



# WATEREYE

Project Title:	O&M tools integrating accurate structural health in offshore energy
Starting date:	2019/11/01
Duration in months:	36
Call (part) identifier:	H2020-LC-SC3-2019-RES-TwoStages
Grant agreement no:	851207

## Consortium Members

Organisation	Abbreviation	Country
Centro Tecnológico Ceit-IK4	CEIT	Spain
Delft Dynamics B.V.	DD	The Netherlands
Semantic Web Company	SWC	Austria
Delft University of Technology	TUD	The Netherlands
SINTEF Energy Research AS	SINTEF-E	Norway
SINTEF AS	SINTEF-I	Norway
FLANDERS MAKE VZW	FMAKE	Belgium
COBRA Instalaciones y Servicios	COBRA	Spain
Plataforma Oceánica de Canarias	PLOCAN	Spain

## Deliverable D3.3

### Services for data acquisition, storage and access

Due date of deliverable	Month 18
Actual submission date	2021-04-30
Organization name of lead contractor for this deliverable	SWC
Dissemination level	PU
Revision	1.0

**The WATEREYE project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 851207**

**Authors**

<b>Author(s)</b>	<b>Partner 1 (SWC)</b> Artem Revenko
<b>Contributor(s)</b>	<b>Partner 7 (FMAKE)</b> Agusmian Partogi Ompusunggu
	<b>Partner 1 (CEIT)</b> Ainhoa Cortés
	<b>Partner 10 (PLOCAN)</b> Adriana García

**Disclaimer**

The content reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

<b>DRAFT PREPARATION</b>			
<b>Version</b>	<b>Publication date</b>	<b>Content</b>	<b>Author</b>
0.0	16/04/2021	<ul style="list-style-type: none"> <li>▪ First Draft</li> </ul>	<ul style="list-style-type: none"> <li>▪ Artem Revenko (SWC)</li> </ul>
0.1	16/04/2021	<ul style="list-style-type: none"> <li>▪ First Draft, definition of sections, their content and the responsible partner</li> </ul>	<ul style="list-style-type: none"> <li>▪ Artem Revenko (SWC)</li> </ul>
0.2	20/04/2021	<ul style="list-style-type: none"> <li>▪ Corrections from FMAKE, CEIT and PLOCAN</li> </ul>	<ul style="list-style-type: none"> <li>▪ Agusmian Partogi Ompusunggu (FMAKE)</li> <li>▪ Ainhoa Cortés (CEIT)</li> <li>▪ Adriana García (PLOCAN)</li> </ul>
1.0	29/04/21	<ul style="list-style-type: none"> <li>▪ Document finalized by SWC and CEIT</li> </ul>	<ul style="list-style-type: none"> <li>▪ Artem Revenko (SWC)</li> <li>▪ Ainhoa Cortés (CEIT)</li> </ul>

<b>HISTORY OF CHANGES</b>		
<b>Version</b>	<b>Publication date</b>	<b>Change</b>
0.0	16/04/2021	<ul style="list-style-type: none"> <li>▪ Document created and structure defined</li> </ul>
1.0	29/04/2021	<ul style="list-style-type: none"> <li>▪ Internal review of the document</li> </ul>

## **EXECUTIVE SUMMARY**

---

The goal of this document is to describe the services developed by SWC for data management in the WATEREYE project. In Section 1, we describe the preliminary investigation of potential solutions. In Section 2, we describe the early stage work where we evaluated different approaches for implementing these services. After settling on a suitable option, we defined the full data schema described in Section 3. Finally, Section 4 describes the current implemented and deployed solution and outlines next steps. The screenshots of the deployed applications as well as several specifications are presented in the Appendix.

# TABLE OF CONTENTS

---

<b>EXECUTIVE SUMMARY .....</b>	<b>4</b>
<b>TABLE OF CONTENTS .....</b>	<b>5</b>
<b>ABBREVIATIONS AND ACRONYMS .....</b>	<b>6</b>
<b>1 PRELIMINARY INVESTIGATIONS .....</b>	<b>7</b>
1.1 COMMUNICATION FORMAT .....	7
1.2 TRIPLE STORE .....	7
1.3 SQL DATABASE .....	8
<b>2 FULL DATA SCHEMA.....</b>	<b>9</b>
<b>3 DEPLOYED SOLUTION .....</b>	<b>10</b>
3.1 DATABASE.....	10
3.2 WEB APPLICATION.....	10
3.3 NEXT STEPS.....	10
<b>4 CONCLUSIONS.....</b>	<b>12</b>
<b>5 APPENDIX .....</b>	<b>13</b>
5.1 ONTOTEXT PLATFORM .....	13
5.2 GRAPHDB DATABASE .....	13
5.3 WATEREYE RDF DATA CUBES DATA STRUCTURE DEFINITIONS .....	13
5.4 RDF DATA CUBES COMPONENTS DEFINITIONS.....	19
5.5 RDF DATA CUBES SOML .....	23
5.6 GRAPHQL YAML DESCRIPTION .....	33
5.7 DATA MANAGEMENT WEB APPLICATION API REFERENCE .....	37
<b>6 REFERENCES.....</b>	<b>38</b>

## ABBREVIATIONS AND ACRONYMS

---

Abbreviation / Acronyms	Description
CBM	Condition-Based Maintenance
CM	Condition Monitoring
DSS	Decision Support Systems
EPC	Engineering, Procurement and Construction
FEM	Finite Element Method
IPS	Indoor Positioning System
LCoE	Levelized Cost of Energy
NLOS	Non-Line-Of-Sight
OPEX	Operational Expenditures
O&M	Operation and Maintenance
PdM	Predictive Maintenance
RUL	Remaining Useful Lifetime
SHM	Structural Health Monitoring
UAV	Unmanned Aerial Vehicle
US	Ultrasound
UWB	Ultra-Wide Band
WT	Wind Turbine
WF	Wind Farm
WFO	Wind Farm Operator
JSON	JavaScript Object Notation
HTTP	Hypertext Transfer Protocol
RDF	Resource Description Framework
SOML	Semantic Objects Modeling Language
YAML	YAML Ain't Markup Language
SQL	Structured Query Language

# 1 PRELIMINARY INVESTIGATIONS

---

## 1.1 Communication Format

The specifications of data exchange were investigated and published in Deliverable D1.5 “DATA, MODELS, DIAGNOSIS, PROGNOSIS, DECISION SUPPORT AND FAULT TOLERANT CONTROL SPECIFICATIONS”. In particular, the different data flows including the data types are presented in Section 8.

In the first stages of preparations, we have investigated different options of communication between different tools. As outlined in Deliverable 1.3 “SMART CORROSION SENSORS & COMMUNICATION SPECIFICATIONS” the different UWB sensors are capable of formatting the data and sending it over the HTTP. In the consortium meeting, the JSON<sup>1</sup> format was identified as the most suitable and accommodated by most partners. We note that for the sake of publishing data and facilitating its findability and reuse, it is beneficial to provide data in RDF format. However, the increasingly popular JSON-LD<sup>2</sup> specification allows to easily convert any JSON data into RDF.

To enable further flexibility in consuming the data by downstream applications, namely, 3D visualization SW tool and diagnostics & prognostics tool, we have further identified GraphQL<sup>3</sup> specification as a valuable extension. This query language enables the data consumer to specify the exact format and shape of the output data, therefore, making a single data source highly customizable and suitable for several different consumers.

## 1.2 Triple Store

As the first option, we have considered a NoSQL triple database. Such databases store the information in native RDF format. We have identified a (commercial) triple store that is capable of serving the GraphQL interface on top of the RDF data - GraphDB<sup>4</sup>. Moreover, this database is enterprise-ready, providing different scaling and deployment options. Unfortunately, the GraphQL feature is not available in the free version, therefore we contacted OntoText, and they agreed to provide a license of the commercial version of the platform. The GraphDB was deployed at a dedicated server prepared by SWC (*WATEREYE server*). See Appendix 6.1 and 6.2 for screenshots.

