

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

Bolzano, 26.05.2021

Preliminary market consultation and contextual request for quote

Dear supplier,

NOI SpA intends to initiate a preliminary market consultation pursuant to art. 20 of LP no. 16/2015 and art. 40 of Directive 2014/24/EC for the implementation of a software components (“**Data Collector**”), for the integration in the Open Data Hub of **real-time traffic data** collected by a network of traffic sensors managed by the company **Famas System** related to the **inter-urban road network of South Tyrol**.

More information and specifications about this market request are described in more detail in the Annex.

NOI SpA invites all interested economic operators to participate by filling in an expression of interest, **also in the form of a quote**, in relation to the products and requirements described in the Annex.

DEADLINE FOR THE DELIVERY OF THE QUOTE (10.06.2022):
The quote is to be sent exclusively by e-mail to r.cavaliere@noi.bz.it

Best regards
Roberto Cavaliere

ANNEX: Details of the preliminary market consultation

1. Introduction	2
2. Integration specification in the Open Data Hub	2
3. Modalities and guidelines for the development	3
4. Timing of implementation	7
5. Contents and evaluation of proposals / quotes	7
6. Invoicing procedures	8
7. Transfer of rights	8

1. Introduction

Famas System is the company selected by the Autonomous Province of Bolzano to manage the network of traffic monitoring stations placed on the inter-urban road network of South Tyrol. All the data is collected on a real-time basis on a back-end controlled by Famas System and made available through a custom API. The specification of this custom API is made available as a separate attachment. Since the access to the API is made secure by a control on the public IP address, also real examples of the data provided by the API are also made available.

2. Integration specification in the Open Data Hub

The model of the Open Data Hub (Mobility) is quite simple, and is mainly made up of three levels:

- **station:** refers to a concept of "station", i.e. a fixed/mobile installation that collects and transmits data (e.g. measuring station);
- **type:** refers to a concept of measured parameter, which can be associated with a particular sensor installed in a measuring station (e.g. temperature sensor in a meteorological station);
- **measurement:** is the measurement value which is associated univocally to a type and a station.

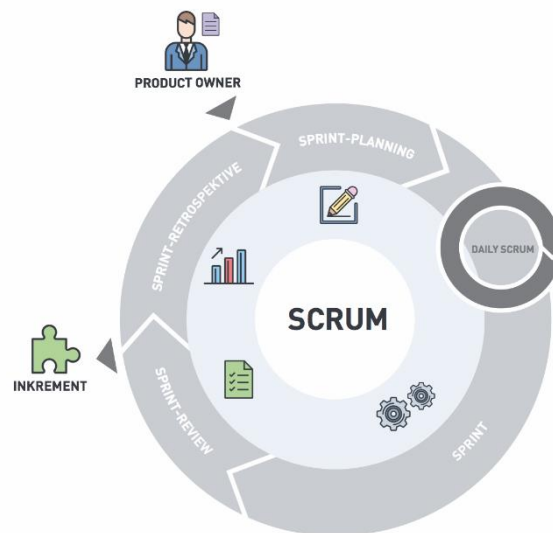
In this case the integration in the Open Data Hub is quite immediate. Each traffic station has to be considered as "station", or better each couple (lane, direction) of each traffic station has to be considered as a separate traffic station. All available measurement parameters have to be considered as "types", and all measurements as "measurements". The data flow has to match the structure of the traffic data (station type "Traffic Sensor", see <https://mobility.api.opendatahub.bz.it/v2/tree,node/TrafficSensor>)

Please note that the available API also provides information collected by additional Bluetooth sensors installed in correspondence to the traffic detection points. The Bluetooth sensors are able to detect and register the MAC address of Bluetooth devices put in visible mode, i.e. typically the devices present on the vehicles (e.g. navigation devices). The MAC addresses are first anonymized before they are stored. In this case, the measurements have to be associated to correspondent Bluetooth stations in the Open Data Hub model (see <https://mobility.api.opendatahub.bz.it/v2/tree,node/Bluetooth-Station>).

In order to speed up and simplify the requested implementation work it is possible to evaluate to reuse the source code developed for the implementation of other Open Data Hub Data Collectors, which is available in the following repository: <https://github.com/noi-techpark/bdp-commons> Additional technical details on how Data Collectors should be implemented can be found in the documentation available in this repository.

3. Modalities and guidelines for the development

The development of the activities covered by this market survey will follow the agile method (scrum). Two weeks sprint sessions are scheduled, unless otherwise agreed during the kick-off meeting with the core team of NOI S.p.A.



The software development will take place in three phases/environments:

- **development environment:** this environment is on supplier's infrastructure and is used during the development of the software components;
- **testing environment:** on infrastructure made available from NOI Techpark. This environment is used in order to test the new working versions of the software components. For the publication of the new versions a Continuous Integration (Jenkins) pipeline will be developed by the NOI team. For this reason, the new versions of the code will have to be "committed" to a dedicated Git Repository according to the instructions provided by the team of the NOI Techpark;
- **production environment:** on infrastructure made available from NOI Techpark. After the testing phase, as soon as the software produced is considered sufficiently stable, the software will be integrated in the production environment. Also, this process is managed automatically with Continuous Integration pipelines.

To coordinate the project NOI S.p.A. will use a Kanban Board in Github. Each functionality or issue will be described by NOI S.p.A. in Github and put on the Kanban Board. The Kanban Board will have the following columns:

- **Backlog:** contains all issues that are on hold and have to be discussed during the next sprint meeting with the supplier;
- **ToDo:** contains all issues that have to be concluded in the actual sprint;
- **In Progress:** contains all issues where the is working on;

- **To Review:** contains all issues where NOI Techpark has to make some reviews and that has to be reviewed during the sprint meeting.

