

Description

- Structure of rule entry
 - Structure of actions entry
 - Structure of conditions entry
- Actions types
 - SET - create a new or redefine an existing metric
 - EXTEND - extend the table with new metrics
 - DROP - delete metrics
- Examples

In the standard mode of the sensor operation, the monitoring process is as follows:

- the specified probe is performed ;
- JSON data is sent to the server;
- the data is displayed in the SAYMON web interface in the "Data" table.

Each new data array overwrites the previous result, removing the missing metrics, that in some cases narrows down the range of setting up rules possibilities of changing states and generating accidents. The functional "Data forming rules (Stat Rules)" eliminates it and expands the possibilities for managing the data displayed in the data table.

Stat Rules provide the user with 3 types of rules:

- set
- extend
- drop

each of which is described below.

Stat Rules works with two data arrays:

- current - the current displayed data in the table;
- new - the newly arrived data.

Structure of rule entry

Each rule consists of conditions under which actions are applied to the data.

The "actions" field is required and must contain at least one action.

In the absence of the "conditions" field, the specified actions will always be performed - each time the server receives data.

The structure of a single rule entry:

```
[
  {
    "conditions": [...],
    "actions": [...]
  }
]
```

The structure of several rules entry:

```
[
  {
    "conditions": [...],
    "actions": [...]
  },
  {
    "conditions": [...],
    "actions": [...]
  },
  ...
]
```

Structure of actions entry

The "actions" field lists the types of actions and additional fields that are characteristic of a particular type of action and are described below.

The structure of the entry of a rule with a single action:

```
[
  {
    "conditions": [...],
    "actions": [
      {
        "type": "extend"
      }
    ]
  }
]
```

Rules can be combined. In this case, they are executed in the order of the sequence:

```
[
  {
    "conditions": [...],
    "actions": [
      {
        "type": "extend"
      },
      {
        "type": "drop"
      }
    ]
  }
]
```

In the absence of conditions, actions can be combined in two equivalent ways:

```
[
  {
    "actions": [
      {
        "type": "extend"
      },
      {
        "type": "drop"
      }
    ]
  }
]
```

```
[
  {
    "actions": [
      {
        "type": "extend"
      }
    ]
  },
  {
    "actions": [
      {
        "type": "drop"
      }
    ]
  }
]
```

Structure of conditions entry

The structure of the entry of performing actions conditions is similar to the [state change conditions](#). You can find it by switching the graphical editor of state change conditions to a textual representation.

Actions types

SET - create a new or redefine an existing metric

This action has two required fields:

- field - the name of the metric to which the value should be written (if there is no metric with the specified name, it will be added to the data table);
- value - the value to write to the metric.

As a value, you can use:

- numeric values, for example, 10;
- text data, for example, "text";
- values of other metrics from the data table with *current* and *new* pointers, for example, "{{new.temperature}}";
- formulas using metrics from a data table with *current* and *new* pointers, for example, "{{current.temperature}} - {{new.temperature}}".

The structure of the SET rule entry:

```
[
  {
    "actions": [
      {
        "type": "set",
        "field": "metric_name",
        "value": "metric_value"
      }
    ]
  }
]
```

EXTEND - extend the table with new metrics

This action complements the data table by new metrics. Metrics that are already present in the data table will not be deleted.

The EXTEND action has 2 optional mutually exclusive fields:

- include - names of metrics that must be added to existing ones;
- exclude - names of metrics that must be ignored when adding.

Example 1. Extension of the data table by all new metrics.

```
[
  {
    "actions": [
      {
        "type": "extend"
      }
    ]
  }
]
```

Example 2. Extension of the data table only by the specified new metrics.

```
[
  {
    "actions": [
      {
        "type": "extend",
        "include": ["metric_name_1", "metric_name_2", ...]
      }
    ]
  }
]
```

Example 3. Extension of the data table by all new metrics, except those specified.

```
[
  {
    "actions": [
      {
        "type": "extend",
        "exclude": ["metric_name_3", "metric_name_4", ...]
      }
    ]
  }
]
```

DROP - delete metrics

This action drops metrics from the data table.

The DROP action has 2 optional mutually exclusive fields:

- include - the names of the metrics that must be dropped;
- exclude - the names of the metrics that must be left.

Example 1. Drop all metrics.

```
[
  {
    "actions": [
      {
        "type": "drop"
      }
    ]
  }
]
```

Example 2. Drop the specified metrics.

```
[
  {
    "actions": [
      {
        "type": "drop",
        "include": ["metric_name_1", "metric_name_2", ...]
      }
    ]
  }
]
```

Example 3. Drop all metrics except those specified.

```
[
  {
    "actions": [
      {
        "type": "drop",
        "exclude": ["metric_name_3", "metric_name_4", ...]
      }
    ]
  }
]
```

Examples

You can find examples in the following articles:

- [SET - examples of use](#)
- [EXTEND - examples of use](#)
- [DROP - examples of use](#)