



Polemarch Documentation

Release 3.0.3

VST Consulting

Jul 22, 2023

CONTENTS

1	Installation	3
1.1	Install from PyPI	3
1.2	Install from docker	7
1.3	Examples	8
1.4	Quickstart	9
1.5	Backup	9
1.6	Update	9
1.7	Update from any old version	9
1.8	Migrate	10
1.9	Create superuser	10
1.10	Change password	10
2	Configuration manual	11
2.1	Introduction	11
2.2	Project architecture	11
2.3	Polemarch clustering overview	12
2.4	Main settings	13
2.5	Database settings	14
2.6	Cache settings	15
2.7	Locks settings	15
2.8	Session cache settings	15
2.9	Rpc settings	15
2.10	Worker settings	16
2.11	Web settings	16
2.12	Centrifugo client settings	16
2.13	Git settings	17
2.14	Archive settings	17
2.15	History output plugins	17
2.16	Production web settings	17
2.17	Configuration options	18
2.18	Inventory plugins config	18
2.19	Execution plugins config	18
3	GUI workflow	21
3.1	Welcome to Polemarch GUI	21
3.2	Before you start	22
3.3	Projects	22
3.4	Project variables	28
3.5	Ansible Module execution	32
3.6	Execution templates	36
3.7	Periodic tasks	44
3.8	Inventory	49
3.9	Import inventory	70
3.10	.polemarch.yaml	77

3.11	Community project samples	83
3.12	Hooks	89
3.13	Users	91
4	Plugins	97
4.1	Execution plugins	97
4.2	Inventory plugins	99
4.3	API reference	101
5	API	107
5.1	Structure	107
6	How to participate	273
7	VST Consulting contributor agreement	275
7.1	1. Definitions	275
7.2	2. Copyright Assignment, License and Waiver	275
7.3	3. Patent License	276
7.4	4. Your Representation and Warranties.	276
7.5	5. Intellectual Property	276
7.6	Miscellaneous	276
	HTTP Routing Table	277
	Index	283

Polemarch is a web interface that makes it easy for IT teams to work with ansible. It's designed to be compatible with all other projects using the ansible cli without complicated adaptation. Polemarch collects all ansible best practices for organizing the code of infrastructure projects and adds such features as execution of scheduled tasks, creating execution templates (instead of complex bash/powershell scripts). Each launch is logged and stored in the history to control and monitor the execution.

Polemarch allows you to control access to infrastructure, credentials without having to transfer these data. You can set up Inventory so that different projects only have access to the host they need. This is possible by reusing hosts and groups, as well as a set of their variables in different inventories. All variables are stored in the database, and key variables (such as passwords and keys) after loading are stored on the server without showing the user their contents.

Since Polemarch is a web app, you can run the infrastructure remotely without actually having access to your infrastructure. This may be handy for remote teams. Besides in some scripts you can organize infrastructure management work without employees having access to the interface.

Polemarch developers have tried to provide the maximum set of options for extending functionality through web or script hooks, as well as through customization of the cli utility for launching playbooks/modules (for example, running in an isolated chroot/docker environment). The well-documented REST API allows the user to integrate Polemarch with any product that can make remote requests.

The project implies maximum compatibility with ansible cli (specifically with ansible and ansible-playbook, at the moment), so it doesn't require special knowledge other than that required to work with ansible. It is not the intention of this documentation to describe all use cases, only those that are necessary to understand how the web interface works. In some cases it's worth having official ansible documentation.

You could find latest releases [there](#). Also you could find source code in [official repository](#).

Contact us on [any questions](#) or [report bugs](#).

Official site: <https://polemarch.org/>

We also help you via:

[Stack Overflow](#) - tag for questions about our application on Stack Overflow

INSTALLATION

1.1 Install from PyPI

1. Install dependencies:

Required packages on Ubuntu 18.04:

```
sudo apt-get install python3-virtualenv python3.8 python3.8-dev gcc libffi-dev  
↳libkrb5-dev libffi6 libssl-dev libyaml-dev libsasl2-dev libldap2-dev default-  
↳libmysqlclient-dev sshpass git virtualenv
```

Required packages on Ubuntu 20.04:

```
sudo apt-get install python3-virtualenv python3.8 python3.8-dev gcc libffi-dev  
↳libkrb5-dev libffi7 libssl-dev libyaml-dev libsasl2-dev libldap2-dev default-  
↳libmysqlclient-dev sshpass git
```

Required packages on Ubuntu 22.04:

```
sudo apt-get install python3-virtualenv python3.10 python3.10-dev gcc libffi-dev  
↳libkrb5-dev libffi7 libssl-dev libyaml-dev libsasl2-dev libldap2-dev default-  
↳libmysqlclient-dev sshpass git
```

Required packages on Debian 11 (as root user):

```
apt-get install python3-virtualenv python3.9 python3.9-dev gcc libffi-dev  
↳libkrb5-dev libffi7 libssl-dev libyaml-dev libsasl2-dev libldap2-dev default-  
↳libmysqlclient-dev sshpass git
```

Required packages on Red Hat/CentOS 7 (sqlite not supported):

```
sudo yum install epel-release centos-release-scl-rh centos-release-scl  
sudo yum --enablerepo=centos-scl-rh install rh-python38 rh-python38-python-  
↳devel gcc openssl-devel libyaml-devel krb5-devel krb5-libs openldap-devel  
↳mysql-devel git sshpass  
sudo /opt/rh/rh-python38/root/usr/bin/python -m pip install virtualenv
```

Required packages on Red Hat/Alma/Rocky 8:

```
sudo dnf install epel-release  
sudo dnf install python38-devel python3-virtualenv gcc openssl-devel libyaml  
↳krb5-devel krb5-libs openldap-devel mysql-devel git sshpass
```

Note: If your OS is not in the list of presented OS, but you understand the differences between these OS and yours, then you can adapt this list of packages to your platform. We don't tie environment to system package

versions as much as possible.

2. Install mysql-server:

The following command is suitable for Debian and Ubuntu. If you have a different operation system, you can use [official documentation](#).

```
sudo apt-get install default-mysql-server
```

Warning: Do not use MySQL version less than 8.0.

3. Create database in mysql with this commands:

```
sudo -H mysql <<QUERY_INPUT
# uncomment this string on old MariaDB/MySQL versions
# SET @@global.innodb_large_prefix = 1;
create user db_user identified by 'db_password';
create database db_name default CHARACTER set utf8 default COLLATE utf8_general_
→ci;
grant all on db_name.* to 'db_user';
QUERY_INPUT
```

Note: You should do it on database host if you connect to remote server.

4. Then, if you use mysql and you have set timezone different from “UTC” you should run next command:

```
mysql_tzinfo_to_sql /usr/share/zoneinfo | sudo -H mysql mysql
```

Note: You should do it on database host if you connect to remote server.

5. Create user:

```
sudo useradd --user-group --create-home --shell /bin/bash polemarch
```

Hint: You can add this user to sudoers for easier installation process and support.

6. Create virtualenv and activate it:

```
# In some cases use sudo for first command.
# For rhel/centos7 use:
/opt/rh/rh-python38/root/usr/bin/python -m virtualenv /opt/polemarch
# For Debian with Python 3.9:
virtualenv --python=python3.9 /opt/polemarch
# For Ubuntu 22.04 and other debian distributions with Python 3.10:
virtualenv --python=python3.10 /opt/polemarch
# For other distributions:
virtualenv --python=python3.8 /opt/polemarch

# Make required directories
sudo mkdir -p /etc/polemarch
sudo chown -R polemarch:polemarch /opt/polemarch /etc/polemarch
```

(continues on next page)

(continued from previous page)

```
sudo -u polemarch -i
source /opt/polemarch/bin/activate
```

Note: If you have more than one Python version, recommended use Python 3.8 or newer for virtualenv.

7. Install Polemarch:

```
pip install -U polemarch[mysql]
```

8. Edit config file:

1. Create directory for *log* and *pid* files:

```
mkdir /opt/polemarch/logs /opt/polemarch/pid
```

2. Open */etc/polemarch/settings.ini*, if it does not exist, create it. Polemarch uses config from this directory.
3. The default database is SQLite3, but MySQL is recommended. Settings needed for correct work MySQL database:

```
[database]
engine = django.db.backends.mysql
name = db_name
user = db_user
password = db_password

[database.options]
connect_timeout = 20
init_command = SET sql_mode='STRICT_TRANS_TABLES', default_storage_
engine=INNODB, NAMES 'utf8', CHARACTER SET 'utf8', SESSION collation_
connection = 'utf8_unicode_ci'
```

Note: Set host and port settings if you connect to remote server.

Warning: If you are using MariaDB, make sure that your *settings.ini* config contains next line:

```
[databases]
databases_without_cte_support = default
```

The problem is that the implementation of recursive queries in the MariaDB does not allow using it in a standard form. MySQL (since 8.0) works as expected.

4. The default cache system is file based cache, but RedisCache is recommended. Settings needed for correct RedisCache work:

```
[cache]
backend = django.core.cache.backends.redis.RedisCache
location = redis://127.0.0.1:6379/1

[locks]
backend = django.core.cache.backends.redis.RedisCache
location = redis://127.0.0.1:6379/2
```

Note: Set host ip and port instead of 127.0.0.1:6379 if you connect to remote server.

5. The default celery broker is file Celery broker, but Redis is recommended. Settings needed for correct Redis work:

```
[rpc]
connection = redis://127.0.0.1:6379/3
heartbeat = 5
concurrency = 8
enable_worker = true
```

Note: Set host ip and port instead of 127.0.0.1:6379 if you connect to remote server.

Hint: Use RabbitMQ in case there can be a big network delay between the Polemarch nodes.

6. For running Polemarch with worker, you need to create follow sections:

```
[uwsgi]
pidfile = /opt/polemarch/pid/polemarch.pid
log_file = /opt/polemarch/logs/polemarch_web.log

# Uncomment it for HTTPS:
# [uvicorn]
# ssl_keyfile = /etc/polemarch/polemarch.key
# ssl_certfile = /etc/polemarch/polemarch.crt

[worker]
# output will be /opt/polemarch/logs/polemarch_worker.log
logfile = /opt/polemarch/logs/{PROG_NAME}_worker.log
# output will be /opt/polemarch/pid/polemarch_worker.pid
pidfile = /opt/polemarch/pid/{PROG_NAME}_worker.pid
loglevel = INFO
```

Also if you need to set your own path for logfile or pidfile, different from the path from example, you can do it, but make sure, that user, which starts Polemarch has write-permissions for these directory and file. If you run it as root, we recommend to add in [uwsgi] params uid and gid ([read more](#)).

Tip: More configuration settings you can find in *Configuration manual*.

9. Make migrations:

```
polemarchctl migrate
```

Note: The first time run this command, the first superuser `admin` will be created in the database with the same password. We recommend changing the user's password immediately after the first login.

1. Start Polemarch:

```
polemarchctl webserver
```

Polemarch starts with web interface on port 8080.

If you need to restart Polemarch use following command:

```
polemarchctl webserver reload=/opt/polemarch/pid/polemarch.pid
```

If you use another directory for storing Polemarch pid file, use path to this file.

If you need to stop Polemarch use following command:

```
polemarchctl webserver stop=/opt/polemarch/pid/polemarch.pid
```

If you use another directory for storing Polemarch pid file, use path to this file.

1.2 Install from docker

1.2.1 Run image

For run Polemarch docker image use command:

```
docker run -d --name polemarch --restart always -v /opt/polemarch/projects:/  
↳projects -v /opt/polemarch/hooks:/hooks -p 8080:8080 vstconsulting/  
↳polemarch
```

Using this command download official docker image and run it with default settings. Dont use default SQLite installation with filecache in production.

Ensure, that `/opt/polemarch/projects` and `/opt/polemarch/hooks` has uid/gid `1000/1000` as owner. Polemarch will be run with web interface on port `8080`

1.2.2 Settings

1.2.3 Main section

- **DEBUG** - status of debug mode. Default value: *false*.
- **DJANGO_LOG_LEVEL** - log level. Default value: *WARNING*.
- **TIMEZONE** - timezone. Default value: *UTC*.

1.2.4 Database section

Setup database connection via `django-environ`: `environ.Env.db_url`.

For example for mysql, **DATABASE_URL** = `'mysql://user:password@host:port/dbname'`. Read more about `django-environ` in the [official django-environ documentation](#).

1.2.5 Cache

For cache environment variables you can also use `django-environ` - `environ.Env.cache_url`. For example for redis, **CACHE_URL** = `redis://host:port/dbname`.

1.2.6 RPC section

- **POLEMARCH_RPC_ENGINE** - connection to rpc service. If not set used as tmp-dir.
- **POLEMARCH_RPC_RESULT_BACKEND** - connection to rpc results service. Default as engine.
- **POLEMARCH_RPC_HEARTBEAT** - Timeout for RPC. Default value: 5.
- **POLEMARCH_RPC_CONCURRENCY** - Number of concurrently tasks. Default value: 4.

1.2.7 Web section

- **POLEMARCH_WEB_REST_PAGE_LIMIT** - Limit elements in answer, when send REST request. Default value: 1000.

1.2.8 UWSGI section

- **POLEMARCH_UWSGI_PROCESSES** - number of uwsgi processes. Default value: 4.

1.2.9 Other settings

If you set *WORKER* to *ENABLE* state, uwsgi run worker as daemon.

If you set *SECRET_KEY*, value of *SECRET_KEY* variable would be written to *secret*

1.3 Examples

Run latest version of Polemarch in docker and connect to MySQL on server, using django-enviro:

```
docker run -d --name polemarch --restart always -v /opt/polemarch/projects:/  
→projects -v /opt/polemarch/hooks:/hooks --env DATABASE_URL=mysql://  
→polemarch:polemarch@polemarch_db:3306/polemarch -p 8080:8080 -p  
→vstconsulting/polemarch
```

Run Polemarch with Memcache and RabbitMQ and SQLite3. Polemarch log-level=INFO, secret-key=mysecretkey

```
docker run -d --name polemarch --restart always -v /opt/polemarch/projects:/  
→projects -v /opt/polemarch/hooks:/hooks --env POLEMARCH_RPC_ENGINE=amqp://  
→polemarch:polemarch@rabbitmq-server:5672/polemarch --env CACHE_  
→URL=memcache://memcached-server:11211/ --env POLEMARCH_LOG_LEVEL=INFO --  
→env SECRET_KEY=mysecretkey -p 8080:8080 vstconsulting/polemarch
```

Also you can use *.env* file with all variable you want use on run docker:

```
docker run -d --name polemarch --restart always -v /opt/polemarch/projects:/  
→projects -v /opt/polemarch/hooks:/hooks --env-file /path/to/file -p  
→8080:8080 vstconsulting/polemarch
```

Run from the sources with docker-compose (PoleMarch+MySQL+Redis):

```
export DOCKER_BUILDKIT=1  
export COMPOSE_DOCKER_CLI_BUILD=1  
docker-compose up -d --build
```

1.4 Quickstart

After you install Polemarch by instructions above you can use it without any further configuration. Interface is pretty intuitive and common for any web application. Read more in [GUI workflow](#).

Default installation is suitable for most simple and common cases, but Polemarch is highly configurable system. If you need something more advanced (scalability, dedicated DB, custom cache, logging or directories) you can always configure Polemarch like it is said in [Configuration manual](#).

1.5 Backup

Regular uploading of backups is a guarantee of the reliability of the application. We recommend use SQL backup or copy database.

There are examples of SQL backup for MySQL and PostgreSQL below.

Making backup in MySQL:

```
shell> mysqldump dbname > dump.sql
```

Here dbname is the name of your database, dump.sql is the file, where all SQL backup statements will be saved.

Uploading of backup in MySQL:

```
shell> mysqladmin create dbname
shell> mysql dbname < dump.sql
```

Making backup in PostgreSQL:

```
pg_dump dbname > dump.sql
```

Uploading of backup in PostgreSQL:

```
createdb dbname
psql dbname < dump.sql
```

1.6 Update

Before updating of package of any type it is strongly recommended to stop all services and create backup for safety.

1.7 Update from any old version

1. Firstly, we strongly recommend you to create a database backup and to stop all Polemarch services for safety.
2. Secondly, if you are updating from 0.x.y to 1.x.y, you need to update you current 0.x.y version to 1.0.0 version.
3. Then update to the latest version. If you don't know how to do it, look ["Install from PyPI"](#).

1.8 Migrate

Migrations are Django's way of propagating changes you make to your models (adding a field, deleting a model, etc.) into your database schema. They're designed to be mostly automatic, but you need to know when to make migrations, when to run them, and the common problems you might run into.

To run a migrate command you should run follow code:

```
polemarchctl migrate
```

1.9 Create superuser

A superuser is the user, who has all permissions.

To create a superuser account use the follow command:

```
polemarchctl createsuperuser
```

This command prompts for all required user's options.

1.10 Change password

To change password use the follow command:

```
polemarchctl changepassword [<username>]
```

It prompts you to enter a new password twice for the given user. If the entries are identical, this immediately becomes the new password. If you do not supply a user, the command will attempt to change the password of user whose username matches the current user.

CONFIGURATION MANUAL

2.1 Introduction

Default installation is suitable for most simple and common cases, but Polemarch is highly configurable system. If you need something more advanced (scalability, dedicated DB, custom cache, logging or directories) you can always configure Polemarch deeply by tweaking `/etc/polemarch/settings.ini`.

This manual does not have purpose to describe all possible configuration options in details because `settings.ini` has commentary for every option which makes clear its purpose and possible values. But here is a brief overview of the most important settings to make clear for you big picture: what you can actually customize in Polemarch.

We advice you to read [Polemarch clustering overview](#) if you want to setup cluster of Polemarch nodes to maintain reliability or speedup things. It will give you understanding of services, which are included into Polemarch and how to distribute them between the nodes to reach your goal.

2.2 Project architecture

Polemarch was created to adapt to any work environment. Almost every service can be easily replaced by another without losing any functionality. The application architecture consists of the following elements:

- **Database** supports all types and versions that django can. The code was written to be vendor agnostic to support as many backends as possible. Database contains information about projects settings, schedule and templates of tasks, execution history, authorisation data, etc. Database performance is a key performance limitation of the entire Polemarch.
- **Cache** services is used for store session data, services locks, etc. Also, PM support all of Django can. Mostly, we recommend to use Redis in small and medium clusters.
- **MQ** or rpc engine is required for notifying celery worker about new task execution request. Redis in most cases can process up to 1000 executions/min. For more complex and high-load implementations, it is recommended to use a distributed RabbitMQ cluster. If technically possible, AWS SQS and its compatible counterparts from other cloud providers are also supported.
- **Centrifugo** (optional) is used for active user interaction. At this point, the service notifies the user of an update or change to the data structure that the user is viewing to complete a data update request. This reduces the load on the database, because without this service, the interface makes periodic requests on a timer.
- **Project storage** at now is directory in filesystem where PM clone or unarchive project files for further executions. Storage must be readable for web-server and writeable for celery worker. It can be mounted dir from shared storage.

Understanding what services the Polemarch application consists of, you can build any architecture of services suitable for the circumstances and infrastructure.

2.3 Polemarch clustering overview

Polemarch actually consists of two services: web-server and worker. Polemarch uses worker for long-running tasks (such as `ansible-playbook` runs, repo synchronizations and so on). Those services are designed as systemd services you can control using regular distro-tools for service manipulation. You can run more than one server with those services. In default configuration those services use local file system to keep data and exchange with each other, but for multiple nodes they must be configured to use shared client-server database, cache server and network filesystem (for multiple workers). All those settings are described in appropriate sections of this documentation. It is up to you to make sure that configuration is identical on every node to prevent discordant behaviour of nodes. If you have multiple web-servers, don't forget to setup HAProxy or something similar for balancing load between them.

Lets assume, that you want to create 2 servers with web-part of Polemarch to maintain reliability of your admin-panel and 4 servers with workers to prevent denial for service because of overloading. Then briefly (it is mostly example than general howto) you must do such steps:

1. Install Polemarch from PyPI at every server with worker and web-server by [installation instructions](#). We recommend to install virtual environment in `/opt/polemarch` and set as owner user `polemarch` (need to be created).
2. Setup some network filesystem (NFS, Samba, GlusterFS, Ceph). NFS, for example. Mount it in the same directory of all worker-intended nodes. Write this directory in [Main settings](#). Example:

```
[main]
projects_dir = /mnt/mystorage/projects
hooks_dir = /mnt/mystorage/hooks
```

3. Setup some http-balancer. HAProxy, for example. Point it to web-intended nodes.

Hint: You can setup ssl for Polemarch endpoints in this step.

4. Create polemarch systemd service:

1. Firstly, create a file `/etc/systemd/system/polemarch.service`:

```
[Unit]
Description=Polemarch Service HTTP Server
After=network.target remote-fs.target nss-lookup.target redis.service_
↳mysql.service

[Service]
Type=forking
ExecStart=/opt/polemarch/bin/polemarchctl webserver
ExecReload=/opt/polemarch/bin/polemarchctl webserver reload=/opt/
↳polemarch/pid/web.pid
ExecStop=/opt/polemarch/bin/polemarchctl webserver stop=/opt/
↳polemarch/pid/web.pid
PIDFile=/opt/polemarch/pid/web.pid
User=polemarch
Group=polemarch
KillSignal=SIGCONT
Restart=always

# Uncomment this if used privileged ports
# Capabilities=CAP_NET_BIND_SERVICE+ep
# AmbientCapabilities=CAP_NET_BIND_SERVICE
```

(continues on next page)

(continued from previous page)

```
[Install]
WantedBy=multi-user.target
```

Note: Notice, that user and group ‘polemarch’ should exist in your system. If they don’t exist, create them.

2. Reload systemctl daemon:

```
systemctl daemon-reload
```

3. Add polemarch.service to autoload:

```
systemctl enable polemarch.service
```

4. Start polemarch.service:

```
systemctl start polemarch.service
```

5. Repeat all steps in other nodes and connect them to one DB, cache, MQ and storage.

Note: You don’t need migrate database on each node. This need only once when you install/update first node in cluster.

Warning: Don’t forget to stop all Polemarch services when update polemarch package.

That’s it.

2.4 Main settings

Section [main].

This section is for settings related to whole Polemarch (both worker and web). Here you can specify verbosity level of Polemarch during work, which can be useful for troubleshoot problems (logging level etc). Also there are settings for changing of timezone for whole app and directory where Polemarch will store ansible projects cloned from repositories.

If you want to use LDAP protocol, you should create next settings in section [main].

```
ldap-server = ldap://server-ip-or-host:port
ldap-default-domain = domain.name
ldap-auth_format = cn=<username>,ou=your-group-name,<domain>
```

ldap-default-domain is an optional argument, that is aimed to make user authorization easier (without input of domain name).

ldap-auth_format is an optional argument, that is aimed to customize LDAP authorization request. Default value: cn=<username>,<domain>

So in this case authorization logic will be the following:

1. System checks combination of login:password in database;
2. System checks combination of login:password in LDAP:

- if domain was mentioned, it will be set during authorization (if user enter login without `user@domain.name` or without `DOMAIN\user`);
- if authorization was successful and there is user with mentioned login in database, server creates session for him.
- **debug** - Enable debug mode. Default: `false`.
- **allowed_hosts** - Comma separated list of domains, which allowed to serve. Default: `*`.
- **ldap-server** - LDAP server connection.
- **ldap-default-domain** - Default domain for auth.
- **timezone** - Timezone of web-application. Default: `UTC`.
- **log_level** - Logging level. Default: `WARNING`.
- **projects_dir** - Path where projects will be stored.
- **hooks_dir** - Path where hook scripts stored.
- **executor_path** - Path for polemarch-ansible wrapper binary.
- **enable_django_logs** - Enable or disable Django logger to output. Useful for debugging. Default: `false`.
- **enable_user_self_remove** - Enable or disable user self-removing. Default: `false`.
- **auth-cache-user** - Enable or disable user instance caching. It increases session performance on each request but saves model instance in unsafe storage (default django cache). The instance is serialized to a string using the [standard python module pickle](#) and then encrypted with [Vigenère cipher](#). Read more in the [vstutils.utils.SecurePickling](#) documentation. Default: `false`.

2.5 Database settings

Section [database].

Here you can change settings related to database system, which Polemarch will use. Polemarch supports all databases supported by django. List of supported out of the box: SQLite (default choice), MySQL, Oracle, or PostgreSQL. Configuration details you can look at [Django database documentation](#). If you run Polemarch at multiple nodes (clusterization), you should use some of client-server database (SQLite not suitable) shared for all nodes.

If you use MySQL there is a list of required settings, that you should create for correct database work.

Firstly, if you use MariaDB and you have set timezone different from “UTC” you should run next command:

```
mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql -u root -p mysql
```

Secondly, for correct MariaDB work you should set next options in `settings.ini` file:

```
[database.options]
connect_timeout = 10
init_command = SET sql_mode='STRICT_TRANS_TABLES', default_storage_engine=INNODB,
↳ NAMES 'utf8', CHARACTER SET 'utf8', SESSION collation_connection = 'utf8_unicode_ci'
```

Finally, you should add some options to MariaDB configuration:

```
[client]
default-character-set=utf8
init_command = SET collation_connection = @@collation_database

[mysqld]
```

(continues on next page)

(continued from previous page)

```
character-set-server=utf8
collation-server=utf8_unicode_ci
```

Note: You can find more database options in [Databases settings](#).

2.6 Cache settings

Section [cache].

This section is for settings related to cache backend used by Polemarch. Polemarch supports all cache backends that Django supports. Currently is: filesystem, in-memory, memcached out of the box and many more by additional plugins. You can find details about cache configuration at [Django caches documentation](#). In clusterization scenario we advice to share cache between nodes to speedup their work using client-server cache realizations. We recommend to use Redis in production environments.

2.7 Locks settings

Section [locks].

Locks is system that Polemarch uses to prevent damage from parallel actions working on something simultaneously. It is based on Django cache, so there is another bunch of same settings as [Cache](#). And why there is another section for them, you may ask. Because cache backend used for locking must provide some guarantees, which does not required to usual cache: it **MUST** be shared for all Polemarch threads and nodes. So, in-memory backend, for example, is not suitable. In case of clusterization we strongly recommend to use Redis or Memcached as backend for that purpose. Cache and locks backend can be same, but don't forget about requirement we said above.

2.8 Session cache settings

Section [session].

Polemarch store sessions in [Database settings](#), but for better performance, we use a cache-based session backend. It is based on Django cache, so there is another bunch of same settings as [Cache](#). By default, settings getted from [Cache](#).

2.9 Rpc settings

Section [rpc].

Polemarch uses Celery for long-running tasks (such as `ansible-playbook` runs, repo synchronizations and so on). Celery is based on message queue concept, so between web-service and workers running under Celery must be some kind of message broker (RabbitMQ or something). Those settings relate to this broker and Celery itself. Those kinds of settings: broker backend, number of worker-processes per node and some settings used for troubleshoot server-broker-worker interaction problems.

- **connection** - Celery broker connection. Read more: [Broker Settings](#). Default: `filesystem:///var/tmp`.
- **concurrency** - Celery count worker threads. Default: `4`.
- **heartbeat** - Interval between sending heartbeat packages, which says that connection still alive. Default: `10`.

- **enable_worker** - Enable or disable worker with webserver. Default: `true`.
- **clone_retry_count** - Retries count on project sync operation.

Note: You can find more RPC options in [Rpc settings](#).

2.10 Worker settings

Section [worker].

Celery worker options for start. Useful settings:

- **loglevel** - Celery worker logging level. Default: from main section `log_level`.
- **pidfile** - Celery worker pidfile. Default: `/run/polemarch_worker.pid`
- **autoscale** - Options for autoscaling. Two comma separated numbers: max,min.
- **beat** - Enable or disable celery beat scheduler. Default: `true`.

Other settings can be getted from command `celery worker --help`.

2.11 Web settings

Section [web].

Here placed settings related to web-server. Those settings like: `session_timeout`, `static_files_url` or `pagination limit`.

- **session_timeout** - Session life-cycle time. Default: `2w` (two weeks).
- **rest_page_limit** - Default limit of objects in API list. Default: `1000`.
- **history_metrics_window** - Timeframe in seconds of collecting execution history statuses. Default: `1min`.

Note: You can find more Web options in [Web settings](#).

2.12 Centrifugo client settings

Section [centrifugo].

To install app with centrifugo client, [centrifugo] section must be set. Centrifugo is used by application to auto-update page data. When user change some data, other clients get notification on `subscriptions_update` channel with model label and primary key. Without the service all GUI-clients get page data every 5 seconds (by default). Centrifugo server v3 is supported.

- **address** - Centrifugo api address. For example, `http://localhost:8000/api`.
- **public_address** - Centrifugo server address. By default used **address** without `/api` prefix (`http` -> `ws`, `https` -> `wss`). Also, can be used relative path, like `/centrifugo`.
- **api_key** - API key for clients.
- **token_hmac_secret_key** - API key for jwt-token generation.
- **timeout** - Connection timeout.
- **verify** - Connection verification.

Note: These settings also add parameters to the OpenApi schema and change how the auto-update system works in the GUI. `token_hmac_secret_key` is used for jwt-token generation (based on session expiration time). Token will be used for Centrifugo-JS client.

Note: `api_key` and `token_hmac_secret_key` come from `config.json` for Centrifugo. Read more in [Official Centrifugo documentation](#)

2.13 Git settings

Sections `[git.fetch]` and `[git.clone]`.

Options for git commands. See options in `git fetch --help` or `git clone --help`.

2.14 Archive settings

Section `[archive]`.

Here you can specify settings used by archive (e.g. TAR) projects.

- **max_content_length** - Maximum download file size. Format: `30<unit>`, where unit is *b*, *kb*, *mb*, *gb*, *tb*.

2.15 History output plugins

Section `[history]`

This section of the configuration provides to configure the output history plugin settings.

- **output_plugins** - a comma-separated list of plugin names that are used to record history lines. Plugins must have the `writable` attribute. Default: `database`
- **read_plugin** - the name of the plugin used to display the history lines in the api. Default is `database`.

Other parameters are set in the plugin options section: `history.plugin.PLUGIN_NAME.options`.

Warning: Be careful. The reader plugin must be able to read the data. Therefore, the storage from which the reading plugin takes data must be filled with one of the writer plugins.

2.16 Production web settings

Section `[uwsgi]`.

Here placed settings related to web-server used by Polemarch in production (for deb and rpm packages by default). Most of them related to system paths (logging, PID-file and so on).

Note: More settings in [Configuring uWSGI](#) (deprecated) and [uvicorn docs](#).

Warning: In production, it is recommended to use Centrifugo in order to reduce the load on the backend from automatic page updates.

2.17 Configuration options

This section contains additional information for configure additional elements.

#. If you need to set `https` for your web settings, you can do it using HAProxy, Nginx or configure it in `settings.ini`.

```
# [uvicorn]
# ssl_keyfile = /etc/polemarch/polemarch.key
# ssl_certfile = /etc/polemarch/polemarch.crt
```

1. We strictly do not recommend running the web server from root. Use HTTP proxy to run on privileged ports.

Note: If you need more options you can find it in [Configuration manual](#) in the official vstutils documentation.

2.18 Inventory plugins config

To connect an inventory plugin to Polemarch, there should be a section

```
[inventory.plugin.<plugin_name>]
backend = import.path.to.plugin.Class
```

Where

- **<plugin_name>** - name that will be available in API to work with
- **backend** - is a python import path to plugin class

Also you may add options which will be available in plugin:

```
[inventory.plugin.<plugin_name>.options]
some_option = some_option
```

To read more about plugins, please see *Plugins*.

2.19 Execution plugins config

To connect an execution plugin to Polemarch, there should be a section

```
[execution.plugin.<plugin_name>]
backend = import.path.to.plugin.Class
compatible_inventory_plugins = <inventory_plugin1>,<inventory_plugin1>
```

Where

- **<plugin_name>** - name that will be available in API to work with
- **backend** - is a python import path to plugin class
- **compatible_inventory_plugins** - inventory plugins which are compatible with this execution plugin.
If omitted,
it's supposed that execution plugin cannot work with any inventory.

Also you may add options which will be available in plugin:

```
[execution.plugin.<plugin_name>.options]
some_option = some_option
```

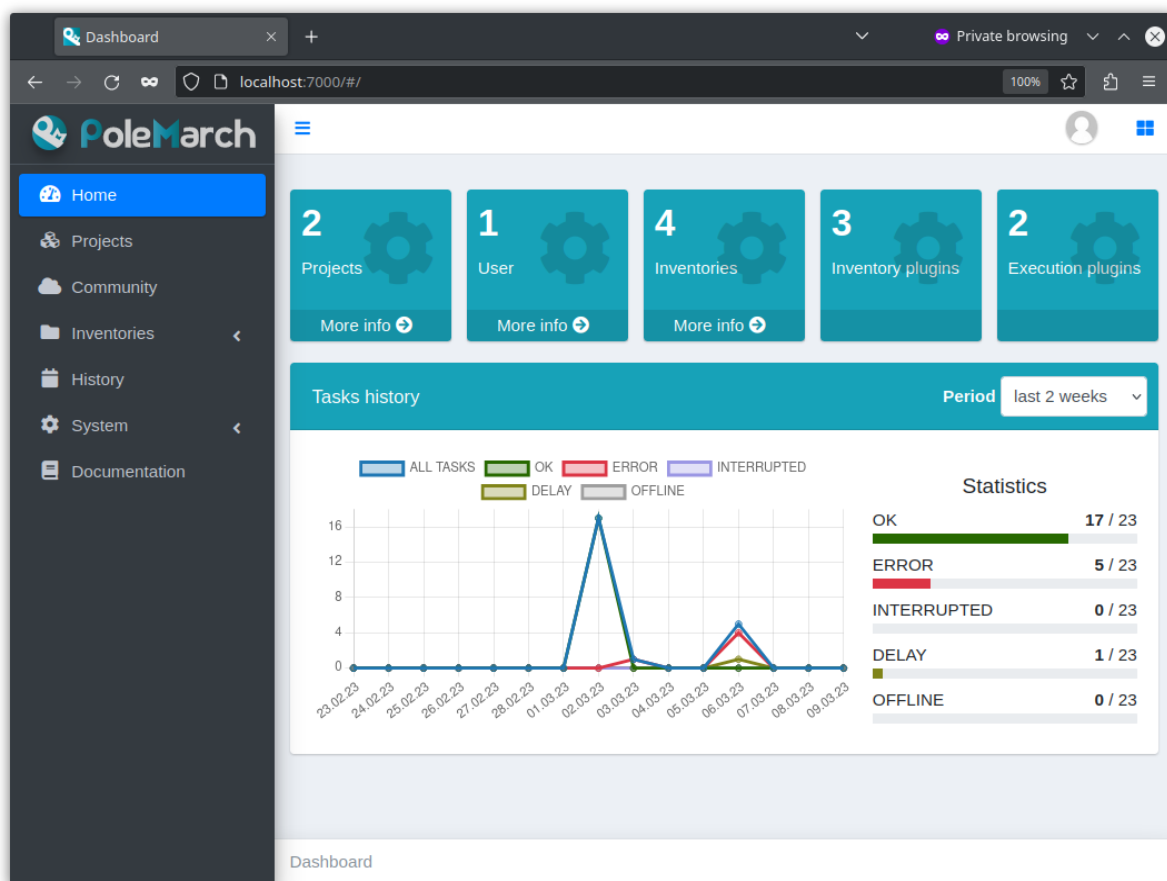
To read more about plugins, please see [*Plugins*](#).

GUI WORKFLOW

3.1 Welcome to Polemarch GUI

In this section of our documentation we will tell you about Polemarch GUI's opportunities.

Let's begin with Dashboard page:



As you can see, Polemarch GUI provides user with 2 menus:

- the first one is located in the left sidebar and it is aimed to provide user with navigation between main system objects, like projects, inventories, history records and etc.
- the second one is located at the top of browser window and it is aimed to navigate user to API section, to user's page and to logout page.

Also there are some widgets on content block of Dashboard page. For example, there you can check the statistics of your tasks for different periods of time.

3.2 Before you start

Before you can do any job with Polemarch you should create at least one project, because all Polemarch's functions are linked to the project.

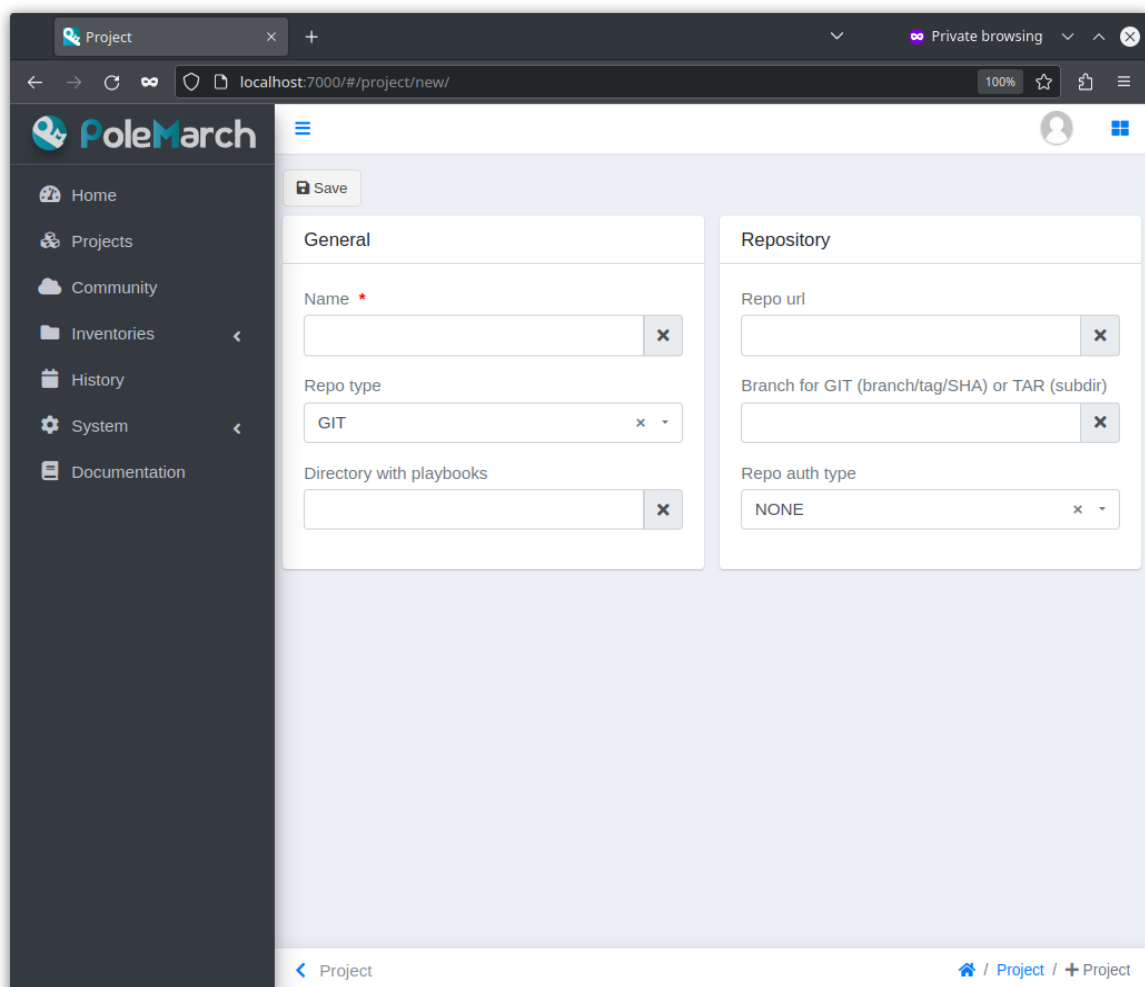
3.3 Projects

So, to start your work with Polemarch you should create project. To do this, you need to select 'Projects' in the left side menu and click the *Create* button.

There are 3 project types in Polemarch:

- GIT - Polemarch can sync with your git project.
- TAR - If you don't have git repository, you can upload tar archive with project files from any http url.
- MANUAL - Empty project. This kind of project will be appropriate for you, if you want run only modules without any playbooks or something. Polemarch will create folder for your project files in *projects_dir* (you can specify it in settings) named as project id (you can see at URL), where you place manually your project files (like playbooks or something).

Let's look at the example of GIT project's creation:



The screenshot shows a web browser window with the Polemarch application running on localhost:7000. The browser is in private browsing mode. The application has a dark sidebar on the left with a menu containing: Home, Projects, Community, Inventories, History, System, and Documentation. The 'Projects' menu item is selected. The main content area is titled 'Project' and contains a 'Save' button at the top left. Below the 'Save' button are two columns of form fields. The left column is titled 'General' and contains: 'Name' (text input), 'Repo type' (dropdown menu with 'GIT' selected), and 'Directory with playbooks' (text input). The right column is titled 'Repository' and contains: 'Repo url' (text input), 'Branch for GIT (branch/tag/SHA) or TAR (subdir)' (text input), and 'Repo auth type' (dropdown menu with 'NONE' selected). At the bottom of the page, there is a breadcrumb trail: < Project / Project / + Project.

As you can see, the form of new project creation consists of 7 fields:

General:

- **Name** - name of your project;
- **Repo type** - type of project repository (GIT, TAR, MANUAL);
- **Directory with playbooks** - you can specify the path to the directory with playbooks.

Repository:

- **Repo URL** - URL to your repository;
- **Branch for GIT (branch/tag/SHA) or TAR (subdir)** - you can specify the branch you want or leave this field blank to get the default branch;
- **Repo auth type** - type of authentication (NONE, KEY, PASSWORD);
- **Repo auth data** - key or password value (if you choose KEY or PASSWORD authentication type);

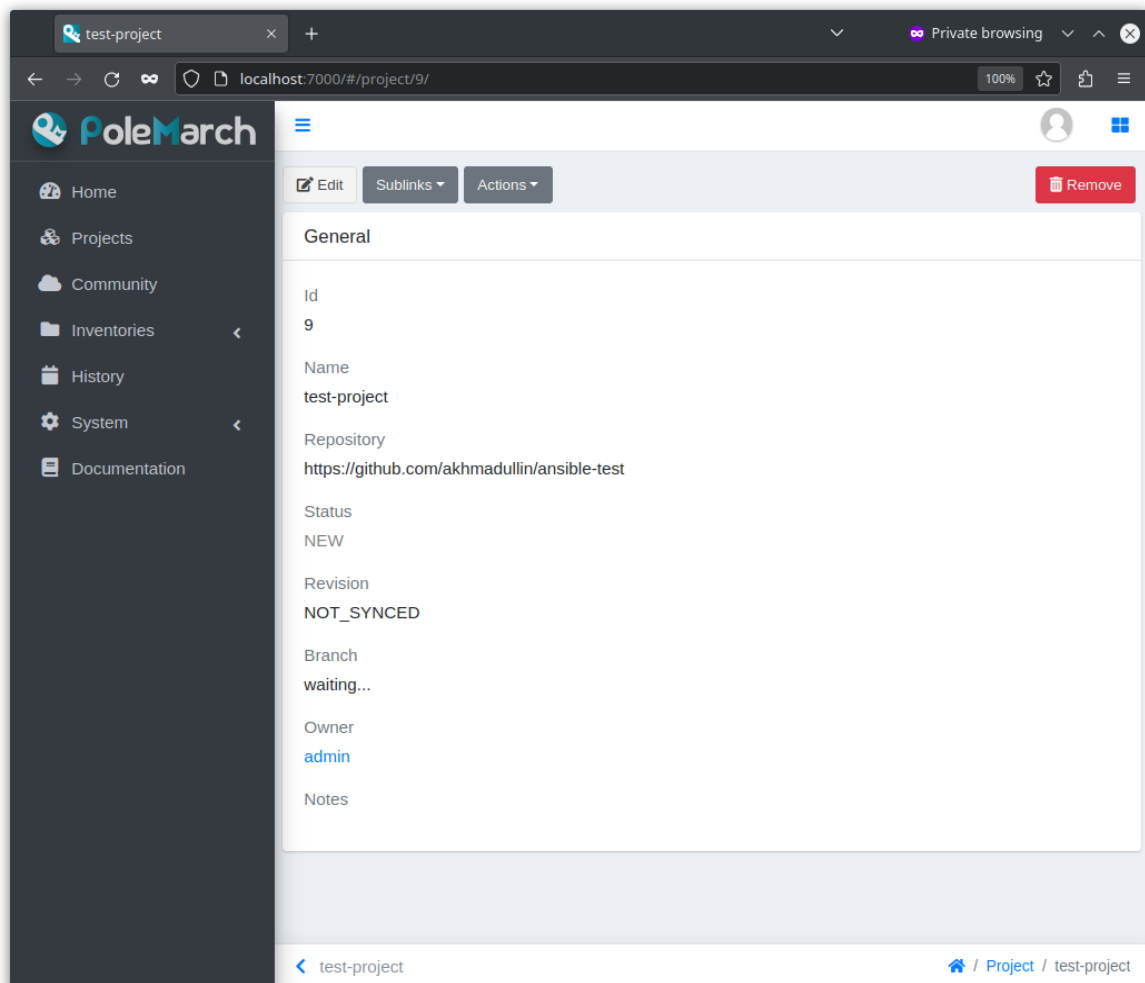
Before we continue talking about the structure of our project, we used [ansible best practice](#) and run playbooks from project:

- `ansible.cfg` - used to setup settings. For detailed information about configuration file read [settings reference](#).

If you setup modules directory, we import all modules from this directory and set name `polemarch.project.[module_name]`.

- Playbook - import all files with `.yaml` extension from root directory of the **Project**.
- If you need to set variables globally, not only for run, you can use `group_vars` and `host_vars` in root directory of project.

After project creation you'll see the next page:



As you can see there are some new fields on this page:

- **Id**: the unique key of object in database.
- **Status**: Polemarch project status. Possible values are:
 - NEW - newly created project,
 - WAIT_SYNC - repository synchronization has been scheduled, but has not started to perform yet,
 - SYNC - synchronization is in progress,
 - ERROR - synchronization failed,
 - OK - project is synchronized.
- **Revision** - GIT project revision.
- **Branch** - branch of your GIT project, to what your Polemarch project was synced.
- **Owner** - user, who owns this project.

Also there are some new buttons:

- *Edit* - this button turns on edit mod for this page;
- *Remove* - this button deletes project.

