



# WiPortal API

**Date: 27 Feb, 2025**  
**Document Revision: 2.02**



**BiPOM Electronics**

Telephone : 1-713-283-9970  
E-mail : [info@bipom.com](mailto:info@bipom.com)  
Web : [www.bipom.com](http://www.bipom.com)



© 2019 by BiPOM Electronics, Inc. All rights reserved.

WiPortal API. No part of this work may be reproduced in any manner without written permission of BiPOM Electronics.

All trademarked names in this manual are the property of respective owners.



## TABLE OF CONTENTS

<b>1. Introduction</b>	<b>4</b>
<b>2. API Commands</b>	<b>5</b>
2.1 Login and get Access Token	6
2.2 Reading Accounts List	7
2.3 Reading Account Info	9
2.4 Reading Users List	10
2.5 Reading User Info	13
2.6 Reading Devices List	14
2.7 Reading Device Info	16
2.8 Reading Tags List	19
2.9 Reading Tag Info	21
2.10 Reading Sensors List	22
2.11 Reading Assets List	24
2.12 Reading Events	25
2.13 Reading Alarms	27
2.14 Reading TriVibe Datasnapshot List	29
2.15 Reading TriVibe Datasnapshot Data	31
2.16 Reading Generic Datasnapshot List	33
2.17 Reading Generic Datasnapshot Data	35
2.18 Reading Tag Data	37
2.19 Reading Sensor Data	38
2.20 Reading GPS Data	40
2.21 Writing DO	42
2.22 Add/Edit Integration	43
2.23 Remove Integration	45
2.24 Find Integrations	46
<b>3. Examples - Exercise the API</b>	<b>47</b>



## 1. Introduction

This document describes WiPortal API 2.0.

This API should be used for following operations:

- Read users/accounts information which registered under account
- Read list of devices registered under account
- Read list of tags / sensors of device
- Read data records for specified tag / sensor or set of tags / sensors
- Read alarm records for specified device
- Read events records for specified device
- Read list of datasnapshots for specified device or account
- Read datasnapshot data for specified datasnapshot

Following versions of API could also include operations for inserting / updating and deleting system entities. But API Version 2.0 doesn't support them.

WiPortal API implemented as RESTful API. This means that:

- **read** operations will be issued with HTTP GET request.
- parameters will be passed as JSON object in request body or on query string
- result of read operation will be response where data put in response body in JSON format
- authentication/authorization implemented involving OAuth2 technology and uses security access tokens
- every client application must use APPKEY to be able issue requests
- only registered accounts will be able use Web API in their client applications
- application must to receive access token using login/password to issue other API requests
- access token will be issued for 1 day. Client application must refresh access token after this time to continue use Web API requests

This Web API can be used to extract data from WiPortal to 3<sup>rd</sup> party clients.



## 2. API Commands

Below will be described all supported commands of Web API for WiPortal.

Every command is exchange of HTTP request and response between Client and WiPortal.

WiPortal – it is web site [www.nanowipom.com](http://www.nanowipom.com)

Client – any software which issue commands to Web portal.

All communication must be over HTTPS channel. HTTP request will be dropped.

Base URI for every WiPortal API request started from <https://api.nanowipom.com/v1/>

V2 – version #2 of Web API.

All read operations pass its parameters using query string.

WiPortal API Version 2 supports following commands:

- <https://api.nanowipom.com/v1/Login>
- <https://api.nanowipom.com/v1/AccountsList>
- <https://api.nanowipom.com/v1/AccountInfo>
- <https://api.nanowipom.com/v1/UsersList>
- <https://api.nanowipom.com/v1/UserInfo>
- <https://api.nanowipom.com/v1/DevicesList>
- <https://api.nanowipom.com/v1/DeviceInfo>
- <https://api.nanowipom.com/v1/TagsList>
- <https://api.nanowipom.com/v1/TagInfo>
- <https://api.nanowipom.com/v1/SensorsList>
- <https://api.nanowipom.com/v1/AssetsList>
- <https://api.nanowipom.com/v1/Events>
- <https://api.nanowipom.com/v1/Alarms>
- <https://api.nanowipom.com/v1/Data>
- <https://api.nanowipom.com/v1/DatasnapshotsList>
- <https://api.nanowipom.com/v1/DatasnapshotData>
- <https://api.nanowipom.com/v1/TriVibeDatasnapshotsList>
- <https://api.nanowipom.com/v1/TriVibeDatasnapshotData>
- <https://api.nanowipom.com/v1/TagData>
- <https://api.nanowipom.com/v1/SensorData>
- <https://api.nanowipom.com/v1/GPSData>
- <https://api.nanowipom.com/v1/SetDO>
- <https://api.nanowipom.com/v1/Integrations>

Every command should use bearer token in Authorization HTTP header of every request. e.g.,

```
GET https://api.nanowipom.com/v1/DatasnapshotsList/229 HTTP/1.1
User-Agent: Fiddler
Host: api.nanowipom.com
Authorization: Bearer 6rrSAW7t0b7-
wvKapjggADI37tahHC0JVAXQ1_7gfn4orXmvJ5Nhtz53YFcBuy85CZZQTVknwV-aquuBwtLG_YVks09a0Gt6kvVoA-
DZ2VUH5m4kc7XeGFTj29IXTPKFP_Do5RIZm6zSu22CjTotv4Beh3ZMZAIBZfZUXKNwyschDiGa5p501xCPTdRdj75WmQqG
ygcaUJ5cRteXwuLmE80ZPUXmvY1qHr8_GdJNyquKws7kRyge44yFA5D-
010MQzpmXE9K61ya8CQCgJHI1ixWEN3FPKE4tARKe8sbl17_qageiQf2gvpxHN0G2wd09JI5wm6gXEgkZUCPqjRt6QCE8nn
Y0PFMG-N1H0WCK7Tgky12WDF5TTWAFth9FDowa7I1ykw13ddqSO_rdbcbdl1y1ke
Content-Length: 0
```

Those red text is bearer token returned by **Login** command. Token valid only 1 day. After this time



## 2.1 Login and get Access Token

### Description

This command must be issued in first turn to login user and receive Access Token.

### URI

<https://api.nanowipom.com/v1/Login>

### HTTP Method

POST

### Parameters

Nothing

### Data

grant\_type=password&username=[Login]&password=[Password]

[Login/Password](#) – Web Portal user login and password (which used to login into web portal)

This can be any user account: Integrator or regular user. This information will restrict access to system resources. So user will see only devices/tags/sensors/users etc. which he can see when logged to web portal.

### Response

```
{
  "access_token"      :string,
  "token_type"        :string "
  "expires_in"        :integer "
  "userName"          :string "
  ".issued"           :Date,
  ".expires"          :Date
}
```

[access\\_token](#) – a string which represent security access token. It must be included in Authorization HTTP header for every other request. It is OAuth2 bearer token, which means that it contains sensitive information. So all communication should be done over protected encrypted channel (HTTPS). This token has expiration date (1 day / 24 hours). During this time token can be used to execute requests to the server. After this time Client will receive error response. This means that Access Token must be re-issued every 24 hours. System can re-issue new token as often as it need (no need wait 24 hours).