Following best RDF practices, we have identified RDF data cubes vocabulary<sup>5</sup> as the most suitable representation of the data in the WATEREYE project. Furthermore, we have defined Data Structure Definitions as required for using the RDF data cubes vocabulary. Most important “MeasureProperty”s are the measured thickness of the wall of a wind turbine structure, the quality of the ultrasound measurement and the temperature. The “DimensionProperty”s include the identifier of a sensor that sends the measurement, which further links to its location on the selected turbine, and the date and time of the measurement. See the full specification in Appendix 6.3 and 6.4.

The support of GraphQL in GraphDB is provided via SOML<sup>6</sup> - proprietary format for defining mappings from RDF to GraphQL. We have defined the SOML mappings for RDF data cubes and included this into the running instance of GraphDB. The GraphQL support feature is quite new for triple stores in general. Therefore, no full specification of capabilities is available and

some extensions of the features are still being implemented, see, for example, the issues at the official repository<sup>7</sup>. See full SOML in Appendix 6.5.

After some experimentation with the deployed solution, it was possible to push the Data Structure Definitions into GraphDB. However, these definitions are not propagated into SOML. Therefore, no mappings are provided for the different properties of the dataset. As a result, it is not possible to push the data through the GraphQL interface. The workaround would suggest defining a new SOML mappings specification and using it to create another GraphQL endpoint. This endpoint is used for pushing the data. However, such an approach diminishes the original benefits of using the out of the box solution.

### 1.3 SQL database

We have identified an alternative solution with more standardized SQL databases. The main idea consists of preparing GraphQL mappings on top of relational tables. This solution has certain advantages compared to using proprietary platform:

- Many enterprise-ready open source solutions are publicly available,
- SQL database have longer development history and are therefore known to a broader audience - this could an advantage to further maintain and support the prepared solution,
- For the same reasons they offer are more complete support for GraphQL,
- Broader customization options.

The disadvantage of this approach is the necessity of developing a custom application to serve a web interface.

## 2 FULL DATA SCHEMA

---

As already mentioned, the data flows and the data types were identified and published in earlier deliverables. However, the presented data flows only take into account a single measurement position. Hence, it was necessary to extend the data flows to accommodate:

1. Different positions of measurements at a wind turbine,
2. Different (fixed) sensors at those positions and a special mobile sensor that can potentially take measurements at any turbine and at any position,
3. Different wind turbines.

The final data schema includes 4 main types of objects: turbines, sensors, positions and measurements. See the Data Management Web Application API reference in Appendix 6.7 for the complete description of the defined data schema.

## 3 DEPLOYED SOLUTION

---

### 3.1 Database

There exists a range of different SQL implementations openly available. The lightest one is SQLite database<sup>8</sup>. SQLite is a relational database management system contained in a C library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program. We use SQLite implementations for local development. However, it is not suitable for a productive server.

The PostgreSQL database is a free and open-source relational database management system emphasizing extensibility and SQL compliance. It requires a separate server-side application. However, it is fully production ready and is used in many commercial software applications. We use this database for productive deployment. It is currently deployed at WATEREYE server.

### 3.2 Web Application

We make use of modern Python<sup>9</sup> frameworks:

- FastAPI<sup>10</sup> - for developing the web application itself,
- graphene<sup>11</sup> - for serving the GraphQL endpoint.

The developed application offers a set of endpoints for pushing, fetching and deleting the data. All the endpoints are described using OpenAPI<sup>12</sup> specification and offer SwaggerUI<sup>13</sup> frontend application that allows developers of downstream applications to conveniently test the input and the output of the different endpoints in the browser.

The full definitions of GraphQL endpoints and the different data management endpoints are available in Appendix 6.6.

At the current stage the downstream applications are already deployed and connected to the data management web application, though these downstream applications are still being actively developed.

### 3.3 Next Steps

Currently, the data management web application provides endpoints to store and retrieve data in JSON format. This is sufficient for the downstream applications in the WATEREYE project. However, as already mentioned, for publishing the open data it would be beneficial to format the data as linked data and comply with FAIR principles<sup>14</sup>. Therefore, in the next steps we will discuss the options to publish open research data and then initiate a discussion with potential consumers about the most convenient RDF vocabulary for such data. Though we have already identified and prepared the RDF data cube Data Structure Definitions, it is still not clear if this format would be convenient for the consumers.

As mentioned in the potential disadvantages of developing a custom application, the developers are responsible for taking care of the additional non-functional requirements, such as user management and authentication. The current deployed solution offers open access. In the further stages of the project, it might be necessary to extend the application to manage individual users, providing for them a way to authenticate.