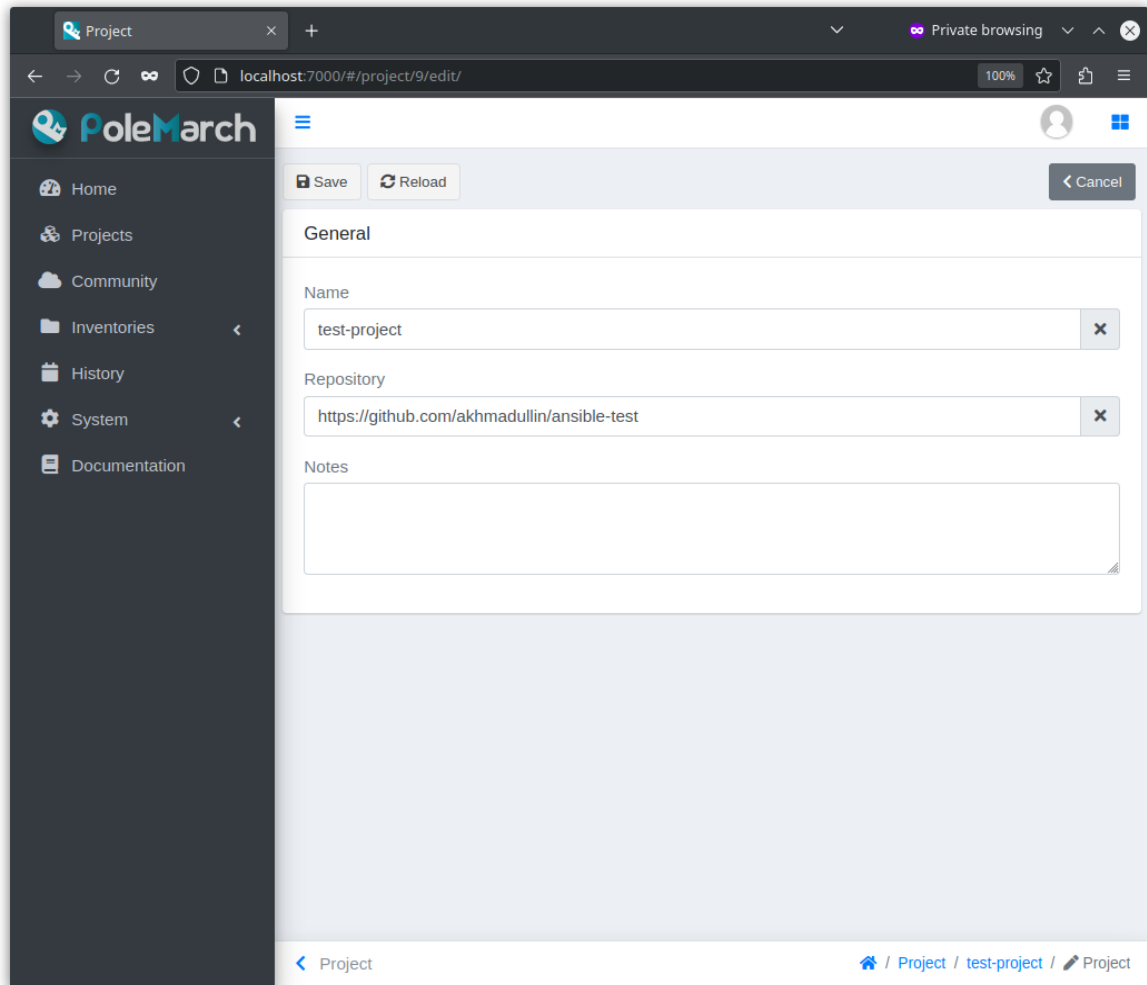
Sublinks buttons:

- *Ansible modules* - this button opens project's module list;
- *Ansible playbooks* - this button opens project's playbook list;
- *Execution templates* - this button opens project's template list;
- *History* - this button opens project executions' history list;
- *Inventory* - this button opens project's inventory list;
- *Variables* - this button opens project's variables list.

Action buttons:

- *Copy* - this button opens "copy" action page;
- *Execute ansible module* - this button opens "execute module" action page;
- *Execute ansible playbook* - this button opens "execute playbook" action page;
- *Set owner* - this button opens "set owner" action page;
- *Sync* - this button syncs your Polemarch project with repository.

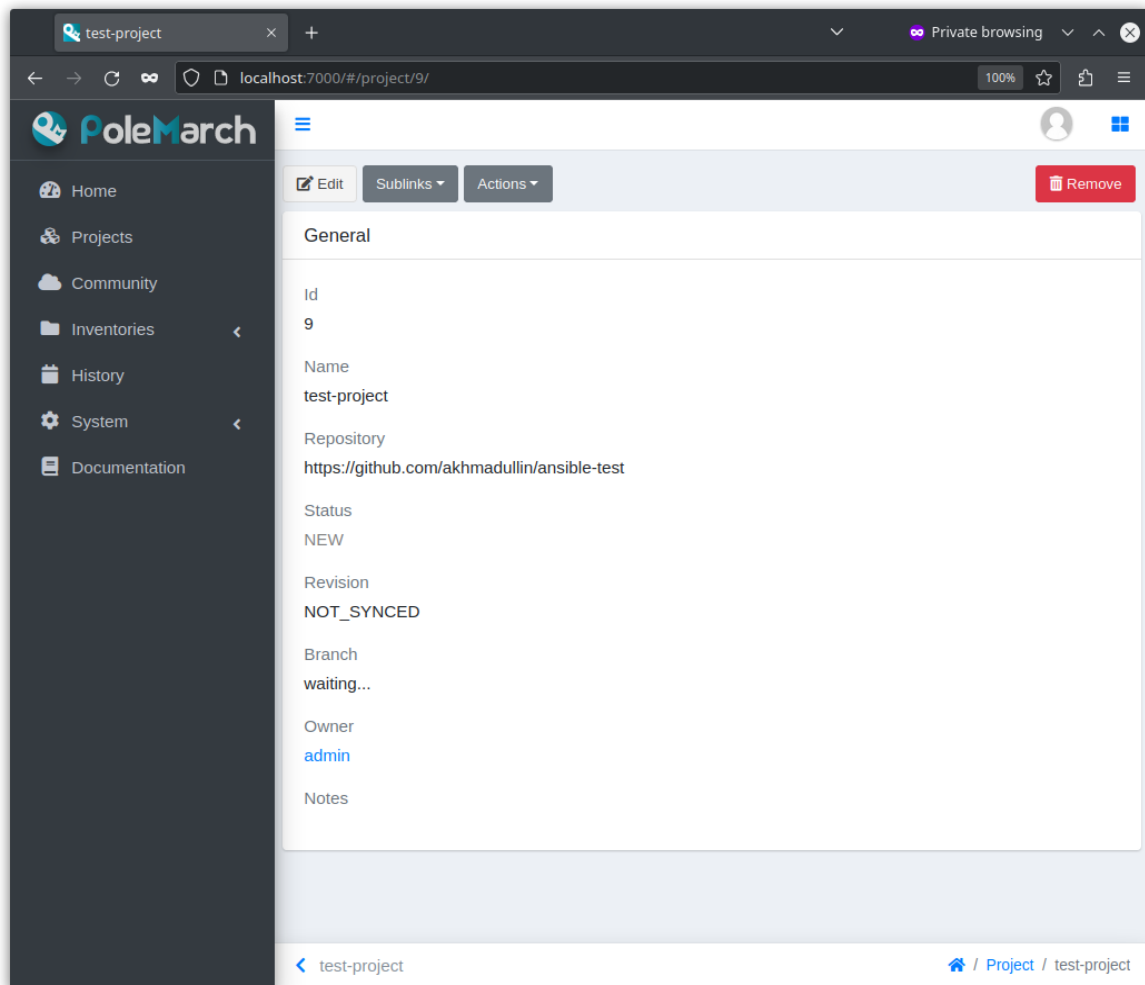
If you want to edit values of some fields on any page in Polemarch, you should click the *Edit* button to activate edit mode.



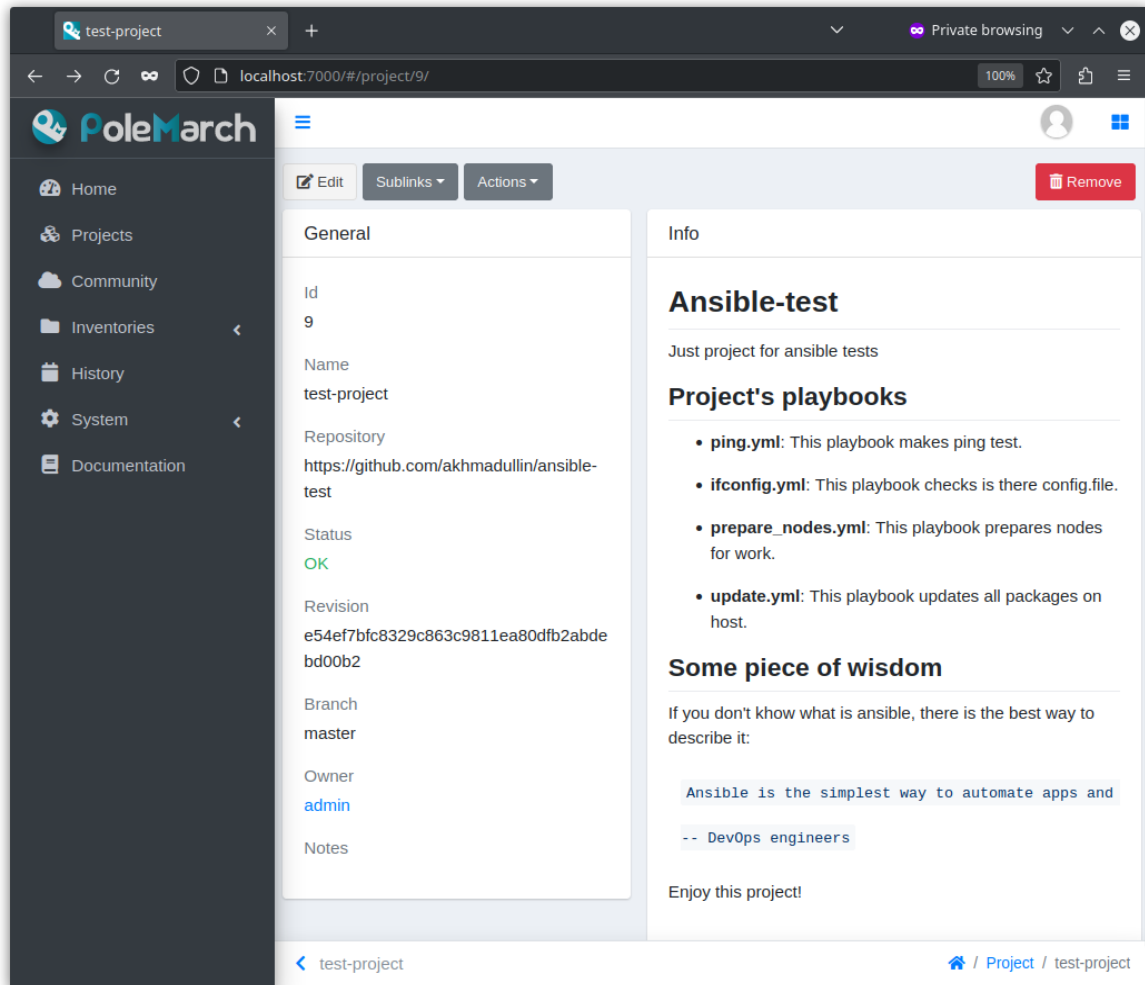
In this mode, you can change the project's name, update the repository URL, and make notes about your project.

- *Save* - this button saves all changes you have made on this page.
- *Reload* - this button resets all changes that you have done on this page.

Now, let's back to read only mode of our project page.



As you can see, now project's status is "NEW", so we need to click the *Sync* button to get all needed data from your GIT project.



Once your project status changes to “OK”, you can start working with Polemarch.

Warning: If you update something in your GIT repository, don’t forget to run sync in Polemarch for pulling your changes.

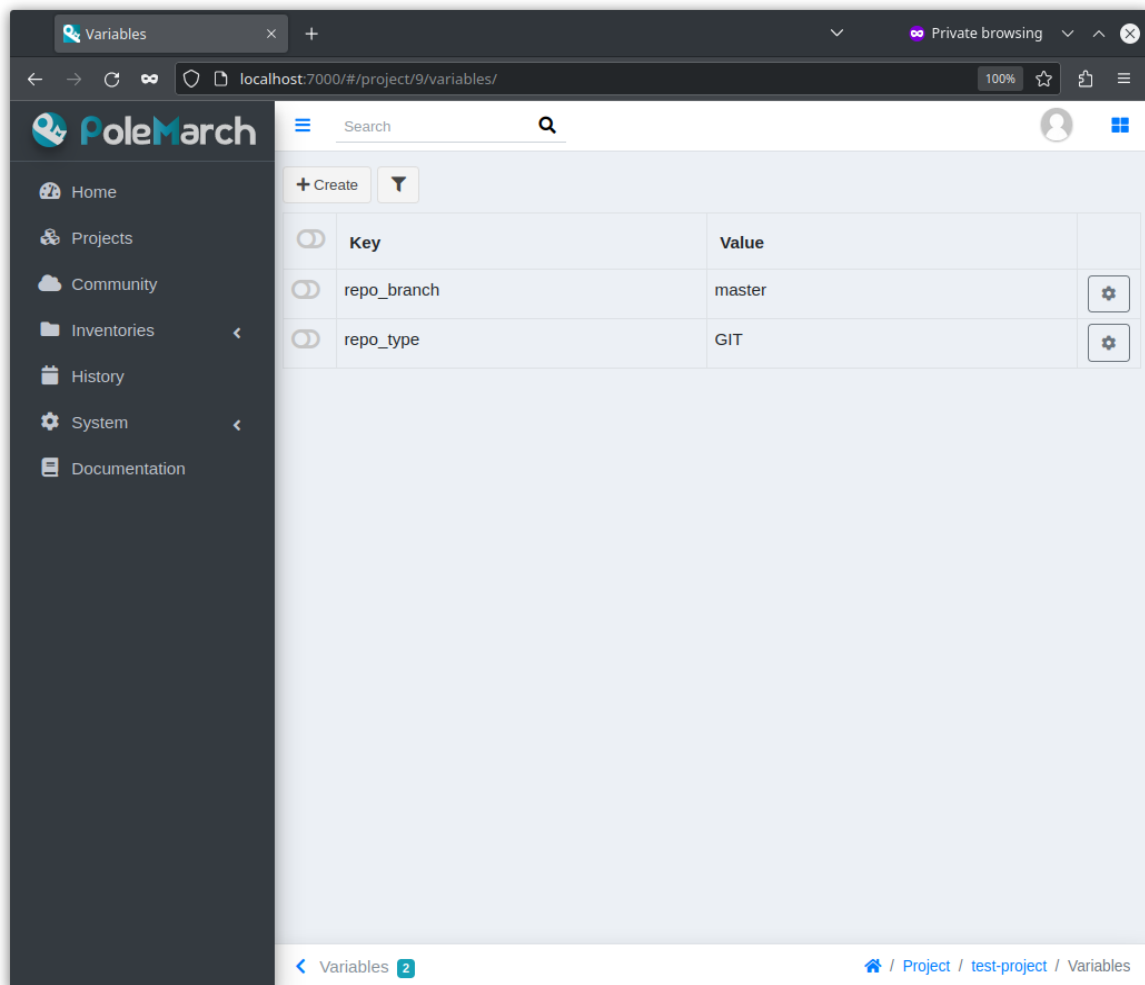
Warning: If you have trouble when cloning from *local* git repository, this may be caused by `protocol.file.allow` setting that is set to `user` in your git config. For more information please see [this topic](#). One of the solutions might be:

```
git config --global protocol.file.allow always
```

Keep in mind that this command allows file clone **globally at git level for all your projects**.

3.4 Project variables

If you need to change some properties of your project (for example, branch, key or password of project), you can do it on the “variables” page of your project:



As you can see, now current project has only 2 variables:

- **repo_branch** - branch of GIT repository;
- **repo_type** - type of Polemarch project.

Other available project variables, that you can add with clicking the *Create* button:

- **repo_sync_on_run** - boolean, if true, on every execution will be used project files obtained directly from the remote repository with revision specified in **repo_branch**, instead of current project stored in Polemarch.
- **repo_sync_on_run_timeout** - time value after which attempt to clone project on execution will be stopped. Works only in conjunction with **repo_sync_on_run**.

Note: In case of GIT projects **repo_sync_on_run_timeout** limits time to whole cloning process. For example, if it set to 3 seconds and after 3 seconds project was cloned only on 70%, execution will be aborted.

This behavior is somewhat different with TAR projects. Timeout specifies how long Polemarch must wait until successfully establish connection with the server. After that download process will start and continue until it completes, regardless of **repo_sync_on_run_timeout**.

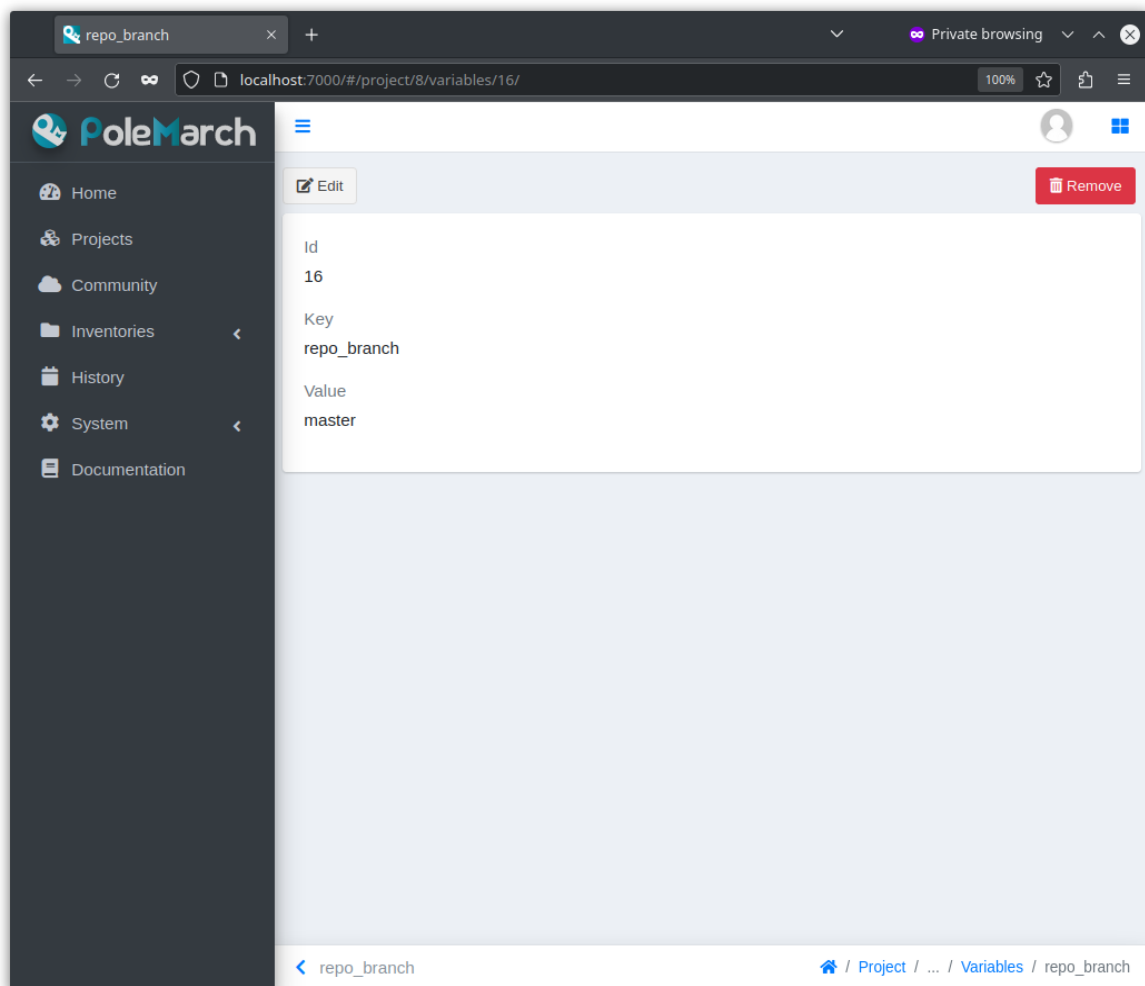
When working with MANUAL projects both **repo_sync_on_run** and **repo_sync_on_run_timeout** options

do not applies.

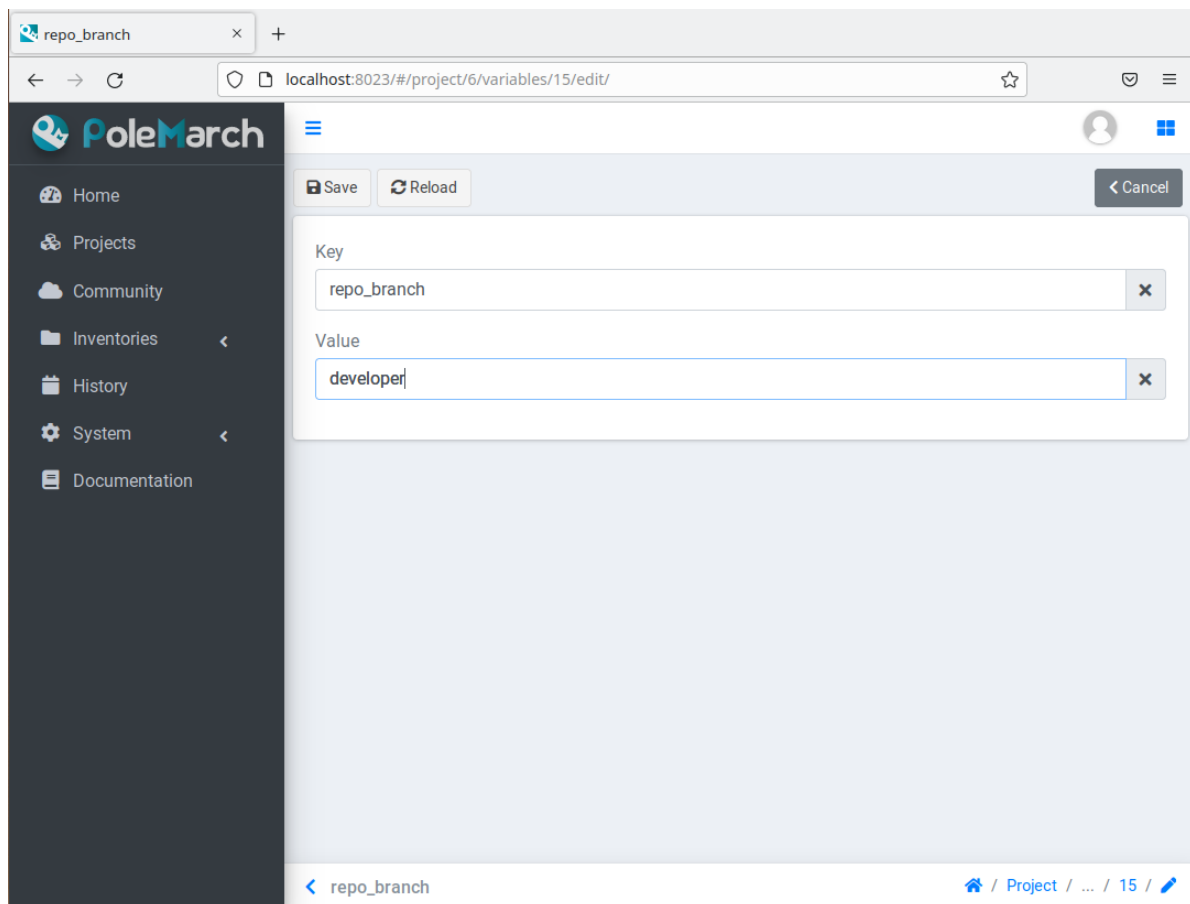
- **repo_password** - GIT repository password;
- **repo_key** - GIT repository key;
- **playbook_path** - path to playbook;
- **ci_template** - runs the selected template during project synchronization;
- Environment variables, with key starting from **env_**. For example **env_test_var** would create environment variable **test_var** on run tasks from this project.

Note: You can add **ANSIBLE_CONFIG** environment variable by creating **env_ANSIBLE_CONFIG** variable in your project. This will make your project to use specified config at every execution. Make sure that path is relative to project directory.

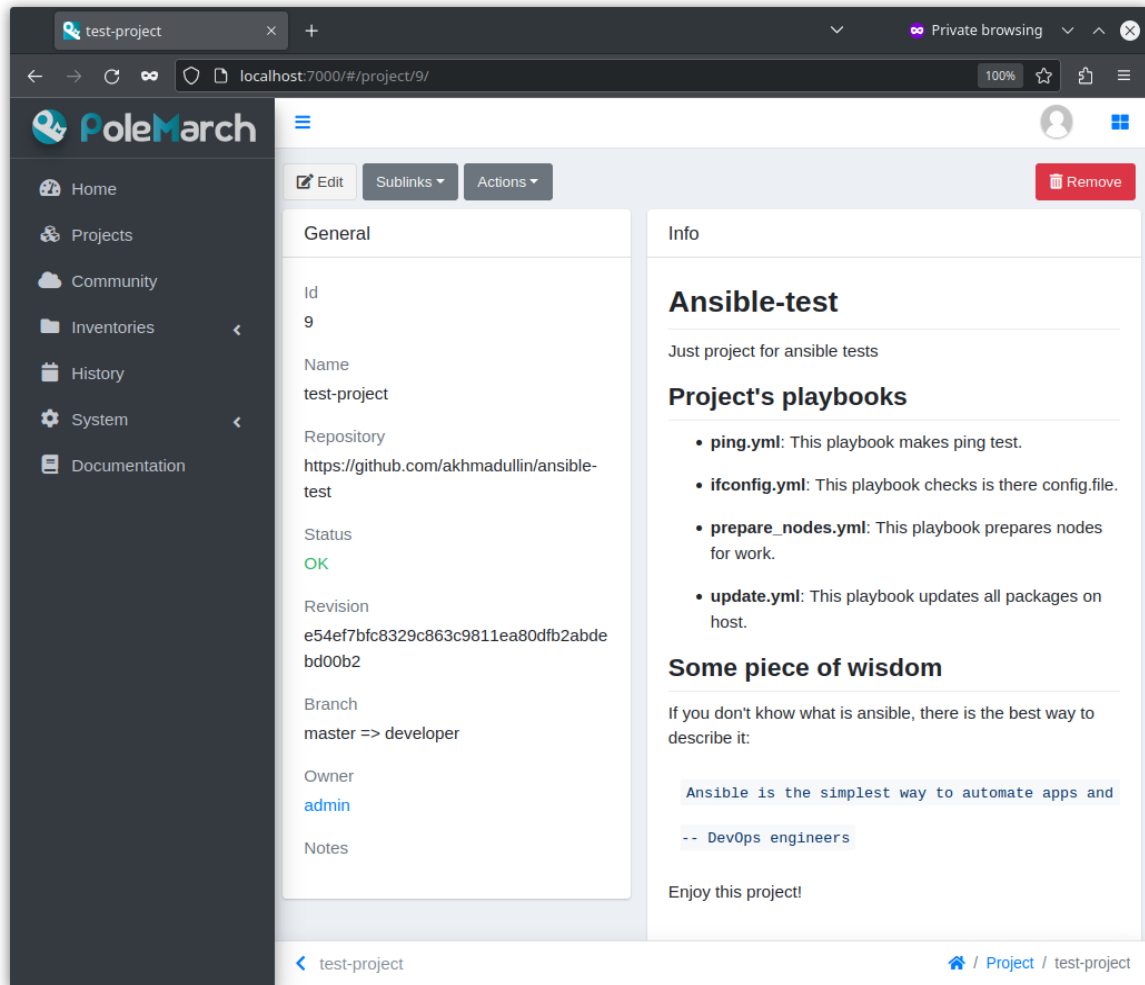
Let's edit **repo_branch** variable. To do it you need click the **repo_branch** item in list. Then you will see the following page:



After clicking the *Edit* button you need to change branch name of 'value' field. Then click the *Save* button.

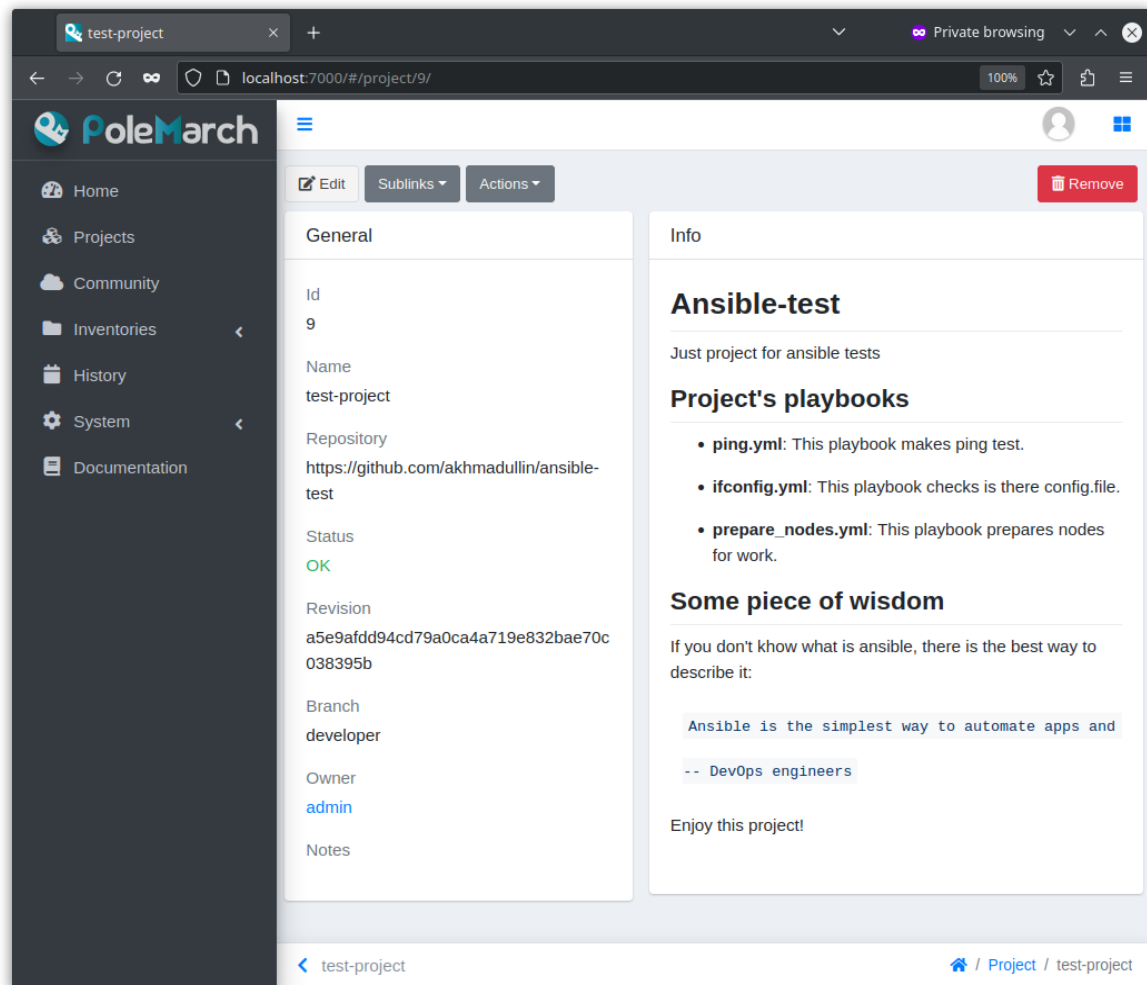


After changing of **repo_branch** variable you need to sync your Polemarch project to get files from selected branch. Until you synchronize the project, you'll see in the “Branch” field your old branch and the new one.



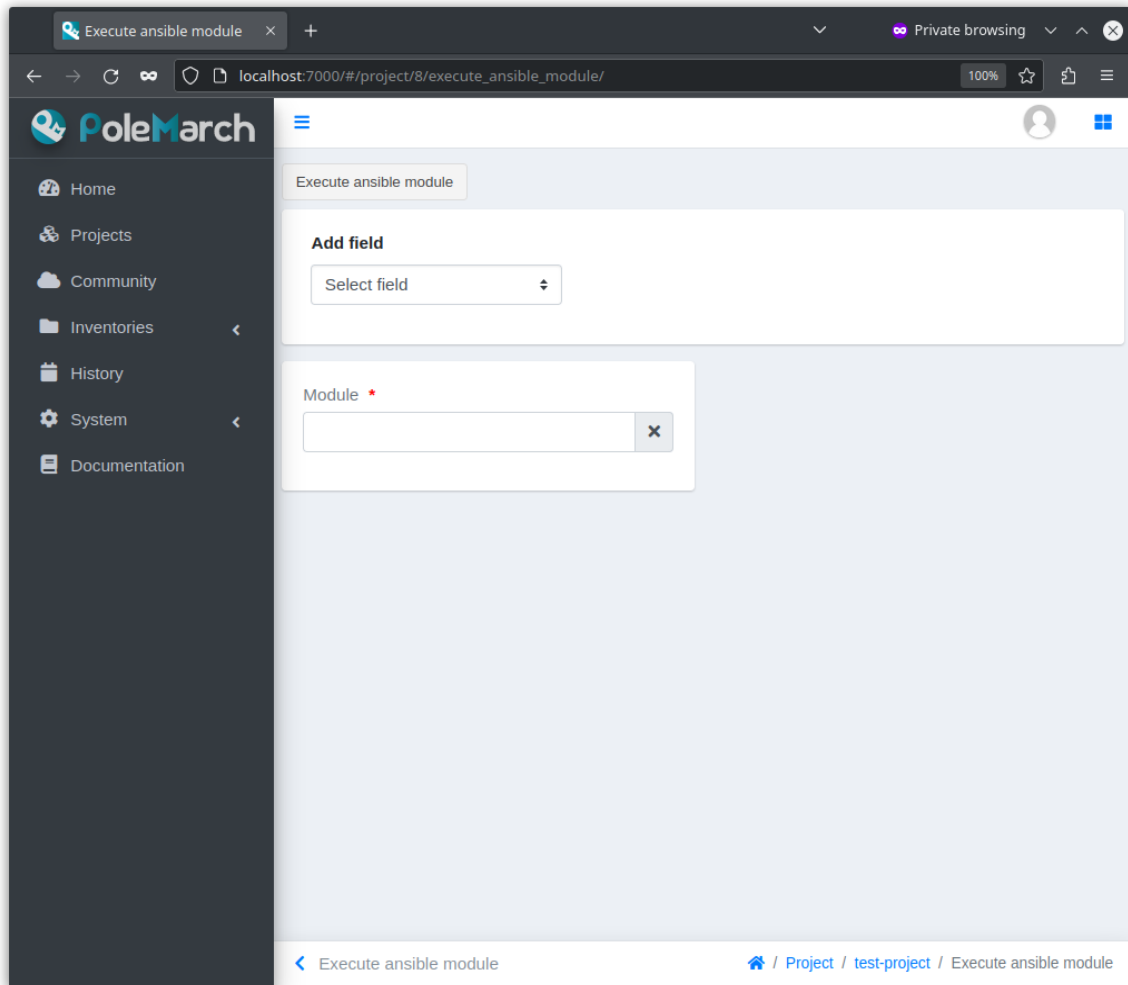
Note: If your need to choose branch with tag name you should use following format of branch name `tags/[tag_name]`.

After successful project synchronization you will see selected branch name in 'branch' field:



3.5 Ansible Module execution

The simplest way to start using Polemarch is to execute ansible module. To make this action click the *Execute ansible module* button on project page.



As you can see, there are two fields on this page:

- **Module** - autocomplete field with the list of project's modules.
- **Add field** - add additional argument.

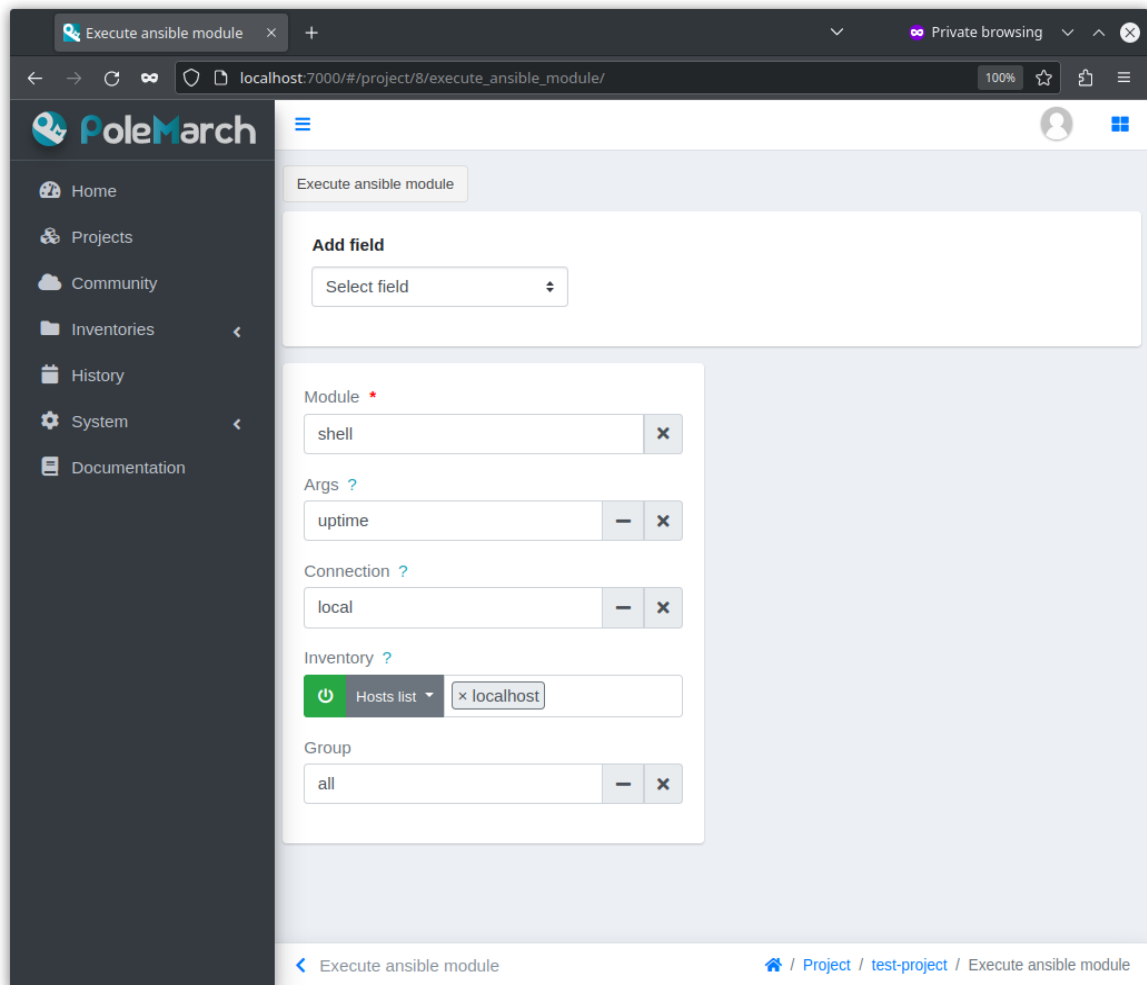
Also there is only one button on this page:

- *Execute ansible module* - this button starts action execution.

For example, let's execute module `shell` on `localhost` with argument `uptime`. To do it we need to add next fields:

- **Inventory** - it can be inventory from Polemarch system, list of hosts, that are separated by `,`, or path to your inventory in project folder.
- **Group** - to which hosts from inventory execute this module.
- **Connection** - type of connection.
- **Args** - list of arguments for current module.

After all fields have been filled, our page started look like:



So, let's execute our first task on Polemarch! To do it click the *Execute module* button.

When status of your module execution changes to "OK" you will see the next page:

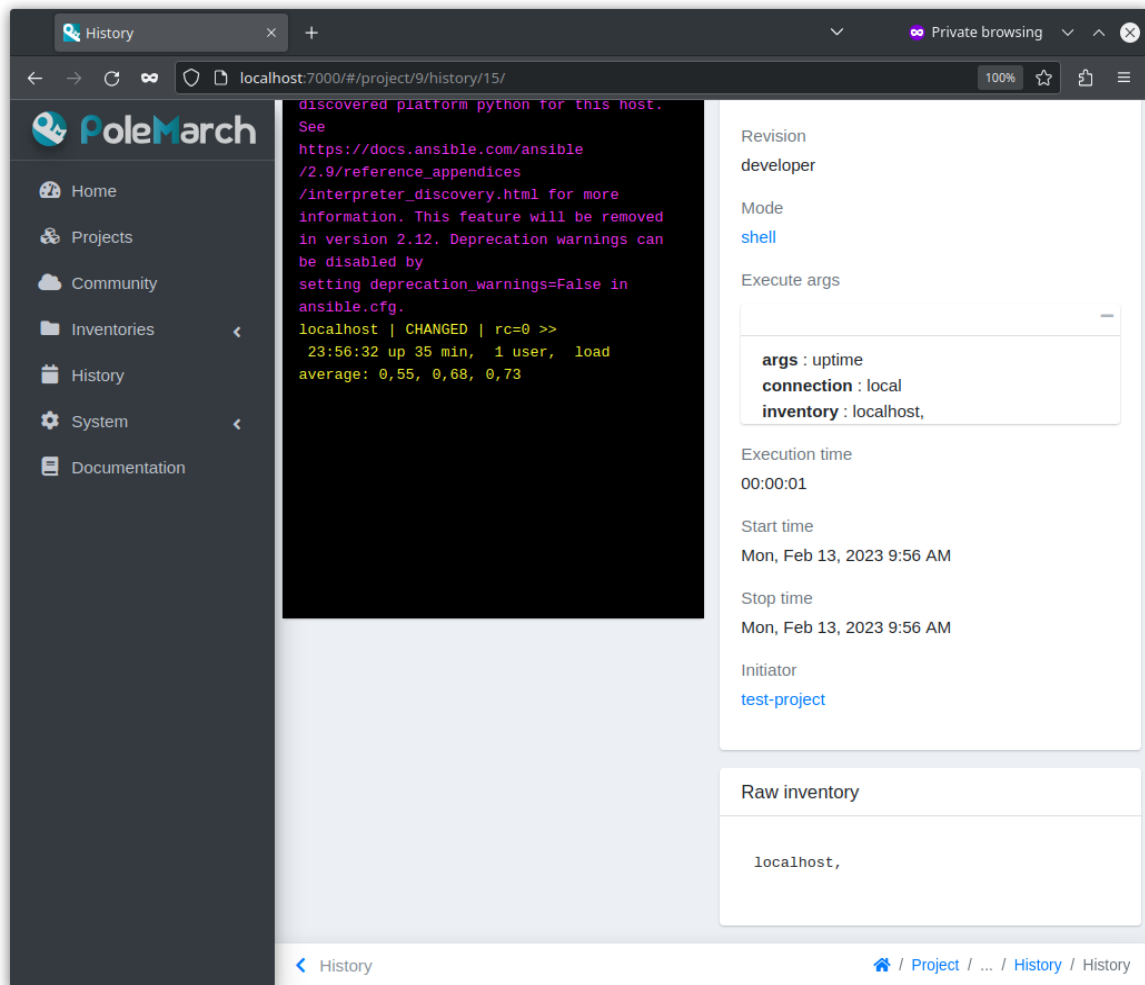
The screenshot shows the Polemarch web interface in a browser window. The address bar indicates the URL is `localhost:7000/#/project/9/history/15/`. The interface has a dark sidebar on the left with navigation links: Home, Projects, Community, Inventories, History, System, and Documentation. The main content area is divided into two panels. The left panel, titled "Execution output (stdout)", displays the output of an Ansible module execution. The right panel shows metadata for the execution, including Id, Status, Executor, Revision, Mode, Execute args, Execution time, Start time, and Stop time.

Execution output (stdout)

```
[DEPRECATION WARNING]: Distribution ubuntu 20.04 on host localhost should use /usr/bin/python3, but is using /usr/bin/python for backward compatibility with prior Ansible releases. A future Ansible release will default to using the discovered platform python for this host. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information. This feature will be removed in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
localhost | CHANGED | rc=0 >>
  23:56:32 up 35 min,  1 user,  load
average: 0,55, 0,68, 0,73
```

Metadata:

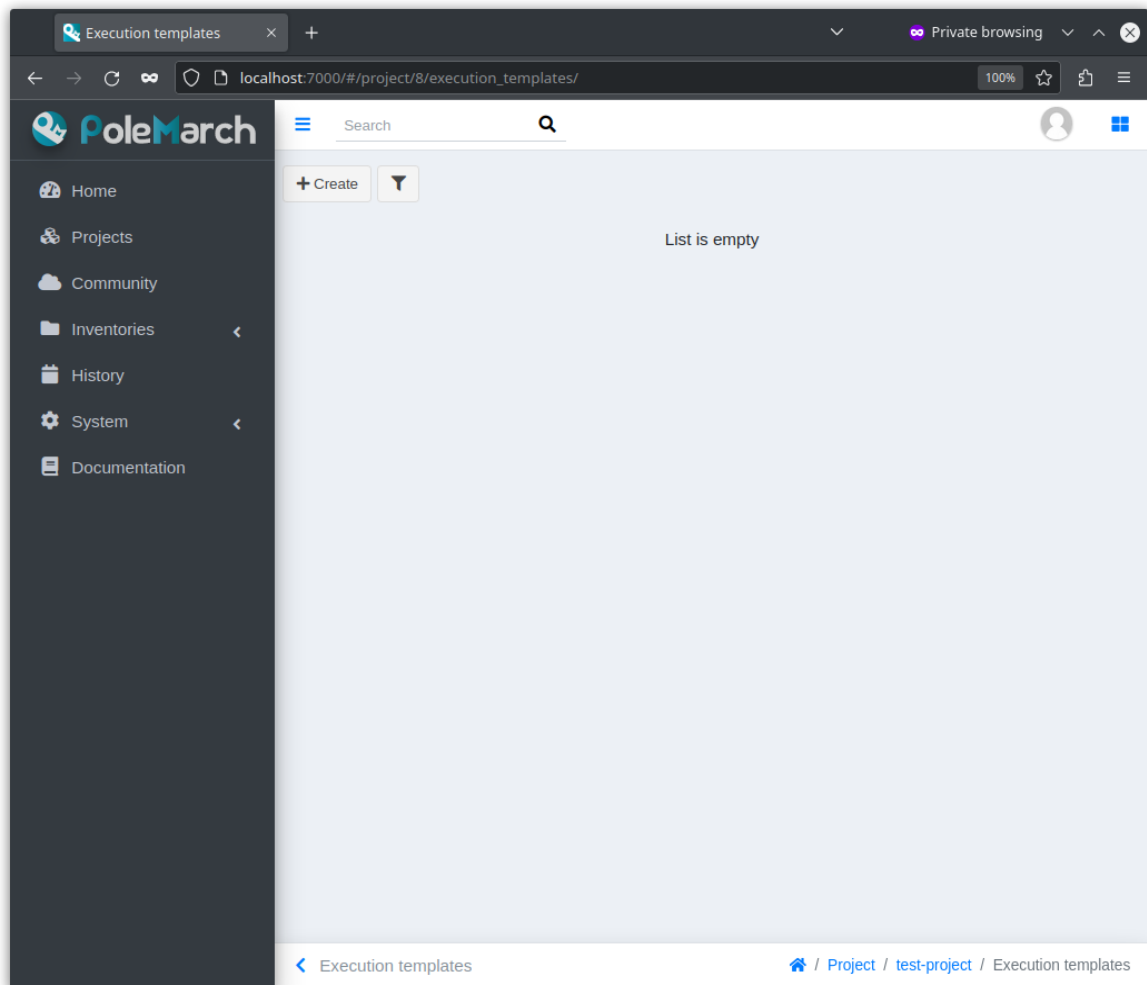
- Id: 15
- Status: OK
- Executor: admin
- Revision: developer
- Mode: shell
- Execute args: args : uptime, connection : local, inventory : localhost,
- Execution time: 00:00:01
- Start time: Mon, Feb 13, 2023 9:56 AM
- Stop time:



3.6 Execution templates

In previous abstract to execute module we needed to fill several fields. To do it before every module/playbook execution is rather inconvenient. In this case Polemarch templates save our time and nerves. Polemarch template is an object, that saves all options that user used during task execution.

