

Comune di Ferrara

webAPI FROST

DEDAGROUP Public Services s.r.l

Versione 1.0
14 Ottobre 2024

INDICE

1. Oggetto	3
1.1 webApi FROST	4
1.1.1 Things.....	5
1.1.2 Locations.....	6
1.1.3 Datastream	7
1.1.4 ObservedProperty.....	9
1.1.5 Sensor	9
1.1.6 Observation.....	10
1.1.7 FeatureOfInterest	11
1.1.8 Caso d'uso complesso	11

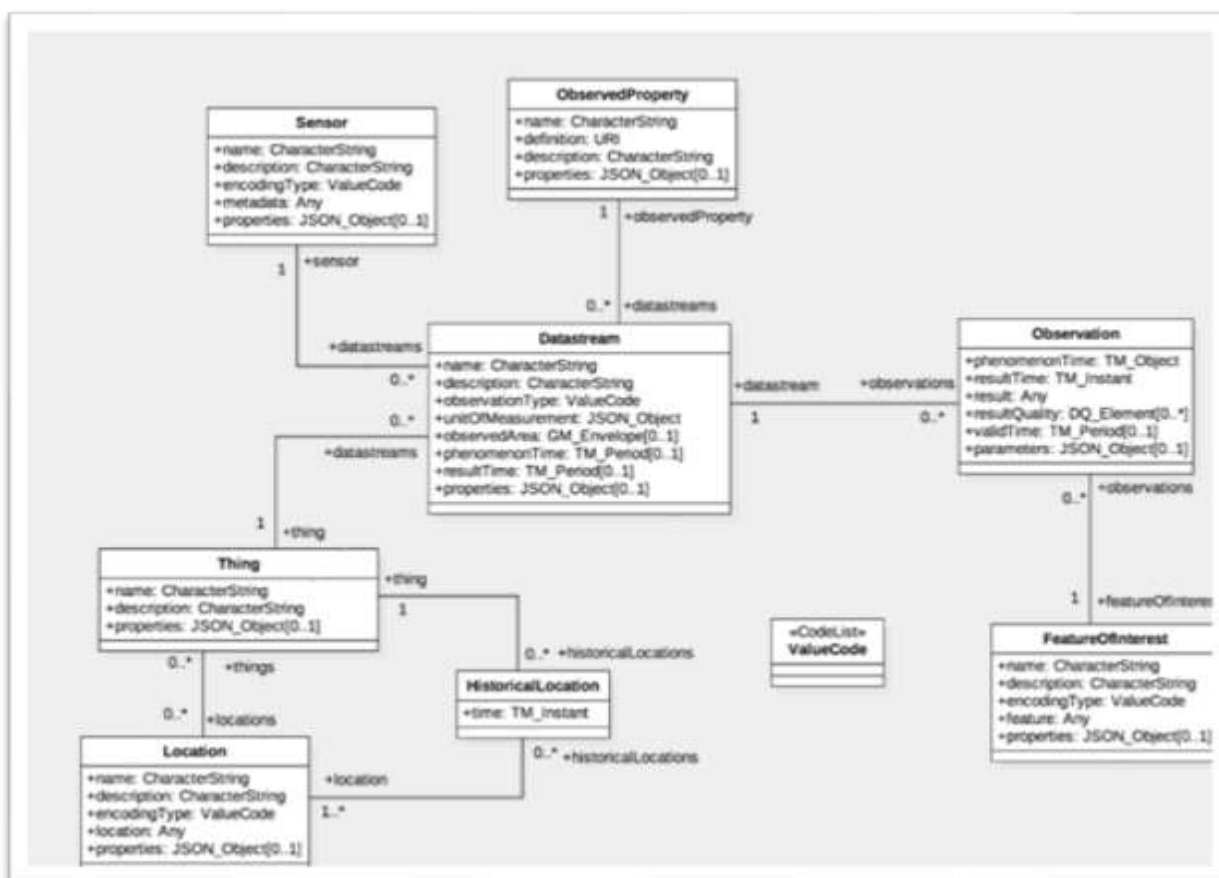
1. OGGETTO

Nel documento si riportano istruzioni e casi d'uso utili alla fruizione delle WebAPI esposte dal server FROST.

