TinySolutions

RESTfull API

Olav Frengstad <olav@fwt.no>

Table of contents

4.1. Message

4.1.2. GET /messages/, GET /network/<network>/messages, GET /node/<node>/messages

Retrieve a list of messages with each child formatted as described in § 3.1

Parameters:

node **= *** *optional*	An API resource to limit the fetch to
network **= *** *optional*	An API network resource to limit the fetch to
type = * *optional*	Which type of messages to be included in the return list, available options are [*command, event, serial, serial_out*]
fields **= *** *optional*	Comma separated list of fields to include in result
payload **= *** *optional*	Comma separated list of payload attributes to include in result. By default none are included.
meta **= *** *optional*	Comma separated list of meta attributes to include in result. By default *timestamp* and *node* are included.
composite.struct = 0|1 *optional*	Integer flag to control return either as a single- or multi-dimensional matrix. Default is *0*
composite.prefix **= *** *optional*	If *composite.struct* is true the group of the attribute will be prepended to the element name using the character specified for separator.

limit = 100 *optional*	Number of message to retrieve
date.from = * *optional*	Date as unix timestamp format to use for limiting a fetch
date.to = * *optional*	Date as unix timestamp format to use for limiting a fetch

```
return HTTP 200 Ok, When content was retrieved.
       HTTP 400 Bad Request, If any parameters are malformed
       HTTP 409 Conflict, If multiple versions exists
```

4.2. Node

4.2.2. GET /node || GET /network/<network-key>/nodes

Fetches a list of node resources. This will by default include the 100 first columns defined for that node, you are encouraged to filter the result set by using *fields, config* and *meta* parameters.

Parameters:

network *required*	The network to fetch nodes from (can be set by using */network/<network>/nodes*)
limit *optional*	The maximum number of nodes to return in the fetch, *default 100*
offset *optional*	The offset to use when fetching a list of nodes, *default 0*
system_id *optional*	Only fetch nodes from the given system, *default 0 - fetches from all*
composite.struct *optional*	Return a 1 dimensional matrix instead of grouped
composite.prefix *optional*	Use in combination with ***flatStruct*** to give all the indices their group as prefix and using the value of ***compositePrefix*** as a seperator (*: . _ -)*
fields* *optional*	The state fields to include in results
config* *optional*	Any node config's to include in group

meta* *optional*	Any meta information to include in group

** Exclusive options, if one is used all other values wanted to return must be set explicit.*

```
return HTTP 200 Ok, When content is fetched
       HTTP 500 Internal Server Error, If parameter value is invalid
```

4.2.2. GET /node/<key>

Fetch the current representation of a node, this returns a structure identical to § 3.2.1. Additional metadata consists of the fields defined in § 3.2 with the exception of *config* which is included as data.

```
return HTTP 200 Ok, When content was read.
       HTTP 404 Not Found, If resource was not found
```

5. Examples

All examples can be run from a terminal that has *curl* installed.

A static auth token can be used for authentication: *cf02f72c*

5.1. Find nodes

List all nodes, you can use ?config=<cfg_p> AND/OR ?meta=<meta_p> AND/OR ? fields=<field_p> to fetch more information.

```
~> curl -H "Accept: application/xml" \
 api.tiny-solution.com/network/statsbygg-remmen/nodes?auth=cf02f72c

Date: Mon, 24 Sep 2012 09:25:48 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
Keep-Alive: timeout=5
Set-Cookie: auth=cf02f72c
Access-Control-Allow-Origin: *

<?xml version=”1.0” encoding=”utf-8”?>
<root>
     <data>
```

```
            <node>
                  <meta>
                        <network>statsbygg-remmen</network>
                        <node>>statsbygg-remmen-c4b5bc5d</node>
                  </meta>
            </node>
            ….
      </data>
</root>
```

Select nodes of 1 type, add the query parameter ?type=(building-sensor|meter|gateway|bridge) to fetch use a predicate. YOU NEED TO FETCH THE TYPE ASWELL UISING &meta=type

```
~> curl -H "Accept: application/xml" \
 api.tiny-solution.com/network/statsbygg-remmen/nodes?auth=cf02f72c \
  &meta=type&type=meter

Date: Mon, 24 Sep 2012 09:25:52 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
Keep-Alive: timeout=5
Set-Cookie: auth=cf02f72c
Access-Control-Allow-Origin: *

<?xml version=”1.0” encoding=”utf-8”?>
<root>
      <data>
            <node>
                  <meta>
                        <name>statsbygg-remmen</name>
                        <node>>statsbygg-remmen-2a117e13</node>
                        <type>gateway</type>
                  </meta>
            </node>
      </data>
</root>
```

