FIWARE Global Summit

From Data to Value

OPEN SOURCE
OPEN STANDARDS
OPEN COMMUNITY

Digital Twin Intermediate Programming using NGSI-LD

Temporal Operations, GeoFencing

Jason Fox, Technical Evangelist, FIWARE Foundation

Vienna, Austria
12-13 June, 2023
#FIWARESummit

FIWARE
Open APIs for Open Minds

Useful links

Latest NGSI-LD specification:
https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.06.01_60/gs_CIM009v010601p.pdf

NGSI-LD Tutorials:
https://ngsi-ld-tutorials.readthedocs.io/

Swagger Specification
https://forge.etsi.org/rep/NGSI-LD/NGSI-LD/raw/master/spec/updated/generated/full_api.json

Guidelines for Creating NGSI-LD Models:
https://github.com/smart-data-models/data-models/blob/master/guidelines.md

Semantic Modelling with NGSI-LD Whitepaper:
https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp_42_NGSI_LD.pdf

Content-Type Header

Supported Content-Types

- `application/json`
- `application/ld+json`

Default is `application/json`, in which case the `@context` must be supplied in a `Link` header see: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Link

`Link` Header is to be preferred as it reduces the size of the payloads

Follow JSON-LD best practices.
see https://w3c.github.io/json-ld-bp

```
{
   "id": "http://dbpedia.org/resource/John_Lennon",
   "type": "Person",
   "name": {"type": "Property", "value": "John Lennon"},
   "born": {"type": "Property", "value": "1940-10-09"},
   "spouse": {"
      "type": "Relationship",
      "object": "http://dbpedia.org/resource/Cynthia_Lennon"
    }
}
```

```
{
   "@context":  [
       "https://fiware.github.io/data-models/context.jsonld",
       "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld"
    ],
   "id": "http://dbpedia.org/resource/John_Lennon",
   "type": "Person",
   "name": {"type": "Property", "value": "John Lennon"},
   "born": {"type": "Property", "value": "1940-10-09"},
   "spouse": {"
      type": "Relationship",
     "object": "http://dbpedia.org/resource/Cynthia_Lennon"
    }
}
```

```
'Link: <http://.../path-to-my-public-server/ngsi-context.jsonld>;
rel="http://www.w3.org/ns/json-ld#context"; type="application/ld+json"'
```


FIWARE

Accept Header for GET /entities and Subscription payloads

Supported Accept Types

- `application/json` - `@context` is returned in a `Link` header
- `application/ld+json` - `@context` is returned in the payload body
- `application/geo+json` - GeoJSON response for GET /entities and subscriptions
  see https://tools.ietf.org/html/rfc7946

The fallback for error messages is `application/json`

Common NGSI-LD Formats

- `options=normalized`
- `options=concise`
- `options=keyValues`

Custom Formats **may** be supported by selected context brokers:

- `options=x-ngsiv2-normalized`
- `options=x-ngsiv2-keyValues`
- `options=x-ngsiv2-keyValues-compacted`

Custom NGSI-LD Formats should be used connection to microservices only
Do not use them for data exchange