All issues in the Kanban, but the one in Backlog, have to be assigned to the user that has to make the next step (e.g. the issues in ToDo will be assigned to the developer who has to develop the functionality, the issue in ToReview will be assigned to the tester, etc.). The supplier will have access to the project Kanban board and will have to check it regularly.

In order to allow the NOI S.p.A. team to properly review and test the code, for each issue in the ToDo lane the service provider has to send a pull request to the development Branch of the repository at least 5 working days before the sprint meeting.

In order to allow a better integration with the systems already in use by NOI Techpak it is required to implement all software components, where possible, using the technologies that are already in use by the Open Data Hub project. These technologies are described in the technical documentation, available at the following link:

<https://docs.opendatahub.bz.it/>

The source code has to be uploaded to the Git repositories provided by NOI Techpark. During the upload the service provider has to take particular attention to the following aspects:

- do not commit usernames or passwords. NOI Techpark uses Jenkins technology to build the code which implements password ingestion based on special keywords in the source code;
- well document the code describing at least:
 - the general architecture of the system;
 - the list of the licences of all the libraries used;
 - the installation process;
 - all other useful information for people who want to fork or install and use the project.

As Open Data Hub we created some boilerplate repositories for the most common project type (es. Java project, Web Component, .Net Core project, etc.). In case you are starting a new project from scratch, before starting your project please look for the boilerplate that best fits your project and use it to initialize your repository

While you are documenting your code please consider that the official language of the Open Data Hub is English. So, the entire documentation, including the comments in the code, has to be in English. Moreover, you have to observe the following guidelines:

- use the right boilerplate of the README.md if exists;
- use only markdown or text (no binaries, no PDF, etc.);
- should be so detailed that a third person, without any connection to the developers can setup the project, run it and develop it further;
- Java Doc and similar tools for other languages should be as complete as possible;
- add the author tags incl. emails;
- README.md should be a good description of the project and should also have a usage instruction (boilerplate does not consider that). Mainly because tools like ****npm**** use it as homepage for each project

In general, the documentation of the project (e.g. readme file, license file, etc.) should be done in order to allow third parties developers, who don't know anything about the project, to understand the whole project and also replicate, install or modify it without the need to get in contact with NOI S.p.A.

Therefore, the documentation (README.md) should include also:

- a short description that allows the user to understand the overall goal and functionalities of the project;
- Longer and detailed description that includes also:
 - description of the different parts of the repository/application;
 - description of different parts of the project (also other repositories, if existing, and a link to them) and how this application is part of the overall project;
 - external services/code/framework/software that are used including their licence and copyright information;
- detailed development setup instructions (including testing);
- detailed deployment setup instructions

In respect to the licensing and copyright information, the service provider has to follow the guidelines defined by the Reuse project:

<https://reuse.software/>

The service provider must provide code where the Reuse linter passes without errors and the licenses must be all compatible with each other.

As mentioned above the service provider, before each sprint meeting, will deliver the source code by making a Pull Request to the Development Branch of the repository Git provided by NOI S.p.A. at the beginning of the project. In general, the service provider has to observe the following guidelines to make the pull requests:

- at the beginning of each sprint the service provider will open a Pull Request (PR) with a prefix [WIP];
- during the sprint the service provider has to regularly push the commits to that PR in order to allow NOI S.p.A. to monitor the status of the project (additional information are available under <https://opendata-hub.readthedocs.io/en/latest/contributors.html>);
- at the end of the sprint (at least 5 days before the sprint meeting) the service provider will close and send the Pull Request.

NOI S.p.A. will analyze the Pull Request before the meeting and eventually send feedback to the service provider. The minimal requirements for a Pull Request to get accepted are:

- the documentation must exist and be as complete as possible in respect to the status of the project
- commits must not contain credentials or any other sensible data
- contributions (e.g. documentation, comments, etc.) must be in english
- merge conflicts must be resolved by the contributor
- all Continuous Integration verifications must pass
- Pull request branches should possibly have a linear history, that is, they should not contain merge commits

During the development cycles the pull request comments and in general the issues and the dedicated Kanban board on Github (original repository) must be tracked by the service provider. The discussion about issues, pull requests, and other specific comments on the code development will be managed on Git in the project repository and NOT through email. That also involves moving user stories to the corresponding column in the Kanban and assigning them to the right user.

These paragraphs contain some guidelines that the service provider should follow while implementing the project:

- commits should contain a single thing/feature, not be too big and specially they should not be a combination of unrelated features or bug-fixes;
- each commit must be described: present tense and active (e.g. "Add logging to commons" not "commons will get logging now" and not "Added logging").

For the deployment of the project NOI S.p.A. will use its CI/CD infrastructure, for this reason it is important that the service provider includes in the documentation of the project the information about how the application should be deployed or updated by a CD pipeline. Therefore, the documentation should point out the following things:

- What parameters must be configured? Which ones are secrets and which are not?
- What services must be used? (e.g.. PostgreSQL database, S3, ..)
- What steps must be made to package the application/project so that it can be copied to the server?
- What steps must be made on the server after deploying? (ex. database migrations executing with special command)
- What must be adjusted on the server only once? (ex. cron job, shared folder)

All projects should include unit tests and the minimal requirements for the service provider are:

- setup a test infrastructure;
- write unit tests to cover the most important features;
- the minimal test coverage should be 20%;
- tests should mainly cover own business logic (even if minimal) and not third party API's / libraries

Finally, a test driven development is appreciated.

