring** (measurementstringhistory), by applying to the values the above mentioned mapping. The storage in these tables shall follow this logic:

* **created\_on**: the current timestamp in which the write operation in the DB is carried out
* **period**: the forecasts are not updated so frequently, they are typically checked on a daily basis. We can put this value to 86400 (seconds), i.e. one day. We can however think to interrogate the web-service e.g. once an hour and check for new values.
* **timestamp**: the timestamp of the forecast. Since we have forecasts for time intervals in the day, let’s consider the center of these intervals, i.e.:
	+ **0\_6** -> 3 AM
	+ **6\_12**-> 9 AM
	+ **12\_18** -> 3 PM
	+ **18** \_**24**-> 9 PM
* **string\_value**: the forecast data.

## **Traffic Events Data Collector**

## **Roadworks Events Data Collector**