FIWARE APIs

FIWARE Platform for Smart Cities

FIWARE is born as the result of a public-private collaboration between the European Commission and the private sector. This open platform, which provides a set of tools for different functionalities, is an innovation ecosystem for the creation of new applications and Internet services. It is especially useful in terms of Smart Cities, as it ensures the interoperability and the creation of standard data models.

In this context, being "Smart" requires first being "Aware", that is, implementing a Smart City requires gathering and managing context information describing the current and historic state of the cities. This context information refers to the values of attributes charterizing relevant entities of city services, governance, and third party apps. In this regard, Smart applications and services for cities do need information about everything happening at every moment, and thus have access to this context information. FIWARE provides a mechanism to generate, collect, publish or query massive context information and use it for applications to react to their context. This is a complex process, as this information may come from different sources: systems, mobile apps' users, sensor networks, etc. It is our Context Broker, through a REST implementation of API OMA NGSI, which allows to shape and access it, whatever the source is.

The use and management from data coming from "Things" (i.e. sensors, actuators and other devices) is also a complex process, as there are many different protocols in the IoT sphere, but FIWARE provides a set of GEs allowing to access the relevant information through only one API (NGSI). It not only allows to read this sensor information, but also to act on some elements. Therefore, Context Broker is an essential part of the architecture to collect data, analyse them on real time, consult archives and their analysis, as well as to publish them as open data from a city. On the other hand, other functionalities such as business intelligence, web interfaces and advanced interfaces allow the creation of very powerful applications and solutions.

FIWARE IoT Stack

The FIWARE IoT Stack Documentation describes how to connect devices and receive data, integrating all device protocols and connectivity methods, understanding and interpreting relevant information. It isolates data processing and application service layers from the device and network complexity, in terms of access, security and network protocols.

FIWARE « ORION » Context Broker

Orion Context Broker allows you to manage all the whole lifecycle of NGSI context information including updates, queries, registrations and subscriptions. Using the Orion Context Broker, you are able to register context elements and manage them through updates and queries. In addition, you can subscribe to context information so when some condition occurs (e.g. an interval of time has passed or the context elements have changed) you receive a notification.

The Orion Context Broker Documentation describes in detail how to interact with the Context Broker, and thus, how to use the NGSI API for managing context information. On the other hand, this documentation describes how to deploy a dedicated instance of the Orion Context broker.

You can find the source code of Orion in the FIWARE Context Broker Github repository

FIWARE Device Backend Gateway (IDAS)

The FIWARE Device Backend Gateway supports several IoT protocols with a modular architecture where modules are called "IoT Agents". This compontents allow to simplify the management and integration of devices. It collects data from devices using heterogeneous protocols and translates them into standard platform language: NGSI entities.

The Device Backend Gateway Documentation describes what are the different IoT Agents available and how to develop new agents for non-supported protocols

FIWARE Application Mashup (WireCloud)

WireCloud builds on cutting-edge end-user (software) development, RIA and semantic technologies to offer a next-generation end-user centred web application mashup platform aimed at allowing end users without programming skills to easily create web applications and dashboards/cockpits (e.g. to visualize their data of interest or to control their domotized home or environment). Web application mashups integrate heterogeneous data, application logic, and UI components (widgets) sourced from the Web to create new coherent and value-adding composite applications.

The WireCloud Documentation describes in detail how to interact with the Wirecloud platform, how to develop WireCloud components, and hoy to deploy a dedicated instance.

You can find a running instance in the Mashup portal of the FIWARE Lab

You can find the source code in the WireCloud Github repository

Available Smart City Real Time Data Endpoints

FIWARE offers a huge amount of Open Data that is available for you for free at https://data.lab.fiware.org. This section covers the most important real-time information that you will be able to find in all the cities that participates in the FIWARE initiative.

Santander (Spain)

The city of Santander in Spain, relies on FIWARE for publishing its smart city real time data. For this purpose, the city has different NGSI endpoints.

There is one NGSI v1 service is deployed on http://mu.tlmat.unican.es:8099/ and can be queried sending NGSI queries to the *queyContext* service using HTTP POST requests. This service responses with a *contextResponses* node containing a list of *contextElement,* each represe nting a entitity or sensor. For these entities, it is included the type, the id, and a list of attributes, which is different depending on the type of the entities.

In this endpoint it is published the following information:

- **Indoor parkings**: Real time information of existing indoor parkings and the number of available spots. In the following screenshot you can find an example of the information returned for an indoor parking:


```
{
  "contextResponses": [
    {
      "contextElement": {
        "type": "ParkingLot",
        "isPattern": "false",
        "id": "urn:x-iot:smartsantander:parking:indoor:APAR1",
        "attributes": [
          {
            "name": "created",
            "type": "ISO8601",
            "value": "2016-02-09T08:46:48.000Z"
          },
          {
            "name": "extraSpotNumber",
            "type": "int",
            "value": "25"
          },
          {
            "name": "location",
            "type": "coords",
            "value": "43.4628890000,-3.7949180000",
            "metadatas": [
              {
                "name": "location",
                "type": "string",
                "value": "WGS84"
              }
            ]
          },
          {
            "name": "name",
            "type": "string",
            "value": "Parking de Puertochico"
          },
          {
            "name": "parkingDisposition",
            "type": "string",
            "value": "parallel"
          },
          {
            "name": "totalSpotNumber",
            "type": "int",
            "value": "475"
          }
        ]
      },
```


