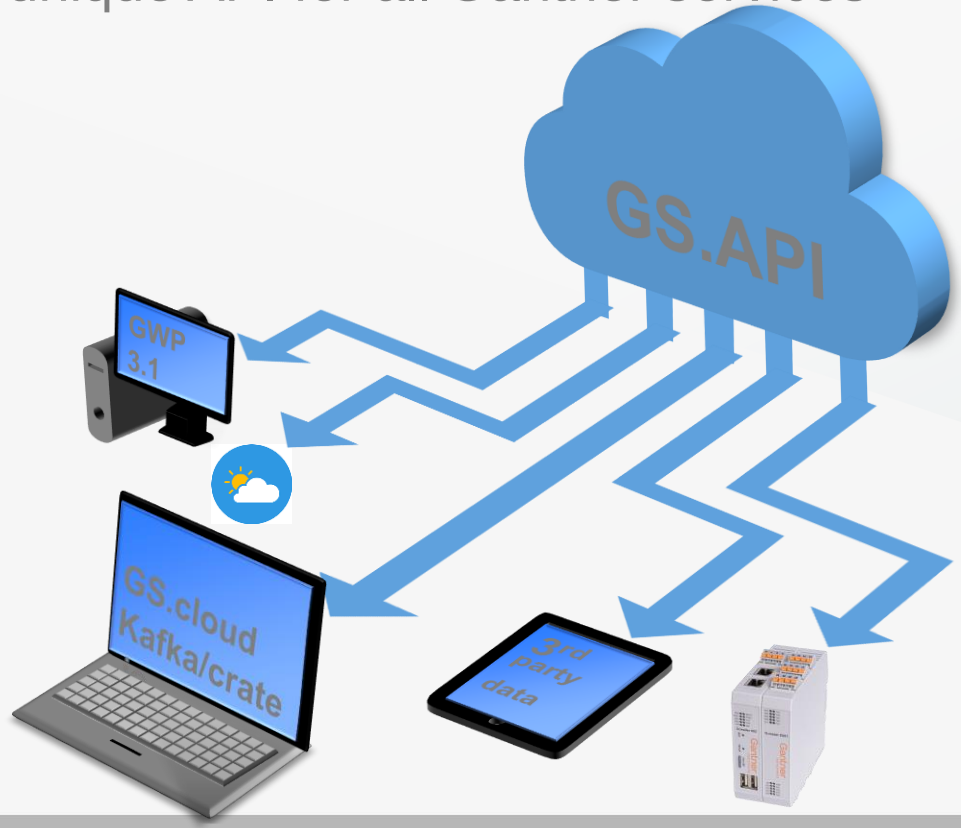
GANTNER ENVIRONMENT GS.API

Cloud Services

Gantner instruments

GS.API

A unique API for all Gantner services



- **GS.API**
- **GS.cloud**
  Gantner.webportal next generation
- **Gantner.webportal 3.1.6** up to 2021
- **Weather and Energy Forecast**
- **3rd Party**
- **Gantner Q.reader / Gantner.RAS**
  remote control of utility scale PV power plants
  energy off taker

GS.cloud

Gantner next generation solar platform for monitor, control and analyses of PV power plants

GS.API

GS.API is unique API for utility scale PV applications

CRATE.IO

CrateDB is a distributed SQL database management system that integrates a fully searchable document-oriented data store. It is open-source, written in Java, based on a shared nothing architecture, and is designed for high scalability and includes components from Presto, Lucene, Elasticsearch and Netty..

KAFKA

Apache Kafka is an open-source stream-processing software platform. The project aims to provide a unified, high-throughput, low-latency platform for handling real-time data feeds.

1. Access-Token

To access the API, you first must fetch a session token. The tokens are user-bound and valid for one year, which is why it is not necessary to generate a new token for every API call. The user's site access restrictions also apply to the API. Every admin user can set up these restrictions and the API call limits via the configuration pages of the portal.