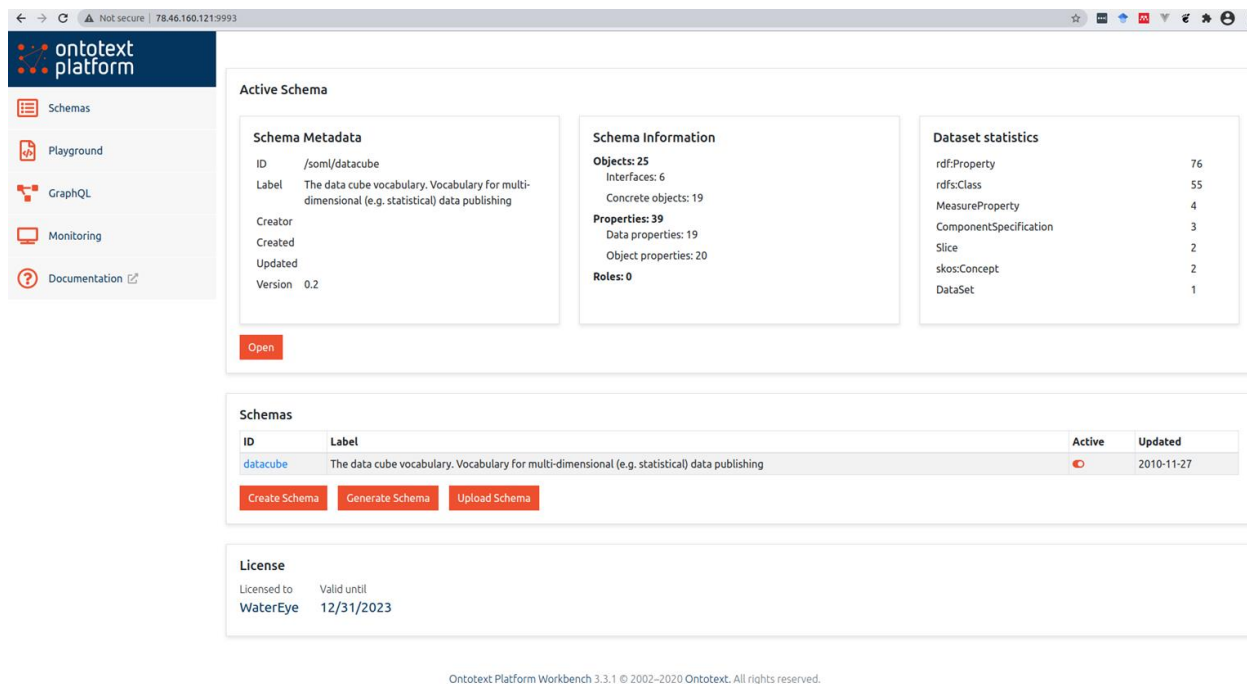
## **4 CONCLUSIONS**

---

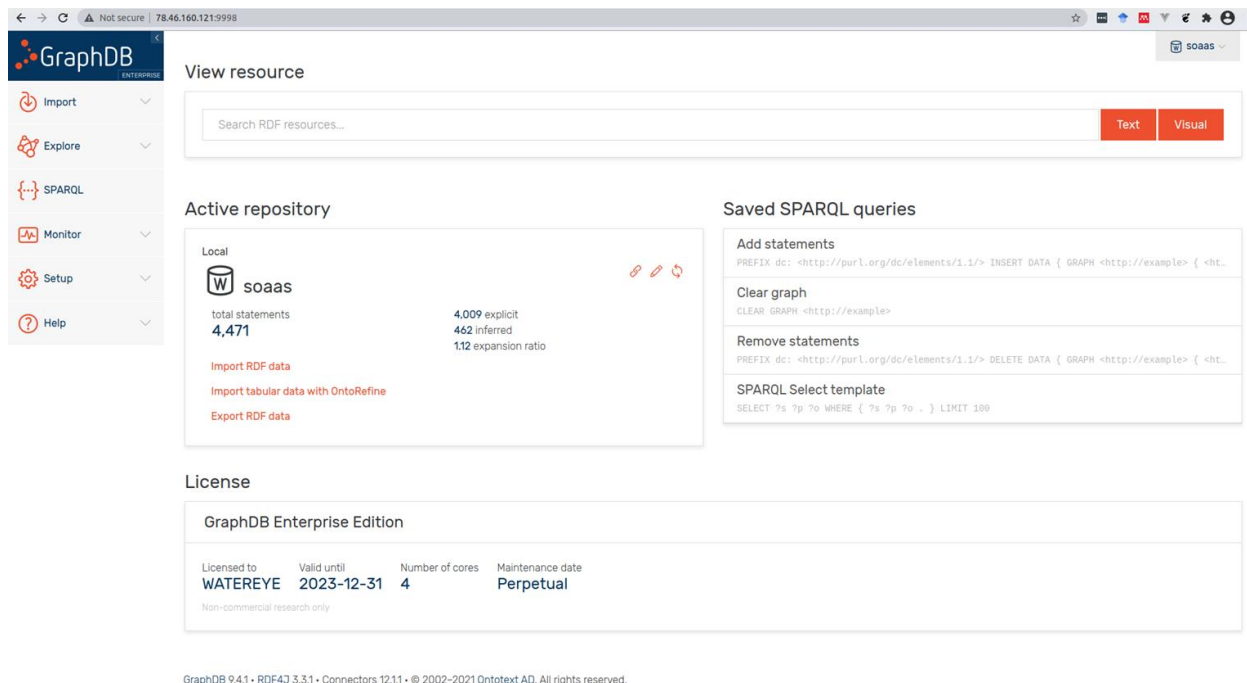
We have analyzed different approaches for providing data acquisition, storage and access solutions. The deployed solution is a custom development for the WATEREYE project. The solution offers 1) a set of endpoints for data management and a SwaggerUI interface for convenient development of downstream applications, 2) GraphQL interface for defining the shape and format of the output in the request. The solution relies on open source production-ready PostgreSQL as the database backend.

# 5 APPENDIX

## 5.1 OntoText Platform



## 5.2 GraphDB database



## 5.3 WATEREYE RDF data cubes Data Structure Definitions

# @prefix owl: <http://www.w3.org/2002/07/owl#> .

# @prefix void: <http://rdfs.org/ns/void#> .

---

```
@prefix dcterms: <http://purl.org/dc/terms/> .
# @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
# @prefix dcat: <http://www.w3.org/ns/dcat#> .
# @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
# @prefix qb: <http://purl.org/linked-data/cube#> .
# @prefix skos: <http://www.w3.org/2004/02/skos/core#> .
# @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

# -----WATEREYE namespace -----
@prefix weye-res:      <https://watereye-project.eu/res#> .
@prefix weye-def:     <https://watereye-project.eu/def#> .

# -----SDMX namespace -----
@prefix sdmx-dimension: <http://purl.org/linked-data/sdmx/2009/dimension#> .
@prefix sdmx-attribute: <http://purl.org/linked-data/sdmx/2009/attribute#> .
@prefix sdmx-concept:  <http://purl.org/linked-data/sdmx/2009/concept#> .
@prefix sdmx-measure:  <http://purl.org/linked-data/sdmx/2009/measure#> .

# -----Generic namespaces -----
@prefix qb:      <http://purl.org/linked-data/cube#> .
@prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:   <http://www.w3.org/2000/01/rdf-schema#> .
@prefix skos:   <http://www.w3.org/2004/02/skos/core#> .
@prefix xsd:    <http://www.w3.org/2001/XMLSchema#> .

# DSD

weye-def: a qb:DataStructureDefinition ;
  rdfs:label "Data structure definition for WATEREYE data"@en ;
  qb:component
```

```
[ qb:dimension weye-def:device_id ;
  qb:componentAttachment qb:Slice ],
[ qb:dimension weye-def:sample_id ;
  qb:componentAttachment qb:Slice ],
[ qb:dimension weye-def:piezo_id ;
  qb:componentAttachment qb:Slice ],
[ qb:dimension weye-def:date_time ;
  qb:componentAttachment qb:Slice ],
[ qb:dimension weye-def:xctype ;
  qb:componentAttachment qb:Slice ],
```

```
[ qb:measure weye-def:echo_n ] ,
[ qb:measure weye-def:amplitude ] ,
[ qb:measure weye-def:xcmax ] ,
[ qb:measure weye-def:pow ] ,
[ qb:measure weye-def:width ] ,
[ qb:measure weye-def:tof ] ,
[ qb:measure weye-def:temp ] ,
[ qb:measure weye-def:sizew_sigma ] ,
[ qb:measure weye-def:gain ] ,
[ qb:measure weye-def:decay_rate ] ,
[ qb:measure weye-def:freq ] ,
[ qb:measure weye-def:dr_sigma ] ;
```

```
qb:sliceKey weye-def:slice-by-deviceId, weye-def:slice-by-sampleId, weye-def:slice-by-piezoId.
```

```
# Slice key definition
```

```
weye-def:slice-by-deviceId a qb:SliceKey ;
  rdfs:label "slice by device Id"@en ;
  qb:componentProperty weye-def:device_id .
```

```
weye-def:slice-by-sampleId a qb:SliceKey ;
    rdfs:label "slice by sample Id"@en ;
    qb:componentProperty weye-def:sample_id .
```

```
weye-def:slice-by-piezoId a qb:SliceKey ;
    rdfs:label "slice by piezo Id"@en ;
    qb:componentProperty weye-def:piezo_id .
```

#### # Dimension definitions

```
weye-def:date_time a qb:DimensionProperty, rdf:Property ;
    dcterms:description "DateTime stamp";
    rdfs:label "date_time";
    rdfs:subPropertyOf sdmx-dimension:timePeriod ;
    rdfs:range xsd:dateTime .
```

```
weye-def:device_id a qb:DimensionProperty, rdf:Property ;
    dcterms:description "The ID of the device" ;
    rdfs:label "device_id" ;
    rdfs:range xsd:string .
```

```
weye-def:sample_id a qb:DimensionProperty, rdf:Property;
    dcterms:description "The ID of the sample";
    rdfs:label "sample_id";
    rdfs:range xsd:string .
```

```
weye-def:piezo_id a qb:DimensionProperty, rdf:Property;
    dcterms:description "The ID of the piezo device";
    rdfs:label "piezo_id";
    rdfs:range xsd:string .
```

```
weye-def:xctype a qb:DimensionProperty, rdf:Property;
```

```
dcterms:description "XC type";  
rdfs:label "xctype";  
rdfs:range xsd:string .
```

```
weye-def:echo_n a qb:MeasureProperty, rdf:Property;  
dcterms:description "Echo number";  
rdfs:label "echo_n";  
rdfs:range xsd:integer ;  
rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue> .
```

```
weye-def:amplitude a qb:MeasureProperty, rdf:Property;  
rdfs:label "amplitude";  
rdfs:range xsd:integer ;  
rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue> .
```

```
weye-def:xcmax a qb:MeasureProperty, rdf:Property;  
rdfs:label "xcmax";  
rdfs:range xsd:integer ;  
rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue> .
```

```
weye-def:pow a qb:MeasureProperty, rdf:Property ;  
rdfs:label "pow";  
rdfs:range xsd:integer ;  
rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue> .
```

```
weye-def:width a qb:MeasureProperty, rdf:Property;  
rdfs:label "width";  
rdfs:range xsd:integer ;  
rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue> .
```

```
weye-def:tof a qb:MeasureProperty, rdf:Property;  
dcterms:description "Time of flight";  
rdfs:label "tof";
```

rdfs:range xsd:decimal ;  
rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue> .

