

# ACQUAOUNT PLATFORM

## Use Manual

### Description Of The Platform

The ACQUAOUNT Platform is a web server hosting many Web Of Things Things. Things, in this use case, represent a field in the real world (to be more precise, a physical location). Everything that can be done with this thing is specified in the Thing Description, a document in JSON format.

Each Thing Description has 3 main sections apart from various separate attributes: the properties, the actions, and the events. Inside each of these sections everything needed to use them is described, from the body required to use them or the format of the data being returned.

The properties represent properties of the Thing, and in normal cases can be read or written (in this case, just read). The actions are physical actions that the Thing can do which can be invoked. Finally, the events are occurrences that happen at a specific moment to which the client can subscribe to to be notified when they happen.

There are two different types of Things in the platform: Fields and Weather Stations. Both types of things function in the same way, but don't have the exact same properties.

The properties available in a Field in the platform are:

- **fieldInformation:** Provides some static information about the field of this thing.
- **sensorsList:** Provides a list of the sensors in this field.
- **sensorInformation:** Provides detailed information about one of the sensors.
- **propertiesList:** Provides a list of all the properties being observed at this field.
- **propertyInformation:** Provides detailed information about one of the properties.
- **datastreamsList:** Provides a list of datastreams in the field. A datastream is the flow of data defined by the combination of a Thing, Sensor, and Property.
- **datastreamInformation:** Provides detailed information about one datastream.
- **datastreamLastMeasure:** Provides the last measure taken in a specified datastream.
- **datastreamMeasures:** Provides all the measures taken in a specified datastream.
- **lastMeasures:** Provides the last measures taken in all datastreams.

The properties available in a Weather Station in the platform are:

- **stationInformation:** Provides some static information about the station of this thing.
- **propertiesList:** Provides a list of all the properties being observed at this station.
- **propertyInformation:** Provides detailed information about one of the properties.
- **datastreamsList:** Provides a list of datastreams in the station.
- **datastreamInformation:** Provides detailed information about one datastream.
- **datastreamLastMeasure:** Provides the last measure taken in a specified datastream.
- **datastreamMeasures:** Provides all the measures taken in a specified datastream.
- **lastMeasures:** Provides the last measures taken in all datastreams.

An action available, in both the Fields and the Weather Stations, is:

- **receiveMeasure:** Used as an endpoint for the sensors to upload data to the platform.

A different action is planned to be made available in the Fields but not yet implemented:

- `sendCommand`: Sends a command to a sensor or device, such as change measurement frequency or perform an action such as closing a gate.

The only event available in both types of Things is:

- `newObservation`: This event is triggered every time there is a new measure in the system.

### Monitoring Platform

Alongside the WoT platform, a very simple monitoring platform has been created. This monitoring platform provides a graphical way to check the data available inside the WoT platform by specifying a pilot site and then one or more Datastreams.

The simple monitoring platform is available in <https://84.88.76.18/acquaount/> and supports all pilot sites, dynamically loading all fields that have Datastreams available in the Platform.

## How To Use The Platform

The platform can be used through HTTP requests or through the node-wot open-source client, which was also used to implement the platform. The various attributes can be tested by forking the Postman collection found [here](#).

Currently, the platform can be accessed through the following address:

[\*http://84.88.76.14/{{thingTitle}}\*](http://84.88.76.14/{{thingTitle}})

The IP seen is the address of the platform, and `{{thingTitle}}` must be the title of the Thing that is being accessed in all lowercase. There are multiple Things available in the platform, which, as said previously, can be of two different types: Fields or Weather Stations. All the Things of a different type use the same base Thing Description, with only small changes in the id, the title, the field or station name and the description.