1.1 WEBAPI FROST

Le API Frost sono di tipo RESTful API e rispettano le specifiche dello standard OGC in merito ai dati forniti dagli *interconnect IoT devices* (SensorsThings STA). Tale si basa su uno standard OGC aperto che affronta l'interoperabilità sintattica e l'interoperabilità semantica dell' Internet of Things, in questo modo i diversi sistemi IoT si scambiano e utilizzano informazioni.

Il modello dati sottostante è raffigurato nell'immagine:



La radice delle API è <https://iot.comune.fe.it/FROST-Server/v1.1>

Doc di riferimento: <https://docs.ogc.org/is/15-078r6/15-078r6.html>

Lo standard OGC SensorThings API consente di utilizzare il sistema di query tipico degli OData, è dunque possibile raffinare le richieste ed avere i risultati desiderati mediante gli operatori:

- \$filter
- \$count
- \$orderby
- \$skip
- \$stop
- \$expand

- \$select

Tali operatori vanno utilizzati conoscendo il modello dati con cui si presentano le entità, possono infatti essere eseguite delle query interrogando gli attributi naturali di ogni entità (name, description) oppure utilizzando le properties che possono variare in funzione dell'entità e degli attributi con i quali sono state ampliate.

Di seguito si riportano degli esempi di utilizzo su casi pratici.

1.1.1 THINGS

Si tratta di un oggetto dotato di uno o più sensori atti alla misurazione di determinati fenomeni (ObservedProperties).

L'elenco delle things presenti in un database è consultabile mediante l'url:

<https://iot.comune.fe.it/FROST-Server/v1.1/Things>

La risposta del servizio è del tipo:

```
{
  "description": "tipo: traffic, zona: Urban",
  "@iot.id": "aq_prepair_emiliaromagna_7000015",
  "name": "Stazione 7000015",
  "properties": {
    "organization": "aq_prepair_emiliaromagna"
  },
  "@iot.selfLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Things('aq_prepair_emiliaromagna_7000015')",
  "HistoricalLocations@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Things('aq_prepair_emiliaromagna_7000015')/HistoricalLocations",
  "Datastreams@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Things('aq_prepair_emiliaromagna_7000015')/Datastreams",
  "Locations@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Things('aq_prepair_emiliaromagna_7000015')/Locations",
  "MultiDatastreams@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Things('aq_prepair_emiliaromagna_7000015')/MultiDatastreams",
  "TaskingCapabilities@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Things('aq_prepair_emiliaromagna_7000015')/TaskingCapabilities"
}
```

La risposta del servizio restituisce due tipi di informazioni, da una parte gli attributi specifici della entità *Things*, dall'altra i link con i quali la particolare things è collegata.

Analizzando gli attributi, si può vedere che ce ne sono alcuni sui quali è possibile utilizzare gli operatori per filtrare le informazioni; ad esempio è possibile avere l'elenco delle things e "selezionare" soltanto il nome:

[https://iot.comune.fe.it/FROST-Server/v1.1/Things?\\$select=name](https://iot.comune.fe.it/FROST-Server/v1.1/Things?$select=name)

Se si desiderano anche altri attributi, ad esempio la description:

[https://iot.comune.fe.it/FROST-Server/v1.1/Things?\\$select=name,description](https://iot.comune.fe.it/FROST-Server/v1.1/Things?$select=name,description)

Se si vuole conoscere anche il numero delle things oltre che il nome si aggiunge l'operatore count=true

[https://iot.comune.fe.it/FROST-Server/v1.1/Things?\\$select=name&\\$count=true](https://iot.comune.fe.it/FROST-Server/v1.1/Things?$select=name&$count=true)

L'insieme ottenuto si può ordinare:

[https://iot.comune.fe.it/FROST-Server/v1.1/Things?\\$select=name&\\$orderby=name%20asc](https://iot.comune.fe.it/FROST-Server/v1.1/Things?$select=name&$orderby=name%20asc)

Dalle query di insieme si può passare alle query di dettaglio, se ad esempio si desidera una thing/stazione all'interno della quale ci sono diversi sensori, ad esempio la thing che ha per id (@iot.id):

aq_prepair_emiliaromagna_7000015 si può eseguire:

[https://iot.comune.fe.it/FROST-Server/v1.1/Things\('aq_prepair_emiliaromagna_7000015'\)](https://iot.comune.fe.it/FROST-Server/v1.1/Things('aq_prepair_emiliaromagna_7000015'))

1.1.2 LOCATIONS

L'entità *location* individua la posizione in cui si trova la thing (specificando la posizione in long/lat).

L'Elenco delle locations sulle quali possono trovarsi delle things è dato dall'url:

<https://iot.comune.fe.it/FROST-Server/v1.1/Locations>

```
{
  "@iot.selflink": "https://iot.comune.modena.it/FROST-Server/v1.1/Locations/'as_prepar_emiliaromagna_7808015'",
  "@iot.id": "as_prepar_emiliaromagna_7808015",
  "name": "PORTA SAN FELICE",
  "description": "quota: 50m, Emilia Romagna (BO)",
  "encodingType": "application/vnd.geo+json",
  "location": {
    "type": "Point",
    "coordinates": [
      11.32752672,
      44.49985998
    ]
  },
  "properties": {
    "organization": "as_prepar_emiliaromagna"
  },
  "HistoricalLocations@iot.navigationlink": "https://iot.comune.modena.it/FROST-Server/v1.1/Locations/'as_prepar_emiliaromagna_7808015'/HistoricalLocations",
  "Things@iot.navigationlink": "https://iot.comune.modena.it/FROST-Server/v1.1/Locations/'as_prepar_emiliaromagna_7808015'/Things"
}
```

Come già fatto in precedenza è possibile richiedere l'elenco delle location selezionando solo il nome o altri attributi, un conteggio e un ordinamento:

[https://iot.comune.fe.it/FROST-Server/v1.1/Locations?\\$select=name](https://iot.comune.fe.it/FROST-Server/v1.1/Locations?$select=name)

[https://iot.comune.fe.it/FROST-Server/v1.1/Locations?\\$select=name,description,location/coordinates](https://iot.comune.fe.it/FROST-Server/v1.1/Locations?$select=name,description,location/coordinates)

[https://iot.comune.fe.it/FROST-Server/v1.1/Locations?\\$select=name&\\$count=true](https://iot.comune.fe.it/FROST-Server/v1.1/Locations?$select=name&$count=true)

[https://iot.comune.fe.it/FROST-Server/v1.1/Locations?\\$select=name&\\$orderby=name%20asc](https://iot.comune.fe.it/FROST-Server/v1.1/Locations?$select=name&$orderby=name%20asc)

Per consultare nel dettaglio una location di cui si conosce l'identificativo (@iot.id):

[https://iot.comune.fe.it/FROST-Server/v1.1/Locations\('traffico_comune_fe_63'\)](https://iot.comune.fe.it/FROST-Server/v1.1/Locations('traffico_comune_fe_63'))

```
{
  "description": "Comune di Ferrara",
  "encodingType": "application/vnd.geo+json",
  "@iot.id": "traffico_comune_fe_63",
  "location": {
    "type": "Point",
    "coordinates": [
      11.6136961071958,
      44.8411181227637
    ]
  },
  "name": "63",
  "properties": {
    "organization": "traffico_comune_fe"
  },
  "@iot.selflink": "https://iot.comune.fe.it/FROST-Server/v1.1/Locations('traffico_comune_fe_63')",
  "Things@iot.navigationlink": "https://iot.comune.fe.it/FROST-Server/v1.1/Locations('traffico_comune_fe_63')/Things",
  "HistoricalLocations@iot.navigationlink": "https://iot.comune.fe.it/FROST-Server/v1.1/Locations('traffico_comune_fe_63')/HistoricalLocations"
}
```