GeoJSON request example

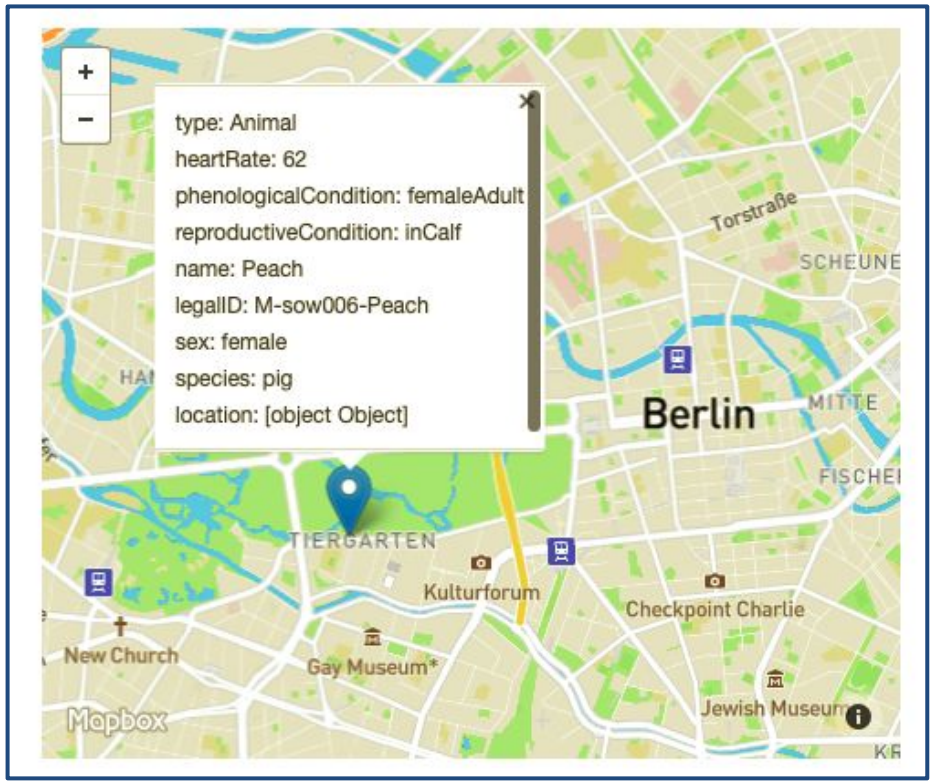
Give me all **Animal** entities which are **pigs inCalf** to be found within 2km of 13.364°N 52.52°E
… and return the data as key-value pairs in GeoJSON format without an `@context` attribute

```
curl -G 'http://localhost:1026/ngsi-ld/v1/entities/' \
  -d 'georel=near;maxDistance==2000' \
  -d 'geometry=Point' \
  -d 'coordinates=%5B13.364,52.52%5D' \
  -d 'q=species==%22pig%22;reproductiveCondition==%22inCalf%22' \
  -d 'type=Animal' \
  -d 'options=keyValues' \
  -H 'NGSILD-Tenant: openiot' \
  -H 'Accept: application/geo+json' \
  -H 'Prefer: body=json'  \
  -H 'Link: <http://.../path-to-my-public-server/ngsi-context.jsonld>;
rel="http://www.w3.org/ns/json-ld#context"; type="application/ld+json'
```

Use `Prefer=ld+json` to return in GeoJSON-LD format
see https://geojson.org/geojson-ld/

GeoJSON response example


```
{
    "type": "FeatureCollection",
    "features": [
        {
            "id": "urn:ngsi-ld:Animal:pig016",
            "type": "Feature",
            "properties": {
                "type": "Animal",
                "heartRate": 62,
                "phenologicalCondition": "femaleAdult",
                "reproductiveCondition": "inCalf",
                "name": "Tango",
                "legalID": "F-sow016-Tango",
                "sex": "female",
                "species": "pig",
                "location": {
                   "type": "Point",
                   "coordinates": [13.355, 52.523]
                }
            },
            "geometry": {
                "type": "Point",
                "coordinates": [ 13.355, 52.523]
            }
        },
...etc
    ]
}
```

- Since entities typically have a `location` they can be plotted onto a map.
- GeoJSON is used as an output format only.
- Any GeoJSON `Feature` and/or `FeatureCollection` can be easily digested by any GIS system.

FIWARE

Normalized vs Concise Payload

Concise **Property** Format

Normalized Property

```
{
    "temperature": {
        "type" : "Property",
        "value" : 100,
    }
}
```

Concise Property

```
{
    "temperature": {
        "value" : 100,
    }
}
```

Super Concise Property

```
{
    "temperature": 100
}
```

- `type` is optional
- `value` is optional (if no sub-attributes present)

The concise format is shorter than normalized but unlike key-values it is still lossless.

Concise **GeoProperty** format

Normalized GeoProperty

```
{
    "location": {
        "type" : "GeoProperty",
        "value" : {
          "type": "Point",
          "coordinates": [-73.97, 40.77]
        }
    }
}
```

Concise GeoProperty

```
{
    "location": {
        "value" : {
          "type": "Point",
          "coordinates": [-73.97, 40.77]
        }
    }
}
```

Super Concise GeoProperty

```
{
    "location": {
        "type": "Point",
        "coordinates": [-73.97, 40.77]
    }
}
```

- `type` is optional
- `value` is optional (if no sub-attributes present)
- **GeoProperty** is inferred if the `type` is a supported GeoJSON type.