  - As can be seen in the screeshot, *ParkingLot* entities contain the following attributes:
    - **created**: The creation date in ISO8601 format
    - **extraSpotNumber**: An integer with the number of available spots for parking
    - **location**: Coordinates of the parking in WGS84 format
    - **name**: Name of the parking
    - **parkingDisposition**: Disposition of the cars in the parking, e.g parallel
    - **totalSpotNumber**: An integer with the total number of parking spots

- **Ambient data**: Real time information of ambient data, retrieved from sensors deployed in city buses. The following screenshot shows the information available for these sensors:

```
{
  "contextResponses": [
    {
      "contextElement": {
        "type": "AmbientObserved",
        "isPattern": "false",
        "id": "urn:x-iot:smartsantander:environmental:mobile:3050",
        "attributes": [
          {
            "name": "created",
            "type": "ISO8601",
            "value": "2013-12-02T11:11:02.000Z"
          },
          {
            "name": "direction:heading",
            "type": "string",
            "value": "0"
          },
          {
            "name": "location",
            "type": "coords",
            "value": "43.455600,-3.834100",
            "metadatas": [
              {
                "name": "location",
                "type": "string",
                "value": "WGS84"
              }
            ]
          },
          {
            "name": "mileage:total",
            "type": "float",
            "value": "17242"
          },
          {
            "name": "pollutants",
            "type": "object",
            "value": {
              "CO": {
                "concentration": "0.1",
                "descripton": "Carbon Monoxide"
              },
              "NO2": {
                "concentration": "50.7",
                "descripton": "Nitrogen Dioxide"
              },
              "O3": {
                "concentration": "2",
                "descripton": "Ozone"
              }
            }
          },
          {
            "name": "position:altitude",
            "type": "float",
            "value": "14.0"
          },
          {
            "name": "sensorType",
            "type": "string",
            "value": "Mobile"
          },
          {
            "name": "speed:instantaneous",
            "type": "float",
            "value": "0"
          }
        ]
      },
```


- As can be seen in the screenshot, *AmbientObserved* entities contain the following attributes:
  - **created**: The creation date in ISO8601 format
  - **direction:heading**: Direction of the bus (in degrees) where the sensor is deployed
  - **location**: Coordinates of the sensor in WGS84 format
  - **mileage:total**: Total distance covered by the bus where the sensor is deployed
  - **pollutants**: Current values of CO, NO2 and O3
  - **position:altitude:** Current altitude (meters) of the bus where the sensor is deployed
  - **sensorType**: Type of the sensor
  - **speed:instantaneous**: Current speed (km/h) of the bus where the sensor is deployed

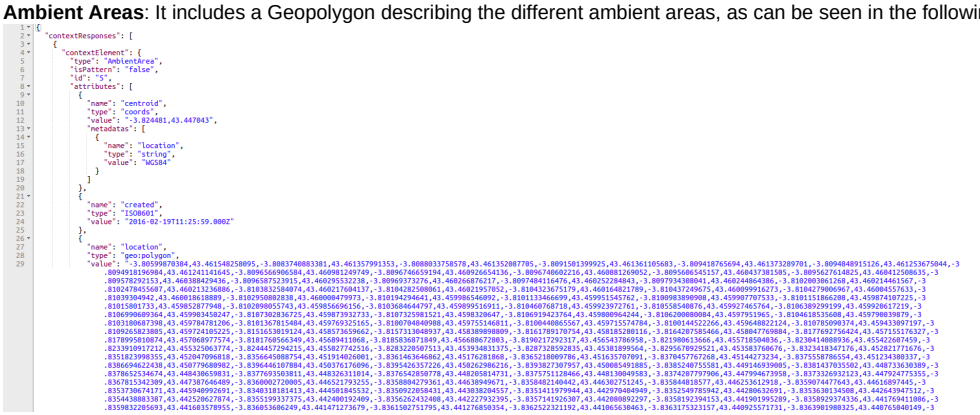
- Additionally, some of the sensors include two extra attributes:

```
{
  "name": "relativeHumidity",
  "type": "float",
  "value": "89"
},
```

```
{
  "name": "temperature",
  "type": "float",
  "value": "11.7"
}
```


- **relativeHumidity**: % of relative humidity
- **temperature**: Temperature in ºC

