

MYSCADA RESTFUL API

mySCADA allow to retrieve historical data using the restful API. Data are retrieved using the HTTP or preferably (due to higher security) HTTPS requests. Please use a POST or GET methods to retrieve the data. To retrieve the data, you must use basic digest authentication. Provided username and password is the same as when downloading the project into myPRO/myBOX/myPANEL from myDESIGNER. Authentication parameters can be changed in setting page.

Default login parameters:

Username: *myscadaupload*

Password: set up a password in the settings page

Retrieving the data-logs configuration

You can retrieve configuration parameters of all data-logs as one JSON object.

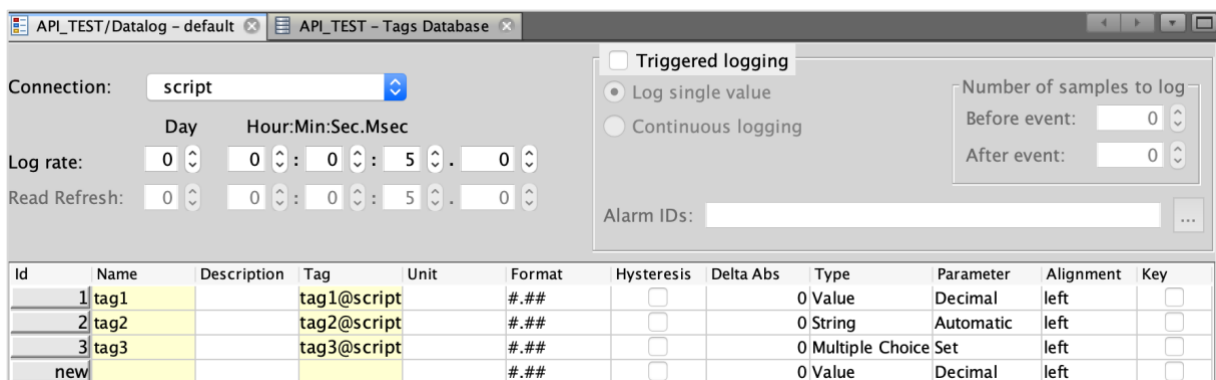
Path: /download/datalogs.json

Method: POST or GET

Reply is a JSON object with configuration.

Example:

Datalog defined in myDESIGNER with 3 tags:



Id	Name	Description	Tag	Unit	Format	Hysteresis	Delta Abs	Type	Parameter	Alignment	Key
1	tag1		tag1@script		###	☐	0	Value	Decimal	left	☐
2	tag2		tag2@script		###	☐	0	String	Automatic	left	☐
3	tag3		tag3@script		###	☐	0	Multiple Choice	Set	left	☐
	new				###	☐	0	Value	Decimal	left	☐

And its representation in the JSON object:

```
object{3}
  projectID:785344
  datalog[1]
    0{19}
      id:1
      name:default
      refreshRead:5000
      refreshLog:5000
      condition:false
      conditionContinuous:false
      samplesBefore:0
      samplesAfter:0
```

```
maxSize:85
autoSize:true
defaultDatalog:true
viewAccess:-1
connection:script
olderThen:31536000
scriptDatalog:false
hideDatalog:false
enableDataFilter:true
keys[0]
tags[3]
  0{17}
    id:1
    name:tag1
    description:
    tag:0.0
    alias:tag1
    unit:
    nameLangs{0}
    descriptionLangs{0}
    unitLangs{0}
    format:#.##
    hysteresis:false
    deltaAbs:0
    digital:false
    type:0
    subType:0
    align:left
    key:false
  1{18}
    id:2
    name:tag2
    description:
    tag:0.1
    alias:tag2
    unit:
    nameLangs{0}
    descriptionLangs{0}
    unitLangs{0}
```

```

format:###
hysteresis:false
deltaAbs:0
digital:false
type:1
subType:0
align:left
key:false
params{3}
2{18}
  id:3
  name:tag3
  description:
  tag:0.2
  alias:tag3
  unit:
  nameLangs{0}
  descriptionLangs{0}
  unitLangs{0}
  format:###
  hysteresis:false
  deltaAbs:0
  digital:false
  type:4
  subType:0
  paramsMultipleChoice{3}
    1.0{2}
      text:closed
      translation{0}
    2.0{2}
      text:open
      translation{0}
    3.0{2}
      text:error
      translation{0}
  align:left
  key:false
datalogViews[1]

```

As you can see, all the parameters user can set up in myDESIGNER are available here.

First parameter is `projectID`, this is a unique id of the project. You will need this number to communicate with API to retrieve data.

Second parameter is a list of data-logs. All data-logs are represented by its unique ID (integer number starting from 1). In the section tags, you can find all the tags that are logged in the data-log including its type (0=value, 1=string, 3=Boolean, 4=multiple choice).

Retrieving the data-logs configuration

Path: /api

Method: POST

Data format: JSON

Path: /api

Method: GET

Data format: values separated by &

Parameter	Description
prj	Unique project number
t	31
start	Start time in seconds (UTC linux time format)
startMs	Start time milliseconds (not required)
end	End time in seconds (UTC linux time format)
limit	Limit number of records from start parameter (ASCENDING order)
limitEnd	Limit number of records from end parameter back in time (DESCENDING order)
	First parameter is datalog unique ID then followed by the ID of tags separated by coma
tags	You can specify multiple data-logs in that case split it by semicolon
	Example: 2 1,2;3 1,2 will retrieve tag 1 and 2 from datalog with ID 2 and tag 1 and 2 from data-log with ID 3
seq	Sequential number, will be increased by 1 in reply. Not required.

To retrieve a data from start in ascending order specify only start and limit parameter. To retrieve a data from start in descending order specify only end and limitEnd parameter.

Example:

To read 10 records in ascending order from given time from 2 different data-logs with IDs 2 and 3. From each data-log we will read two tags with IDs 1 and 2.

```
{
  limit:10
  prj:581546
  start:1538681082
  startMs:200
  t:31
  tags:2|1,2;3|1,2
  seq:1
}
```

Server reply:

```
  d: //object with data
    2: //values from data-log with id 2
      d: Array(10). //array of values (only first 1 shown)
        0: //first item of data
          1: 4433 //tag value with ID 1
          2: "string" //tag value with ID 2
          tm: Array(2) //Array of time
          0: 1538681082 //seconds UTC Linux time
          1: 140000 //microseconds
        3: {d: Array(10)}. //values from data-log with id 3
      r: 31 //reply on request ID 31
      s: 1 //status 1=OK, 0=error
```