In case that within the project it is foreseen also the development or the change of APIs, the service provider should observe the following guidelines:

- all API calls must be documented in the README.md;
- Swagger UI should be used;
- in case of errors the API should return to the consumer valid and descriptive error messages;
- the API should be RESTful, if possible, but, in case of need, other formats will be considered. In case of non RESTful APIs the service provider should present to NOI S.p.A. enough documentation to allow NOI S.p.A. to decide whether to go on with the new technology or stick to RESTful;
- the API must include also:
 - Response codes,
 - HTTP methods,
 - validity errors,
 - logging: JSON format for production and plain-text for local development written to stdout

In case that the project foresees Access Control List management, the service provider should observe the following guidelines:

- every login to a webapp needs ACL;
- the passwords must be complex enough to be secure;
- Oauth 2.0 standard is required
- Session management for webapps should be present, logout after an inactivity time (the length of the inactivity time depends on the single projects and has to be agreed with NOI S.p.A.)

As an Access Management tool NOI S.p.A. uses Keycloak (<https://www.keycloak.org/>) instance. More details are available at the following links:

<https://docs.opendatahub.bz.it/en/latest/guidelines/authentication.html>

<https://docs.opendatahub.bz.it/en/latest/guidelines/authentication.html#authentication-to-internal-infrastructure>

NOI S.p.A. is using Docker (<https://www.docker.com/>) to automate the deployment of the application and we strongly recommend to:

- use docker for local development;
- keep local docker setup, staging and production as similar as possible (these will be provided and updated by the NOI S.p.A. team).
- use environmental variables to configure different stages (i.e., .env files)

4. Timing of implementation

The Data Collector has to be implemented by the end of August.

5. Contents and evaluation of proposals / quotes

The quotes will be evaluated according to the criteria summarized in the table below.

Criteria	Points
1. Technical evaluation	70
1.1 Technical proposal for the implementation of the Data Collectors	20
1.2 Temporal planning and ability to complete the required developments in the shortest possible time	30
1.3 References and overall experience in the requested software developments	20
2. Economical quote	30

As far as **criterion 1.1** is concerned, it is expected to receive some technical indications about how Data Collectors are going to be implemented (e.g. with which programming language, how is the source code going to be structured, e.g.).

As far as **criterion 1.2** is concerned, it is expected to receive some concrete indications about how the activities are going to be internally organized in order to work requested in order to match the temporal deadline given. The temporal plan should take in consideration the bi-weekly time frame in which developments' sprints are going to be organized.

As far as **criterion 1.3** is concerned, it is expected to receive a CV of the personnel staff to be involved in this activity and up to 3 project references about similar development activities in which this staff (or alternatively the company) was involved. Such project references should have taken place from 1.1.2018 onwards.

As far as the **economical quote** is concerned, it is expected to receive a cost indication for the implementation of the whole Data Collector implementation work. The points are going to be assigned according to the following formulas:

$$C_i = \frac{O_{min}}{O_i}$$

$$PE_i = C_i * P_{max}$$

where:

- O_i is the economical quote of the i -th proposal.
- O_{min} is the economical quote of the best quote (i.e. with the lowest price)
- C_i is the coefficient associated to the i -th proposal
- P_{max} is the maximum number of points related to the economical quote (i.e. 30 points)
- PE_i is the number of points associated to the i -th proposal.

The technical evaluation and economical quote shall be provided in a document that should not exceed 10 pages of documentation. Any collaboration with other companies and the presence of any subcontracts must be explicitly indicated.

NOI S.p.A reserves the right to activate a cooperation with several economic operators if it considers it functional and efficient from a technical and economic point of view. Therefore, it is possible to present a quote also for just a subset of the datasets described in this document. In this case, the comparison of the quotes will be limited to the solely Data Collectors that are included in the proposal.

6. Invoicing procedures

The invoicing of the activities concluded by the supplier will be sent to NOI S.p.A via electronic invoice only after the outputs produced have been successfully tested by NOI S.p.A. Before to proceed with the testing of the outputs, the supplier must provide to NOI S.p.A.:

- the entire documentation;
- if code development is planned, the code must be uploaded to the Git repository provided by NOI S.p.A;
- in the case of multimedia contents (e.g. photos, videos, illustrations, documents), the service provider has to upload it on specific platforms (e.g. Vimeo, Flickr, etc.) and provide the source files or open versions through appropriate file hosting services indicated by NOI S.p.A.

All invoices must include that the transaction is subject to the Split Payment discipline as mentioned in the art.17-ter del DPR 633/197 and must be issued exclusively in electronic format (Unique Office code: T04ZHR3).

7. Transfer of rights

Where the creation of material subject to proprietary rights, including copyrights, sui generis data rights, and related rights, including solely of photographs, industrial design, all rights of economic exploitation arising from achieved results are reserved to NOI S.p.A., excepting those expressly excluded when the order is placed.

Further, if the material includes a software development project, all source code from libraries or other modules used in the realisation of an assignment and belonging to a third party must be released under an Open Source license (open-source.org/licenses) in a manner compatible with the scope of the "outbound" software license, without requirement for adaptation, addition, cancellation or requests for permission from third parties on the part of NOI S.p.A. In the absence of any expressly indicated license, the terms of the GPL v3 or AGPL v3 (depending on the project type) license shall apply. The use of material belonging to third parties must be expressly declared at the time of the quote, or be easily and immediately

understandable from the description of the project. In the event that code is developed during the realisation of this assignment, NOI S.p.A. will initiate a Git repository on which the supplier must develop and publish the source code.

If the material consists of data, creative works (drawings, literary works, cinematographic works, figurative art, photographs), industrial design or other material which are subject in whole or in part to the proprietary rights of a third party, the use of such material is permitted provided it is licensed under conditions compatible with the license under which said material will be published, if indicated. If no license is indicated, the material will be subject to conditions compatible with the Creative Commons Zero (CC0) license.