For example, let's create template which executes ansible playbook. To do it click the *Execution templates* button on project page. As you can see, there are no templates in the project's template list now.



There is only one button here:

- *Create* - this button opens the page for creation of new template.

To create template click the *Create* button on this page.

As you can see, there are several fields on this page:

- **Name** - name of template at your choice.
- **Plugin** - plugin which executes our task. There are two built in plugins: *ANSIBLE_PLAYBOOK* and *ANSIBLE_MODULE* allows you work with Ansible.

Note: You can create your own execution plugin allows you to run any shell command. If you're interested in, checkout [Execution plugins documentation](#).

- **Playbook** - name of playbook which will be used during execution.
- **Add fields** - here you can add additional arguments for this template. In our case we're going to use inventory and connection arguments.

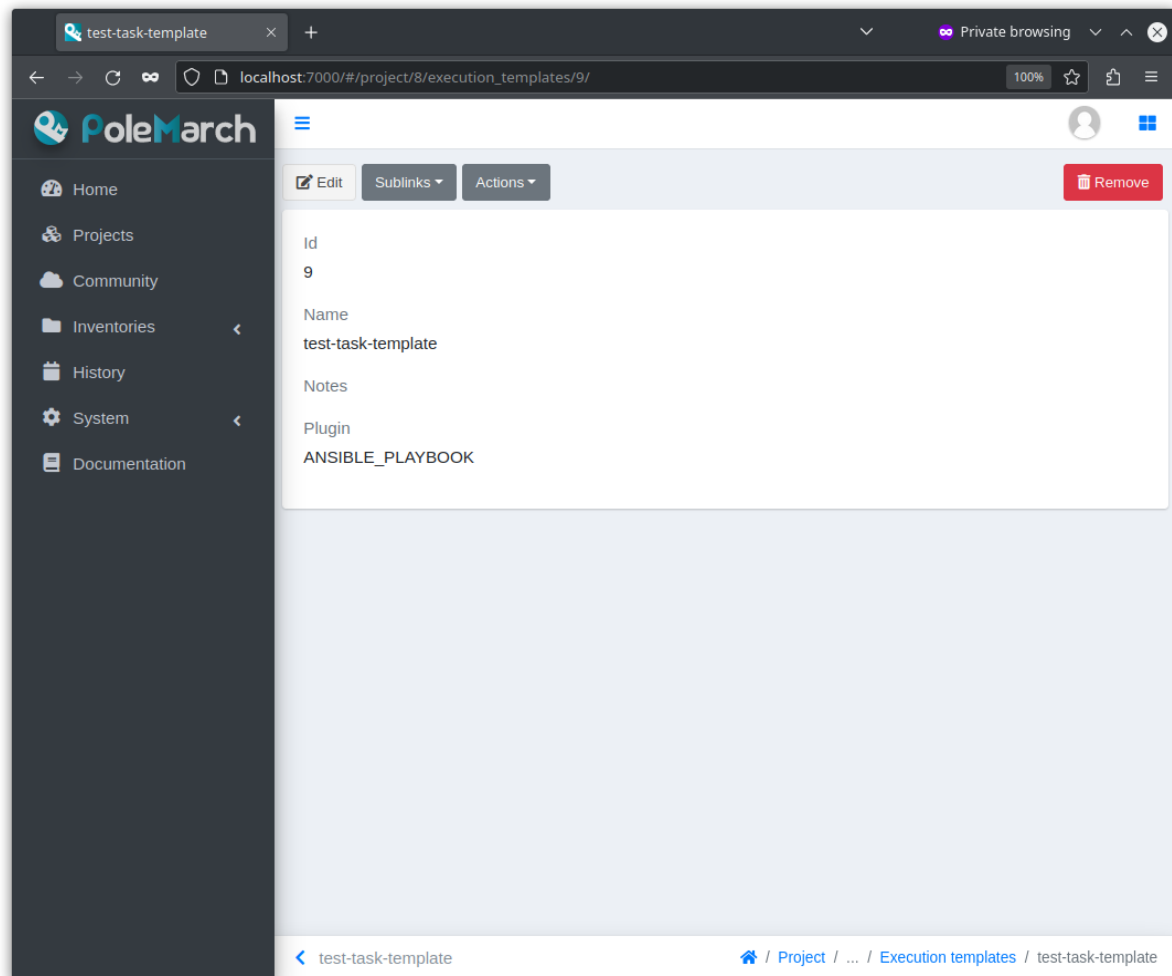
After all fields have been filled, our page started look like:

The screenshot shows a web browser window with the URL `localhost:7000/#/project/8/execution_templates/new/`. The browser is in 'Private browsing' mode. The Polemarch application interface is visible, featuring a dark sidebar with navigation links: Home, Projects, Community, Inventories, History, System, and Documentation. The main content area is titled 'Execution templates' and contains a 'Save' button at the top. Below the 'Save' button, there are several form fields:

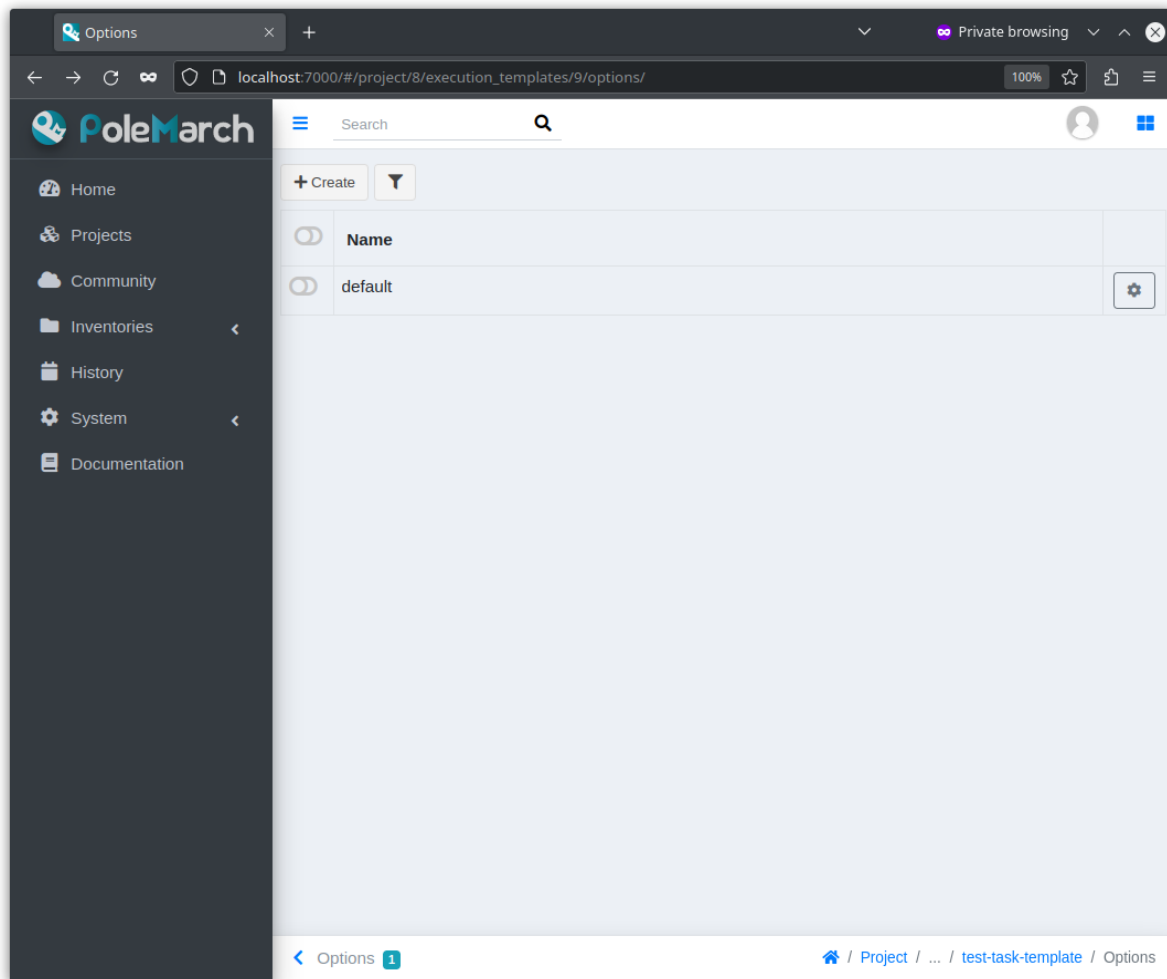
- Name ***: A text input field containing 'test-task-template'.
- Plugin ***: A dropdown menu showing 'ANSIBLE_PLAYBOOK'.
- Arguments ***: A section with an 'Add field' button and a 'Select field' dropdown.
- Playbook ***: A text input field containing 'ping.yml'.
- Connection ?**: A text input field containing 'local'.
- Inventory ?**: A section with a 'Hosts list' button and a text input field containing 'localhost'.

The bottom of the interface shows a breadcrumb trail: `< Execution templates` and a navigation bar with links: `Home / Project / ... / Execution templates / + Execution templates`.

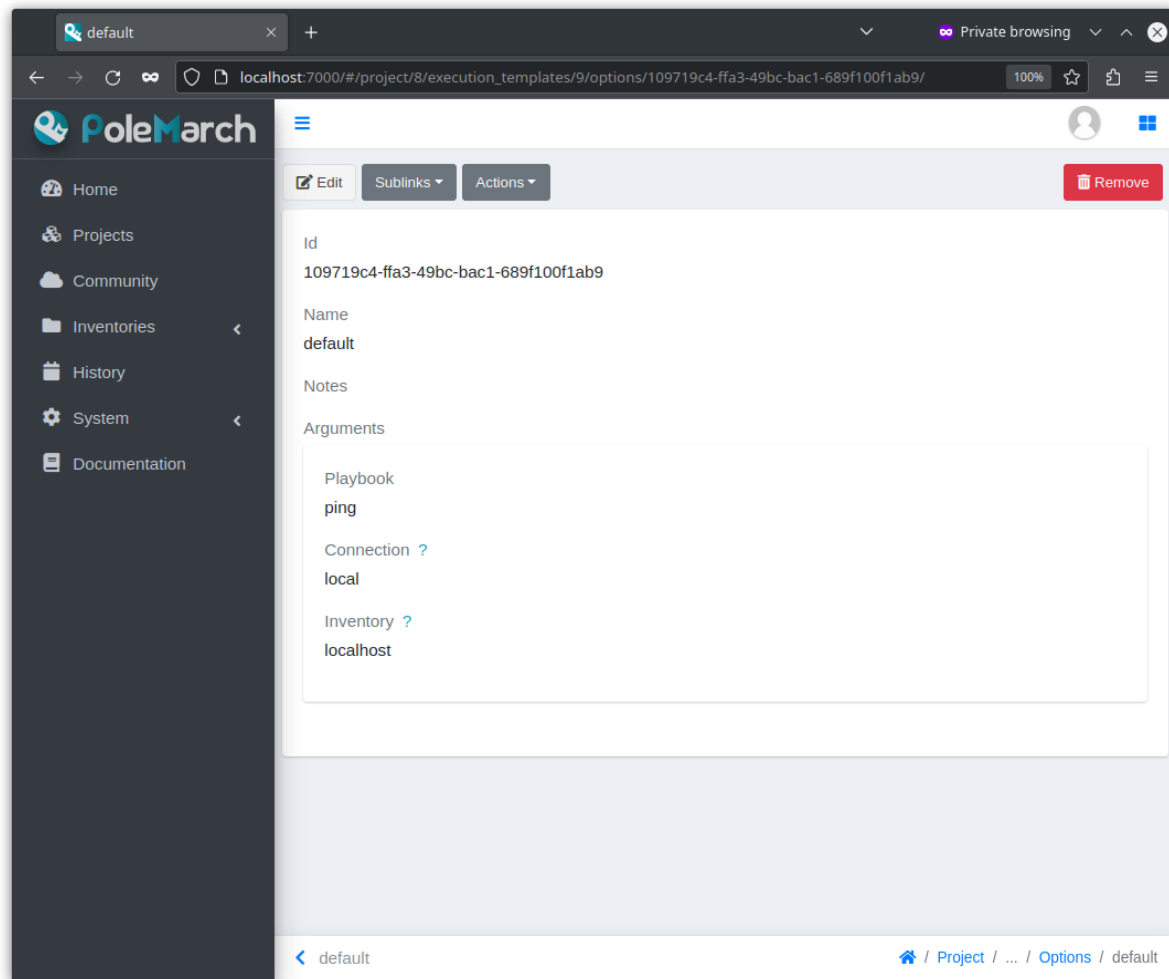
After template creation you will see the next page:



Thought all our arguments are gone? Don't worry. When you create a template, Polemarch also creates a default *option* that holds all arguments. Let's check this out by going to *Sublinks > Options*:



Here is the list of all options which our template can use. You can always create a new option. Let's click on *default*:

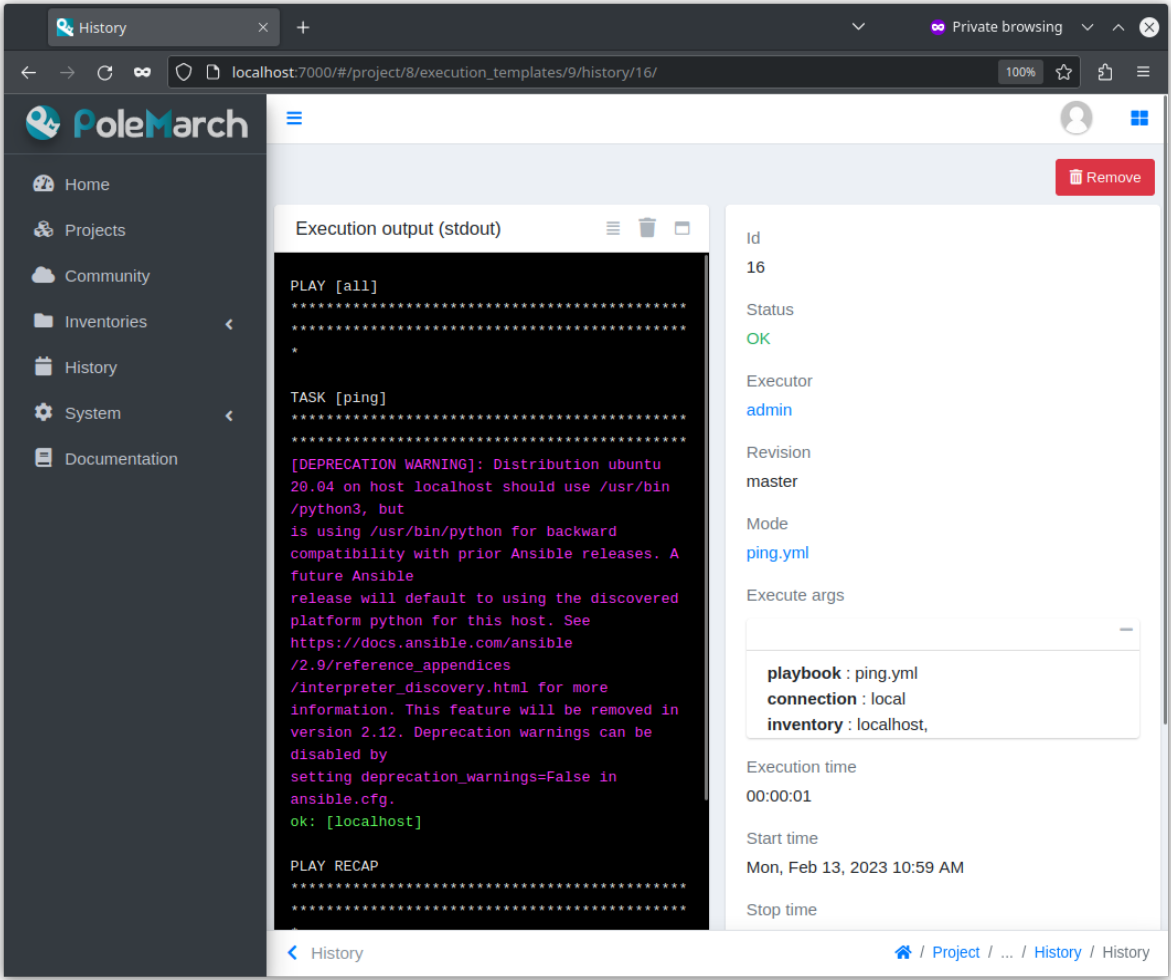


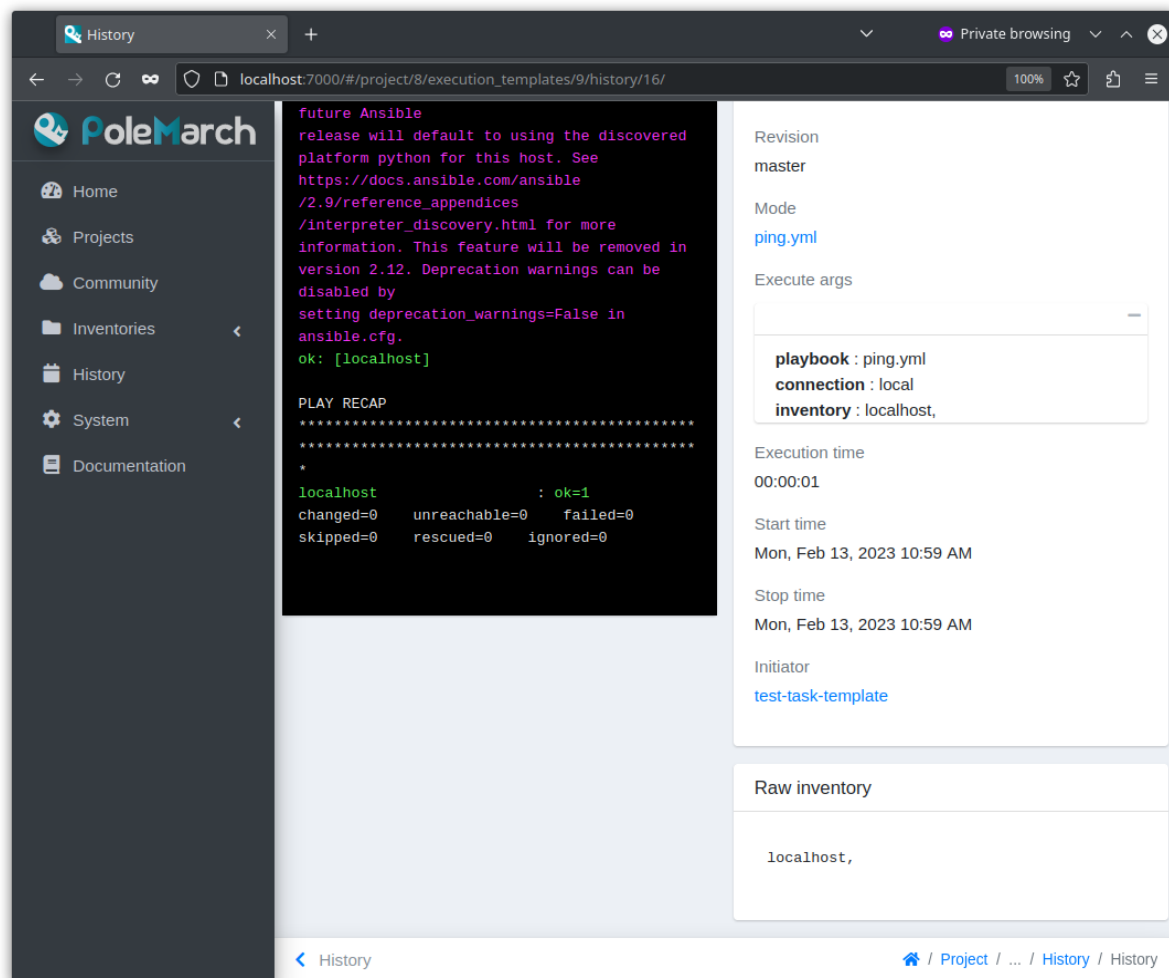
Now we are ready to execute our template. To do it you have two ways:

- Being on the option page, click *Actions* > *Execute*
- Being on the template page, click *Actions* > *Execute*, select an appropriate option and click *Execute* again.

These scenarios are identical. Use the one that suites you best.

As we are already on the option page, let's use the first approach. So, clicking the *Execute* gives us the execution history page:





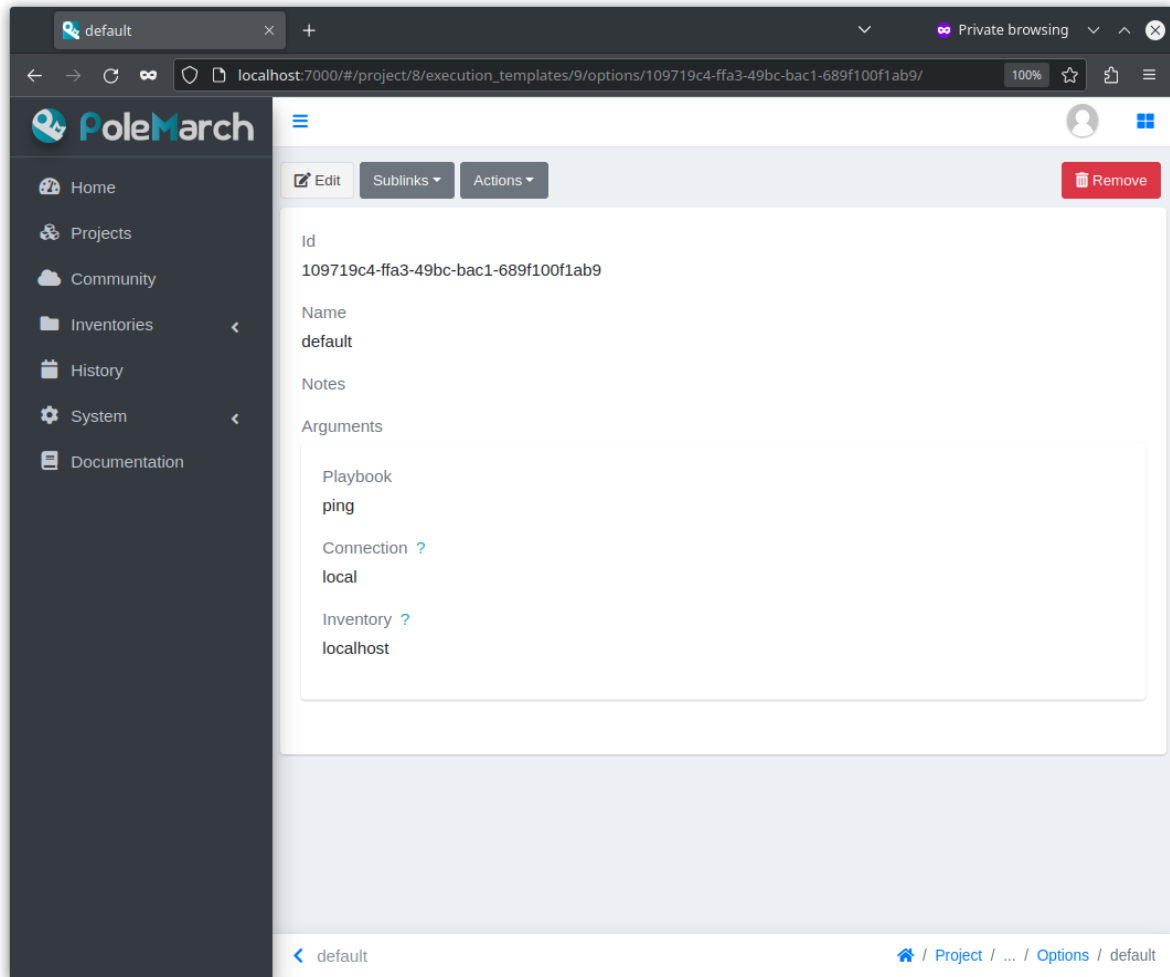
3.7 Periodic tasks

Now let's imagine, that you need to execute task with some interval or on the first day of month, for example, and you do not want to execute it every time by yourself.

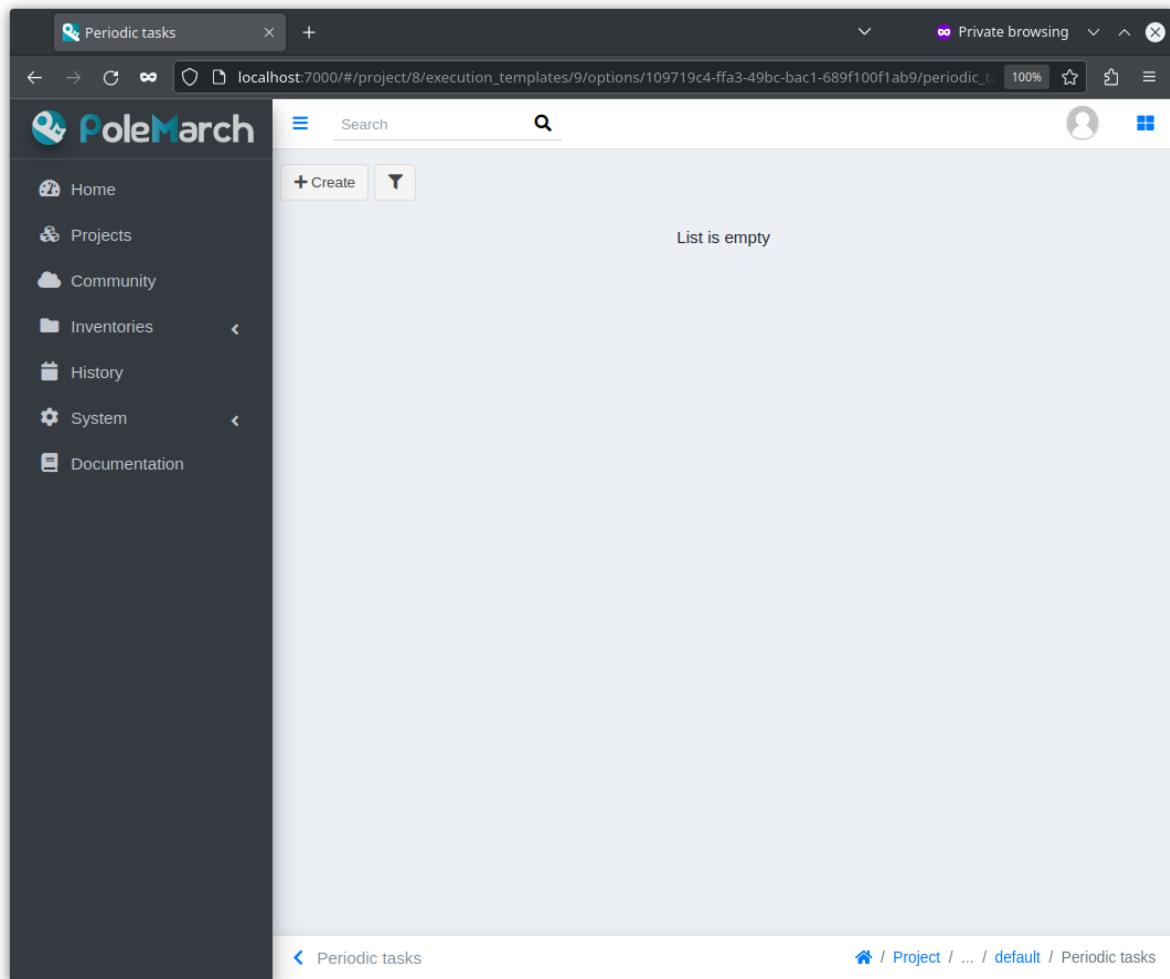
For this case Polemarch has periodic tasks.

Polemarch executes periodic tasks regularly after specified period of time or by crontab schedule.

Remember our *test-task-template* from previous example? Let's create a periodic task for it's *default* option.



Click on the *Sublinks > Periodic tasks*.



As you can see, there are no periodic tasks for now.

There is only one button here:

- *Create* - this button opens the page for creation of new periodic task.

To create a periodic task click the *Create* button on this page.

The screenshot shows a web browser window with the address bar displaying a localhost URL. The application interface has a dark sidebar on the left with navigation links: Home, Projects, Community, Inventories, History, System, and Documentation. The main content area is titled 'Periodic tasks' and contains a form with the following fields:

- Name**: A text input field with a red asterisk indicating it is required.
- Type**: A dropdown menu currently set to 'INTERVAL'.
- Schedule**: A time input field showing '00:00:00' with left and right navigation arrows.
- Enabled**: A toggle switch currently turned on.
- Save result**: A toggle switch currently turned on.
- Notes**: A large text area for additional information.

A 'Save' button is located at the top left of the form area. The bottom of the page shows a breadcrumb trail: [Home](#) / [Project](#) / ... / [Periodic tasks](#) / + Periodic tasks.

As you can see, the form of new periodic task creation consists of following fields:

- **Name** - name of periodic task at your choice.
- **Type** - type of schedule (INTERVAL or CRONTAB).
- **Schedule** - value of execution interval.
 - If “type” is INTERVAL, task runs every time after specified interval has elapsed.
 - If “type” is CRONTAB, task runs according specified schedule.
- **Enabled** - if active, the task will run, otherwise it won't.
- **Save result** - if active than each execution will generate a history page, otherwise it won't.
- **Notes** - not required field for some user's notes, for example, for what purpose this periodic task was created or something like this.

After all fields have been filled, our page started look like:

Periodic tasks

Save

Name *

test-periodic-task

Type *

INTERVAL

Schedule *

00:00:10

Enabled

Save result

Notes

Figuring out how periodic tasks work!

Periodic tasks

Project / ... / Periodic tasks / + Periodic tasks

Let's save the task.

Now you can check history list page. Every 10 seconds an execution history will be generated here. Periodic tasks differ from other tasks by “system” executor:

Start time	Executor	Inventory	Mode	Status
Mon, Feb 13, 2...	system		ping.yml	OK
Mon, Feb 13, 2...	system		ping.yml	OK
Mon, Feb 13, 2...	admin		ping.yml	OK
Mon, Feb 13, 2...	admin		shell	OK
Fri, Feb 10, 202...	admin		aws_codecommit	OK
Fri, Feb 10, 202...	admin		aws_acm_info	OK
Fri, Feb 10, 202...	admin		aws_application_s	OK
Fri, Feb 10, 202...	admin		deploy_gitlab.yml	OK
Fri, Feb 10, 202...	admin		a10_server	OK
Fri, Feb 10, 202...	admin		make_giltab_backi	OFFLINE
Fri, Feb 10, 202...	admin		make_giltab_backi	OK
Fri, Feb 10, 202...	admin		deploy_gitlab.yml	OK
Fri, Feb 10, 202...	admin		deploy.yml	OK
Fri, Feb 10, 202...	admin		deploy.yml	OK

3.8 Inventory

If you don't want to use 'inventory' as just list of hosts separated by , or do not have inventory file in you GIT project ("./inventory", for example), you need to create it in Polemarch.

Warning: Do not forget to add you inventory to project after it's creation. To do it click the *Inventory* button on project page.

Since v3.0.0 inventories can be state managed or not depending on plugin which is manages inventory. We will check them both with an example of built-in *ANSIBLE_STRING* and *ANSIBLE_FILE* inventories which are uses state, and *POLEMARCH_DB* inventory which is not.

Note: You can create your own inventory plugin which will manage its state as you define. If you're interested in, checkout [Inventory plugins documentation](#).

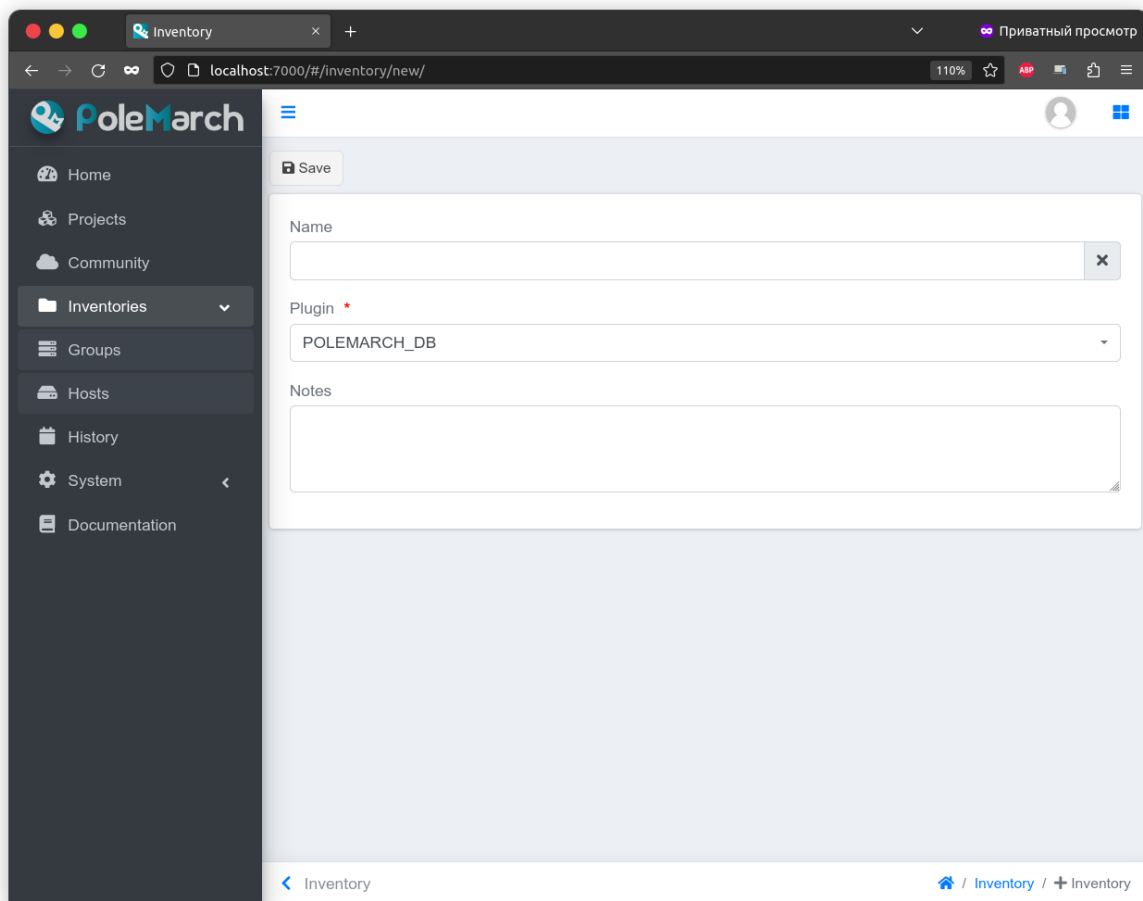
3.8.1 POLEMARCH_DB inventory

POLEMARCH_DB inventory stores all data in separate entities - groups, hosts and variables and not uses state.

By inventory's creation in this case, we mean creation of inventory that includes at least one group, which, in turn, includes at least one host. In other words, in addition to the inventory, the user must create a host and a group.

To better understand this, let's look at next images, which will explain you how to create inventory.

To create inventory you should choose 'Inventories' in left side menu and click the *Create* button. Here you can see the inventory creation form.

The image is a screenshot of a web browser displaying the Polemarch application's 'Inventory' creation page. The browser's address bar shows 'localhost:7000/#/inventory/new/'. The application has a dark sidebar on the left with a menu containing 'Home', 'Projects', 'Community', 'Inventories' (which is expanded), 'Groups', 'Hosts', 'History', 'System', and 'Documentation'. The main content area is light blue and contains a form with a 'Save' button at the top left. The form has three fields: 'Name' (a text input), 'Plugin' (a dropdown menu with 'POLEMARCH_DB' selected), and 'Notes' (a large text area). The bottom of the page shows a breadcrumb trail: '< Inventory' and a footer with a home icon, '/ Inventory', and '+ Inventory'.

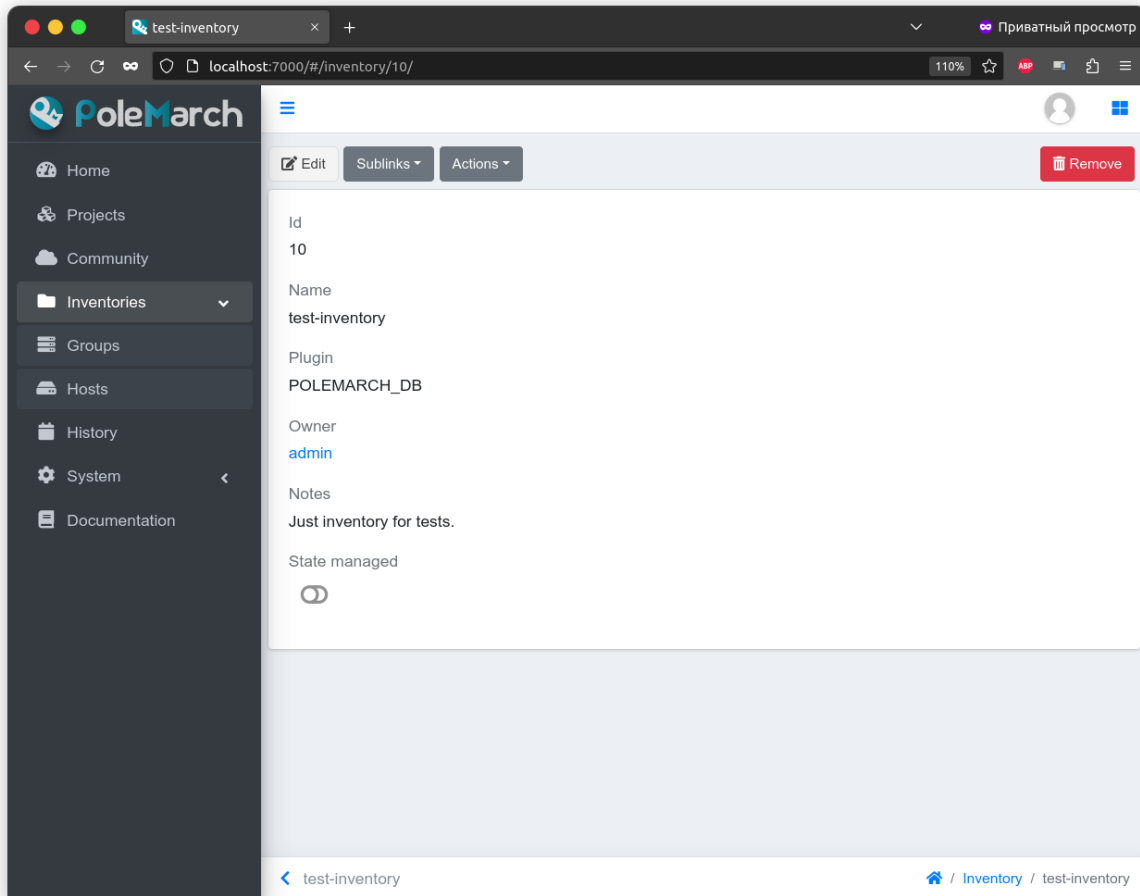
As you can see, there are only 3 fields on this page:

- **Name** - name of inventory.
- **Plugin** - plugin which will manage our inventory. We will select *POLEMARCH_DB* plugin here.
- **Notes** - not required field for some user's notes, for example, for what purpose this inventory was created or something like this.

And there is only one button here:

- *Save* - this button saves new inventory.

After inventory creation you will see the next page:



As you can see there are two new fields on this page:

- **Id** - the unique key of object in database.
- **Owner** - user, who owns this inventory.

Also there are some new buttons here:

- *Edit* - this button turns on edit mod for this page.
- *Remove* - this button deletes inventory.

Sublinks buttons:

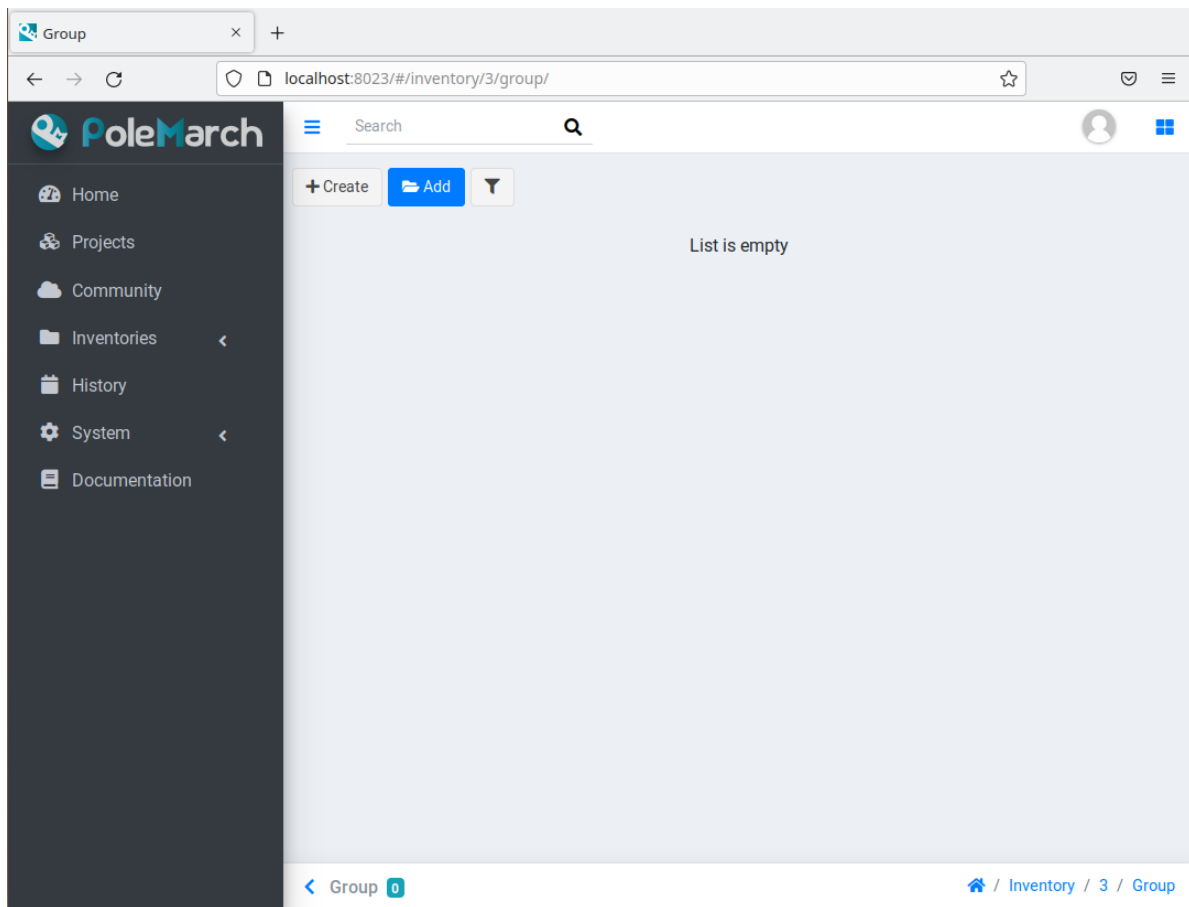
- *All groups* - this button opens inventory's all groups list (list of groups, which includes also groups that are nested into inventory groups).
- *All hosts* - this button opens inventory's all hosts list. (list of hosts, which includes also hosts that are nested into inventory groups).
- *Group* - this button opens inventory's group list.
- *Host* - this button opens inventory's host list.
- *Variables* - this button opens inventory's variables list.

Action buttons:

- *Copy* - this button opens "copy" action page.
- *Set owner* - this button opens "set owner" action page.

Let's look how you can create a group for this inventory. To do it click the *Group* button.

Group

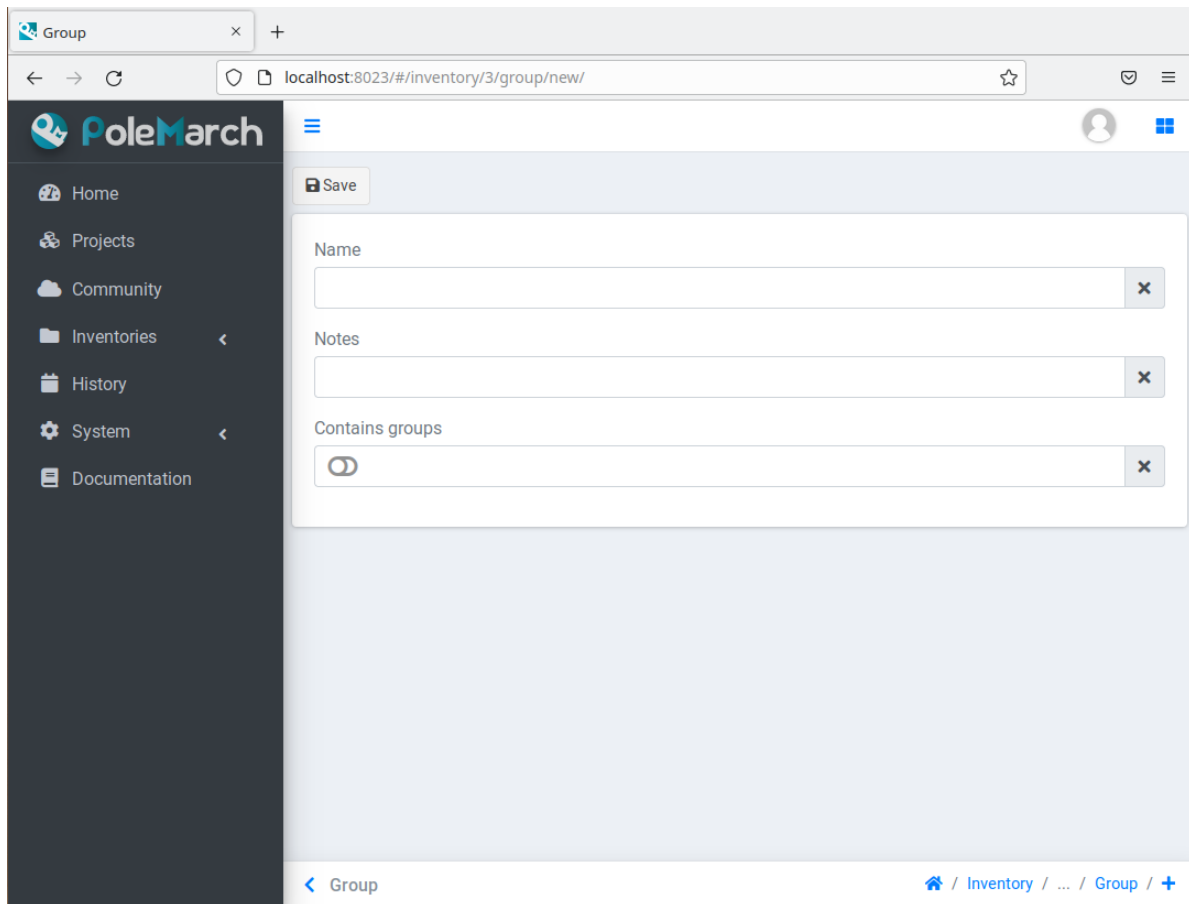


As you can see, there are no groups in the inventory's group list now.

There are two buttons here:

- *Create* - this button opens the page for creation of new group.
- *Add* - this button opens the all group list from database, from which you can choose group for this inventory.

We need to create a group. To do it click the *Create* button.