### Properties

Properties can be read using a GET request. By making a request to [\*http://84.88.76.14/{{thingTitle}}/properties\*](http://84.88.76.14/{{thingTitle}}/properties), information of the properties to call will be returned. To access a specific property, the URL is:

[\*http://84.88.76.14/{{thingTitle}}/properties/{{propertyName}}\*](http://84.88.76.14/{{thingTitle}}/properties/{{propertyName}})

Some properties do not require any extra information, but some require a parameter in the URL to specify which instance in detail the user is interested in.

### Actions

The only action to be called is called using a POST request. Its URL is:

[\*http://84.88.76.14/{{thingTitle}}/actions/receiveMeasure\*](http://84.88.76.14/{{thingTitle}}/actions/receiveMeasure)

This request does not take parameters, but it takes a body. This body is a JSON object with two keys, "info" and "values". The value of the "info" key is an object with another key, "deviceId". The value of this key is the ID of the sensor that took the measurement. The value of the "values" key, on the other hand, is an object with multiple key/value pairs. In the "values" object, each key is the name of a property, and each value is the value observed in the measurement of that property. This way, each measure has a Property, Sensor, and Thing, so it specifies a Datastream where the measure will be stored.

It is vital that the specified Datastream is present in the Datastream Definition Google Sheet, or else the measures will not be uploaded to the platform.

### Events

Events cannot be subscribed to with standard HTTP requests since they require a handshake and constant connection. The only way to subscribe to an event is using the node-wot client, with the fetch command and the subscribeEvent handler. An example can be found in the source files of the project.

## Data Integration

To integrate data from external platforms, a Python service which fetches the data has been created. This service can run different jobs which fetch data from external sources, and then upload it to the platform as a standard Datastream, therefore, the external sources will also be listed in the Datastream Definition Google Sheet.

To specify an external source in the Datastream Definition, most attributes are to be filled in the same way as one would fill a normal Datastream, with a few exceptions:

- Device ID: Specify a new ID for each data source, differentiating from the way sensor IDs are specified, for example, use DX001 for an external data source related to the Demo pilot.
- Device Type and Description: Specify the type of device is “External”.
- Device EUI: Can be blank.

## Annex

### Field Base Thing Description

```
{
  "id": "acquaount:demosite",
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "title": "AcquaountThingBase",
  "fieldName": "Base",
  "href": "84.88.76.14",
  "securityDefinitions": {
    "basic_sc": {
      "scheme": "nosec"
    }
  },
  "security": [
    "basic_sc"
  ],
  "description": "Base Thing Description",
  "properties": {
    "fieldInformation": {
      "type": "object",
      "description": "The information of the site this thing represents",
      "readOnly": true,
      "writeOnly": false,
      "observable": false,
      "forms": [
        {
          "href": "/properties/thingInformation",
          "contentType": "application/json",
          "op": "readproperty"
        }
      ]
    },
    "properties": {
      "name": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The name of the field"
      },
      "description": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "A description about the field"
      },
      "pilot": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The pilot of this field"
      },
      "location": {
        "type": "object",
        "readOnly": true,
        "writeOnly": false,
        "format": "geojson",
        "href": "https://datatracker.ietf.org/doc/html/rfc7946#section-3.1",
        "description": "The location of the field in GeoJSON Geometry format"
      }
    }
  }
}
```

```

    },
    "sensorsList": {
      "type": "array",
      "description": "The list of sensors of this thing",
      "readOnly": true,
      "writeOnly": false,
      "observable": false,
      "forms": [
        {
          "href": "/properties/sensorList",
          "contentType": "application/json",
          "op": "readproperty"
        }
      ]
    },
    "items": {
      "type": "object",
      "readOnly": true,
      "writeOnly": false,
      "properties": {
        "name": {
          "type": "string",
          "readOnly": true,
          "writeOnly": false,
          "description": "The name of the sensor"
        },
        "description": {
          "type": "string",
          "readOnly": true,
          "writeOnly": false,
          "description": "A description of the sensor"
        },
        "properties": {
          "type": "object",
          "readOnly": true,
          "writeOnly": false,
          "description": "The properties of the sensor",
          "properties": {
            "device_id": {
              "type": "string",
              "readOnly": true,
              "writeOnly": false,
              "description": "The device's ID according to the list
(D001, D002, I001, ...)\"
            },
            "device_type": {
              "type": "string",
              "readOnly": true,
              "writeOnly": false,
              "description": "The device's type according to the list
(ST01, ST02, ...)\"
            },
            "device_eui": {
              "type": "string",
              "readOnly": true,
              "writeOnly": false,
              "description": "The device's EUI according to the list"
            }
          }
        }
      }
    },
    "sensorInformation": {
      "type": "object",
      "description": "The information of a sensor of this thing",