Per vedere la things (o le things) collegata a questa location è sufficiente aggiungere /Things:

[https://iot.comune.fe.it/FROST-Server/v1.1/Locations\('traffico_comune_fe_63'\)/Things](https://iot.comune.fe.it/FROST-Server/v1.1/Locations('traffico_comune_fe_63')/Things)

```
"value": {
  {
    "description": "Comune di Ferrara",
    "@iot.id": "traffico_comune_fe_63",
    "name": "63",
    "properties": {
      "organization": "traffico_comune_fe"
    },
    "@iot.selflink": "https://iot.comune.fe.it/FROST-Server/v1.1/Things('traffico_comune_fe_63')",
    "HistoricalLocations@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Things('traffico_comune_fe_63')/HistoricalLocations",
    "Datastreams@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Things('traffico_comune_fe_63')/Datastreams",
    "Locations@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Things('traffico_comune_fe_63')/Locations",
    "MultiDatastreams@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Things('traffico_comune_fe_63')/MultiDatastreams",
    "TaskingCapabilities@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Things('traffico_comune_fe_63')/TaskingCapabilities"
  }
}
```

Come si può vedere dalla risposta ottenuta, a partire da questa location è stato possibile arrivare alla Thing e da questa è poi possibile consultare i Datastreams (o multidatastreams) dei sensori collegati.

Le richieste eseguite, possono essere eseguite con dei filtri secondo la sintassi:

[https://iot.comune.fe.it/FROST-Server/v1.1/Locations?\\$select=name&filter ...](https://iot.comune.fe.it/FROST-Server/v1.1/Locations?$select=name&filter ...)

Per filtrare, dall'elenco delle Location, quella che ha uno specifico nome, ad esempio quella che ha nome "Ferrara - Via Cassoli":

[https://iot.comune.fe.it/FROST-Server/v1.1/Locations?\\$filter=name eq 'Ferrara - Via Cassoli'](https://iot.comune.fe.it/FROST-Server/v1.1/Locations?$filter=name eq 'Ferrara - Via Cassoli')

Se non si conosce il nome esatto, si può utilizzare l'operatore che valuta una parte della stringa utilizzando l'operatore *substringof*:

[https://iot.comune.fe.it/FROST-Server/v1.1/Locations?\\$filter=substringof\(%27PORTA%27,name\)](https://iot.comune.fe.it/FROST-Server/v1.1/Locations?$filter=substringof(%27PORTA%27,name))

oppure:

[https://iot.comune.fe.it/FROST-Server/v1.1/Locations?\\$filter=substringof\(%27Ferrara%27,name\)](https://iot.comune.fe.it/FROST-Server/v1.1/Locations?$filter=substringof(%27Ferrara%27,name))

Si possono applicare filtri anche sugli altri attributi di secondo livello, ad esempio se si vogliono conoscere le locations relative a sensori che appartengono alla organization "traffico_comune_fe" si può eseguire:

[https://iot.comune.fe.it/FROST-Server/v1.1/Locations?\\$filter=properties/organization%20eq%20%27traffico_comune_fe%27](https://iot.comune.fe.it/FROST-Server/v1.1/Locations?$filter=properties/organization%20eq%20%27traffico_comune_fe%27)

Altri esempi di ricerca:

[https://iot.comune.fe.it/FROST-Server/v1.1/Locations?\\$filter=substringof\(%27bici%27,properties/organization\)](https://iot.comune.fe.it/FROST-Server/v1.1/Locations?$filter=substringof(%27bici%27,properties/organization))

[https://iot.comune.fe.it/FROST-Server/v1.1/Locations?\\$filter=substringof\(%27traffico%27,properties/organization\)](https://iot.comune.fe.it/FROST-Server/v1.1/Locations?$filter=substringof(%27traffico%27,properties/organization))

Se si volessero conoscere le altre organization quindi l'elenco totale in tutto l'archivio si può eseguire:

[https://iot.comune.fe.it/FROST-Server/v1.1/Locations?\\$select=properties/organization](https://iot.comune.fe.it/FROST-Server/v1.1/Locations?$select=properties/organization)

Il concetto di **HistoricalLocations** non è stato gestito.