- **Ambient Areas**: It includes a Geopolygon describing the different ambient areas, as can be seen in the following screenshot:



```
{
 "contextResponses": [
  {
   "contextElement": {
    "type": "AmbientArea",
    "isPattern": "false",
    "id": "5",
    "attributes": [
     {
      "name": "centroid",
      "type": "coords",
      "value": "-3.824481,43.447043",
      "metadatas": [
       {
        "name": "location",
        "type": "string",
        "value": "WGS84"
       }
      ]
     },
     {
      "name": "created",
      "type": "ISO8601",
      "value": "2016-02-19T11:25:59.000Z"
     },
     {
      "name": "location",
      "type": "geo:polygon",
      "value": "-3.80599070384,43.461548258095,-3.8003740083381,43.461357991353,-3.8008033758578,43.461352087705,-3.8091501399925,43.461361105683,-3.8094187656694,43.461373289701,-3.8094840915126,43.461253675044,-3
.8094918196984,43.461241141645,-3.8096566906584,43.460981249749,-3.8096746659194,43.460926654136,-3.8096746602216,43.460881269652,-3.8095606543157,43.460437381585,-3.8095627614825,43.460412508635,-3
.809578292153,43.460388429436,-3.8096587523915,43.460295532238,-3.80969373276,43.460266876217,-3.8097484116476,43.460252284843,-3.8097934300041,43.460244864386,-3.8102003061268,43.460214461567,-3
.8102478455607,43.460213236886,-3.8103832584074,43.460217604137,-3.8104282580861,43.460219578052,-3.8104323675179,43.460164821789,-3.8104372496754,43.460099916273,-3.8104279006967,43.46004557633,-3
.8103930494?,43.460018618889,-3.8102950802838,43.460000479973,-3.8101942946411,43.459986546092,-3.8101133466699,43.459951545762,-3.8100983890908,43.459907707533,-3.8101151866288,43.459874107225,-3
.810158017333,43.459852877948,-3.8102098055743,43.45985669615 6,-3.8103684644797,43.459899516911,-3.8104607687 18,43.459923972761,-3.8105585400 76,43.459927465764,-3.8106308299199,43.459920617219,-3
.8106990609364,43.459903450247,-3.8107302836725,43.459073932733,-3.8107325981523,43.4590328647,-3.8106919423764,43.459080964244,-3.8106200080084,43.45979319 65,-3.8104618535608,43.459790039879,-3
.8103180687398,43.459784781206,-3.8101367815404,43.459769325165,-3.8100704840988,43.459755146811,-3.8100440865567,43.459715574784,-3.8100144522266,43.459648822124,-3.8107850983 74,43.459433097197,-3
.8109265823885,43.459724105225,-3.8151653019124,43.458573659662,-3.8157313048937,43.458389898809,-3.8161789170754,43.458185280116,-3.8164207585466,43.458047769884,-3.8177692756424,43.457155176327,-3
.8170959810874,43.457068977574,-3.8181760566349,43.456894110868,-3.8185836871849,43.456608672803,-3.8190217292317,43.456543786958,-3.8219806136 66,43.455718504036,-3.8230414088936,43.455422687459,-3
.8233910917212,43.455325063774,-3.8244457294215,43.455027742516,-3.8283220507513,43.455934831375,-3.8287328592835,43.455383899564,-3.8295670929521,43.455583760676,-3.8323418347176,43.452821771676,-3
.8351823998355,43.452047096818,-3.8356645088754,43.451914026001,-3.8361463646862,43.451762818 68,-3.8365218009786,43.451635707091,-3.8370457767268,43.451442732 34,-3.8375558786554,43.451234380337,-3
.8386694622438,43.450779680982,-3.8396446107884,43.450376176096,-3.8395426357226,43.450262986216,-3.8393827307957,43.450085491885,-3.8385240755581,43.449146939085,-3.8381437035502,43.448733630389,-3
.8378652534674,43.448430659833,-3.8377693503811,43.448326311014,-3.8376542859778,43.448205814731,-3.8375751128466,43.448130049583,-3.8374207797906,43.447994679958,-3.8373326932323,43.447924775355,-3
.8367815342309,43.447387646489,-3.8360002720095,43.446521793255,-3.8358084279361,43.446389949671,-3.8358402140442,43.446302751245,-3.8358440818577,43.446253612918,-3.8359074477643,43.446168974 45,-3
.8353730674171,43.445940992691,-3.8340318181413,43.444581845532,-3.8350922058431,43.443038204557,-3.8351411979944,43.442970404949,-3.8352549785942,43.442806326 91,-3.8353630134508,43.442643947512,-3
.8354430883387,43.442520627874,-3.8355199337375,43.442408192409,-3.8356262432408,43.442227932395,-3.8357141926307,43.442080892297,-3.8358192394153,43.441901995289,-3.8359929374336,43.441769411086,-3
.8359832285693,43.441683579955,-3.8360853606249,43.441471273679,-3.8361502751795,43.441276850354,-3.8362522321192,43.441065630463,-3.8363175323157,43.440925571731,-3.8363981988325,43.440765840149,-3
```


  - As can be seen in the screenshot, *AmbientArea* entities contain the following attributes:
    - **created**: The creation date in ISO8601 format
    - **centroid**: Coordinates of the geo poligon centroid in WGS84 format
    - **location**: Coordinates of the different points that build up the geo poligon in in WGS84

There is also another NGSI v1 deployed on http://130.206.85.12:1026 that offers information about the different buses on service in Santander:

- **Buses**: Real time information about the different buses on service in the city of Santander. The following screenshot shows the data returned from these sensors:

```
{
 "contextResponses": [
  {
   "contextElement": {
    "type": "device",
    "isPattern": "false",
    "id": "bus.182",
    "attributes": [
     {
      "name": "position",
      "type": "coords",
      "value": "43.464067921,-3.83063232272",
      "metadatas": [
       {
        "name": "location",
        "type": "string",
        "value": "WGS84"
       }
      ]
     },
     {
      "name": "line",
      "type": "string",
      "value": "18"
     },
     {
      "name": "id",
      "type": "string",
      "value": "http://datos.santander.es/api/datos/control_flotas_posiciones/323"
     },
     {
      "name": "TimeInstant",
      "type": "ISO8601",
      "value": "2016-05-03T13:52:50.967Z"
     },
     {
      "name": "index",
      "type": "string",
      "value": "0"
     },
     {
      "name": "service",
      "type": "string",
      "value": "1002"
     },
     {
      "name": "status",
      "type": "string",
      "value": "S"
     },
     {
      "name": "vehicle",
      "type": "string",
      "value": "3446"
     },
     {
      "name": "TimeInstantModified",
      "type": "ISO8601",
      "value": "2016-05-03T13:52:53.905Z"
     },
     {
      "name": "speed",
      "type": "string",
      "value": "2.66400003433227 54"
     }
    ]
   },
```


  - As can be seen, the following attributes are returned:
    - **position**: The current position of the bus
    - **line**: The bus line covered by the given bus

- **TimeInstant**: Relates to the time when the information was extracted
- **vehicle**: The number that identifies the bus
- **speed**: The current speed of the bus

Additionally, the NGSI v1 Global Instance deployed on the FIWARE Lab (http://orion.lab.fiware.org:1026) offers some real time information of Santander. Please, take note that to access these data, you have to have a FIWARE Lab account. You can create a new one at: https://account.lab.fiware.org/sign_up/ . The information provided by this endpoint covers:

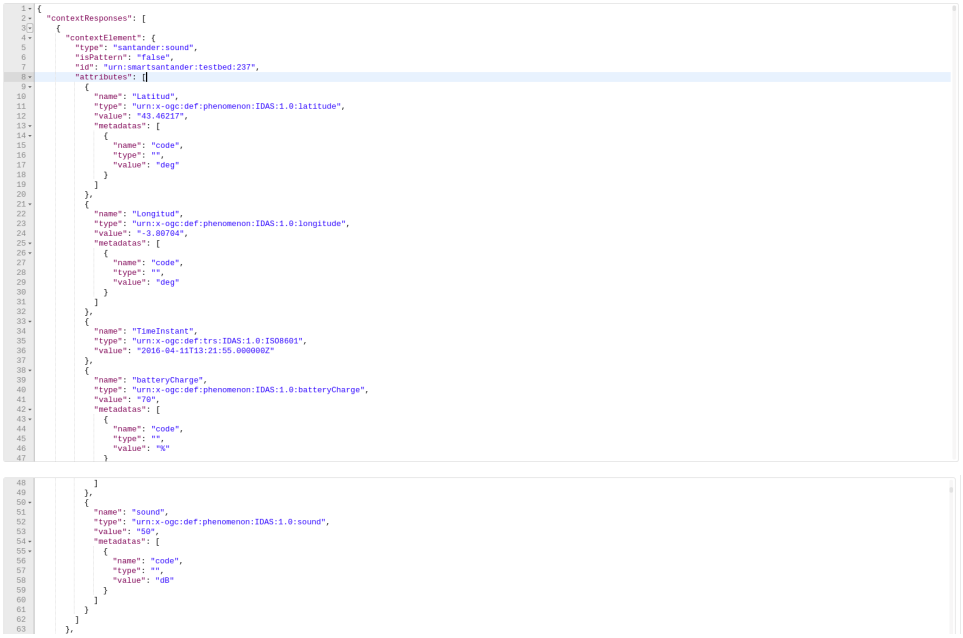
- **Mobile Sensors**: Real time information of ambient data, retrieved from sensors deployed in mobile vehicles (public transport, taxis, police cars,...) in Santander. The following screenshot shows the data returned from these sensors:


```
{
  "contextResponses": [
    {
      "contextElement": {
        "type": "santander:device",
        "isPattern": "false",
        "id": "urn:x-iot:smartsantander:u7jcfa:mobile:fa3028",
        "attributes": [
          {
            "name": "COConcentration",
            "type": "urn:x-ogc:def:phenomenon:IDAS:1.0:COConcentration",
            "value": "0.1",
            "metadatas": [
              {
                "name": "code",
                "type": "",
                "value": "milliGM3"
              }
            ]
          },
          {
            "name": "Latitud",
            "type": "urn:x-ogc:def:phenomenon:IDAS:1.0:latitude",
            "value": "43.4573",
            "metadatas": [
              {
                "name": "code",
                "type": "",
                "value": "deg"
              }
            ]
          },
          {
            "name": "Longitud",
            "type": "urn:x-ogc:def:phenomenon:IDAS:1.0:longitude",
            "value": "-3.83049",
            "metadatas": [
              {
                "name": "code",
                "type": "",
                "value": "deg"
              }
            ]
          },
          {
            "name": "NO2Concentration",
            "type": "urn:x-ogc:def:phenomenon:IDAS:1.0:NO2Concentration",
            "value": "116",
            "metadatas": [
              {
                "name": "code",
                "type": "",
                "value": "microGM3"
              }
            ]
          },
          {
            "name": "O3Concentration",
            "type": "urn:x-ogc:def:phenomenon:IDAS:1.0:O3Concentration",
            "value": "85",
            "metadatas": [
              {
                "name": "code",
                "type": "",
                "value": "microGM3"
              }
            ]
          },
          {
            "name": "TimeInstant",
            "type": "urn:x-ogc:def:trs:IDAS:1.0:ISO8601",
            "value": "2016-04-11T13:18:16.000000Z"
          },
          {
            "name": "airParticles",
            "type": "urn:x-ogc:def:phenomenon:SmartS:1.0:airParticles",
            "value": "9.99",
            "metadatas": [
              {
                "name": "code",
                "type": "",
                "value": "microGM3"
              }
            ]
          },
          {
            "name": "altitude",
            "type": "urn:x-ogc:def:phenomenon:IDAS:1.0:altitude",
            "value": "23",
            "metadatas": [
              {
                "name": "code",
                "type": "",
                "value": "m"
              }
            ]
          },
          {
            "name": "course",
            "type": "urn:x-ogc:def:phenomenon:SmartS:1.0:course",
            "value": "0"
          },
          {
            "name": "odometer",
            "type": "urn:x-ogc:def:phenomenon:SmartS:1.0:odometer",
            "value": "116290",
            "metadatas": [
              {
                "name": "code",
                "type": "",
                "value": "km"
              }
            ]
          },
          {
            "name": "relativeHumidity",
            "type": "urn:x-ogc:def:phenomenon:IDAS:1.0:relativeHumidity",
            "value": "24",
            "metadatas": [
              {
                "name": "code",
                "type": "",
                "value": "%"
              }
            ]
          },
          {
            "name": "temperature",
            "type": "urn:x-ogc:def:phenomenon:IDAS:1.0:temperature",
            "value": "20.9",
            "metadatas": [
              {
                "name": "code",
                "type": "",
                "value": "Cel"
              }
            ]
          }
        ]
      },
```