```

```

"readOnly": true,
"writeOnly": false,
"observable": false,
"uriVariables": {
  "device_id": {
    "type": "string",
    "readOnly": true,
    "writeOnly": false,
    "description": "The device ID of the sensor"
  }
},
"forms": [
  {
    "href": "/properties/sensorInformation",
    "contentType": "application/json",
    "op": "readproperty"
  }
],
"properties": {
  "name": {
    "type": "string",
    "readOnly": true,
    "writeOnly": false,
    "description": "The name of the sensor"
  },
  "description": {
    "type": "string",
    "readOnly": true,
    "writeOnly": false,
    "description": "A description of the sensor"
  },
  "properties": {
    "type": "object",
    "readOnly": true,
    "writeOnly": false,
    "description": "The properties of the sensor",
    "properties": {
      "device_id": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The device's ID according to the list
(D001, D002, I001, ...)"
      },
      "device_type": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The device's type according to the list
(ST01, ST02, ...)"
      },
      "device_eui": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The device's EUI according to the list"
      }
    }
  }
},
"propertiesList": {
  "type": "array",
  "description": "The list of observed properties of this thing",
  "readOnly": true,
  "writeOnly": false,

```

```

"observable": false,
"forms": [
  {
    "href": "/properties/propertiesList",
    "contentType": "application/json",
    "op": "readproperty"
  }
],
"items": {
  "type": "object",
  "readOnly": true,
  "writeOnly": false,
  "properties": {
    "name": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "The name of the property"
    },
    "description": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "A description of the property"
    }
  }
},
},
"propertyInformation": {
  "type": "object",
  "description": "The information of a property of this thing",
  "readOnly": true,
  "writeOnly": false,
  "observable": false,
  "uriVariables": {
    "name": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "The name of the property"
    }
  },
  "forms": [
    {
      "href": "/properties/propertyInformation",
      "contentType": "application/json",
      "op": "readproperty"
    }
  ],
  "properties": {
    "name": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "The name of the property"
    },
    "description": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "A description of the property"
    }
  }
},
},
"datastreamsList": {
  "type": "array",

```

```

    "description": "The list of datasources of this thing",
    "readOnly": true,
    "writeOnly": false,
    "observable": false,
    "forms": [
      {
        "href": "/properties/datasourcesList",
        "contentType": "application/json",
        "op": "readproperty"
      }
    ],
    "items": {
      "type": "object",
      "readOnly": true,
      "writeOnly": false,
      "properties": {
        "name": {
          "type": "string",
          "readOnly": true,
          "writeOnly": false,
          "description": "The name of the datasource"
        },
        "description": {
          "type": "string",
          "readOnly": true,
          "writeOnly": false,
          "description": "A description of the datasource"
        },
        "unit_of_measurement": {
          "type": "string",
          "readOnly": true,
          "writeOnly": false,
          "description": "The unit of measurement of the observations of
this datasource"
        }
      }
    }
  },
  "datastreamInformation": {
    "description": "The information of a datasource of this thing",
    "readOnly": true,
    "writeOnly": false,
    "observable": false,
    "uriVariables": {
      "name": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The name of the datasource"
      }
    }
  },
  "forms": [
    {
      "href": "/properties/propertyInformation",
      "contentType": "application/json",
      "op": "readproperty"
    }
  ],
  "type": "object",
  "properties": {
    "name": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "The name of the datasource"
    },