1.1.3 DATASTREAM

Il Datastream è un insieme di dati derivanti dalle osservazioni prodotte dallo stesso Sensore.

L'elenco dei datastream presenti nel DB è dato dall'url:

<https://iot.comune.fe.it/FROST-Server/v1.1/Datastreams>

La risposta si presenta nella forma:

```
{
  "description": "Polveri sottili",
  "@iot.id": "aq_prepar_lombardia_IT0689A_PM25",
  "name": "PM2.5",
  "observationType": "http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement",
  "properties": {
    "organization": "aq_prepar_lombardia"
  },
  "resultTime": null,
  "@iot.selflink": "https://iot.comune.fe.it/FROST-Server/v1.1/Datastreams('aq_prepar_lombardia_IT0689A_PM25')",
  "unitOfMeasurement": {
    "name": null,
    "symbol": "µg/m³",
    "definition": null
  },
  "Thing@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Datastreams('aq_prepar_lombardia_IT0689A_PM25')/Thing",
  "Sensor@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Datastreams('aq_prepar_lombardia_IT0689A_PM25')/Sensor",
  "Observations@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Datastreams('aq_prepar_lombardia_IT0689A_PM25')/Observations",
  "ObservedProperty@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Datastreams('aq_prepar_lombardia_IT0689A_PM25')/ObservedProperty"
}
```

Come già visto in precedenza, per ogni attributo è possibile effettuare delle richieste utilizzando gli operatori di selezione attributi o di filtro.

I Datastream o i MultiDatastream (quelli che forniscono più observedProperties) presi in sé non forniscono informazioni utili, occorre infatti collegarli ai sensori o alle things.

Si possono richiedere i Datastream di una certa stazione/things, ad esempio alla things

'aq_labservice_LS0621020069'

[https://iot.comune.fe.it/FROST-Server/v1.1/Things\('aq_labservice_LS0621020069'\)/Datastreams](https://iot.comune.fe.it/FROST-Server/v1.1/Things('aq_labservice_LS0621020069')/Datastreams)

Se si vogliono mantenere anche gli attributi dell'entità da cui si è partiti (in questo caso la things) è disponibile l'opzione *expand* grazie alla quale si può espandere il risultato combinando i dati delle diverse entità

[https://iot.comune.fe.it/FROST-Server/v1.1/Things\('aq_labservice_LS0621020069'\)?\\$expand=Datastreams](https://iot.comune.fe.it/FROST-Server/v1.1/Things('aq_labservice_LS0621020069')?$expand=Datastreams)


```
{
  "description": "tipo: Progetto AirBreak",
  "@iot.id": "aq_labservice_LS0621020069",
  "name": "Stazione LS0621020069",
  "properties": {
    "organization": "aq_labservice"
  },
  "@iot.selflink": "https://iot.comune.fe.it/FROST-Server/v1.1/Things('aq_labservice_LS0621020069')",

```

Thing

```
"Datastreams": {
  {
    "description": "Polveri sottili (media oraria)",
    "@iot.id": "aq_labservice_LS0621020069_PM25-h",
    "name": "PM2.5-h",
    "observationType": "http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement",
    "observedArea": {
      "type": "Point",
      "coordinates": [
        11.58301,
        44.84227
      ]
    },
    "phenomenonTime": "2021-09-10T13:00:00.000Z/2024-08-09T23:00:00.000Z",
    "properties": {
      "organization": "aq_labservice"
    },
    "resultTime": null,
    "@iot.selflink": "https://iot.comune.fe.it/FROST-Server/v1.1/Datastreams('aq_labservice_LS0621020069_PM25-h')",
    "unitOfMeasurement": {
      "name": null,
      "symbol": "µg/m³",
      "definition": null
    }
  }
}
}
```