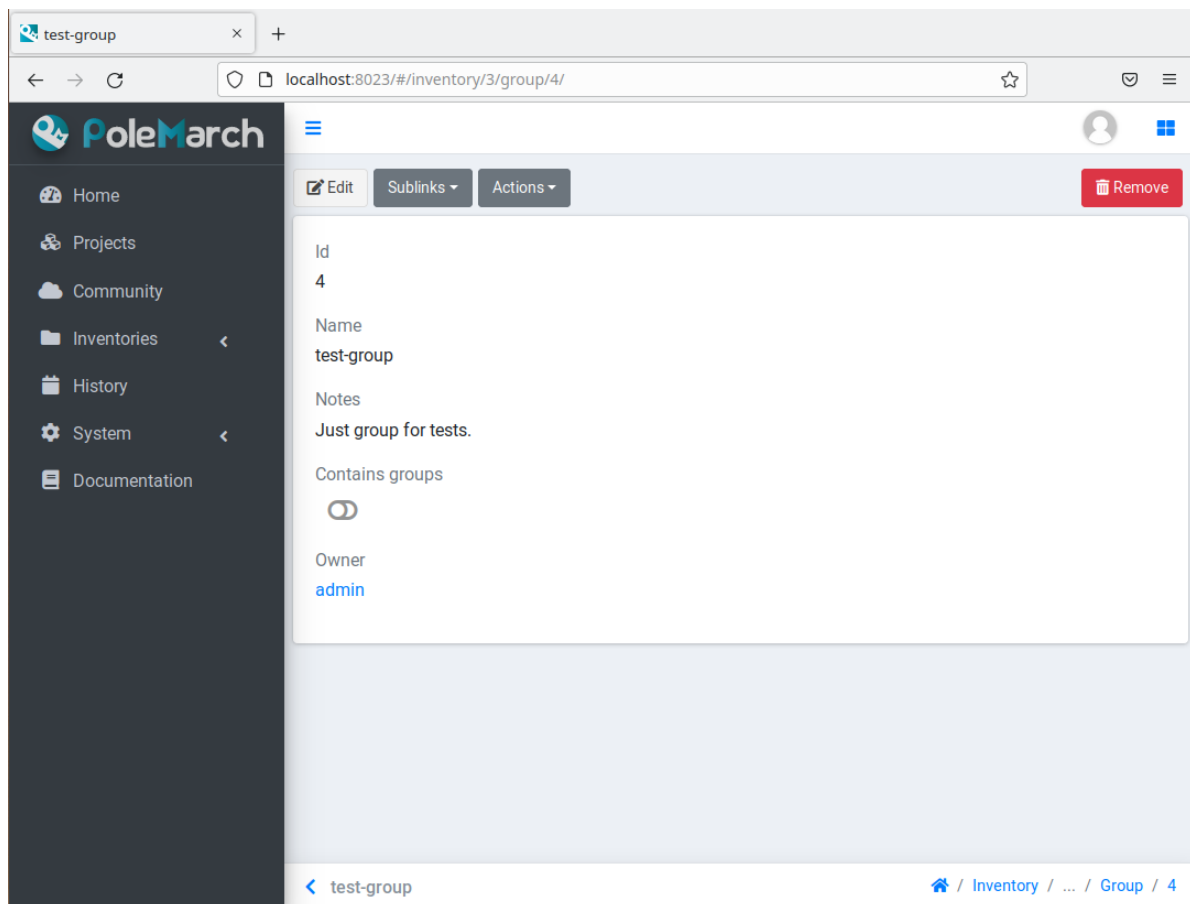
The screenshot shows a web browser window with the URL `localhost:8023/#/inventory/3/group/new/`. The application has a dark sidebar on the left with the 'Polemarch' logo and navigation links: Home, Projects, Community, Inventories, History, System, and Documentation. The main content area is light blue and contains a 'Save' button at the top left. Below it is a form with three fields: 'Name' (a text input), 'Notes' (a text input), and 'Contains groups' (a checkbox). Each field has a small 'x' icon on the right side of its input box. At the bottom of the form, there is a breadcrumb trail: '< Group'. The bottom right of the page shows a navigation bar with links: Home / Inventory / ... / Group / +.

As you can see, the form of new group creation consists of following fields:

- **Name** - name of your group.
- **Contains groups** - boolean field, it means ability of group to contain child groups.
- **Notes** - not required field for some user's notes, for example, for what purpose this group was created or something like this.

Warning: By default SQLite's maximum expression tree depth is 1000. This could create problems with very nested groups. If you encounter so, please refer to [documentation](#).

After group creation you will see the next page:



As you can see there are two new fields on this page:

- **Id** - the unique key of object in database.
- **Owner** - user, who owns this group.

Also there are some buttons here:

- *Edit* - this button turns on edit mod for this page.
- *Remove* - this button deletes group.

Sublinks buttons:

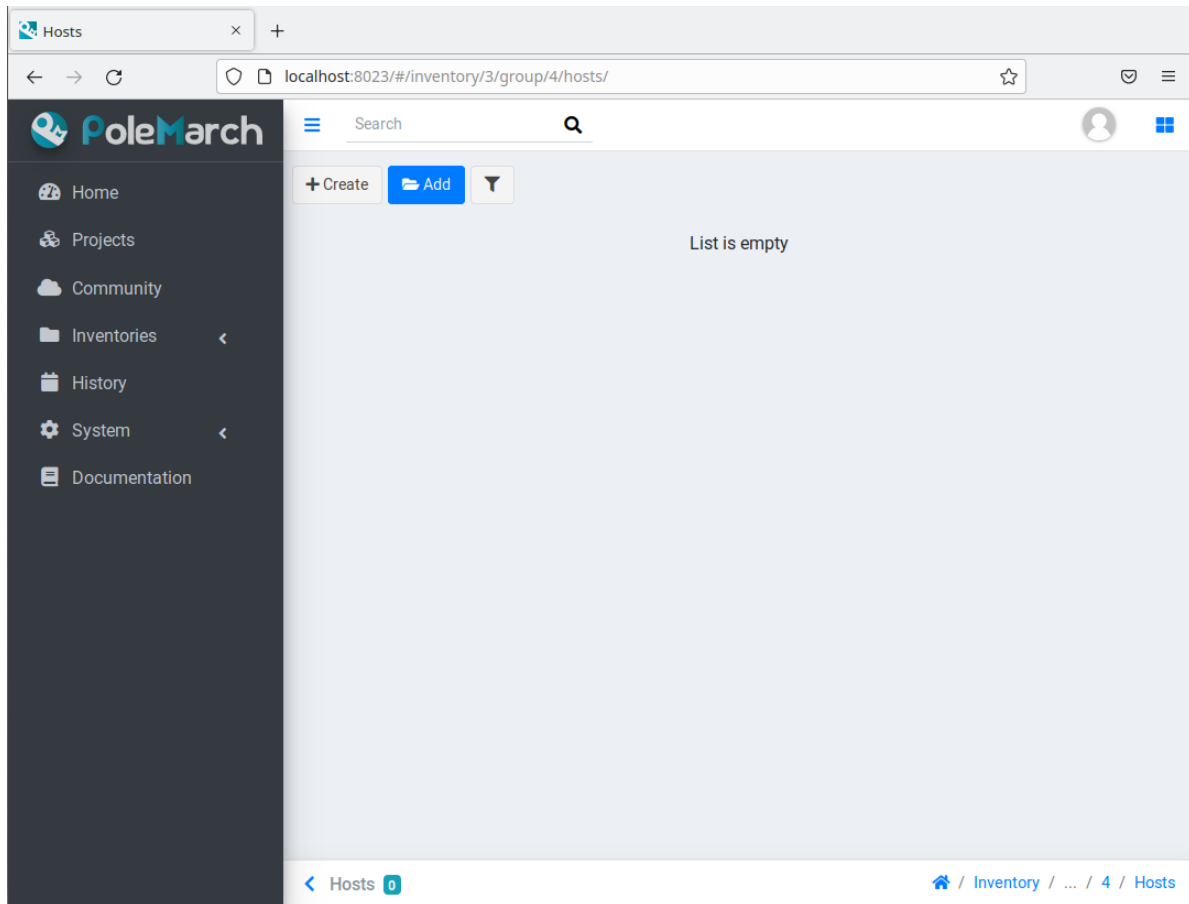
- *Hosts* - this button opens group's host list.
- *Variables* - this button opens group's variables list.

Action buttons:

- *Copy* - this button opens "copy" action page.
- *Set owner* - this button opens "set owner" action page.

Hosts

Let's look how you can create a host for this group. To do it click the *Create* button.



As you can see, there are no hosts in the group's host list now.

There are two buttons here:

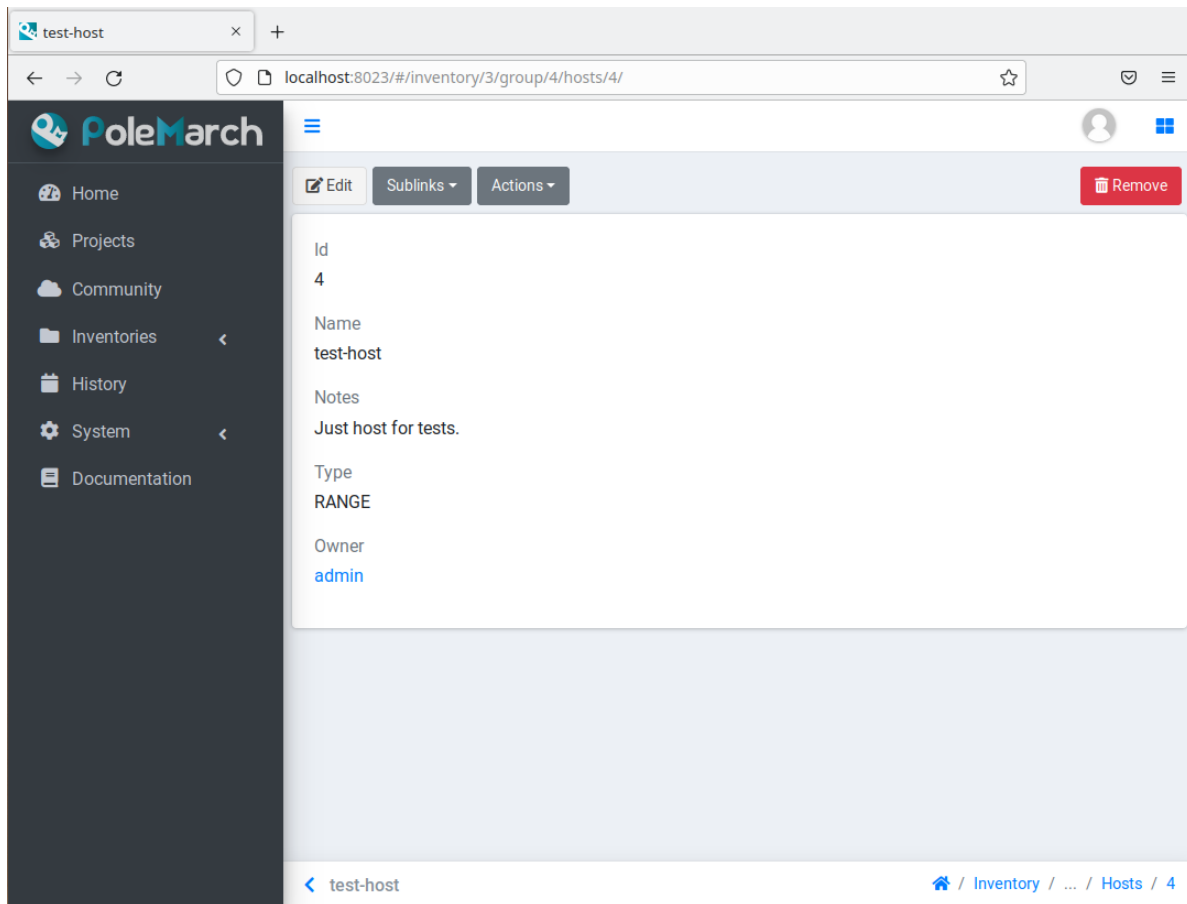
- *Create* - this button opens the page for creation of new host.
- *Add* - this button opens the all host list from database, from which you can choose host for this group.

We need to create host. To do it click the *Create* button.

As you can see, the form of new host creation consists of following fields:

- **Name** - name of your host.
- **Notes** - not required field for some user's notes, for example, for what purpose this host was created or something like this.
- **Type** - type of host (RANGE, HOST).
 - RANGE - range of IPs or hosts.
 - HOST - single host.

After host creation you will see the next page:



As you can see there are two new fields on this page:

- **Id** - the unique key of object in database
- **Owner** - user, who owns this host.

Also there are some buttons here:

- *Edit* - this button turns on edit mod for this page.
- *Remove* - this button deletes host.

Sublinks buttons:

- *Variables* - this button opens host's variables list.

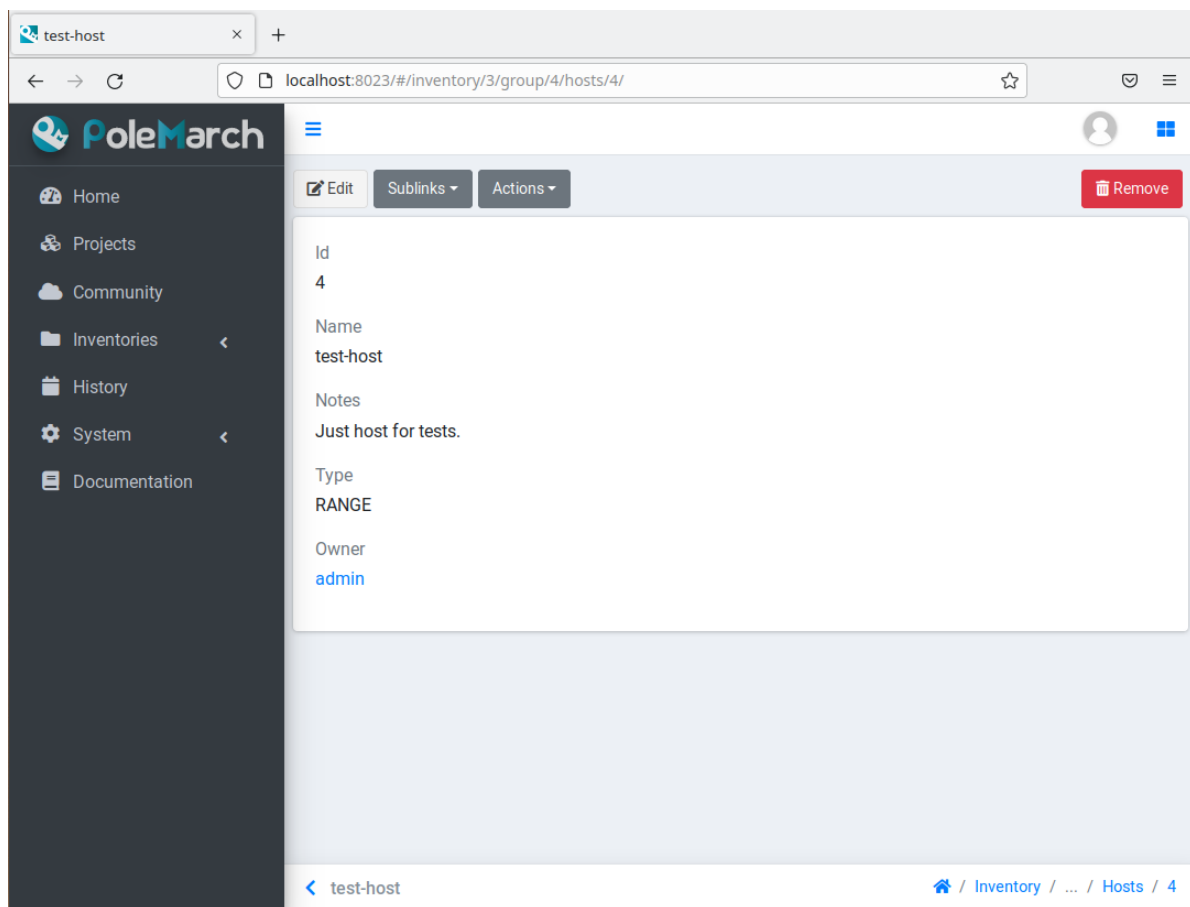
Action buttons:

- *Copy* - this button opens "copy" action page.
- *Set owner* - this button opens "set owner" action page.

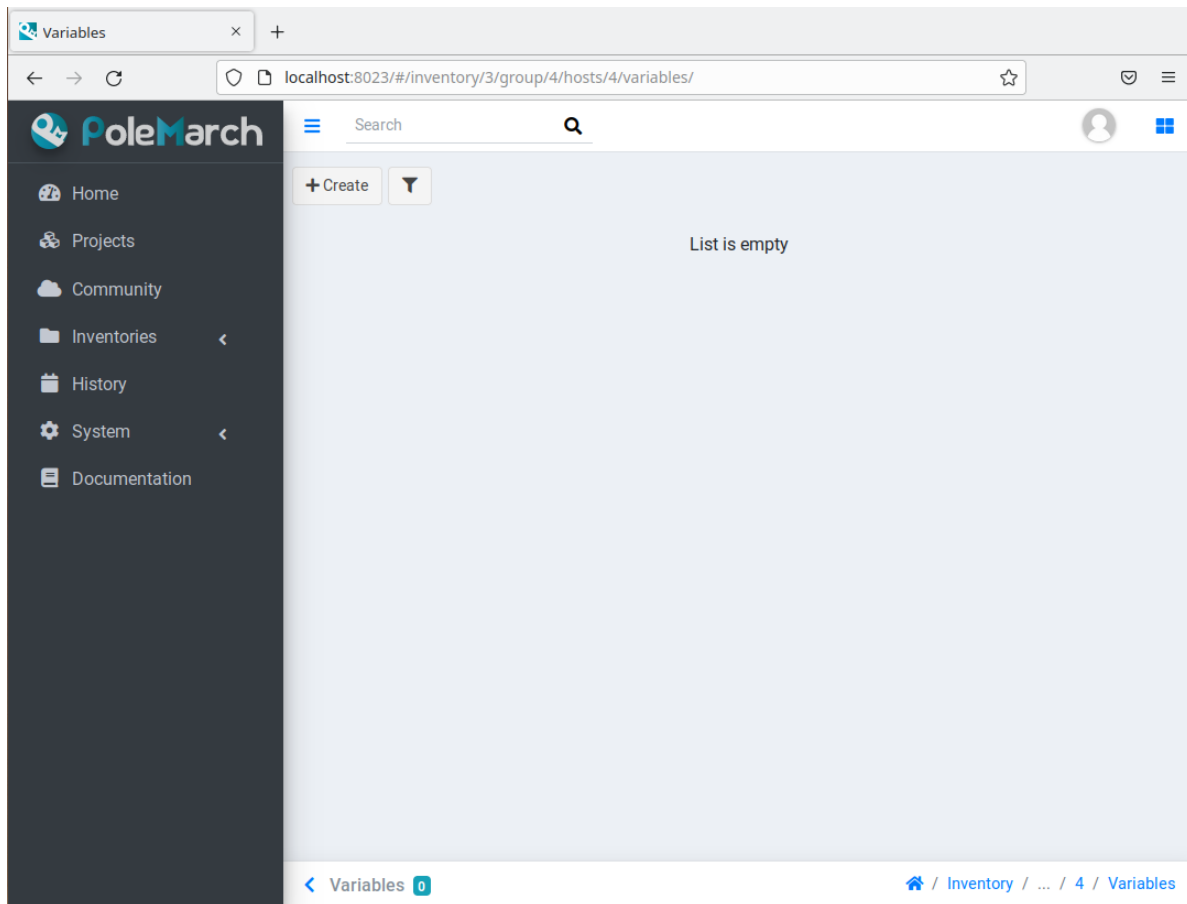
Let's look how you can create a variables for host, group and inventory.

Variables for inventory, group, hosts

The process of variable creation for inventory is the same as for group or host. So, let's look it at the example of variable creation for host.



To do it click the *Variables* button on the host page:



As you can see, there are no variables in the host's variables list now.

There is only one button here:

- *Create* - this button opens the page for creation of new variable.

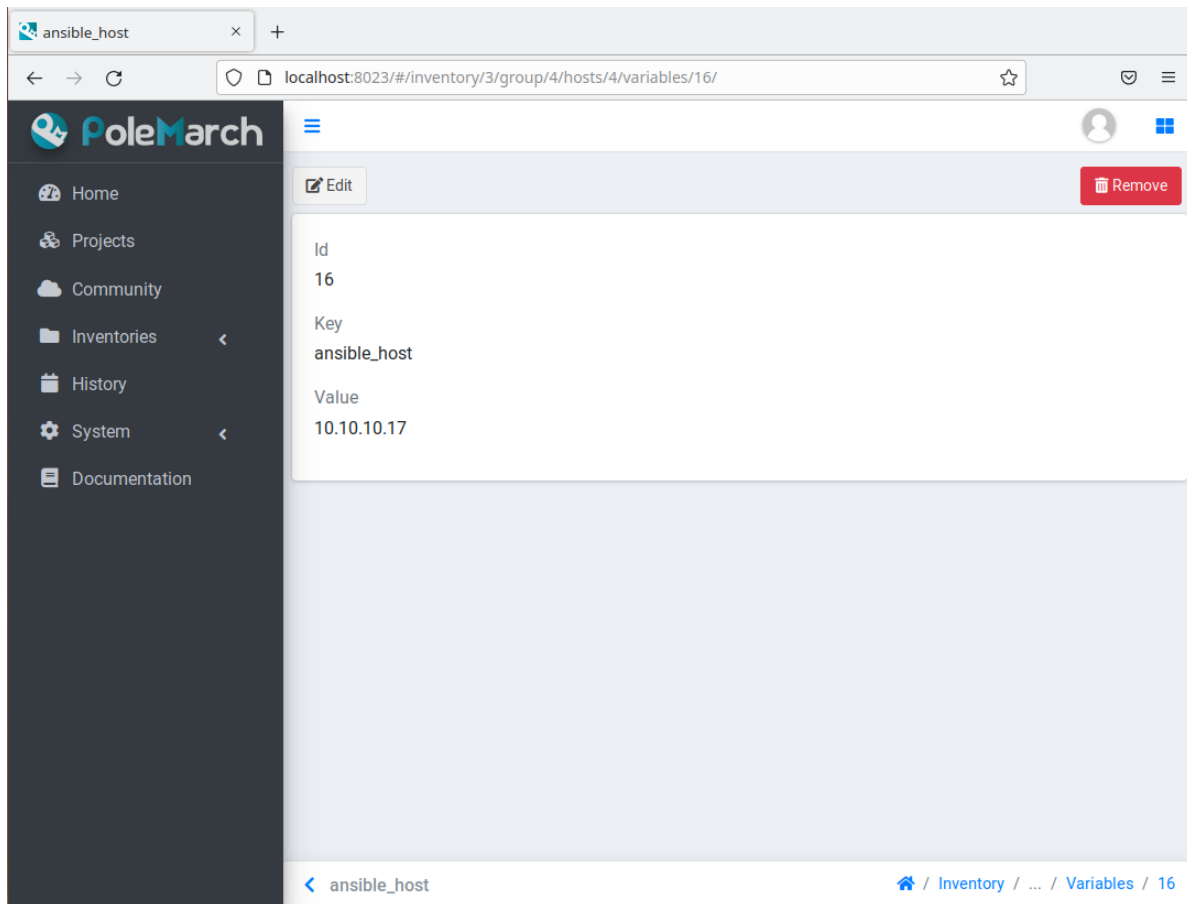
To create variable click the *Create* button:

The screenshot shows a web browser window with the URL `localhost:8023/#/inventory/3/group/4/hosts/4/variables/new/`. The interface features a dark sidebar on the left with the Polemarch logo and navigation links: Home, Projects, Community, Inventories, History, System, and Documentation. The main content area has a light blue header with a 'Save' button and a user profile icon. Below the header, there is a form with two input fields: 'Key' and 'Value', each with a clear button (X). The bottom of the page shows a breadcrumb trail: `< Variables` and `/ Inventory / ... / Variables / +`.

As you can see, the form of new host variable creation consists of following fields:

- **Key** - key of variable.
- **Value** - value of variable.

After variable creation you will see the next page:

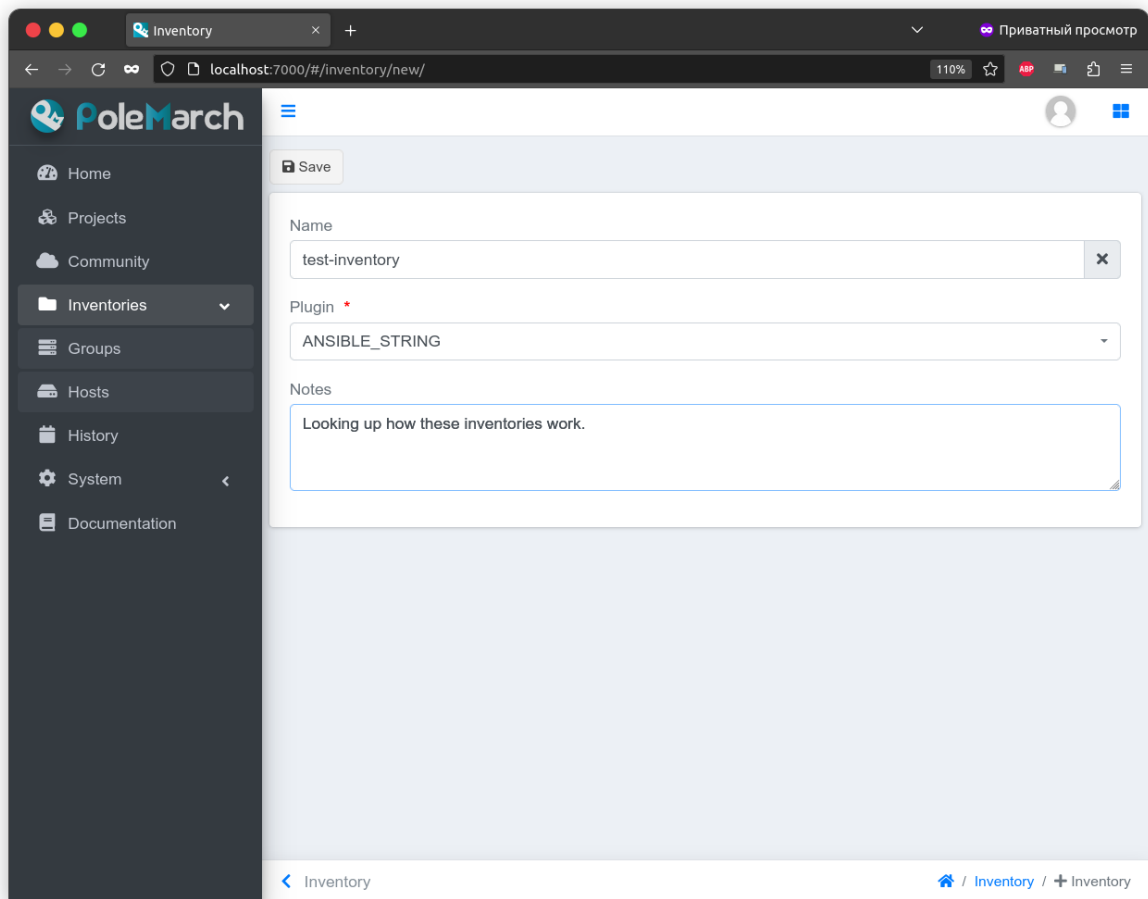


As you can see there is only 1 new field on this page:

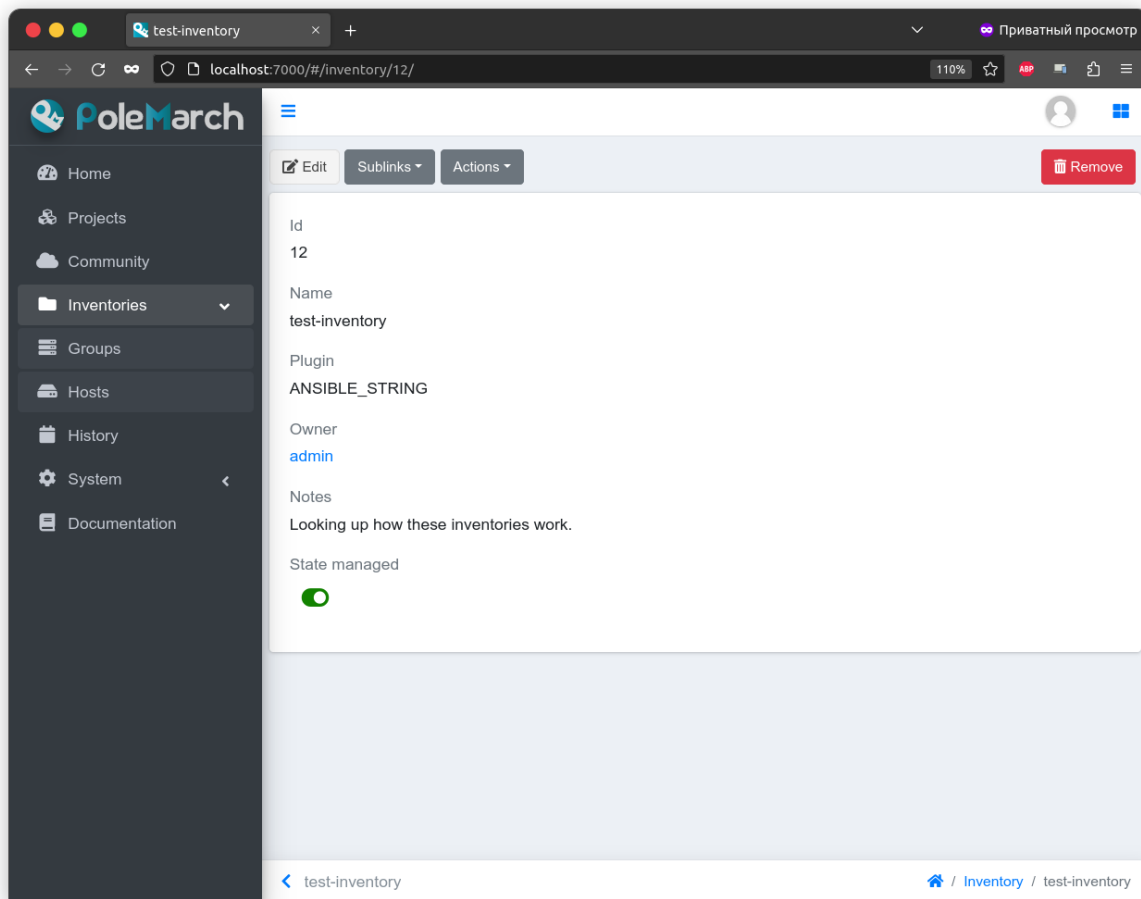
- **Id** - the unique key of object in database.

3.8.2 ANSIBLE_STRING inventory

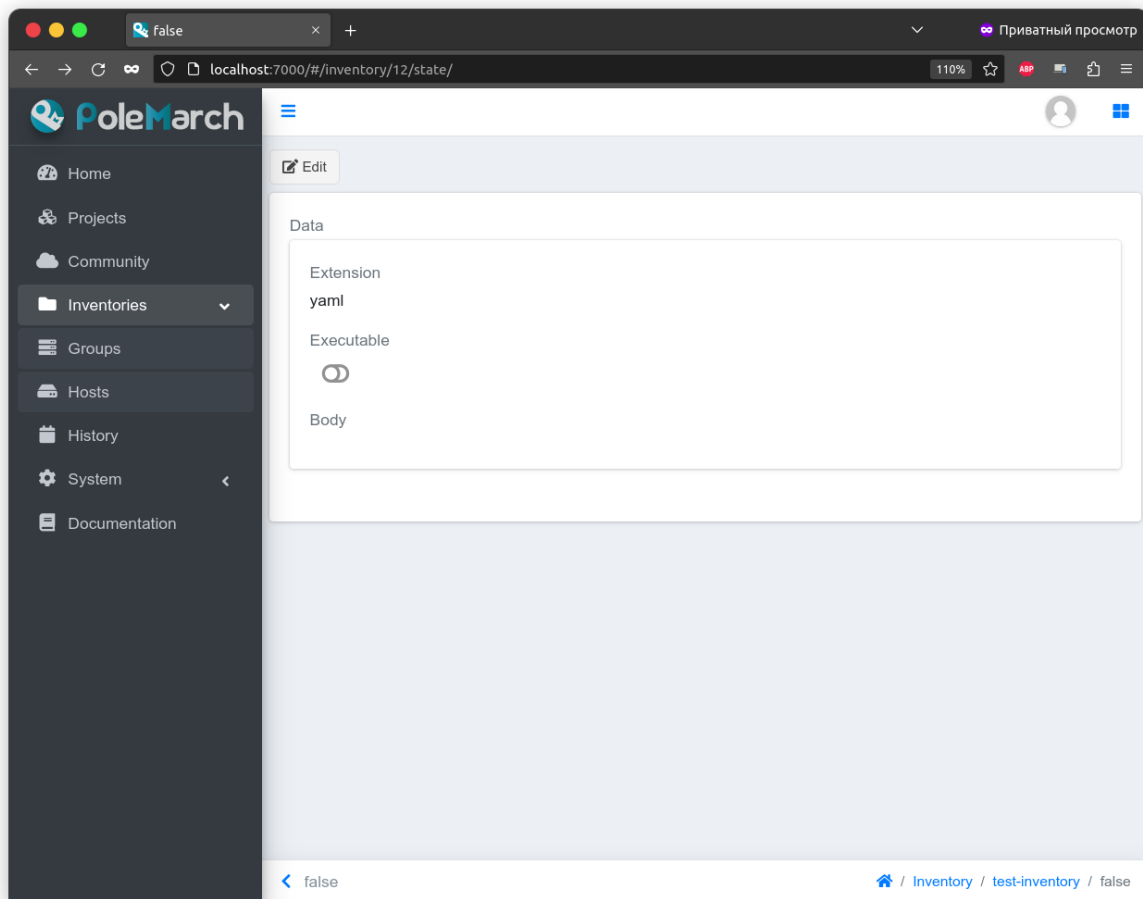
Let's return to inventory list page and create a new inventory:



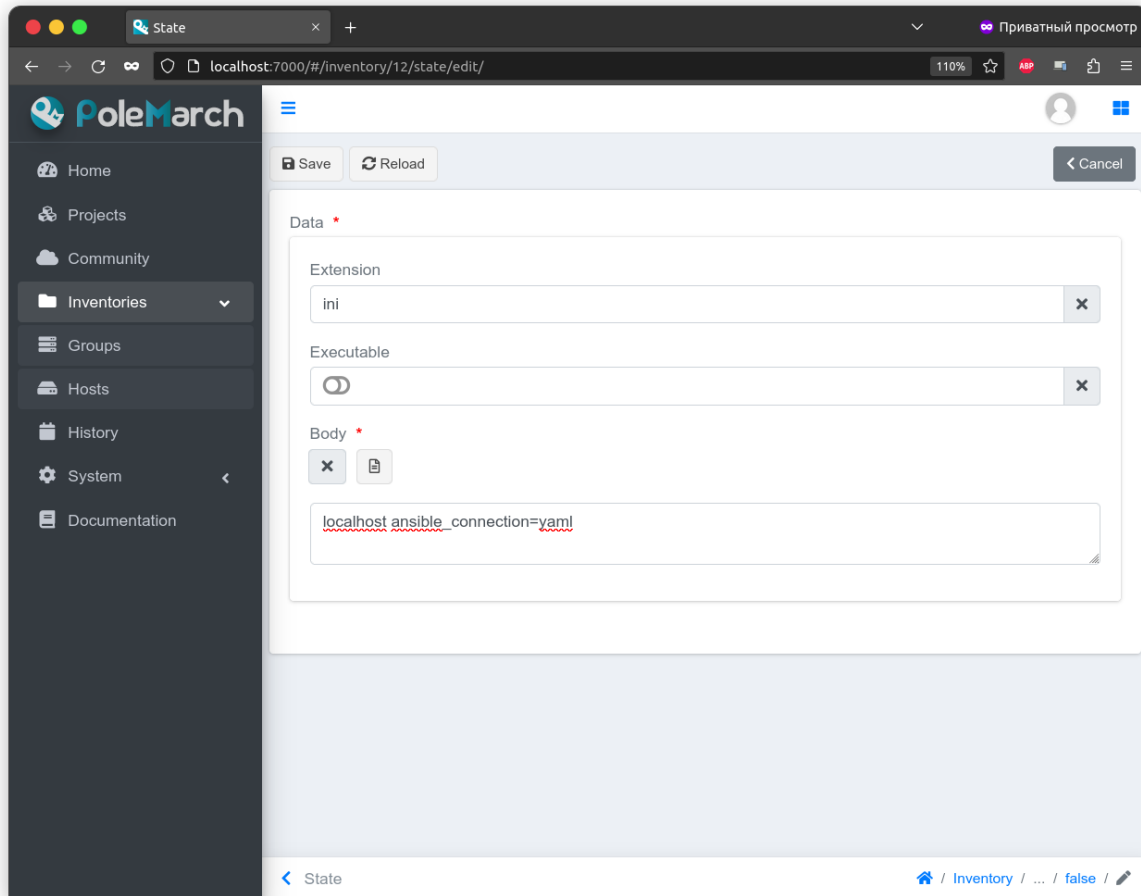
Our created inventory looks like this:



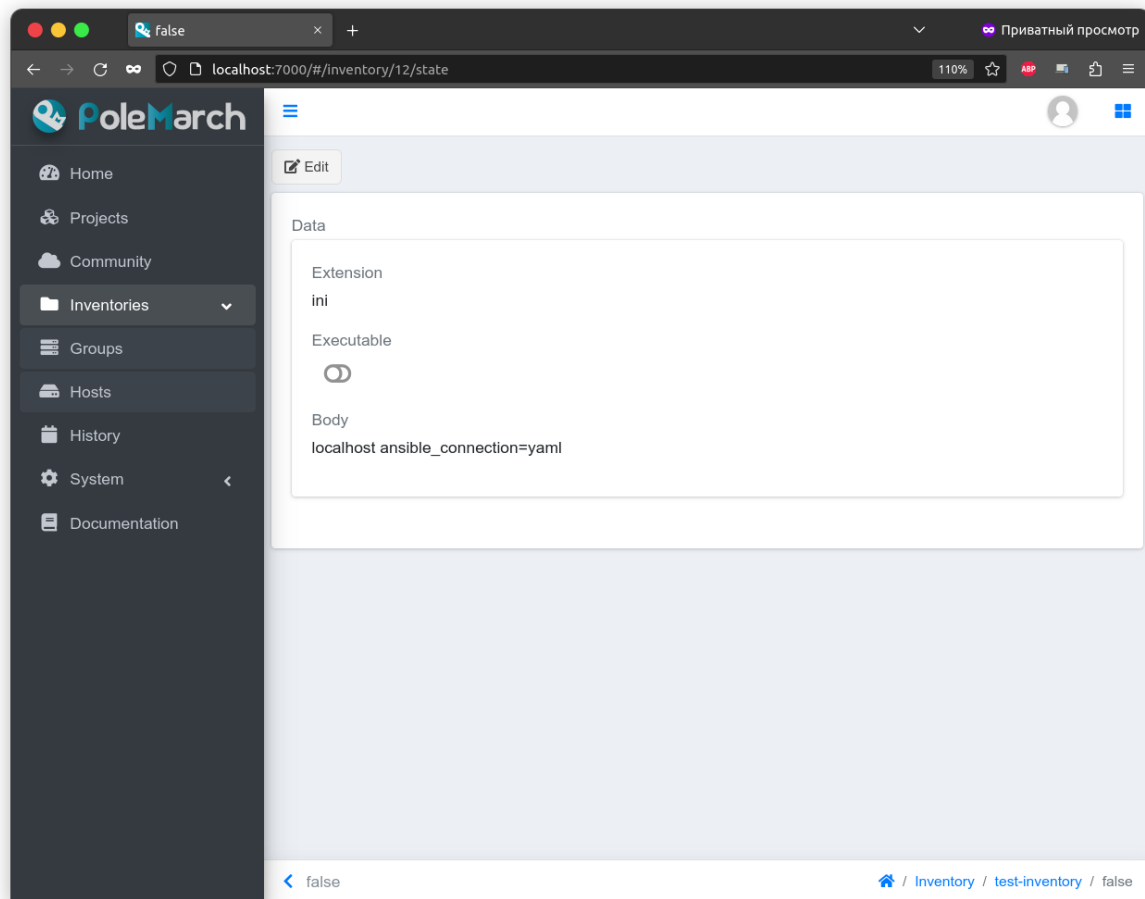
As you can see, compared to *POLEMARCH_DB* inventory, this one is state managed. Its sublinks will have the only *state* link. Let's click it and check the *state* page:



These types of inventory stores an extension of file, its body, filename and specifies either file should be executable or not. Let's edit the state. Click *Edit* button:



After saving:

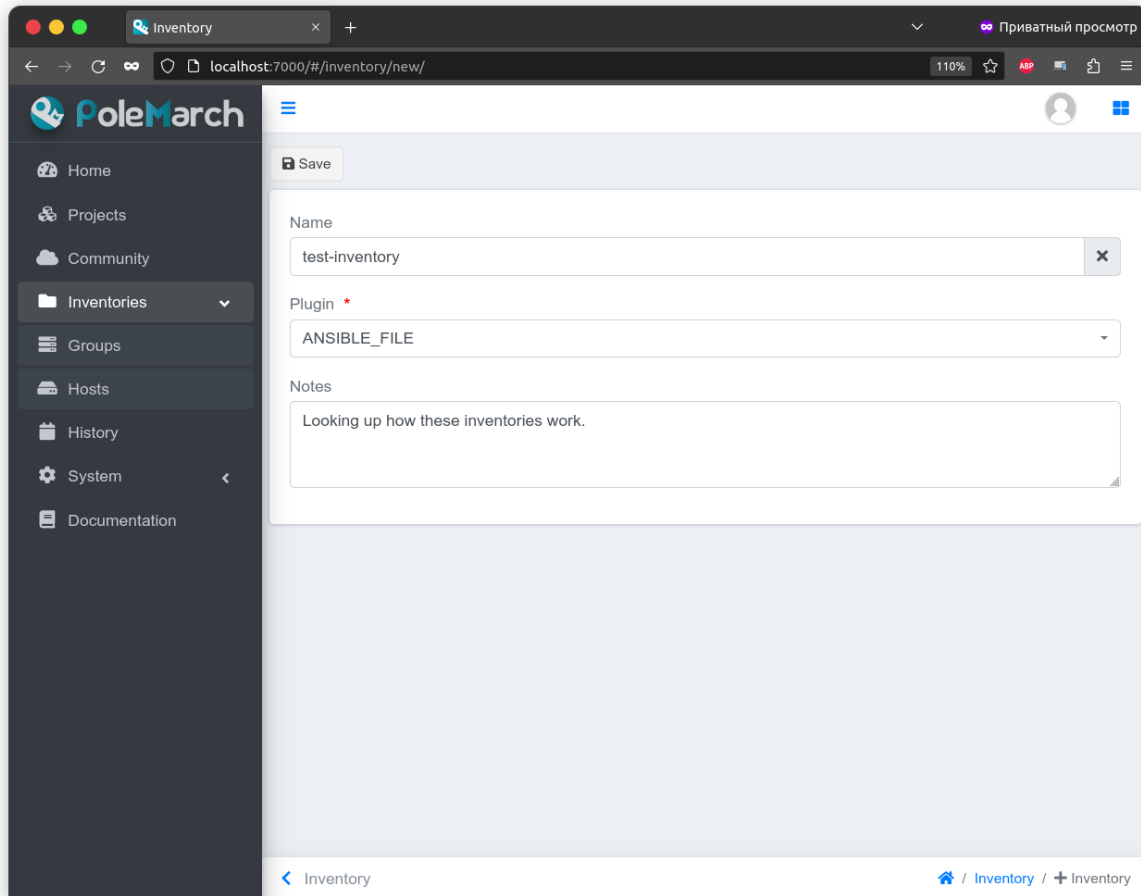


Now inventory is ready for using.

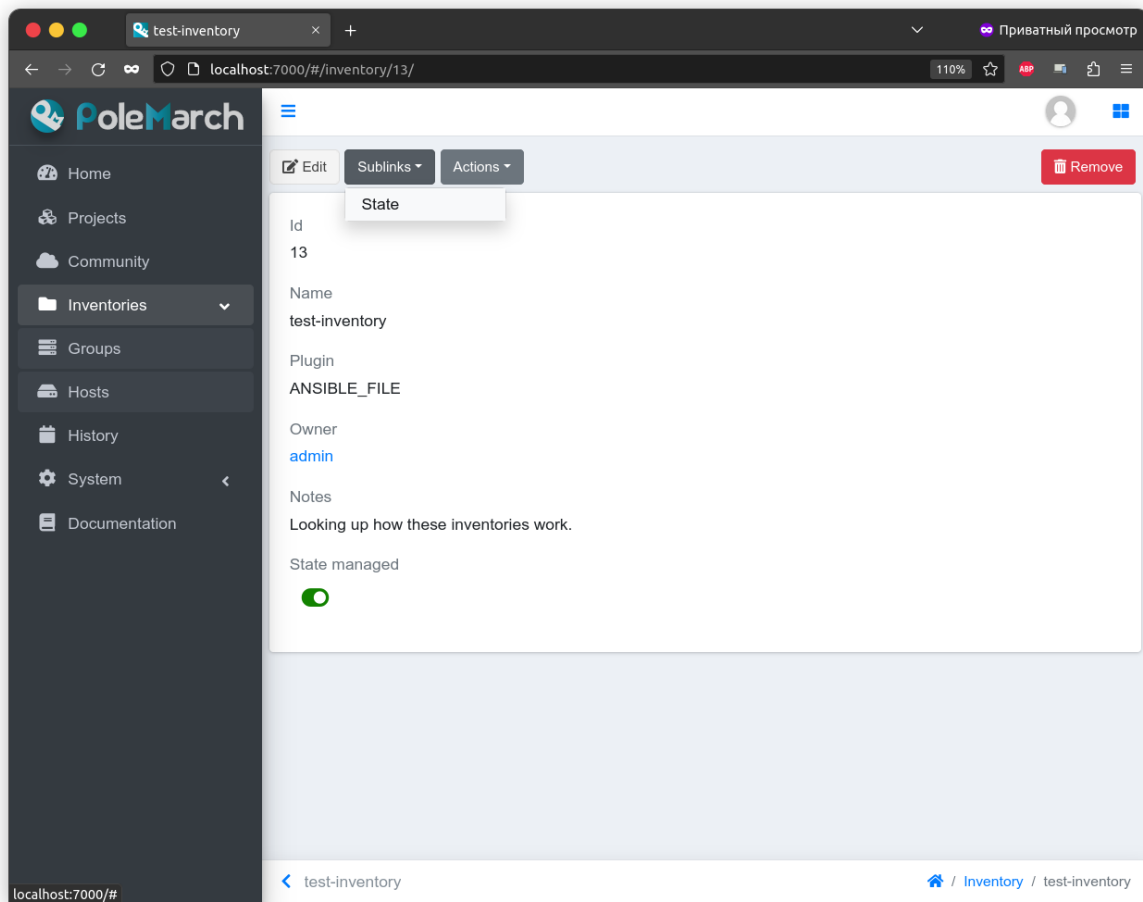
This type of inventory is used to store the completed file in the project directory at the time the task is run. It is helpful when you need to use one of the dynamic ansible inventory plugins.