FIWARE

Concise **Relationship** format

Normalized Relationship

```
{
    "providedBy": {
        "type" : "Relationship",
        "object" : "urn:ngsi-ld:Entity:001"
    }
}
```

Concise Relationship

```
{
    "providedBy": {
        "object" : "urn:ngsi-ld:Entity:001"
    }
}
```

- **`type`** is optional
- **`object`** is mandatory

FIWARE

Concise **LanguageProperty** format

Input and Output format. Potentially NGSI-LD 1.6.1

Normalized LanguageProperty

```
{
  "name": {
      "type": "LanguageProperty",
      "languageMap": {
          "el": "Κωνσταντινούπολις",
          "en": "Constantinople",
          "tr": "İstanbul"
      }
  }
}
```

- `type` is optional
- `languageMap` is mandatory

Concise LanguageProperty (as Map)

```
{
  "name": {
      "languageMap": {
          "el": "Κωνσταντινούπολις",
          "en": "Constantinople",
          "tr": "İstanbul"
      }
  }
}
```

Concise LanguageProperty (as Property)

```
{
  "name": {
    "value": "Constantinople",
    "lang": "en"
  }
}
```


FIWARE

Concise Format allowed for all `/entities` endpoints

- **GET** `/ngsi-ld/v1/entities/?options=concise`
- **POST** `/ngsi-ld/v1/entities/`
- **GET** `/ngsi-ld/v1/entities/<entity-id>?options=concise`
- **POST** **PATCH** `/ngsi-ld/v1/entities/<entity-id>/attrs`
- **PATCH** `/ngsi-ld/v1/entities/<entity-id>/attrs/<attr-id>`
- Plus all relevant Batch Operation endpoints:
  POST `/ngsi-ld/v1/entityOperations/xxx`

FIWARE

Temporal Interface

NGSI-LD Temporal interface

Give me the last 5 readings about a single entity and return in default (normalized) format:

```
curl -G -X GET 'http://localhost:8080/temporal/entities/urn:ngsi-ld:Animal:cow001' \
  -d 'lastN=5' \
  -H 'NGSILD-Tenant: openiot' \
  -H 'Link: <http://.../path-to-my-public-server/ngsi-context.jsonld>;
rel="http://www.w3.org/ns/json-ld#context"; type="application/ld+json"'
```

- Temporal endpoints are found under `/temporal/entities`
- Temporal endpoints are **optional** - not directly supported by all context brokers
- Gives a context broker a “memory” at the cost of data storage and maintenance.
- Expect a performance hit - don’t run as `DEBUG`

Sample docker-compose:
https://github.com/FIWARE/tutorials.Short-Term-History/blob/NGSI-LD/docker-compose/orion-ld.yml

Normalized Temporal request

- 1.4 kB Unlimited Temporal Responses get very long very quickly


```
{
  "id": "urn:ngsi-ld:Animal:cow001",
  "type": "Animal",
  "legalID": [
    {
      "type": "Property",
      "value": "M-bull001-Beany",
      "instanceId": "urn:ngsi-ld:attribute:instance:ec12e7fc-a45d-11eb-a739-0242ac120106"
    },
    … etc
  ],
  "name": [
    {
      "type": "Property",
      "value": "Beany",
      "instanceId": "urn:ngsi-ld:attribute:instance:ec1284c4-a45d-11eb-a739-0242ac120106"
    },
    … etc
  ],
  "sex": [
    {
      "type": "Property",
      "value": "male",
      "instanceId": "urn:ngsi-ld:attribute:instance:ec12aad0-a45d-11eb-a739-0242ac120106"
    },
    … etc
  ],

...etc
  ],
```