```

```

        "description": {
            "type": "string",
            "readOnly": true,
            "writeOnly": false,
            "description": "A description of the datasource"
        },
        "unit_of_measurement": {
            "type": "string",
            "readOnly": true,
            "writeOnly": false,
            "description": "The unit of measurement of the observations of this
datasource"
        }
    },
    "datastreamLastMeasure": {
        "description": "The last measure of the specified datastream",
        "readOnly": true,
        "writeOnly": false,
        "observable": false,
        "uriVariables": {
            "name": {
                "type": "string",
                "readOnly": true,
                "writeOnly": false,
                "description": "The name of the datastream"
            }
        },
        "forms": [
            {
                "href": "/properties/datasourceLastMeasure",
                "contentType": "application/json",
                "op": "readproperty"
            }
        ],
        "type": "object",
        "properties": {
            "device_id": {
                "type": "string",
                "readOnly": true,
                "writeOnly": false,
                "description": "The ID of the sensor that took this measure"
            },
            "property_name": {
                "type": "string",
                "readOnly": true,
                "writeOnly": false,
                "description": "The name of the property observed in this measure"
            },
            "datastream_name": {
                "type": "string",
                "readOnly": true,
                "writeOnly": false,
                "description": "The name of the datastream containing this measure"
            },
            "unit_of_measurement": {
                "type": "string",
                "readOnly": true,
                "writeOnly": false,
                "description": "The unit of measurement of the observation"
            },
            "value": {
                "type": "string",
                "readOnly": true,
                "writeOnly": false,
                "description": "The value of the observation"
            }
        }
    }
}

```

```

    },
    "time_of_measure": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "The time when the observation was taken"
    }
  },
  "datastreamMeasures": {
    "description": "All the measures of a specified datastream",
    "readOnly": true,
    "writeOnly": false,
    "observable": false,
    "uriVariables": {
      "name": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The name of the datasource"
      }
    }
  },
  "forms": [
    {
      "href": "/properties/datasourceMeasures",
      "contentType": "application/json",
      "op": "readproperty"
    }
  ],
  "type": "array",
  "items": {
    "readOnly": true,
    "writeOnly": false,
    "properties": {
      "device_id": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The ID of the sensor that took this measure"
      },
      "property_name": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The name of the property observed in this
measure"
      },
      "datastream_name": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The name of the datastream containing this
measure"
      },
      "unit_of_measurement": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The unit of measurement of the observation"
      },
      "value": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The value of the observation"
      }
    }
  },

```

```

        "time_of_measure": {
            "type": "string",
            "readOnly": true,
            "writeOnly": false,
            "description": "The time when the observation was taken"
        }
    },
    },
    "datastreamTimeRangeMeasures": {
        "description": "All the measures in a range of time of a specified
datastream",
        "readOnly": true,
        "writeOnly": false,
        "observable": false,
        "uriVariables": {
            "name": {
                "type": "string",
                "readOnly": true,
                "writeOnly": false,
                "description": "The name of the datasource"
            },
            "start_time": {
                "type": "string",
                "readOnly": true,
                "writeOnly": false,
                "description": "The start time of the range of time, in format
\\\"YYYY-MM-DDThh:mm:ssZ\\\" in the UTC timezone"
            },
            "end_time": {
                "type": "string",
                "readOnly": true,
                "writeOnly": false,
                "description": "The start time of the range of time, in format
\\\"YYYY-MM-DDThh:mm:ssZ\\\" in the UTC timezone"
            }
        },
        "forms": [
            {
                "href": "/properties/datasourceMeasures",
                "contentType": "application/json",
                "op": "readproperty"
            }
        ],
        "type": "array",
        "items": {
            "readOnly": true,
            "writeOnly": false,
            "properties": {
                "device_id": {
                    "type": "string",
                    "readOnly": true,
                    "writeOnly": false,
                    "description": "The ID of the sensor that took this measure"
                },
                "property_name": {
                    "type": "string",
                    "readOnly": true,
                    "writeOnly": false,
                    "description": "The name of the property observed in this
measure"
                },
                "datastream_name": {
                    "type": "string",
                    "readOnly": true,
                    "writeOnly": false,

