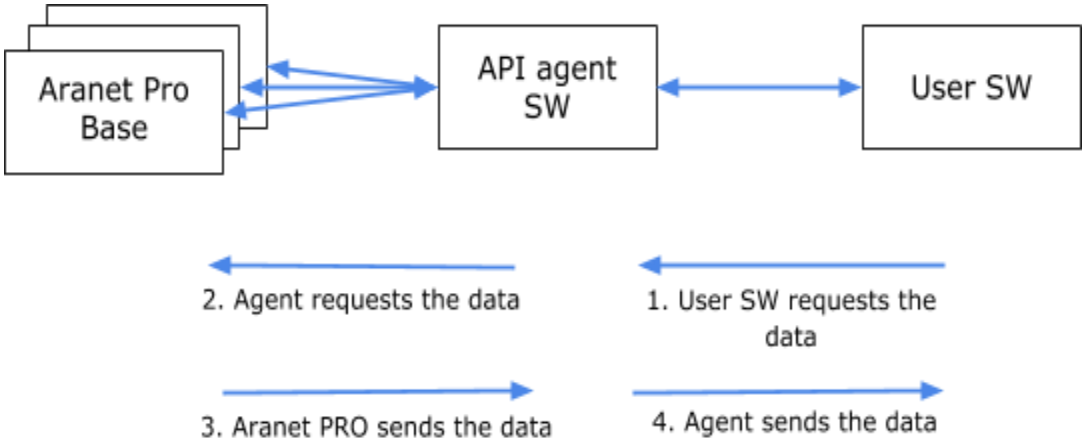
Aranet PRO API description

Overview

Aranet PRO does not have a direct API but uses a separate application *aranet-agent* that acts as an API provider for one or several Aranet PRO devices. A simplified picture is shown below.



To limit the load of Aranet Pro Base stations, *aranet-agent* stores the gathered data in cache. The latest reading data from particular sensor are updated when new request from User SW arrives and data in cache are more than 60 seconds old.

The *aranet-agent* application can be run under 64-bit Windows, Linux or macOS and can be run either on the same server/workstation as the User SW or it can run on a seperate server/workstation.
It is intended to use *aranet-agent* as a long-running backend service application.

aranet-agent can provide data over HTTP, HTTPS, SNMPv2 and SNMPv3 protocols.
The default format for HTTP responses is `application/senml+json` (https://datatracker.ietf.org/doc/draft-ietf-core-senml).

Also custom `application/json` format responses are supported.

MIB file *saf-AranetSensors.mib* for SNMP managers is included in delivery package along with *aranet-agent*

Configuration options

aranet-agent stores its configuration data in the *config.yml* file.

The configuration file contains several sections:

- The `aranet` section specifies the IP address(es) and user credentials of the Aranet PRO(s).
- The `http` section contains options for the HTTP and HTTPS provider.
- The `snmp` section holds options for the SNMP provider.
- The `autoconfig` section is used to save some runtime data and this section should not be edited by a user.

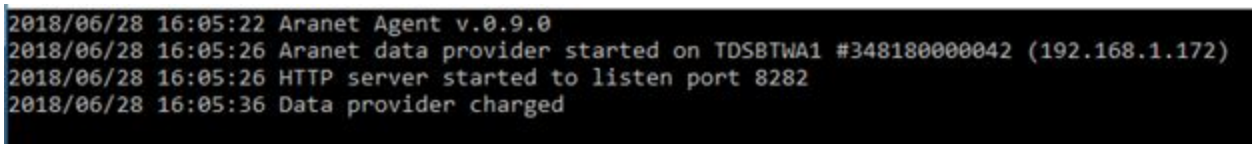
aranet-agent needs permissions to read and write the *config.yml* file.