5.2 Find messages

Get the latest messages from the network:

```
~> curl  -H "Accept: application/xml" \
 api.tiny-solution.com/network/statsbygg-remmen/messages? \
  auth=cf02f72c&meta=event&limit=1

<?xml version=”1.0” encoding=”utf-8”?>
<root>
  <node>
    <meta>
      <network>statsbygg-remmen</network>
      <node>92781e03</node>
      <state>recv</state>
      <timestamp>1348484068.5454</timestamp>
    </meta>
    <field>
      <meter>67</meter>
    </field>
    <payload>
      <analog_io_0>2087</analog_io_0>
      <analog_io_1>8191</analog_io_1>
      <detail>9</detail>
      <detail_e>ima</detail_e>
      <digital_io_0>0</digital_io_0>
      <digital_io_1>0</digital_io_1>
      <digital_io_2>0</digital_io_2>
      <digital_io_3>1</digital_io_3>
      <digital_io_4>0</digital_io_4>
      <digital_io_5>0</digital_io_5>
      <digital_io_6>0</digital_io_6>
      <digital_io_7>0</digital_io_7>
      <fw_version>1.340000000000000007994e+00</fw_version>
      <hw_version>2.0000000000000000000e+00</hw_version>
      <jump_count>1</jump_count>
      <jump_level>1</jump_level>
      <locator>2130781991</locator>
      <msg_data>30486</msg_data>
      <msg_id>52713</msg_id>
      <node_id>258</node_id>
      <packet_latency>0</packet_latency>
      <raw>IwEAAAACAQAAVAEBzekAAAIJdxZ/AScnm3HvCCcf/wIAASI=</raw>
      <rssi>84</rssi>
      <system_id>1</system_id>
      <temp>27</temp>
      <type>event</type>
      <voltage>3.390000000000000012434e+00</voltage>
    </payload>
```

```
  </node>
</root>
```

Get messages from a node:

```
~> curl -H "Accept: application/xml" \
 api.tiny-solution.com/node/statsbygg-remmen-92781e03/messages? \
  auth=cf02f72c&meta=event&limit=1

<?xml version=”1.0” encoding=”utf-8”?>
<root>
  <node>
    <meta>
      <network>statsbygg-remmen</network>
      <node>92781e03</node>
      <state>recv</state>
      <timestamp>1348484068.5454</timestamp>
    </meta>
    <field>
      <meter>67</meter>
    </field>
    <payload>
      <analog_io_0>2087</analog_io_0>
      <analog_io_1>8191</analog_io_1>
      <detail>9</detail>
      <detail_e>ima</detail_e>
      <digital_io_0>0</digital_io_0>
      <digital_io_1>0</digital_io_1>
      <digital_io_2>0</digital_io_2>
      <digital_io_3>1</digital_io_3>
      <digital_io_4>0</digital_io_4>
      <digital_io_5>0</digital_io_5>
      <digital_io_6>0</digital_io_6>
      <digital_io_7>0</digital_io_7>
      <fw_version>1.340000000000000007994e+00</fw_version>
      <hw_version>2.000000000000000000000e+00</hw_version>
      <jump_count>1</jump_count>
      <jump_level>1</jump_level>
      <locator>2130781991</locator>
      <msg_data>30486</msg_data>
      <msg_id>52713</msg_id>
      <node_id>258</node_id>
      <packet_latency>0</packet_latency>
      <raw>IwEAAAACAQAAVAEBzekAAAIJdxZ/AScnm3HvCCcf/wIAASI=</raw>
      <rssi>84</rssi>
```

```
      <system_id>1</system_id>
      <temp>27</temp>
      <type>event</type>
      <voltage>3.3900000000000012434e+00</voltage>
    </payload>
  </node>
</root>
```

Get only relevant fields of a message:

```
~> curl -H "Accept: application/xml" \
 api.tiny-solution.com/node/statsbygg-remmen-92781e03/messages? \
  auth=cf02f72c&meta=event&limit=1&fields=meter&composite.prefix=_

<?xml version="1.0" encoding="utf-8"?>
<root>
      <node>
            <field_meter>71</field_meter>
            <meta_timestamp>1348490400.234</meta_timestamp>
      </node>
</root>
```

Get CSV file format:

```
~> curl -H "Accept: text/csv" \
 api.tiny-solution.com/node/statsbygg-remmen-92781e03/messages? \
  auth=cf02f72c&meta=event&limit=10&fields=meter&composite.prefix=_

field_meter,meta_timestamp
92,1348490520.8453
81,1348490460.5529
71,1348490400.234
74,1348490339.9468
73,1348490279.6454
73,1348490219.3279
74,1348490159.0287
75,1348490098.7311
75,1348490038.4327
75,1348489978.1199
```

Get specific timeranges:
Here you can use -<secs> to use current time, i.e. date.from=-600 to get messages from the latest 10min

Selects messages in range 15 min to 10 min ago (*note: you can't do*

```
~> curl -D - -H "Accept: text/csv" \
 api.tiny-solution.com/node/statsbygg-remmen-92781e03/messages?\
 auth=cf02f72c&meta=event&date.from=-900&date.to=-
600&fields=meter&composite.prefix=_

field_meter,meta_timestamp
73,1348490219.3279
74,1348490159.0287
75,1348490098.7311
75,1348490038.4327
75,1348489978.1199
```