The following are mandated by the core **`@context`**

- `value`
- `unitCode`
- `observedAt`


```
  "location": [
    {
      "type": "GeoProperty",
      "value": {
        "type": "Point",
        "coordinates": [13.409,52.471,0]
      },
      "observedAt": "2021-04-26T09:35:16.814Z",
      "instanceId": "urn:ngsi-ld:attribute:...",
      "providedBy": {
        "object": "urn:ngsi-ld:Device:cowcollar001",
        "type": "Relationship",
        "instanceId": "urn:ngsi-ld:attribute:..."
      }
    },
    … etc
  ],

    "heartRate": [
    {
      "type": "Property",
      "value": 52,
      "observedAt": "2021-04-26T09:35:16.814Z",
      "instanceId": "urn:ngsi-ld:attribute:..",
      "unitCode": "5K",
      "providedBy": {
        "object": "urn:ngsi-ld:Device:cowcollar001",
        "type": "Relationship",
        "instanceId": "urn:ngsi-ld:attribute:..."
      }
    },
 ...etc
  ]
}
```


FIWARE

Temporal Queries on attributes without `observedAt`

Give me the last 5 readings about all **female Animals**, and return them 2 at a time

```
curl -G -X GET 'http://localhost:8080/temporal/entities' \
  -d 'type=Animal' \
  -d 'pageSize=2' \
  -d 'lastN=5' \
  -d 'q=sex==%22female%22' \
  -d 'timeproperty=modifiedAt' \
  -d 'options=count' \
  -H 'NGSILD-Tenant: openiot' \
  -H 'Link: <http://.../path-to-my-public-server/ngsi-context.jsonld>;
rel="http://www.w3.org/ns/json-ld#context"; type="application/ld+json"
```

- Default temporal attribute is `observedAt`.
- static attributes are usually not observed - cannot be queried in the `q` parameter directly
- Use `timeproperty=modifiedAt` to query static properties

Temporal Response including `modifiedAt`

```
{        "id": "urn:ngsi-ld:Animal:cow003",
        "type": "Animal",
        "heartRate": [
            {
                "type": "Property",
                "value": 51.0,
                "observedAt": "2021-04-26T09:36:36.577Z",
                "modifiedAt": "2021-04-26T09:38:09.579Z",
                "instanceId": "urn:ngsi-ld:attribute:instance:627f4202-a673-11eb-89a1-0242ac120106",
                "unitCode": "5K",
                "providedBy": {
                    "object": "urn:ngsi-ld:Device:cowcollar003",
                    "type": "Relationship",
                    "modifiedAt": "2021-04-26T09:38:09.579Z",
                    "instanceId": "urn:ngsi-ld:attribute:instance:62816672-a673-11eb-89a1-0242ac120106"
                }
            }
 … etc
```

- `modifiedAt` is returned in the response.
- There may be a significant lag between `observedAt` and `modifiedAt`
- `modifiedAt` identifies the last confirmed value, not necessarily the last change of value

Pagination options

Query Parameters

- `lastN` - limits the number of returned Attributes
- `pageSize` - limits the number of returned Entities
- `pageAnchor` - id of the first returned Entity
- `options=count` - includes the number of entities as a header in the response

Relevant Headers in response

- `Content-Range` - `date-time 2021-04-26T09:41:15.752-2021-04-26T09:29:10.834/5`
- `NGSILD-Results-Count` - 174
- `Page-Size` - 2
- `Next-Page` - `urn:ngsi-ld:Animal:cow004`

```
curl -G -X GET
'http://localhost:8080/temporal/entities' \
  -d 'type=Animal' \
  -d 'pageSize=2' \
  -d 'lastN=5' \
  -d 'q=sex==%22female%22' \
  -d 'timeproperty=modifiedAt' \
  -d 'options=count' \
  -d pageAnchor=urn:ngsi-ld:Animal:cow004 \
...etc
```

Time limiting and Geofencing Temporal Queries

Give me the `heartRate`, `location` and `controlledAsset` attributes of all **Device** entities, found within 800m of 13.364°N 52.52°E and return all readings taken since 8:30 a.m on 22nd April, returning them 2 devices at a time and in temporal values format

```
curl -L -g -X GET 'http://localhost:8080/temporal/entities' \
  -d 'type=Device' \
  -d 'attrs=location,controlledAsset' \
  -d 'options=temporalValues' \
  -d 'georel=near%3BmaxDistance==800' \
  -d 'geometry=Point' \
  -d 'coordinates=[13.364,52.52]' \
  -d 'timerel=after' \
  -d 'timeAt=2021-04-22T08:33:51.255Z' \
  -d 'pageSize=2' \
  -H 'NGSILD-Tenant: openiot' \
  -H 'Link: <http://.../path-to-my-public-server/ngsi-context.jsonld>;
rel="http://www.w3.org/ns/json-ld#context"; type="application/ld+json"' \
  -H 'Accept: application/json'
```

