
NodeConductor Zabbix Documentation

Release 0.6.0

OpenNode

December 05, 2016

1	Guide	3
2	API	5
3	Endpoints	13
4	License	23
5	Indices and tables	25

Zabbix service provides an interface to Zabbix monitoring system.

1.1 Installation from source

- Install NodeConductor
- Clone NodeConductor Zabbix repository

```
git clone git@code.opennodecloud.com:nodeconductor/nodeconductor-zabbix.git
```

- Install NodeConductor Zabbix into NodeConductor virtual environment

```
cd /path/to/zabbix/  
python setup.py install
```

1.1.1 Configuration

Zabbix plugin settings should be defined in NodeConductor's settings.py file under **NODECONDUCTOR_ZABBIX** section.

For example,

```
NODECONDUCTOR_ZABBIX = {  
    'SMS_SETTINGS': {  
        'SMS_EMAIL_FROM': 'zabbix@example.com',  
        'SMS_EMAIL_RCPT': '{phone}@example.com',  
    },  
}
```

Available settings:

SMS_SETTINGS A dictionary of configurations for Zabbix SMS notifications.

SMS_EMAIL_FROM Defines the email from which SMS notification will be sent.

SMS_EMAIL_RCPT Defines the email to which SMS notification will be sent. It should include *{phone}* string, which will be after replaced with a phone number.

2.1 Services

2.1.1 List services

To get a list of services, run GET against **/api/zabbix/** as authenticated user.

2.1.2 Create a Zabbix service

To create a new Zabbix service, issue a POST with service details to **/api/zabbix/** as a customer owner.

Request parameters:

- name - service name,
- customer - URL of service customer,
- settings - URL of Zabbix settings, if not defined - new settings will be created from server parameters,
- dummy - is service dummy.

The following rules for generation of the service settings are used:

- backend_url - Zabbix API URL (e.g. http://example.com/zabbix/api_jsonrpc.php);
- username - Zabbix user username (e.g. admin);
- password - Zabbix user password (e.g. zabbix);
- host_group_name - Zabbix group name for registered hosts (default: "nodeconductor");
- interface_parameters - default parameters for hosts interface. (default: {"dns": "", "ip": "0.0.0.0", "main": 1, "port": "10050", "type": 1, "useip": 1});
- templates_names - List of Zabbix hosts templates. (default: ["NodeConductor"]);
- database_parameters - Zabbix database parameters. (default: {"host": "localhost", "port": "3306", "name": "zabbix", "user": "admin", "password": ""})

Example of a request:

```
POST /api/zabbix/ HTTP/1.1
Content-Type: application/json
Accept: application/json
Authorization: Token c84d653b9ec92c6cbac41c706593e66f567a7fa4
Host: example.com
```

```
{
  "name": "My Zabbix"
  "customer": "http://example.com/api/customers/2aadad6a4b764661add14dfdda26b373/",
  "backend_url": "http://example.com/zabbix/api_jsonrpc.php",
  "username": "admin",
  "password": "zabbix"
}
```

2.2 Service-project links

2.2.1 Create and delete link

In order to be able to provision Zabbix resources, it must first be linked to a project. To do that, POST a connection between project and a service to **/api/zabbix-service-project-link/** as staff user or customer owner. For example,

```
POST /api/zabbix-service-project-link/ HTTP/1.1
Content-Type: application/json
Accept: application/json
Authorization: Token c84d653b9ec92c6cbac41c706593e66f567a7fa4
Host: example.com

{
  "project": "http://example.com/api/projects/e5f973af2eb14d2d8c38d62bcbaccb33/",
  "service": "http://example.com/api/zabbix/b0e8a4cbd47c4f9ca01642b7ec033db4/"
}
```

To remove a link, issue DELETE to url of the corresponding connection as staff user or customer owner.

2.2.2 List links

To get a list of connections between a project and a Zabbix service, run GET against **/api/zabbix-service-project-link/** as authenticated user. Note that a user can only see connections of a project where a user has a role.