To get an access token, you must pass your login data via form fields and send it to the “login_check” node. Using curl such a request could look like this:

```
curl -sX POST -F "_username=mylogin" -F "_password=mypassword" "https://myportal.gantner-webportal.com/app/api/v1/login_check"
```

The returned session token is contained in a JSON object. For all further API calls you must insert this token in the header of the request. As already mentioned, you can use the token for a year before you must fetch a new one. With curl you can access the “sites” node like this:

```
curl -sH "Authorization: Bearer MY_API_TOKEN" "https://myportal.gantner-webportal.com/app/api/v1/sites"
```

2. Content Encoding

In order to save traffic for requests that deliver huge amounts of data back content encoding via gzip is for such calls available. Please note: This functionality is not available for every API function. If you make heavy usage of compressing you may not reach very high poll rates, depending on the size of content to be compressed.

3. HTML response codes

The GS.API call results are based on the standard HTML server response codes.

200	OK
400	Bad API Request
401	No valid License
403	Access denied
404	Object not found
429	Call limit exceeded
500	Internal Server Error

Note: The Accept-Encoding request HTTP header can be used to compress the requested data.
Also note: The amount of data you can request with one call is limited by the product of the channels and the data sets.
If you request too much data at once you may receive an error code instead of your data.

3.1. Request limits

In order to prevent the server from being overloaded by too many API requests, there are limits to the number of requests allowed. By default this is 15 requests per minute and 600 requests per day. However, these settings can be adjusted in the user configuration by an administrator if required.

4. Retrieve Sites

The very first step is to retrieve all available sites. Since components are directly linked to the sites it is necessary to know the Site-ID to which the component is linked to. This can be easily done with:

myportal.gantner-webportal.com/app/api/v1/sites

The result looks as follows (Example with two available sites):


```
{
        "1": "Tannenberg",
        "2": "Annaberg"
}
```


Please note:
In order to keep it clean and simple you can later retrieve based on the Site-ID additional site information if necessary.

5. Usage of Site-ID’s

After you got your unique Site-ID you can construct your API-calls like:

myportal.gantner-webportal.com/app/api/v1/sites/1/<API-Command>

to ensure all API calls refer to site 1 (Tannenberg in this example).

If you want to use the site Annaberg then your next API-Command should start with:

myportal.gantner-webportal.com/app/api/v1/sites/2/<API-Command>

5.1. Detailed site information

With the “info” node / command you can get all the meta information, that is stored in the portal for a specific site:

myportal.gantner-webportal.com/app/api/v1/sites/47/info


```
{
    "0": {
        "altitude": 22,
        "comment_on_site_layout": null,
        "date_of_commissioning": "22.03.2016",
        "dc_power_nominal": 4999,
        "id": 47,
        "inverter_concept": "CENTRAL",
        "latitude": 12.34,
        "longitude": 5.432,
        "public_ip_address": null,
        "site_address": "xyz Solar Power Plant\r\nUK",
        "site_type": "GSM",
        "timezone": "UTC+0",
        "timezone_name": "Europe/London",
        "title": "Annaberg"
    },
    "assets": [
        {
            "amount": "420",
            "article": "JKM260P",
            "producttype": "PV module"
        },
        {
            "amount": "6",
            "article": "Conext Core XC 680",
            "producttype": "Inverter"
        },
        {
            "amount": "35",
            "article": "String.CC24/12",
            "producttype": "Combiner box"
        },
        {
            "amount": "2",
            "article": "SMP10",
            "producttype": "Irradiance_inclined"
        }
    ],
    "preview_image": "/app/gantnerTheme/Images/annaberg.jpg"
}
```

6. Retrieve Component Types

The second step is to retrieve all available component types and subtypes for the given site. This function is used to check what device types are installed on site. For the sake of simplicity, the example shows only a small subset of the possible component types. Other types (without claim to completeness) could be:

Site, Substation (Transformer Station), Station, CombinerBox, Inverter, MPPTracker, String, Meter, ProtectionRelay, UPS, Controller and Sensor

myportal.gantner-webportal.com/app/api/v1/sites/1/types&language=de

returns all available component types and quantity on the given site (translated into the requested language).


```
{
        "CombinerBox": {
                " CombinerBox ": {
                        "count": "26",
                        „sub_type": "CombinerBox",
                        "type": " CombinerBox "
                }
        },
        „Sensor": {
                "GhPyr": {
                        "count": "2",
                        „sub_type": "GhPyr",
                        "type": "Sensor"
                },
                "GiPyr": {
                        "count": "2",
                        „sub_type": "GiPyr",
                        "type": "Sensor"
                }
        }
}
```

7. Retrieve Location List

myportal.gantner-webportal.com/app/api/v1/sites/1/locations?type=CombinerBox&meta_tags=dc

This function returns a list of all locations where components of the requested type are installed.
Input: Site (via URL), Component Type (via argument) and language (via optional argument)

The language parameter is supported. If not specified it returns (in this example) the English component and parameter list.

Gantner-naming (colorized purple) is used as reference ID for further API-calls.

Valid types are the key values of the “Retrieve Component Types” query from the previous chapter.

Please note:
To keep the example simple, it only shows a few parameters for the CombinerBox. The real parameter list is much longer. The purple keys in the result can be used to perform further API-calls.

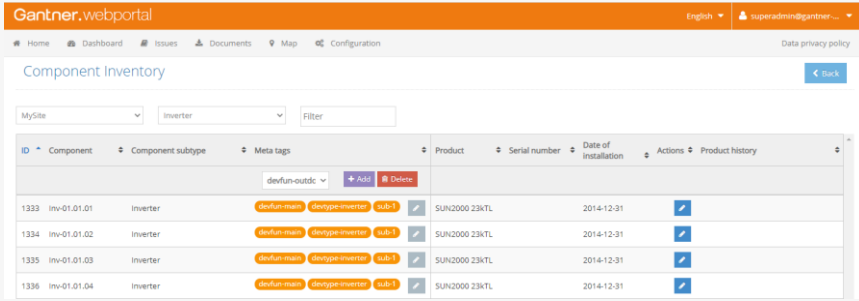

```
{
        "CB-1.A.1.01": {
                "manufacturer": "Gantner Instruments",
                "name": "CombinerBox-1.A.1.01",
                "parameters": {
                        "Pdc": "DC Power",
                        "Vdc": "DC Voltage"
                },
                "product": "String.CB 116",
                "meta_tags": [ "combiner", "dc" ]
        }
        "CB-1.A.1.02": {
                "manufacturer": "Gantner instruments",
                "name": "CombinerBox-1.A.1.02",
                "parameters": {
                        "Pdc": "DC Power",
                        "Vdc": "DC Voltage"
                },
                "product": "String.CB 116"
                "meta_tags": [ "combiner", "dc", "shaded" ]
        }
}
```

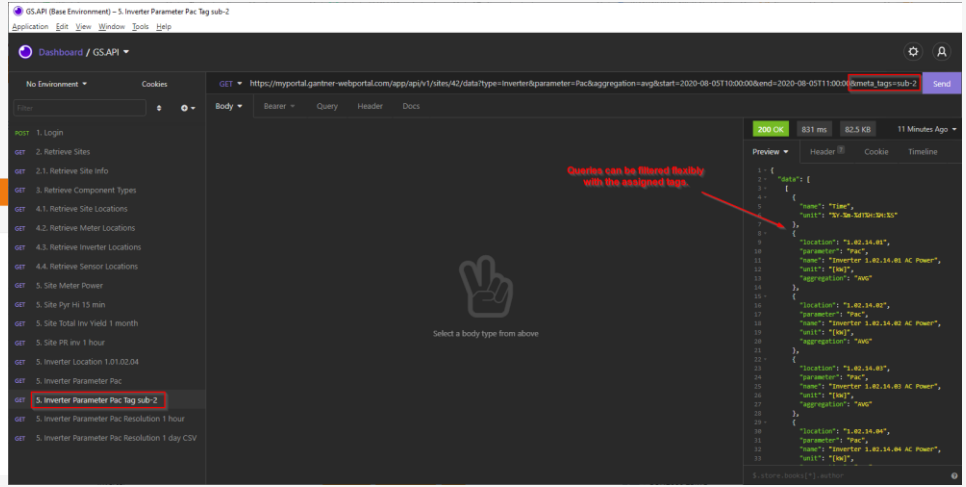
7.1. Usage of Tags to flexibly filter results