weye-def:temp a qb:MeasureProperty, rdf:Property;  
dcterms:description "Temperature";  
rdfs:label "temp";  
rdfs:range xsd:decimal ;  
rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue> .

weye-def:sizew\_sigma a qb:MeasureProperty, rdf:Property;  
rdfs:label "sizew\_sigma";  
rdfs:range xsd:decimal ;  
rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue> .

weye-def:gain a qb:MeasureProperty, rdf:Property;  
rdfs:label "gain";  
rdfs:range xsd:decimal ;  
rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue> .

weye-def:decay\_rate a qb:MeasureProperty, rdf:Property;  
rdfs:label "decay\_rate";  
rdfs:range xsd:decimal ;  
rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue> .

weye-def:freq a qb:MeasureProperty, rdf:Property;  
rdfs:label "freq";  
rdfs:range xsd:decimal ;  
rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue> .

weye-def:dr\_sigma a qb:MeasureProperty, rdf:Property;  
rdfs:label "dr\_sigma";  
rdfs:range xsd:decimal ;  
rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue> .

## 5.4 RDF data cubes components definitions

@prefix owl: <http://www.w3.org/2002/07/owl#> .

@prefix void: <http://rdfs.org/ns/void#> .

@prefix dcterms: <http://purl.org/dc/terms/> .

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix dcat: <http://www.w3.org/ns/dcat#> .

@prefix sdmx-dimension: <http://purl.org/linked-data/sdmx/2009/dimension#> .

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

@prefix sdmx-attribute: <http://purl.org/linked-data/sdmx/2009/attribute#> .

@prefix qb: <http://purl.org/linked-data/cube#> .

@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

@prefix sdmx-concept: <http://purl.org/linked-data/sdmx/2009/concept#> .

<https://watereye-project.eu/def/ontology/components> a owl:Ontology;

dcterms:title "Components Ontology"@en;

rdfs:label "Components Ontology"@en .

<https://watereye-project.eu/def/dimension/date\_time> a qb:DimensionProperty, rdf:Property;

dcterms:description "DateTime stamp";

rdfs:isDefinedBy <https://watereye-project.eu/def/ontology/components>;

rdfs:label "date\_time";

rdfs:range <https://watereye-project.eu/def/Date\_time>;

skos:notation "date\_time" .

<https://watereye-project.eu/def/dimension/device\_id> a qb:DimensionProperty, rdf:Property;

dcterms:description "The ID of the device";

rdfs:isDefinedBy <https://watereye-project.eu/def/ontology/components>;

rdfs:label "device\_id";

rdfs:range <https://watereye-project.eu/def/Device\_id>;

skos:notation "device\_id" .

---

```
<https://watereye-project.eu/def/dimension/sample_id> a qb:DimensionProperty, rdf:Property;
  dcterms:description "The ID of the sample";
  rdfs:isDefinedBy <https://watereye-project.eu/def/ontology/components>;
  rdfs:label "sample_id";
  rdfs:range <https://watereye-project.eu/def/Sample_id>;
  skos:notation "sample_id" .
```

```
<https://watereye-project.eu/def/dimension/piezo_id> a qb:DimensionProperty, rdf:Property;
  dcterms:description "The ID of the piezo device";
  rdfs:isDefinedBy <https://watereye-project.eu/def/ontology/components>;
  rdfs:label "piezo_id";
  rdfs:range <https://watereye-project.eu/def/Piezo_id>;
  skos:notation "piezo_id" .
```

```
<https://watereye-project.eu/def/dimension/xctype> a qb:DimensionProperty, rdf:Property;
  dcterms:description "XC type";
  rdfs:isDefinedBy <https://watereye-project.eu/def/ontology/components>;
  rdfs:label "xctype";
  rdfs:range <https://watereye-project.eu/def/Xctype>;
  skos:notation "xctype" .
```

```
<https://watereye-project.eu/def/measure/echo_n> a qb:MeasureProperty, rdf:Property;
  dcterms:description "Echo number";
  rdfs:isDefinedBy <https://watereye-project.eu/def/ontology/components>;
  rdfs:label "echo_n";
  rdfs:range <https://watereye-project.eu/def/Echo_n>;
  rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue>;
  skos:notation "echo_n" .
```

```
<https://watereye-project.eu/def/measure/tof> a qb:MeasureProperty, rdf:Property;
  dcterms:description "Time of flight";
  rdfs:isDefinedBy <https://watereye-project.eu/def/ontology/components>;
  rdfs:label "tof";
```

rdfs:range <https://watereye-project.eu/def/Tof>;  
rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue>;  
skos:notation "tof" .

<https://watereye-project.eu/def/measure/temp> a qb:MeasureProperty, rdf:Property;  
dcterms:description "Temperature";  
rdfs:isDefinedBy <https://watereye-project.eu/def/ontology/components>;  
rdfs:label "temp";  
rdfs:range <https://watereye-project.eu/def/Temp>;  
rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue>;  
skos:notation "temp" .

<https://watereye-project.eu/def/measure/sizew\_sigma> a qb:MeasureProperty, rdf:Property;  
rdfs:isDefinedBy <https://watereye-project.eu/def/ontology/components>;  
rdfs:label "sizew\_sigma";  
rdfs:range <https://watereye-project.eu/def/Sizew\_sigma>;  
rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue>;  
skos:notation "sizew\_sigma" .

<https://watereye-project.eu/def/measure/gain> a qb:MeasureProperty, rdf:Property;  
rdfs:isDefinedBy <https://watereye-project.eu/def/ontology/components>;  
rdfs:label "gain";  
rdfs:range <https://watereye-project.eu/def/Gain>;  
rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue>;  
skos:notation "gain" .

<https://watereye-project.eu/def/measure/amplitude> a qb:MeasureProperty, rdf:Property;  
rdfs:isDefinedBy <https://watereye-project.eu/def/ontology/components>;  
rdfs:label "amplitude";  
rdfs:range <https://watereye-project.eu/def/Amplitude>;  
rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue>;  
skos:notation "amplitude" .

---

```
<https://watereye-project.eu/def/measure/decay_rate> a qb:MeasureProperty, rdf:Property;
  rdfs:isDefinedBy <https://watereye-project.eu/def/ontology/components>;
  rdfs:label "decay_rate";
  rdfs:range <https://watereye-project.eu/def/Decay_rate>;
  rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue>;
  skos:notation "decay_rate" .
```

```
<https://watereye-project.eu/def/measure/xcmax> a qb:MeasureProperty, rdf:Property;
  rdfs:isDefinedBy <https://watereye-project.eu/def/ontology/components>;
  rdfs:label "xcmax";
  rdfs:range <https://watereye-project.eu/def/Xcmax>;
  rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue>;
  skos:notation "xcmax" .
```

```
<https://watereye-project.eu/def/measure/pow> a qb:MeasureProperty, rdf:Property;
  rdfs:isDefinedBy <https://watereye-project.eu/def/ontology/components>;
  rdfs:label "pow";
  rdfs:range <https://watereye-project.eu/def/Pow>;
  rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue>;
  skos:notation "pow" .
```

```
<https://watereye-project.eu/def/measure/width> a qb:MeasureProperty, rdf:Property;
  rdfs:isDefinedBy <https://watereye-project.eu/def/ontology/components>;
  rdfs:label "width";
  rdfs:range <https://watereye-project.eu/def/Width>;
  rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue>;
  skos:notation "width" .
```

```
<https://watereye-project.eu/def/measure/freq> a qb:MeasureProperty, rdf:Property;
  rdfs:isDefinedBy <https://watereye-project.eu/def/ontology/components>;
  rdfs:label "freq";
  rdfs:range <https://watereye-project.eu/def/Freq>;
  rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue>;
```

skos:notation "freq" .

```
<https://watereye-project.eu/def/measure/dr_sigma> a qb:MeasureProperty, rdf:Property;
  rdfs:isDefinedBy <https://watereye-project.eu/def/ontology/components>;
  rdfs:label "dr_sigma";
  rdfs:range <https://watereye-project.eu/def/Dr_sigma>;
  rdfs:subPropertyOf <http://purl.org/linked-data/sdmx/2009/measure#obsValue>;
  skos:notation "dr_sigma" .
```