Below you'll find an example of the *config.yml* file.

```
aranet:
- url: 192.168.206.100
  user: agent
  password: <Enter password here>
  hash: 47db63b09a2128e0caf408c87f5bef8a44ba17c0583236896f5c81f5581aaf6b
- url: 192.168.1.172
  user: agent
  password: <Enter password here>
  hash: b68b84e203072c41ce62318d96288a308c81d4a3dfca813d22bd59911d959a25
http:
  status: "on"
  port: 8080
  ssl:
      status: disabled
      comment: To secure traffic provide Certificate and Private Key files OR turn
the Let's Encrypt automated mechanism on and fill the domain names to serve
      certfile: ""
      keyfile: ""
      autocert: disabled
    autodomains: [mydomain.com]
  templates:
  - name: example
      fields:
      name:
      senml: "n"
      sensor:
      senml: bn
      time:
      senml: bt
      type: string
      unit:
```

```
      senml: u
      value:
      senml: v
      type: string
snmp:
  status: "off"
  port: 161
  v2: enabled
  v3: enabled
  community: public
  node:
      contact: <Enter the contact person here>
      name: <Enter the node name here>
      location: <Enter physical location of this node here>
  users:
  - name: <SNMPv3User>
      pass: <Enter the Pass Phrase here>
      authtype: NoAuth
      privtype: NoPriv
      comment: 'Possible values -> authtype: SHA | MD5 | NoAuth, privtype: DES |
AES | NoPriv, pass length >= 8 characters'
      hashl: ""
  - name: <SNMPv3User2>
      pass: <Enter the Pass Phrase here>
      authtype: NoAuth
    privtype: NoPriv
      hashl: ""
autoconfig:
  comment: The following values are assigned and changed automatically!
  snmpv3:
      engineID: 00001d9341e58164574dc7a8
      engineBoots: 4
```

Once you run the *aranet-agent*, it will start up and establish connection(s) with Aranet PRO based on the data specified in *config.yml* file.



To test and discover various options open a web browser and enter the following url: "http://127.0.0.1:8080" (port can be changed in the c*onfig.yml* file). It will open a help page as shown below that can be used as guidance on how to use the different request and filter parameters.

```
Aranet data agent provides data from AranetPro in SenML format:
       /last - last measurement and technical data from all sensors
       /last/xxx - last measurement and technical data from the sensor with ID xxx
       /history - measurement data for requested period from all sensors
       /history/xxx - measurement data for requested period from the sensor with ID xxx
       /telemetry - technical data for requested period from all sensors
       /telemetry/xxx - technical data for requested period from the sensor with ID xxx
       /version - version of Aranet data agent software

Possible parameters are:
       - filter by data kind
           m[easure]=
              t | temperature - temperature
              h | humidity - humidity
              c | co2 - carbon dioxide
              ap | atmpressure | atmosphericpressure - atmospheric pressure
              v | ev | voltage - electric voltage
              ec | current - electric current
              w | weight - weight
              uw | untared - untared weight
              vwc | volumetricwatercontent - volumetric water content of soil / substrate
              pec | poreec - pore water electrical conductivity
              bec | bulkec - bulk electrical conductivity
              dp | dielectricpermittivity - Dielectric permittivity
              pp | ppfd - photosynthetic photon flux density
              r | rssi - received signal strength indicator
              b | battery - battery charge level
       - filter by sensors
              s[ensor]=<comma separated list of sensor ID's>
       - filter by period
              from= YYYY-MM-DD | YYYY-MM-DDThh:mm:ssZ - start date or timestamp
              to= YYYY-MM-DD | YYYY-MM-DDThh:mm:ssZ - end date or timestamp. Current time
                  if omitted
              days=n - last n days
              hours=n - last n hours
              minutes=n - last n minutes
              seconds=n - last n seconds
       - template for formatting JSON response
              t[emplate]=nnn - response is formatted using template nnn

Examples:
       /last?sensor=125424,265314,336542&measure=humidity
       returns relative humidity measured by sensors 125424,265314 and 336542

       /last?s=125424&s=265314&s=336542&m=h&t=myJSON
       the same data formatted using template myJSON. Templates are defined in agent
       configuration file.
```

/history/125424?from=2018-03-18&to=2018-03-24&measure=humidity
returns relative humidity measured by sensors 125424 during period from March 18th, 2018 to March 24th, 2018

/history?minutes=60&s=125424&s=265314&s=336542&m=h&t=myJSON
returns relative humidity measured by sensors 125424,265314 and 336542 during last 60 minutes formatted using template myJSON.