The meta_tags are available for the **location** and the **data** query node, that is explained in the following chapters.

myportal.gantner-webportal.com/app/api/v1/sites/42/data?type=Inverter&parameter=Pac&aggregation=avg&start=2020-08-05T10:00:00&end=2020-08-05T11:00:00&meta_tags=sub-2

Tags can be freely assigned to the devices in the web frontend.
Afterwards you can use the optional API parameter meta_tags to filter your results with any tag combination.
Several tags can be separated in the query using a comma.





8. Retrieve Measurement Data

8.1. General Parameters

The data can be accessed in two ways using the “data” or the “datagroups” function. The following parameters are required or available in both cases.

Parameter	Required	Description
language	optional	language of FixedChannel titles • en (default) • de • logger
start	yes	start time of requested interval (local site time zone) format: YYYY-MM-DDThh:mm:ss
end	yes	end time of requested interval (local site time zone) format: YYYY-MM-DDThh:mm:ss
resolution	optional	valid data resolutions: • raw (default) • 15min • 30min • 1hour • 1day • 1month • 1year
aggregation	optional	type of aggregation over time for longer resolutions - valid methods are: • avg • sum • min • max
format	optional	format of requested data: • json (default) • csv

8.2. Retrieve Data by Component

The component list result can be used to query the data in a component-based way. Since the component list does rarely change, it is recommended to cache this list for further data queries. This type of access requires some additional parameters to filter for the required channels:

Parameter	Required	Description
type	optional	query all components of a component type
location	optional	query parameters of a single component
parameter	optional	query only this parameter of the requested components
meta_tags	optional	only return data from devices with the assigned tags

8.2.1. Valid parameter combination examples

- Component Type + Parameter - query the DC-Power of all inverters:

myportal.gantner-webportal.com/app/api/v1/sites/1/data?start=2019-01-31T14:40:00&end=2019-02-04T11:50:00&type=Inverter&parameter=Pdc

- Component Type + Location – query all parameters of a single device:

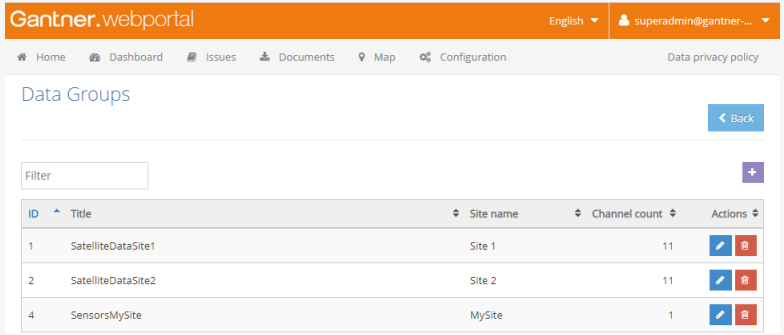
myportal.gantner-webportal.com/app/api/v1/sites/1/data?start=2019-01-31T14:40:00&end=2019-02-04T11:50:00&type=CombinerBox&location=CB-1.A.1.01