Interfaccia per NOI per lo scaricamento dei dati di traffico della Provincia Autonoma di Bolzano

Tramite le apposite chiamate http REST si potranno richiedere le informazioni di anagrafica, copertura, passaggi Bluetooth e aggregati veicolari. L'architettura REST utilizza il protocollo HTTP per "manipolare" lo stato della risorsa indicato nella riga di richiesta (URL). I metodi supportati sono:

- GET: questo metodo viene utilizzato per le operazioni di lettura semplici. La rappresentazione della risorsa richiesta è contenuta nel body della risposta inviata dal Server.
- POST: questo metodo viene utilizzato per le operazioni di lettura che richiedono la trasmissione di una struttura complessa di parametri nel body della richiesta. La rappresentazione della risorsa richiesta è sempre contenuta nel body della risposta inviata dal Server.

Nota: Nell'Header dei pacchetti HTTP inviati dal client al server è fondamentale che il campo "Content-Type" sia impostato al valore "application/json". La stessa cosa vale per i pacchetti di risposta inviati dal server, il campo "Content-Type" sarà sempre impostato al valore "application/json".

L'interfaccia implementata consiste di 5 chiamate:

1. **SchemiDiClassificazione:** Permette di scaricare i diversi schemi di classificazione in uso. Nel caso del sistema di raccolta traffico della PAB esiste, intanto, un solo schema di classificazione.
2. **AnagraficaPostazioni:** Permette di scaricare l'anagrafica delle postazioni di rilevamento traffico.
3. **DatiAggregatiSuPostazioni:** Permette di scaricare i dati di traffico aggregati per intervalli di 5 minuti
4. **DatiPassaggiSuPostazioni:** Permette di scaricare i singoli MAC Address Bluetooth rilevati (codificati MD5)
5. **PeriodiConAssenzaCopertura:** Permette di ottenere i periodi con assenza dati o sensori guasti

Il client che intende usufruire dei dati dovrebbe innanzitutto chiamare il servizio SchemiDiClassificazione e costruire il database degli schemi di classificazione per poi chiamare la AnagraficaPostazioni per costruire il database delle anagrafiche. Dopodiché può chiamare, a intervalli periodici, i servizi per lo scaricamento dei dati. A intervalli periodici dovrebbero anche essere ripetute le chiamate per aggiornare i schemi di classificazione e l'anagrafica delle postazioni, per gestire eventuali postazioni aggiunte o modificate. Per quanto riguarda le postazioni aggiunte, la chiamata all'anagrafica potrebbe anche essere triggerata se i servizi per la richiesta dei dati restituiscono dati per una postazione sconosciuta, perché significherebbe che è stata aggiunta di recente.

- I servizi sono raggiungibili all'URL:
`https://trafficoPAB.famassystem.it/idm/api/v1/NomeServizio`
- Un'interfaccia swagger è accessibile all'URL:
`https://trafficoPAB.famassystem.it/idm/swagger`

SchemiDiClassificazione:

Il servizio accetta una chiamata di tipo **GET** alla seguente url:

<https://trafficoab.famassystem.it/idm/api/v1/SchemiDiClassificazione>

La chiamata non accetta parametri in ingresso e restituisce a ogni chiamata l'elenco dei schemi di classificazione definite a sistema. La risposta segue il seguente modello json:

```
[
  {
    "Id": 0,
    "Nome": "string",
    "Classi": [
      {
        "Codice": 0,
        "Descrizione": "string"
      }
    ]
  }
]
```

Si tratta di un array di oggetti, ciascuno contenente i dati per uno schema.

Nome campo	Descrizione	
Id	Identificativo dello schema di classificazione	
Nome	Nome descrittivo dello schema	
Classi	Array di oggetti che contiene il codice e la descrizione per ciascuna classe dello schema:	
	Codice	Codice delle classe
	Descrizione	Descrizione della classe

AnagrafichePostazioni

Il servizio accetta una chiamata di tipo **GET** alla seguente url:

<https://trafficoab.famassystem.it/idm/api/v1/AnagrafichePostazioni>

La chiamata non accetta parametri in ingresso e restituisce a ogni chiamata l'elenco delle postazioni di rilevamento dati. La risposta segue il seguente modello json:

```
[
  {
    "Id": 0,
    "Nome": "string",
    "GeoInfo": {
      "Latitudine": 0,
      "Longitudine": 0,
      "Altitudine": 0,
      "Regione": "string",
      "Provincia": "string",
      "Comune": "string"
    },
    "StradaInfo": {
      "Nome": "string",
      "Chilometrica": 0
    },
    "Direzioni": [
```

```

    {
      "Tipo": "sconosciuta",
      "Descrizione": "string"
    }
  ],
  "SchemaDiClassificazione": 0,
  "NumeroCorsie": 0,
  "CorsieInfo": [
    {
      "Id": 0,
      "Descrizione": "string",
      "SensoDiMarcia": "sconosciuta"
    }
  ]
}
]

```

Si tratta di un array di oggetti, ciascuno contenente i dati per una postazione:

Nome campo	Descrizione												
Id	Identificativo numerico della postazione (attenzione: Non è lo stesso ID del numero di postazione utilizzato dalla provincia, anche se in alcuni casi corrisponde)												
Nome	Nome descrittivo della postazione (contiene il numero assegnato dalla provincia di Bolzano)												
GeoInfo	<table border="1"> <tr> <td>Latitudine</td> <td>Coordinata di installazione in formato WGS84 decimale</td> </tr> <tr> <td>Longitudine</td> <td>Coordinata di installazione in formato WGS84 decimale</td> </tr> <tr> <td>Altitudine</td> <td>Altitudine sul livello del mare in metri</td> </tr> <tr> <td>Regione</td> <td>Nome della regione di appartenenza</td> </tr> <tr> <td>Provincia</td> <td>Nome della provincia di appartenenza</td> </tr> <tr> <td>Comune</td> <td>Nome del comune di appartenenza</td> </tr> </table> <p>Struttura che contiene i dati geografici della postazione:</p>	Latitudine	Coordinata di installazione in formato WGS84 decimale	Longitudine	Coordinata di installazione in formato WGS84 decimale	Altitudine	Altitudine sul livello del mare in metri	Regione	Nome della regione di appartenenza	Provincia	Nome della provincia di appartenenza	Comune	Nome del comune di appartenenza
Latitudine	Coordinata di installazione in formato WGS84 decimale												
Longitudine	Coordinata di installazione in formato WGS84 decimale												
Altitudine	Altitudine sul livello del mare in metri												
Regione	Nome della regione di appartenenza												
Provincia	Nome della provincia di appartenenza												
Comune	Nome del comune di appartenenza												
StradaInfo	<p>Struttura che contiene le informazioni della strada su cui insorge la postazione:</p> <table border="1"> <tr> <td>Nome</td> <td>Nome della strada</td> </tr> <tr> <td>Chilometrica</td> <td>Progressiva di installazione</td> </tr> </table>	Nome	Nome della strada	Chilometrica	Progressiva di installazione								
Nome	Nome della strada												
Chilometrica	Progressiva di installazione												
Direzione	<p>Array di 2 strutture che contiene le info per ciascuna delle 2 direzioni di marcia:</p> <table border="1"> <tr> <td>Tipo</td> <td>Tipo direzione (ascendente oppure discendente)</td> </tr> <tr> <td>Descrizione</td> <td>Descrizione della rispettiva direzione di marcia</td> </tr> </table>	Tipo	Tipo direzione (ascendente oppure discendente)	Descrizione	Descrizione della rispettiva direzione di marcia								
Tipo	Tipo direzione (ascendente oppure discendente)												
Descrizione	Descrizione della rispettiva direzione di marcia												
SchemaDiClassificazione	Id dello schema di classificazione in uso per la postazione in questione. (Corrisponde all'ID restituito dalla chiamata a SchemiClassificazione)												

NumeroCorsie	Indica il numero di corsie presenti sul sito						
CorsieInfo	Array di strutture, una per ciascuna corsie, che contiene le informazioni per la rispettiva corsia: <table border="1" data-bbox="462 336 1428 526"> <tr> <td>Id</td> <td>Identificativo numerico della corsia (univoco all'interno della stessa postazione)</td> </tr> <tr> <td>Descrizione</td> <td>Descrizione della corsia</td> </tr> <tr> <td>SensoDiMarcia</td> <td>Normale senso di marcia della corsia (ascendente oppure discendente)</td> </tr> </table>	Id	Identificativo numerico della corsia (univoco all'interno della stessa postazione)	Descrizione	Descrizione della corsia	SensoDiMarcia	Normale senso di marcia della corsia (ascendente oppure discendente)
Id	Identificativo numerico della corsia (univoco all'interno della stessa postazione)						
Descrizione	Descrizione della corsia						
SensoDiMarcia	Normale senso di marcia della corsia (ascendente oppure discendente)						

DatiAggregatiSuPostazioni:

Il servizio accetta una chiamata di tipo **POST** alla seguente url:

<https://trafficoab.famassystem.it/idm/api/v1/DatiAggregatiSuPostazioni>

La chiamata accetta, nel body della richiesta, la seguente struttura di parametri:

```
{
  "IdPostazioni": [
    0
  ],
  "InizioPeriodo": "2021-11-12T14:31:54.546Z",
  "FinePeriodo": "2021-11-12T14:31:54.546Z"
}
```

Nome Campo	Descrizione
IdPostazioni	Array degli identificativi delle postazioni per cui richiedere i dati. Se l'array è vuoto vengono restituiti i dati per tutte le postazioni. L'identificativo delle postazioni deve corrispondere all'identificativo restituito dalla chiamata AnagraficaPostazioni.
InizioPeriodo	Data/Ora di inizio del periodo di richiesta dati in formato ISO8601.
FinePeriodo	Data/Ora di fine del periodo di richiesta dati in formato ISO8601.

Il periodo massimo richiedibile è limitato a 7 giorni. Qualsiasi richiesta che non rispetti tale limite otterrà come risposta un errore 400 con il seguente testo:

```
L'intervallo di dati richiesti è troppo grande! [(FinePeriodo - InizioPeriodo) > 7 giorni]
```

Il servizio restituisce i dati dei flussi di traffico nel periodo richiesto e per le postazioni richieste, aggregati per intervalli di 5 minuti. La risposta è un array, non ordinato, di oggetti, ciascuno contenente un intervallo da 5 minuti. I dati sono divisi per corsia e direzione, ovvero per ciascuna corsia di marcia saranno restituiti 2 record di dati, uno per ciascuna direzione di marcia. Intervalli in cui mancano i dati non sono trasmessi. Nel caso in cui non ci fossero dati per nessuna postazione e per tutto l'intervallo richiesto, la risposta conterrà un array vuoto.