2.3 Hosts

2.3.1 Create host

A new Zabbix host can be created by users with project administrator role, customer owner role or with staff privilege (is_staff=True). To create a host, client must issue POST request to **/api/zabbix-hosts/** with parameters:

- name - host name;
- service_project_link - url of service-project-link;
- visible_name - host visible name (optional if scope is defined);
- scope - optional url of related object, for example of OpenStack instance;
- description - host description (optional);
- interface_parameters - host interface parameters (optional);
- host_group_name - host group name (optional);

- templates - list of template urls (optional).

For optional fields, such as interface_parameters, host_group_name, templates if value is not specified in request, default value will be taken from service settings.

Example of a valid request:

```
POST /api/zabbix-hosts/ HTTP/1.1
Content-Type: application/json
Accept: application/json
Authorization: Token c84d653b9ec92c6cbac41c706593e66f567a7fa4
Host: example.com

{
  "name": "test host",
  "visible_name": "test host",
  "description": "sample description",
  "service_project_link": "http://example.com/api/zabbix-service-project-link/1/",
  "templates": [
    {
      "url": "http://example.com/api/zabbix-templates/99771937d38d41ceba3352b99e01b00b/"
    }
  ]
}
```

2.3.2 Get host

To get host data - issue GET request against `/api/zabbix-hosts/<host_uuid>/`.

Example rendering of the host object:

```
{
  "url": "http://example.com/api/zabbix-hosts/c2c29036f6e441908e5f7ca0f2441431/",
  "uuid": "c2c29036f6e441908e5f7ca0f2441431",
  "name": "a851fa75-5599-467b-be11-3d15858e8673",
  "description": "",
  "start_time": null,
  "service": "http://example.com/api/zabbix/1ffaa994d8424b6e9a512ad967ad428c/",
  "service_name": "My Zabbix",
  "service_uuid": "1ffaa994d8424b6e9a512ad967ad428c",
  "project": "http://example.com/api/projects/8dc8f34f27ef4a4f916184ab71e178e3/",
  "project_name": "Default",
  "project_uuid": "8dc8f34f27ef4a4f916184ab71e178e3",
  "customer": "http://example.com/api/customers/7313b71bd1cc421ea297dcb982e40260/",
  "customer_name": "Alice",
  "customer_native_name": "",
  "customer_abbreviation": "",
  "project_groups": [],
  "tags": [],
  "error_message": "",
  "resource_type": "Zabbix.Host",
  "state": "Online",
  "created": "2015-10-16T11:18:59.596Z",
  "backend_id": "2535",
  "visible_name": "a851fa75-5599-467b-be11-3d15858e8673",
  "interface_parameters": "{u'ip': u'0.0.0.0', u'useip': 1, u'dns': u'', u'main': 1, u'type': 1, u'group': u'nodeconductor'",
  "host_group_name": "nodeconductor",
  "scope": null,
```

```
"templates": [
  {
    "url": "http://example.com/api/zabbix-templates/99771937d38d41ceba3352b99e01b00b/",
    "uuid": "99771937d38d41ceba3352b99e01b00b",
    "name": "Template NodeConductor Instance",
    "items": [
      {
        "name": "Host name of zabbix_agentd running",
        "key": "agent.hostname"
      },
      {
        "name": "Agent ping",
        "key": "agent.ping"
      },
      {
        "name": "Version of zabbix_agent(d) running",
        "key": "agent.version"
      }
    ]
  }
],
"agreed_sla": 91.5,
"actual_sla": 100.0
}
```

2.3.3 Delete host

To delete host - issue DELETE request against `/api/zabbix-hosts/<host_uuid>/`.

2.3.4 Host statistics

URL: `/api/zabbix-hosts/<host_uuid>/items_history/`

Request should specify datetime points and items. There are two ways to define datetime points for historical data.