Datastream

1.1.4 OBSERVEDPROPERTY

Si tratta della grandezza osservata, l'elenco delle grandezze osservate memorizzate nel DB è ricavabile dall'elenco: <https://iot.comune.fe.it/FROST-Server/v1.1/ObservedProperties>

```
{
  "description": "Biossido di azoto",
  "@iot.id": "aq_preair_emiliaromagna_NO2",
  "name": "NO2",
  "properties": {
    "organization": "aq_preair_emiliaromagna"
  },
  "@iot.selflink": "https://iot.comune.fe.it/FROST-Server/v1.1/ObservedProperties('aq_preair_emiliaromagna_NO2')",
  "Datastreams@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/ObservedProperties('aq_preair_emiliaromagna_NO2')/Datastreams",
  "MultiDatastreams@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/ObservedProperties('aq_preair_emiliaromagna_NO2')/MultiDatastreams"
}
```

Per avere l'elenco e la descrizione di tutte le grandezze osservate si può eseguire:
[https://iot.comune.fe.it/FROST-Server/v1.1/ObservedProperties?\\$Select=name,description](https://iot.comune.fe.it/FROST-Server/v1.1/ObservedProperties?$Select=name,description)

1.1.5 SENSOR

Strumento che osserva un fenomeno. L'elenco dei sensori che misurano delle grandezze è dato dal link:
<https://iot.comune.fe.it/FROST-Server/v1.1/Sensors>

```
"@iot.id": "aq_preparir_emiliaromagna_Sensore",
"name": "ARPAF Emilia-Romagna",
"properties": {
  "organization": "aq_preparir_emiliaromagna"
},
"@iot.selfLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Sensors('aq_preparir_emiliaromagna_Sensore')",
"Datastreams@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Sensors('aq_preparir_emiliaromagna_Sensore')/Datastreams",
"MultiDatastreams@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Sensors('aq_preparir_emiliaromagna_Sensore')/MultiDatastreams"
```

Per avere l'elenco dei sensori che registrano fenomeni si può eseguire:
[https://iot.comune.fe.it/FROST-Server/v1.1/Sensors?\\$select=name](https://iot.comune.fe.it/FROST-Server/v1.1/Sensors?$select=name)

1.1.6 OBSERVATION

Valore assunto dalla grandezza osservata, si tratta dei valori veri e propri.
L'elenco delle osservazioni è disponibile mediante l'url:
<https://iot.comune.fe.it/FROST-Server/v1.1/Observations>

```
"@iot.id": "63b9339a-ebb-11eb-95dc-c78ecce542b",
"phenomenonTime": "2021-07-25T23:00:00.000Z",
"parameters": {
  "validated": "0",
  "zone_type": "Urban",
  "station_type": "Background"
},
"result": "11",
"resultTime": null,
"@iot.selfLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Observations('63b9339a-ebb-11eb-95dc-c78ecce542b')",
"MultiDatastream@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Observations('63b9339a-ebb-11eb-95dc-c78ecce542b')/MultiDatastream",
"Datastream@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Observations('63b9339a-ebb-11eb-95dc-c78ecce542b')/Datastream",
"FeatureOfInterest@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/Observations('63b9339a-ebb-11eb-95dc-c78ecce542b')/FeatureOfInterest"
```

Come di può vedere dagli attributi di questa entità, in questo caso le ricerche possono essere effettuate per valore della grandezza (quindi *result*) oppure per data di registrazione del fenomeno (*phenomenonTime*).