La risposta è strutturata come da modello seguente:

```
[
  {
    "IdPostazione": 0,
    "Data": "2021-11-12T14:40:15.862Z",
    "Corsia": 0,
    "Direzione": "sconosciuta",
    "TotaleVeicoli": 0,
    "TotaliPerClasseVeicolare": {
      "IdClasse1": 0,
      "IdClasse2": 0,
      "IdClasse3": 0
    },
    "MediaArmonicaVelocita": 0,
    "HeadwayMedioSecondi": 0,
    "VarianzaHeadwayMedioSecondi": 0,
    "GapMedioSecondi": 0,
    "VarianzaGapMedioSecondi": 0
  }
]
```

Nome Campo	Descrizione
IdPostazione	Identificativo numerico della postazione a cui si riferisce l'intervallo di dati. Corrisponde all'identificativo restituito dalla chiamata alle anagrafiche.
Data	Data/Ora di inizio dell'intervallo di 5 minuti in formato ISO8601.
Corsia	Identificativo numerico della corsia sulla quale sono stati rilevati i flussi. Corrisponde all'identificativo restituito dalla chiamata alle anagrafiche.
Direzione	Direzione di marcia del flusso di traffico, (ascendente o discendente)
TotaleVeicoli	Il numero totale di passaggi veicolari per la corsia e la direzione in questione
TotaliPerClasseVeicolare	Flusso veicolare nell'intervallo e per la corsia e la direzione in questione diviso per singola classe veicolare. I nomi dei campi della struttura corrispondono agli identificatori numerici della rispettiva classe così come restituiti dal apposito servizio degli schemi di classificazione e i valori corrispondono ai passaggi nella rispettiva classe. Sono trasmessi solo le classi con almeno 1 passaggio, le classi con passaggi a zero non sono trasmessi. Nel caso in cui il totale veicoli è zero, la struttura sarà quindi vuota (senza campi)
MediaArmonicaVelocità	Velocità media armonica nell'intervallo in questione in km/h
HeadwayMedioSecondi	Headway (distanza tra 2 veicoli testa-testa) medio in secondi
VarianzaHeadwayMedioSecondi	Rispettiva varianza
GapMedioSecondi	Gap (distanza tra 2 veicoli coda-testa) medio in secondi
VarianzaGapMedioSecondi	Rispettiva varianza

PeriodiConAssenzaCopertura:

Il servizio accetta una chiamata di tipo **POST** alla seguente url:

<https://trafficoab.famassystem.it/idm/api/v1/PeriodiConAssenzaCopertura>

La chiamata accetta, nel body della richiesta, la seguente struttura di parametri:

```
{
  "IdPostazioni": [
    0
  ],
  "InizioPeriodo": "2021-11-12T14:31:54.546Z",
  "FinePeriodo": "2021-11-12T14:31:54.546Z"
}
```

Nome Campo	Descrizione
IdPostazioni	Array degli identificativi delle postazioni per cui richiedere i dati. Se l'array è vuoto vengono restituiti i dati per tutte le postazioni abilitate. L'identificativo delle postazioni deve corrispondere all'identificativo restituito dalla chiamata AnagraficaPostazioni.
InizioPeriodo	Data/Ora di inizio del periodo di richiesta dati in formato ISO8601.
FinePeriodo	Data/Ora di fine del periodo di richiesta dati in formato ISO8601.

Il periodo massimo richiedibile è limitato a 7 giorni. Qualsiasi richiesta che non rispetti tale limite otterrà come risposta un errore 400 con il seguente testo:

```
L'intervallo di dati richiesti è troppo grande! [(FinePeriodo - InizioPeriodo) > 7 giorni]
```

Il servizio restituisce i cosiddetti periodi anomali. I periodi anomali sono gli intervalli di 5 minuti per i quali i dati non sono (ancora) pervenuti oppure durante i quali era presente almeno un guasto dei sensori di rilevamento. La chiamata al servizio restituisce quindi tutti i periodi anomali verificatosi nell'intervallo richiesto e per le postazioni richieste. Per tutti gli intervalli **non** presenti nella risposta invece non ci sono segnalazioni di anomalie. Il servizio ricevente può utilizzare tale informazioni per eventualmente invalidare i periodi con sensori guasti e/o per riconoscere i periodi con mancanza di dati.

La risposta è un array di strutture, ciascuna relativa a una postazione come da modello seguente:

```
[
  {
    "IdPostazione": 0,
    "PeriodiAnomali": [
      {
        "Periodo": {
          "Da": "2021-11-12T14:49:04.597Z",
          "A": "2021-11-12T14:49:04.597Z"
        },
        "StatoSensoriOk": true,
        "CoperturaCompleta": true
      }
    ]
  }
]
```


Nome Campo	Descrizione						
IdPostazione	Identificativo numerico della postazione a cui si riferiscono i periodi . Corrisponde all'identificativo restituito dalla chiamata alle anagrafiche.						
Periodi anomali	Array di strutture, ciascuna contenente i dati di un periodo anomalo: <table border="1" data-bbox="561 423 1426 784"> <tbody> <tr> <td>Periodo</td> <td>Struttura composta da 2 campi ("Da" e "A") che indicano rispettivamente l'inizio e la fine del periodo in questione. I campi "Da" e "A" contengono una data/ora in formato ISO8601</td> </tr> <tr> <td>StatoSensoriOk</td> <td>Posto a false se nel periodo in questione sono stati rilevati guasti sui sensori. Altrimenti è posto a true.</td> </tr> <tr> <td>CoperturaCompleta</td> <td>Posto a true se il periodo in questione è coperto, ovvero sono pervenuti i dati dalla postazione periferica, false altrimenti</td> </tr> </tbody> </table> <p>Nota: Per definizione, per comparire tra i periodi anomali, uno dei 2 campi "StatoSensoriOk" e "CoperturaCompleta" è sempre posto a false.</p>	Periodo	Struttura composta da 2 campi ("Da" e "A") che indicano rispettivamente l'inizio e la fine del periodo in questione. I campi "Da" e "A" contengono una data/ora in formato ISO8601	StatoSensoriOk	Posto a false se nel periodo in questione sono stati rilevati guasti sui sensori. Altrimenti è posto a true.	CoperturaCompleta	Posto a true se il periodo in questione è coperto, ovvero sono pervenuti i dati dalla postazione periferica, false altrimenti
Periodo	Struttura composta da 2 campi ("Da" e "A") che indicano rispettivamente l'inizio e la fine del periodo in questione. I campi "Da" e "A" contengono una data/ora in formato ISO8601						
StatoSensoriOk	Posto a false se nel periodo in questione sono stati rilevati guasti sui sensori. Altrimenti è posto a true.						
CoperturaCompleta	Posto a true se il periodo in questione è coperto, ovvero sono pervenuti i dati dalla postazione periferica, false altrimenti						