1. Send `?point=<timestamp>` parameter that can list. Response will contain historical data for each given point in the same order.
2. Send `?start=<timestamp>`, `?end=<timestamp>`, `?points_count=<integer>` parameters. Result will contain `<points_count>` points from `<start>` to `<end>`.

Also you should specify one or more name of host template items, for example 'openstack.instance.cpu_util'

Response is list of datapoint, each of which is dictionary with following fields:

- 'point' - timestamp;
- 'value' - values are converted from bytes to megabytes, if possible;
- 'item' - name of host template item.

Example response:

```
[
  {
    "point": 1441935000,
    "value": 0.1393,
    "item": "openstack.instance.cpu_util"
  },
]
```

```
[
  {
    "point": 1442163600,
    "value": 10.2583,
    "item": "openstack.instance.cpu_util"
  },
  {
    "point": 1442392200,
    "value": 20.3725,
    "item": "openstack.instance.cpu_util"
  },
  {
    "point": 1442620800,
    "value": 30.3426,
    "item": "openstack.instance.cpu_util"
  },
  {
    "point": 1442849400,
    "value": 40.3353,
    "item": "openstack.instance.cpu_util"
  },
  {
    "point": 1443078000,
    "value": 50.3574,
    "item": "openstack.instance.cpu_util"
  }
]
```

2.3.5 Aggregated host statistics

URL: `/api/zabbix-hosts/aggregated_items_history/`

Request should specify host filtering parameters, datetime points, and items. Host filtering parameters are the same as for `/api/resources/` endpoint. Input/output format is the same as for `/api/zabbix-hosts/<host_uuid>/items_history/` endpoint.

Example request and response:

```
GET /api/zabbix-hosts/aggregated_items_history/?point=1436094582&point=1443078000&customer_uuid=73131
Content-Type: application/json
Accept: application/json
Authorization: Token c84d653b9ec92c6cbac41c706593e66f567a7fa4
Host: example.com

[
  {
    "point": 1436094582,
    "item": "openstack.instance.cpu_util",
    "value": 40.3353
  },
  {
    "point": 1443078000,
    "item": "openstack.instance.cpu_util",
    "value": 50.3574
  }
]
```

2.4 IT services and SLA calculation

The status of IT Service is affected by the status of its trigger.

2.4.1 List triggers

Triggers are available as Zabbix service properties under `/api/zabbix-triggers/` endpoint. You may filter triggers by template by passing its ID as GET query parameter.

```
[
  {
    "url": "http://example.com/api/zabbix-triggers/3e19dc77279d42ccb6c2e21f2a2f6ced/",
    "uuid": "3e19dc77279d42ccb6c2e21f2a2f6ced",
    "name": "Host name of zabbix_agentd was changed on {HOST.NAME}",
    "template": "http://example.com/api/zabbix-templates/8780ebf60ac448c4a3d083f0c71106ff/"
  }
]
```

2.4.2 List IT services

IT services are available as Zabbix service properties under `/api/zabbix-itservices/` endpoint.

```
{
  "url": "http://example.com/api/zabbix-itservices/db075c3c8d494f5886fc0f6686390624/",
  "uuid": "db075c3c8d494f5886fc0f6686390624",
  "name": "example-it-service",
  "description": "",
  "start_time": null,
  "service": "http://example.com/api/zabbix/18931f568b344b3fbc8d048cbe806ff6/",
  "service_name": "TST Zabbix",
  "service_uuid": "18931f568b344b3fbc8d048cbe806ff6",
  "project": "http://example.com/api/projects/f43171f9374442b78ce7e842effea0aa/",
  "project_name": "TST PaaS project",
  "project_uuid": "f43171f9374442b78ce7e842effea0aa",
  "customer": "http://example.com/api/customers/691f62f8d89e44d6a69d02b3b5334f7c/",
  "customer_name": "TST Paas customer",
  "customer_native_name": "",
  "customer_abbreviation": "",
  "project_groups": [],
  "tags": [],
  "error_message": "",
  "resource_type": "Zabbix.ITService",
  "state": "Online",
  "created": "2016-02-22T06:56:37.393Z",
  "backend_id": "1590",
  "access_url": null,
  "host": "http://example.com/api/zabbix-hosts/f8e46835e4654410915bd24c2f784876/",
  "algorithm": "problem, if at least one child has a problem",
  "sort_order": 1,
  "agreed_sla": "99.0000",
  "actual_sla": 100.0,
  "trigger": "http://example.com/api/zabbix-triggers/765b979ec9b34038b1b214f6be2bb0b5/",
  "trigger_name": "PostgreSQL is not available",
  "is_main": true
}
```