```

```

        "description": "The name of the datastream containing this
measure"
    },
    "unit_of_measurement": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The unit of measurement of the observation"
    },
    "value": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The value of the observation"
    },
    "time_of_measure": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The time when the observation was taken"
    }
}

},
"lastMeasures": {
    "description": "All the latest measures of the thing",
    "readOnly": true,
    "writeOnly": false,
    "observable": false,
    "forms": [
        {
            "href": "/properties/lastMeasures",
            "contentType": "application/json",
            "op": "readproperty"
        }
    ],
    "type": "array",
    "items": {
        "readOnly": true,
        "writeOnly": false,
        "properties": {
            "device_id": {
                "type": "string",
                "readOnly": true,
                "writeOnly": false,
                "description": "The ID of the sensor that took this measure"
            },
            "property_name": {
                "type": "string",
                "readOnly": true,
                "writeOnly": false,
                "description": "The name of the property observed in this
measure"
            },
            "datastream_name": {
                "type": "string",
                "readOnly": true,
                "writeOnly": false,
                "description": "The name of the datastream containing this
measure"
            },
            "unit_of_measurement": {
                "type": "string",
                "readOnly": true,
                "writeOnly": false,
                "description": "The unit of measurement of the observation"
            }
        }
    }
}

```

```

    },
    "value": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "The value of the observation"
    },
    "time_of_measure": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "The time when the observation was taken"
    }
  }
}
},
"actions": {
  "receiveMeasure": {
    "safe": true,
    "idempotent": false,
    "input": {
      "type": "object",
      "description": "The message that the sensor sends",
      "readOnly": true,
      "writeOnly": false,
      "properties": {
        "info": {
          "type": "object",
          "readOnly": true,
          "writeOnly": false,
          "description": "The necessary information to identify from
which sensor is this data coming from",
          "properties": {
            "deviceID": {
              "type": "string",
              "readOnly": true,
              "writeOnly": false,
              "description": "The device's ID according to the list
(D001, D002, I001, ...)"
            }
          }
        }
      }
    },
    "values": {
      "type": "object",
      "readOnly": true,
      "writeOnly": false,
      "description": "The observation values in key/value pairs. The
keys must match the keys agreed on before.",
      "properties": {}
    }
  }
},
"output": {
  "description": "Returns true/false if measures were identified
correctly, and a message",
  "type": "object",
  "properties": {
    "result": {
      "type": "boolean",
      "readOnly": true,
      "writeOnly": false
    },
    "message": {
      "type": "string",
      "readOnly": true,

```

```

        "writeOnly": false
    },
    {
        "readOnly": true,
        "writeOnly": false
    },
    "forms": [
        {
            "href": "/actions/receiveMeasure",
            "contentType": "application/json",
            "op": "invokeaction"
        }
    ]
},
{
    "events": {
        "newObservation": {
            "description": "A new observation of the specified datastream has been
added to the system",
            "data": {
                "type": "object",
                "readOnly": true,
                "writeOnly": false,
                "properties": {
                    "device_id": {
                        "type": "string",
                        "readOnly": true,
                        "writeOnly": false,
                        "description": "The device's ID according to the list (D001,
D002, I001, ...)"
                    },
                    "observed_property": {
                        "type": "string",
                        "readOnly": true,
                        "writeOnly": false,
                        "description": "The device's observed property, such as
soilTemperature, soilMoisture,..."
                    },
                    "value": {
                        "type": "number",
                        "readOnly": true,
                        "writeOnly": false,
                        "description": "The value of the measurement"
                    },
                    "time": {
                        "type": "string",
                        "readOnly": true,
                        "writeOnly": false,
                        "description": "The time when this measurement was taken"
                    }
                }
            },
            "forms": [
                {
                    "href": "/events/newObservation",
                    "contentType": "application/json",
                    "op": "subscribeevent"
                }
            ]
        }
    }
}