3.8.3 ANSIBLE_FILE inventory

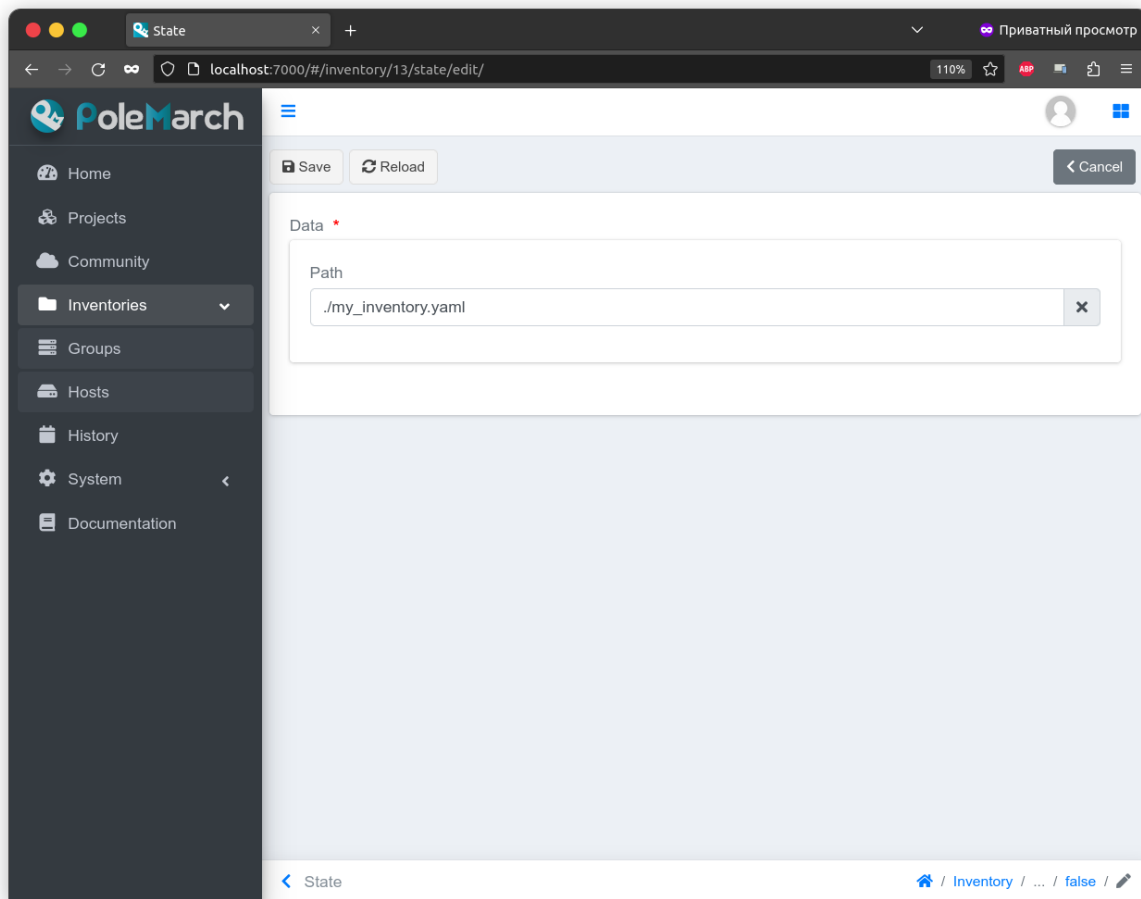
Again, returning to inventory list page, let's create an inventory with *ANSIBLE_FILE* plugin:



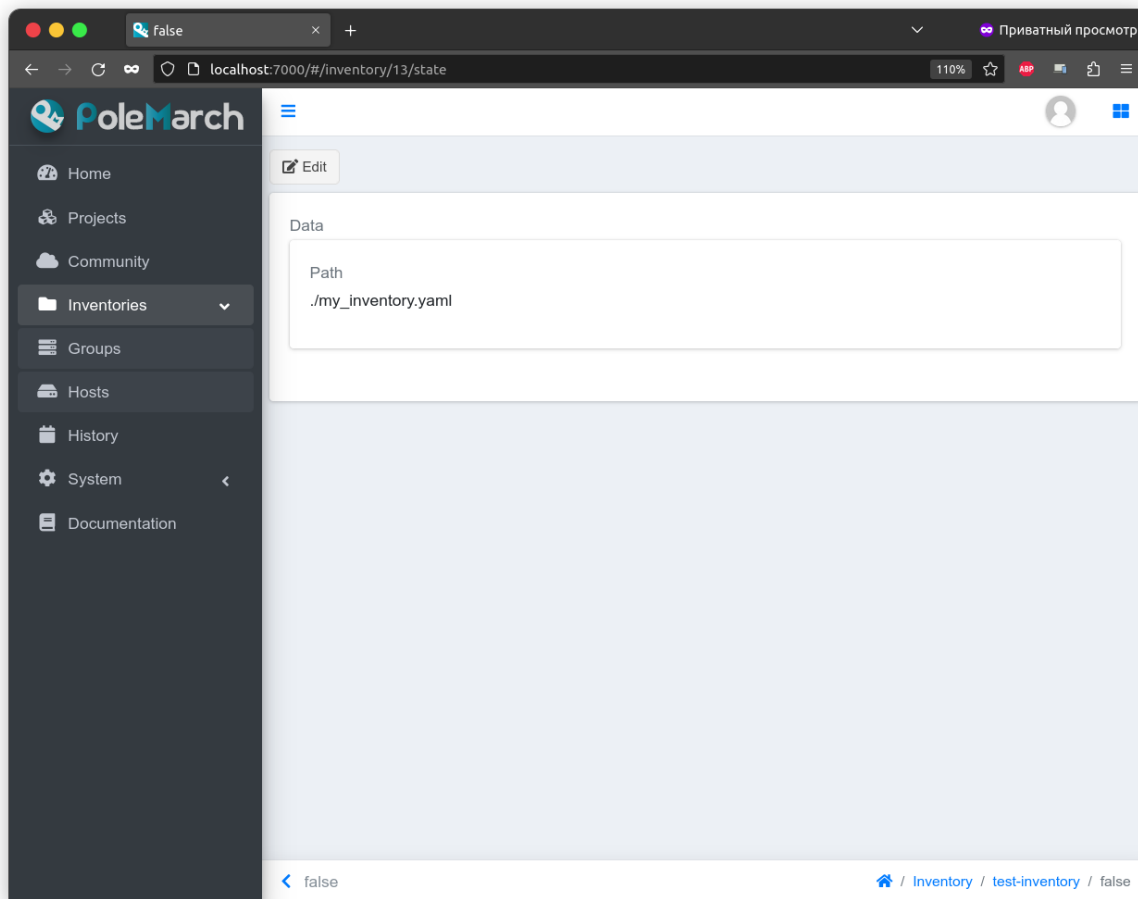
These inventories are also stores data in state as *State managed* hints us:



Let's edit the state (click *Sublinks*, then *State* and then *Edit*):



Here we specify the relative path of inventory file which should be inside the directory with project that will be executed.



Done. Inventory is ready for use.

This type of inventory is used to use file from the project directory at the time the task is run. It is helpful when you have GitOps infrastructure where project files is single store of truth.

Note: The use of strings when specifying an inventory path to describe a launch will be deprecated in future releases and removed from the interface. This inventory type is a direct string replacement.

3.9 Import inventory

If you have an inventory file and you want to add items from it to Polemarch, you can do it quickly using “Import Inventory”.

Note: Import action may be available or not depending on which plugin inventory uses. For example, built-in *POLEMARCH_DB*, *ANSIBLE_STRING* supports import but *ANSIBLE_FILE* is not.

Here we will look at import with an example of *POLEMARCH_DB* inventory. Import with *ANSIBLE_STRING* inventories works similar except that imported data saves to inventory state.

For example, let’s use next inventory file:

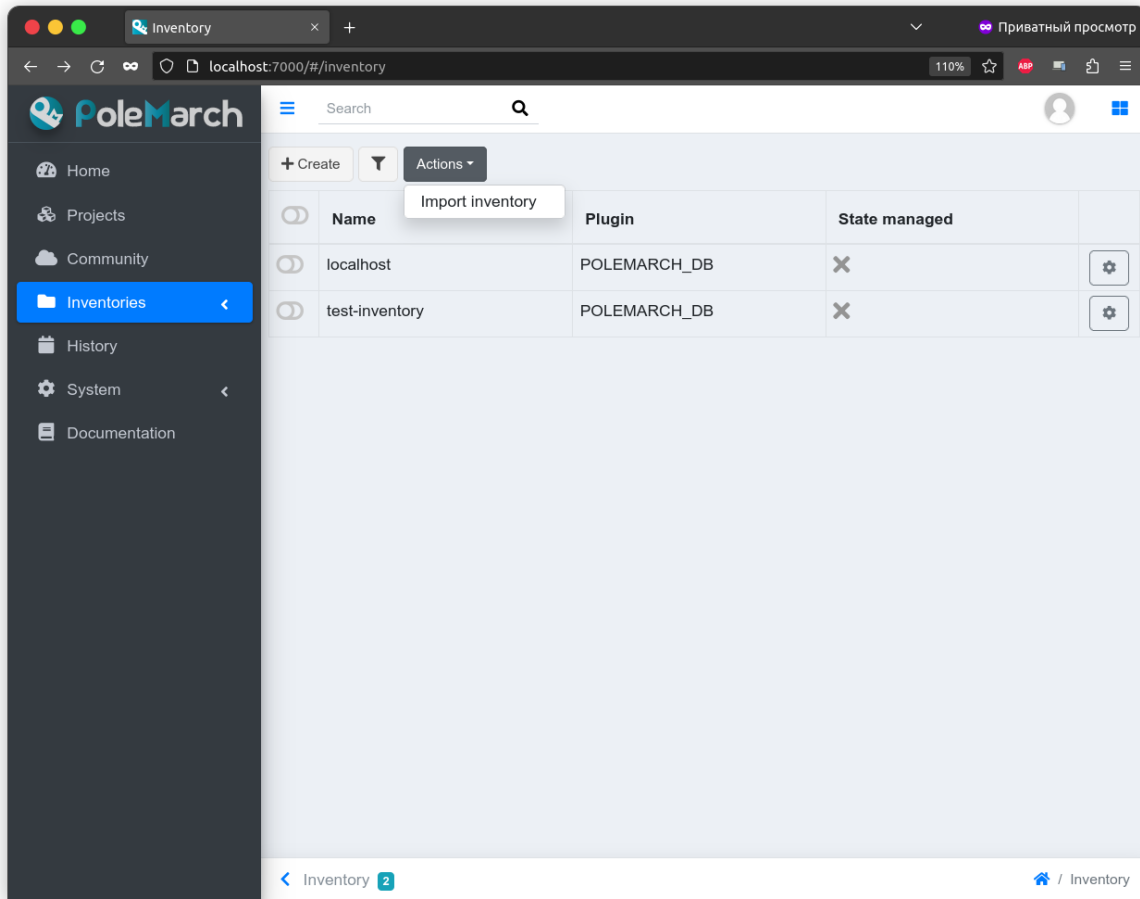
```
[imported-test-group]
imported-test-host ansible_host=10.10.10.17
```

(continues on next page)

(continued from previous page)

```
[imported-test-group:vars]
ansible_user=ubuntu
ansible_ssh_private_key_file=example_key
```

To import inventory you should open inventory list page:



And click the *Import inventory* button. Then you will see the next page:

The screenshot shows the 'Import inventory' form in the Polemarch web application. The form is located on the right side of the interface, with a dark sidebar on the left containing navigation links. The form has three main sections: 'Name', 'Plugin', and 'Data'. The 'Name' section has a text input field containing 'imported-test-inventory'. The 'Plugin' section has a dropdown menu set to 'POLEMARCH_DB'. The 'Data' section has a text area containing the following Ansible inventory content:

```
[imported-test-group]
imported-test-host ansible_host=10.10.10.17

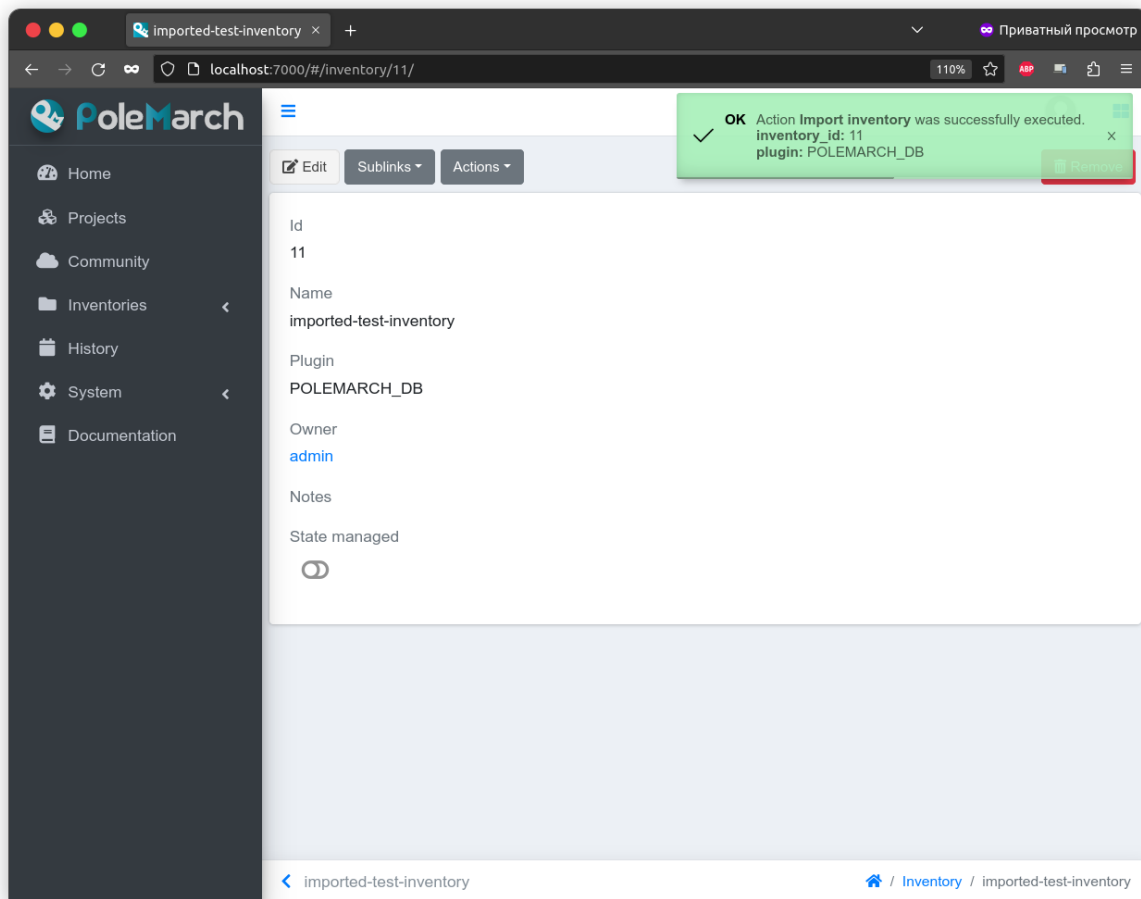
[imported-test-group:vars]
ansible_user=ubuntu
ansible_ssh_private_key_file=example_key
```

The browser address bar shows 'localhost:7000/#/inventory/import_inventory/'. The bottom of the page shows a breadcrumb trail: 'Inventory / Import inventory'.

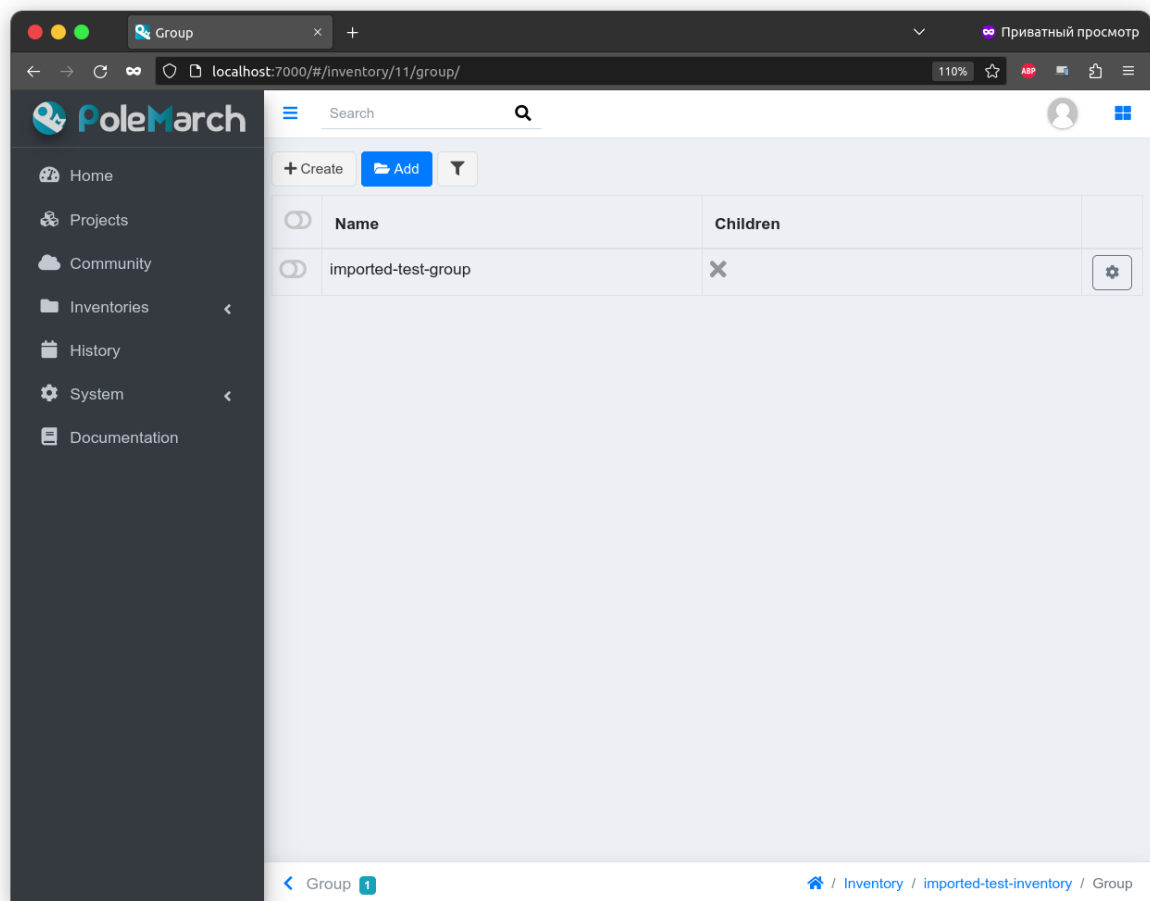
As you can see, the form of “Import inventory” action consists of two fields:

- **Name** - name of your inventory.
- **Row data** - content of your inventory file.

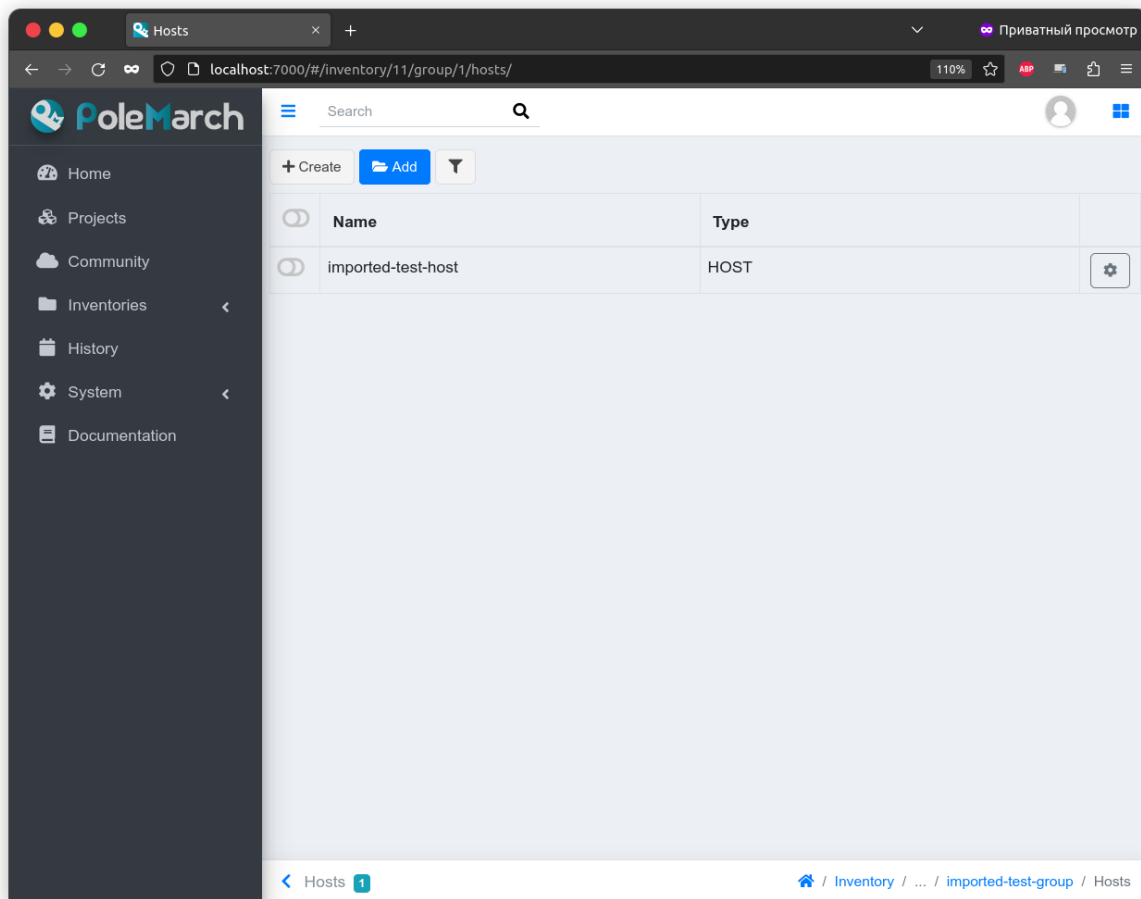
After filling of all fields you should click the *Import inventory* button and then you will see page of your imported inventory:



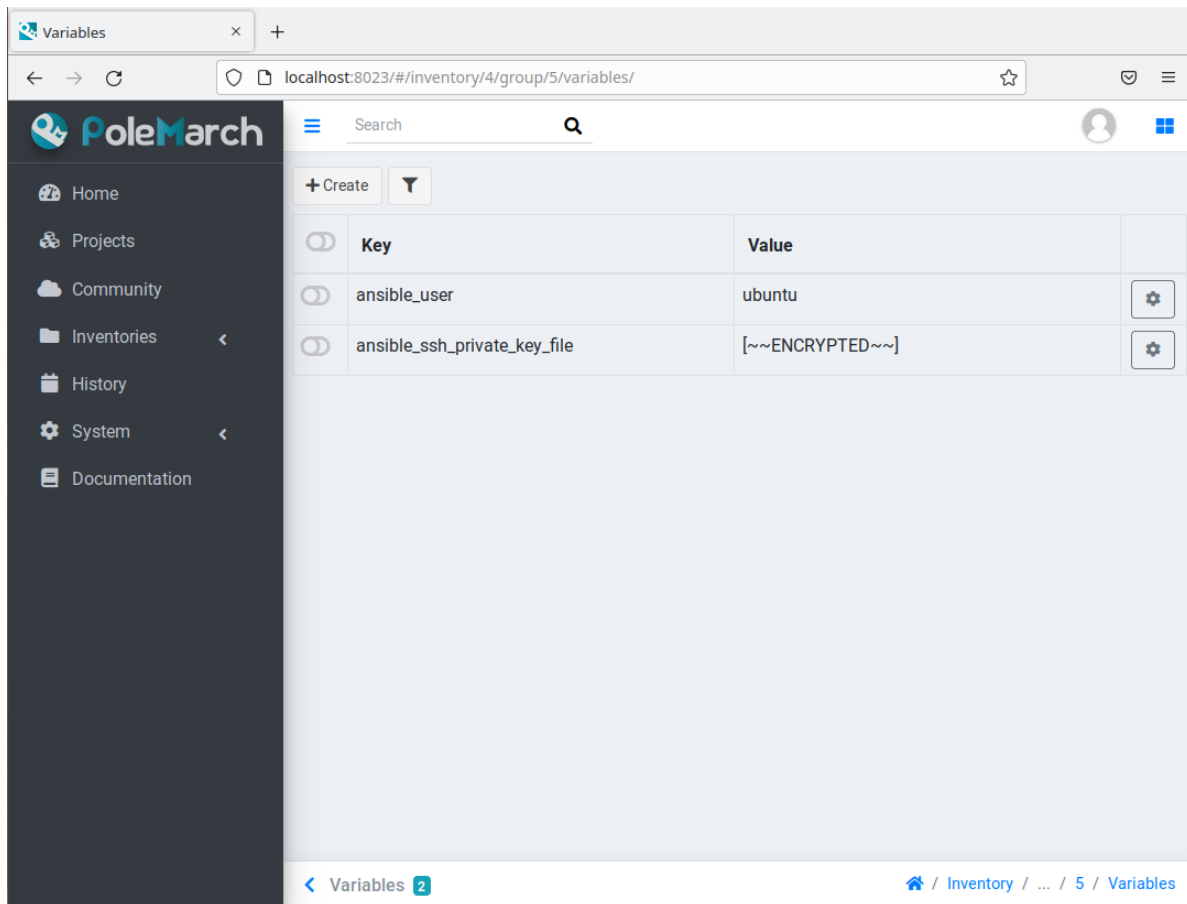
This inventory includes “imported-test-group” from imported inventory file:



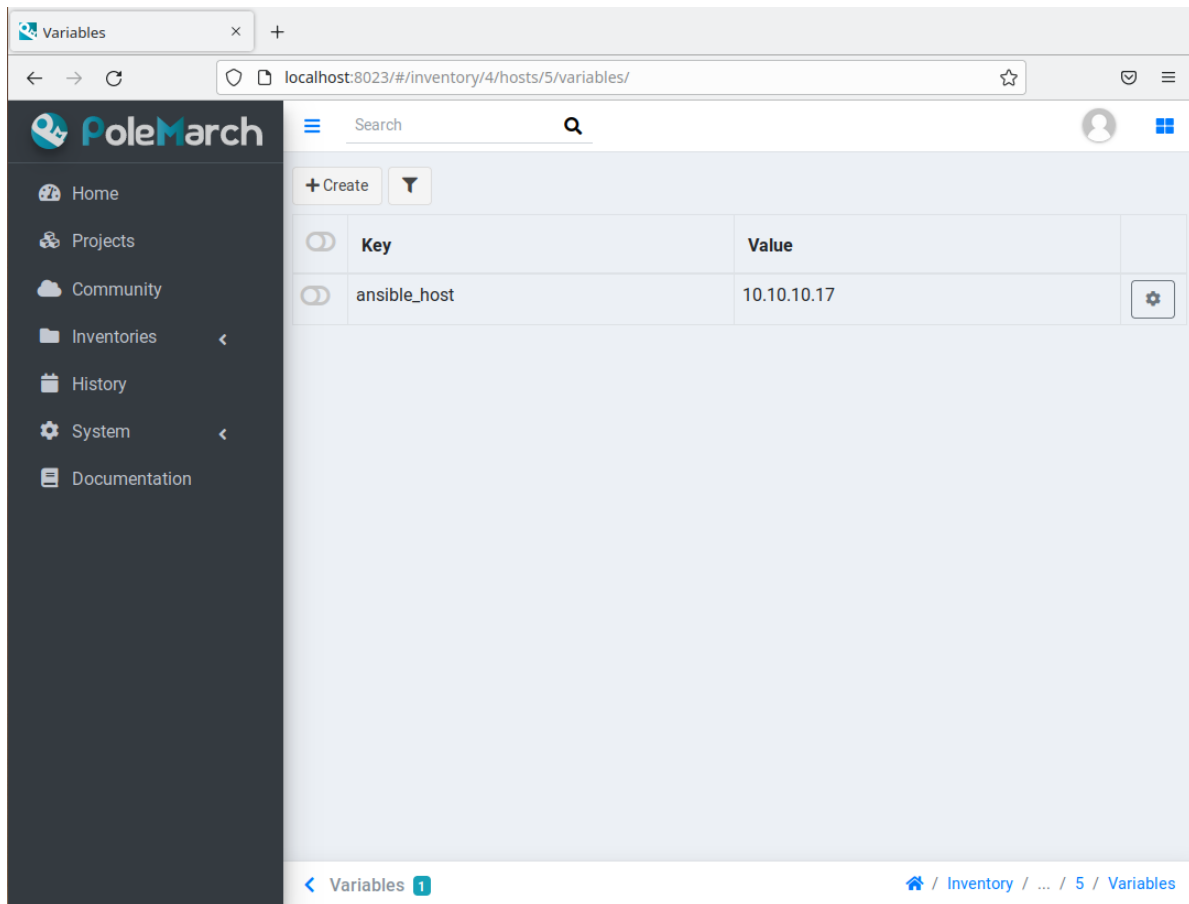
And “imported-test-group” includes “imported-test-host” from imported inventory file:



Also “imported-test-group” includes variables “ansible_user” and “ansible_ssh_private_key_file” from imported inventory file:



“imported-test-host” includes variable “ansible-host” from imported inventory file:



3.10 .polemarch.yaml

`.polemarch.yaml` is a file for a quick deployment of Polemarch project. By quick deployment of Polemarch project we mean automatic creation of some templates for this project (during project sync) and using of additional interface for quick task execution.

`.polemarch.yaml` is not required file for Polemarch work, but if you want to use features of `.polemarch.yaml`, you have to store it in the base directory of (GIT, MANUAL, TAR) project.

Structure of `.polemarch.yaml` consists of next fields:

- **sync_on_run** - boolean, it means to get or not to get settings from `.polemarch.yaml` during each project sync.
- **templates** - dictionary, consists of template objects (their structure is similar to template's API structure except the 'name' field).
- **templates_rewrite** - boolean, it means to rewrite or not to rewrite templates in project with names equal to templates' names from `.polemarch.yaml`.
- **view** - dictionary, it is a description of web-form, that will be generated from `.polemarch.yaml`. It consists of:
 - **fields** - dictionary, it consists of objects, that describe fields properties:
 - * **title**: title of field, that Polemarch will show in web-form.
 - * **default**: default value of field. Default: ' ' - for strings, 0 - for numbers.
 - * **format**: format of field. For today next field formats are available: string, integer, float, boolean. Default: string.

- * **help**: some help text for this field.
- **playbooks** - dictionary, it consists of objects, that describes playbook properties:
 - * **title**: title of playbook, that Polemarch will use during playbook execution.
 - * **help**: some help text for this playbook.

Example of .polemarch.yaml:

```
---
sync_on_run: true
test-module:
  plugin: ANSIBLE_MODULE
  options:
    default:
      module: ping
      group: all
      inventory: localhost,
    uptime:
      module: ping
      args: uptime
      inventory: 127.0.0.1,
test-playbook:
  plugin: ANSIBLE_PLAYBOOK
  options:
    update:
      playbook: main.yml
      become: true
templates_rewrite: true
view:
  fields:
    string:
      title: Field string
      default: 0
      format: string
      help: Some help text
    integer:
      title: Field integer
      default: 0
      format: integer
      help: Some help text
    float:
      title: Field float
      default: 0
      format: float
      help: Some help text
    boolean:
      title: Field boolean
      default: 0
      format: boolean
      help: Some help text
    enum_string:
      title: Field enum_string
      default: 0
      format: string
      help: Some help text
      enum: ['Choice1', 'Choice2', 'Choice3']
    unknown:
```

(continues on next page)

(continued from previous page)

```

    title: Field unknown
    default: 0
    format: invalid_or_unknown
    help: Some help text
  playbooks:
    main.yml:
      title: Execute title
      help: Some help text

```

Note: Since Polemarch v3.0.0 execution templates system has been changed a lot. Along with it, the format for describing templates has also changed. We still support old format like in the next example:

```

...
templates:
  test-module:
    notes: Module test template
    kind: Module
    data:
      group: all
      vars: {}
      args: ''
      module: ping
      inventory: localhost,
    options:
      uptime:
        args: uptime
        module: shell
  test playbook:
    notes: Playbook test template
    kind: Task
    data:
      vars: {"become": true}
      playbook: main.yml
      inventory: localhost,
    options:
      update: {"playbook": "other.yml"}
...

```

Here **kind** (Task, Module) is converted to **plugin** (ANSIBLE_PLAYBOOK, ANSIBLE_MODULE)

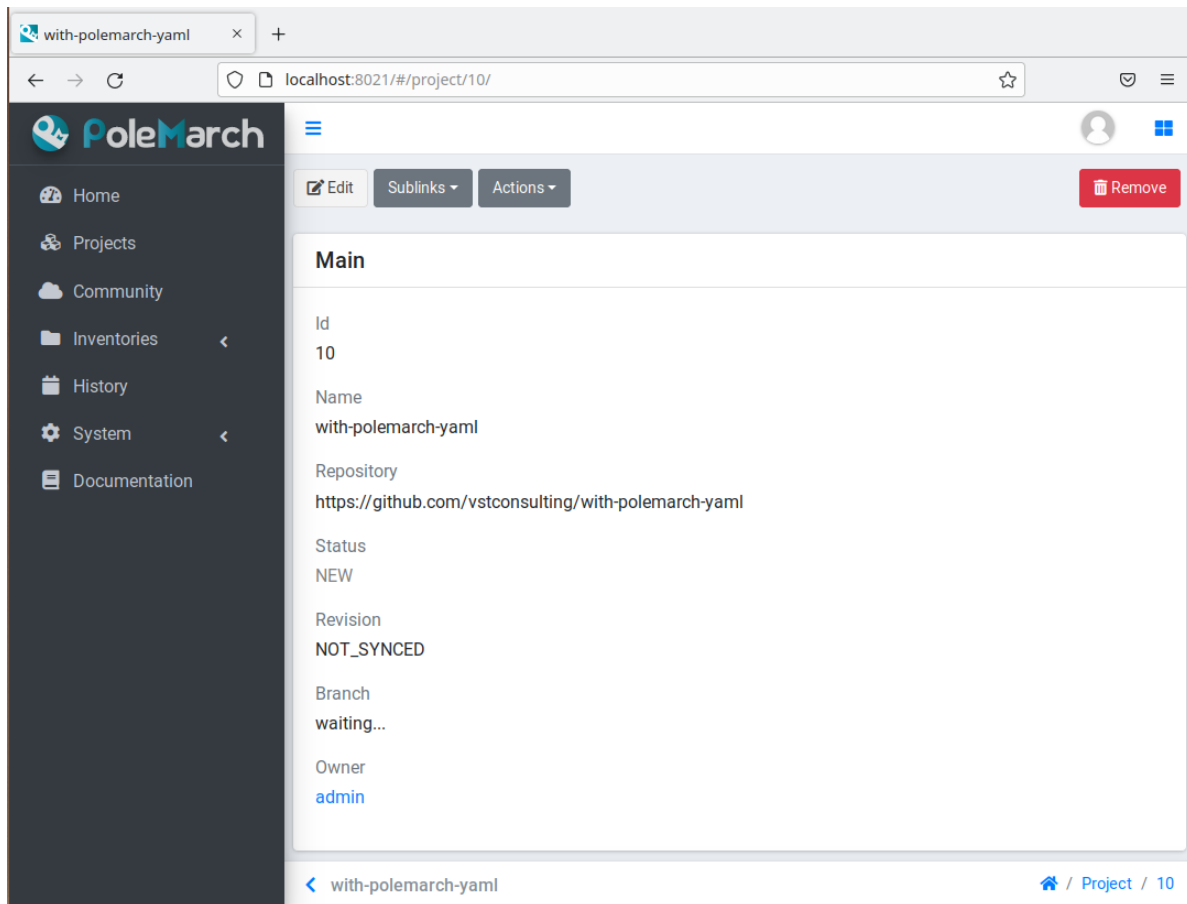
But keep in mind that this format has been deprecated and it's desirable to rewrite templates to match the new format.

In GUI process of working with `.polemarch.yaml` will be the following:

Firstly, you need to create a project with `.polemarch.yaml` (or to add `.polemarch.yaml` to existing project). For example, let's create new GIT project, that has in its base directory `.polemarch.yaml` file from the example above:

The screenshot shows a web browser window with the address bar displaying `localhost:8023/#/project/new/`. The page features a dark sidebar on the left with the Polemarch logo and navigation links: Home, Projects, Community, Inventories, History, System, and Documentation. The main content area is titled 'Project' and contains a 'Save' button at the top left. Below this, there are two side-by-side form panels. The 'General' panel on the left includes fields for 'Name' (containing 'with-polemarch-yaml'), 'Repo type' (a dropdown menu set to 'GIT'), and 'Directory with playbooks'. The 'Repository' panel on the right includes fields for 'Repo url' (containing 'https://github.com/akhmadullin/with-pole'), 'Branch for GIT (branch/tag/SHA) or TAR (subdir)', and 'Repo auth type' (a dropdown menu set to 'NONE'). Each input field has a small 'x' icon to its right. At the bottom of the page, there is a breadcrumb trail showing a home icon, 'Project', and a plus icon.

In the field 'Repo url' you should insert URL of project that has in its base directory `.polemarch.yaml` file. After project creation you will see the ordinary project page:



Then you need to click the *Sync* button. After project synchronization you will see the next page:

The screenshot shows the Polemarch web interface in a browser window. The address bar indicates the URL is `localhost:8021/#/project/10/`. The left sidebar contains navigation links: Home, Projects, Community, Inventories, History, System, and Documentation. The main content area is titled "Quick playbook execution form" and includes the following fields:

- Inventory ?**: A dropdown menu with "Inventory" selected.
- User ?**: A text input field with a clear button (X).
- Private key ?**: A text input field with a clear button (X) and a file upload icon.
- Field boolean ?**: A radio button input field with a clear button (X).

The right sidebar, titled "Main", displays project details:

- Id**: 10
- Name**: with-polemarch-yaml
- Repository**: <https://github.com/vstconsulting/with-polemarch-yaml>
- Status**: OK
- Revision**: d1b5526ee58efab4e2d9b8b1c7ebccac53b12fb7
- Branch**: master
- Owner**: [admin](#)

At the bottom of the main content area, there is a button labeled "Execute title".

This screenshot shows the same Polemarch web interface, but with a different set of fields in the "Quick playbook execution form":

- Field boolean ?**: A radio button input field with a clear button (X).
- Field enum_string ?**: A dropdown menu.
- Field float ?**: A text input field with a clear button (X).
- Field integer ?**: A text input field with a clear button (X).
- Field string ?**: A text input field with a clear button (X).
- Field unknown ?**: A text input field with a clear button (X).

The "Main" sidebar on the right remains the same, showing project details for "with-polemarch-yaml". The "Execute title" button is still present at the bottom of the form.

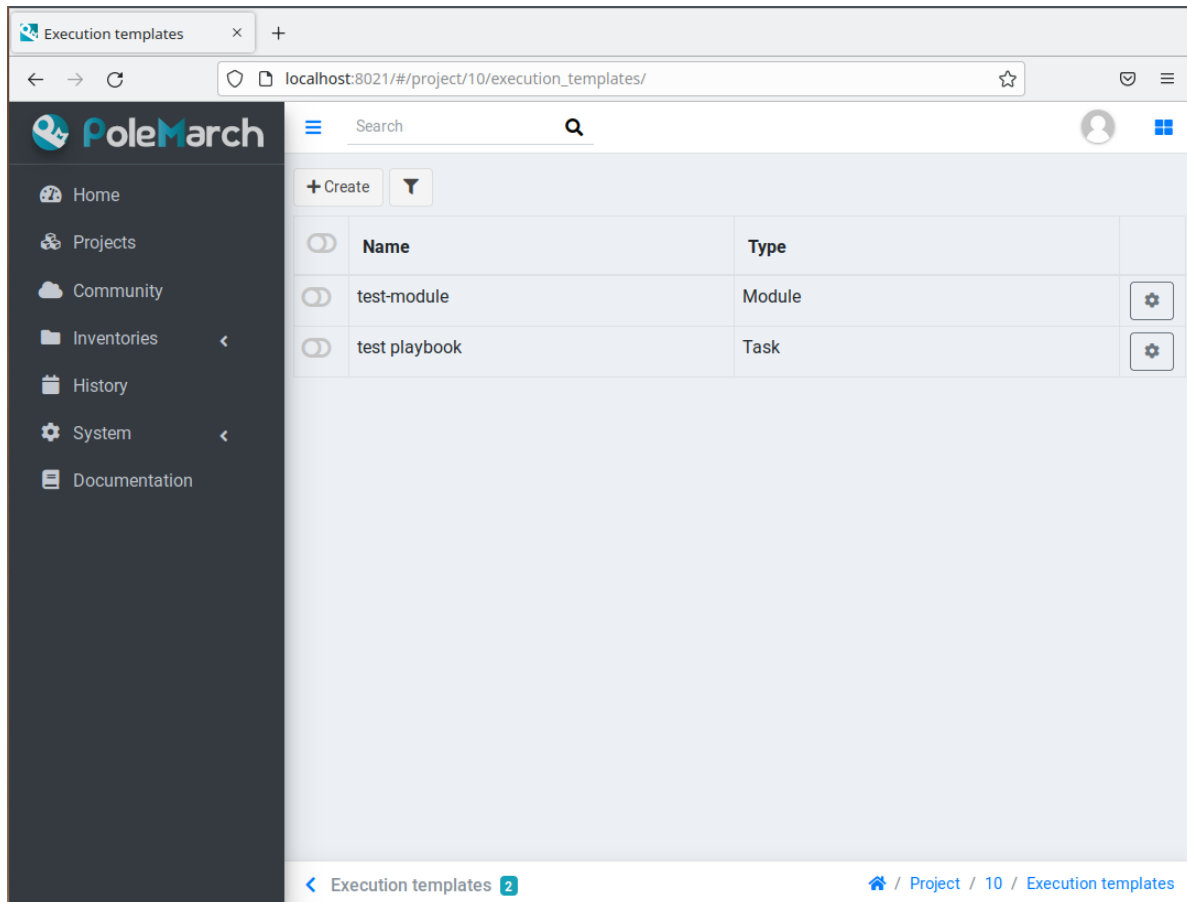
As you can see, all fields that we mentioned in the example `.polemarch.yaml` were added to this web-form. Values of all these fields will be used as ansible `extra_vars` parameter during task execution.

The remaining fields mean following:

- **Inventory** - it can be inventory from Polemarch system, list of hosts, that are separated by `,`, or path to your inventory in project folder.
- **User** - set it if your don't have `ansible_user` variable in your inventory.
- **Key file** - set it if your don't have `ansible_ssh_private_key_file` variable in your inventory.

Button *Execute title* will execute playbook "main.yml", as we mentioned in `.polemarch.yaml` file.

Also templates from `.polemarch.yaml` were added to just created Polemarch project.

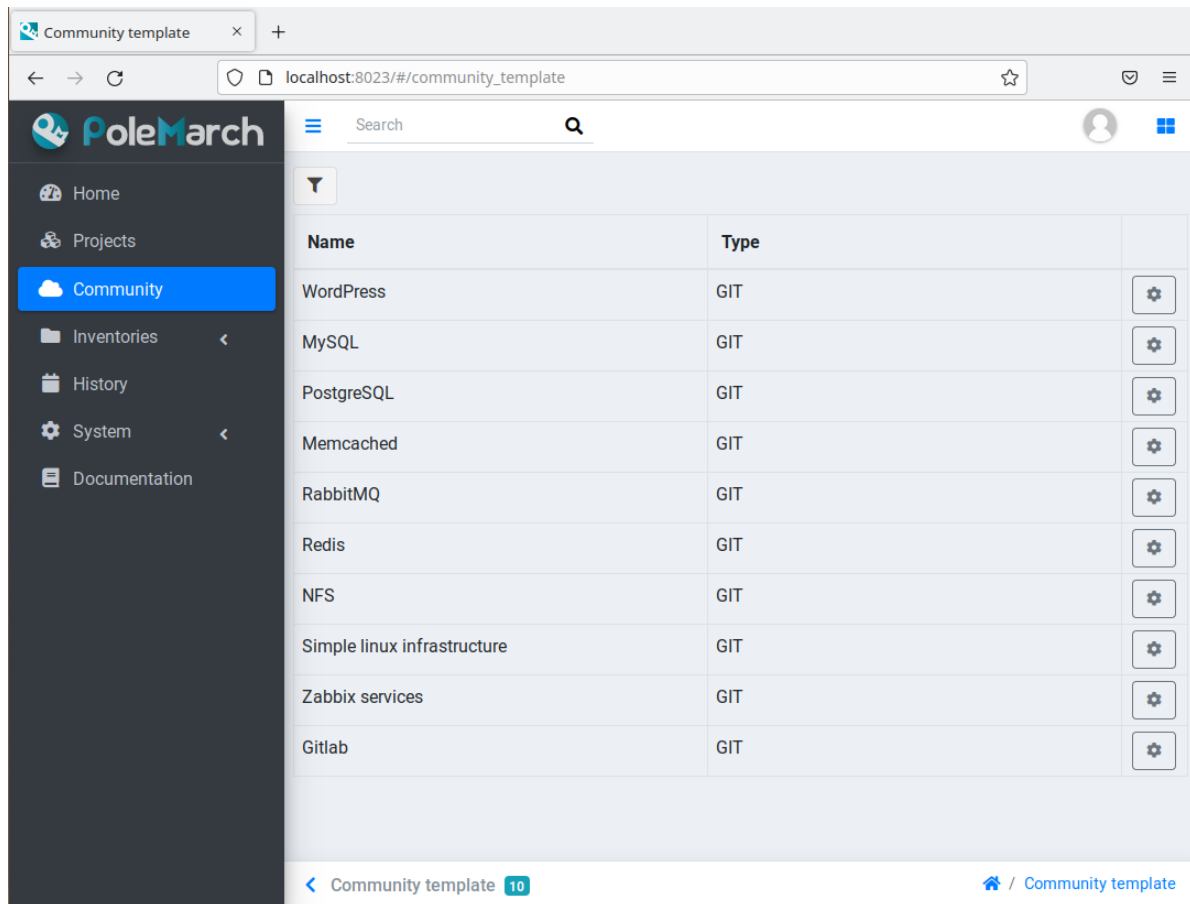


3.11 Community project samples

Polemarch has his own list of community repositories, which provide you with stuff, that is able to deploy different services in extremely quick and convenient way. Community project sample is an entity, that clones repository from community list into your Polemarch. Some community project samples have only list of playbooks, but most have `.polemarch.yaml` file, that helps you to deploy tasks even faster.

Let's look how it works on examples.