2.4.3 SLA periods

IT services list is displaying current SLAs for each of the items. By default, SLA period is set to the current month. To change the period pass it as a query argument:

- `?period=YYYY-MM` - return a list with SLAs for a given month
- `?period=YYYY` - return a list with SLAs for a given year

If SLA for the given period is not known or not present, it will be shown as **null** in the response.

2.4.4 SLA events

IT service SLAs are connected with occurrences of events. To get a list of such events issue a GET request to `/zabbix-itservices/<service_uuid>/events/`. Optionally period can be supplied using the format defined above.

The output contains a list of states and timestamps when the state was reached. The list is sorted in descending order by the timestamp.

Example output:

```
[
  {
    "timestamp": 1418043540,
    "state": "U"
  },
  {
    "timestamp": 1417928550,
    "state": "D"
  },
  {
    "timestamp": 1417928490,
    "state": "U"
  }
]
```

Endpoints

3.1 NodeConductor Zabbix

NodeConductor Zabbix

3.1.1 `/api/zabbix/`

A filter backend that uses django-filter. Supported actions and methods:

/api/zabbix/

Methods: GET, POST

Supported fields for creation:

- **name** – string
- **project** – link to /api/projects/<uuid>/
- **customer** – link to /api/customers/<uuid>/
- **settings** – link to /api/service-settings/<uuid>/
- **backend_url** – URL (Zabbix API URL (e.g. http://example.com/zabbix/api_jsonrpc.php))
- **username** – string (Zabbix user username (e.g. admin))
- **password** – string (Zabbix user password (e.g. zabbix))
- **available_for_all** – boolean (Service will be automatically added to all customers projects if it is available for all)
- **scope** – link to any: /api/zabbix-hosts/<uuid>/, /api/zabbix-it-services/<uuid>/ (VM that contains service)
- **database_parameters** – string (Zabbix database parameters (default: {"user": "admin", "host": "localhost", "password": "", "port": "3306", "name": "zabbix"}))
- **templates_names** – string (List of Zabbix hosts templates (default: []))
- **interface_parameters** – string (Default parameters for hosts interface (will be used if interface is not specified) (default: {"ip": "0.0.0.0", "useip": 1, "dns": "", "main": 1, "type": 1, "port": "10050"}))
- **host_group_name** – string (Zabbix host group name for registered hosts (default: "nodeconductor"))

Filter fields:

- ?customer = UUIDFilter
- ?name = string
- ?settings = link
- ?project_uuid = UUIDFilter
- ?project = link
- ?tag = ModelMultipleChoiceField
- ?rtag = ModelMultipleChoiceField
- ?shared = boolean
- ?type = ServiceTypeFilter

To list all services without regard to its type, run **GET** against /api/services/ as an authenticated user.To list services of specific type issue **GET** to specific endpoint from a list above as a customer owner. Individual endpoint used for every service type.To create a service, issue a **POST** to specific endpoint from a list above as a customer owner. Individual endpoint used for every service type.