```

## Weather Station Base Field Description

```
{
  "id": "acquaount:demostation",
  "@context": "https://www.w3.org/2022/wot/td/v1.1",
  "title": "AcquaountStationBase",
  "stationName": "Base",
  "href": "84.88.76.14",
  "securityDefinitions": {
    "basic_sc": {
      "scheme": "nosec"
    }
  },
  "security": [
    "basic_sc"
  ],
  "description": "Base Station Thing Description",
  "properties": {
    "stationInformation": {
      "type": "object",
      "description": "The information of the station this thing represents",
      "readOnly": true,
      "writeOnly": false,
      "observable": false,
      "forms": [
        {
          "href": "/properties/thingInformation",
          "contentType": "application/json",
          "op": "readproperty"
        }
      ]
    },
    "name": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "The name of the field"
    },
    "description": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "A description about the field"
    },
    "pilot": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "The pilot of this field"
    },
    "location": {
      "type": "object",
      "readOnly": true,
      "writeOnly": false,
      "format": "geojson",
      "href": "https://datatracker.ietf.org/doc/html/rfc7946#section-3.1",
      "description": "The location of the field in GeoJSON Geometry format"
    }
  }
}
```

```

"propertiesList": {
  "type": "array",
  "description": "The list of observed properties of this thing",
  "readOnly": true,
  "writeOnly": false,
  "observable": false,
  "forms": [
    {
      "href": "/properties/propertiesList",
      "contentType": "application/json",
      "op": "readproperty"
    }
  ],
  "items": {
    "type": "object",
    "readOnly": true,
    "writeOnly": false,
    "properties": {
      "name": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The name of the property"
      },
      "description": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "A description of the property"
      }
    }
  }
},
"propertyInformation": {
  "type": "object",
  "description": "The information of a property of this thing",
  "readOnly": true,
  "writeOnly": false,
  "observable": false,
  "uriVariables": {
    "name": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "The name of the property"
    }
  },
  "forms": [
    {
      "href": "/properties/propertyInformation",
      "contentType": "application/json",
      "op": "readproperty"
    }
  ],
  "properties": {
    "name": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "The name of the property"
    },
    "description": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "A description of the property"
    }
  }
}

```

```

    }
  },
  "datastreamsList": {
    "type": "array",
    "description": "The list of datasources of this thing",
    "readOnly": true,
    "writeOnly": false,
    "observable": false,
    "forms": [
      {
        "href": "/properties/datasourcesList",
        "contentType": "application/json",
        "op": "readproperty"
      }
    ],
    "items": {
      "type": "object",
      "readOnly": true,
      "writeOnly": false,
      "properties": {
        "name": {
          "type": "string",
          "readOnly": true,
          "writeOnly": false,
          "description": "The name of the datasource"
        },
        "description": {
          "type": "string",
          "readOnly": true,
          "writeOnly": false,
          "description": "A description of the datasource"
        },
        "unit_of_measurement": {
          "type": "string",
          "readOnly": true,
          "writeOnly": false,
          "description": "The unit of measurement of the observations of
this datasource"
        }
      }
    }
  },
  "datastreamInformation": {
    "description": "The information of a datasource of this thing",
    "readOnly": true,
    "writeOnly": false,
    "observable": false,
    "uriVariables": {
      "name": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The name of the datasource"
      }
    },
    "forms": [
      {
        "href": "/properties/propertyInformation",
        "contentType": "application/json",
        "op": "readproperty"
      }
    ],
    "type": "object",
    "properties": {
      "name": {

