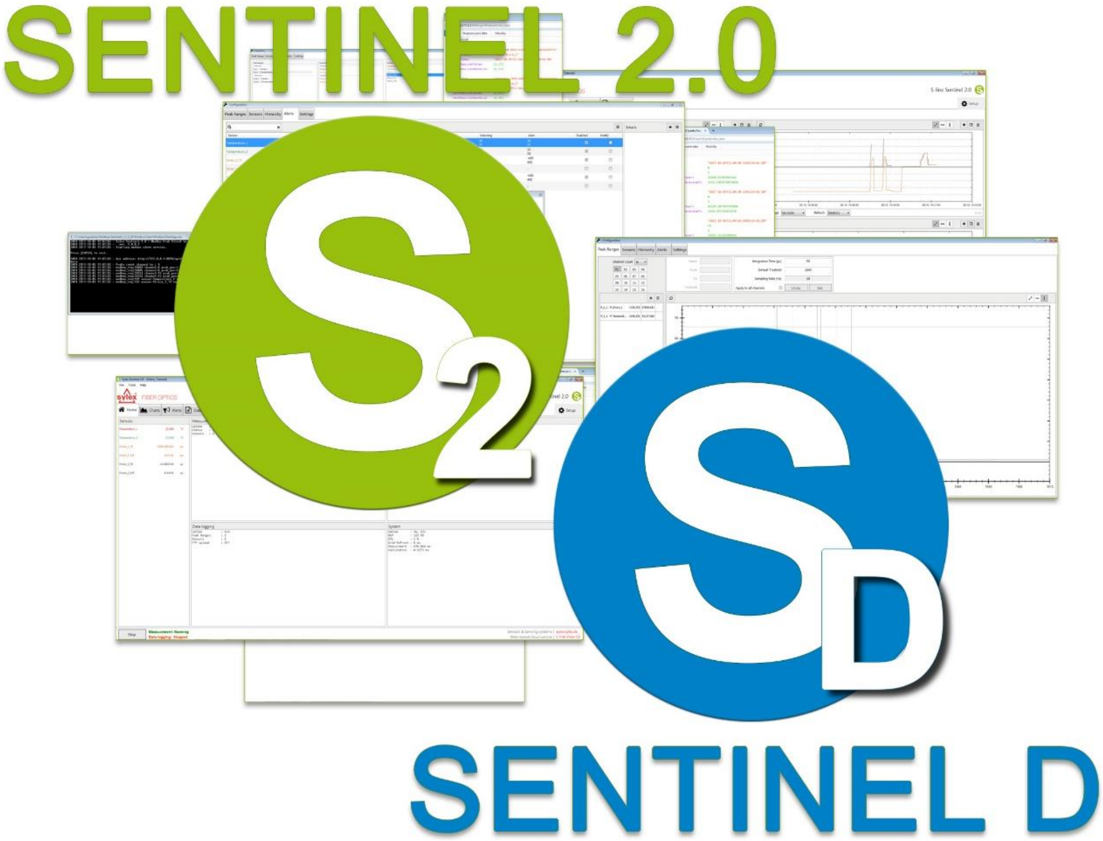
sylex® FIBER OPTICS

Sentinel API Manual



Sentinel API Manual

The current version, Sentinel: 1.1.60

The current version, Sentinel-D: 0.4.3

Release date: 27th of June, 2024 | **Version:** 01 | **Revision:** 00

Index

List of API calls

List of API results

List of tables

1. Introduction

1.1. Hardware requirements

The Sentinel software was designed to have low hardware requirements for the driving PC, however, a certain performance of the computer is necessary for the proper working of the software. The Sentinel software is a Windows-based application and thus bear in mind that the operating system needs to be Windows-based.

A list of the hardware specifications for the driving computer with the Sentinel software is listed below.

- Operating system (**OS**): x64, Windows 7 - Windows 10[1] (Ubuntu 22.04[2])
- **CPU**: Intel Pentium N4200 or better (for optimal performance, 4 cores/threads are recommended with a frequency of 2,5 GHz)
- **RAM**: 8 GB RAM or more
- **GPU**: Intel® HD Graphics 505 or better
- **Storage**: SSD
- Available Storage **Space**: 10GB or more
- **I/O:** 1x free USB 2.0 (or later) port

1.2. Sentinel API

The Sentinel API allows the user to use the API calls to read out the data from the unit without considering the data log or even starting the data logging at all. The API calls can be sent through a TCP protocol or called locally. The API function of Sentinel creates an API server at the local host at port 8024 – for this reason, the application needs to be launched with administration privileges.