Un esempio di richiesta di osservazioni di un certo Datastream è ad esempio la ricerca dei valori di un datastream con data inferiore o uguale ad un certo valore:
[http://iot.comune.fe.it/FROST-Server/v1.1/Observations?\\$orderby=phenomenonTime+desc&\\$filter=phenomenonTime%20le%202024-06-01T21%3A59%3A59.999Z](http://iot.comune.fe.it/FROST-Server/v1.1/Observations?$orderby=phenomenonTime+desc&$filter=phenomenonTime%20le%202024-06-01T21%3A59%3A59.999Z)

In alternativa si possono richiedere i dati di oggi (supponendo che oggi sia il 14/10/2024):
[https://iot.comune.fe.it/FROST-Server/v1.1/Observations?\\$orderby=phenomenonTime+desc&\\$filter=phenomenonTime%20gt%202024-10-14T00:00:00.000Z](https://iot.comune.fe.it/FROST-Server/v1.1/Observations?$orderby=phenomenonTime+desc&$filter=phenomenonTime%20gt%202024-10-14T00:00:00.000Z)

Altro esempio di richiesta è ad esempio per valore di temperatura maggiore di 35 gradi:
[https://iot.comune.fe.it/FROST-Server/v1.1/Datastreams\('bici contabici provincia 2ViaBologna NumeroPassaggi'\)/Observations?\\$orderby=result+desc&\\$filter=result%20ge%20%27950%27](https://iot.comune.fe.it/FROST-Server/v1.1/Datastreams('bici contabici provincia 2ViaBologna NumeroPassaggi')/Observations?$orderby=result+desc&$filter=result%20ge%20%27950%27)

Se si vuole conoscere il numero delle osservazioni totale presenti nel DB:

[https://iot.comune.modena.it/FROST-Server/v1.1/Observations?\\${count}=true](https://iot.comune.modena.it/FROST-Server/v1.1/Observations?${count}=true)

1.1.7 FEATUREOFINTEREST

Si tratta dell'ambiente in cui si trovano le things:

```
{
  "description": "quota: 50e, Emilia Romagna",
  "@iot.id": "aq_prepair_emiliaromagna_7000015",
  "name": "PORTA SAN FELICE",
  "properties": {
    "organization": "aq_prepair_emiliaromagna"
  },
  "@iot.selflink": "https://iot.comune.fe.it/FROST-Server/v1.1/featuresOfInterest('aq_prepair_emiliaromagna_7000015')",
  "Observations@iot.navigationLink": "https://iot.comune.fe.it/FROST-Server/v1.1/featuresOfInterest('aq_prepair_emiliaromagna_7000015')/Observations"
}
```

1.1.8 CASO D'USO COMPLESSO

Se si vuole mettere sotto osservazione una grandezza e capire quando supera un certo valore occorre fare una valutazione del tipo:

La thing XXX ha superato il valore YYY per l'obsProperty ZZZ nelle ultime 24h?

```
<URL base>/Things(<id XXX>)?
$expand=
  Datastreams/ObservedProperties($filter=name eq 'ZZZ'),
  Datastreams/Observations($filter=phenomenonTime ge '2024-09-30T08:00:00Z' and result ge YYY)
```

```
https://iot.comune.fe.it/FROST-Server/v1.1/Things('aq_labservice_LS0621020069')?
$expand=
  Datastreams($filter=name%20eq%20%27T%27),
  Datastreams/Observations($filter=phenomenonTime%20ge%202024-08-06T08:00:00Z and result ge
  '38';$orderby=phenomenonTime)
```

Che consiste in:

[https://iot.comune.fe.it/FROST-Server/v1.1/Things\('aq_labservice_LS0621020069'\)?\\$expand=Datastreams\(\\$filter=name%20eq%20%27T%27\),Datastreams/Observations\(\\$filter=phenomenonTime%20ge%202024-08-06T08:00:00Z%20and%20result%20ge%20%2738%27;\\$orderby=phenomenonTime\)](https://iot.comune.fe.it/FROST-Server/v1.1/Things('aq_labservice_LS0621020069')?$expand=Datastreams($filter=name%20eq%20%27T%27),Datastreams/Observations($filter=phenomenonTime%20ge%202024-08-06T08:00:00Z%20and%20result%20ge%20%2738%27;$orderby=phenomenonTime))