```

```

        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The name of the datasource"
    },
    "description": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "A description of the datasource"
    },
    "unit_of_measurement": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The unit of measurement of the observations of this
datasource"
    }
},
"datastreamLastMeasure": {
    "description": "The last measure of the specified datastream",
    "readOnly": true,
    "writeOnly": false,
    "observable": false,
    "uriVariables": {
        "name": {
            "type": "string",
            "readOnly": true,
            "writeOnly": false,
            "description": "The name of the datastream"
        }
    },
    "forms": [
        {
            "href": "/properties/datasourceLastMeasure",
            "contentType": "application/json",
            "op": "readproperty"
        }
    ],
    "type": "object",
    "properties": {
        "device_id": {
            "type": "string",
            "readOnly": true,
            "writeOnly": false,
            "description": "The ID of the sensor that took this measure"
        },
        "property_name": {
            "type": "string",
            "readOnly": true,
            "writeOnly": false,
            "description": "The name of the property observed in this measure"
        },
        "datastream_name": {
            "type": "string",
            "readOnly": true,
            "writeOnly": false,
            "description": "The name of the datastream containing this measure"
        },
        "unit_of_measurement": {
            "type": "string",
            "readOnly": true,
            "writeOnly": false,
            "description": "The unit of measurement of the observation"
        }
    },

```

```

    "value": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "The value of the observation"
    },
    "time_of_measure": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "The time when the observation was taken"
    }
  },
  "datastreamMeasures": {
    "description": "All the measures of a specified datastream",
    "readOnly": true,
    "writeOnly": false,
    "observable": false,
    "uriVariables": {
      "name": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The name of the datasource"
      }
    }
  },
  "forms": [
    {
      "href": "/properties/datasourceMeasures",
      "contentType": "application/json",
      "op": "readproperty"
    }
  ],
  "type": "array",
  "items": {
    "readOnly": true,
    "writeOnly": false,
    "properties": {
      "device_id": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The ID of the sensor that took this measure"
      },
      "property_name": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The name of the property observed in this
measure"
      },
      "datastream_name": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The name of the datastream containing this
measure"
      },
      "unit_of_measurement": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The unit of measurement of the observation"
      }
    },
    "value": {

```

```

        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The value of the observation"
    },
    "time_of_measure": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The time when the observation was taken"
    }
}

},
"datastreamTimeRangeMeasures": {
    "description": "All the measures in a range of time of a specified
datastream",
    "readOnly": true,
    "writeOnly": false,
    "observable": false,
    "uriVariables": {
        "name": {
            "type": "string",
            "readOnly": true,
            "writeOnly": false,
            "description": "The name of the datasource"
        },
        "start_time": {
            "type": "string",
            "readOnly": true,
            "writeOnly": false,
            "description": "The start time of the range of time, in format
\\\"YYYY-MM-DDThh:mm:ssZ\\\" in the UTC timezone"
        },
        "end_time": {
            "type": "string",
            "readOnly": true,
            "writeOnly": false,
            "description": "The start time of the range of time, in format
\\\"YYYY-MM-DDThh:mm:ssZ\\\" in the UTC timezone"
        }
    },
    "forms": [
        {
            "href": "/properties/datasourceMeasures",
            "contentType": "application/json",
            "op": "readproperty"
        }
    ],
    "type": "array",
    "items": {
        "readOnly": true,
        "writeOnly": false,
        "properties": {
            "device_id": {
                "type": "string",
                "readOnly": true,
                "writeOnly": false,
                "description": "The ID of the sensor that took this measure"
            },
            "property_name": {
                "type": "string",
                "readOnly": true,
                "writeOnly": false,
                "description": "The name of the property observed in this
measure"
            }
        }
    }
}

