

ODA - DATA FORMAT

Gianpaolo Cugola (gianpaolo.cugola@polimi.it) - Alessandro Margara (alessandro.margara@polimi.it)
Luca Mottola (luca.mottola@polimi.it) - Matteo Visotto (matteo.visotto@polimi.it)

The data published in ODA by the Fiware-ODA Bridge is represented as a valid JSON object, adhering to the structure and syntax defined by RFC 8259. This ensures that the data is serialized in a standardized format, allowing for consistent interpretation and interoperability across systems that consume JSON-based data.

The object includes a fixed set of attributes that identify the record, along with an inner object containing a distinct set of attributes for the actual data.

The fixed attributes are:

Attribute name	Attribute description
timestamp	Data generation timestamp
generator_id	Bridge Identifier + Unique identifier of the source
topic	Kafka topic the record is published on
data	JSON object containing the actual data

Table 1. ODA object fixed attributes

For instance, the `AirQuality` sensor `sc-40659` generates the following object, ignoring the data content:

```
{
  "timestamp":1732641538000,
  "generator_id":"neslabpolimi_bridge_sc-40659",
  "topic":"AirQualitySensor",
  "data":{ }
}
```

Data object

We have designed the data object in such a way that each data is represented by a key in the JSON object. The corresponding value for each key is a JSON object containing both the actual data value and its associated unit of measurement.

```
{
  "attribute_name_1":{
    "value": "Any supported type"
    "unit": "String or None"
  },
  "attribute_name_2":{
    "value": "Any supported type"
    "unit": "String or None"
  }
}
```

This structure provides several advantages: it allows for easy retrieval of individual data using the attribute name as the key, and it also facilitates the extraction of all data by iterating over the key-value pairs. This approach enhances flexibility and ensures consistency in handling both the data values and their respective

units across the entire dataset.

For example, retrieving the value and unit of `attribute_name_1` from a received ODA message in Python requires only:

```
value = oda_object['data']['attribute_name_1']['value']
unit = oda_object['data']['attribute_name_1']['unit']
```

and printing all the data attributes and their corresponding values:

```
for key, value in oda_object['data'].items():
    data_value = value['value']
    unit = value['unit']
    print(f"{key} - Value: {data_value}, Unit: {unit}")
```

Below, we provide an example for each data source from which we publish information.

Example: Mitreo Temperature and Humidity Sensors

```
{
  "timestamp":1732641538000,
  "generator_id":"neslabpolimi_bridge_14",
  "topic":"MitreoTHSensor",
  "data":{
    "deploymentSite":{
      "value":"urn:ngsi-ld:DeploymentSite:Mitreo",
      "unit":"None"
    },
    "humidity":{
      "value":0,
      "unit":"%"
    },
    "location":{
      "value":"41.887882;12.482475",
      "unit":"latitude;longitude"
    },
    "pm10":{
      "value":28.77,
      "unit":"µg/m³"
    },
    "pm25":{
      "value":13.27,
      "unit":"µg/m³"
    },
    "spectralPower":{
      "value":6.280822587,
      "unit":"W/m"
    },
    "temperature":{
      "value":289.56,
      "unit":"K"
    }
  }
}
```

Example: Sensor Community Milano Air Quality

```
{
  "timestamp":1732636993000,
  "generator_id":"neslabpolimi_bridge_sc-40659",
  "topic":"AirQualitySensor",
  "data":{
    "co":{
      "value":210.9,
      "unit":"None"
    },
    "deploymentGroup":{
      "value":"urn:ngsi-ld:DeploymentGroup:Milano-sensor.community",
      "unit":"None"
    },
    "deploymentSite":{
      "value":"urn:ngsi-ld:DeploymentSite:Milano",
      "unit":"None"
    },
    "humidity":{
      "value":77.2,
      "unit":"%"
    },
    "indoor":{
      "value":false,
      "unit":"None"
    },
    "location":{
      "value":"45.544;9.232",
      "unit":"latitude;longitude"
    },
    "manufacturer":{
      "value":"Nova Fitness",
      "unit":"None"
    },
    "nh3":{
      "value":"None",
      "unit":"None"
    },
    "no":{
      "value":"None",
      "unit":"None"
    },
    "no2":{
      "value":"None",
      "unit":"None"
    },
    "o3":{
      "value":"None",
      "unit":"None"
    },
    "pm10":{
      "value":34.45,
      "unit":"µg/m³"
    },
    "pm25":{
```

```

        "value":22.33,
        "unit":"µg/m³"
    },
    "sensorName":{
        "value":"SDS011",
        "unit":"None"
    },
    "so2":{
        "value":"None",
        "unit":"None"
    },
    "temperature":{
        "value":"286.3",
        "unit":"K"
    }
}
}

```

Example: UpTown Energy Distribution

```

{
    "timestamp":1712444401,
    "generator_id":"neslabpolimi_bridge_02",
    "topic":"UpTownEnergyDistributionSensor",
    "data":{
        "deploymentGroup":{
            "value":"urn:ngsi-ld:DeploymentSite:BirdNESt",
            "unit":"None"
        },
        "deploymentSite":{
            "value":"urn:ngsi-ld:DeploymentSite:UpTown",
            "unit":"None"
        },
        "energy_consumption":{
            "value":"None",
            "unit":"kWh"
        },
        "external_temperature":{
            "value":"None",
            "unit":"°C"
        },
        "heating_power":{
            "value":"None",
            "unit":"kW"
        },
        "heating_temperature_delivery":{
            "value":"None",
            "unit":"°C"
        },
        "heating_temperature_return":{
            "value":"None",
            "unit":"°C"
        },
        "heating_volume":{
            "value":"None",
            "unit":"m³/h"
        }
    }
}

```

```
    },
    "pod_coordinates": {
      "value": "9.100249508;45.511895182",
      "unit": "latitude;longitude"
    },
    "pod_identifier": {
      "value": "72161307",
      "unit": "None"
    },
    "user_identifier": {
      "value": "02",
      "unit": "None"
    },
    "user_type": {
      "value": "Residenziale",
      "unit": "None"
    }
  }
}
```