[token\\_type](#) – it will be “bearer” in general.

[expires\\_in](#) – total seconds the Access Token will be valid

[userName](#) – Web Portal login, same as .username in request data

[.issued](#) – time when Access Token was issued by Authorization server.

[.expires](#) – time when Access Token will reach end of life and new token must be issued.



## 2.2 Reading Accounts List

### Description

This command request list of all sub accounts which belong to the account of logged user, or if logged user is Super Administrator, all accounts

### URI

<https://api.nanowipom.com/v1/AccountsList>

### HTTP Method

GET

### Parameters

**includeSubAcc** : **bool**

If set to TRUE (default assignment) then this parameter forces server include in result list also unlimited level of sub accounts which belong to logged user's account.

If set to FALSE then this parameter forces server return only 1-level subaccounts of account of logged user.

**activeOnly** : **bool**

If set to TRUE (default assignment) then this parameter forces server include in result list only active accounts.

If set to FALSE then this parameter forces server return as active so and inactive accounts.

### Data

Nothing

### Response

```
[
  {
    "id": integer,
    "active": bool,
    "name": string,
    "emails": string,
    "supportEmail": string,
    "address": string,
    "phones": string,
    "fax": string,
    "website": string,
    "createDate": date,
    "logoUrl": string,
    "integrator": bool,
    "parentIntegratorId": integer,
    "companyName": string,
    "country": string,
    "state": string,
    "city": string,
    "postalCode": string,
    "county": string,
    "customLicenseAgreement": bool
  },
  ...
]
```

Response is a list of objects which describes Account resource.

### Id

Unique integer number which identify **Account** resource in database



**Active**

Bool value which indicate if user is active or inactive

**Name**

Name of the account

**Emails**

Emails list separated by comma or semicolon for owner of this account

**SupportEmail**

Support Email

**Address**

Address of owner of the account

**Phones**

Phones list separated by comma or semicolon for owner of this account

**Fax**

Fax of owner of the account

**WebSite**

WebSite of owner of the account

**CreateDate**

Date and time when account was added to WiPortal.

**LogoUrl**

Web URL where to display Logo of owner of the account

**Integrator**

Bool value which indicate if the account is an integrator account who can have subaccounts

**ParentIntegratorId**

The parent account's ID is this is a subaccount, or 0 if this is not a subaccount

**CompanyName**

CompanyName of owner of the account

**Country/ State/ City/ County**

Country/ State/ City/ County location of the owner of the account

**PostalCode**

PostalCode of the owner of the account

**CustomLicenseAgreement**

Bool value which indicate if Custom License is agreed by the account.





## 2.3 Reading Account Info

### Description

This command request account information for specified account or subaccount.

### URI

<https://api.nanowipom.com/v1/AccountInfo>

### HTTP Method

GET

### Parameters

**Id** : integer

The account ID whose account information will be retrieved.

If Id is not set, server will return account information for the account of logged user.

If the Id is not logged user's account or not belong to any subaccount of logged user's account, server will return "null".

### Data

Nothing

### Response

```
{
  "id": integer,
  "active": bool,
  "name": string,
  "emails": string,
  "supportEmail": string,
  "address": string,
  "phones": string,
  "fax": string,
  "website": string,
  "createDate": date,
  "logoUrl": string,
  "integrator": bool,
  "parentIntegratorId": integer,
  "companyName": string,
  "country": string,
  "state": string,
  "city": string,
  "postalCode": string,
  "county": string,
  "customLicenseAgreement": bool
}
```

(Fields description is same as above.)



## 2.4 Reading Users List

### Description

This command request list of all users which belong to logged user's account or specified subaccount.

### URI

<https://api.nanowipom.com/v1/UsersList>

### HTTP Method

GET

### Parameters

**accountId:** *integer*

The account ID whose users will be list.

If omitted, we will list all users which belong to logged user's account.

If the accountId is not logged user's account or not belong to any subaccount of logged user's account, server will return empty list.

**includeSubAcc :** *bool*

If set to TRUE then this parameter forces server include in result list users under unlimited level of sub accounts which belong to specified account.

If set to FALSE then this parameter forces server return only direct users of specified account.

**activeOnly :** *bool*

If set to TRUE then this parameter forces server include in result list only active users.

If set to FALSE then this parameter forces server return as active so and inactive users.

### Data

Nothing

### Response

```
[
  {
    "Id": integer,
    "Active": bool,
    "Type": integer,
    "Login": string,
    "Name": string,
    "SurName": string,
    "Email": string,
    "Phone": string,
    "SkypeId": string,
    "AccountId": integer,
    "CreateDate": date,
    "IsLuaAccepted": bool,
    "LastLoggedAt": date,
    "TimeZone": string,
    "DeviceCreator": bool,
    "DisableLogin": bool,
    "Email2": string,
    "Phone2": string,
    "DefaultDataRange": string,
    "AccountReady": bool
  },
  ...
]
```

Response is a list of objects which describes User resource.



**Id**

Unique integer number which identify **User** resource in database

**Active**

Bool value which indicate if user is active or inactive

**Type**

Enum value, SuperAdministrator = 0, Administrator = 1, User = 2

**Login**

User's login name

**Name**

User's first name

**surname**

User's last name

**Email/Email2**

User's Email

**Phone/Phone**

User's Phone

**skypeId**

User's Skype ID

**AccountId**

Unique integer number which identify **Account** resource in database. This is user's account.

**CreateDate**

Date and time when user was added to WiPortal.

**IsLuaAccepted**

Bool value which indicate if user has accepted License Agreement.

**LastLoggedAt**

Date and time when user login the last time.

**TimeZone**

Timezone ID user locate at.

**DeviceCreator**

Bool value indicate if user has authority to create device.

**DisableLogin**

Bool value indicate if user is only a contact in system, who has no authority to login.

**DefaultDataRange**

A string indicate default time range when user retrieve data during a period. It starts with a number and ends with "d", "w" or "m" which means day, week, month, so "1d" means default time range is 1 day and "2m" means user's default time range is 2 months.



**AccountReady**

Bool value indicate if user's account is ready to use. (for online-pay user, if payment not finished or expired, AccountReady will be set false)



## 2.5 Reading User Info

### Description

This command request user information for specified user.

### URI

<https://api.nanowipom.com/v1/UserInfo>

### HTTP Method

GET

### Parameters

**Id** : integer

The user ID whose user information will be retrieved.

If Id is not set, server will return user information for the logged user.

If the Id is not logged user's ID or not belong to any subaccount of logged user's account, server will return "null".

### Data

Nothing

### Response

```
{
  "Id": integer,
  "Active": bool,
  "Type": integer,
  "Login": string,
  "Name": string,
  "SurName": string,
  "Email": string,
  "Phone": string,
  "SkypeId": string,
  "AccountId": integer,
  "CreateDate": date,
  "IsLuaAccepted": bool,
  "LastLoggedAt": date,
  "TimeZone": string,
  "DeviceCreator": bool,
  "DisableLogin": bool,
  "Email2": string,
  "Phone2": string,
  "DefaultDataRange": string,
  "AccountReady": bool
}
```

(Fields description is same as above.)