```

```

    },
    "datastream_name": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "The name of the datastream containing this
measure"
    },
    "unit_of_measurement": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "The unit of measurement of the observation"
    },
    "value": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "The value of the observation"
    },
    "time_of_measure": {
      "type": "string",
      "readOnly": true,
      "writeOnly": false,
      "description": "The time when the observation was taken"
    }
  }
},
"lastMeasures": {
  "description": "All the latest measures of the thing",
  "readOnly": true,
  "writeOnly": false,
  "observable": false,
  "forms": [
    {
      "href": "/properties/lastMeasures",
      "contentType": "application/json",
      "op": "readproperty"
    }
  ],
  "type": "array",
  "items": {
    "readOnly": true,
    "writeOnly": false,
    "properties": {
      "device_id": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The ID of the sensor that took this measure"
      },
      "property_name": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The name of the property observed in this
measure"
      },
      "datastream_name": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false,
        "description": "The name of the datastream containing this
measure"
      }
    }
  },

```

```

        "unit_of_measurement": {
            "type": "string",
            "readOnly": true,
            "writeOnly": false,
            "description": "The unit of measurement of the observation"
        },
        "value": {
            "type": "string",
            "readOnly": true,
            "writeOnly": false,
            "description": "The value of the observation"
        },
        "time_of_measure": {
            "type": "string",
            "readOnly": true,
            "writeOnly": false,
            "description": "The time when the observation was taken"
        }
    },
    },
    },
    "actions": {
        "receiveMeasure": {
            "safe": true,
            "idempotent": false,
            "input": {
                "type": "object",
                "description": "The message that the data retriever sends",
                "readOnly": true,
                "writeOnly": false,
                "properties": {
                    "info": {
                        "type": "object",
                        "readOnly": true,
                        "writeOnly": false,
                        "description": "The necessary information to identify from
which platform is this data coming from",
                        "properties": {
                            "stationID": {
                                "type": "string",
                                "readOnly": true,
                                "writeOnly": false,
                                "description": "The device's ID according to the list
(D001, D002, I001, ...) "
                            }
                        }
                    }
                },
                "values": {
                    "type": "object",
                    "readOnly": true,
                    "writeOnly": false,
                    "description": "The observation values in key/value pairs. The
keys must match the keys agreed on before.",
                    "properties": {}
                }
            },
            "output": {
                "description": "Returns true/false if measures were identified
correctly, and a message",
                "type": "object",
                "properties": {
                    "result": {
                        "type": "boolean",
                        "readOnly": true,

```

```

        "writeOnly": false
      },
      "message": {
        "type": "string",
        "readOnly": true,
        "writeOnly": false
      }
    },
    "readOnly": true,
    "writeOnly": false
  },
  "forms": [
    {
      "href": "/actions/receiveMeasure",
      "contentType": "application/json",
      "op": "invokeaction"
    }
  ]
},
"events": {
  "newObservation": {
    "description": "A new observation of the specified datastream has been
added to the system",
    "data": {
      "type": "object",
      "readOnly": true,
      "writeOnly": false,
      "properties": {
        "device_id": {
          "type": "string",
          "readOnly": true,
          "writeOnly": false,
          "description": "The device's ID according to the list (D001,
D002, I001, ...)"
        },
        "observed_property": {
          "type": "string",
          "readOnly": true,
          "writeOnly": false,
          "description": "The device's observed property, such as
soilTemperature, soilMoisture,..."
        },
        "value": {
          "type": "number",
          "readOnly": true,
          "writeOnly": false,
          "description": "The value of the measurement"
        },
        "time": {
          "type": "string",
          "readOnly": true,
          "writeOnly": false,
          "description": "The time when this measurement was taken"
        }
      }
    }
  },
  "forms": [
    {
      "href": "/events/newObservation",
      "contentType": "application/json",
      "op": "subscribeevent"
    }
  ]
}

```

}  
}