You can create service based on shared service settings. Example:

```
POST /api/digitalocean/ HTTP/1.1
Content-Type: application/json
Accept: application/json
Authorization: Token c84d653b9ec92c6cbac41c706593e66f567a7fa4
Host: example.com

{
  "name": "Common DigitalOcean",
  "customer": "http://example.com/api/customers/1040561ca9e046d2b74268600c7e1105/",
  "settings": "http://example.com/api/service-settings/93ba615d6111466ebe3f792669059cb/"
}
```

Or provide your own credentials. Example:

```
POST /api/oracle/ HTTP/1.1
Content-Type: application/json
Accept: application/json
Authorization: Token c84d653b9ec92c6cbac41c706593e66f567a7fa4
Host: example.com
```

/api/zabbix/<uuid>/

Methods: GET, PUT, PATCH, DELETE

Supported fields for update:

- **name** – string
- **available_for_all** – boolean (Service will be automatically added to all customers projects if it is available for all)

/api/zabbix/<uuid>/credentials/

Methods: GET, POST

On GET request - return superadmin user data. On POST - reset superuser password and return new one.

/api/zabbix/<uuid>/link/

Methods: GET, POST

To get a list of resources available for import, run **GET** against **/<service_endpoint>/link/** as an authenticated user. Optionally **project_uuid** parameter can be supplied for services requiring it like OpenStack.

To import (link with NodeConductor) resource issue **POST** against the same endpoint with resource id.

POST **/api/openstack/08039f01c9794efc912f1689f4530cf0/link/ HTTP/1.1**

Content-Type: application/json

Accept: application/json

Authorization: Token c84d653b9ec92c6cbac41c706593e66f567a7fa4

Host: example.com

```
{
  "backend_id": "bd5ec24d-9164-440b-a9f2-1b3c807c5df3",
  "project": "http://example.com/api/projects/e5f973af2eb14d2d8c38d62bcbaccb33/"
}
```

/api/zabbix/<uuid>/managed_resources/

Methods: GET

/api/zabbix/<uuid>/unlink/

Methods: POST

Unlink all related resources, service project link and service itself.

3.1.2 /api/zabbix-users/

A filter backend that uses django-filter. Supported actions and methods:

/api/zabbix-users/

Methods: GET, POST

Supported fields for creation:

- **alias** – string
- **name** – string
- **surname** – string
- **type** – choice('admin', 'default', 'superadmin')
- **groups** – list of [{name: string, settings: link to /api/service-settings/<uuid>/}]
- **settings** – link to /api/service-settings/<uuid>/
- **phone** – string

Filter fields:

- ?alias = string
- ?surname = string
- ?name = string
- ?settings_uuid = UUIDFilter
- ?settings = link

/api/zabbix-users/<uuid>/

Methods: GET, PUT, PATCH, DELETE

Supported fields for update:

- **alias** – string
- **name** – string
- **surname** – string
- **type** – choice('admin', 'default', 'superadmin')
- **groups** – list of [{name: string, settings: link to /api/service-settings/<uuid>/}]
- **phone** – string

/api/zabbix-users/<uuid>/password/

Methods: POST

3.1.3 /api/zabbix-user-groups/

A filter backend that uses django-filter. Supported actions and methods:

/api/zabbix-user-groups/

Methods: GET

Filter fields:

- ?name = string
- ?settings_uuid = UUIDFilter
- ?settings = link

```
/api/zabbix-user-groups/<uuid>/
```

Methods: GET

3.1.4 /api/zabbix-triggers/

A filter backend that uses django-filter. Supported actions and methods:

```
/api/zabbix-triggers/
```

Methods: GET

Filter fields:

- ?template_uuid = UUIDFilter
- ?template = link
- ?name = string
- ?settings_uuid = UUIDFilter
- ?settings = link

```
/api/zabbix-triggers/<uuid>/
```

Methods: GET

3.1.5 /api/zabbix-templates/

A filter backend that uses django-filter. Supported actions and methods:

```
/api/zabbix-templates/
```

Methods: GET

Filter fields:

- ?name = string
- ?settings_uuid = UUIDFilter
- ?settings = link

```
/api/zabbix-templates/<uuid>/
```

Methods: GET

3.1.6 /api/zabbix-service-project-link/

A filter backend that uses django-filter. Supported actions and methods:

/api/zabbix-service-project-link/

Methods: GET, POST

Supported fields for creation:

- **project** – link to /api/projects/<uuid>/
- **service** – link to /api/zabbix/<uuid>/

Filter fields:

- ?project = link
- ?service_uuid = UUIDFilter
- ?customer_uuid = UUIDFilter
- ?project_uuid = UUIDFilter

To get a list of connections between a project and an service, run **GET** against service_project_link_url as authenticated user. Note that a user can only see connections of a project where a user has a role.

If service has *available_for_all* flag, project-service connections are created automatically. Otherwise, in order to be able to provision resources, service must first be linked to a project. To do that, **POST** a connection between project and a service to service_project_link_url as stuff user or customer owner.

/api/zabbix-service-project-link/<pk>/

Methods: GET, DELETE

To remove a link, issue **DELETE** to URL of the corresponding connection as stuff user or customer owner.

3.1.7 /api/zabbix-itservices/

SLA filter

Allows to filter or sort resources by actual_sla Default period is current year and month.

Example query parameters for filtering list of OpenStack instances:

```
/api/openstack-instances/?actual_sla=90&period=2016-02
```

Example query parameters for sorting list of OpenStack instances:

```
/api/openstack-instances/?o=actual_sla&period=2016-02
```

Monitoring filter

Filter and order resources by monitoring item. For example, given query dictionary

```
{
  'monitoring__installation_state': True
}
```

it produces following query

```
{
  'monitoring_item__name': 'installation_state',
  'monitoring_item__value': True
}
```

Example query parameters for sorting list of OpenStack instances:

```
/api/openstack-instances/?o=monitoring__installation_state
```

Tags ordering. Filtering for complex tags.

Example: ?tag__license-os=centos7 - will filter objects with tag “license-os:centos7”.

Allow to define next parameters in view:

- tags_filter_db_field - name of tags field in database. Default: tags.
- tags_filter_request_field - name of tags in request. Default: tag.

In PostgreSQL NULL values come *last* with ascending sort order. In MySQL NULL values come *first* with ascending sort order. This filter provides unified sorting for both databases. Supported actions and methods:

```
/api/zabbix-it-services/
```

Methods: GET, POST

Supported fields for creation:

- **name** - string
- **description** - string
- **service_project_link** - link to /api/zabbix-service-project-link/<pk>/
- **host** - link to /api/zabbix-hosts/<uuid>/
- **algorithm** - choice('do not calculate', 'problem, if all children have problems', 'problem, if at least one child has a problem')
- **sort_order** - integer
- **agreed_sla** - float
- **trigger** - link to /api/zabbix-triggers/<uuid>/
- **is_main** - boolean (Main IT service SLA will be added to hosts resource as monitoring item.)

```
/api/zabbix-it-services/<uuid>/
```

Methods: GET, PUT, PATCH, DELETE

Supported fields for update:

- **name** - string
- **description** - string
- **host** - link to /api/zabbix-hosts/<uuid>/
- **algorithm** - choice('do not calculate', 'problem, if all children have problems', 'problem, if at least one child has a problem')
- **sort_order** - integer
- **agreed_sla** - float
- **trigger** - link to /api/zabbix-triggers/<uuid>/
- **is_main** - boolean (Main IT service SLA will be added to hosts resource as monitoring item.)

```
/api/zabbix-it-services/<uuid>/events/
```

Methods: GET

3.1.8 /api/zabbix-hosts/

Representation of Zabbix hosts and related actions.

SLA filter

Allows to filter or sort resources by actual_sla Default period is current year and month.