## 2.6 Reading Devices List

### Description

This command request list of all devices which belong to the account of logged user

### URI

<https://api.nanowipom.com/v1/DevicesList>

### HTTP Method

GET

### Parameters

**accounts** : **number array**

Command will collect only devices which belong to provided accounts. This parameter accepts comma delimited list of **Account Id** values. **Account Id** could be read with command **AccountsList**.

**includeSubAcc** : **bool**

If set to TRUE then this parameter forces server include in result list also devices which belong to all subaccounts. This will include subaccounts of logged user account.

If set to FALSE then this parameter forces server return only devices which belong to account of logged user.

**includeSharedDev** : **bool**

If set to TRUE then this parameter forces server include in result list also devices which shared with the account of logged user.

If set to FALSE then this parameter forces server return only devices which belong to account of logged user.

**activeOnly** : **bool**

If set to TRUE then this parameter forces server include in result list only active devices.

If set to FALSE then this parameter forces server return as active so and inactive devices.

### **Data**

Nothing

### **Response**

```
[
  {
    "Id"           :integer,
    "IsActive"    :bool,
    "Serial"      :string,
    "IPAddress"   :string,
    "AccountId"   :integer,
    "CreateAt"    :date,
    "CreatedbyUser" :integer,
    "deviceName"  :string
  },
  ...
]
```

Response is a list of objects which describes Device resource.



**Id**

Unique integer number which identify **Device** resource in database

**IsActive**

Bool value which indicate if device is active or inactive

**Serial**

Unique string identifier of device. It is usually serial number of device hardware

**IPAddress**

IPv4 address of device. Can be NULL.

**AccountId**

Unique integer number which identify **Account** resource in database. This is device's owner account.

**CreatedAt**

Date and time when device was added to WiPortal.

**CreatedbyUser**

Unique integer number which identify **User** resource in database. This is user who created the device on WiPortal.

**DeviceName**

Name of device



## 2.7 Reading Device Info

### Description

This command request device information for specified device.

### URI

<https://api.nanowipom.com/v1/DeviceInfo>

### HTTP Method

GET

### Parameters

**Id** : integer

The device ID whose device information will be retrieved.

If the device is not under logged user's account any subaccount of logged user's account, server will return "null".

### Data

Nothing

### Response

```

{
  "Id"                :integer,
  "AccountId"        :integer,
  "Serial"            :string,
  "IsActive"         :bool,
  "IPAddress"        :string,
  "ConfigData"       :string,
  "CreateAt"         :date,
  "CreatedbyUser"    :integer,
  "GpsColor"         :string,
  "Location"         :string,
  "IsSimplexDevice"  :bool,
  "DeviceExtraInfo"  :object*,
  "VtbDeviceExtraInfo" :object**,
  "DeviceName"       :string
}

```

#### Id

Unique integer number which identify **Device** resource in database

#### AccountId

Unique integer number which identify **Account** resource in database. This is device's owner account.

#### Serial

Unique string identifier of device. It is usually serial number of device hardware

#### IsActive

Bool value which indicate if device is active or inactive

#### IPAddress

IPv4 address of device. Can be NULL.

#### ConfigData

Config data of the device, (blank for now) .





**CreatedAt**

Date and time when device was added to WiPortal.

**CreatedbyUser**

This is user Id who created the device on WiPortal.

**Location**

Description of where the device is located.

**IsSimplexDevice**

Bool value which indicate if this is One-directional device (nnot be reachable from web portal side).

**DeviceExtraInfo**

This will be null for VTB device, otherwise it will be object include following infomations:

```
{
    "Name"                : string,
    "LastSessionTime"    : DateTime,
    "HardwareModel"      : string,
    "RtuNumber"          : int,
    "LoggerState"        : bool,
    "ModemModel"         : string,
    "FirmwareVersion"    : string,
    "Latitude"           : double,
    "Longitude"          : double,
    "GpsSpeed"           : float,
    "GpsHeading"         : string,
    "Satellite"          : int,
    "ValidGpsRecordsCount" : int,
    "InvalidGpsRecordsCount" : int,
    "HasNewAlarms"       : bool,
    "Time"               : DateTime,
    "CGTime"             : DateTime,
    "TagCount"           : int,
    "TagValuesCount"     : int,
    "AlarmsCount"        : int,
    "EventsCount"        : int,
    "isMetrologicDevice" : bool,
    "ModelName"          : string,
    "GPSTime"            : DateTime,
    "GPSColor"           : string,
    "PushFrequency"      : int,
    "LastTimeDataPushed" : DateTime,
    "isBadRTCTime"       : bool
}
```



**VtbDeviceExtraInfo**

This will be null for WRTU device, otherwise it will be object include following information:

```
{
    "Name" : string,
    "DeviceModel" : string,
    "LastSessionTime" : DateTime,
    "Location" : string,
    "Country" : string,
    "Employee" : string,
    "Customer" : string,
    "OSInformation" : string,
    "ServerVersion" : string,
    "UpTime" : long,
    "CanErrors" : long,
    "EthErrors" : long,
    "IOErrors" : long,
    "CAN1Enabled" : bool,
    "CAN2Enabled" : bool,
    "CAN3Enabled" : bool,
    "CAN4Enabled" : bool,
    "CAN1Sensors" : long,
    "CAN2Sensors" : long,
    "CAN3Sensors" : long,
    "CAN4Sensors" : long,
    "PushFrequency" : int,
    "LastTimeDataPushed" : DateTime
}
```

**DeviceName**

Name of the device



## 2.8 Reading Tags List

### Description

This command request list of all Tags under specified device

### URI

<https://api.nanowipom.com/v1/TagsList>

### HTTP Method

GET

### Parameters

**deviceId** : integer

The device ID under which tags will be list.

**activeOnly** : bool

If set to TRUE then this parameter forces server include in result list only active tags.

If set to FALSE then this parameter forces server return as active so and inactive tags.

### Data

Nothing

### Response

```
[
  {
    "Id": integer,
    "IsActive": bool,
    "DeviceId": integer,
    "Name": string,
    "Type": string,
    "Address": integer,
    "VirtualAddress": integer,
    "Units1": string,
    "Units2": string,
    "LogPeriod": integer,
    "MBRtu": integer,
    "MBType": string,
    "MBAddress": integer,
    "MBValueType": integer,
    "MBValueByteOrder": integer,
    "IsVMapEnabled": bool,
    "IsBMapEnabled": bool,
    "MaxLoggedAt": date,
    "RecordsNumber": integer,
    "VMapPresent": bool,
    "BMapPresent": bool
  },
  ...
]
```

Response is a list of objects which describes **PushDataTag** resource.