/telemetry?hours=2&s=125424,265314,336542
returns telemetry data for sensors 125424,265314 and 336542 during last 2 hours

Output in SenML format


```
[{"bn":"aranet:394260700035:100021:","bt":1520934027,"n":"Temperature","u":"Cel","v":29.45
},{"n":"Humidity","u":"%RH","v":51},

{"bn":"aranet:394260700035:1000EE:","bt":1520934199,"n":"Temperature","u":"Cel","v":23.6},
 {"n":"Humidity","u":"%RH","v":64}]
```


Output formatted by template myJSON

```
templates:
- name: myJSON
  fields:
       name:
         senml: "n"
       sensor:
         senml: bn
       time:
         senml: bt
         type: string
       unit:
         senml: u
       value:
         senml: v
         type: string
```


```
[{"name":"Temperature","sensor":"aranet:394260700035:100021","time":"2018-03-13T09:50:27Z"
,"unit":"Cel","value":"29.65"},{"name":"Temperature","sensor":"aranet:394260700035:1000EE"
,"time":"2018-03-13T09:43:19Z","unit":"Cel","value":"23.6"}]
```

Some examples of using the commands:

http://127.0.0.1:8080/last
will return all the latest measurement readings for all connected sensors in SenML(default) format.

http://127.0.0.1:8080/last?m=t
will return the latest temperature measurement readings for all connected sensors.

http://127.0.0.1:8080/last/1000EE?m=t
will return the latest temperature measurement reading for sensor with id 1000EE.

http://127.0.0.1:8080/last?t=myJSON
will return the latest measurement readings in format which has been defined as template with name "myJSON".

http://127.0.0.1:8080/last/1000EE?m=t&t=example
will return the latest temperature measurement readings for sensor with id 1000EE in format which has been defined as template with name "example".

http://127.0.0.1:8080/history/1000EE?m=t&from=2018-07-09T18:00:00Z&to=2018-07-10T10:00:00Z
will return the temperature measurement readings for sensor with id 1000EE made from July 9, 2018 6:00:00 PM GMT to July 10, 2018 10:00:00 AM GMT.

http://127.0.0.1:8080/telemetry?minutes=150&s=1000EE,10010A
will return the received signal strength indicator and battery charge level readings for sensors with id 1000EE and 10010A made during last 150 minutes.

Configuration details

`aranet` section

Options for each Aranet PRO device start with "-" symbol

Option	Description
url	The host name or IP address of the Aranet PRO
user	The Aranet PRO user name for authentication. It is recommended to create a separate account for *aranet-agent*
password	The Aranet PRO user password for authentication. Enter the password when you add Aranet PRO to *config.yml* or the user name or password has been changed. At startup *aranet-agent* looks for a password and, if entered, calculates the hash value and replaces password with placeholder.
hash	Auto-generated credential data hash. Should not be edited by a user.

`http` section

Option	Description
status	Enables or disables the HTTP service
port	The port number for the HTTP service
ssl	SSL options *aranet-agent* can use automatically deployed certificates from Let's Encrypt (https://letsencrypt.org) or manually installed certificates obtained from some Certificate Authority
status	Enables or disables SSL for the HTTP service.
certfile	The pathname to the certificate file .crt

keyfile	The pathname to the key file .key
autocert	Enables or disables auto-renewing SSL certificates from Let’s Encrypt
autodomains	One or more comma separated domain names to obtain certificates for
templates	Field mappings for custom JSON responses
name	Template identifier
fields	List of response tags. Each tag has mandatory attribute `senml` for corresponding SenML field and optional attribute `type: string` to convert field value from native format to string

`snmp` section

Option	Description
status	Enables or disables the SNMP service
port	The port number for the SNMP service
v2	Enables or disables support for the SNMPv2 protocol messages
v3	Enables or disables support for the SNMPv3 protocol messages
community	SNMP community string for use with managers which support SNMPv1 and SNMPv2c protocol
node	Values for contact, name and physical location information for the SNMP service
contact	The person who manages the *aranet-agent*
name	A name identifying the *aranet-agent*
location	Physical location of the *aranet-agent*
users	List of SNMPv3 user accounts and credentials
name	The username

pass	The authentication and privacy pass phrase. After a password is entered and *aranet-agent* started, a localized hash key is generated and written to *config.yml*. After the generation of the key, the password is replaced by a placeholder.
authtype	The authentication method. Possible methods are: MD5 message-digest 5 algorithm SHA secure hash algorithm NoAuth no authentication
privtype	The privacy (encryption) protocol. Possible values are: AES message-digest 5 algorithm DES secure hash algorithm NoPriv no encryption
hashl	Auto-generated localized key hash. Should not be edited by a user.