  - The attributes returned for each entity are:
    - **COConcentration**: Concentration of CO registered by the sensor
    - **Latitud**: The latitude where the measure were taken
    - **Longitud**: The longitude where the measure were taken
    - **NO2Concentration**: Concentration of NO_2 registered by the sensor

- **O3Concentration**: Concentration of O_3 registered by the sensor
- **TimeInstant**: Relates to the time when the information was extracted
- **airParticles**: The number of particles in the air registered by the sensor
- **altitude**: The altitude where the measure were taken
- **odometer**: The number of kms that the sensor has covered until now
- **relativeHumidity**: % of relative humidity
- **temperature**: Temperature in ºC

- **Sound Sensors**: Sound monitoring entities that show information about the sound level, time instant, battery charge and geolocation. The sound level measured is provided by the sound field. The following screenshot shows the data returned from these sensors:

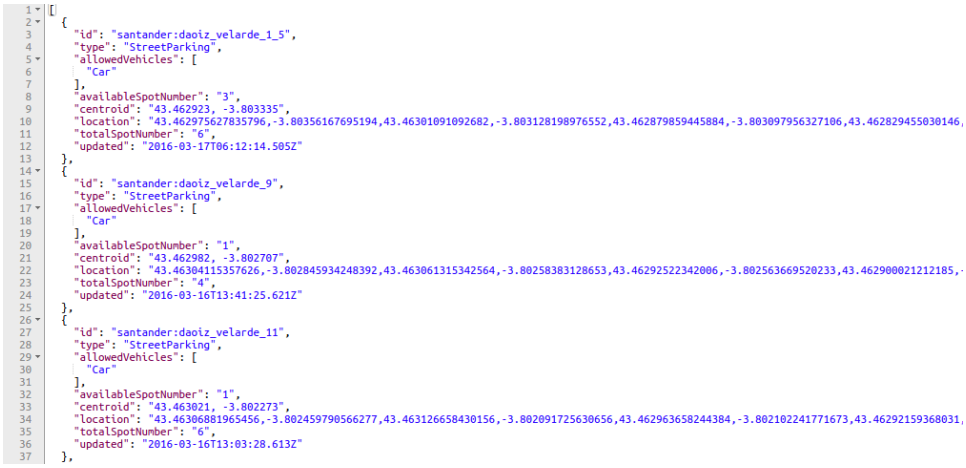


```
{
  "contextResponses": [
    {
      "contextElement": {
        "type": "santander:sound",
        "isPattern": "false",
        "id": "urn:smartsantander:testbed:237",
        "attributes": [
          {
            "name": "Latitud",
            "type": "urn:x-ogc:def:phenomenon:IDAS:1.0:latitude",
            "value": "43.46217",
            "metadatas": [
              {
                "name": "code",
                "type": "",
                "value": "deg"
              }
            ]
          },
          {
            "name": "Longitud",
            "type": "urn:x-ogc:def:phenomenon:IDAS:1.0:longitude",
            "value": "-3.80704",
            "metadatas": [
              {
                "name": "code",
                "type": "",
                "value": "deg"
              }
            ]
          },
          {
            "name": "TimeInstant",
            "type": "urn:x-ogc:def:trs:IDAS:1.0:ISO8601",
            "value": "2016-04-11T13:21:55.000000Z"
          },
          {
            "name": "batteryCharge",
            "type": "urn:x-ogc:def:phenomenon:IDAS:1.0:batteryCharge",
            "value": "70",
            "metadatas": [
              {
                "name": "code",
                "type": "",
                "value": "%"
              }
            ]
          },
          {
            "name": "sound",
            "type": "urn:x-ogc:def:phenomenon:IDAS:1.0:sound",
            "value": "50",
            "metadatas": [
              {
                "name": "code",
                "type": "",
                "value": "dB"
              }
            ]
          }
        ]
      },
```