After the creation of the API server, the user can use the API calls to return the data in JSON format. To see the full list of commands, please contact SYLEX support.

[1] The software can operate even on Windows XP; however, the OS is already discontinued and without the required security updates. We do not recommend running the software on an OS older than (IOT) Windows 7 x64.
[2] Sentinel can be run under Ubuntu 22.04 LTS in Demo mode using „wine“ emulator.

```
Example of the API call: http://127.0.0.1:8024/api/v1/peaks/last_data
```

API call 1: Example of an API call returning the raw wavelengths.

2. API related calls - Channels

2.1. Measurement Configuration Request

To list all defined configurations within the Sentinel set-up:

```
/api/v1/channels/
```

API call 2: List all defined configurations within peak-ranges set-up.

To list all specific configurations within the Sentinel set-up:

```
/api/v1/channels/CH_IDX/config
```

API call 3: List channel-specific configurations within peak-ranges set-up.

Where in both cases above CH_IDX is the index of the acquired channel of the interrogator set-up, i.e. 1 for Channel 1.

```
[
  {
    "Index": 0,
    "IntegrationTimeMcs": 1338,
    "CorrectionCycleMs": 10000,
    "DefaultTreshold": 2000,
    "SamplingRateHz": 10
  },
  {
    "Index": 1,
    "IntegrationTimeMcs": 40,
    "CorrectionCycleMs": 10000,
    "DefaultTreshold": 1000,
    "SamplingRateHz": 10
  },
]
```

API result 1: Example results from API call returning peak ranges configuration.

2.2. Raw measurement data per Channel

To return raw measurement data by channel:

```
/api/v1/channels/CH_IDX/last_data
```

API call 4: Returns RAW wavelength data for a specific channel.

Where CH_IDX is the index of the acquired channel of the interrogator set-up, i.e. 1 for Channel 1. The API call 4 will result in an array described in Table 1.

Field	Variable type	Description
valid	Boolean	True if the measurement was successful.
caseTemperature	Number	Temperature of case
channelIndex	Number	Channel index (starts at 1)
correctionCycleMs	Number	The integration time of the channel
pdblData	Array Of Numbers	The array of measured powers
pdblWavelength	Array Of Numbers	Array of measured wavelengths
peakCount	Number	The number of found raw peaks
noiseThreshold	Number	Noise threshold
peakPowers	Array Of Numbers	Array of powers for detected peaks
pixelCount	Number	Number of pixels (defines the size of pblData and pblWavelength array)
timeStamp	DateTime	Time of Measurement
lastCorrectionTimeStamp	Number	Last time of temperature correction

Table 1: Resulting array from the API call, **last_data**.

```
 [ {
  "caseTemperature": 36.677169708616645,
  "channelIndex": 1,
  "correctionCycleMs": 60000,
  "lastCorrectionTimeStamp": "2019-04-23T15:03:47.2400954+02:00",
  "noiseThreshold": 36347.0,
  "pdblData": [
    0.41392303526754404,
    -1.2429741135460972,
        ...
  ],
  "pdblWavelength": [
    1507.5174806369328,
    1507.6874806369344,
    ...
  ],
  "peakCount": 1,
  "peakSearchMode": 0,
  "peakPowers": [
    42326.857754871431,
    0.0,
        ...
  ],
  "pixelCount": 512,
  "timeStamp": "2019-04-23T15:04:12.4396829+02:00",
  "valid": true

} ]
```

API result 2: Example results from **?last_data** call.

3. API related calls - Peaks

3.1. Actual peaks values

To list all detected peaks:

```
/api/v1/channels/CH_IDX/last_data
```

API call 5: List all detected peaks within the peak-ranges set-up.

To list a specific channel detected peaks:

```
/api/v1/peaks/last_data?ch=CH_IDX
```

API call 6: List all detected peaks for a specific channel within the peak-ranges set-up.

Where in both cases above CH_IDX is the index of the acquired channel of the interrogator set-up, i.e. 1 for Channel 1.

