NodeConductor Organization Documentation

Release 0.3.0

OpenNode

December 05, 2016

Contents

Organizations allow to separate users into different groups and connect them to the customers.

CHAPTER 1

Guide

1.1 Installation

- Install NodeConductor
- Clone NodeConductor Organization repository

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

- Install NodeConductor Organization into NodeConductor virtual environment

```
cd /path/to/nodeconductor-organization/
python setup.py install
```

1.2 Joining organization

Prerequisites

- User do not have any organization user.
- User wants to join the organization.

Steps

1. User creates organization user providing URL of the organization he wants to join.
2. Before user will be member of the organization, his organization user should be approved.
3. Staff user or owner of the organization's customer approves or rejects the organization user.
4. When organization user is approved, user becomes member of the organization.

Note that user can delete his organization user only when it is not approved yet.

1.3 Leaving organization

Prerequisites

- User has organization user in approved state.
- User wants to leave current organization.

Steps

1. User contacts staff user or owner of the organization's customer he wants to leave.
2. Staff user or owner of the organization's customer rejects or removes organization user.
3. User is not member of the organization anymore.

CHAPTER 2

API

2.1 List organizations

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

Filtering organization list is supported through HTTP query parameters, the following fields are supported:

- ?name=<organization name>
- ?native_name=<organization native name>
- ?abbreviation=<organization abbreviation>
- ?customer_uuid=<customer uuid>
- ?customer=<customer url>

Sorting is supported in ascending and descending order by specifying a field to an **?o=** parameter.

Ascending:

- ?o=name - sort by name
- ?o=native_name - sort by native name
- ?o=abbreviation - sort by abbreviation

Descending:

- ?o=-name - sort by name
- ?o=-native_name - sort by native name
- ?o=-abbreviation - sort by abbreviation

2.2 Create a organization

To create a new organization, issue a POST with organization details to **/api/organizations/** as a staff user.

Request parameters:

- name - organization name (required)
- native_name - organization native name (required)
- abbreviation - organization abbreviation (required and unique)
- customer - URL of organization customer (optional)

Example of a request:

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

{
    "customer": "http://example.com/api/customers/8bdbcd5be4d5452db1390199fa0a4756/",
    "name": "My organization",
    "abbreviation": "MO",
    "native_name": "Minu organisatsioon"
}
```

2.3 Display organization

To get organization data - issue GET request against **/api/organizations/<organization_uuid>/**.

Example rendering of the organization object for staff user:

```
{
    "url": "http://example.com/api/organizations/ab937a628b194c2dba9b414741e918ec/",
    "customer": "http://example.com/api/customers/8bdbcd5be4d5452db1390199fa0a4756/",
    "name": "My organization",
    "uuid": "ab937a628b194c2dba9b414741e918ec",
    "abbreviation": "MO",
    "native_name": "Minu organisatsioon"
}
```

2.4 Delete organization

To delete organization - issue DELETE request against **/api/organizations/<organization_uuid>/** as staff user.

2.5 List organization users

To get list of all organization users - issue GET request against **/api/organization-users/**.

Permissions:

- Staff user can see users of all organizations.
- Customer owner can see users of the connected organization.
- Regular user can see only his organization user.

Response example:

```
[
    {
        "url": "http://example.com/api/organization-users/2de6bc633e56403f9a73c7a3baf9677e/",
        "user": "http://example.com/api/users/ba3f5a1c36a94075a53fd9ab180967de/",
        "username": "Alice",
        "uuid": "2de6bc633e56403f9a73c7a3baf9677e",
        "is_approved": false,
        "organization": "http://example.com/api/organizations/33e17c07683a4d0db0b9376a14b9e2a1/"
    },
    {
        "url": "http://example.com/api/organization-users/701e29fbc30f44c895cafc6848a9bda8/",
        "user": "http://example.com/api/users/95db7c9c9a7c4109b3791a1fa3a7a6a8/",
        "username": "Bob",
        "uuid": "701e29fbc30f44c895cafc6848a9bda8",
        "is_approved": false,
        "organization": "http://example.com/api/organizations/c2a045dfc2bb4cf981cafd35a2e88368/"
    }
]
```

Filtering organization users list is supported through HTTP query parameters, the following fields are supported:

- ?organization=<organization url>
- ?organization_uuid=<organization uuid>
- ?user=<user url>
- ?user_uuid=<user uuid>
- ?is_approved=<True or False>

Sorting is supported in ascending and descending order by specifying a field to an **?o=** parameter.

- ?o=is_approved - approved at the end
- ?o=-is_approved - approved at the beginning

2.6 Create organization user

To create new organization user - issue POST request against **/api/organization-users/**.

Permissions:

- Staff user can create organization user for every user account.
- Regular user can create organization user only for his user account.

Note that every user can have only one user account.

Request parameters:

- user - user url (required, unique)
- organization - organization url (required)

Example of a request:

```
POST /api/organization-users/24156c367e3a41eea81e374073fa1060/ HTTP/1.1
Content-Type: application/json
Accept: application/json
Authorization: Token c84d653b9ec92c6cbac41c706593e66f567a7fa4
Host: example.com

{
    "user": "http://example.com/api/users/95db7c9c9a7c4109b3791a1fa3a7a6a8/",
    "organization": "http://example.com/api/organizations/33e17c07683a4d0db0b9376a14b9e2a1/"
}
```

2.7 Display organization user

To get organization user data - issue GET request against **/api/organization-users/<user_uuid>/**.

Example rendering of the organization object for staff user:

```
{
    "url": "http://example.com/api/organization-users/2de6bc633e56403f9a73c7a3baf9677e/",
    "user": "http://example.com/api/users/ba3f5a1c36a94075a53fd9ab180967de/",
    "username": "Alice",
    "uuid": "2de6bc633e56403f9a73c7a3baf9677e",
    "is_approved": false,
    "organization": "http://example.com/api/organizations/33e17c07683a4d0db0b9376a14b9e2a1/"
}
```

2.8 Approve organization user

To approve user participation in the organization issue POST request against **/api/organization-users/<user_uuid>/approve/** as organization customer owner or staff user. *is_approved* field will change to **True**.

2.9 Reject organization user

To reject user participation in the organization issue POST request against **/api/organization-users/<user_uuid>/reject/** as organization customer owner or staff user. *is_approved* field will change to **False**.

2.10 Delete organization user

To delete organization user - issue DELETE request against **/api/organization-users/<user_uuid>/**. Note that user can delete his organization user only when it is not approved.

CHAPTER 3

Endpoints

3.1 NodeConductor Organization

NodeConductor Organization

3.1.1 /api/organizations/

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

`/api/organizations/`

Methods: `GET`, `POST`
Supported fields for creation:

- **name** – `string`
- **abbreviation** – `string`
- native_name – `string`

Filter fields:

- ?customer = `link`
- ?customer_uuid = `UUIDFilter`

Order fields: `abbreviation-name`, `name`, `native_name`

`/api/organizations/<uuid>/`

Methods: `GET`, `PUT`, `PATCH`, `DELETE`
Supported fields for update:

- **name** – `string`
- **abbreviation** – `string`
- native_name – `string`

3.1.2 /api/organization-users/

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

`/api/organization-users/`

Methods: `GET`, `POST`

Supported fields for creation:

- **user** – `link to /api/users/<uuid>/`
- is_approved – `boolean`
- **organization** – `link to /api/organizations/<uuid>/`

Filter fields:

- ?organization_uuid = `UUIDFilter`
- ?organization = `link`
- ?user_uuid = `UUIDFilter`
- ?user = `link`

Order fields: `is_approved`

`/api/organization-users/<uuid>/`

Methods: `GET`, `DELETE`

`/api/organization-users/<uuid>/approve/`

Methods: `POST`

`/api/organization-users/<uuid>/reject/`

Methods: `POST`

CHAPTER 4

License

NodeConductor Organization plugin is open-source under MIT license.

CHAPTER 5

Indices and tables

- genindex
- search