## 5.5 RDF data cubes SOML

id: /soml/datacube

label: The data cube vocabulary. Vocabulary for multi-dimensional (e.g. statistical)  
data publishing

versionInfo: '0.2'

created: '2010-07-12'

updated: '2010-11-27'

prefixes:

dcterms: "http://purl.org/dc/terms/"

foaf: "http://xmlns.com/foaf/0.1/"

owl: http://www.w3.org/2002/07/owl#

qb: http://purl.org/linked-data/cube#

rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#

rdfs: "http://www.w3.org/2000/01/rdf-schema#"

scovo: http://purl.org/NET/scovo#

skos: http://www.w3.org/2004/02/skos/core#

void: http://rdfs.org/ns/void#

xsd: http://www.w3.org/2001/XMLSchema#

sdmx-dimension: http://purl.org/linked-data/sdmx/2009/dimension#

sdmx-attribute: http://purl.org/linked-data/sdmx/2009/attribute#

sdmx-concept: http://purl.org/linked-data/sdmx/2009/concept#

sdmx-measure: http://purl.org/linked-data/sdmx/2009/measure#

weye-res: https://watereye-project.eu/res#

weye-def: https://watereye-project.eu/def#

## specialPrefixes:

base\_iri: <https://watereye-project.eu/res#>

vocab\_prefix: qb

vocab\_iri: <http://purl.org/linked-data/cube#>

## properties:

## attribute:

descr: An alternative to qb:componentProperty which makes explicit that the component is a attribute

kind: object

label: attribute

max: inf

range: AttributeProperty

rdfProp: qb:attribute

## codeList:

descr: gives the code list associated with a CodedProperty

kind: object

label: code list

max: inf

range: HierarchicalCodeList

rdfProp: qb:codeList

## component:

descr: indicates a component specification which is included in the structure of the dataset

kind: object

label: component specification

max: inf

range: ComponentSpecification

rdfProp: qb:component

## componentAttachment:

descr: Indicates the level at which the component property should be attached, this might an qb:DataSet, qb:Slice or qb:Observation, or a qb:MeasureProperty

kind: object

label: component attachment

max: inf

range: rdfs:Class

rdfProp: qb:componentAttachment

componentProperty:

descr: indicates a ComponentProperty (i.e. attribute/dimension) expected on a DataSet, or a dimension fixed in a SliceKey

kind: object

label: component

max: inf

range: ComponentPropertyInterface

rdfProp: qb:componentProperty

componentRequired:

descr: |-

Indicates whether a component property is required (true) or optional (false) in the context of a DSD. Only applicable

to components correspond to an attribute. Defaults to false (optional)

kind: literal

label: component required

max: inf

range: boolean

rdfProp: qb:componentRequired

concept:

descr: gives the concept which is being measured or indicated by a ComponentProperty

kind: object

label: concept

max: inf

range: skos:Concept

rdfProp: qb:concept

dataSet:

descr: indicates the data set of which this observation is a part

kind: object

label: data set

max: inf

range: DataSet

rdfProp: qb:dataSet

dimension:

descr: An alternative to qb:componentProperty which makes explicit that the component is a dimension

kind: object

label: dimension

max: inf

range: DimensionProperty

rdfProp: qb:dimension

hierarchyRoot:

descr: Specifies a root of the hierarchy. A hierarchy may have multiple roots but must have at least one

kind: object

max: inf

range: iri

rdfProp: qb:hierarchyRoot

measure:

descr: An alternative to qb:componentProperty which makes explicit that the component is a measure

kind: object

label: measure

max: inf

range: MeasureProperty

rdfProp: qb:measure

measureDimension:

descr: An alternative to qb:componentProperty which makes explicit that the component is a measure dimension

kind: object

label: measure dimension

max: inf

range: DimensionProperty

rdfProp: qb:measureDimension

measureType:

descr: Generic measure dimension, the value of this dimension indicates which measure (from the set of measures in the DSD) is being given by the obsValue (or other primary measure)

kind: object

label: measure type

max: inf

range: MeasureProperty

rdfProp: qb:measureType

observation:

descr: indicates a observation contained within this slice of the data set

kind: object

label: observation

max: inf

range: Observation

rdfProp: qb:observation

observationGroup:

descr: Indicates a group of observations. The domain of this property is left open so that a group may be attached to different resources and need not be restricted to a single DataSet

kind: object

label: observation group

max: inf

range: ObservationGroup

rdfProp: qb:observationGroup

order:

descr: indicates a priority order for the components of sets with this structure, used to guide presentations - lower order numbers come before higher numbers, un-numbered components come last

kind: literal

label: order

max: inf

range: int

rdfProp: qb:order

parentChildProperty:

descr: Specifies a property which relates a parent concept in the hierarchy to  
a child concept

kind: object

label: parent-child property

max: inf

range: rdf:PropertyInterface

rdfProp: qb:parentChildProperty

slice:

descr: Indicates a subset of a DataSet defined by fixing a subset of the dimensional  
values

kind: object

label: slice

max: inf

range: Slice

rdfProp: qb:slice

sliceKey:

descr: indicates a slice key which is used for slices in this dataset

kind: object

label: slice key

max: inf

range: SliceKey

rdfProp: qb:sliceKey

sliceStructure:

descr: indicates the sub-key corresponding to this slice

kind: object

label: slice structure

max: inf

range: SliceKey

rdfProp: qb:sliceStructure

structure:

descr: indicates the structure to which this data set conforms

kind: object

label: structure

max: inf

range: DataStructureDefinition

rdfProp: qb:structure

objects:

Attachable:

descr: Abstract superclass for everything that can have attributes and dimensions

inherits: AttachableInterface

label: Attachable (abstract)

type: qb:Attachable

AttachableInterface:

descr: Abstract superclass of Attachable

kind: abstract

AttributeProperty:

descr: The class of components which represent attributes of observations in the  
cube, e.g. unit of measurement

inherits: ComponentPropertyInterface

label: Attribute property

type: qb:AttributeProperty

CodedProperty:

descr: Superclass of all coded ComponentProperties

inherits: CodedPropertyInterface

label: Coded property

type: qb:CodedProperty

CodedPropertyInterface:

descr: Abstract superclass of CodedProperty

inherits: ComponentPropertyInterface

kind: abstract

props:

codeList: {}

**ComponentProperty:**

descr: Abstract super-property of all properties representing dimensions, attributes or measures

inherits: ComponentPropertyInterface

label: Component property (abstract)

type: qb:ComponentProperty

**ComponentPropertyInterface:**

descr: Abstract superclass of ComponentProperty

inherits: rdf:PropertyInterface

kind: abstract

props:

concept: {}

**ComponentSet:**

descr: Abstract class of things which reference one or more ComponentProperties

inherits: ComponentSetInterface

label: Component set

type: qb:ComponentSet

**ComponentSetInterface:**

descr: Abstract superclass of ComponentSet

kind: abstract

props:

componentProperty: {}

**ComponentSpecification:**

descr: Used to define properties of a component (attribute, dimension etc) which are specific to its usage in a DSD

inherits: ComponentSetInterface

label: Component specification

props:

componentAttachment: {}

componentRequired: {}

order: {}

type: qb:ComponentSpecification

**DataSet:**

descr: Represents a collection of observations, possibly organized into various slices, conforming to some common dimensional structure

inherits: AttachableInterface

label: Data set

props:

slice: {}

structure: {}

type: qb:DataSet

DataStructureDefinition:

descr: Defines the structure of a DataSet or slice

inherits: ComponentSetInterface

label: Data structure definition

props:

component: {}

sliceKey: {}

type: qb:DataStructureDefinition

DimensionProperty:

descr: The class of components which represent the dimensions of the cube

inherits: CodedPropertyInterface

label: Dimension property

type: qb:DimensionProperty

HierarchicalCodeList:

descr: Represents a generalized hierarchy of concepts which can be used for coding.