#### Id

Unique integer number which identify **PushDataTag** resource in database

#### IsActive

Bool value which indicate if tag is active or inactive



**Deviceld**

Device ID of the tag

**Name**

Name of the tag

**Type**

Type of the tag

**Address**

Address of the tag

**Deviceld**

Device ID of the tag

**Units1**

Units1 of the tag

**Units2**

Units2 of the tag

**LogPeriod**

LogPeriod of the tag

**MBRtu**

MBRtu of the tag

**MBType**

MBType of the tag

**MBAAddress**

MBAAddress of the tag

**MBValueType**

MBValueType of the tag

**MBValueByteOrder**

MBValueByteOrder of the tag

**IsVMapEnabled/ IsBMapEnabled**

Bool value which indicate if tag can setup ValueMap/ BitMap

**VMapPresent/BMapPresent**

Bool value which indicate if tag has ValueMap/ BitMap

**MaxLoggedAt**

MaxLoggedAt of the tag

**RecordsNumber**

RecordsNumber of the tag



## 2.9 Reading Tag Info

### Description

This command request tag information for specified Tag.

### URI

<https://api.nanowipom.com/v1/TagInfo>

### HTTP Method

GET

### Parameters

**Id** : integer

The Tag ID whose information will be retrieved.

If the tag is not belong to any device under logged user's account or belong to any subaccount of logged user's account, server will return "null".

### **Data**

Nothing

### **Response**

```

{
  "Id": integer,
  "IsActive": bool,
  "DeviceId": integer,
  "Name": string,
  "Type": string,
  "Address": integer,
  "VirtualAddress": integer,
  "Units1": string,
  "Units2": string,
  "LogPeriod": integer,
  "MBRtu": integer,
  "MBType": string,
  "MBAddress": integer,
  "MBValueType": integer,
  "MBValueByteOrder": integer,
  "IsVMapEnabled": bool,
  "IsBMapEnabled": bool,
  "MaxLoggedAt": date,
  "RecordsNumber": integer,
  "VMapPresent": bool,
  "BMapPresent": bool
}

```

(Fields description is same as above.)



## 2.10 Reading Sensors List

### Description

This command request list of all sensors under specified device, or, if no deviceId specified, under all devices under the login user's account.

### URI

<https://api.nanowipom.com/v1/SensorsList>

### HTTP Method

GET

### Parameters

**deviceId** : integer (optional)

The device ID under which sensors will be listed.

### Data

Nothing

### Response

```
[
  {
    "DeviceId":      integer,
    "AccountId":    integer,
    "Type":          string,
    "Tags":          array of Tag Info,
    "SensorName":   string,
    "RTU":           integer,
    "DevuceEUI":    string,
  },
  ...
]
```

Response is a list of objects which describes **Sensor** resource.

#### **Deviceld**

Device ID of the sensor

#### **Accountld**

Account ID of the sensor

#### **Type**

Type of the sensor

#### **Tags**

List of [Tag info]s which build the sensor.

#### **SensorName**

Name of the sensor

#### **RTU**

RTU of the sensor (this applies to TRIVIBE type sensors)



**DevuceEUI**

Device EUI of the sensor (this applies to only AIRVIBE type sensors)



## 2.11 Reading Assets List

### Description

This command request list of all assets under login user's account.

### URI

<https://api.nanowipom.com/v1/AssetsList>

### HTTP Method

GET

### Parameters

None

### Data

Nothing

### Response

```
[
  {
    "Id": integer,
    "AssetName": string,
    "AssetType": string,
    "AccountId": integer,
    "Devices": array of simple devices info,
    "Tags": array of simple tags info,
    "Sensors": array of simple sensors info
  },
  ...
]
```

Response is a list of objects which describes **Sensor** resource.

#### Id

ID of the asset

#### AssetName

Name of the asset

#### AssetType

Type of the asset

#### AccountId

Account ID of the asset

#### Devices

Devices list under the asset. (Each device info contains only ID and Name)

#### Tags

Tags list under the asset. (Each tag info contains only device ID, tag ID and tag Name)

#### Sensors

Sensors list under the asset. (Each sensor info contains only device ID, sensor hash and sensor Name)





## 2.12 Reading Events

### Description

This command request list of Events for specified device for specified period.

### URI

<https://api.nanowipom.com/v1/Events>

### HTTP Method

GET

### Parameters

**deviceId** : *integer*

The device ID for which Events will be list.

**startTime** : *datetime* (in MM-dd-yyyy hh:mm:ss format)

Specify the start time from when to list events. If not set, list will begin with the earliest event record.

**endTime** : *datetime* (in MM-dd-yyyy hh:mm:ss format)

Specify the end time to when to list events. If not set, list will end with the latest event record.

### Data

Nothing

### Response

```
[
  {
    "Id": integer,
    "DeviceId": integer,
    "LogId": integer,
    "LoggedAt": date,
    "EventType": integer,
    "EventId": integer,
    "ErrorCode": integer
  },
  ...
]
```

Response is a list of objects which describes **Event** resource.

#### Id

Unique integer number which identify **Event** resource in database

#### DeviceId

Device ID of the event

#### LogId

LogId of the event

#### LoggedAt

Date and time when event was logged to system

#### EventType



Event type code

**EventId**

Event Id of the event

**ErrorCode**

Error Code of the event



## 2.13 Reading Alarms

### Description

This command request list of Alarms for specified device for specified period.

### URI

<https://api.nanowipom.com/v1/Alarms>

### HTTP Method

GET

### Parameters

**deviceId** : integer

The device ID for which Alarms will be list.

**startTime** : datetime (in MM-dd-yyyyThh:mm:ss format)

Specify the start time from when to list alarms. If not set, list will begin with the earliest alarm record.

**endTime** : datetime (in MM-dd-yyyyThh:mm:ss format)

Specify the end time to when to list alarms. If not set, list will end with the latest alarm record.

**activeOnly** : bool

If set to TRUE then this parameter forces server include in result list only alarms for active tags.

If set to FALSE then this parameter forces server return list alarms for as active tags so and inactive tags.

### Data

Nothing

### Response

```
[
  {
    "Id": integer,
    "DeviceId": integer,
    "TagId": integer,
    "LogId": integer,
    "LoggedAt": date,
    "RawValue": float,
    "CalcValue": float,
    "AlarmCondition": integer
  },
  ...
]
```

Response is a list of objects which describes **Alarm** resource.

#### Id

Unique integer number which identify **Alarm** resource in database

#### DeviceId

Device ID who trigger the alarm

#### TagId

DbTagId who trigger the alarm



**LogId**

LogId of the alarm

**LoggedAt**

Date and time when alarm was logged to system

**RawValue**

Raw value of the alarm

**CalcValue**

Calculated value of the alarm

**AlarmCondition**

Alarm Condition of the alarm



## 2.14 Reading TriVibe Datasnapshot List

### Description

This command request list of TriVibe Datasnapshots for specified device.

### URI

<https://api.nanowipom.com/v1/TriVibeDatasnapshotsList>

### HTTP Method

GET

### Parameters

**deviceId** : integer

The device ID for which TriVibe Data snapshots will be list.

### Data

Nothing

### Response

```
[
  {
    "Id": integer,
    "DeviceId": integer,
    "SensorId": integer,
    "TagId": integer,
    "UserId": integer,
    "Samples": integer,
    "TimePeriod": integer,
    "SensorPosition": integer,
    "Axis": integer,
    "SensorType": integer,
    "SensorSerial": integer,
    "Name": string,
    "RecTime": date
  },
  ...
]
```

Response is a list of objects which describes **TriVibe Datasnapshot** resource.