DatiPassaggiSuPostazioni

Il servizio accetta una chiamata di tipo **POST** alla seguente url:

<https://trafficoFab.famassystem.it/idm/api/v1/DatiPassaggiSuPostazioni>

La chiamata accetta, nel body della richiesta, la seguente struttura di parametri:

```
{
  "IdPostazioni": [
    0
  ],
  "InizioPeriodo": "2021-11-12T14:31:54.546Z",
  "FinePeriodo": "2021-11-12T14:31:54.546Z"
}
```

Nome Campo	Descrizione
IdPostazioni	Array degli identificativi delle postazioni per cui richiedere i dati. Se l'array è vuoto vengono restituiti i dati per tutte le postazioni abilitate. L'identificativo delle postazioni deve corrispondere all'identificativo restituito dalla chiamata AnagraficaPostazioni.
InizioPeriodo	Data/Ora di inizio del periodo di richiesta dati in formato ISO8601.
FinePeriodo	Data/Ora di fine del periodo di richiesta dati in formato ISO8601.

Il periodo massimo richiedibile è limitato a massimo 12 ore. Qualsiasi richiesta che non rispetti tale limite otterrà come risposta un errore 400 con il seguente testo:

L'intervallo di dati richiesti è troppo grande! [(FinePeriodo - InizioPeriodo) > 12 ore]

Il servizio restituisce nella risposta l'elenco dei MAC Address Bluetooth rilevati per le postazioni richieste e nel periodo richiesto. La risposta è un array non ordinato di oggetti, ciascun relativo a un singolo passaggio Bluetooth ed è composta come indicato nel modello seguente:

```
[
  {
    "IdPostazione": 0,
    "Data": "2021-11-15T08:45:26.890Z",
    "IdVeicolo": "string"
  }
]
```

Nome Campo	Descrizione
IdPostazione	Identificativo della postazione sulla quale è stato rilevato il passaggio. L'identificativo corrisponde all'identificativo restituito dalla chiamata AnagraficaPostazioni.
Data	Data/Ora di rilevamento del passaggio in formato ISO8601.
IdVeicolo	MAC Address Bluetooth sotto forma di hash MD5 del valore originale

Nota: A prescindere dall'intervallo di dati richiesti, il servizio implementa un limite di al massimo 150000 veicoli che potranno essere restituiti per singola richiesta. Nel caso in cui la query restituirebbe più di 150.000 veicoli si otterrà una risposta con errore 400 e il seguente testo:

Troppi veicoli nell'intervallo richiesto! [> 150k veicoli]

AnagrafichePostazioni

```
[
  {
    "Id": 3,
    "Nome": "3",
    "GeoInfo": {
      "Latitudine": 46.4497009548582,
      "Longitudine": 11.3448734664564,
      "Regione": "Trentino-Alto Adige",
      "Provincia": "Bolzano",
      "Comune": "Laives"
    },
    "StradaInfo": {
      "Nome": "SS 12 dell'Abetone e del Brennero",
      "Chilometrica": 432.69
    },
    "Direzioni": [
      {
        "Tipo": "ascendente",
        "Descrizione": "Verso Bolzano"
      },
      {
        "Tipo": "discendente",
        "Descrizione": "Verso Trento"
      }
    ],
    "SchemaDiClassificazione": 1,
    "NumeroCorsie": 2,
    "CorsiInfo": [
      {
        "Id": 1,
        "Descrizione": "verso Bolzano",
        "SensoDiMarcia": "ascendente"
      },
      {
        "Id": 2,
        "Descrizione": "verso Trento",
        "SensoDiMarcia": "discendente"
      }
    ]
  },
  {
    "Id": 4,
    "Nome": "4",
    "GeoInfo": {
      "Latitudine": 46.4933369069905,
      "Longitudine": 11.4048027067281,
      "Regione": "Trentino-Alto Adige",
      "Provincia": "Bolzano",
      "Comune": "Cornedo all'Isarco"
    },
    "StradaInfo": {
      "Nome": "SS 12 dell'Abetone e del Brennero",
      "Chilometrica": 442.35
    },
    "Direzioni": [
      {
        "Tipo": "ascendente",
```

```
    "Descrizione": "Verso Brennero"  
  },  
  {  
    "Tipo": "discendente",  
    "Descrizione": "Verso Bolzano"  
  }  
],  
"SchemaDiClassificazione": 1,  
"NumeroCorsie": 2,  
"CorsiInfo": [  
  {  
    "Id": 1,  
    "Descrizione": "verso Brennero",  
    "SensoDiMarcia": "ascendente"  
  },  
  {  
    "Id": 2,  
    "Descrizione": "verso Bolzano",  
    "SensoDiMarcia": "discendente"  
  }  
]  
},  
]
```