Field	Variable type	Description
channelIdx	Number	Channel index (starts at 1)
timeStamp	Number	Measurement time
channelOrder	Number	Index of the detected peak in the channel
lastMeasuredPower	Number	Value of peak power
lastMeasuredWaveLenght	Number	Value of peak wavelength

Table 2: Resulting array from the API call, **peaks/last_data**.

```
[
  {
    "timeStamp": "2019-04-23T12:04:43.8453736+02:00",
    "channelIdx": 4,
    "channelOrder": 1,
    "lastMeasuredPower": 53911.8726820779,
    "lastMeasuredWaveLenght": 1527.4816899859832
  },
  {
    "timeStamp": "2019-04-23T12:04:43.876403+02:00",
    "channelIdx": 4,
    "channelOrder": 2,
    "lastMeasuredPower": 49345.949226305856,
    "lastMeasuredWaveLenght": 1530.5062792002664
  },
  {
    "timeStamp": "2019-04-23T12:04:43.9069326+02:00",
    "channelIdx": 4,
    "channelOrder": 3,
    "lastMeasuredPower": 49823.337657721051,
    "lastMeasuredWaveLenght": 1535.2860759566856
  }
]
```

API result 3: Example API call result from **?peaks/last_data**.

4. API related calls - Sensors

4.1. Actual calculated sensor values

To list all sensors:

```
/api/v1/sensors/last_data
```

API call 7: List all defined sensors within the sensor set-up.

To list a specific sensor:

```
/api/v1/sensors/last_data?id=SENS_ID
/api/v1/sensors/last_data?name=SENS_NAME
```

API call 8: List specific sensors within the sensor set-up according to the sensor ID or Name.

Where SENS_ID and SENS_NAME are the identifier/name of the requested sensor.

Field	Variable type	Description
Id	Text	Sensor GUID
Name	Text	Name of sensor.
Description	Text	Sensor description
Function	Number	Enum of statistic function.
		0 – None 1 – Average 2 – Min 3 – Max 4 – St. Dev
FunctionSampleCount	Number	Sample count for statistic function.
IsReferenced	Boolean	Flag set to true if sensor use reference value.
ReferenceValue	Number	Reference value.
ReferenceDeltaValue	Number	Used for internal calculations.
Template	JSON Object	Sensor Template

Table 3: Resulting array from the API call, **last_data?id=SENS_ID**.

```
[
  {
    "Id": "00cddd72-83c5-4200-ba0b-0c4424770856",
    "Name": "SENSOR_0",
    "Description": null,
    "TemplateId": "TEMPL_ID-2258-4361-a2e4-abd1ed4d4e5d",
    "Function": 0,
    "FunctionSampleCount": 1,
    "IsReferenced": false,
    "ReferenceValue": 0.0,
    "ReferenceDeltaValue": "NaN",
    "Template": {
      "BuildIn": true,
      "OutputUnit": "°C",
      "Id": "TEMPL_ID-2258-4361-a2e4-abd1ed4d4e5d",
      "Name": "Temperature, 2nd grade",
      "Description": "2nd grade used for example with TP-02",
      "Formula": "Ts1*pow(((λTact-λTref)/λTref),2)+Ts2*((λTact-
λTref)/λTref)+Ts3",
      "CanReference": true,
      "DecimalPrecision": 3
    }
  }
]
```

API result 4: Example API call result from **last_data?id=SENS_ID**.

5. API related calls - Alerts

5.1. Alert settings

To return alert settings:

```
/api/v1/alerts
/api/v1/alerts/config
```

API call 9: API call to return “alert” settings.