The hierarchy is defined by one or more roots together with a property which relates concepts in the hierarchy to their child concept. The same concepts may be members of multiple hierarchies provided that different qb:parentChildProperty values are used for each hierarchy

label: Hierarchical Code List

props:

hierarchyRoot: {}

parentChildProperty: {}

type: qb:HierarchicalCodeList

MeasureProperty:

descr: The class of components which represent the measured value of the phenomenon being observed

inherits: ComponentPropertyInterface

label: Measure property

type: qb:MeasureProperty

#### Observation:

descr: A single observation in the cube, may have one or more associated measured values

inherits: AttachableInterface

kind: abstract

label: Observation

props:

dataSet: {}

type: qb:Observation

#### ObservationGroup:

descr: A, possibly arbitrary, group of observations

label: Observation Group

props:

observation: {}

type: qb:ObservationGroup

#### Slice:

descr: Denotes a subset of a DataSet defined by fixing a subset of the dimensional values, component properties on the Slice

inherits: AttachableInterface

label: Slice

props:

sliceStructure: {}

type: qb:Slice

#### SliceKey:

descr: Denotes a subset of the component properties of a DataSet which are fixed in the corresponding slices

inherits: ComponentSetInterface

label: Slice key

```

    type: qb:SliceKey
  rdf:Property:
    inherits: rdf:PropertyInterface
    type: rdf:Property
  rdf:PropertyInterface:
    descr: Abstract superclass of rdf:Property
    kind: abstract
  rdfs:Class:
    type: rdfs:Class
  skos:Concept:
    type: skos:Concept
  weye-def:WEYEObservation:
    inherits: Observation
    props:
      deviceId: {range: weye-def:Device, min: 1}
      tof: {range: positiveFloat, typeCast: true, min: 1}
      dateTime: {range: dateTime, typeCast: true, min: 1}
  weye-def:Device:
    descr: Device

```

## 5.6 GraphQL YAML description

```

schema {
  query: GQLQuery
}

```

```

"""Exposes a URL that specifies the behaviour of this scalar."""

```

```

directive @specifiedBy(

```

```

  """The URL that specifies the behaviour of this scalar."""

```

```

  url: String!

```

```

) on SCALAR

```

```

"""

```

The `Date` scalar type represents a Date

value as specified by

[iso8601]([https://en.wikipedia.org/wiki/ISO\\_8601](https://en.wikipedia.org/wiki/ISO_8601)).

""

scalar Date

""

The `DateTime` scalar type represents a DateTime value as specified by

[iso8601]([https://en.wikipedia.org/wiki/ISO\\_8601](https://en.wikipedia.org/wiki/ISO_8601)).

""

scalar DateTime

type GQLDDSLocation {

  xn: Float!

  ye: Float!

  zd: Float!

  phi: Float!

  theta: Float!

  psi: Float!

}

type GQLItem {

  date: Date!

  time: Time!

  tofQuality: GQLTOFQuality!

  temp: Float!

  thickness: Float!

  raw: [Float]

  deviceId: String

  sampleId: String

  piezoId: String

  tof: Float

  sizeus: Int

  echoN: Int

```
gain: Float
decayRate: Float
xcmax: Int
pow: Int
width: Int
freq: Float
xctype: String
dateTime: DateTime!
id: String!
position: GQLPositionBase!
turbine: GQLTurbineBase!
}
```

```
type GQLPosition {
  waypointIdentifier: String!
  xn: Float!
  ye: Float!
  zd: Float!
  referenceThickness: Float
  hostTurbine: GQLTurbineBase!
  measurements: [GQLItem]
  fixedSensor: GQLSensorBase
}
```

```
type GQLPositionBase {
  waypointIdentifier: String!
  xn: Float!
  ye: Float!
  zd: Float!
  referenceThickness: Float
}
```

```
type GQLQuery {
```

```
measurements(skip: Int = null, limit: Int = null, turbineIdentifier: String = null, verbose: Boolean = true, positionId: String = null, ordering: Sorting = null, dateTimeStart: DateTime = null, dateTimeEnd: DateTime = null): [GQLItem]
```

```
turbines(skip: Int = null, limit: Int = null): [GQLTurbine]
```

```
sensors(turbineId: String = null, skip: Int = null, limit: Int = null): [GQLSensor]
```

```
positions(turbineId: String = null, skip: Int = null, limit: Int = null): [GQLPosition]
```

```
}
```

```
type GQLSensor {
```

```
  sensorIdentifier: String!
```

```
  hostTurbine: GQLTurbineBase!
```

```
  sensorPosition: GQLPositionBase!
```

```
}
```

```
type GQLSensorBase {
```

```
  sensorIdentifier: String!
```

```
}
```

```
type GQLTOFQuality {
```

```
  sizeSigma: Float!
```

```
  amplitude: Float!
```

```
  drSigma: Float!
```

```
}
```

```
type GQLTurbine {
```

```
  turbineIdentifier: String!
```

```
  turbineLocation: GQLTurbineLocation!
```

```
  ddsLocation: GQLDDSLocation!
```

```
  sensors: [GQLSensorBase]
```

```
  positions: [GQLPositionBase]
```

```
}
```

```
type GQLTurbineBase {
```

```
  turbineIdentifier: String!
```

```
turbineLocation: GQLTurbineLocation!  
ddsLocation: GQLDDSLocation!  
}
```

```
type GQLTurbineLocation {  
  xn: Float!  
  ye: Float!  
}
```

```
enum Sorting {  
  ""oldest first""  
  asc  
  
  ""latest first""  
  desc  
}
```

```
""
```

The `Time` scalar type represents a Time value as specified by [iso8601]([https://en.wikipedia.org/wiki/ISO\\_8601](https://en.wikipedia.org/wiki/ISO_8601)).

```
""
```

```
scalar Time
```

## 5.7 Data Management Web Application API reference

API Reference

# Items

API Version: 0.1.0

## Data Management Web Application

For managing turbines, positions, sensors and measurements from both mobile and fixed sensors.

[contact](#)

## Acknowledgement

This service is developed for the [WaterEYE](#) project.

# INDEX

<b>1. MEASUREMENTS</b>	<b>3</b>
1.1 GET /measurements/	3
1.2 DELETE /measurements/	5
1.3 POST /mobile_measurements/	6
1.4 POST /fixed_measurements/	8
<b>2. POSITIONS</b>	<b>11</b>
2.1 GET /positions/	11
2.2 POST /positions/	11
2.3 DELETE /positions/	13
<b>3. SENSORS</b>	<b>16</b>
3.1 GET /sensors/	16
3.2 POST /sensors/	17
3.3 DELETE /sensors/	18
<b>4. TURBINES</b>	<b>20</b>
4.1 GET /turbines/	20
4.2 POST /turbines/	21
4.3 DELETE /turbine/	22

# API

## 1. MEASUREMENTS

### 1.1 GET /measurements/

#### Read Measurements

#### REQUEST

##### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
skip	integer	
limit	integer	
turbine_identifier	string	
verbose	boolean	
position_id	string	
ordering	enum	ALLOWED: oldest first, latest first
date_time_start	date-time	
date_time_end	date-time	

#### RESPONSE

##### STATUS CODE - 200: Successful Response

##### RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
ARRAY OF OBJECT WITH BELOW STRUCTURE		
ANY:OF	object	
OPTION:1	object	
date*	string	
time*	string	
tof_quality*	object	
sizew_sigma*	number	
amplitude*	number	
dr_sigma*	number	
temp*	number	
thickness*	number	
raw	array	
device_id	string	
sample_id	string	
piezo_id	string	
tof	number	
sizeus	integer	