SchemiDiClassificazione

```
[
  {
    "Id": 1,
    "Nome": "Schema Famas 9+1",
    "Classi": [
      {
        "Codice": 0,
        "Descrizione": "Conteggio"
      },
      {
        "Codice": 1,
        "Descrizione": "Moto"
      },
      {
        "Codice": 2,
        "Descrizione": "Auto"
      },
      {
        "Codice": 3,
        "Descrizione": "Auto con rimorchio"
      },
      {
        "Codice": 4,
        "Descrizione": "Furgoni"
      },
      {
        "Codice": 5,
        "Descrizione": "Camion <7,5m"
      },
      {
        "Codice": 6,
        "Descrizione": "Camion >7,5m"
      },
      {
        "Codice": 7,
        "Descrizione": "Autotreni"
      },
      {
        "Codice": 8,
        "Descrizione": "Autoarticolati"
      },
      {
        "Codice": 9,
        "Descrizione": "Autobus"
      },
      {
        "Codice": 10,
        "Descrizione": "Altri"
      }
    ]
  }
]
```

Response headers

content-encoding: br

PeriodiConAssenzaCopertura (IdPostazioni: [], InizioPeriodo: "2021-12-03T11:06:08.236Z",
"FinePeriodo": "2021-12-03T11:06:08.236Z")

```
[
  {
    "IdPostazione": 3,
    "PeriodiAnomali": [
      {
        "Periodo": {
          "Da": "2021-12-03T11:05:00Z",
          "A": "2021-12-03T11:10:00Z"
        },
        "StatoSensoriOk": false,
        "CoperturaCompleta": false
      }
    ]
  },
  {
    "IdPostazione": 4,
    "PeriodiAnomali": [
      {
        "Periodo": {
          "Da": "2021-12-03T11:05:00Z",
          "A": "2021-12-03T11:10:00Z"
        },
        "StatoSensoriOk": false,
        "CoperturaCompleta": false
      }
    ]
  }
]
```

DatiAggregatiSuPostazioni (IdPostazioni: [3], InizioPeriodo: 2021-12-02T11:11:08.679Z, FinePeriodo: 2021-12-03T11:11:08.679Z)

```
[
  {
    "IdPostazione": 3,
    "Data": "2021-12-02T11:10:00Z",
    "Corsia": 0,
    "Direzione": "ascendente",
    "TotaleVeicoli": 64,
    "TotaliPerClasseVeicolare": {
      "2": 59,
      "4": 5
    },
    "MediaArmonicaVelocita": 79.3,
    "HeadwayMedioSecondi": 4.68,
    "VarianzaHeadwayMedioSecondi": 26.01,
    "GapMedioSecondi": 4.42,
    "VarianzaGapMedioSecondi": 26.12
  },
  {
    "IdPostazione": 3,
    "Data": "2021-12-02T11:10:00Z",
    "Corsia": 0,
    "Direzione": "discendente",
    "TotaleVeicoli": 0
  },
  {
    "IdPostazione": 3,
    "Data": "2021-12-02T11:10:00Z",
    "Corsia": 1,
    "Direzione": "ascendente",
    "TotaleVeicoli": 0
  },
  {
    "IdPostazione": 3,
    "Data": "2021-12-02T11:10:00Z",
    "Corsia": 1,
    "Direzione": "discendente",
    "TotaleVeicoli": 94,
    "TotaliPerClasseVeicolare": {
      "1": 5,
      "2": 77,
      "4": 3,
      "5": 2,
      "6": 3,
      "8": 3,
      "9": 1
    },
    "MediaArmonicaVelocita": 67.9,
    "HeadwayMedioSecondi": 2.96,
    "VarianzaHeadwayMedioSecondi": 27.47,
    "GapMedioSecondi": 2.65,
    "VarianzaGapMedioSecondi": 27.75
  },
  {
    "IdPostazione": 3,
```

```
"Data": "2021-12-02T11:15:00Z",
"Corsia": 0,
"Direzione": "ascendente",
"TotaleVeicoli": 50,
"TotaliPerClasseVeicolare": {
  "2": 46,
  "3": 2,
  "4": 1,
  "8": 1
},
"MediaArmonicaVelocita": 78.1,
"HeadwayMedioSecondi": 5.97,
"VarianzaHeadwayMedioSecondi": 47.25,
"GapMedioSecondi": 5.69,
"VarianzaGapMedioSecondi": 47.69
},
{
  "IdPostazione": 3,
  "Data": "2021-12-02T11:15:00Z",
  "Corsia": 0,
  "Direzione": "discendente",
  "TotaleVeicoli": 0
},
{
  "IdPostazione": 3,
  "Data": "2021-12-02T11:15:00Z",
  "Corsia": 1,
  "Direzione": "ascendente",
  "TotaleVeicoli": 0
},
{
  "IdPostazione": 3,
  "Data": "2021-12-02T11:15:00Z",
  "Corsia": 1,
  "Direzione": "discendente",
  "TotaleVeicoli": 107,
  "TotaliPerClasseVeicolare": {
    "1": 5,
    "2": 87,
    "4": 9,
    "6": 3,
    "8": 2,
    "9": 1
  },
  "MediaArmonicaVelocita": 70.1,
  "HeadwayMedioSecondi": 3.07,
  "VarianzaHeadwayMedioSecondi": 24.89,
  "GapMedioSecondi": 2.76,
  "VarianzaGapMedioSecondi": 25.04
},
]
```


DatiPassaggiSuPostazioni (IdPostazioni: [3], InizioPeriodo: 2021-12-03T08:14:20.126Z, FinePeriodo: 2021-12-03T11:14:20.126Z)

```
[
  {
    "IdPostazione": 3,
    "Data": "2021-12-03T08:25:06Z",
    "IdVeicolo": "9532E31173B863BE28A5B76CF1BB91C5"
  },
  {
    "IdPostazione": 3,
    "Data": "2021-12-03T08:25:08Z",
    "IdVeicolo": "A032FA4CC79C8EB1342A2F4A53D2260E"
  },
  {
    "IdPostazione": 3,
    "Data": "2021-12-03T08:25:12Z",
    "IdVeicolo": "E51B97BB2C56050F1F91C74E5AAF738E"
  },
]
```