#### Id

Unique integer number which identify **TriVibe Datasnapshot** resource in database

#### DeviceId

Device ID who the snapshot belong to.

#### TagId

DbTagId who the snapshot belong to.

#### SensorId

vtb Sensor Id who the snapshot belong to.

#### UserId

User who record the snapshot.



**Samples**

Samples count of the snapshot.

**TimePeriod**

TimePeriod of the snapshot.

**SensorPosition**

SensorPosition of the snapshot.

**Axis**

Axis of the snapshot.

**SensorType**

SensorType of the snapshot.

**SensorSerial**

SensorSerial of the snapshot.

**Name**

Name of the snapshot.

**RecTime**

Date and time when the snapshot was record to system.



## 2.15 Reading TriVibe Datasnapshot Data

### Description

This command request data for specified TriVibe Datasnapshot.

### URI

<https://api.nanowipom.com/v1/TriVibeDatasnapshotData>

### HTTP Method

GET

### Parameters

**Id** : integer

The TriVibe Data snapshot ID whose data will be retrieved.

**window** : string

Window function which is used to convert raw data of the snapshot to Acceleration and Velocity. This should be one of following window functions name:

- "HanningWindow",
- "InverseHanningWindow",
- "HammingWindow",
- "InverseHammingWindow".

If not set, WiPortal will not use any window function when convert raw data.

**highPass** : double

The highPass Frequency Filter used to convert raw data of the snapshot to Acceleration and Velocity. It will be 1.5 by default if not set.

**lowPass** : double

The lowPass Frequency Filter used to convert raw data of the snapshot to Acceleration and Velocity. It will be 2000 by default if not set.

### **Data**

Nothing

### **Response**

```

{
  "Id": integer,
  "DeviceId": integer,
  "SensorId": integer,
  "TagId": integer,
  "UserId": integer,
  "Samples": integer,
  "TimePeriod": integer,
  "SensorPosition": integer,
  "Axis": integer,
  "SensorType": integer,
  "SensorSerial": integer,
  "Name": string,
  "RecTime": date,
  "TimewaveData": array of (array of double),
  "FftAccData": array of (array of double),
  "FftVelData": array of (array of double),
  "RawData": array of (array of double)
}

```

Fields description is same as above except following:



**TimewaveData**

If Axis is 9 (which means ALL\_AXIS), this field will contain 3 double array for Timewave Data for each axis. Otherwise one double array for specified axis

**FftAccData**

If Axis is 9 (which means ALL\_AXIS), this field will contain 3 double array for fft Acceleration Data for each axis. Otherwise one double array for specified axis

**FftVelData**

If Axis is 9 (which means ALL\_AXIS), this field will contain 3 double array for fft Velocity\_Data for each axis. Otherwise one double array for specified axis

**RawData**

If Axis is 9 (which means ALL\_AXIS), this field will contain 3 double array for Raw Data for each axis. Otherwise one double array for specified axis





## 2.16 Reading Generic Datasnapshot List

### Description

This command request list of Generic Datasnapshots for specified device.

### URI

<https://api.nanowipom.com/v1/DatasnapshotsList>

### HTTP Method

GET

### Parameters

**deviceId** : integer

The device ID for which Generic Data snapshots will be list.

**tagId** : integer (optional)

The tag ID, if specified, for whose Generic Data snapshots will be list.

### Data

Nothing

### Response

```
[
  {
    "Id": integer,
    "DeviceId": integer,
    "TagId": integer,
    "Name": string,
    "XUnits": string,
    "YUnits": string,
    "Timestamp": date
  },
  ...
]
```

Response is a list of objects which describes Generic **Datasnapshot** resource.

#### Id

Unique integer number which identify Generic **Datasnapshot** resource in database

#### DeviceId

Device ID who the snapshot belong to.

#### TagId

DbTagId who the snapshot belong to.

#### Name

Name of the snapshot.

#### XUnits

Units of the snapshot on X-axis.

#### YUnits



Units of the snapshot on Y-axis.

**Timestamp**

Date and time when the snapshot was record to system.



## 2.17 Reading Generic Datasnapshot Data

### Description

This command request data for specified Generic Datasnapshot.

### URI

<https://api.nanowipom.com/v1/DatasnapshotData>

### HTTP Method

GET

### Parameters

**Id** : integer

The Generic Data snapshot ID whose data will be retrieved.

**window** : string

Window function which is used to convert raw data of the snapshot to Acceleration and Velocity. This should be one of following window functions name:

- "HanningWindow",
- "InverseHanningWindow",
- "HammingWindow",
- "InverseHammingWindow".

If not set, WiPortal will not use any window function when convert raw data.

**highPass** : double

The highPass Frequency Filter used to convert raw data of the snapshot to Acceleration and Velocity. It will be 1.5 by default if not set.

**lowPass** : double

The lowPass Frequency Filter used to convert raw data of the snapshot to Acceleration and Velocity. It will be 2000 by default if not set.

### Data

Nothing

### Response

```
{
  "Id": integer,
  "DeviceId": integer,
  "TagId": integer,
  "Name": string,
  "XUnits": string,
  "YUnits": string,
  "Timestamp": date,
  "XData": array of double,
  "YData": array of double,
  "AccelData": array of double,
  "AccelFFTData": array of double,
  "VelocityData": array of double,
  "VelocityFFTData": array of double
}
```

Fields description is same as above except following:

#### XData

Double array for micro-seconds value (start from 0) on X-axis.



**YData**

Double array for raw snapshot values on Y-axis.

**AccelData**

Double array for Acceleration Data represented by YData.

**AccelFFTData**

Double array for fft Acceleration Data calculated from Acceleration Data.

**VelocityData**

Double array for velocity Data converted through Acceleration Data.

**VelocityFFTData**

Double array for fft velocity Data calculated from velocity Data.



## 2.18 Reading Tag Data

### Description

This command request list of Values for specified Tag for specified period.

### URI

<https://api.nanowipom.com/v1/TagData>

### HTTP Method

GET

### Parameters

**tagId** : integer

The tag ID for whose tag values will be list.

**startTime** : *datetime* (in MM-dd-yyyyThh:mm:ss format)

Specify the start time from when to list tag values. If not set, list will begin with the earliest value record.

**endTime** : *datetime* (in MM-dd-yyyyThh:mm:ss format)

Specify the end time to when to list tag values. If not set, list will end with the latest value record.

### Data

Nothing

### Response

```
[
  {
    "Id": integer,
    "LoggedAt": date,
    "RawValue": float,
    "CalcValue": float,
    "TextValue": string
  },
  ...
]
```

Response is a list of objects which describes **TagValue** resource.

#### Id

Unique integer number which identify **TagValue** resource in database

#### LoggedAt

Date and time when the value was logged to system

#### RawValue

Raw Value

#### CalcValue

Calculated value

#### TextValue

Text Value (if applicable)



## 2.19 Reading Sensor Data

### Description

This command request list of Values for specified Sensor for specified period.