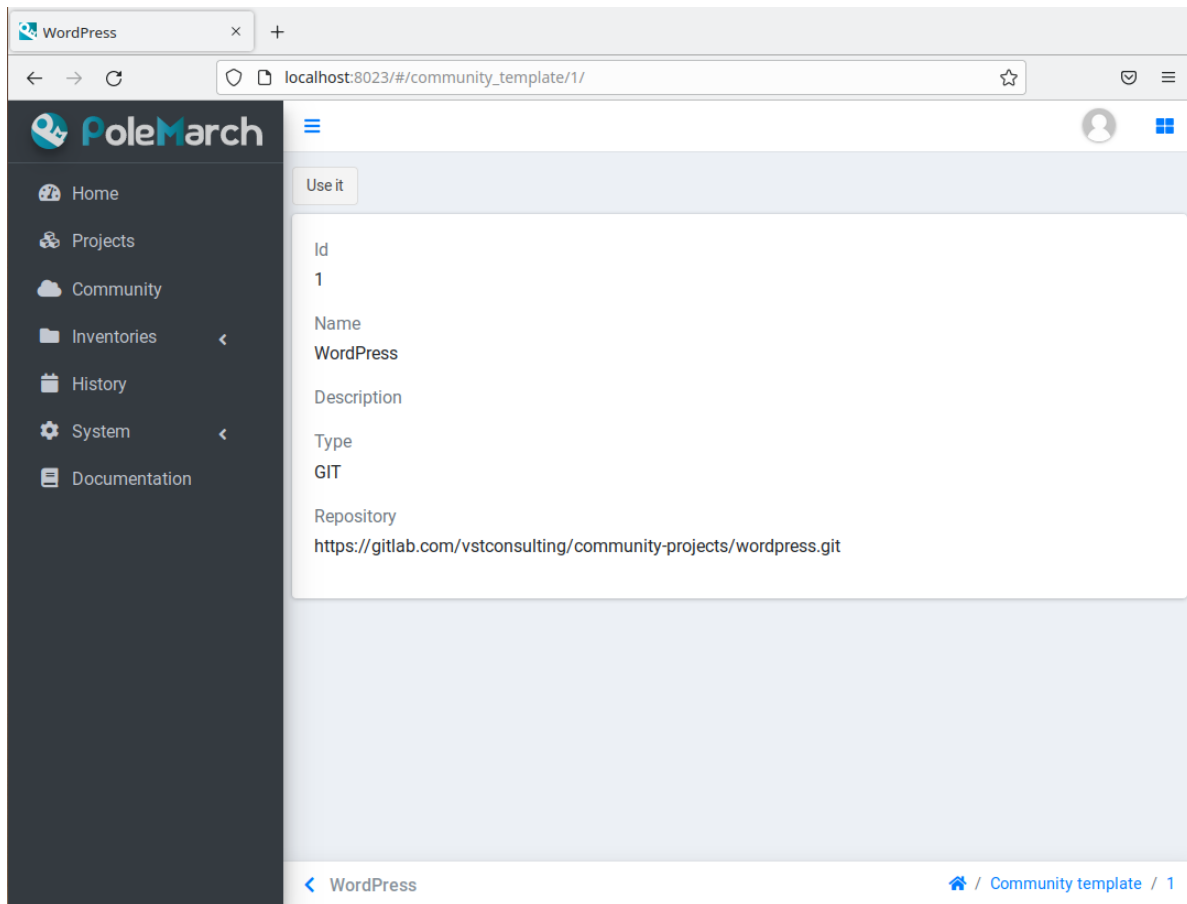
Firstly, open page with community project samples list.



As you can see, there is a table with list of community repositories, that consists of following columns:

- **Name** - name of community repository.
- **Type** - type of community repository.

Let's open "WordPress" community repository page.



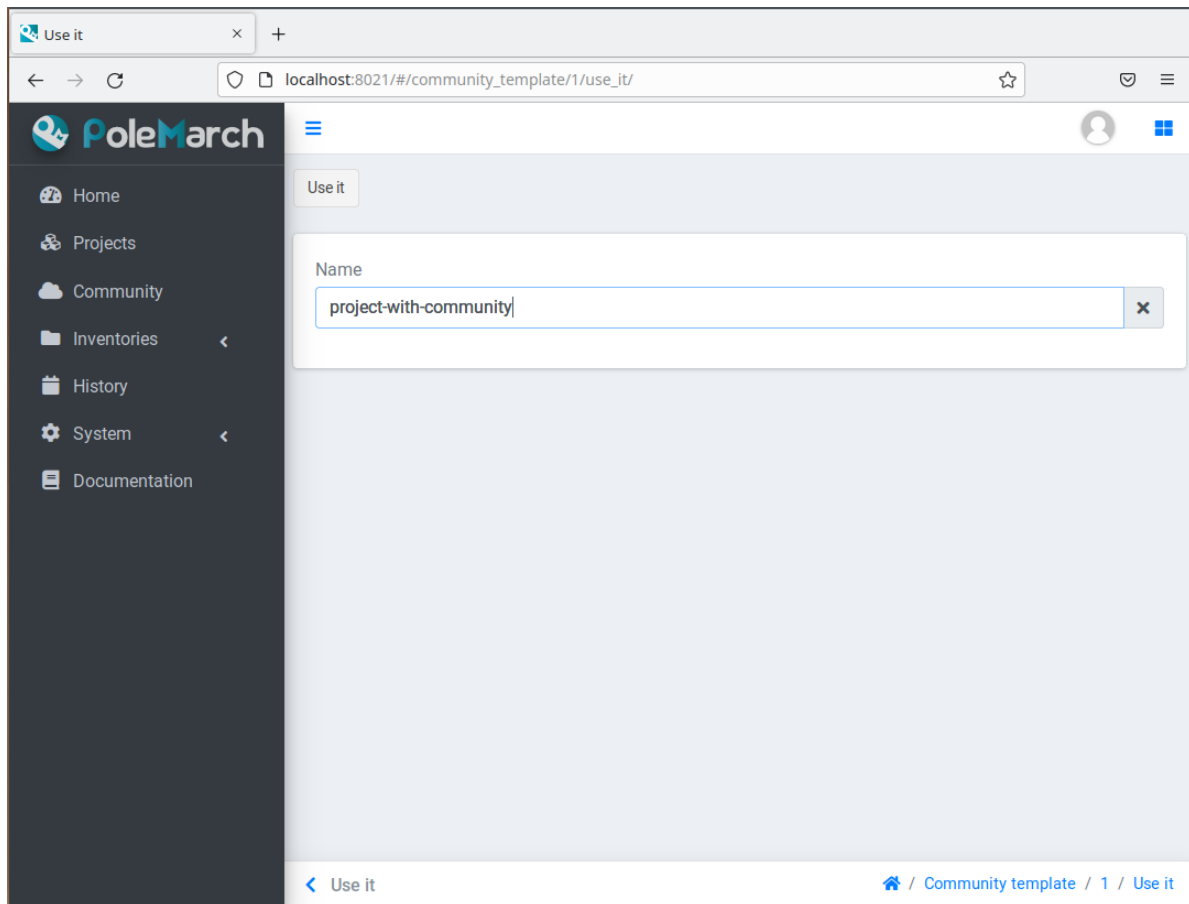
As you can see, the community repository page consists of following fields:

- **Id** - id of community repository;
- **Name** - name of community repository;
- **Description** - description of community repository;
- **Type** - type of community repository;
- **Repository** - URL of community repository.

Also there is only one action button on this page:

- *Use it* - this button opens page where you can set project's name.

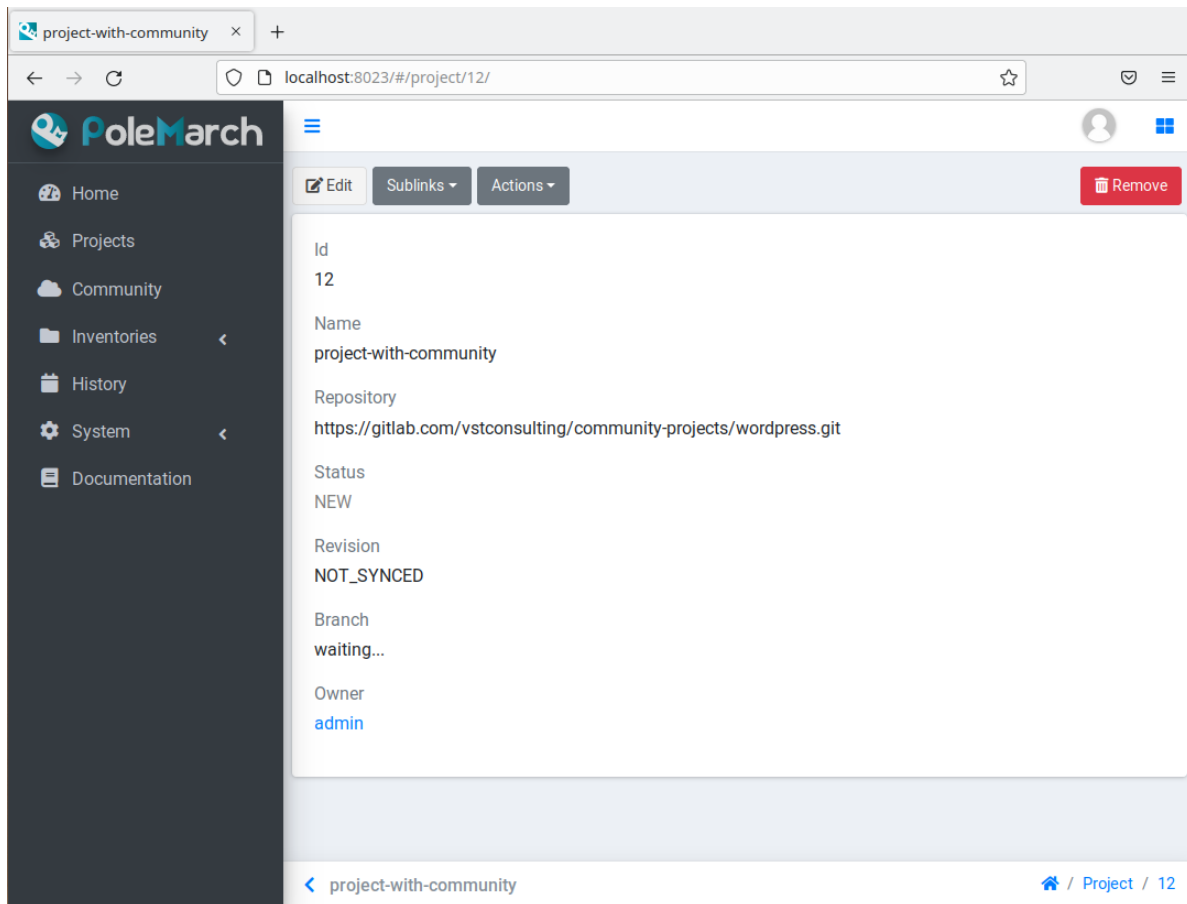
Let's click the *Use it* button.



As you can see, this action page has only one field:

- **Name** - name of project, that will be clone of current community repository in your Polemarch. By default it will be name of community repository with UUID.

After that you need to click *Use it* button once more time and you will be redirected to the page of just created project:



As you can see value of field “Repository” is the same as in “WordPress” community repository. Then you need to click the *Sync* button. After project synchronization you will see the next page:

The screenshot shows the Polemarch web interface in a browser window. The left sidebar contains navigation links: Home, Projects, Community, Inventories, History, System, and Documentation. The main content area is titled 'Quick playbook execution form' and includes several input fields: 'Inventory' (a dropdown menu), 'User' (a text input), 'Private key' (a text input with a file upload icon), and 'MySQL user' (a text input). A 'Deploy' button is visible at the bottom of the form. On the right, a 'Main' sidebar displays project details for 'project-with-community': Id 12, Name project-with-community, Repository https://gitlab.com/vstconsulting/community-projects/wordpress.git, Status OK, Revision 0d4868b95a6666a152d6fb434228d10d83af2aa1, Branch master, and Owner admin.

The screenshot shows the Polemarch web interface in a browser window. The left sidebar contains navigation links: Home, Projects, Community, Inventories, History, System, and Documentation. The main content area is titled 'Quick wordpress deployment' and includes a 'MySQL password' input field and a 'Deploy wordpress' button. On the right, an 'Info' sidebar displays project details for 'project-with-community': Information, Quick wordpress deployment, Project for quick Wordpress deployment with Polemarch, Supported OS (Ubuntu, Debian, RedHat, CentOS), and Project's playbook (deploy.yml: This playbook deploys latest Wordpress version on all hosts from selected inventory).

This page has *Deploy wordpress* button (because project has `.polemarch.yaml` file) and all that you need to do for WordPress deployment on your hosts is filling following fields:

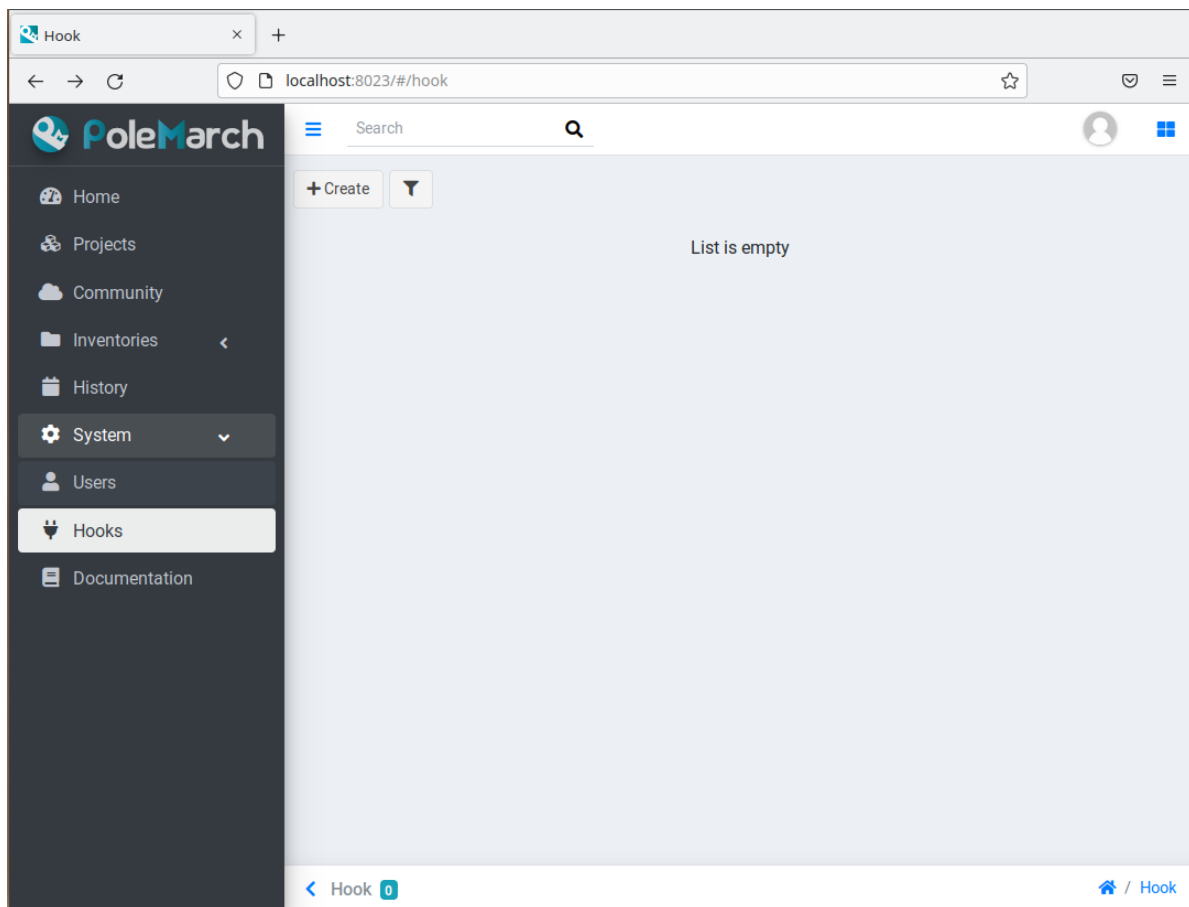
- **Inventory** - it can be inventory from Polemarch system, list of hosts, that are separated by `,`, or path to your inventory in project folder.
- **User** - set it if your do not have `ansible_user` variable in your inventory.
- **Key file** - set it if your do not have `ansible_ssh_private_key_file` variable in your inventory.
- **Mysql user** - name of MySQL user of your wordpress site's database.
- **Mysql password** - password of MySQL user of your wordpress site's database.

After all required fields have been filled, you need to click the *Deploy wordpress* button to start wordpress deployment on your hosts.

3.12 Hooks

Polemarch has his own system of hooks. Polemarch hooks are synchronous and you can appoint them on different events like `"on_execution"`, `"after_execution"` and so on.

Warning: You should be accurate with hooks appointment, because the more hooks you have, the more time they need for execution and, finally, the more time Ansible needs for task execution.

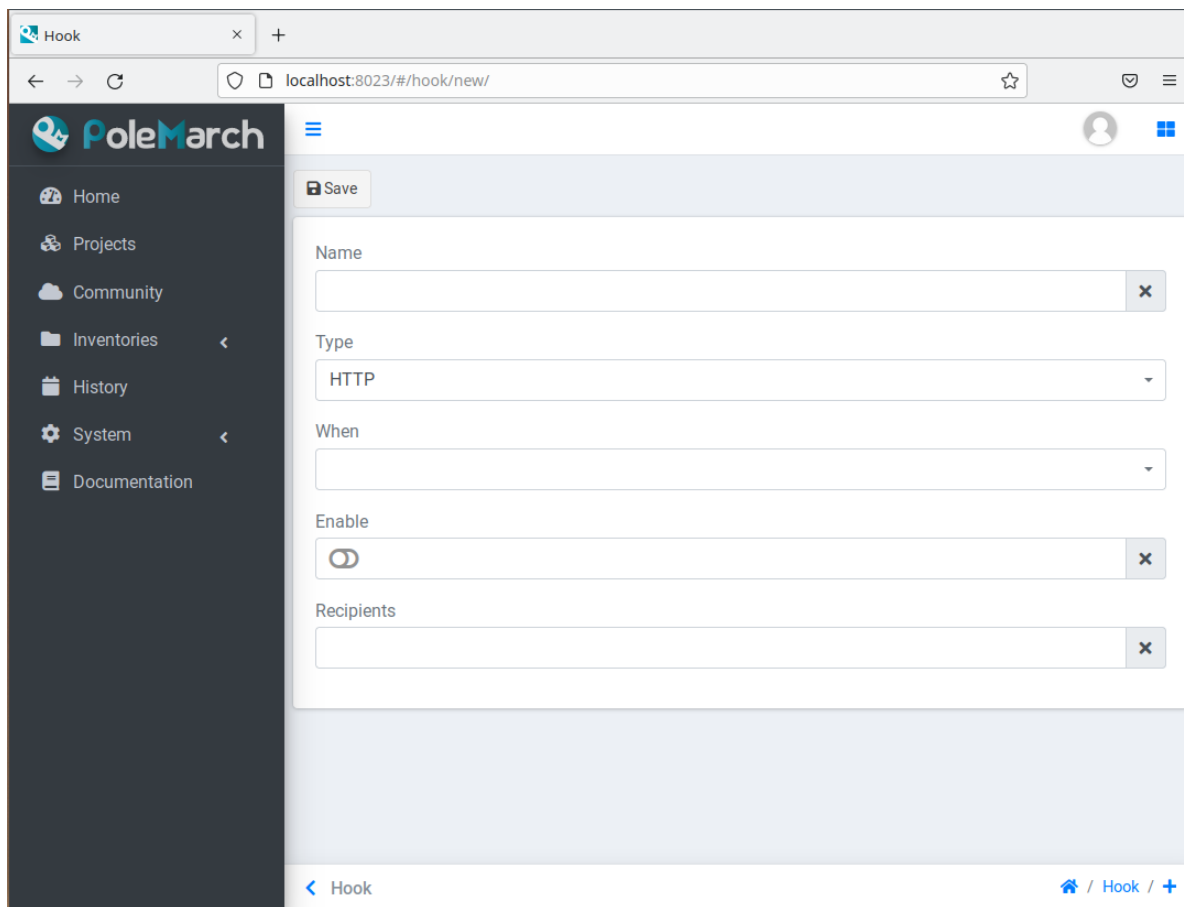


As you can see, there are no hooks in the system now.

There is only one button here:

- *Create* - this button opens the page for creation of new hook.

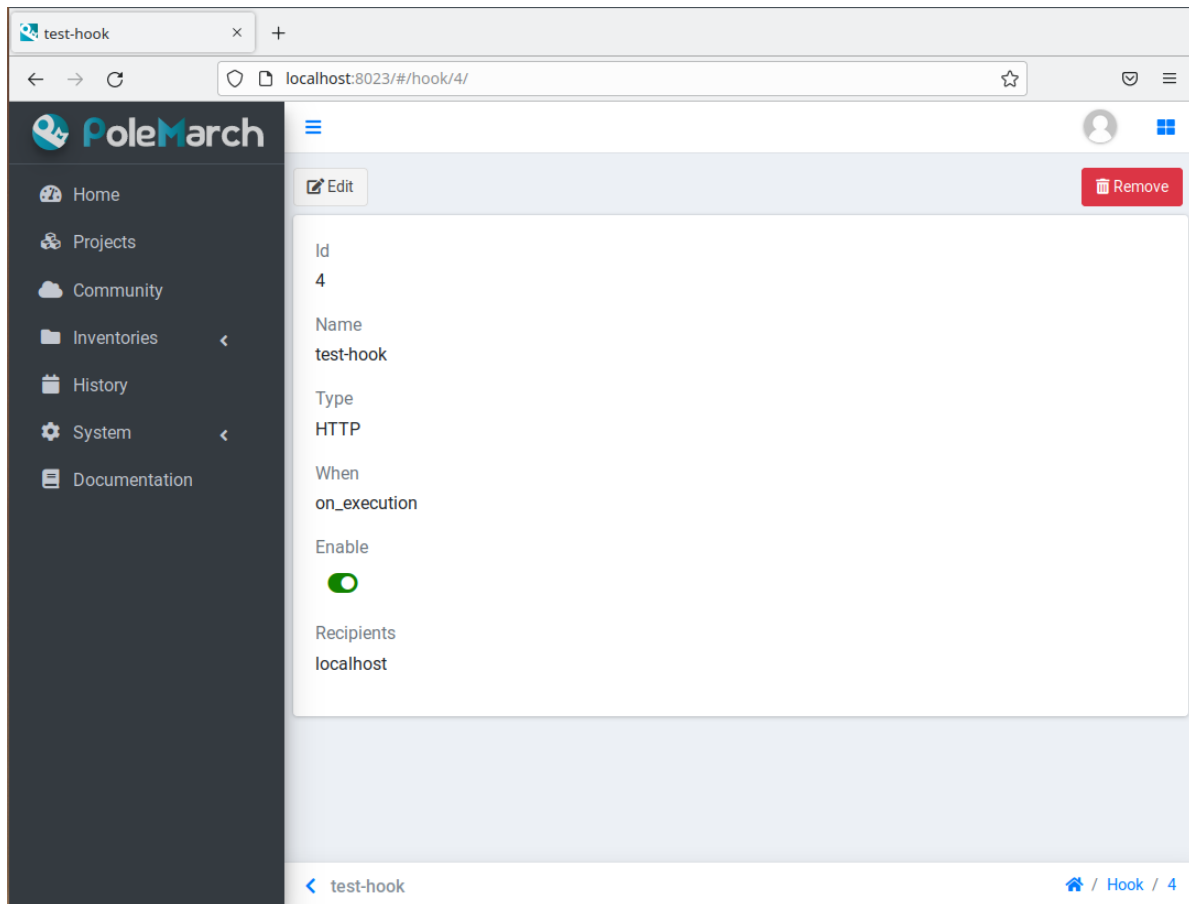
To create hook click the *Create* button.



The screenshot shows a web browser window with the URL `localhost:8023/#/hook/new/`. The application has a dark sidebar with the Polemarch logo and navigation links: Home, Projects, Community, Inventories, History, System, and Documentation. The main content area is titled 'Hook' and contains a 'Save' button at the top left. The form fields are: 'Name' (text input), 'Type' (dropdown menu with 'HTTP' selected), 'When' (dropdown menu), 'Enable' (toggle switch), and 'Recipients' (text input). Each input field has a small 'x' button to clear its content. At the bottom of the form area, there is a breadcrumb trail: `< Hook` and a home link `/ Hook / +`.

As you can see, the form of new hook creation consists of following fields:

- **Name** - name of your hook.
- **Type** - type of hook (HTTP, SCRIPT). * If type is “HTTP”, Polemarch will send “POST” request with JSON to all recipients. * If type is “SCRIPT”, Polemarch will execute script.
- **When** - event on each Polemarch have to execute hook.
- **Enable** - boolean field, it means to activate or to deactivate hook.
- **Recipients** - if type of hook is HTTP, then this field represents list of recipients, separated by ” | “. For example, “ex1.com | ex2.com | ex3.com”. If type of hook is SCRIPT, then recipients are actually list of files containing scripts, similarly separated by ” | “. For example: “notify.sh | counter.sh”. Files must be in `HOOKS_DIR` directory.



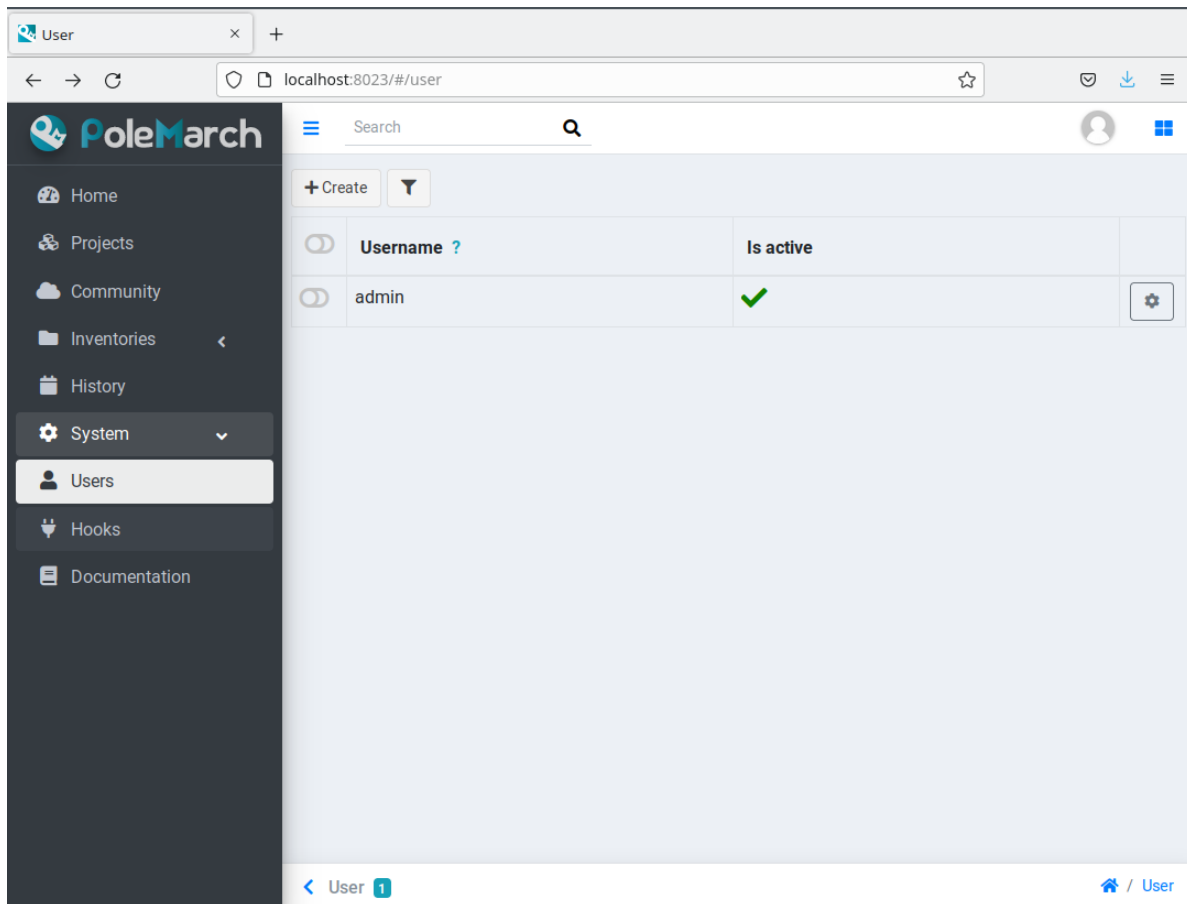
3.13 Users

Polemarch provides you with several types of user:

- superuser;
- staff.

If you need to create a superuser, you need to do it with terminal command. Look for more information here [“Create superuser”](#).

If you need to create user with “staff” rights you can do it with Polemarch GUI:



To create new user click the *Create* button.

The screenshot displays the Polemarch web application's 'New User' creation interface. The browser window shows the URL 'localhost:8023/#/user/new/'. The application has a dark sidebar on the left with navigation links: Home, Projects, Community, Inventories, History, System, and Documentation. The main content area features a form with the following fields and controls:

- Username ?**: A text input field with a clear button (X).
- Is active**: A toggle switch currently set to 'On' (green), with a clear button (X).
- First name**: A text input field with a clear button (X).
- Last name**: A text input field with a clear button (X).
- Email**: A text input field with a clear button (X).
- Password**: A text input field with toggle visibility (eye icon), copy, and clear (X) buttons.
- Repeat password**: A text input field with toggle visibility (eye icon), copy, and clear (X) buttons.

At the top of the form, there are two buttons: 'Save' and 'Generate password'.

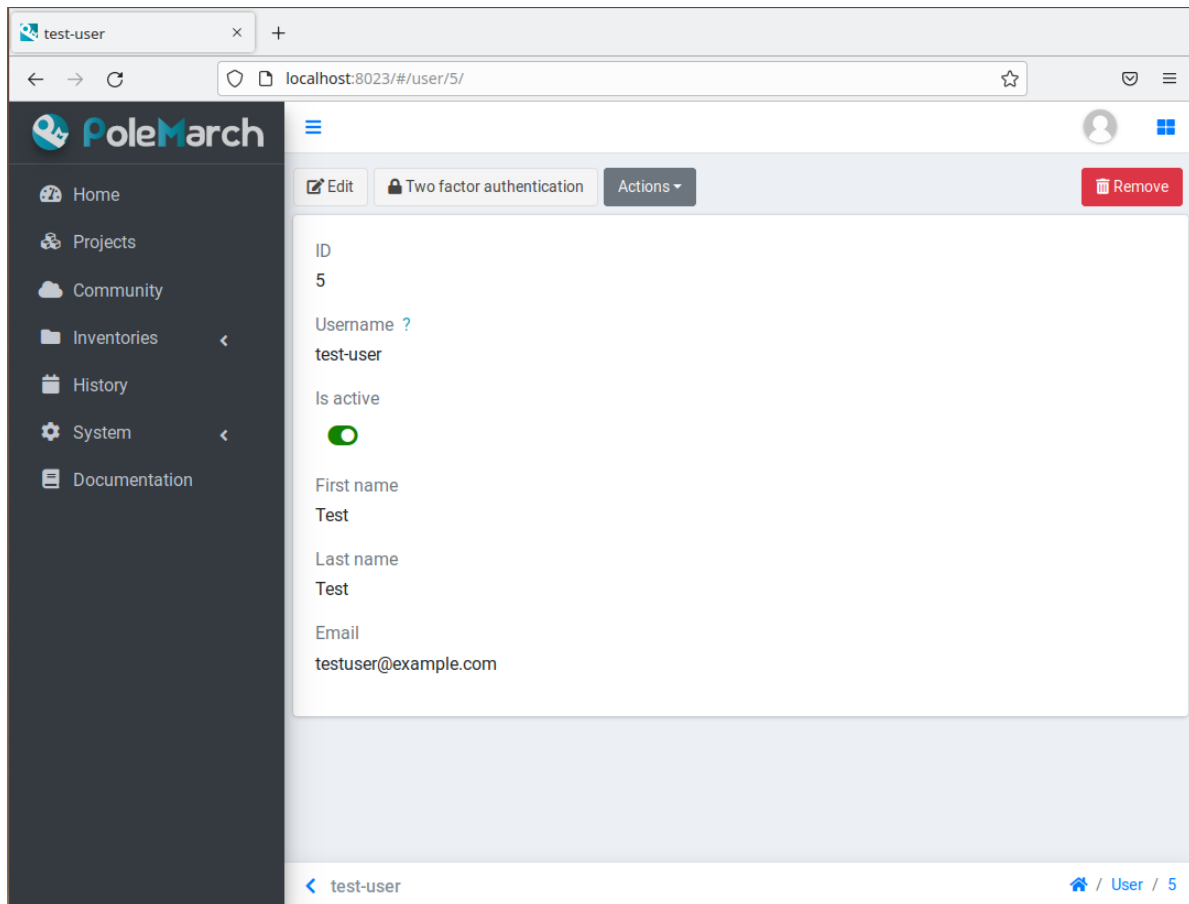
As you can see, the form of new user creation consists of following fields:

- **Username** - name of new user.
- **Is active** - boolean field, it means to activate or to deactivate user.
- **First name** - first name of user.
- **Last name** - last name of user.
- **Email** - email of user.
- **Password** - password of user.
- **Repeat password** - password of user.

Also you can use action *Generate password*.

- *Generate password* - generates random password from 12 to 20 characters long. Password will contain capital and lower letters, numbers and special symbols.

After user creation you will see next page:



As you can see there is only one new fields on this page:

- **Id** - the unique key of object in database.

Also there are several buttons here:

- *Edit* - this button turns on edit mod for this page.
- *Two factor authentication* - you can use TFA if you want to protect your account.
- *Remove* - this button deletes periodic task.

Action buttons:

- *Change password* - this button opens “change password” action page.
- *Copy* - this button opens “copy” action page.

Let's look on “change password” action page.

The screenshot shows a web browser window with the address bar displaying `localhost:8023/#/user/5/change_password/`. The page features a dark sidebar on the left with the Polemarch logo and navigation links: Home, Projects, Community, Inventories, History, System, and Documentation. The main content area has a light blue header with a hamburger menu, a user profile icon, and a blue flag icon. Below the header, there are two buttons: 'Change password' (active) and 'Generate password'. The form contains three input fields: 'Old password', 'New password', and 'Confirm new password'. Each field has a toggle icon (eye) and a copy icon (document) to its right. At the bottom of the page, there is a breadcrumb trail: `< Change password` and a navigation path: `Home / User / 5 / Change password`.

As you can see, the form of “change password” action consists of following fields:

- **Old password** - current password of user.
- **New password** - new password of user.
- **Confirm new password** - new password of user.

There are two buttons here:

- *Change password* - this button saves your new password.
- *Generate password* - generates random password from 12 to 20 characters long. Password will contain capital and lower letters, numbers and special symbols.

PLUGINS

Polemarch can be extended with different type of plugins. This manual describes how to create, setup and start using your first Execution or Inventory plugin and provides full API reference to learn more about them.

4.1 Execution plugins

Execution plugins system allows you to execute any external command from Polemarch. To create an execution plugin you need to

- create a python class describing the argument processing and API talking logic;
- configure this plugin in your `settings.ini`.

Warning: Since v3.0.0 path to execution plugins has been changed from `polemarch.plugins` to `polemarch.plugins.execution`.

4.1.1 Execution plugins: quick start

All built-in execution backends, such as ansible playbook or module execution, use plugins (starting from 2.2.0).

To get started let's create a simple plugin allows you to run echo command.

```
from rest_framework.fields import BooleanField, empty
from vstutils.api.fields import VSTCharField
from polemarch.plugins.execution.base import BasePlugin

# Any plugin must be inherited from BasePlugin
class TestEcho(BasePlugin):
    # Define fields which will be used to execute this plugin and create template with
    # serializer_fields = {
    #     # You can use fields either from rest_framework or vstutils
    #     'string': VSTCharField(),
    #     'n': BooleanField(default=empty, required=False, label='No trailing newlines
    ↪'),
    #     'e': BooleanField(default=empty, required=False, label='Interpret backslash_
    ↪escapes'),
    # }
    # Value of this field will be shown on history detail page as Mode
    arg_shown_on_history_as_mode = 'string'

    # This is our binary from which any command starts
    @property
```

(continues on next page)

(continued from previous page)

```

def base_command(self):
    return ['echo']

    # This method is called by get_args method of BasePlugin for each argument
    # received from API. As we defined
    # 'string', 'n', and 'e' arguments in serializer_fields, we can get their keys and
    # values here.
    def _process_arg(self, key, value):
        # As 'string' argument in this case is a positional argument, let's return just
        # its value, so the final
        # command will be something like ['echo', 'string to output', ...]
        if key == 'string':
            return value
        # As this guys are just boolean flags, let's return them as '-n' or '-e'
        # accordingly
        if key in ('n', 'e') and value:
            return f'-{key}'
        # Note, that if we received for example 'n' argument with False value, we are
        # returning None,
        # means that it won't be included to execution command. But of course, you
        # may override this behavior
        # in get_args method.

```

Supposing that described plugin is located at `polemarch.plugins.execution.custom.Echo`, let's connect it to Polemarch. In your `/etc/polemarch/settings.ini` add following section:

```

[execution.plugin.echo]
backend = polemarch.plugins.execution.custom.Echo

```

Also you may want to provide additional options directly to plugin:

```

[execution.plugin.echo.options]
some_option = 'some_option'

```

In this example `some_option` will be available in any plugin's instance method as `self.config['some_option']`, as all options are initialized in the `__init__` method.

So it's all done! After restarting your polemarch server and resetting schema, you can check `#/project/<your_project_id>/execute_echo/`. Here you should be able to execute echo plugin as any built-in one. Also you should be able to create a *template* with it at `#/project/<your_project_id>/execution_templates/new/`.

If you tried executing echo plugin with flags, you may see that this flags are being outputted too. This is because they goes after string argument. To fix this issue, we may do something like this:

```

...

class TestEcho(BasePlugin):
    ...

    def get_args(self, raw_args):
        # We know that 'string' is required so no default for .pop() is needed
        string_value = raw_args.pop('string')
        args = super().get_args(raw_args)
        # Push our string to the end of command
        args += [string_value]
        return args

```

(continues on next page)

(continued from previous page)

```

@property
def base_command(self):
    return ['echo']

def _process_arg(self, key, value):
    if key in ('n', 'e') and value:
        return f'-{key}'

```

Now if you are passing flags to execution, they should work the same except not being outputted.

Note: If your execution plugin may work with inventories, you should specify which inventory plugins are compatible with your execution plugin. By default it's assumed that execution plugin can't work with inventories. For more information about inventory plugins please see *Inventory plugins*.

To learn more about what plugins are provide, please check API reference.

4.2 Inventory plugins

Inventory plugins system allows you to define how inventory stores, manages and displays its state. To create an inventory plugin you need to

- create a python class which manages inventory state and API talking logic;
- configure this plugin in your `settings.ini`.

4.2.1 Inventory plugins: quick start

To get started let's create a simple plugin which works like a simplified version of built-in `AnsibleString` with some additional changes.

```

from vstutils.api import fields as vstfields
from polemarch.plugins.inventory.base import BasePlugin

class CustomAnsibleString(BasePlugin):
    # Our plugin will support import and we will implement corresponding method
    supports_import = True

    # These fields will be used for working with inventory state in API
    serializer_fields = {
        # You can use fields either from rest_framework or vstutils
        'body': vstfields.TextareaField(allow_blank=True, default=''),
    }
    # Default values for our fields which will be used to initialize inventory state
    defaults = {
        'body': 'localhost ansible_connection=local',
    }
    # These fields will be used for import action in API
    serializer_import_fields = {
        'body': vstfields.FileInStringField(),
    }

    def render_inventory(self, execution_dir):
        # Getting inventory state. This is possible because by default state_managed_
        ↪ attribute of plugin class

```

(continues on next page)

(continued from previous page)

```

    # is True
    state_data = self.instance.inventory_state.data
    filename = str(uuid1())
    # Any created files must be in execution_dir
    filepath = Path(execution_dir) / filename
    filepath.write_text(state_data['body'])
    # We doesn't need any additional files so the second argument is empty list
    return filepath, []

def get_raw_inventory(self, inventory_string):
    # This string will be shown on history page
    return f'File contents:\n{inventory_string}'

@classmethod
def import_inventory(cls, instance, data):
    # Here we got data which structure corresponds to serializer_import_fields
    # and created inventory instance. It's recommended to always use update_
    ↪inventory_state method
    # rather than accessing inventory state directly.
    instance.update_inventory_state(data=data)
    return instance

```

Supposing that described plugin is located at `polemarch.plugins.inventory.custom.CustomInventoryString`, let's connect it to Polemarch. In your `/etc/polemarch/settings.ini` add following section:

```

[inventory.plugin.custom_inventory_string]
backend = polemarch.plugins.inventory.custom.CustomInventoryString

```

Also you may want to provide additional options directly to plugin:

```

[inventory.plugin.custom_inventory_string.options]
some_option = 'some_option'

```

In this example `some_option` will be available in any plugin's instance method as `self.options['some_option']`, as all options are initialized in the `__init__` method.

To start working with the created plugin we also need to allow some execution plugin work with this one. Let's say that `ANSIBLE_MODULE` plugin can play with our new `CUSTOM_ANSIBLE_STRING`:

```

[execution.plugin.ansible_module.options]
; Make sure you not disabled other built-in inventory plugins
compatible_inventory_plugins = polemarch_db,ansible_file,ansible_string,custom_
    ↪inventory_string

```

Done! Now you can create inventory with your plugin at `#/project/<your_project_id>/inventory/new/`, execute `ANSIBLE_MODULE` plugin at `#/project/<your_project_id>/execute_ansible_module/` selecting created inventory.

4.3 API reference

class `polemarch.plugins.execution.base.BasePlugin(options=None, output_handler=None)`

Base execution plugin class from which any other plugin should inherit. The plugin itself is an entity which, on the one hand provides appropriate fields for the API, and on the other hand, processes arguments received from it to build execution command.

For each configured plugin an endpoint will be generated allows you to choose arguments and execute it. Also, this plugin will be available to create template with.

Parameters

- **options** (`dict`) – `settings.ini` options mapping for this plugin.
- **output_handler** (`typing.Optional[typing.Callable]`) – executor's function which handles logging and outputting to history. Used by `verbose_output` method.

_get_serializer_fields(`exclude_fields=()`)

Returns field name and field instance mapping used to generate fields for serializer.

Parameters

exclude_fields (`tuple`) – field names that should not be presented in serializer.

Return type

`typing.Mapping[str, rest_framework.fields.Field]`

_get_serializer_metaclass(`exclude_fields=()`)

Returns serializer metaclass used to generate fields in serializer.

Parameters

exclude_fields (`tuple`) – field names that should not be presented in serializer.

Return type

`typing.Type[typing.Type[vstutils.api.serializers.BaseSerializer]]`

_process_arg(`key, value`)

Returns single argument with value for `get_args` method. Should return `None` if argument must not be included to the execution command.

Parameters

- **key** (`str`) – argument key (e.g. `verbose`).
- **value** (`typing.Any`) – argument value (e.g. `2`).

Return type

`typing.Optional[str]`

arg_shown_on_history_as_inventory: `typing.Optional[str] = None`

Name of argument presented in generated serializer which will be shown on list history page as *Inventory*.

arg_shown_on_history_as_mode: `typing.Optional[str] = None`

Name of argument presented in generated serializer which will be shown on detail history page as *Mode*. For example, if you are executing some module with additional arguments and fields contains *module* field, than you can set `'module'` here, and it's value will be shown after execution. If not set, *Mode* in the history will show `[<plugin name> plugin]` string.

base_command: `typing.List[str]`

Base command (usually binary) from which execution command starts, e.g. `['echo']`. You may also override this attribute as a property if more complex logic needs to be performed.

error_codes: `typing.Mapping[int, str] = {}`

This mapping will be looked up to choose an appropriate error message for history output if execution finished with errors. If no code found, then just "ERROR" string outputs.

get_args(*raw_args*)

Returns list of processed arguments which will be substituted into execution command.

Parameters

raw_args (*dict*) – argument name-value mapping which should be processed.

Return type

typing.List[str]

get_env_vars(*project_data*)

Returns env variables which will be used in execution, project's env variables by default.

Parameters

project_data – proxy of the project instance, allows you to access it's readonly properties, such as `config vars`, `env_vars` etc.

Return type

typing.Mapping[str, str]

get_execution_data(*execution_dir*, *raw_args*, *project_data*)

Returns tuple of execution command and env variables. This method will be called directly by executor before execution starts.

Parameters

- **execution_dir** (*pathlib.Path*) – path to execution directory in which project copy located. All additional files that should be generated (e.g. inventory file) must be placed here.
- **raw_args** (*dict*) – argument name-value mapping which should be processed.
- **project_data** – proxy of the project instance, allows you to access it's readonly properties, such as `config vars`, `env_vars` etc.

Return type

typing.Tuple[typing.List[str], dict]

get_pre_commands(*raw_args*)

This method will be called before execution. Returns list of commands which are needed to be executed before main execution command.

Parameters

raw_args (*dict*) – dictionary with arguments received from API.

Return type

typing.List[typing.List[str]]

get_raw_inventory()

Returns raw inventory string used to show it on history page.

Return type

str

get_serializer_class(*exclude_fields=()*)

Returns serializer class which will be used to generate fields for arguments. Uses metaclass returned by `_get_serializer_metaclass` method.

Parameters

exclude_fields (*tuple*) – field names that should not be presented in serializer.

Return type

typing.Type[vstutils.api.serializers.BaseSerializer]

get_verbose_level(*raw_args*)

Returns verbose level used for history output and logging. Should be taken from execution arguments (usually from `verbose` argument). This method will be called directly by executor.

Parameters

raw_args (`dict`) – argument name-value mapping which should be processed.

Return type

`int`

property name: `str`

Returns name of plugin, class name by default. Primarily used to generate an appropriate model name for OpenAPI schema.

post_execute_hook(*cmd*, *raw_args*)

This method will be called after execution.

Parameters

- **cmd** (`typing.List[str]`) – list of arguments which were used for execution.
- **raw_args** (`dict`) – dictionary with arguments received from API.

Return type

`None`

prepare_execution_dir(*dir*)

Gets execution directory with copied project. All files needed for execution (e.g. generated inventory file) should be here.

Parameters

dir (`pathlib.Path`) – path to execution directory in which project copy located. All additional files that should be generated (e.g. inventory file) must be placed here.

Return type

`None`

serializer_fields: `typing.Mapping[str, rest_framework.fields.Field] = {}`

Fields mapping used to generate serializer. By default returned by `_get_serializer_fields` method.

verbose_output(*message*, *level=3*)

Logs value with logger and outputs it to history.

Parameters

- **message** (`str`) – message to output.
- **level** (`int`) – verbosity level from which message should be outputted.

Return type

`None`

class `polemarch.plugins.inventory.base.BasePlugin`(*options*)

Base inventory plugin class from which any other plugin should inherit. The plugin itself is an entity which, on the one hand provides appropriate fields for the API, and on the other hand, manages inventory state.

Parameters

options – `settings.ini` options for this plugin.

_get_serializer_fields()

Returns field name and field instance mapping used to generate fields for serializer used for working with state in API.

Return type

`typing.Mapping[str, rest_framework.fields.Field]`

_get_serializer_import_fields()

Returns field name and field instance mapping used to generate fields for serializer used for import action.

Return type`typing.Mapping[str, rest_framework.fields.Field]`**`_get_serializer_import_metaclass()`**

Returns serializer metaclass used to generate fields in serializer for import action.

Return type`typing.Type[typing.Type[vstutils.api.serializers.BaseSerializer]]`**`_get_serializer_metaclass()`**

Returns serializer metaclass used to generate fields in serializer for working with state in API.