  - The attributes returned for each entity are:
    - **Latitud**: The latitude where the measure were taken
    - **Longitud**: The longitude where the measure were taken
    - **TimeInstant**: Relates to the time when the information was extracted
    - **batteryCharge**: The level of battery of the given sensor
    - **altitude**: The altitude where the measure were taken.
    - **sound**: The current level of sound (given in dB) registered by the sensor

Last but not least, there is a NGSI v2 endpoint deployed on http://130.206.83.68:1026 and can be queried with simple HTTP GET requests. This service responses with a list of entities containing an id, a type and a set of attributes. This endpoint contains the following information:

- **Street Parking Lots**: Real time information of street parking lots retrieved from some sensors deployed on the street. The following screenshot shows the data returned from these sensors:



```
[
  {
    "id": "santander:daoiz_velarde_1_5",
    "type": "StreetParking",
    "allowedVehicles": [
      "Car"
    ],
    "availableSpotNumber": "3",
    "centroid": "43.462923, -3.803335",
    "location": "43.462975627835796,-3.80356167695194,43.46301091092682,-3.803128198976552,43.462879859445884,-3.803097956327106,43.462829455030146,
    "totalSpotNumber": "6",
    "updated": "2016-03-17T06:12:14.505Z"
  },
  {
    "id": "santander:daoiz_velarde_9",
    "type": "StreetParking",
    "allowedVehicles": [
      "Car"
    ],
    "availableSpotNumber": "1",
    "centroid": "43.462982, -3.802707",
    "location": "43.46304115357626,-3.802845934248392,43.463061315342564,-3.80258383128653,43.46292522342006,-3.802563669520233,43.462900021212185,
    "totalSpotNumber": "4",
    "updated": "2016-03-16T13:41:25.621Z"
  },
  {
    "id": "santander:daoiz_velarde_11",
    "type": "StreetParking",
    "allowedVehicles": [
      "Car"
    ],
    "availableSpotNumber": "1",
    "centroid": "43.463021, -3.802273",
    "location": "43.463068819654 56,-3.802459790566277,43.463126658430156,-3.802091725630656,43.462963658244384,-3.802102241771673,43.46292159368031,
    "totalSpotNumber": "6",
    "updated": "2016-03-16T13:03:28.613Z"
  },
```


  - As can be seen in the screenshot, *StreetParking* entities contain the following attributes:

- **allowedVehicles**: Type of vehicles that can park in the available spots
- **availableSpotNumber**: Number of available spots for parking
- **location**: Coordinates of the different points that build up the geo poligon which demarcate the parking area in WGS84 format
- **centroid**: Coordinates of the geo poligon centroid in WGS84 format
- **totalSpotNumber**: Total number of spots for parking
- **updated**: The last update date in ISO8601 format

You can find the endpoints and examples on how to query Santander Sensors in the Santander Sensors Postman

Porto (Portugal)

The city of Portugal makes use of FIWARE technology to provide citizens real time information. This information is provided through three different NGSI instances.

There is one NGSI v2 instance deployed on http://130.206.83.68:1026 that provides information of the different Ambient Areas in which the city of Porto is divided:

- **Ambient Zones**: The different ambient zones in which the city of Porto is divided. The following screenshot shows the data returned:


```
[
  {
    "id": "Porto-AmbientArea-Boavista",
    "type": "AmbientArea",
    "centroid": {
      "type": "coords",
      "value": "41.155471,-8.6289",
      "metadata": {}
    },
    "location": {
      "type": "geo:polygon",
      "value": "41.163854282188204,-8.633629796628234,41.16523029413228,-8.630357419242978,41.164981131496575,-8.627010741427831,41.16463453133977,-8.624898630516782,41.16308595014499,-8.62354
      "metadata": {}
    }
  },
```


  - Each entity contains one of the different ambient zones of Porto. For each entity, the following attributes are returned:
    - **centroid**: The center of the polygon
    - **location**: Coordinates of the different points that build up the geo poligon in in WGS84

There is NGSI v1 instance deployed on http://fiware-porto.citibrain.com:1026 that provides the following entities:

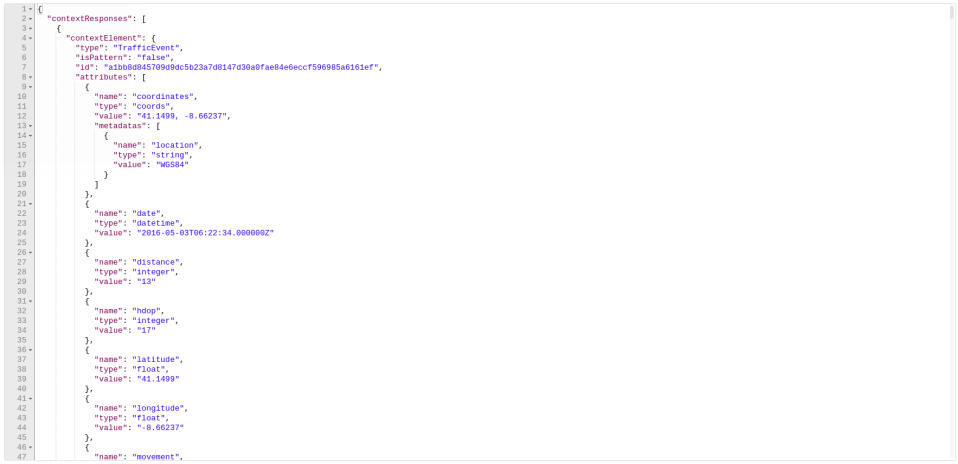
- **Environmental Events**: Environmental measures across the city of Porto. The following screenshot shows the data returned from those sensors:


```
{
  "contextResponses": [
    {
      "contextElement": {
        "type": "EnvironmentEvent",
        "isPattern": "false",
        "id": "8",
        "attributes": [
          {
            "name": "carbon_monoxide",
            "type": "float",
            "value": "1.21389"
          },
          {
            "name": "coordinates",
            "type": "coords",
            "value": "41.1629371643, -8.58449554443",
            "metadatas": [
              {
                "name": "location",
                "type": "string",
                "value": "WGS84"
              }
            ]
          },
          {
            "name": "humidity",
            "type": "float",
            "value": "77.1"
          },
          {
            "name": "nitrogen_dioxide",
            "type": "float",
            "value": "1.59365"
          },
          {
            "name": "noise_level",
            "type": "float",
            "value": "104.7"
          },
          {
            "name": "ozone",
            "type": "float",
            "value": "1.12238"
          },
          {
            "name": "processed_nitrogen_dioxide",
            "type": "float",
            "value": "4.605"
          },
          {
            "name": "processed_ozone",
            "type": "float",
            "value": "4.605"
          },
          {
            "name": "temperature",
            "type": "float",
            "value": "14.2"
          },
          {
            "name": "timestamp",
            "type": "string",
            "value": "2015-12-05T15:19:17Z"
          }
        ]
      },
      "statusCode": {
        "code": "200",
        "reasonPhrase": "OK"
      }
    },
```


  - Each entity contains the information provided by one sensor deployed on the city of Porto. The following attributes are returned:
    - **carbon_monoxide**: The amount of CO registered by the sensor

- **coordinates**: The position of the sensor
- **humidity**: % of relative humidity
- **nitrogen_dioxide**: The amount of NO_2 registered by the sensor
- **noise_level**: The amount of noise registered by the sensor
- **ozone**: The amount of O_3 registered by the sensor
- **processed_nitrogen_dioxide**: The amount of processed NO_2 registered by the sensor
- **processed_ozone**: The amount of processed O_3 registered by the sensor
- **temperature**: The temperature in ºC registered by the sensor
- **timestamp**: The time when the measures were taken

- **Traffic Events**: Traffic information about the city of Porto extracted from mobile sensors. The following screenshot shows the data returned by these sensors:



```
{
 "contextResponses": [
  {
   "contextElement": {
    "type": "TrafficEvent",
    "isPattern": "false",
    "id": "a1bb8d845709d9dc5b23a7d8147d30a0fae84e6eccf596985a6161ef",
    "attributes": [
     {
      "name": "coordinates",
      "type": "coords",
      "value": "41.1499, -8.66237",
      "metadatas": [
       {
        "name": "location",
        "type": "string",
        "value": "WGS84"
       }
      ]
     },
     {
      "name": "date",
      "type": "datetime",
      "value": "2016-05-03T06:22:34.000000Z"
     },
     {
      "name": "distance",
      "type": "integer",
      "value": "13"
     },
     {
      "name": "hdop",
      "type": "integer",
      "value": "17"
     },
     {
      "name": "latitude",
      "type": "float",
      "value": "41.1499"
     },
     {
      "name": "longitude",
      "type": "float",
      "value": "-8.66237"
     },
     {
      "name": "movement",
      "type": "integer",
      "value": "1"
     },
     {
      "name": "speed",
      "type": "integer",
      "value": "5"
     },
     {
      "name": "vehicle",
      "type": "string",
      "value": "a1bb8d845709d9dc5b23a7d8147d30a0fae84e6eccf596985a6161ef"
     }
    ]
   },
   "statusCode": {
    "code": "200",
    "reasonPhrase": "OK"
   }
  },
```


  - Each entity contains information about one traffic sensor that contains the following attributes:
    - **coordinates**: The current position of the sensor
    - **date**: The date when the information of the sensor was updated
    - **distance**: The distance covered by the sensor
    - **hdop**: The precision of the geographical information returned by the sensor GPS
    - **latitude**: The current latitude where the sensor is located
    - **longitude**: The current longitude where the sensor is located
    - **movement**: 1 if the sensor is in movement
    - **speed**: The current speed of the sensor
    - **vehicle**: The ID of the vehicle