### URI

<https://api.nanowipom.com/v1/SensorData>

### HTTP Method

GET

### Parameters

**sensorId** : integer

The sensor ID for whose values will be list.

**startTime** : datetime (in MM-dd-yyyy hh:mm:ss format)

Specify the start time from when to list sensor values. If not set, list will begin with the earliest value record.

**endTime** : datetime (in MM-dd-yyyy hh:mm:ss format)

Specify the end time to when to list sensor values. If not set, list will end with the latest value record.

### Data

Nothing

### Response

```
[
  {
    "Id"           : integer,
    "RecTime"     : date,
    "A1A"         : float,
    "A2A"         : float,
    "A3A"         : float,
    "A1V"         : float,
    "A2V"         : float,
    "A3V"         : float,
    "T"           : float,
    "AC"          : integer,
    "DC"          : integer,
    "AA"          : float,
    "AD"          : float,
    "IW"          : float,
    "ErrCode"    : integer
  }
  ...
]
```

Response is a list of objects which describes **TagValue** resource.

#### Id

Unique integer number which identify **Sensor Data** resource in database

#### RecTime

Date and time when the value was logged to system

#### A1A

Value of A1 Acceleration



**A2A**

Value of A2 Acceleration

**A3A**

Value of A3 Acceleration

**A1V**

Value of A1 Velocity

**A2V**

Value of A2 Velocity

**A3V**

Value of A3 velocity

**T**

Value of Temperature

**AC**

Value of Impact Counts

**DC**

Value of Higher Amplitude Impact Count

**AA**

Value of Avg. Alert

**AD**

Value of Higher Amplitude Impact Level

**IW**

Value of Impact Window

**ErrCode**

Value of Error Code



## 2.20 Reading GPS Data

### Description

This command request list of associated GPS trace data for specified device for specified period.

### URI

<https://api.nanowipom.com/v1/GPSData>

### HTTP Method

GET

### Parameters

**deviceId** : integer

The device ID for whose GPS trace data will be list.

**startTime** : *datetime* (in MM-dd-yyyy hh:mm:ss format)

Specify the start time from when to list GPS trace data. If not set, list will begin with the earliest value record.

**endTime** : *datetime* (in MM-dd-yyyy hh:mm:ss format)

Specify the end time to when to list GPS trace data. If not set, list will end with the latest value record.

### Data

Nothing

### Response

```
[
  {
    "Id" : integer,
    "SessionTime" : date,
    "Latitude" : double,
    "Longitude" : double,
    "GpsSpeed" : float,
    "GpsHeading" : string,
    "Satellite" : integer,
    "LatitudeDegree" : string,
    "LongitudeDegree": string
  }
  ...
]
```

Response is a list of objects which describes **TagValue** resource.

#### Id

Unique integer number which identify **GPS Data** resource in database

#### SessionTime

Date and time when the GPS data was logged to system

#### Latitude

Raw Latitude value of GPS date

#### Longitude

Raw Longitude value of GPS date





**GpsSpeed**

GpsSpeed value of GPS date

**GpsHeading**

GpsHeading of GPS date

**Satellite**

Satellite count value of GPS date

**LatitudeDegree**

Degree format Latitude value of GPS date

**LongitudeDegree**

Degree format Longitude value of GPS date



## 2.21 Writing DO

### Description

This command requests to write DO on the remote device.

### URI

<https://api.nanowipom.com/v1/SetDO/<deviceId>>

### HTTP Method

POST

### Parameters

**deviceId** : integer

The device ID which should be connected.

### Data

```
{
  "state": [
    {
      "number": integer,
      "value" integer
    }
    ...
  ]
}
```

Request is an object with “state” array.  
Every item of “state” array specify desired state of individual DO.

### number

DO number on device (1 for DO1, 2 for DO2,...)

### value

Desired DO state. 0 – set DO LOW state, 1 – set DO HIGH state.

### Response

```
{"State":boolean, "Message":""}
```

### State

Boolean value, true – DO written successfully, false – DO was not written.

### Message

Error message if request failed. Empty string if request is successful.



## 2.22 Add/Edit Integration

### Description

This command add a new Integration item or update a Integration item for login user's account.

### URI

<https://api.nanowipom.com/v1/Integrations>

### HTTP Method

POST

### Parameters

Nothing

### Data

JSON data that represent an **Integration** item, e.g.

```
{
  "Name": "NewIntegration",
  "AccountId": 71,
  "Type": "HTTP",
  "URL": "http://localhost:9000",
  "Login": null,
  "Password": null,
  "Topic": null,
  "Details": [{
    "Event": "NewDataReceived",
    "DeviceId": 315
  }],
  ...
}
```

### Name

Name of the Integration (should be unique under user's account).

**Note**, If this is a new **Integration** name, then we will add this new **Integration**, Otherwise, update the existing one.

### AccountId

Account who owns the Integration (can be user's account ID or sub account ID).

### Type

"HTTP" or "MQTT" (0 Or 1 is also applicable).

### URL

URL where will be made POST HTTP request. Or URL of MQTT broker.

### Login

The user name used for MQTT connection. Or null if type is "HTTP".

### Password

The user password used for MQTT connection. Or null if type is "HTTP".

### Topic

The topic of message published through MQTT. Or null if type is "HTTP".

### Details



Array of devices and events which we will monitor to issue integration request.

**Event**

"NewDataReceived" or 0 - new data received (pushed from remote device)  
"NewAlarmReceived" or 1 - new alarms received (pushed from remote device)  
"NewAlarmGenerated" or 2 - new alarm generated by web portal system  
"NewNotificationGenerated" or 3 - new notification generated by web portal system

**DeviceId**

The Monitored device ID.

**Response**

```
{"State":boolean, "Message":""}
```

**State**

Boolean value, true - add/edit integration successfully, false - error occurred.

**Message**

Error message or "OK".



## 2.23 Remove Integration

### Description

This command delete an Integration item from login user's account.

### URI

<https://api.nanowipom.com/v1/Integrations>

### HTTP Method

DELETE

### Parameters

**name** : string

The exact Integration name to be deleted.

### Data

Nothing

### Response

```
{"State":boolean, "Message":""}
```

### State

Boolean value, true - delete Integration successfully, false - error occurred.

### Message

Error message or "OK".



## 2.24 Find Integrations

### Description

This command search for Integrations under login user's account.

### URI

<https://api.nanowipom.com/v1/Integrations>

### HTTP Method

GET

### Parameters

**pattern** : string

The Integration name pattern to be searched.

\* All Integrations whose name starts with this pattern will be list.

### Data

Nothing

### Response

Response is a list of Integration items, e.g:

```
[
  {
    "Name": "MyIntegration",
    "AccountId": 1,
    "Type": "HTTP",
    "URL": "http://localhost:9000",
    "Login": null,
    "Password": null,
    "Topic": null,
    "Details": [
      {
        "Event": "NewDataReceived",
        "DeviceId": 297
      },
      ...
    ]
  }
  ...
]
```

Refer the description of "Add/Edit Integration".