Return type`typing.Type[typing.Type[vstutils.api.serializers.BaseSerializer]]`**`defaults: typing.Mapping[str, typing.Any] = {}`**

Field name and its default value mapping used to initialize inventory state. Only works if plugin is `state_managed`.

`classmethod get_raw_inventory(inventory_string)`

Returns raw inventory string used to show it on history page.

Parameters

`inventory_string` (`str`) – inventory string which is needed to be processed.

Return type`str`**`get_serializer_class()`**

Returns serializer class which will be used for working with state in API. Uses metaclass returned by `_get_serializer_metaclass` method.

Return type`typing.Type[vstutils.api.serializers.BaseSerializer]`**`get_serializer_import_class()`**

Returns serializer class which will be used for import action. Uses metaclass returned by `_get_serializer_metaclass` method.

Return type`typing.Type[vstutils.api.serializers.BaseSerializer]`**`classmethod import_inventory(instance, data)`**

Method which implements importing inventory from external source. Must be implemented if `supports_import` is `True`.

Parameters

- **`data`** (`dict`) – data received from user in API as a result of import action.
- **`instance`** (`polemarch.main.models.hosts.Inventory`) – created `Inventory` instance.

`render_inventory(instance, execution_dir)`

Renders inventory into text file and puts it into `execution_dir` directory. Additional files may be returned by second argument as list (or empty list, if no any).

Parameters

- **`instance`** (`polemarch.main.models.hosts.Inventory`) – `Inventory` instance.
- **`execution_dir`** (`pathlib.Path`) – path

Return type`typing.Tuple[pathlib.Path, list]`

serializer_fields: `typing.Mapping[str, rest_framework.fields.Field] = {}`

Fields mapping used to generate serializer for working with state in API. By default returned by `_get_serializer_fields` method.

serializer_import_fields: `typing.Mapping[str, rest_framework.fields.Field] = {}`

Fields mapping used to generate serializer for import action. By default returned by `_get_serializer_import_fields` method.

state_managed: `bool = True`

Flag indicating whether Inventory stores all its data in json field or in separate models.

supports_import: `bool = False`

Flag indicating whether Inventory supports import from external source or not. If True `import_inventory` method must be implemented.

Polemarch provides REST API for all its functionality accessible via web GUI, because our GUI also uses this API to work. Below there is an information about every entity we have in Polemarch and methods applicable to it. This is an example of api schema, but you can find actual schema on Polemarch host at `/api/v3/openapi/`.

5.1 Structure

In Polemarch we have some entities that can be nested to another entities. Below examples of such entities:

Inventory can be nested into **Project**

Group can be nested into **Inventory** or into another **Group** with `children=true`

Host can be nested into **Inventory** or into another **Group** with `children=false`

User can be nested into **Team**

To add entities into another, you only need send `[{"id": [instance_id]}, ...]` to subpath. Also you can insert instead of data results of bulk request, inner mechanism add all entities in result to parent entity.

GET /community_template/

List of community project templates.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **type** (*string*) –
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- **200 OK** – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –

- **results[]**.**id** (*integer*) – (required)
- **results[]**.**name** (*string*) – (required)
- **results[]**.**type** (*string*) –

GET /community_template/{id}/

Return a community project template instance.

Parameters

- **id** (*integer*) – A unique value identifying this project community template.

Status Codes

- 200 OK – OK

Response JSON Object

- **description** (*string*) – (required)
- **id** (*integer*) – (required)
- **name** (*string*) – (required)
- **repository** (*string*) – (required)
- **type** (*string*) –

POST /community_template/{id}/use_it/

Create a project based on current template.

Parameters

- **id** (*integer*) – A unique value identifying this project community template.

Request JSON Object

- **name** (*string*) –
- **project_id** (*integer*) – (read only)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **name** (*string*) –
- **project_id** (*integer*) – (read only)

GET /group/

Return all groups.

Query Parameters

- **__deep_parent** (*integer*) –
- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.

- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]**.**children** (*boolean*) –
- **results[]**.**id** (*integer*) – (read only)
- **results[]**.**name** (*string*) –

POST /group/

Create a new group.

Request JSON Object

- **children** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **children** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

GET /group/{id}/

Return a group instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.

Status Codes

- 200 OK – OK

Response JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

PUT /group/{id}/

Update a group.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.

Request JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

Status Codes

- 200 OK – OK

Response JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

PATCH /group/{id}/

Update one or more fields on an existing group.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.

Request JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

Status Codes

- 200 OK – OK

Response JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

DELETE /group/{id}/

Remove an existing group.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.

Status Codes

- 204 No Content – NO CONTENT

POST /group/{id}/copy/

Endpoint which copy instance with deps.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

GET /group/{id}/groups/

Return all groups.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.

Query Parameters

- **__deep_parent** (*integer*) –
- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]**.children (*boolean*) –
- **results[]**.id (*integer*) – (read only)
- **results[]**.name (*string*) –

POST /group/{id}/groups/

Create a new group.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.

Request JSON Object

- **children** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **children** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

GET /group/{id}/groups/{groups_id}/

Return a group instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Status Codes

- 200 OK – OK

Response JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

PUT /group/{id}/groups/{groups_id}/

Update a group.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Request JSON Object

- **children** (*boolean*) – (read only)

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

Status Codes

- 200 OK – OK

Response JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

PATCH /group/{id}/groups/{groups_id}/

Update one or more fields on an existing group.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Request JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

Status Codes

- 200 OK – OK

Response JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

DELETE /group/{id}/groups/{groups_id}/

Remove an existing group.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Status Codes

- 204 No Content – NO CONTENT

POST /group/{id}/groups/{groups_id}/copy/

Endpoint which copy instance with deps.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

GET /group/{id}/groups/{groups_id}/hosts/

Return all hosts.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **type** (*string*) – Instance type.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**name** (*string*) –

- **results[]**.**type** (*string*) –

POST /group/{id}/groups/{groups_id}/hosts/

Create a new host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 201 **Created** – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

GET /group/{id}/groups/{groups_id}/hosts/{hosts_id}/

Return a host instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Status Codes

- 200 **OK** – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

PUT /group/{id}/groups/{groups_id}/hosts/{hosts_id}/

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.

- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

PATCH /group/{id}/groups/{groups_id}/hosts/{hosts_id}/

Update one or more fields on an existing host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

DELETE /group/{id}/groups/{groups_id}/hosts/{hosts_id}/

Remove an existing host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Status Codes

- 204 **No Content** – NO CONTENT

POST /group/{id}/groups/{groups_id}/hosts/{hosts_id}/copy/

Endpoint which copy instance with deps.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

Status Codes

- 201 **Created** – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

PATCH /group/{id}/groups/{groups_id}/hosts/{hosts_id}/set_owner/

Change instance owner.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **owner** (*integer*) – (required)

Status Codes

- 200 **OK** – OK

Response JSON Object

- **owner** (*integer*) – (required)

GET /group/{id}/groups/{groups_id}/hosts/{hosts_id}/variables/

Return all variables of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.

- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **key** (*string*) – A key name string value (or comma separated list) of instance.
- **value** (*string*) – A value of instance.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**key** (*string*) – (required)
- **results[]** .**value** (*string*) – (required)

POST /group/{id}/groups/{groups_id}/hosts/{hosts_id}/variables/

Create a new variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

GET /group/{id}/groups/{groups_id}/hosts/{hosts_id}/variables/{variables_id}/

Return a variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PUT /group/{id}/groups/{groups_id}/hosts/{hosts_id}/variables/{variables_id}/

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PATCH /group/{id}/groups/{groups_id}/hosts/{hosts_id}/variables/{variables_id}/

Update one or more fields on an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

DELETE /group/{id}/groups/{groups_id}/hosts/{hosts_id}/variables/{variables_id}/

Remove an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 204 No Content – NO CONTENT

PATCH /group/{id}/groups/{groups_id}/set_owner/

Change instance owner.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.

Request JSON Object

- **owner** (*integer*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **owner** (*integer*) – (required)

GET /group/{id}/groups/{groups_id}/variables/

Return all variables of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **key** (*string*) – A key name string value (or comma separated list) of instance.
- **value** (*string*) – A value of instance.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]**.**id** (*integer*) – (read only)
- **results[]**.**key** (*string*) – (required)
- **results[]**.**value** (*string*) – (required)

POST /group/{id}/groups/{groups_id}/variables/

Create a new variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

GET /group/{id}/groups/{groups_id}/variables/{variables_id}/

Return a variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.

- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PUT /group/{id}/groups/{groups_id}/variables/{variables_id}/

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PATCH /group/{id}/groups/{groups_id}/variables/{variables_id}/

Update one or more fields on an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

DELETE /group/{id}/groups/{groups_id}/variables/{variables_id}/

Remove an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 204 No Content – NO CONTENT

GET /group/{id}/hosts/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **type** (*string*) – Instance type.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.

- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**name** (*string*) –
- **results[]** .**type** (*string*) –

POST /group/{id}/hosts/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

- **type** (*string*) –

GET /group/{id}/hosts/{hosts_id}/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

PUT /group/{id}/hosts/{hosts_id}/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

PATCH /group/{id}/hosts/{hosts_id}/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

DELETE /group/{id}/hosts/{hosts_id}/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Status Codes

- 204 No Content – NO CONTENT

POST /group/{id}/hosts/{hosts_id}/copy/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

PATCH /group/{id}/hosts/{hosts_id}/set_owner/

Change instance owner.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **owner** (*integer*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **owner** (*integer*) – (required)

GET /group/{id}/hosts/{hosts_id}/variables/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.

- **key** (*string*) – A key name string value (or comma separated list) of instance.
- **value** (*string*) – A value of instance.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**key** (*string*) – (required)
- **results[]** .**value** (*string*) – (required)

POST /group/{id}/hosts/{hosts_id}/variables/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

GET /group/{id}/hosts/{hosts_id}/variables/{variables_id}/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PUT /group/{id}/hosts/{hosts_id}/variables/{variables_id}/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PATCH /group/{id}/hosts/{hosts_id}/variables/{variables_id}/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

DELETE /group/{id}/hosts/{hosts_id}/variables/{variables_id}/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 204 No Content – NO CONTENT

PATCH /group/{id}/set_owner/

Change instance owner.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.

Request JSON Object

- **owner** (*integer*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **owner** (*integer*) – (required)

GET /group/{id}/variables/

Inventory hosts variables.

retrieve:

Return a variable of inventory host.

list:

Return all variables of inventory host.

create:

Create a new variable of inventory host.

destroy:

Remove an existing variable.

partial_update:

Update one or more fields on an existing variable.

update:

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **key** (*string*) – A key name string value (or comma separated list) of instance.
- **value** (*string*) – A value of instance.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- **200 OK** – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**key** (*string*) – (required)
- **results[]** .**value** (*string*) – (required)

POST /group/{id}/variables/

Inventory hosts variables.

retrieve:

Return a variable of inventory host.

list:

Return all variables of inventory host.

create:

Create a new variable of inventory host.

destroy:

Remove an existing variable.

partial_update:

Update one or more fields on an existing variable.

update:

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 201 *Created* – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

GET /group/{id}/variables/{variables_id}/

Inventory hosts variables.

retrieve:

Return a variable of inventory host.

list:

Return all variables of inventory host.

create:

Create a new variable of inventory host.

destroy:

Remove an existing variable.

partial_update:

Update one or more fields on an existing variable.

update:

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PUT /group/{id}/variables/{variables_id}/
Inventory hosts variables.

retrieve:

Return a variable of inventory host.

list:

Return all variables of inventory host.

create:

Create a new variable of inventory host.

destroy:

Remove an existing variable.

partial_update:

Update one or more fields on an existing variable.

update:

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PATCH /group/{id}/variables/{variables_id}/
Inventory hosts variables.

retrieve:

Return a variable of inventory host.

list:

Return all variables of inventory host.

create:

Create a new variable of inventory host.

destroy:

Remove an existing variable.

partial_update:

Update one or more fields on an existing variable.

update:

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

DELETE /group/{id}/variables/{variables_id}/

Inventory hosts variables.

retrieve:

Return a variable of inventory host.

list:

Return all variables of inventory host.

create:

Create a new variable of inventory host.

destroy:

Remove an existing variable.

partial_update:

Update one or more fields on an existing variable.

update:

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this group.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 204 No Content – NO CONTENT

GET /history/

Return all history instances of executions.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **status** (*string*) – Status of execution.
- **mode** (*string*) – Module or playbook name.
- **kind** (*string*) – Kind of execution.
- **older** (*string*) – Older then this time
- **newer** (*string*) – Newer then this time
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]**.**executor** (*string*) – (required)
- **results[]**.**id** (*integer*) – (read only)
- **results[]**.**initiator** (*integer*) –
- **results[]**.**initiator_type** (*string*) –
- **results[]**.**inventory** (*string*) – (read only)
- **results[]**.**kind** (*string*) –
- **results[]**.**mode** (*string*) – (required)
- **results[]**.**options** (*string*) – (read only)
- **results[]**.**project** (*integer*) –
- **results[]**.**start_time** (*string*) –
- **results[]**.**status** (*string*) – (required)
- **results[]**.**stop_time** (*string*) –

GET /history/{id}/

Return an execution history instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this history.

Status Codes

- 200 OK – OK

Response JSON Object

- **execute_args** (*string*) – (required)
- **execution_time** (*integer*) – (required)
- **executor** (*string*) – (required)
- **id** (*integer*) – (read only)
- **initiator** (*integer*) –
- **initiator_type** (*string*) –
- **inventory** (*integer*) –
- **kind** (*string*) –
- **mode** (*string*) – (required)
- **options** (*string*) – (read only)
- **project** (*integer*) –
- **raw_args** (*string*) –
- **raw_inventory** (*string*) –
- **raw_stdout** (*string*) – (read only)
- **revision** (*string*) –
- **start_time** (*string*) –
- **status** (*string*) – (required)
- **stop_time** (*string*) –

DELETE /history/{id}/

Remove an existing history record.

Parameters

- **id** (*integer*) – A unique integer value identifying this history.

Status Codes

- 204 No Content – NO CONTENT

PATCH /history/{id}/cancel/

Cancel working task.

Parameters

- **id** (*integer*) – A unique integer value identifying this history.

Status Codes

- 200 OK – OK

Response JSON Object

- **detail** (*string*) – (read only)

DELETE /history/{id}/clear/

Clear execution output.

Parameters

- **id** (*integer*) – A unique integer value identifying this history.

Status Codes

- 204 No Content – NO CONTENT

GET /history/{id}/facts/

Get facts from “setup” ansible module execution.

Parameters

- **id** (*integer*) – A unique integer value identifying this history.

Status Codes

- 200 OK – OK

Response JSON Object

- **facts** (*string*) – (read only)

GET /hook/

Return all hooks.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **when** (*string*) –
- **type** (*string*) – Instance type.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**enable** (*boolean*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**name** (*string*) –
- **results[]** .**recipients** (*string*) – (required)
- **results[]** .**type** (*string*) – (required)
- **results[]** .**when** (*string*) –

POST /hook/

Create a new hook.

Request JSON Object

- **enable** (*boolean*) –
- **id** (*integer*) – (read only)

- **name** (*string*) –
- **recipients** (*string*) – **HTTP**: list of URLs, separated by “|”. **
SCRIPT**: script files, separated by “|” Files must be in HOOKS_DIR directory. (required)
- **type** (*string*) – (required)
- **when** (*string*) – (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **enable** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **recipients** (*string*) – **HTTP**: list of URLs, separated by “|”. **
SCRIPT**: script files, separated by “|” Files must be in HOOKS_DIR directory. (required)
- **type** (*string*) – (required)
- **when** (*string*) – (required)

GET /hook/{id}/

Return a hook instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this hook.

Status Codes

- 200 OK – OK

Response JSON Object

- **enable** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **recipients** (*string*) – **HTTP**: list of URLs, separated by “|”. **
SCRIPT**: script files, separated by “|” Files must be in HOOKS_DIR directory. (required)
- **type** (*string*) – (required)
- **when** (*string*) – (required)

PUT /hook/{id}/

Update a hook.

Parameters

- **id** (*integer*) – A unique integer value identifying this hook.

Request JSON Object

- **enable** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –

- **recipients** (*string*) – **HTTP**: list of URLs, separated by “|”.
**
SCRIPT**: script files, separated by “|” Files must be in HOOKS_DIR directory. (required)
- **type** (*string*) – (required)
- **when** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **enable** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **recipients** (*string*) – **HTTP**: list of URLs, separated by “|”.
**
SCRIPT**: script files, separated by “|” Files must be in HOOKS_DIR directory. (required)
- **type** (*string*) – (required)
- **when** (*string*) – (required)

PATCH /hook/{id}/

Update one or more fields on an existing hook.

Parameters

- **id** (*integer*) – A unique integer value identifying this hook.

Request JSON Object

- **enable** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **recipients** (*string*) – **HTTP**: list of URLs, separated by “|”.
**
SCRIPT**: script files, separated by “|” Files must be in HOOKS_DIR directory. (required)
- **type** (*string*) – (required)
- **when** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **enable** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **recipients** (*string*) – **HTTP**: list of URLs, separated by “|”.
**
SCRIPT**: script files, separated by “|” Files must be in HOOKS_DIR directory. (required)
- **type** (*string*) – (required)
- **when** (*string*) – (required)

DELETE /hook/{id}/

Remove an existing hook.

Parameters

- **id** (*integer*) – A unique integer value identifying this hook.

Status Codes

- 204 No Content – NO CONTENT

GET /host/

Return all hosts.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **type** (*string*) – Instance type.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**name** (*string*) –
- **results[]** .**type** (*string*) –

POST /host/

Create a new host.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

GET /host/{id}/

Return a host instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this host.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

PUT /host/{id}/

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this host.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

PATCH /host/{id}/

Update one or more fields on an existing host.

Parameters

- **id** (*integer*) – A unique integer value identifying this host.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

DELETE /host/{id}/

Remove an existing host.

Parameters

- **id** (*integer*) – A unique integer value identifying this host.

Status Codes

- 204 No Content – NO CONTENT

POST /host/{id}/copy/

Endpoint which copy instance with deps.

Parameters

- **id** (*integer*) – A unique integer value identifying this host.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

PATCH /host/{id}/set_owner/

Change instance owner.

Parameters

- **id** (*integer*) – A unique integer value identifying this host.

Request JSON Object

- **owner** (*integer*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **owner** (*integer*) – (required)

GET /host/{id}/variables/

Inventory hosts variables.

retrieve:

Return a variable of inventory host.

list:

Return all variables of inventory host.

create:

Create a new variable of inventory host.

destroy:

Remove an existing variable.

partial_update:

Update one or more fields on an existing variable.

update:

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this host.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **key** (*string*) – A key name string value (or comma separated list) of instance.
- **value** (*string*) – A value of instance.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**key** (*string*) – (required)
- **results[]** .**value** (*string*) – (required)

POST /host/{id}/variables/

Inventory hosts variables.

retrieve:

Return a variable of inventory host.

list:

Return all variables of inventory host.

create:

Create a new variable of inventory host.

destroy:

Remove an existing variable.

partial_update:

Update one or more fields on an existing variable.

update:

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this host.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- **201 Created** – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

GET /host/{id}/variables/{variables_id}/

Inventory hosts variables.

retrieve:

Return a variable of inventory host.

list:

Return all variables of inventory host.

create:

Create a new variable of inventory host.

destroy:

Remove an existing variable.

partial_update:

Update one or more fields on an existing variable.

update:

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this host.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PUT /host/{id}/variables/{variables_id}/

Inventory hosts variables.

retrieve:

Return a variable of inventory host.

list:

Return all variables of inventory host.

create:

Create a new variable of inventory host.

destroy:

Remove an existing variable.

partial_update:

Update one or more fields on an existing variable.

update:

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this host.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PATCH /host/{id}/variables/{variables_id}/

Inventory hosts variables.

retrieve:

Return a variable of inventory host.

list:

Return all variables of inventory host.

create:

Create a new variable of inventory host.

destroy:

Remove an existing variable.

partial_update:

Update one or more fields on an existing variable.

update:

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this host.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- **200 OK** – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

DELETE /host/{id}/variables/{variables_id}/

Inventory hosts variables.

retrieve:

Return a variable of inventory host.

list:

Return all variables of inventory host.

create:

Create a new variable of inventory host.

destroy:

Remove an existing variable.

partial_update:

Update one or more fields on an existing variable.

update:

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this host.

- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 204 No Content – NO CONTENT

GET /inventory/

Return all inventories.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **plugin** (*array*) –
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]**.**id** (*integer*) – (read only)
- **results[]**.**name** (*string*) –
- **results[]**.**plugin** (*string*) – (read only)
- **results[]**.**state_managed** (*boolean*) – (read only)

POST /inventory/

Create a new inventory.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **plugin** (*string*) – (required)
- **state_managed** (*boolean*) – (read only)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **plugin** (*string*) – (required)
- **state_managed** (*boolean*) – (read only)

POST /inventory/import_inventory/

Import inventory from file.

Request JSON Object

- **data** (*string*) – (required)
- **inventory_id** (*integer*) – (read only)
- **name** (*string*) – (required)
- **plugin** (*string*) – (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **data** (*string*) – (required)
- **inventory_id** (*integer*) – (read only)
- **name** (*string*) – (required)
- **plugin** (*string*) – (required)

GET /inventory/{id}/

Return a inventory instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **plugin** (*string*) – (read only)
- **state_managed** (*boolean*) – (read only)

PUT /inventory/{id}/

Update a inventory.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.

Request JSON Object

- **id** (*integer*) – (read only)

- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **plugin** (*string*) – (read only)
- **state_managed** (*boolean*) – (read only)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **plugin** (*string*) – (read only)
- **state_managed** (*boolean*) – (read only)

PATCH /inventory/{id}/

Update one or more fields on an existing inventory.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **plugin** (*string*) – (read only)
- **state_managed** (*boolean*) – (read only)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **plugin** (*string*) – (read only)
- **state_managed** (*boolean*) – (read only)

DELETE /inventory/{id}/

Remove an existing inventory.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.

Status Codes

- 204 No Content – NO CONTENT

GET /inventory/{id}/all_groups/

Return all groups.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.

Query Parameters

- **__deep_parent** (*integer*) –
- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[].children** (*boolean*) –
- **results[].id** (*integer*) – (read only)
- **results[].name** (*string*) –

GET /inventory/{id}/all_groups/{all_groups_id}/

Return a group instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **all_groups_id** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

GET /inventory/{id}/all_hosts/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **type** (*string*) – Instance type.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- **200 OK** – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**name** (*string*) –
- **results[]** .**type** (*string*) –

GET /inventory/{id}/all_hosts/{all_hosts_id}/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **all_hosts_id** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

POST /inventory/{id}/copy/

Endpoint which copy instance with deps.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

GET /inventory/{id}/group/

Return all groups.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.

Query Parameters

- **__deep_parent** (*integer*) –

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]**.**children** (*boolean*) –
- **results[]**.**id** (*integer*) – (read only)
- **results[]**.**name** (*string*) –

POST /inventory/{id}/group/

Create a new group.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.

Request JSON Object

- **children** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **children** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

GET /inventory/{id}/group/{group_id}/

Return a group instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

PUT /inventory/{id}/group/{group_id}/

Update a group.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –

Request JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

Status Codes

- 200 OK – OK

Response JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

PATCH /inventory/{id}/group/{group_id}/

Update one or more fields on an existing group.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –

Request JSON Object

- **children** (*boolean*) – (read only)

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

Status Codes

- 200 OK – OK

Response JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

DELETE /inventory/{id}/group/{group_id}/

Remove an existing group.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –

Status Codes

- 204 No Content – NO CONTENT

POST /inventory/{id}/group/{group_id}/copy/

Endpoint which copy instance with deps.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

GET /inventory/{id}/group/{group_id}/groups/

Return all groups.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –

Query Parameters

- **__deep_parent** (*integer*) –

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]**.**children** (*boolean*) –
- **results[]**.**id** (*integer*) – (read only)
- **results[]**.**name** (*string*) –

POST /inventory/{id}/group/{group_id}/groups/

Create a new group.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –

Request JSON Object

- **children** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **children** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

GET /inventory/{id}/group/{group_id}/groups/{groups_id}/

Return a group instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Status Codes

- 200 OK – OK

Response JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

PUT /inventory/{id}/group/{group_id}/groups/{groups_id}/

Update a group.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Request JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

Status Codes

- 200 OK – OK

Response JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

PATCH /inventory/{id}/group/{group_id}/groups/{groups_id}/

Update one or more fields on an existing group.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.

- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Request JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

Status Codes

- 200 OK – OK

Response JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

DELETE /inventory/{id}/group/{group_id}/groups/{groups_id}/

Remove an existing group.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Status Codes

- 204 No Content – NO CONTENT

POST /inventory/{id}/group/{group_id}/groups/{groups_id}/copy/

Endpoint which copy instance with deps.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)

- **name** (*string*) –

GET /inventory/{id}/group/{group_id}/groups/{groups_id}/hosts/

Return all hosts.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **type** (*string*) – Instance type.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**name** (*string*) –
- **results[]** .**type** (*string*) –

POST /inventory/{id}/group/{group_id}/groups/{groups_id}/hosts/

Create a new host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –

- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

GET /inventory/{id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/

Return a host instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

PUT /inventory/{id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

PATCH /inventory/{id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/

Update one or more fields on an existing host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

DELETE /inventory/{id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/

Remove an existing host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Status Codes

- 204 No Content – NO CONTENT

POST /inventory/{id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/copy/

Endpoint which copy instance with deps.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

Status Codes

- **201 Created** – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

PATCH /inventory/{id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/set_owner/

Change instance owner.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **owner** (*integer*) – (required)

Status Codes

- **200 OK** – OK

Response JSON Object

- **owner** (*integer*) – (required)

GET /inventory/{id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/variables/

Return all variables of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.

- **key** (*string*) – A key name string value (or comma separated list) of instance.
- **value** (*string*) – A value of instance.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]**.**id** (*integer*) – (read only)
- **results[]**.**key** (*string*) – (required)
- **results[]**.**value** (*string*) – (required)

POST /inventory/{id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/variables/

Create a new variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

GET /inventory/{id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/variables/{variables_id}/

Return a variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.

- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PUT /inventory/{id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/variables/{variables_id}/

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PATCH /inventory/{id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/variables/{variables_id}/

Update one or more fields on an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.

- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

DELETE /inventory/{id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/variables/{variables_id}/

Remove an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 204 No Content – NO CONTENT

PATCH /inventory/{id}/group/{group_id}/groups/{groups_id}/set_owner/

Change instance owner.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.

Request JSON Object

- **owner** (*integer*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **owner** (*integer*) – (required)

GET /inventory/{id}/group/{group_id}/groups/{groups_id}/variables/

Return all variables of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **key** (*string*) – A key name string value (or comma separated list) of instance.
- **value** (*string*) – A value of instance.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**key** (*string*) – (required)
- **results[]** .**value** (*string*) – (required)

POST /inventory/{id}/group/{group_id}/groups/{groups_id}/variables/

Create a new variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

GET /inventory/{id}/group/{group_id}/groups/{groups_id}/variables/{variables_id}/

Return a variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PUT /inventory/{id}/group/{group_id}/groups/{groups_id}/variables/{variables_id}/

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PATCH /inventory/{id}/group/{group_id}/groups/{groups_id}/variables/{variables_id}/

Update one or more fields on an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.

- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

DELETE /inventory/{id}/group/{group_id}/groups/{groups_id}/variables/{variables_id}/

Remove an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 204 No Content – NO CONTENT

GET /inventory/{id}/group/{group_id}/hosts/

Return all hosts.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **type** (*string*) – Instance type.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.

- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**name** (*string*) –
- **results[]** .**type** (*string*) –

POST /inventory/{id}/group/{group_id}/hosts/

Create a new host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

GET /inventory/{id}/group/{group_id}/hosts/{hosts_id}/

Return a host instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

PUT /inventory/{id}/group/{group_id}/hosts/{hosts_id}/

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

PATCH /inventory/{id}/group/{group_id}/hosts/{hosts_id}/

Update one or more fields on an existing host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

DELETE /inventory/{id}/group/{group_id}/hosts/{hosts_id}/

Remove an existing host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Status Codes

- 204 **No Content** – NO CONTENT

POST /inventory/{id}/group/{group_id}/hosts/{hosts_id}/copy/

Endpoint which copy instance with deps.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

Status Codes

- 201 **Created** – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

PATCH /inventory/{id}/group/{group_id}/hosts/{hosts_id}/set_owner/

Change instance owner.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **owner** (*integer*) – (required)

Status Codes

- 200 **OK** – OK

Response JSON Object

- **owner** (*integer*) – (required)

GET /inventory/{id}/group/{group_id}/hosts/{hosts_id}/variables/

Return all variables of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **key** (*string*) – A key name string value (or comma separated list) of instance.
- **value** (*string*) – A value of instance.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]**.**id** (*integer*) – (read only)
- **results[]**.**key** (*string*) – (required)
- **results[]**.**value** (*string*) – (required)

POST /inventory/{id}/group/{group_id}/hosts/{hosts_id}/variables/

Create a new variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)

- **key** (*string*) – (required)
- **value** (*string*) – (required)

GET /inventory/{id}/group/{group_id}/hosts/{hosts_id}/variables/{variables_id}/

Return a variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PUT /inventory/{id}/group/{group_id}/hosts/{hosts_id}/variables/{variables_id}/

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PATCH /inventory/{id}/group/{group_id}/hosts/{hosts_id}/variables/{variables_id}/

Update one or more fields on an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

DELETE /inventory/{id}/group/{group_id}/hosts/{hosts_id}/variables/{variables_id}/

Remove an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 204 No Content – NO CONTENT

PATCH /inventory/{id}/group/{group_id}/set_owner/

Change instance owner.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –

Request JSON Object

- **owner** (*integer*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **owner** (*integer*) – (required)

GET /inventory/{id}/group/{group_id}/variables/

Return all variables of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.

- **key** (*string*) – A key name string value (or comma separated list) of instance.
- **value** (*string*) – A value of instance.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]**.**id** (*integer*) – (read only)
- **results[]**.**key** (*string*) – (required)
- **results[]**.**value** (*string*) – (required)

POST /inventory/{id}/group/{group_id}/variables/

Create a new variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

GET /inventory/{id}/group/{group_id}/variables/{variables_id}/

Return a variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PUT /inventory/{id}/group/{group_id}/variables/{variables_id}/

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PATCH /inventory/{id}/group/{group_id}/variables/{variables_id}/

Update one or more fields on an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

DELETE /inventory/{id}/group/{group_id}/variables/{variables_id}/

Remove an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **group_id** (*string*) –
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 204 No Content – NO CONTENT

GET /inventory/{id}/hosts/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **type** (*string*) – Instance type.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)

- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**name** (*string*) –
- **results[]** .**type** (*string*) –

POST /inventory/{id}/hosts/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 201 [Created](#) – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

GET /inventory/{id}/hosts/{hosts_id}/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

PUT /inventory/{id}/hosts/{hosts_id}/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

PATCH /inventory/{id}/hosts/{hosts_id}/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –

- **owner** (*integer*) – (read only)
- **type** (*string*) –

DELETE /inventory/{id}/hosts/{hosts_id}/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Status Codes

- **204 No Content** – NO CONTENT

POST /inventory/{id}/hosts/{hosts_id}/copy/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

PATCH /inventory/{id}/hosts/{hosts_id}/set_owner/

Change instance owner.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **owner** (*integer*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **owner** (*integer*) – (required)

GET /inventory/{id}/hosts/{hosts_id}/variables/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **key** (*string*) – A key name string value (or comma separated list) of instance.
- **value** (*string*) – A value of instance.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.

- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]**.**id** (*integer*) – (read only)
- **results[]**.**key** (*string*) – (required)
- **results[]**.**value** (*string*) – (required)

POST /inventory/{id}/hosts/{hosts_id}/variables/
Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

GET `/inventory/{id}/hosts/{hosts_id}/variables/{variables_id}/`

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PUT `/inventory/{id}/hosts/{hosts_id}/variables/{variables_id}/`

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.

- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PATCH /inventory/{id}/hosts/{hosts_id}/variables/{variables_id}/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)

- **key** (*string*) – (required)
- **value** (*string*) – (required)

DELETE /inventory/{id}/hosts/{hosts_id}/variables/{variables_id}/

Manage inventory hosts.

retrieve:

Return a host instance.

list:

Return all hosts.

create:

Create a new host.

destroy:

Remove an existing host.

partial_update:

Update one or more fields on an existing host.

update:

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- **204 No Content** – NO CONTENT

PATCH /inventory/{id}/set_owner/

Change instance owner.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.

Request JSON Object

- **owner** (*integer*) – (required)

Status Codes

- **200 OK** – OK

Response JSON Object

- **owner** (*integer*) – (required)

GET /inventory/{id}/state/

State

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.

Status Codes

- **200 OK** – OK

Response JSON Object

- **data** (*string*) – (required)

PUT /inventory/{id}/state/

State

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.

Request JSON Object

- **data** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **data** (*string*) – (required)

GET /inventory/{id}/variables/

Inventory hosts variables.

retrieve:

Return a variable of inventory host.

list:

Return all variables of inventory host.

create:

Create a new variable of inventory host.

destroy:

Remove an existing variable.

partial_update:

Update one or more fields on an existing variable.

update:

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **key** (*string*) – A key name string value (or comma separated list) of instance.
- **value** (*string*) – A value of instance.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)

- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**key** (*string*) – (required)
- **results[]** .**value** (*string*) – (required)

POST /inventory/{id}/variables/

Inventory hosts variables.

retrieve:

Return a variable of inventory host.

list:

Return all variables of inventory host.

create:

Create a new variable of inventory host.

destroy:

Remove an existing variable.

partial_update:

Update one or more fields on an existing variable.

update:

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- **201 Created** – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

GET /inventory/{id}/variables/{variables_id}/

Inventory hosts variables.

retrieve:

Return a variable of inventory host.

list:

Return all variables of inventory host.

create:

Create a new variable of inventory host.

destroy:

Remove an existing variable.

partial_update:

Update one or more fields on an existing variable.

update:

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PUT /inventory/{id}/variables/{variables_id}/

Inventory hosts variables.

retrieve:

Return a variable of inventory host.

list:

Return all variables of inventory host.

create:

Create a new variable of inventory host.

destroy:

Remove an existing variable.

partial_update:

Update one or more fields on an existing variable.

update:

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PATCH /inventory/{id}/variables/{variables_id}/

Inventory hosts variables.

retrieve:

Return a variable of inventory host.

list:

Return all variables of inventory host.

create:

Create a new variable of inventory host.

destroy:

Remove an existing variable.

partial_update:

Update one or more fields on an existing variable.

update:

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

DELETE /inventory/{id}/variables/{variables_id}/

Inventory hosts variables.

retrieve:

Return a variable of inventory host.

list:

Return all variables of inventory host.

create:

Create a new variable of inventory host.

destroy:

Remove an existing variable.

partial_update:

Update one or more fields on an existing variable.

update:

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this inventory.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 204 No Content – NO CONTENT

GET /project/

Return all projects.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **status** (*array*) – Project sync status.
- **status__not** (*array*) – Project sync status.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**name** (*string*) –
- **results[]** .**status** (*string*) – (read only)
- **results[]** .**type** (*string*) – (read only)

POST /project/

Create a new project.

Request JSON Object

- **additional_playbook_path** (*string*) –
- **auth_data** (*string*) –
- **branch** (*string*) –

- **id** (*integer*) – (read only)
- **name** (*string*) – (required)
- **repo_auth** (*string*) –
- **repository** (*string*) –
- **status** (*string*) – (read only)
- **type** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **additional_playbook_path** (*string*) –
- **auth_data** (*string*) –
- **branch** (*string*) –
- **id** (*integer*) – (read only)
- **name** (*string*) – (required)
- **repo_auth** (*string*) –
- **repository** (*string*) –
- **status** (*string*) – (read only)
- **type** (*string*) –

GET /project/{id}/

Return a project instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

Status Codes

- 200 OK – OK

Response JSON Object

- **branch** (*string*) – (read only)
- **execute_view_data** (*object*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **readme_content** (*string*) – (read only)
- **repository** (*string*) –
- **revision** (*string*) – (read only)
- **status** (*string*) – (read only)

PUT /project/{id}/

Update a project.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

Request JSON Object

- **branch** (*string*) – (read only)
- **execute_view_data** (*object*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **readme_content** (*string*) – (read only)
- **repository** (*string*) –
- **revision** (*string*) – (read only)
- **status** (*string*) – (read only)

Status Codes

- 200 OK – OK

Response JSON Object

- **branch** (*string*) – (read only)
- **execute_view_data** (*object*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **readme_content** (*string*) – (read only)
- **repository** (*string*) –
- **revision** (*string*) – (read only)
- **status** (*string*) – (read only)

PATCH /project/{id}/

Update one or more fields on an existing project.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

Request JSON Object

- **branch** (*string*) – (read only)
- **execute_view_data** (*object*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **readme_content** (*string*) – (read only)
- **repository** (*string*) –
- **revision** (*string*) – (read only)
- **status** (*string*) – (read only)

Status Codes

- 200 OK – OK

Response JSON Object

- **branch** (*string*) – (read only)
- **execute_view_data** (*object*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **readme_content** (*string*) – (read only)
- **repository** (*string*) –
- **revision** (*string*) – (read only)
- **status** (*string*) – (read only)

DELETE /project/{id}/

Remove an existing project.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

Status Codes

- 204 No Content – NO CONTENT

GET /project/{id}/_project_template_options/

ExecutionTemplateOption(id, name, template, notes, arguments)

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **template** (*string*) – Search by template's primary key or name
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[] .extended_name** (*string*) – (required)
- **results[] .id** (*string*) –

- **results[].template** (*integer*) – (required)

GET /project/{id}/_project_template_options/{_project_template_options_id}/

ExecutionTemplateOption(id, name, template, notes, arguments)

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **_project_template_options_id** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **extended_name** (*string*) – (required)
- **id** (*string*) –
- **template** (*integer*) – (required)

GET /project/{id}/ansible_modules/

Ansible modules for project.

retrieve:

Return a module details of project instance.

list:

Return all available modules of project.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

Query Parameters

- **path** (*string*) – Full path to module.
- **path__not** (*string*) – Full path to module.
- **name** (*string*) – Full path to module.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[].id** (*integer*) – (read only)
- **results[].path** (*string*) – (required)

GET /project/{id}/ansible_modules/{ansible_modules_id}/

Ansible modules for project.

retrieve:

Return a module details of project instance.

list:

Return all available modules of project.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **ansible_modules_id** (*integer*) – A unique integer value identifying instance of this ansible_modules sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **data** (*string*) – (required)
- **id** (*integer*) – (read only)
- **path** (*string*) – (required)

GET /project/{id}/ansible_playbooks/

Ansible playbooks for project.

retrieve:

Return a playbook of project instance.

list:

Return all playbooks of project.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

Query Parameters

- **playbook** (*string*) – Playbook filename.
- **playbook__not** (*string*) – Playbook filename.
- **pb_filter** (*string*) – Playbook filename - filter for prefetch.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)

- **results[]**.**name** (*string*) –
- **results[]**.**playbook** (*string*) – (required)

GET /project/{id}/ansible_playbooks/{ansible_playbooks_id}/

Ansible playbooks for project.

retrieve:

Return a playbook of project instance.

list:

Return all playbooks of project.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **ansible_playbooks_id** (*integer*) – A unique integer value identifying instance of this ansible_playbooks sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **playbook** (*string*) – (required)

POST /project/{id}/copy/

Endpoint which copy instance with deps.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **status** (*string*) – (read only)
- **type** (*string*) – (read only)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **status** (*string*) – (read only)
- **type** (*string*) – (read only)

POST /project/{id}/execute_ansible_module/

Execute ANSIBLE_MODULE plugin.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

Request JSON Object

- **args** (*string*) – host pattern
- **background** (*integer*) – run asynchronously, failing after X seconds (default=N/A)
- **become** (*boolean*) – run operations with become (does not imply password prompting)
- **become_method** (*string*) – privilege escalation method to use (default=sudo), use *ansible-doc -t become -l* to list valid choices.
- **check** (*boolean*) – don't make any changes; instead, try to predict some of the changes that may occur
- **connection** (*string*) – connection type to use (default=smart)
- **diff** (*boolean*) – when changing (small) files and templates, show the differences in those files; works great with `--check`
- **extra_vars** (*string*) – set additional variables as key=value or YAML/JSON, if filename prepend with `@`
- **forks** (*integer*) – specify number of parallel processes to use (default=5)
- **group** (*string*) –
- **inventory** (*string*) – specify inventory host path or comma separated host list. `--inventory-file` is deprecated
- **limit** (*string*) – further limit selected hosts to an additional pattern
- **list_hosts** (*boolean*) – outputs a list of matching hosts; does not execute anything else
- **module** (*string*) – (required)
- **one_line** (*boolean*) – condense output
- **playbook_dir** (*string*) – Since this tool does not use playbooks, use this as a substitute playbook directory. This sets the relative path for many features including roles/group_vars/ etc.
- **poll** (*integer*) – set the poll interval if using `-B` (default=15)
- **private_key** (*string*) – use this file to authenticate the connection
- **scp_extra_args** (*string*) – specify extra arguments to pass to scp only (e.g. `-l`)
- **sftp_extra_args** (*string*) – specify extra arguments to pass to sftp only (e.g. `-f, -l`)
- **ssh_common_args** (*string*) – specify common arguments to pass to sftp/scp/ssh (e.g. `ProxyCommand`)
- **ssh_extra_args** (*string*) – specify extra arguments to pass to ssh only (e.g. `-R`)
- **syntax_check** (*boolean*) – perform a syntax check on the playbook, but do not execute it
- **timeout** (*integer*) – override the connection timeout in seconds (default=10)
- **tree** (*string*) – log output to this directory
- **user** (*string*) – connect as this user (default=None)
- **vault_password_file** (*string*) – vault password file
- **verbose** (*integer*) – verbose mode (`-vvv` for more, `-vvvv` to enable connection debugging)

Status Codes

- 201 `Created` – CREATED

Response JSON Object

- **detail** (*string*) – (read only)

- **executor** (*integer*) – (read only)
- **history_id** (*integer*) – (read only)

POST /project/{id}/execute_ansible_playbook/

Execute ANSIBLE_PLAYBOOK plugin.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

Request JSON Object

- **args** (*string*) – Playbook(s)
- **become** (*boolean*) – run operations with become (does not imply password prompting)
- **become_method** (*string*) – privilege escalation method to use (default=sudo), use *ansible-doc -t become -l* to list valid choices.
- **check** (*boolean*) – don't make any changes; instead, try to predict some of the changes that may occur
- **connection** (*string*) – connection type to use (default=smart)
- **diff** (*boolean*) – when changing (small) files and templates, show the differences in those files; works great with *--check*
- **extra_vars** (*string*) – set additional variables as key=value or YAML/JSON, if filename prepend with *@*
- **flush_cache** (*boolean*) – clear the fact cache for every host in inventory
- **force_handlers** (*boolean*) – run handlers even if a task fails
- **forks** (*integer*) – specify number of parallel processes to use (default=5)
- **inventory** (*string*) – specify inventory host path or comma separated host list. *--inventory-file* is deprecated
- **limit** (*string*) – further limit selected hosts to an additional pattern
- **list_hosts** (*boolean*) – outputs a list of matching hosts; does not execute anything else
- **list_tags** (*boolean*) – list all available tags
- **list_tasks** (*boolean*) – list all tasks that would be executed
- **playbook** (*string*) – (required)
- **private_key** (*string*) – use this file to authenticate the connection
- **scp_extra_args** (*string*) – specify extra arguments to pass to scp only (e.g. *-l*)
- **sftp_extra_args** (*string*) – specify extra arguments to pass to sftp only (e.g. *-f, -l*)
- **skip_tags** (*string*) – only run plays and tasks whose tags do not match these values
- **ssh_common_args** (*string*) – specify common arguments to pass to sftp/scp/ssh (e.g. *ProxyCommand*)
- **ssh_extra_args** (*string*) – specify extra arguments to pass to ssh only (e.g. *-R*)
- **start_at_task** (*string*) – start the playbook at the task matching this name
- **step** (*boolean*) – one-step-at-a-time: confirm each task before running
- **syntax_check** (*boolean*) – perform a syntax check on the playbook, but do not execute it
- **tags** (*string*) – only run plays and tasks tagged with these values
- **timeout** (*integer*) – override the connection timeout in seconds (default=10)

- **user** (*string*) – connect as this user (default=None)
- **vault_password_file** (*string*) – vault password file
- **verbose** (*integer*) – verbose mode (-vvv for more, -vvvv to enable connection debugging)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **detail** (*string*) – (read only)
- **executor** (*integer*) – (read only)
- **history_id** (*integer*) – (read only)

GET /project/{id}/execution_templates/

Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **plugin** (*string*) –
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)

- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**name** (*string*) – (required)
- **results[]** .**plugin** (*string*) – (required)

POST /project/{id}/execution_templates/

Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

Request JSON Object

- **arguments** (*string*) – (required)
- **id** (*integer*) – (read only)
- **name** (*string*) – (required)
- **plugin** (*string*) – (required)

Status Codes

- 201 [Created](#) – CREATED

Response JSON Object

- **arguments** (*string*) – (required)
- **id** (*integer*) – (read only)
- **name** (*string*) – (required)
- **plugin** (*string*) – (required)

GET /project/{id}/execution_templates/{execution_templates_id}/

Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) – (required)
- **notes** (*string*) –
- **plugin** (*string*) – (read only)

PUT /project/{id}/execution_templates/{execution_templates_id}/

Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) – (required)
- **notes** (*string*) –

- **plugin** (*string*) – (read only)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) – (required)
- **notes** (*string*) –
- **plugin** (*string*) – (read only)

PATCH /project/{id}/execution_templates/{execution_templates_id}/

Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) – (required)
- **notes** (*string*) –
- **plugin** (*string*) – (read only)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) – (required)
- **notes** (*string*) –
- **plugin** (*string*) – (read only)

DELETE /project/{id}/execution_templates/{execution_templates_id}/

Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.

Status Codes

- **204 No Content** – NO CONTENT

POST /project/{id}/execution_templates/{execution_templates_id}/execute/

Execute template with option.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.

Request JSON Object

- **option** (*integer*) – (required)

Status Codes

- **201 Created** – CREATED

Response JSON Object

- **detail** (*string*) – (read only)
- **executor** (*integer*) – (read only)
- **history_id** (*integer*) – (read only)

GET /project/{id}/execution_templates/{execution_templates_id}/history/

Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **status** (*string*) – Status of execution.
- **mode** (*string*) – Module or playbook name.
- **kind** (*string*) – Kind of execution.
- **older** (*string*) – Older then this time
- **newer** (*string*) – Newer then this time
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]**.**executor** (*string*) – (required)
- **results[]**.**id** (*integer*) – (read only)
- **results[]**.**initiator** (*integer*) –
- **results[]**.**initiator_type** (*string*) –
- **results[]**.**inventory** (*string*) – (read only)
- **results[]**.**kind** (*string*) –
- **results[]**.**mode** (*string*) – (required)
- **results[]**.**options** (*string*) – (read only)
- **results[]**.**project** (*integer*) –
- **results[]**.**start_time** (*string*) –

- **results[]**.**status** (*string*) – (required)
- **results[]**.**stop_time** (*string*) –

GET /project/{id}/execution_templates/{execution_templates_id}/history/{history_id}/
Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.
- **history_id** (*integer*) – A unique integer value identifying instance of this history sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **execute_args** (*string*) – (required)
- **execution_time** (*integer*) – (required)
- **executor** (*string*) – (required)
- **id** (*integer*) – (read only)
- **initiator** (*integer*) –
- **initiator_type** (*string*) –
- **inventory** (*integer*) –
- **kind** (*string*) –
- **mode** (*string*) – (required)
- **options** (*string*) – (read only)
- **project** (*integer*) –
- **raw_args** (*string*) –
- **raw_inventory** (*string*) –
- **raw_stdout** (*string*) – (read only)
- **revision** (*string*) –

- **start_time** (*string*) –
- **status** (*string*) – (required)
- **stop_time** (*string*) –

DELETE /project/{id}/execution_templates/{execution_templates_id}/history/{history_id}/

Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.
- **history_id** (*integer*) – A unique integer value identifying instance of this history sublist.