Temporal Values Response

```
[
    {
        "id": "urn:ngsi-ld:Device:pigcollar001",
        "type": "Device",
        "heartRate": {
            "type": "Property",
            "values": [
                [ 61.0, "2021-04-26T08:55:56.100Z"]
                ...etc
            ]
        },
        "location": {
            "type": "GeoProperty",
            "values": [
                [{"type": "Point", "coordinates": [13.355, 52.516, 0.0]},"2021-04-26T08:55:56.100Z"],
                ...etc
            ]
        },
        "controlledAsset": {
            "type": "Relationship",
            "objects": [
                ["urn:ngsi-ld:Animal:pig001", "2021-04-26T08:55:56.100Z"],
                ... etc
            ]
        }
    },
    … etc
]
```

- The response holds an array of attribute value-time stamp pairs for each observed reading.
- Properties are held in `values` arrays, Relationships use `objects`

RE

Updating Entities

Partial Update vs Merge Patch

Partial Update of an Entity

PATCH `/ngsi-ld/v1/entities/<entity-id>/attrs`

Original Entity

```
{
    "id": "urn:ngsi-ld:Sensor:001",
    "type": "TemperatureSensor",
    "temperature": {
        "type" : "Property",
        "value" : 25,
        "unitCode": "CEL"
        "observedAt": "2022-01-01"
    }
}
```

Normalized Payload:

```
{
    "temperature": {
        "type" : "Property",
        "value" : 100,
        "observedAt": "2022-03-14"
    }
}
```

Result: Updated Entity

```
{
    "id": "urn:ngsi-ld:Sensor:001",
    "type": "TemperatureSensor",
    "temperature": {
        "type" : "Property",
        "value" : 100,
        "observedAt": "2022-03-14"
    }
}
```

- `value` updated to 100
- `observedAt` updated
- `unitCode` **removed**
- Other Attributes unchanged

`temperature` attribute **replaced** with payload contents

FIWARE

Partial Update of an Attribute

PATCH `/ngsi-ld/v1/entities/<entity-id>/attrs/temperature`

Original Entity

```
{
    "id": "urn:ngsi-ld:Sensor:001",
    "type": "TemperatureSensor",
    "temperature": {
        "type" : "Property",
        "value" : 25,
        "unitCode": "CEL",
        "observedAt": "2022-01-01"
    }
}
```

Normalized Payload:

```
{
    "type" : "Property",
    "value" : 100,
    "observedAt": "2022-03-14"
}
```

- `value` updated to 100
- `observedAt` updated
- `unitCode` **not** removed
- Other Attributes unchanged

Result: Updated Entity

```
{
    "id": "urn:ngsi-ld:Sensor:001",
    "type": "TemperatureSensor",
    "temperature": {
        "type" : "Property",
        "value" : 100,
        "unitCode": "CEL"
        "observedAt": "2022-03-14"
    }
}
```

`temperature` sub-attribute **replaced** with payload contents

FIWARE

Merge of an Entity (1) - **Normalized** Payload Support

PATCH `/ngsi-ld/v1/entities/<entity-id>`

Original Entity

```
{
    "id": "urn:ngsi-ld:Sensor:001",
    "type": "TemperatureSensor",
    "temperature": {
        "type" : "Property",
        "value" : 25,
        "unitCode": "CEL",
        "observedAt": "2022-01-01"
    }
}
```

Normalized Payload:

```
{
    "temperature": {
        "type" : "Property",
        "value" : 100,
    }
}
```

Result: Merged Entity

```
{
    "id": "urn:ngsi-ld:Sensor:001",
    "type": "TemperatureSensor",
    "temperature": {
        "type" : "Property",
        "value" : 100,
        "unitCode": "CEL",
        "observedAt": "2022-01-01"
    }
}
```