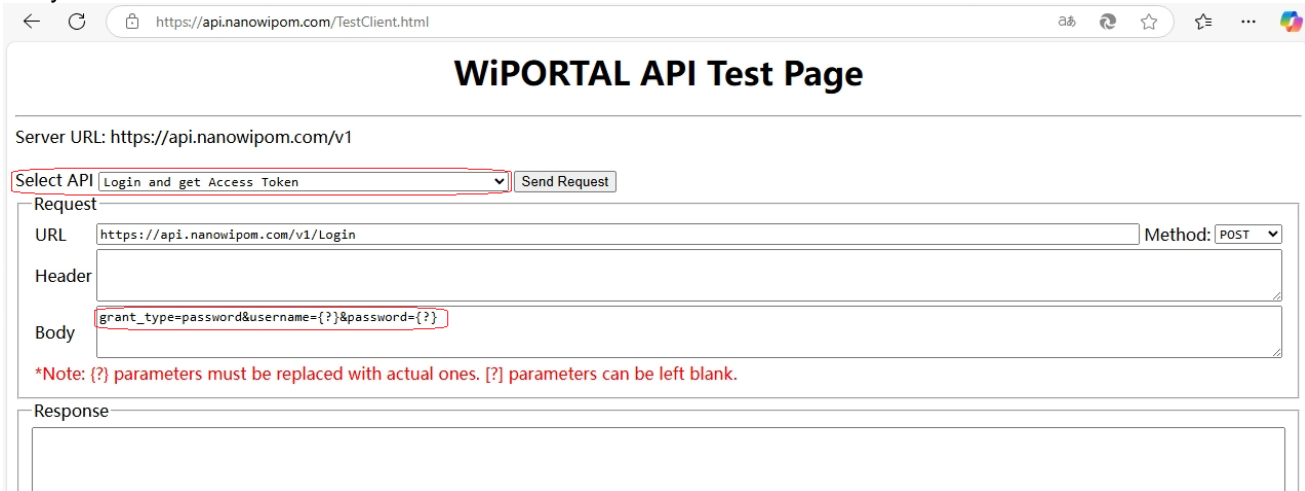
### 3. Examples - Exercise the API

User can use following link to exercise Web APIs for WiPortal.  
<https://api.nanowipom.com/TestClient.html>

**Example #1** - Request sensors on the device.

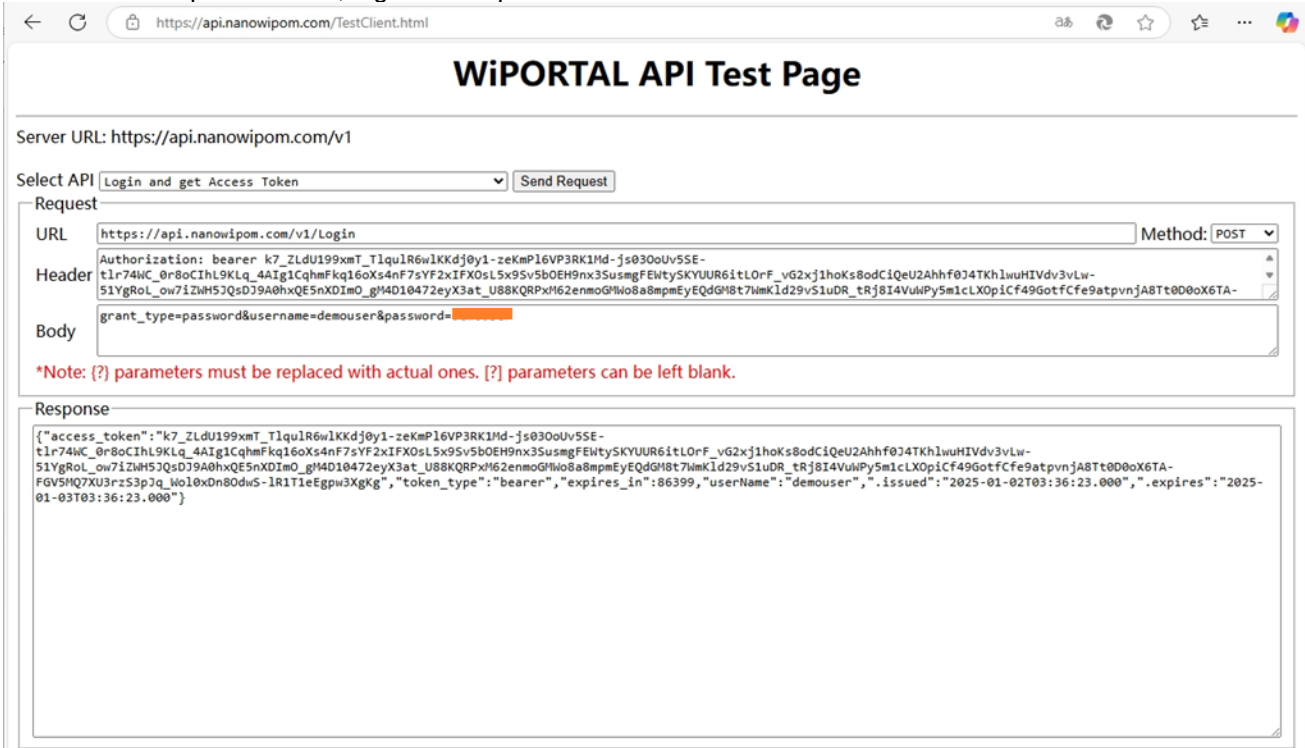
Step 1: Login at first.

Select "Login and get Access Token" API and type your user name and password to replace 2 {?} masks in Body area.



e.g. type "grant\_type=password&username=demouser&password=demopassword".

Click "Send Request" button, login will be processed and access token will be auto-filled into Header area.



Step 2: List all devices under login account.

Select "Reading Devices List" API and keep all parameters as default [?], Click "Send Request" button. All demouser account devices info in JSON format will be list in Response area.

Server URL: <https://api.nanowipom.com/v1>

Select API Reading Devices List Send Request

Request

URL:  Method: GET

Header: Authorization: bearer k7\_ZldU199xmT\_TlqlR6w1KKdJ0y1-zeKmp16VP3RK1Md-js030oUv5SE-tlr74mC\_0r8oCIh9Klq\_4AIg1CqhmFkq16oXs4nF7sYF2xIFX0sL5x9Sv5b0EH9nx3SusmgFEntySKYUUR6itL0rF\_vG2xj1hoKs8odCiQeU2Ahhf0J4TKhluuHIVdv3vLw-51YgRoL\_ow7iZiNH5QsDj9A0hXQE5nXDImo\_gM4D10472eyX3at\_U88KQRPxM62enmoGfWo8a8mpmEyEQdG8t7WmKld29v51uDR\_trj8I4VvWPy5m1cLXOpICf49GofCfe9atpvnjA8Tt00oX6TA-