Finally, there is another NGSI v1 endpoint deployed on https://api.ost.pt/ngsi10/contextEntityTypes that provides information about **garages** (category 9), **g as stations** (category 417) and **parking lots** (category 418). The following screenshot shows the data returned from these sensors:

```
{
  "contextResponses": [
    {
      "contextElement": {
        "attributes": [
          {
            "type": "string",
            "name": "name",
            "value": "Pascoal & Salgado - Reparação de Automóveis Lda."
          },
          {
            "type": "coords",
            "name": "geom_feature",
            "value": "40.279146, -8.571103",
            "metadatas": [
              {
                "type": "string",
                "name": "location",
                "value": "WGS84"
              }
            ]
          },
          {
            "type": "datetime",
            "name": "last_modified",
            "value": "2014-12-17 21:38:58.861327"
          },
          {
            "type": "datetime",
            "name": "publication_date",
            "value": null
          },
          {
            "type": "datetime",
            "name": "start_time",
            "value": null
          },
          {
            "type": "datetime",
            "name": "end_time",
            "value": null
          },
          {
            "type": "boolean",
            "name": "is_crowdsource",
            "value": "false"
          },
          {
            "type": "string",
            "name": "parent_poi",
            "value": null
          },
          {
            "type": "string",
            "name": "location",
            "value": "Rua Principal 25"
          },
          {
            "type": "string",
            "name": "street",
            "value": "Rua Principal 25"
          },
          {
            "type": "string",
            "name": "parish",
            "value": "União das Freguesias de São Martinho de Árvore e Lamarosa"
          },
          {
            "type": "string",
            "name": "municipality",
            "value": "Coimbra"
          },
          {
            "type": "string",
            "name": "district",
            "value": "Coimbra"
          },
          {
            "type": "string",
            "name": "country",
            "value": "Portugal"
          },
          {
            "type": "string",
            "name": "user",
            "value": null
          },
          {
            "type": "object",
            "name": "metadata",
            "value": null
          },
          {
            "type": "array",
            "name": "categories",
            "value": [
              "Manutenção e Reparação"
            ]
          },
          {
            "type": "array",
            "name": "tags",
            "value": []
          }
        ],
        "type": "poi",
        "id": "224"
      },
      "statusCode": {
        "code": 200,
        "reasonPhrase": "OK"
      }
    },
```


As can be seen in the screenshot, *POIs* entities contain the following attributes:

- **name**: The name of the POI
- **geom_feature**: The coordinates where the POI is located
- **last_modified**: The last time when the information of the POI was updated
- **start_time**: The first time when the system retrieved information from that POI
- **end_time**: The expected time when the POI is not going to be updated any more
- **parent_poi**: The ID of the parent POI
- **location**: Textual information about the POI position
- **street**: The name of the street where the POI is located
- **parish**: The name of the parish where the POI is located

- **municipality**: The name of the city where the POI is located
- **district**: The name of the district where the POI is located
- **country**: The name of the country where the POI is located
- **metadata**: Extra information about the POI
- **categories**: The categories the POI belongs to
- **tags**: A set of tags to categorize the POI

You can find the endpoints and examples on how to query Oporto Sensors in the Oporto Sensors Postman

Valencia (Spain)

The city of Valencia does not provide its data in NGSI format yet. However, it relies on FIWARE technology to offer real time information in other formats such as GeoJSON. Concretely, you can access to the following real-time information about:

- **ValenBisi**: The system of bike renting provided by the city. The following screenshot shows the data returned from these sensors:


```
{
  "type": "Feature",
  "properties": {
    "name": "115_C/ DR. VICENTE ZARAGOZA",
    "number": "115",
    "address": "Cataluña - Doctor Vicente Zaragozá",
    "open": "T",
    "available": "1",
    "free": "15",
    "total": "16",
    "ticket": "F",
    "updated_at": "03/05/2016 16:45:58"
  },
  "geometry": {
    "type": "Point",
    "coordinates": [
      727735.506,
      4373565.502
    ]
  }
},
```


  - As can be seen in the screenshot, each feature relates to one bike rental station and the following properties are attached:
    - **name**: The name of the station
    - **number**: The number of the street where the station is located
    - **address**: The name of the street where the station is located
    - **open**: "T" if the station is open, "F" otherwise
    - **available**: The number of bikes available
    - **free**: The number of free bike terminals
    - **total**: The total number of bike terminals
    - **ticket**: "T" if payment with credit card is allowed, "F" otherwise
    - **updated_at**: the time when the information was updated

- **Traffic**: Traffic information if real time in segments. The following screenshot shows the data returned from these sensors:


```
      "type": "Feature",
      "properties": {
        "denominacion": "SAN VICENTE (DE GIORGETA A TOMAS SALA)",
        "estado": "0",
        "idtramo": "8"
      },
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [
            724892.833,
            4369607.29
          ],
          [
            724900.229,
            4369637.225
          ],
          [
            724900.386,
            4369637.818
          ],
          [
            724900.564,
            4369638.404
          ],
          [
            724900.76,
            4369638.985
          ],
          [
            724900.977,
            4369639.558
```


    - As can be seen, each segment contains a "estado" field that represents the state of the segment:
        - 0: Fluid
        - 1: Heavy
        - 2: Congested
        - 3: Closed

You can find the endpoints and examples on how to query Valencia Sensors in the Valencia Sensors Postman

Slides used in the OpenHack

FIWARE Training Sesion