- `value` updated to 100
- `observedAt` **not** removed
- `unitCode` **not** removed
- Other Attributes unchanged

Values from the payload contents **merged** with existing entity. Unchanged data does not need to be supplied

FIWARE

Merge of an **Entity** (2) - **Concise** Payload Support

PATCH `/ngsi-ld/v1/entities/<entity-id>`

Merge means unchanged data no longer needs to be supplied

Normalized Payload

```
{
    "temperature": {
        "type" : "Property",
        "value" : 100,
    }
}
```

Concise Property Payload

```
{
    "temperature": {
        "value" : 100,
    }
}
```

Super Concise Property

```
{
    "temperature" 100
}
```

- `value` updated to 100
- `observedAt` **not** removed
- Other sub-attributes (e.g. `unitCode`) **not** removed
- Other Attributes unchanged

Values from the payload contents **merged** with existing entity.

FIWARE

Merge of an **Entity** (3) - **Key-Values** Payload Support

PATCH `/ngsi-ld/v1/entities/<entity-id>?options=keyValues`

Indicates a lossy payload where only **values** have been supplied

Key-Values Payload (Lossy)

```
{
  "name": "John Ono Lennon",
  "spouse": "http://dbpedia.org/resource/Yoko_Ono"
}
```

Result

- `name` - **Property** type is maintained. `value` updated
- `spouse` - **Relationship** type is maintained. `object` updated
- Other attributes (e.g. `born`) remain unchanged
- All sub-attributes remain unchanged

Values from the payload contents intelligently **merged** with existing entity.

FIWARE

Filters

Use filtering for Queries and Subscriptions

Filtering using the q parameter

- equal - ==
- unequal - !=
- greater - >
- greaterEq - >=
- less - <
- lessEq - <=
- regex pattern - ~=
- not regex Pattern - !~=
- dots (range) - ..
- andOp - ;
- orOp - |

```
?q=((speed>50|rpm>3000);brandName=="Mercedes")

?q=(temperature>=20;temperature<=25)|capacity<=10

?q=(temperature==20..25)|capacity<=10

?q=address[city]!="D%C3%BCsseldorf"

?q=temperature.observedAt>=2017-12-24T12:00:00Z

?q=category=="barn","farm_auxiliary"
```

The geoQ parameters

- `geometry` - any supported GeoJSON type
- `coordinates`
- `georel`
  - `near;maxDistance`
  - `near;minDistance`
  - `within`
  - `contains`
  - `intersects`
  - `equals`
  - `disjoint`
  - `overlaps`

- `geoproperty` - Optional default is `location`

```
?georel=near;maxDistance==2000
   &geometry=Point
   &coordinates=[8,40]
   &geoproperty=observationSpace


?georel=within&
  geometry=Polygon&
  coordinates=[[[100.0,0.0],[101.0,0.0],
    [101.0,1.0],[100.0,1.0],[100.0,0.0]]]&
  geoproperty=location
```

The `temporalQ` parameters

- `timeAt` - any DateTime
- `endTimeAt` - any DateTime
- `timerel`
  - `before`
  - `after`
  - `between`

- `timeproperty` - Optional default is `observedAt`

```
?timerel=before&
  timeAt=2020-04-13T14:20:00Z&
  timeproperty=modifiedAt


?timerel=between&
  timeAt=2021-04-26T09:00:00Z&
  endTimeAt=2021-05-21T14:40:00Z&
  timeproperty=observedAt
```

FIWARE Global Summit

Find Us On

Stay up to date

JOIN OUR NEWSLETTER

Be certified and featured

FIWAREMarketplace

Hosting Partner

City of Vienna

vienna business agency

Keystone Sponsors

aws

ms.gis

Raseel

Red Hat

Media Partners

bee smart city

CITIES FORUM

CitiesToday

Our Future Water

RM RENEWABLE MATTER

Smart Cities World

the smartcity journal

FIWARE
Global
Summit

Thanks!

Vienna, Austria
12-13 June, 2023
#FIWARESummit

FIWARE
Open APIs for Open Minds