Body: grant\_type=password&username=demouser&password=

**\*Note: (?) parameters must be replaced with actual ones. (?) parameters can be left blank.**

Response

```
[{"Id":197,"AccountId":1,"Serial":"M819D4832N-55550001","IsActive":true,"IPAddress":"","CreateAt":"2016-01-27T15:48:36.284","CreatedByUser":28,"IsCGVtCom":false,"DeviceName":"CG9104-900NT"}, {"Id":358,"AccountId":1,"Serial":"XJ29HAD021-936218","IsActive":true,"IPAddress":"","CreateAt":"2018-12-17T11:55:51.581","CreatedByUser":28,"IsCGVtCom":false,"DeviceName":"Monnit ALTA Test Unit 2"}, {"Id":1093,"AccountId":1,"Serial":"05D02D31-35384152-51214842","IsActive":true,"IPAddress":"166.130.0.14","CreateAt":"2019-02-17T16:16:35.184","CreatedByUser":28,"IsCGVtCom":false,"DeviceName":"WMCMS Marathon Compactor Test"}, {"Id":1149,"AccountId":1,"Serial":"TEST MOMENTARY BUTTON","IsActive":true,"IPAddress":"127.0.0.1","CreateAt":"2023-11-29T13:35:59.651","CreatedByUser":28,"IsCGVtCom":false,"DeviceName":"Not Set"}, {"Id":1194,"AccountId":1,"Serial":"ZW108185120410","IsActive":true,"IPAddress":"","CreateAt":"2024-03-11T01:06:13.203","CreatedByUser":28,"IsCGVtCom":false,"DeviceName":"WiPOM [Linux] FX30"}, {"Id":1230,"AccountId":1,"Serial":"KM4AKB5015","IsActive":true,"IPAddress":"10.11.0.11","CreateAt":"2024-08-15T15:23:02.091","CreatedByUser":28,"IsCGVtCom":false,"DeviceName":"Cloudgate Mini"}, {"Id":1208,"AccountId":1,"Serial":"TEST1","IsActive":true,"IPAddress":"","CreateAt":"2024-05-23T09:32:32.167","CreatedByUser":28,"IsCGVtCom":false,"DeviceName":"Not Set"}]
```

You can also fill parameters to list only required devices. E.g. <https://api.nanowipom.com/v1/DevicesList?accounts=1&includeSubAcc=false&includeSharedDev=false&activeOnly=false>. See [section 2.6](#) for parameter details.

Step 3: Request all sensors on the device whose Id is 1230. Select "Reading Sensors List" API and type 1230 as deviceId parameter, Click "Send Request" button. All sensors under device 1230 will be list in Response area, in JSON format.

Server URL: <https://api.nanowipom.com/v1>

Select API Reading Sensors List Send Request

Request

URL:  Method: GET

Header: Authorization: bearer k7\_ZldU199xmT\_TlqlR6w1KKdJ0y1-zeKmp16VP3RK1Md-js030oUv5SE-tlr74mC\_0r8oCIh9Klq\_4AIg1CqhmFkq16oXs4nF7sYF2xIFX0sL5x9Sv5b0EH9nx3SusmgFEntySKYUUR6itL0rF\_vG2xj1hoKs8odCiQeU2Ahhf0J4TKhluuHIVdv3vLw-51YgRoL\_ow7iZiNH5QsDj9A0hXQE5nXDImo\_gM4D10472eyX3at\_U88KQRPxM62enmoGfWo8a8mpmEyEQdG8t7WmKld29v51uDR\_trj8I4VvWPy5m1cLXOpICf49GofCfe9atpvnjA8Tt00oX6TA-

Body: grant\_type=password&username=demouser&password=

**\*Note: (?) parameters must be replaced with actual ones. (?) parameters can be left blank.**

Response

```
[{"DeviceId":1230,"AccountId":1,"Type":"AIRVIBE","Tags":[{"Id":1133783,"IsActive":true,"DeviceId":1230,"Name":"11111111111111 - Acceleration Axis 1","Type":"DS","Address":"","VirtualAddress":"","Units1":"g","Units2":"","LogPeriod":0,"MBRtu":0,"MBType":null,"MBAddress":0,"MBValueType":null,"MBValueByteOrder":null,"IsVMapEnabled":false,"IsBMapEnabled":false,"MaxLoggedAt":null,"RecordsNumber":0,"VMapPresent":false,"BMapPresent":false}, {"Id":1133784,"IsActive":true,"DeviceId":1230,"Name":"11111111111111 - Acceleration Axis 2","Type":"DS","Address":"","VirtualAddress":"","Units1":"g","Units2":"","LogPeriod":0,"MBRtu":0,"MBType":null,"MBAddress":0,"MBValueType":null,"MBValueByteOrder":null,"IsVMapEnabled":false,"IsBMapEnabled":false,"MaxLoggedAt":null,"RecordsNumber":0,"VMapPresent":false,"BMapPresent":false}, {"Id":1133785,"IsActive":true,"DeviceId":1230,"Name":"11111111111111 - Acceleration Axis 3","Type":"DS","Address":"","VirtualAddress":"","Units1":"g","Units2":"","LogPeriod":0,"MBRtu":0,"MBType":null,"MBAddress":0,"MBValueType":null,"MBValueByteOrder":null,"IsVMapEnabled":false,"IsBMapEnabled":false,"MaxLoggedAt":null,"RecordsNumber":0,"VMapPresent":false,"BMapPresent":false}, {"Id":1133786,"IsActive":true,"DeviceId":1230,"Name":"11111111111111 - Temperature","Type":"HR","Address":"40200","VirtualAddress":"","Units1":"F","Units2":"","LogPeriod":60,"MBRtu":0,"MBType":null,"MBAddress":0,"MBValueType":null,"MBValueByteOrder":null,"IsVMapEnabled":false,"IsBMapEnabled":false,"MaxLoggedAt":null,"RecordsNumber":0,"VMapPresent":false,"BMapPresent":false}, {"Id":1133787,"IsActive":true,"DeviceId":1230,"Name":"11111111111111 - Humidity","Type":"HR","Address":"40201","VirtualAddress":"","Units1":"%",Units2":"","LogPeriod":60,"MBRtu":0,"MBType":null,"MBAddress":0,"MBValueType":null,"MBValueByteOrder":null,"IsVMapEnabled":false,"IsBMapEnabled":false,"MaxLoggedAt":null,"RecordsNumber":0,"VMapPresent":false,"BMapPresent":false}, {"Id":1133788,"IsActive":true,"DeviceId":1230,"Name":"11111111111111 - Battery","Type":"HR","Address":"40202","VirtualAddress":"","Units1":"%",Units2":"","LogPeriod":60,"MBRtu":0,"MBType":null,"MBAddress":0,"MBValueType":null,"MBValueByteOrder":null,"IsVMapEnabled":false,"IsBMapEnabled":false,"MaxLoggedAt":null,"RecordsNumber":0,"VMapPresent":false,"BMapPresent":false}, {"Id":1133789,"IsActive":true,"DeviceId":1230,"Name":"11111111111111 - SupplyVoltage","Type":"HR","Address":"40203","VirtualAddress":"","Units1":"V","Units2":"","LogPeriod":60,"MBRtu":0,"MBType":null,"MBAddress":0,"MBValueType":null,"MBValueByteOrder":null,"IsVMapEnabled":false,"IsBMapEnabled":false,"MaxLoggedAt":null,"RecordsNumber":0,"VMapPresent":false,"BMapPresent":false}]]
```