- Component Type + Location + Parameter – query single channel:

myportal.gantner-webportal.com/app/api/v1/sites/1/data?start=2019-01-31T14:40:00&end=2019-02-04T11:50:00&type=CombinerBox&location=CB-1.A.1.01&parameter=Pdc

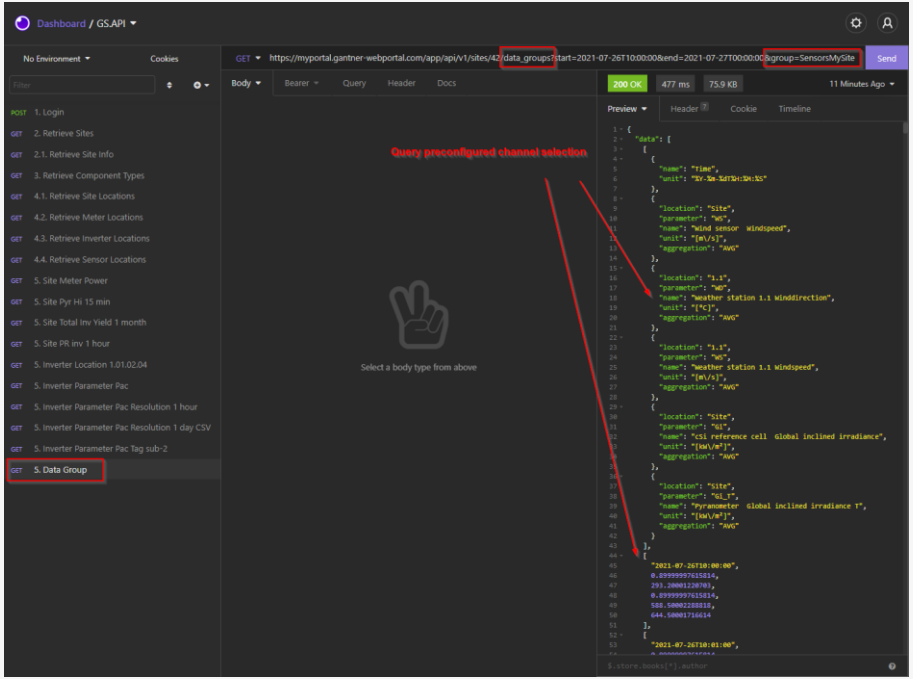
8.3. Retrieve Preconfigured Data Group

For this type of query, it is required to preconfigure data groups in the web portal first. Each data group is a selection of channels linked to a group id. This group id is added as parameter to access the data for the selected channels.



In this example a selection of weather sensor channels was grouped under the id “SensorsMySite”:

myportal.gantner-webportal.com/app/api/v1/sites/42/datagroups?group=SensorsMySite&start=2021-07-26T10:00:00&end=2021-07-27T00:00:00



8.4. Response examples
8.4.1. CSV result

Request URL:

myportal.gantner-webportal.com/app/api/v1/sites/1/data?start=2019-01-31T14:45:00&end=2019-01-31T14:50:00&type=Inverter&location=Inv-1.1.2.3&format=csv
→ first row contains channel names, second units, third the time aggregation type

```
Date;Time;IV.1.1.2.3 Active Power;IV.1.1.2.3 Temperature;IV.1.1.2.3 Power DC;IV.1.1.2.3 Serial
[dd.mm.YYYY];[HH:MM:SS];[kW];[°C];[kW];[]
Aggregation;[AGG];AVG;AVG;AVG;MAX
31.01.2019;14:45:00;1.31;-2;5.44;9554asdfasdk2o29
31.01.2019;14:50:00;1.35;-2.2;5.46;9554asdfasdk2o29
```

GS.API

Json reply example

Gantner instruments

8.4.2. Json result

Request URL:

myportal.gantner-webportal.com/app/api/v1/sites/1/data?start=2019-06-01T12:00:00&end=2019-06-01T12:05:00&type=Inverter&location=Inv-1.1.2.3