Field	Variable type	Description
Id	Text	Sensor GUID
Source	Text	For internal use only.
SourceId	Text	For internal use only.
Enabled	Boolean	True if the alert definition is in use.
SendNotications	Boolean	True if this alert definition is also sending notifications.
AlertLo	Number	Alert value.
AlertHi	Boolean	Alert value.
WarnLo	Number	Alert value.
WarnHi	Number	Alert value.
Treshold	Number	Threshold of alert triggering.
SourceSensor	JSON Object	Sensor Object

Table 4: Resulting array from the API call, **alerts/config**.

```
  {
    "Id": "609a4e28-8239-404b-855d-307ed766d7f2",
    "Source": 2,
    "SourceId": "00cddd72-83c5-4200-ba0b-0c4424770856",
    "Enabled": true,
    "SendNotications": false,
    "AlertLo": 2393245.24,
    "AlertHi": 2394229.728,
    "WarnLo": 2393481.629,
    "WarnHi": 2394037.837,
    "Threshold": 1.0,
    "SourceSensor": {
      "Id": "00cddd72-83c5-4200-ba0b-0c4424770856",
      "Name": "SENSOR_0",
      "Description": null,
      "TemplateId": "TEMPL_ID-2258-4361-a2e4-abd1ed4d4e5d",
      "Function": 0,
      "FunctionSampleCount": 1,
      "IsReferenced": false,
      "ReferenceValue": 0.0,
      "ReferenceDeltaValue": "NaN",
      "Template": {
        "BuildIn": true,
        "OutputUnit": "°C",
        "Id": "TEMPL_ID-2258-4361-a2e4-abd1ed4d4e5d",
        "Name": "Temperature, 2nd grade",
        "Description": "2nd grade used for example with TP-02",
        "Formula": "Ts1*pow(((λTact-λTref)/λTref),2)+Ts2*((λTact-
λTref)/λTref)+Ts3",
        "CanReference": true,
        "DecimalPrecision": 3
      }
    }
```

API result 5: Example API call result example from sensors/config.

5.2. Alert values

To list all active alerts:

```
/api/v1/alerts/active
```

API call 10: API call to return and list active alerts.

To list all finished alerts:

```
/api/v1/alerts/finished
```

API call 11: API call to return and list finished alerts.

Field	Description
level	Type of alert (alert / warning / all)
count	Max number of returned alerts
from	Timestamp of oldest alert.
sensor	name of specific sensor

Table 5: Optional filter parameters.

Example queries:

```
/api/v1/alerts/active?level=warning&from=2019-04-23T12:00:00
/api/v1/alerts/active?sensor_name=SENSOR_0&from=2019-04-23T12:00:00&count=10
```

API call 12: API call to return active alerts using a query filter.

Field	Variable type	Description
filter	JSON Object	Describes query filter
events	Array of JSON Objects	List all events

Table 6: Optional filter array fields.

Field	Variable type	Description
NotificationStatus	Number	For internal use only.
ModificationTime	DateTime	Timestamp of last modification.
SensorId	GUID	ID of sensor that triggered the alert event.
Start	DateTime	Start of event.
End	DateTime	End of event. (null if event is not finished)
Level	Number	Alert orWarning
Zone	Number	For internal use only.
Value	Number	Sensor value that triggered this event.
Message	Text	Event message
SensorName	Text	Name of sensor that triggered the alert event.

Table 7: Event object description.

```
  {
  "filter": {
    "sensor_name": null,
    "level": "warning",
    "count": 0,
    "from": "2019-04-23T12:00:00"
  },
  "events": [
    {
      "NotificationStatus": 0,
      "ModificationTime": "2019-04-23T15:53:20.45031+02:00",
      "SensorId": "00cddd72-83c5-4200-ba0b-0c4424770856",
      "Start": "2019-04-23T15:53:20.4498096+02:00",
      "End": null,
      "Level": 0,
      "Zone": 1,
      "Value": 2393874.8,
      "Message": "Over high treshold. Treshold = 2393858.259. Sensor value =
2393874.8",
      "Status": 0,
      "SensorName": "SENSOR_0"
    }
  ]
}
```

API result 6: API call result example from **sensors/config**.

6. Sentinel API troubleshooting guide

The table below aims to help you with troubleshooting issues related to the Sentinel API. A general troubleshooting guide related to the S-line hardware and Sentinel software is included in the Sentinel manual.

Behavior or Error message	Error cause and fix
Behavior: The API calls are returning a “not reachable” error without any data.	Please, ensure the software was started with enough priviligies to open the default API port 8024.
Behavior: The API calls are returning a “not reachable” error without any data.	Please, ensure the API server is started within the Sentinel GUI. To start the API software navigate to “Tools – Start API server mode”.

If none of the above helps to solve your issue, please, feel free to contact us.

For more information contact our sales team at sales@sylex.sk

* Specifications are subject to change without notice