NAME	TYPE	DESCRIPTION
echo_n	integer	
gain	number	
decay_rate	number	
xcmx	integer	
pow	integer	
width	integer	
freq	number	
xctype	string	
date_time*	string	
id*	string	
<b>position*</b>	object	
waypoint_identifer*	string	
xn*	number	
ye*	number	
zd*	number	
reference_thickness	number	
<b>turbine*</b>	object	
turbine_identifer*	string	
<b>turbine_location*</b>	object	
xn*	number	
ye*	number	
<b>dds_location*</b>	object	
xn*	number	
ye*	number	
zd*	number	
phi*	number	
theta*	number	
psi*	number	
<b>OPTION:2</b>	object	
date*	string	
time*	string	
<b>tof_quality*</b>	object	
sizew_sigma*	number	
amplitude*	number	
dr_sigma*	number	
temp*	number	
thickness*	number	
<b>raw</b>	array	
device_id	string	
sample_id	string	
piezo_id	string	
tof	number	

NAME	TYPE	DESCRIPTION
sizeus	integer	
echo_n	integer	
gain	number	
decay_rate	number	
xcmx	integer	
pow	integer	
width	integer	
freq	number	
xctype	string	
position_id*	string	

STATUS CODE - 400: Bad request

STATUS CODE - 403: Access denied

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
detail	array	
loc*	array	
msg*	string	
type*	string	

STATUS CODE - 500: Internal server error

## 1.2 DELETE /measurements/

### Delete Measurement

#### REQUEST

QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*id_	string	

#### RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

STATUS CODE - 400: Bad request

STATUS CODE - 403: Access denied

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
------	------	-------------

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
detail	array	
loc*	array	
msg*	string	
type*	string	

STATUS CODE - 500: Internal server error

### 1.3 POST /mobile\_measurements/

#### Create Mobile Measurement

#### REQUEST

REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
date*	string	
time*	string	
tof_quality*	object	
sizew_sigma*	number	
amplitude*	number	
dr_sigma*	number	
temp*	number	
thickness*	number	
raw	array	
device_id	string	
sample_id	string	
piezo_id	string	
tof	number	
sizeus	integer	
echo_n	integer	
gain	number	
decay_rate	number	
xcmax	integer	
pow	integer	
width	integer	
freq	number	
xctype	string	
waypoint_identifier*	string	

#### RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
------	------	-------------

NAME	TYPE	DESCRIPTION
<b>OBJECT WITH BELOW STRUCTURE</b>		
date*	string	
time*	string	
<b>tof_quality*</b>	object	
sizew_sigma*	number	
amplitude*	number	
dr_sigma*	number	
temp*	number	
thickness*	number	
<b>raw</b>	array	
device_id	string	
sample_id	string	
piezo_id	string	
tof	number	
sizeus	integer	
echo_n	integer	
gain	number	
decay_rate	number	
xcmax	integer	
pow	integer	
width	integer	
freq	number	
xctype	string	
date_time*	string	
id*	string	
<b>position*</b>	object	
waypoint_identifier*	string	
xn*	number	
ye*	number	
zd*	number	
reference_thickness	number	
<b>turbine*</b>	object	
turbine_identifier*	string	
<b>turbine_location*</b>	object	
xn*	number	
ye*	number	
<b>dds_location*</b>	object	
xn*	number	
ye*	number	
zd*	number	
phi*	number	
theta*	number	

NAME	TYPE	DESCRIPTION
psi*	number	

STATUS CODE - 400: Bad request

STATUS CODE - 403: Access denied

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
detail	array	
loc*	array	
msg*	string	
type*	string	

STATUS CODE - 500: Internal server error

## 1.4 POST /fixed\_measurements/

### Create Fixed Measurement

#### REQUEST

REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
date*	string	
time*	string	
tof_quality*	object	
sizew_sigma*	number	
amplitude*	number	
dr_sigma*	number	
temp*	number	
thickness*	number	
raw	array	
device_id	string	
sample_id	string	
piezo_id	string	
tof	number	
sizeus	integer	
echo_n	integer	
gain	number	
decay_rate	number	
xcmax	integer	
pow	integer	
width	integer	
freq	number	

NAME	TYPE	DESCRIPTION
xctype	string	
sensor_identifier*	string	

## RESPONSE

STATUS CODE - 200: Successful Response

### RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
<b>OBJECT WITH BELOW STRUCTURE</b>		
date*	string	
time*	string	
<b>tof_quality*</b>	object	
sizew_sigma*	number	
amplitude*	number	
dr_sigma*	number	
temp*	number	
thickness*	number	
raw	array	
device_id	string	
sample_id	string	
piezo_id	string	
tof	number	
sizeus	integer	
echo_n	integer	
gain	number	
decay_rate	number	
xcmax	integer	
pow	integer	
width	integer	
freq	number	
xctype	string	
date_time*	string	
id*	string	
<b>position*</b>	object	
waypoint_identifier*	string	
xn*	number	
ye*	number	
zd*	number	
reference_thickness	number	
<b>turbine*</b>	object	
turbine_identifier*	string	
<b>turbine_location*</b>	object	
xn*	number	

NAME	TYPE	DESCRIPTION
ye*	number	
dds_location*	object	
xn*	number	
ye*	number	
zd*	number	
phi*	number	
theta*	number	
psi*	number	

STATUS CODE - 400: Bad request

STATUS CODE - 403: Access denied

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
detail	array	
loc*	array	
msg*	string	
type*	string	

STATUS CODE - 500: Internal server error

## 2. POSITIONS

### 2.1 GET /positions/

#### Read Positions

#### REQUEST

##### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
turbine_id	string	
skip	integer	
limit	integer	

#### RESPONSE

STATUS CODE - 200: Successful Response

##### RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
ARRAY OF OBJECT WITH BELOW STRUCTURE		
waypoint_identifier*	string	
xn*	number	
ye*	number	
zd*	number	
reference_thickness	number	

STATUS CODE - 400: Bad request

STATUS CODE - 403: Access denied

STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
detail	array	
loc*	array	
msg*	string	
type*	string	

STATUS CODE - 500: Internal server error

### 2.2 POST /positions/

#### Create Position

#### REQUEST

## REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
waypoint_identifier*	string	
xn*	number	
ye*	number	
zd*	number	
reference_thickness	number	
turbine_identifier*	string	

## RESPONSE

### STATUS CODE - 200: Successful Response

#### RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
<b>OBJECT WITH BELOW STRUCTURE</b>		
waypoint_identifier*	string	
xn*	number	
ye*	number	
zd*	number	
reference_thickness	number	
<b>host_turbine*</b>	object	
turbine_identifier*	string	
<b>turbine_location*</b>	object	
xn*	number	
ye*	number	
<b>dds_location*</b>	object	
xn*	number	
ye*	number	
zd*	number	
phi*	number	
theta*	number	
psi*	number	
<b>measurements</b>	array	
date*	string	
time*	string	
<b>tof_quality*</b>	object	
sizew_sigma*	number	
amplitude*	number	
dr_sigma*	number	
temp*	number	
thickness*	number	
<b>raw</b>	array	
device_id	string	
sample_id	string	

NAME	TYPE	DESCRIPTION
piezo_id	string	
tof	number	
sizeus	integer	
echo_n	integer	
gain	number	
decay_rate	number	
xcmx	integer	
pow	integer	
width	integer	
freq	number	
xctype	string	
fixed_sensor	object	
sensor_identifier*	string	

STATUS CODE - 400: Bad request

STATUS CODE - 403: Access denied

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
detail	array	
loc*	array	
msg*	string	
type*	string	