Reply on failure:


```
{
        "status_code": "404",
        "message": "Installation not found."
}
```


Reply on success:


```
{
        "data": [[
                {
                        "name": "Time",
                        "unit": "%Y-%m-%dT%H:%M:%S"
                },
                {
                        "type": "Inverter",
                        "location": "1.1.2.3",
                        "parameter": "Pac",
                        "name": "IV.1.1.2.3 Active Power",
                        "unit": "[kW]",
                        "aggregation": "AVG"
                },
                {
                        "type": "Inverter",
                        "location": "1.1.2.3",
                        "parameter": "Freq",
                        "name": "IV.1.1.2.3 Frequency",
                        "unit": "[Hz]",
                        "aggregation": "AVG"
                },
                {
                        "type": "Inverter",
                        "location": "1.1.2.3",
                        "parameter": "Status",
                        "name": "IV.1.1.2.3 Status",
                        "unit": "[]",
                        "aggregation": "MAX"
                }
        ]
        [ "2019-06-01T12:00:00", 128.7, 49.9, 2 ],
        [ "2019-06-01T12:00:00", 132.3, 50, 2 ]]
}
```

9. Alarms

9.1. Alarm Configuration

To get a list of the Alarms that are configured in the portal you can use a request like this:

myportal.gantner-webportal.com/app/api/v1/sites/1/alertconfigurations

```
[
        {
                "id": "123456",
                " title": "alertconfiguration title",
                " category": "alertconfiguration category title",
                " executiontime": "2020-10-09T13:04:04",
                "last_checked_timestamp": "2020-10-09T12:45:00"
        },
        …
]
```

9.2. Alarm Configuration details

The ids from the alertconfigurations list can then be used to get detailed information about the alert check conditions. You just must add the alarm id to the request path like this:

myportal.gantner-webportal.com/app/api/v1/sites/1/alertconfigurations/123456

The result contains also information, that you can also see when you configure the alarm in the web portal.


```
{
        "id": 123456,
        "active": true,
        "category": "others",
        "recipients": [ "mymail@somewhere.com" ],
        "title": "Grid failure",
        "alert activation": {
                "active": true,
                "aoi_dependency": false,
                "dynamic_component": "Grid connection point",
                "dynamic_parameter": "Eac_Sum_tt",
                "fixedchannel_id": 1560,
                "max": 0,
                "min": 100000,
                "shading_angle_dependency": false
        },
        "alert modes": {
                "average": 0,
                "check_from": -1,
                "check_to": -1,
                "duration": 15,
                "max": 0,
                "measurement_values": [ "1" ],
                "min": 0,
                "range_definition_type": null,
                "severity": "failure",
                "value_range_type": "exactNot",
                "variance": 0
        }
}
```

9.3. Alarm history

To determine which alarms have been triggered, you can use the alarm node:

myportal.gantner-webportal.com/app/api/v1/sites/1/alerts/alerts

The alarm list only contains the alarms configured in the portal. In addition, however, it is also checked whether there are gaps in the measurement data ("dataloss" in the case of missing measurement values due to the failure of a device) or if no data is supplied to the portal ("delayed import" due to network disruptions). These **system errors** can be queried with:

myportal.gantner-webportal.com/app/api/v1/sites/1/alerts/dataloss

myportal.gantner-webportal.com/app/api/v1/sites/1/alerts/importdelay


```
[
    {
        "appeared_at": "2021-01-05T13:20:00",
        "configuration": "Inverter 1.C.2 CB deviation",
        "disappeared_at": null,
        "message": "String.C2.02 DC Power Normalized Var: Measurement is less than<span> </span>-15.00<span> [%]</span> <br>Observation period: <strong>60<span> </span>Minutes</strong> <br>Dependent channel<span>: </span><strong>Irradiance inclined pyranometer Global inclined irradiance Temperature Corrected: </strong>is greater than<span> </span>0.10<span> [kW/m\u00b2]</span>",
        "severity": "warning",
        "status": "open"
    },
    {
        "appeared_at": "2020-10-29T13:35:00",
        "configuration": "Inverter State",
        "disappeared_at": null,
        "message": "Inverter.A1 Status: Measurement is out of range<span> </span>4&#039;036.00<span> \u2013 </span>6&#039;645.00<span> [n/a]</span> <br>Observation period: <strong>30<span> </span>Minutes</strong> <br>Dependent channel<span>: </span><strong>Irradiance inclined pyranometer Global inclined irradiance Temperature  Corrected: </strong>is greater than<span> </span>0.10<span> [kW/m\u00b2]</span>",
        "severity": "failure",
        "status": "open"
    },
    ...
]
```

GS.API

Retrieve Measurement Data

You can filter the returned alarms by adding additional parameters. By default, you only see alarms that are still open. For creating reports, you may want to go back further in time and see alert issues that were already solved.

Parameter	Required	Description
status	optional	Alarm state filter: • open (default) • closed • suspended • all
severity	optional	Alarm severity filter: • failure • warning • all (default)
active_after	optional	active after this date format: YYYY-MM-DD only use in combination with active_before parameter
active_before	optional	active before this date format: YYYY-MM-DD only use in combination with active_after parameter

Example:

myportal.gantner-webportal.com/app/api/v1/sites/1/alerts/alerts?status=all&active_after=2020-06-01&active_before=2020-06-16

10. Issues

10.1. List Issues

The issue list for the current user can be acquired with:

myportal.gantner-webportal.com/app/api/v1/issues

By using the returned ID, detailed information on an issue can be requested.


```
[
        {
                "contactPerson": "Superadmin",
                "creationTimestampOfIssue": 1591705248,
                "deadline": "2020-06-09 00:00",
                "id": 38,
                "installation": "Annaberg",
                "issueCategory": "Calibration",
                "issuePriority": "Normal",
                "issueState": "Open",
                "issueType": "Repair",
                "responsibleUser": "Superadmin",
                "title": "Test Datumsbug",
                "updatedAt": 1591705911
        } {
                "contactPerson": "Superadmin",
                "creationTimestampOfIssue": 1591704405,
                "deadline": "",
                "id": 37,
                "installation": "Annaberg",
                "issueCategory": "Calibration",
                "issuePriority": "Normal",
                "issueState": "Open",
                "issueType": "Repair",
                "responsibleUser": "Superadmin",
                "title": "Test 2020_06_09",
                "updatedAt": 1591706024
        },
        ...
]
```

10.2. Issues Details

Detailed issue information can be quarried with the ID like this:

myportal.gantner-webportal.com/app/api/v1/issues/38

To keep the example short, only part of the reply has been included.


```
{
        "activity": [
                {
                        "attachments": "",
                        "authorName": "Superadmin",
                        "date": "",
                        "emailNotifications": "",
                        "internalNote": "",
                        "message": "",
                        "timeTrack": "0:0h",
                        "timeTrackTimeData": { … },
                        "timestamp": 1591705856
                }
        ],
        "author": "Superadmin",
        "category": "Calibration",
        "channel": "",
        "component": "CB-1.A.1.01",
        "componentType": "Combiner box",
        "contactPerson": "Superadmin",
        "contactPersonPhone": "",
        "contractContact": "",
        "creationTimestampOfIssue": 1591705248,
        "customer": "",
        "deadlinetimestamp": 1591653600,
        "deliveryContact": "",
        "description": "Ticket description",
        "escalationLevel": "",
        "estimatedHours": "",
        "excludeFromReports": false,
        "externalUrl": "",
        "fixedChannel": "",
        "foreignIssueId": "",
        "id": 38,
        "installation": "Annaberg",
        …
        "issueState": "Open",
}
```