Example query parameters for filtering list of OpenStack instances:

```
/api/openstack-instances/?actual_sla=90&period=2016-02
```

Example query parameters for sorting list of OpenStack instances:

```
/api/openstack-instances/?o=actual_sla&period=2016-02
```

Monitoring filter

Filter and order resources by monitoring item. For example, given query dictionary

```
{
  'monitoring__installation_state': True
}
```

it produces following query

```
{
  'monitoring_item__name': 'installation_state',
  'monitoring_item__value': True
}
```

Example query parameters for sorting list of OpenStack instances:

```
/api/openstack-instances/?o=monitoring__installation_state
```

Tags ordering. Filtering for complex tags.

Example: ?tag__license-os=centos7 - will filter objects with tag “license-os:centos7”.

Allow to define next parameters in view:

- tags_filter_db_field - name of tags field in database. Default: tags.
- tags_filter_request_field - name of tags in request. Default: tag.

In PostgreSQL NULL values come *last* with ascending sort order. In MySQL NULL values come *first* with ascending sort order. This filter provides unified sorting for both databases.

Supported actions and methods:

/api/zabbix-hosts/

Methods: GET, POST

Supported fields for creation:

- **name** – string
- **description** – string
- **service_project_link** – link to /api/zabbix-service-project-link/<pk>/
- **visible_name** – string
- **host_group_name** – string
- **scope** – link to any: /api/zabbix-hosts/<uuid>/, /api/zabbix-it-services/<uuid>/
- **templates** – list of [{parents: list of [link to /api/zabbix-templates/<uuid>/], children: list of [link to /api/zabbix-templates/<uuid>/], name: string, settings: link to /api/service-settings/<uuid>/}]
- **interface_ip** – IP address (IP of host interface.)

Representation of Zabbix hosts and related actions.

/api/zabbix-hosts/aggregated_items_history/

Methods: GET

Representation of Zabbix hosts and related actions. Get sum of hosts historical values.

Request should specify host filtering parameters, datetime points, and items. Host filtering parameters are the same as for /api/zabbix-hosts/ endpoint. Input/output format is the same as for /api/zabbix-hosts/<host_uuid>/items_history/ endpoint.

/api/zabbix-hosts/items_aggregated_values/

Methods: GET

Representation of Zabbix hosts and related actions. Get sum of aggregated hosts values.

Request parameters:

- ?start – start of aggregation period as timestamp. Default: 1 hour ago.
- ?end – end of aggregation period as timestamp. Default: now.
- ?method – aggregation method. Default: MAX. Choices: MIN, MAX.
- ?item – item key. Can be list. Required.

Response format: {<item key>: <aggregated value>, ...}

Endpoint will return status 400 if there are no hosts or items that match request parameters.

/api/zabbix-hosts/<uuid>/

Methods: GET, POST, PUT, PATCH, DELETE

Representation of Zabbix hosts and related actions.

```
/api/zabbix-hosts/<uuid>/items_history/
```

Methods: GET

Representation of Zabbix hosts and related actions. Get host items historical values.

Request should specify datetime points and items. There are two ways to define datetime points for historical data.

1. Send *?point=<timestamp>* parameter that can list. Response will contain historical data for each given point in the same order.
2. Send *?start=<timestamp>*, *?end=<timestamp>*, *?points_count=<integer>* parameters. Result will contain *<points_count>* points from *<start>* to *<end>*.

Also you should specify one or more name of host template items, for example 'openstack.instance.cpu_util'

Response is list of datapoints, each of which is dictionary with following fields:

- 'point' - timestamp;
- 'value' - values are converted from bytes to megabytes, if possible;
- 'item' - key of host template item;
- 'item_name' - name of host template item.

```
/api/zabbix-hosts/<uuid>/items_values/
```

Methods: GET

Representation of Zabbix hosts and related actions. The same as *items_aggregated_values*, only for one host

License

NodeConductor Zabbix plugin is open-source under MIT license.

Indices and tables

- `genindex`
- `search`

S

SMS_SETTINGS, 3