STATUS CODE - 500: Internal server error

## 2.3 DELETE /positions/

### Delete Position

#### REQUEST

QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*id_	string	

#### RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
waypoint_identifier*	string	

NAME	TYPE	DESCRIPTION
xn*	number	
ye*	number	
zd*	number	
reference_thickness	number	
<b>host_turbine*</b>	object	
turbine_identifier*	string	
<b>turbine_location*</b>	object	
xn*	number	
ye*	number	
<b>dds_location*</b>	object	
xn*	number	
ye*	number	
zd*	number	
phi*	number	
theta*	number	
psi*	number	
<b>measurements</b>	array	
date*	string	
time*	string	
<b>tof_quality*</b>	object	
sizew_sigma*	number	
amplitude*	number	
dr_sigma*	number	
temp*	number	
thickness*	number	
<b>raw</b>	array	
device_id	string	
sample_id	string	
piezo_id	string	
tof	number	
sizeus	integer	
echo_n	integer	
gain	number	
decay_rate	number	
xcmx	integer	
pow	integer	
width	integer	
freq	number	
xctype	string	
<b>fixed_sensor</b>	object	
sensor_identifier*	string	

STATUS CODE - 400: Bad request

STATUS CODE - 403: Access denied

STATUS CODE - 422: Validation Error

**RESPONSE MODEL - application/json**

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
detail	array	
loc*	array	
msg*	string	
type*	string	

STATUS CODE - 500: Internal server error

---

# 3. SENSORS

## 3.1 GET /sensors/

### Read Sensors

#### REQUEST

##### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
turbine_id	string	
skip	integer	
limit	integer	

#### RESPONSE

STATUS CODE - 200: Successful Response

##### RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
ARRAY OF OBJECT WITH BELOW STRUCTURE		
sensor_identifier*	string	
host_turbine*	object	
turbine_identifier*	string	
turbine_location*	object	
xn*	number	
ye*	number	
dds_location*	object	
xn*	number	
ye*	number	
zd*	number	
phi*	number	
theta*	number	
psi*	number	
sensor_position*	object	
waypoint_identifier*	string	
xn*	number	
ye*	number	
zd*	number	
reference_thickness	number	

STATUS CODE - 400: Bad request

STATUS CODE - 403: Access denied

STATUS CODE - 422: Validation Error

## RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
<b>detail</b>	array	
<b>loc*</b>	array	
<b>msg*</b>	string	
<b>type*</b>	string	

STATUS CODE - 500: Internal server error

## 3.2 POST /sensors/

### Create Sensor

## REQUEST

### REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
<b>sensor_identifier*</b>	string	
<b>turbine_identifier*</b>	string	
<b>sensor_position*</b>	string	

## RESPONSE

STATUS CODE - 200: Successful Response

### RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
<b>sensor_identifier*</b>	string	
<b>host_turbine*</b>	object	
<b>turbine_identifier*</b>	string	
<b>turbine_location*</b>	object	
<b>xn*</b>	number	
<b>ye*</b>	number	
<b>dds_location*</b>	object	
<b>xn*</b>	number	
<b>ye*</b>	number	
<b>zd*</b>	number	
<b>phi*</b>	number	
<b>theta*</b>	number	
<b>psi*</b>	number	
<b>sensor_position*</b>	object	
<b>waypoint_identifier*</b>	string	
<b>xn*</b>	number	
<b>ye*</b>	number	
<b>zd*</b>	number	

NAME	TYPE	DESCRIPTION
reference_thickness	number	

STATUS CODE - 400: Bad request

STATUS CODE - 403: Access denied

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
detail	array	
loc*	array	
msg*	string	
type*	string	

STATUS CODE - 500: Internal server error

### 3.3 DELETE /sensors/

#### Delete Sensor

#### REQUEST

QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*id_	string	

#### RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
sensor_identifier*	string	
host_turbine*	object	
turbine_identifier*	string	
turbine_location*	object	
xn*	number	
ye*	number	
dds_location*	object	
xn*	number	
ye*	number	
zd*	number	
phi*	number	
theta*	number	
psi*	number	

NAME	TYPE	DESCRIPTION
sensor_position*	object	
waypoint_identifier*	string	
xn*	number	
ye*	number	
zd*	number	
reference_thickness	number	

STATUS CODE - 400: Bad request

STATUS CODE - 403: Access denied

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
detail	array	
loc*	array	
msg*	string	
type*	string	

STATUS CODE - 500: Internal server error

# 4. TURBINES

## 4.1 GET /turbines/

### Read Turbines

#### REQUEST

##### QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
skip	integer	
limit	integer	

#### RESPONSE

STATUS CODE - 200: Successful Response

##### RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
ARRAY OF OBJECT WITH BELOW STRUCTURE		
turbine_identifier*	string	
turbine_location*	object	
xn*	number	
ye*	number	
dds_location*	object	
xn*	number	
ye*	number	
zd*	number	
phi*	number	
theta*	number	
psi*	number	
sensors	array	
sensor_identifier*	string	
positions	array	
waypoint_identifier*	string	
xn*	number	
ye*	number	
zd*	number	
reference_thickness	number	

STATUS CODE - 400: Bad request

STATUS CODE - 403: Access denied

STATUS CODE - 422: Validation Error

##### RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
detail	array	
loc*	array	
msg*	string	
type*	string	

STATUS CODE - 500: Internal server error

## 4.2 POST /turbines/

### Create Turbine

#### REQUEST

REQUEST BODY - application/json

NAME	TYPE	DESCRIPTION
turbine_idenfifier*	string	
turbine_location*	object	
xn*	number	
ye*	number	
dds_location*	object	
xn*	number	
ye*	number	
zd*	number	
phi*	number	
theta*	number	
psi*	number	

#### RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
turbine_idenfifier*	string	
turbine_location*	object	
xn*	number	
ye*	number	
dds_location*	object	
xn*	number	
ye*	number	
zd*	number	
phi*	number	
theta*	number	
psi*	number	

NAME	TYPE	DESCRIPTION
<b>sensors</b>	array	
sensor_identifier*	string	
<b>positions</b>	array	
waypoint_identifier*	string	
xn*	number	
ye*	number	
zd*	number	
reference_thickness	number	

STATUS CODE - 400: Bad request

STATUS CODE - 403: Access denied

STATUS CODE - 422: Validation Error

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
<b>detail</b>	array	
loc*	array	
msg*	string	
type*	string	

STATUS CODE - 500: Internal server error

## 4.3 DELETE /turbine/

### Delete Turbine

#### REQUEST

QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
*id_	string	

#### RESPONSE

STATUS CODE - 200: Successful Response

RESPONSE MODEL - application/json

NAME	TYPE	DESCRIPTION
OBJECT WITH BELOW STRUCTURE		
turbine_identifier*	string	
turbine_location*	object	
xn*	number	
ye*	number	
dds_location*	object	
xn*	number	

NAME	TYPE	DESCRIPTION
ye*	number	
zd*	number	
phi*	number	
theta*	number	
psi*	number	
<b>sensors</b>	array	
sensor_identifier*	string	
<b>positions</b>	array	
waypoint_identifier*	string	
xn*	number	
ye*	number	
zd*	number	
reference_thickness	number	

STATUS CODE - 400: Bad request

STATUS CODE - 403: Access denied

STATUS CODE - 422: Validation Error

**RESPONSE MODEL - application/json**

NAME	TYPE	DESCRIPTION
<b>OBJECT WITH BELOW STRUCTURE</b>		
<b>detail</b>	array	
loc*	array	
msg*	string	
type*	string	

STATUS CODE - 500: Internal server error

## 6 REFERENCES

---

- <sup>1</sup> <https://www.json.org/json-en.html>
- <sup>2</sup> <https://json-ld.org/>
- <sup>3</sup> <https://graphql.org/>
- <sup>4</sup> <https://www.ontotext.com/try-graphdb-se>
- <sup>5</sup> <https://www.w3.org/TR/vocab-data-cube/>
- <sup>6</sup> <https://platform.ontotext.com/3.0/soml/index.html>
- <sup>7</sup> <https://github.com/graphql/graphql-spec/issues>
- <sup>8</sup> <https://www.sqlite.org/index.html>
- <sup>9</sup> <https://www.python.org/>
- <sup>10</sup> <https://fastapi.tiangolo.com/>
- <sup>11</sup> <https://graphene-python.org/>
- <sup>12</sup> <https://swagger.io/specification/>
- <sup>13</sup> <https://swagger.io/tools/swagger-ui/>
- <sup>14</sup> <https://www.go-fair.org/fair-principles/>