Status Codes

- **204 No Content** – NO CONTENT

PATCH /project/{id}/execution_templates/{execution_templates_id}/history/{history_id}/cancel/

Cancel working task.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.
- **history_id** (*integer*) – A unique integer value identifying instance of this history sublist.

Status Codes

- **200 OK** – OK

Response JSON Object

- **detail** (*string*) – (read only)

DELETE /project/{id}/execution_templates/{execution_templates_id}/history/{history_id}/clear/

Clear execution output.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.
- **history_id** (*integer*) – A unique integer value identifying instance of this history sublist.

Status Codes

- **204 No Content** – NO CONTENT

GET /project/{id}/execution_templates/{execution_templates_id}/history/{history_id}/facts/

Get facts from “setup” ansible module execution.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.
- **history_id** (*integer*) – A unique integer value identifying instance of this history sublist.

Status Codes

- **200 OK** – OK

Response JSON Object

- **facts** (*string*) – (read only)

GET /project/{id}/execution_templates/{execution_templates_id}/options/

Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*string*) –
- **results[]** .**name** (*string*) – (required)

POST /project/{id}/execution_templates/{execution_templates_id}/options/

Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.

Request JSON Object

- **arguments** (*string*) – (required)
- **id** (*string*) – (read only)
- **name** (*string*) – (required)
- **notes** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **arguments** (*string*) – (required)
- **id** (*string*) – (read only)
- **name** (*string*) – (required)
- **notes** (*string*) –

GET /project/{id}/execution_templates/{execution_templates_id}/options/{options_id}/
Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.
- **options_id** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **arguments** (*string*) – (required)
- **id** (*string*) – (read only)
- **name** (*string*) – (required)
- **notes** (*string*) –

PUT /project/{id}/execution_templates/{execution_templates_id}/options/{options_id}/
Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.
- **options_id** (*string*) –

Request JSON Object

- **arguments** (*string*) – (required)
- **id** (*string*) – (read only)
- **name** (*string*) – (required)
- **notes** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **arguments** (*string*) – (required)
- **id** (*string*) – (read only)
- **name** (*string*) – (required)
- **notes** (*string*) –

PATCH /project/{id}/execution_templates/{execution_templates_id}/options/{options_id}/
Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.
- **options_id** (*string*) –

Request JSON Object

- **arguments** (*string*) – (required)
- **id** (*string*) – (read only)
- **name** (*string*) – (required)
- **notes** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **arguments** (*string*) – (required)
- **id** (*string*) – (read only)
- **name** (*string*) – (required)
- **notes** (*string*) –

DELETE /project/{id}/execution_templates/{execution_templates_id}/options/{options_id}/

Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.
- **options_id** (*string*) –

Status Codes

- 204 No Content – NO CONTENT

POST /project/{id}/execution_templates/{execution_templates_id}/options/{options_id}/execute/

Execute template with option.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.
- **options_id** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **detail** (*string*) – (read only)
- **executor** (*integer*) – (read only)
- **history_id** (*integer*) – (read only)

GET /project/{id}/execution_templates/{execution_templates_id}/options/{options_id}/periodic_tasks/

Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.
- **options_id** (*string*) –

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **schedule** (*string*) –
- **enabled** (*boolean*) –
- **save_result** (*boolean*) –
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.

- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]**.**enabled** (*boolean*) –
- **results[]**.**id** (*integer*) – (read only)
- **results[]**.**name** (*string*) – (required)
- **results[]**.**save_result** (*boolean*) –
- **results[]**.**schedule** (*string*) – (required)

POST /project/{id}/execution_templates/{execution_templates_id}/options/{options_id}/periodic_tasks/

Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.
- **options_id** (*string*) –

Request JSON Object

- **enabled** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) – (required)
- **notes** (*string*) –
- **save_result** (*boolean*) –
- **schedule** (*string*) – (required)

- **type** (*string*) – (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **enabled** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) – (required)
- **notes** (*string*) –
- **save_result** (*boolean*) –
- **schedule** (*string*) – (required)
- **type** (*string*) – (required)

GET /project/{id}/execution_templates/{execution_templates_id}/options/{options_id}/periodic_tasks/{periodic_tasks_id}/

Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.
- **options_id** (*string*) –
- **periodic_tasks_id** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **enabled** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) – (required)
- **notes** (*string*) –
- **save_result** (*boolean*) –

- **schedule** (*string*) – (required)
- **type** (*string*) – (required)

PUT /project/{id}/execution_templates/{execution_templates_id}/options/{options_id}/periodic_tasks/{periodic_tasks_id}/

Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.
- **options_id** (*string*) –
- **periodic_tasks_id** (*string*) –

Request JSON Object

- **enabled** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) – (required)
- **notes** (*string*) –
- **save_result** (*boolean*) –
- **schedule** (*string*) – (required)
- **type** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **enabled** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) – (required)
- **notes** (*string*) –
- **save_result** (*boolean*) –
- **schedule** (*string*) – (required)

- **type** (*string*) – (required)

PATCH /project/{id}/execution_templates/{execution_templates_id}/options/{options_id}/periodic_tasks/{periodic_tasks_id}/

Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.
- **options_id** (*string*) –
- **periodic_tasks_id** (*string*) –

Request JSON Object

- **enabled** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) – (required)
- **notes** (*string*) –
- **save_result** (*boolean*) –
- **schedule** (*string*) – (required)
- **type** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **enabled** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) – (required)
- **notes** (*string*) –
- **save_result** (*boolean*) –
- **schedule** (*string*) – (required)
- **type** (*string*) – (required)

DELETE /project/{id}/execution_templates/{execution_templates_id}/options/{options_id}/periodic_tasks/{periodic_tasks_id}/

Manage execution templates of project.

retrieve:

Return an execution template instance.

list:

Return all execution templates in project.

create:

Create a new execution template.

destroy:

Remove an existing execution template.

partial_update:

Update one or more fields on an existing execution template.

update:

Update an execution template.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **execution_templates_id** (*integer*) – A unique integer value identifying instance of this execution_templates sublist.
- **options_id** (*string*) –
- **periodic_tasks_id** (*string*) –

Status Codes

- **204 No Content** – NO CONTENT

GET /project/{id}/history/

View previous executions.

retrieve:

Return an execution history instance.

list:

Return all history instances of executions.

destroy:

Remove an existing history record.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **status** (*string*) – Status of execution.
- **mode** (*string*) – Module or playbook name.
- **kind** (*string*) – Kind of execution.
- **older** (*string*) – Older then this time
- **newer** (*string*) – Newer then this time

- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]**.**executor** (*string*) – (required)
- **results[]**.**id** (*integer*) – (read only)
- **results[]**.**initiator** (*integer*) –
- **results[]**.**initiator_type** (*string*) –
- **results[]**.**inventory** (*string*) – (read only)
- **results[]**.**kind** (*string*) –
- **results[]**.**mode** (*string*) – (required)
- **results[]**.**options** (*string*) – (read only)
- **results[]**.**project** (*integer*) –
- **results[]**.**start_time** (*string*) –
- **results[]**.**status** (*string*) – (required)
- **results[]**.**stop_time** (*string*) –

GET /project/{id}/history/{history_id}/

View previous executions.

retrieve:

Return an execution history instance.

list:

Return all history instances of executions.

destroy:

Remove an existing history record.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **history_id** (*integer*) – A unique integer value identifying instance of this history sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **execute_args** (*string*) – (required)
- **execution_time** (*integer*) – (required)
- **executor** (*string*) – (required)
- **id** (*integer*) – (read only)
- **initiator** (*integer*) –
- **initiator_type** (*string*) –
- **inventory** (*integer*) –
- **kind** (*string*) –
- **mode** (*string*) – (required)
- **options** (*string*) – (read only)
- **project** (*integer*) –
- **raw_args** (*string*) –
- **raw_inventory** (*string*) –
- **raw_stdout** (*string*) – (read only)
- **revision** (*string*) –
- **start_time** (*string*) –
- **status** (*string*) – (required)
- **stop_time** (*string*) –

DELETE /project/{id}/history/{history_id}/

View previous executions.

retrieve:

Return an execution history instance.

list:

Return all history instances of executions.

destroy:

Remove an existing history record.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **history_id** (*integer*) – A unique integer value identifying instance of this history sublist.

Status Codes

- **204 No Content** – NO CONTENT

PATCH /project/{id}/history/{history_id}/cancel/

Cancel working task.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **history_id** (*integer*) – A unique integer value identifying instance of this history sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **detail** (*string*) – (read only)

DELETE /project/{id}/history/{history_id}/clear/

Clear execution output.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **history_id** (*integer*) – A unique integer value identifying instance of this history sublist.

Status Codes

- 204 No Content – NO CONTENT

GET /project/{id}/history/{history_id}/facts/

Get facts from “setup” ansible module execution.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **history_id** (*integer*) – A unique integer value identifying instance of this history sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **facts** (*string*) – (read only)

GET /project/{id}/inventory/

Inventory(id, hidden, notes, owner, name, plugin, _inventory_state)

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **plugin** (*array*) –
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**name** (*string*) –
- **results[]** .**plugin** (*string*) – (read only)
- **results[]** .**state_managed** (*boolean*) – (read only)

POST /project/{id}/inventory/

Inventory(id, hidden, notes, owner, name, plugin, _inventory_state)

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **plugin** (*string*) – (required)
- **state_managed** (*boolean*) – (read only)

Status Codes

- 201 [Created](#) – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **plugin** (*string*) – (required)
- **state_managed** (*boolean*) – (read only)

POST /project/{id}/inventory/import_inventory/

Import inventory from file.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

Request JSON Object

- **data** (*string*) – (required)
- **inventory_id** (*integer*) – (read only)
- **name** (*string*) – (required)
- **plugin** (*string*) – (required)

Status Codes

- 201 [Created](#) – CREATED

Response JSON Object

- **data** (*string*) – (required)
- **inventory_id** (*integer*) – (read only)
- **name** (*string*) – (required)
- **plugin** (*string*) – (required)

GET /project/{id}/inventory/{inventory_id}/

Inventory(id, hidden, notes, owner, name, plugin, _inventory_state)

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **plugin** (*string*) – (read only)
- **state_managed** (*boolean*) – (read only)

PUT /project/{id}/inventory/{inventory_id}/

Inventory(id, hidden, notes, owner, name, plugin, _inventory_state)

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **plugin** (*string*) – (read only)
- **state_managed** (*boolean*) – (read only)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **plugin** (*string*) – (read only)
- **state_managed** (*boolean*) – (read only)

PATCH /project/{id}/inventory/{inventory_id}/

Inventory(id, hidden, notes, owner, name, plugin, _inventory_state)

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **plugin** (*string*) – (read only)
- **state_managed** (*boolean*) – (read only)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **plugin** (*string*) – (read only)
- **state_managed** (*boolean*) – (read only)

DELETE /project/{id}/inventory/{inventory_id}/

Inventory(id, hidden, notes, owner, name, plugin, _inventory_state)

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –

Status Codes

- 204 No Content – NO CONTENT

GET /project/{id}/inventory/{inventory_id}/all_groups/

Return all groups.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –

Query Parameters

- **__deep_parent** (*integer*) –
- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.

- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]**.children (*boolean*) –
- **results[]**.id (*integer*) – (read only)
- **results[]**.name (*string*) –

GET /project/{id}/inventory/{inventory_id}/all_groups/{all_groups_id}/

Return a group instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **all_groups_id** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

GET /project/{id}/inventory/{inventory_id}/all_hosts/

Return all hosts.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **type** (*string*) – Instance type.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.

- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**name** (*string*) –
- **results[]** .**type** (*string*) –

GET /project/{id}/inventory/{inventory_id}/all_hosts/{all_hosts_id}/

Return a host instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **all_hosts_id** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

POST /project/{id}/inventory/{inventory_id}/copy/

Endpoint which copy instance with deps.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

GET /project/{id}/inventory/{inventory_id}/group/

Return all groups.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –

Query Parameters

- **__deep_parent** (*integer*) –
- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[].children** (*boolean*) –
- **results[].id** (*integer*) – (read only)
- **results[].name** (*string*) –

POST /project/{id}/inventory/{inventory_id}/group/

Create a new group.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –

Request JSON Object

- **children** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **children** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

GET /project/{id}/inventory/{inventory_id}/group/{group_id}/

Return a group instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

PUT /project/{id}/inventory/{inventory_id}/group/{group_id}/

Update a group.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –

Request JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

Status Codes

- 200 OK – OK

Response JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –

- **notes** (*string*) –
- **owner** (*integer*) – (read only)

PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/

Update one or more fields on an existing group.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –

Request JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

Status Codes

- 200 OK – OK

Response JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

DELETE /project/{id}/inventory/{inventory_id}/group/{group_id}/

Remove an existing group.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –

Status Codes

- 204 No Content – NO CONTENT

POST /project/{id}/inventory/{inventory_id}/group/{group_id}/copy/

Endpoint which copy instance with deps.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

GET /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/

Return all groups.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –

Query Parameters

- **__deep_parent** (*integer*) –
- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[].children** (*boolean*) –
- **results[].id** (*integer*) – (read only)
- **results[].name** (*string*) –

POST /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/

Create a new group.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –

Request JSON Object

- **children** (*boolean*) –

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **children** (*boolean*) –
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

GET /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/

Return a group instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Status Codes

- 200 OK – OK

Response JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

PUT /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/

Update a group.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Request JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –

- **notes** (*string*) –
- **owner** (*integer*) – (read only)

Status Codes

- 200 OK – OK

Response JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/

Update one or more fields on an existing group.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Request JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

Status Codes

- 200 OK – OK

Response JSON Object

- **children** (*boolean*) – (read only)
- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

DELETE /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/

Remove an existing group.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Status Codes

- 204 No Content – NO CONTENT

POST /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/copy/

Endpoint which copy instance with deps.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

GET /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/hosts/

Return all hosts.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **type** (*string*) – Instance type.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**name** (*string*) –
- **results[]** .**type** (*string*) –

POST /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/hosts/

Create a new host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- **201 Created** – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

GET /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/

Return a host instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

PUT /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/

Update one or more fields on an existing host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

DELETE /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/

Remove an existing host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Status Codes

- 204 No Content – NO CONTENT

POST /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/copy/

Endpoint which copy instance with deps.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/set_owner/

Change instance owner.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **owner** (*integer*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **owner** (*integer*) – (required)

GET /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/variables/

Return all variables of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **key** (*string*) – A key name string value (or comma separated list) of instance.
- **value** (*string*) – A value of instance.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**key** (*string*) – (required)
- **results[]** .**value** (*string*) – (required)

POST /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/variables/

Create a new variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

GET /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/variables/{variables_id}/

Return a variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PUT /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/variables/{variables_id}/

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/variables/{variables_id}/

Update one or more fields on an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

DELETE /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/variables/{variables_id}/

Remove an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 204 No Content – NO CONTENT

PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/set_owner/

Change instance owner.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sublist.

Request JSON Object

- **owner** (*integer*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **owner** (*integer*) – (required)

GET /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/variables/

Return all variables of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **key** (*string*) – A key name string value (or comma separated list) of instance.
- **value** (*string*) – A value of instance.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**key** (*string*) – (required)
- **results[]** .**value** (*string*) – (required)

POST /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/variables/

Create a new variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

GET /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/variables/{variables_id}/

Return a variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PUT /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/variables/{variables_id}/

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/variables/{variables_id}/

Update one or more fields on an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

DELETE /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/variables/{variables_id}/

Remove an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **groups_id** (*integer*) – A unique integer value identifying instance of this groups sub-list.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 204 No Content – NO CONTENT

GET /project/{id}/inventory/{inventory_id}/group/{group_id}/hosts/

Return all hosts.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

- **inventory_id** (*string*) –
- **group_id** (*string*) –

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **type** (*string*) – Instance type.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]**.**id** (*integer*) – (read only)
- **results[]**.**name** (*string*) –
- **results[]**.**type** (*string*) –

POST /project/{id}/inventory/{inventory_id}/group/{group_id}/hosts/

Create a new host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)

- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

GET /project/{id}/inventory/{inventory_id}/group/{group_id}/hosts/{hosts_id}/

Return a host instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

PUT /project/{id}/inventory/{inventory_id}/group/{group_id}/hosts/{hosts_id}/

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)

- **type** (*string*) –

PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/hosts/{hosts_id}/

Update one or more fields on an existing host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

DELETE /project/{id}/inventory/{inventory_id}/group/{group_id}/hosts/{hosts_id}/

Remove an existing host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Status Codes

- 204 No Content – NO CONTENT

POST /project/{id}/inventory/{inventory_id}/group/{group_id}/hosts/{hosts_id}/copy/

Endpoint which copy instance with deps.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/hosts/{hosts_id}/set_owner/

Change instance owner.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **owner** (*integer*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **owner** (*integer*) – (required)

GET /project/{id}/inventory/{inventory_id}/group/{group_id}/hosts/{hosts_id}/variables/

Return all variables of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **key** (*string*) – A key name string value (or comma separated list) of instance.
- **value** (*string*) – A value of instance.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**key** (*string*) – (required)
- **results[]** .**value** (*string*) – (required)

POST /project/{id}/inventory/{inventory_id}/group/{group_id}/hosts/{hosts_id}/variables/

Create a new variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

GET /project/{id}/inventory/{inventory_id}/group/{group_id}/hosts/{hosts_id}/variables/{variables_id}/

Return a variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)

- **key** (*string*) – (required)
- **value** (*string*) – (required)

PUT /project/{id}/inventory/{inventory_id}/group/{group_id}/hosts/{hosts_id}/variables/{variables_id}/

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/hosts/{hosts_id}/variables/{variables_id}/

Update one or more fields on an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)

- **key** (*string*) – (required)
- **value** (*string*) – (required)

DELETE /project/{id}/inventory/{inventory_id}/group/{group_id}/hosts/{hosts_id}/variables/{variables_id}/

Remove an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- **204 No Content** – NO CONTENT

PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/set_owner/

Change instance owner.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –

Request JSON Object

- **owner** (*integer*) – (required)

Status Codes

- **200 OK** – OK

Response JSON Object

- **owner** (*integer*) – (required)

GET /project/{id}/inventory/{inventory_id}/group/{group_id}/variables/

Return all variables of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **key** (*string*) – A key name string value (or comma separated list) of instance.
- **value** (*string*) – A value of instance.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.

- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]**.**id** (*integer*) – (read only)
- **results[]**.**key** (*string*) – (required)
- **results[]**.**value** (*string*) – (required)

POST /project/{id}/inventory/{inventory_id}/group/{group_id}/variables/

Create a new variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

GET /project/{id}/inventory/{inventory_id}/group/{group_id}/variables/{variables_id}/

Return a variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)

- **value** (*string*) – (required)

PUT /project/{id}/inventory/{inventory_id}/group/{group_id}/variables/{variables_id}/

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/variables/{variables_id}/

Update one or more fields on an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

DELETE /project/{id}/inventory/{inventory_id}/group/{group_id}/variables/{variables_id}/

Remove an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **group_id** (*string*) –
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 204 No Content – NO CONTENT

GET /project/{id}/inventory/{inventory_id}/hosts/

Return all hosts.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name** (*string*) – A name string value (or comma separated list) of instance.
- **type** (*string*) – Instance type.
- **variables** (*string*) – List of variables to filter. Comma separated “key:value” list.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **name__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**name** (*string*) –
- **results[]** .**type** (*string*) –

POST /project/{id}/inventory/{inventory_id}/hosts/

Create a new host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

GET /project/{id}/inventory/{inventory_id}/hosts/{hosts_id}/

Return a host instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

PUT /project/{id}/inventory/{inventory_id}/hosts/{hosts_id}/

Update a host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

PATCH /project/{id}/inventory/{inventory_id}/hosts/{hosts_id}/

Update one or more fields on an existing host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –
- **notes** (*string*) –
- **owner** (*integer*) – (read only)
- **type** (*string*) –

DELETE /project/{id}/inventory/{inventory_id}/hosts/{hosts_id}/

Remove an existing host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Status Codes

- 204 No Content – NO CONTENT

POST /project/{id}/inventory/{inventory_id}/hosts/{hosts_id}/copy/

Endpoint which copy instance with deps.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

Status Codes

- **201 Created** – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **name** (*string*) –

PATCH /project/{id}/inventory/{inventory_id}/hosts/{hosts_id}/set_owner/

Change instance owner.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **owner** (*integer*) – (required)

Status Codes

- **200 OK** – OK

Response JSON Object

- **owner** (*integer*) – (required)

GET /project/{id}/inventory/{inventory_id}/hosts/{hosts_id}/variables/

Return all variables of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **key** (*string*) – A key name string value (or comma separated list) of instance.
- **value** (*string*) – A value of instance.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.

- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]**.**id** (*integer*) – (read only)
- **results[]**.**key** (*string*) – (required)
- **results[]**.**value** (*string*) – (required)

POST /project/{id}/inventory/{inventory_id}/hosts/{hosts_id}/variables/

Create a new variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

GET /project/{id}/inventory/{inventory_id}/hosts/{hosts_id}/variables/{variables_id}/

Return a variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)

- **value** (*string*) – (required)

PUT /project/{id}/inventory/{inventory_id}/hosts/{hosts_id}/variables/{variables_id}/

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PATCH /project/{id}/inventory/{inventory_id}/hosts/{hosts_id}/variables/{variables_id}/

Update one or more fields on an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

DELETE /project/{id}/inventory/{inventory_id}/hosts/{hosts_id}/variables/{variables_id}/

Remove an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **hosts_id** (*integer*) – A unique integer value identifying instance of this hosts sublist.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 204 No Content – NO CONTENT

PATCH /project/{id}/inventory/{inventory_id}/set_owner/

Change instance owner.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –

Request JSON Object

- **owner** (*integer*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **owner** (*integer*) – (required)

GET /project/{id}/inventory/{inventory_id}/state/

State

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **data** (*string*) – (required)

PUT /project/{id}/inventory/{inventory_id}/state/

State

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –

Request JSON Object

- **data** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **data** (*string*) – (required)

GET /project/{id}/inventory/{inventory_id}/variables/

Return all variables of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **key** (*string*) – A key name string value (or comma separated list) of instance.
- **value** (*string*) – A value of instance.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]**.**id** (*integer*) – (read only)
- **results[]**.**key** (*string*) – (required)
- **results[]**.**value** (*string*) – (required)

POST /project/{id}/inventory/{inventory_id}/variables/

Create a new variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)

- **value** (*string*) – (required)

GET /project/{id}/inventory/{inventory_id}/variables/{variables_id}/

Return a variable of inventory host.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PUT /project/{id}/inventory/{inventory_id}/variables/{variables_id}/

Update variable value.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PATCH /project/{id}/inventory/{inventory_id}/variables/{variables_id}/

Update one or more fields on an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)

- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

DELETE /project/{id}/inventory/{inventory_id}/variables/{variables_id}/

Remove an existing variable.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **inventory_id** (*string*) –
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 204 No Content – NO CONTENT

PATCH /project/{id}/set_owner/

Change instance owner.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

Request JSON Object

- **owner** (*integer*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **owner** (*integer*) – (required)

PATCH /project/{id}/sync/

Synchronize project with repository.

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

Status Codes

- 200 OK – OK

Response JSON Object

- **detail** (*string*) – (read only)

GET /project/{id}/variables/

Variable(id, hidden, content_type, object_id, key, value)

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **key** (*string*) – A key name string value (or comma separated list) of instance.
- **value** (*string*) – A value of instance.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.
- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]** .**id** (*integer*) – (read only)
- **results[]** .**key** (*string*) – (required)
- **results[]** .**value** (*string*) – (required)

POST /project/{id}/variables/

Variable(id, hidden, content_type, object_id, key, value)

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

GET /project/{id}/variables/{variables_id}/

Variable(id, hidden, content_type, object_id, key, value)

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PUT /project/{id}/variables/{variables_id}/

Variable(id, hidden, content_type, object_id, key, value)

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

PATCH /project/{id}/variables/{variables_id}/

Variable(id, hidden, content_type, object_id, key, value)

Parameters

- **id** (*integer*) – A unique integer value identifying this project.
- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Request JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **id** (*integer*) – (read only)
- **key** (*string*) – (required)
- **value** (*string*) – (required)

DELETE /project/{id}/variables/{variables_id}/

Variable(id, hidden, content_type, object_id, key, value)

Parameters

- **id** (*integer*) – A unique integer value identifying this project.

- **variables_id** (*integer*) – A unique integer value identifying instance of this variables sublist.

Status Codes

- 204 No Content – NO CONTENT

GET /stats/

Status Codes

- 200 OK – OK

Response JSON Object

- **execution_plugins** (*integer*) – (required)
- **inventories** (*integer*) – (required)
- **inventory_plugins** (*integer*) – (required)
- **jobs.day[]**.all (*integer*) – (required)
- **jobs.day[]**.day (*string*) – (required)
- **jobs.day[]**.status (*string*) – (required)
- **jobs.day[]**.sum (*integer*) – (required)
- **jobs.month[]**.all (*integer*) – (required)
- **jobs.month[]**.month (*string*) – (required)
- **jobs.month[]**.status (*string*) – (required)
- **jobs.month[]**.sum (*integer*) – (required)
- **jobs.year[]**.all (*integer*) – (required)
- **jobs.year[]**.status (*string*) – (required)
- **jobs.year[]**.sum (*integer*) – (required)
- **jobs.year[]**.year (*string*) – (required)
- **projects** (*integer*) – (required)
- **users** (*integer*) – (required)

GET /user/

Return all users.

Query Parameters

- **id** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **username** (*string*) – A name string value (or comma separated list) of instance.
- **is_active** (*boolean*) – Boolean value meaning status of user.
- **first_name** (*string*) – Users first name.
- **last_name** (*string*) – Users last name.
- **email** (*string*) – Users e-mail value.
- **id__not** (*array*) – A unique integer value (or comma separated list) identifying this instance.
- **username__not** (*string*) – A name string value (or comma separated list) of instance.
- **ordering** (*array*) – Which field to use when ordering the results.
- **__search** (*string*) – A search term.
- **limit** (*integer*) – Number of results to return per page.

- **offset** (*integer*) – The initial index from which to return the results.

Status Codes

- 200 OK – OK

Response JSON Object

- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[]**.**email** (*string*) –
- **results[]**.**id** (*integer*) – (read only)
- **results[]**.**is_active** (*boolean*) –
- **results[]**.**username** (*string*) – Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only. (required)

POST /user/

Create a new user.

Request JSON Object

- **email** (*string*) –
- **first_name** (*string*) –
- **id** (*integer*) – (read only)
- **is_active** (*boolean*) –
- **last_name** (*string*) –
- **password** (*string*) – (required)
- **password2** (*string*) – (required)
- **username** (*string*) – Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only. (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **email** (*string*) –
- **first_name** (*string*) –
- **id** (*integer*) – (read only)
- **is_active** (*boolean*) –
- **last_name** (*string*) –
- **password** (*string*) – (required)
- **password2** (*string*) – (required)
- **username** (*string*) – Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only. (required)

GET /user/{id}/

Return a user instance.

Parameters

- **id** (*integer*) – A unique integer value identifying this user.

Status Codes

- 200 OK – OK

Response JSON Object

- **email** (*string*) –
- **first_name** (*string*) –
- **id** (*integer*) – (read only)
- **is_active** (*boolean*) –
- **last_name** (*string*) –
- **username** (*string*) – Required. 150 characters or fewer. Letters, digits and @/./+/_ only. (required)

PUT /user/{id}/

Update a user.

Parameters

- **id** (*integer*) – A unique integer value identifying this user.

Request JSON Object

- **email** (*string*) –
- **first_name** (*string*) –
- **id** (*integer*) – (read only)
- **is_active** (*boolean*) –
- **last_name** (*string*) –
- **username** (*string*) – Required. 150 characters or fewer. Letters, digits and @/./+/_ only. (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **email** (*string*) –
- **first_name** (*string*) –
- **id** (*integer*) – (read only)
- **is_active** (*boolean*) –
- **last_name** (*string*) –
- **username** (*string*) – Required. 150 characters or fewer. Letters, digits and @/./+/_ only. (required)

PATCH /user/{id}/

Update one or more fields on an existing user.

Parameters

- **id** (*integer*) – A unique integer value identifying this user.

Request JSON Object

- **email** (*string*) –
- **first_name** (*string*) –
- **id** (*integer*) – (read only)

- **is_active** (*boolean*) –
- **last_name** (*string*) –
- **username** (*string*) – Required. 150 characters or fewer. Letters, digits and @/./+/_ only. (required)

Status Codes

- 200 OK – OK

Response JSON Object

- **email** (*string*) –
- **first_name** (*string*) –
- **id** (*integer*) – (read only)
- **is_active** (*boolean*) –
- **last_name** (*string*) –
- **username** (*string*) – Required. 150 characters or fewer. Letters, digits and @/./+/_ only. (required)

DELETE /user/{id}/

Remove an existing user.

Parameters

- **id** (*integer*) – A unique integer value identifying this user.

Status Codes

- 204 No Content – NO CONTENT

GET /user/{id}/_settings/

Manage users.

Parameters

- **id** (*integer*) – A unique integer value identifying this user.

Status Codes

- 200 OK – OK

Response JSON Object

- **custom** (*string*) –
- **main.dark_mode** (*boolean*) –
- **main.language** (*string*) –

PUT /user/{id}/_settings/

Manage users.

Parameters

- **id** (*integer*) – A unique integer value identifying this user.

Request JSON Object

- **custom** (*string*) –
- **main.dark_mode** (*boolean*) –
- **main.language** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **custom** (*string*) –
- **main.dark_mode** (*boolean*) –
- **main.language** (*string*) –

POST /user/{id}/change_password/

Manage users.

Parameters

- **id** (*integer*) – A unique integer value identifying this user.

Request JSON Object

- **old_password** (*string*) – (required)
- **password** (*string*) – (required)
- **password2** (*string*) – (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **old_password** (*string*) – (required)
- **password** (*string*) – (required)
- **password2** (*string*) – (required)

POST /user/{id}/copy/

Endpoint which copy instance with deps.

Parameters

- **id** (*integer*) – A unique integer value identifying this user.

Request JSON Object

- **email** (*string*) –
- **id** (*integer*) – (read only)
- **is_active** (*boolean*) –
- **username** (*string*) – Required. 150 characters or fewer. Letters, digits and @/./+/_ only. (required)

Status Codes

- 201 Created – CREATED

Response JSON Object

- **email** (*string*) –
- **id** (*integer*) – (read only)
- **is_active** (*boolean*) –
- **username** (*string*) – Required. 150 characters or fewer. Letters, digits and @/./+/_ only. (required)

GET /user/{id}/twofa/

Manage users.

Parameters

- **id** (*integer*) – A unique integer value identifying this user.

Status Codes

- 200 OK – OK

Response JSON Object

- **enabled** (*boolean*) – (read only)
- **pin** (*string*) –
- **recovery** (*string*) –
- **secret** (*string*) –

PUT /user/{id}/twofa/

Manage users.

Parameters

- **id** (*integer*) – A unique integer value identifying this user.

Request JSON Object

- **enabled** (*boolean*) – (read only)
- **pin** (*string*) –
- **recovery** (*string*) –
- **secret** (*string*) –

Status Codes

- 200 OK – OK

Response JSON Object

- **enabled** (*boolean*) – (read only)
- **pin** (*string*) –
- **recovery** (*string*) –
- **secret** (*string*) –

HOW TO PARTICIPATE

If you found some bug in the codebase or documentation, or you want some improvement in ‘Polemarch’ project, you can easily inform us about it. Here is how to proceed:

1. Create (or find if it already exists) an [issue](#) for problem you want to solve. Maybe somebody is already working on it or maybe during discussion becomes clear that it is not a problem at all.
2. To investigate problem, you probably must be able to run cloned project. To do so install required dependencies. See “[Install from PyPI](#)” for more detailed explanation, which dependencies and why they are needed. System packages (example for Ubuntu 20.04):

```
sudo apt-get install python3-virtualenv python3.8 python3.8-dev gcc libffi-dev  
↳ libkrb5-dev libffi7 libssl-dev libyaml-dev libsasl2-dev libldap2-dev default-  
↳ libmysqlclient-dev sshpass git tox
```

Note: Your patch must work at Python 3.8 because older Python may not work correctly with Polemarch.

3. Make fork of our repository and clone it to your local development machine.

```
git clone https://gitlab.com/vstconsulting/polemarch.git
```

4. Create a branch for your work named with number of issue:

```
cd polemarch  
git checkout -b issue/1
```

5. Create and activate virtualenv:

```
tox -e contrib  
source env/bin/activate
```

6. Initialize empty database with all required stuff (tables and so on) for Polemarch:

```
python -m polemarch migrate
```

7. For your comfort you can edit `polemarch/main/settings.ini`, but make sure that changes in that file won't go to your commit. Or you can copy `polemarch/main/settings.ini` to `/etc/polemarch/settings.ini` (default settings location for Polemarch).

- First of all you can enable debug in settings.ini. Set `debug = true` in section `[main]`. Otherwise there will be no available static files and debug features when you start development web-server.
- You may also want to change `log_level` in section `[main]` for easy debugging.
- If you want to see console output of Polemarch webserver during it work, you need to add following option `daemon = false` in section `[uwsgi]`.

8. Run Polemarch GUI with web-server and investigate with your debugger how it works to find out what need to be changed:

```
# run web-server
python -m polemarch webserver
```

This command also starts worker, if you have added worker options in `/etc/polemarch/settings.ini`. More about worker section you can find in *“Install from PyPI”*.

9. Write tests for your changes (we prefer TDD approach). Execute those tests with all other Polemarch’s tests by:

```
tox
```

This command do PEP8 check of codebase and static analyzing with pylint and flake and run main python tests. Make sure that your code meets those checks.

Warning: Some tests linked to git may fail because local `file://` clones are considered unsafe by default. For more information please see [this topic](#). If you are encountered this problem, one of the solutions might be:

```
git config --global protocol.file.allow always
```

Keep in mind that this command allows file clone **globally at git level for all your projects**.

10. Reflect your changes in documentation (if needed). Build documentation, read what you have changed and make sure that all is right. To build documentation use:

```
tox -e bulddoc
```

11. Make commit. We prefer commit messages with briefly explanations of your changes. Unacceptable: “issue #1” or “fix”. Acceptable: “fix end slashes for GET in docs”.
12. Create pull request and refer it in issue.

Attention: You must agree to [our contributor agreement](#) to prevent any license problems to project in future with your contribution.

We prefer to format requests as follows:

- **Title** should start with a description of your changes and follow these rules:
 - Feature - if you added new functionality for the user;
 - Chore - if you refactored the code or optimized performance, but nothing changed for the user;
 - Fix - if you fixed some bug and didn’t add new independent functionality.
 - Docs - if you add some documents and didn’t make any changes to the code.
- **Description** should contain structured information about the work done:
 - BREAKING CHANGES - list of changes that break backward compatibility;
 - Changelog - list of common changes;
 - Closes/WIP - link to the issue you were working on.
 - Add screenshots for easy review of your changes.

That’s it. Thank you for your contribution.

VST CONSULTING CONTRIBUTOR AGREEMENT

This VST Consulting contributor Agreement (this “**Agreement**”) applies to any Contribution you make to any Work.

This is a binding legal agreement on you and any organization you represent. If you are signing this Agreement on behalf of your employer or other organization, you represent and warrant that you have the authority to agree to this Agreement on behalf of the organization.

7.1 1. Definitions

“**Contribution**” means any original work, including any modification of or addition to an existing work, that you submit to VST Consulting in any manner for inclusion in any Work.

“**VST Consulting**”, “**we**” and “**us**” means OOO VST Consulting.

“**Work**” means any project, work or materials owned or managed by OOO VST Consulting.

“**You**” and “**your**” means you and any organization on whose behalf you are entering this Agreement.

7.2 2. Copyright Assignment, License and Waiver

(a) **Assignment.** By submitting a Contribution, you assign to VST Consulting all right, title and interest in any copyright you have in the Contribution, and you waive any rights, including any moral rights, database rights, etc., that may affect your ownership of the copyright in the Contribution.

(b) **License to VST Consulting.** If your assignment in Section 2(a) is ineffective for any reason, you grant to us and to any recipient of any Work distributed by us, a perpetual, worldwide, transferable, non-exclusive, no-charge, royalty-free, irrevocable, and sublicensable licence to use, reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute Contributions and any derivative work created based on a Contribution. If your license grant is ineffective for any reason, you irrevocably waive and covenant to not assert any claim you may have against us, our successors in interest, and any of our direct or indirect licensees and customers, arising out of our, our successors in interest’s, or any of our direct or indirect licensees’ or customers’ use, reproduction, preparation of derivative works, public display, public performance, sublicense, and distribution of a Contribution. You also agree that we may publicly use your name and the name of any organization on whose behalf you’re entering into this Agreement in connection with publicizing the Work.

(c) **License to you.** We grant to you a perpetual, worldwide, transferable, non-exclusive, no-charge, royalty-free, irrevocable, and sublicensable license to use, reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute a Contribution and any derivative works you create based on a Contribution.

7.3 3. Patent License

You grant to us and to any recipient of any Work distributed by us, a perpetual, worldwide, transferable, non-exclusive, no-charge, royalty-free, irrevocable, and sublicensable patent license to make, have made, use, sell, offer to sell, import, and otherwise transfer the Contribution in whole or in part, along or included in any Work under any patent you own, or license from a third party, that is necessarily infringed by the Contribution or by combination of the Contribution with any Work.

7.4 4. Your Representation and Warranties.

By submitting a Contribution, you represent and warrant that: (a) each Contribution you submit is an original work and you can legally grant the rights set out in this Agreement; (b) the Contribution does not, and any exercise of the rights granted by you will not, infringe any third party's intellectual property or other right; and (c) you are not aware of any claims, suits, or actions pertaining to the Contribution. You will notify us immediately if you become aware or have reason to believe that any of your representations and warranties is or becomes inaccurate.

7.5 5. Intellectual Property

Except for the assignment and licenses set forth in this Agreement, this Agreement does not transfer any right, title or interest in any intellectual property right of either party to the other. If you choose to provide us with suggestions, ideas for improvement, recommendations or other feedback, on any Work we may use your feedback without any restriction or payment.

7.6 Miscellaneous

Russian law governs this Agreement, excluding any applicable conflict of laws rules or principles, and the parties agree to the exclusive jurisdiction of the courts in Russian Federation. This Agreement does not create a partnership, agency relationship, or joint venture between the parties. We may assign this Agreement without notice or restriction. If any provision of this Agreement is unenforceable, that provision will be modified to render it enforceable to the extent possible to effect the parties' intention and the remaining provisions will not be affected. The parties may amend this Agreement only in a written amendment signed by both parties. This Agreement comprises the parties' entire agreement relating to the subject matter of this Agreement.

Agreed and accepted on my behalf and on behalf of my organization

Our contributor agreement is based on the mongoDB contributor agreement: <https://www.mongodb.com/legal/contributor-agreement>.

HTTP ROUTING TABLE

/community_template

GET /community_template/, 107
GET /community_template/{id}/, 108
POST /community_template/{id}/use_it/, 108

/group

GET /group/, 108
GET /group/{id}/, 109
GET /group/{id}/groups/, 111
GET /group/{id}/groups/{groups_id}/, 112
GET /group/{id}/groups/{groups_id}/hosts/, 114
GET /group/{id}/groups/{groups_id}/hosts/{hosts_id}/, 115
GET /group/{id}/groups/{groups_id}/hosts/{hosts_id}/variables/, 117
GET /group/{id}/groups/{groups_id}/hosts/{hosts_id}/variables/{variables_id}/, 118
GET /group/{id}/groups/{groups_id}/variables/, 120
GET /group/{id}/groups/{groups_id}/variables/{variables_id}/, 121
GET /group/{id}/hosts/, 123
GET /group/{id}/hosts/{hosts_id}/, 125
GET /group/{id}/hosts/{hosts_id}/variables/, 128
GET /group/{id}/hosts/{hosts_id}/variables/{variables_id}/, 130
GET /group/{id}/variables/, 132
GET /group/{id}/variables/{variables_id}/, 134
POST /group/, 109
POST /group/{id}/copy/, 111
POST /group/{id}/groups/, 112
POST /group/{id}/groups/{groups_id}/copy/, 113
POST /group/{id}/groups/{groups_id}/hosts/, 115
POST /group/{id}/groups/{groups_id}/hosts/{hosts_id}/copy/, 117
POST /group/{id}/groups/{groups_id}/hosts/{hosts_id}/variables/, 118
POST /group/{id}/groups/{groups_id}/variables/, 121
POST /group/{id}/hosts/, 124

POST /group/{id}/hosts/{hosts_id}/copy/, 127
POST /group/{id}/hosts/{hosts_id}/variables/, 129
POST /group/{id}/variables/, 133
PUT /group/{id}/, 109
PUT /group/{id}/groups/{groups_id}/, 112
PUT /group/{id}/groups/{groups_id}/hosts/{hosts_id}/, 115
PUT /group/{id}/groups/{groups_id}/hosts/{hosts_id}/variables/, 119
PUT /group/{id}/groups/{groups_id}/variables/{variables_id}/, 122
PUT /group/{id}/hosts/{hosts_id}/, 125
PUT /group/{id}/hosts/{hosts_id}/variables/{variables_id}/, 130
PUT /group/{id}/variables/{variables_id}/, 135
DELETE /group/{id}/, 110
DELETE /group/{id}/groups/{groups_id}/, 113
DELETE /group/{id}/groups/{groups_id}/hosts/{hosts_id}/, 116
DELETE /group/{id}/groups/{groups_id}/hosts/{hosts_id}/variables/, 120
DELETE /group/{id}/groups/{groups_id}/variables/{variables_id}/, 123
DELETE /group/{id}/hosts/{hosts_id}/, 127
DELETE /group/{id}/hosts/{hosts_id}/variables/{variables_id}/, 132
DELETE /group/{id}/variables/{variables_id}/, 136
PATCH /group/{id}/, 110
PATCH /group/{id}/groups/{groups_id}/, 113
PATCH /group/{id}/groups/{groups_id}/hosts/{hosts_id}/, 116
PATCH /group/{id}/groups/{groups_id}/hosts/{hosts_id}/set_owner/, 117
PATCH /group/{id}/groups/{groups_id}/hosts/{hosts_id}/variables/, 119
PATCH /group/{id}/groups/{groups_id}/variables/{variables_id}/, 120
PATCH /group/{id}/hosts/{hosts_id}/, 126
PATCH /group/{id}/hosts/{hosts_id}/set_owner/, 128

PATCH /group/{id}/hosts/{hosts_id}/variables/{variables_id}/,	131	GET /inventory/{id}/group/{group_id}/groups/{groups_id}/,	161
PATCH /group/{id}/set_owner/,	132	GET /inventory/{id}/group/{group_id}/groups/{groups_id}/,	162
PATCH /group/{id}/variables/{variables_id}/,	135	GET /inventory/{id}/group/{group_id}/groups/{groups_id}/,	164
/history		GET /inventory/{id}/group/{group_id}/groups/{groups_id}/,	165
GET /history/,	137	GET /inventory/{id}/group/{group_id}/groups/{groups_id}/,	167
GET /history/{id}/,	137	GET /inventory/{id}/group/{group_id}/groups/{groups_id}/,	169
GET /history/{id}/facts/,	139	GET /inventory/{id}/group/{group_id}/hosts/,	170
DELETE /history/{id}/,	138	GET /inventory/{id}/group/{group_id}/hosts/{hosts_id}/,	171
DELETE /history/{id}/clear/,	138	GET /inventory/{id}/group/{group_id}/hosts/{hosts_id}/va	174
PATCH /history/{id}/cancel/,	138	GET /inventory/{id}/group/{group_id}/hosts/{hosts_id}/va	175
/hook		GET /inventory/{id}/group/{group_id}/variables/,	176
GET /hook/,	139	GET /inventory/{id}/group/{group_id}/variables/{variable	177
GET /hook/{id}/,	140	GET /inventory/{id}/hosts/,	179
POST /hook/,	139	GET /inventory/{id}/hosts/{hosts_id}/,	180
PUT /hook/{id}/,	140	GET /inventory/{id}/hosts/{hosts_id}/variables/,	184
DELETE /hook/{id}/,	141	GET /inventory/{id}/hosts/{hosts_id}/variables/{variable	185
PATCH /hook/{id}/,	141	GET /inventory/{id}/state/,	188
/host		GET /inventory/{id}/variables/,	189
GET /host/,	142	GET /inventory/{id}/variables/{variables_id}/,	190
GET /host/{id}/,	143	POST /inventory/,	149
GET /host/{id}/variables/,	145	POST /inventory/import_inventory/,	150
GET /host/{id}/variables/{variables_id}/,	146	POST /inventory/{id}/copy/,	154
POST /host/,	142	POST /inventory/{id}/group/,	155
POST /host/{id}/copy/,	144	POST /inventory/{id}/group/{group_id}/copy/,	157
POST /host/{id}/variables/,	145	POST /inventory/{id}/group/{group_id}/groups/,	158
PUT /host/{id}/,	143	POST /inventory/{id}/group/{group_id}/groups/{groups_id}	160
PUT /host/{id}/variables/{variables_id}/,	147	POST /inventory/{id}/group/{group_id}/groups/{groups_id}	161
DELETE /host/{id}/,	144	POST /inventory/{id}/group/{group_id}/groups/{groups_id}	163
DELETE /host/{id}/variables/{variables_id}/,	148	POST /inventory/{id}/group/{group_id}/groups/{groups_id}	165
PATCH /host/{id}/,	143	POST /inventory/{id}/group/{group_id}/groups/{groups_id}	168
PATCH /host/{id}/set_owner/,	144	POST /inventory/{id}/group/{group_id}/hosts/,	171
PATCH /host/{id}/variables/{variables_id}/,	147	POST /inventory/{id}/group/{group_id}/hosts/{hosts_id}/c	173
/inventory		POST /inventory/{id}/group/{group_id}/hosts/{hosts_id}/v	174
GET /inventory/,	149		
GET /inventory/{id}/,	150		
GET /inventory/{id}/all_groups/,	152		
GET /inventory/{id}/all_groups/{all_groups_id}/,	152		
GET /inventory/{id}/all_hosts/,	152		
GET /inventory/{id}/all_hosts/{all_hosts_id}/,	153		
GET /inventory/{id}/group/,	154		
GET /inventory/{id}/group/{group_id}/,	155		
GET /inventory/{id}/group/{group_id}/groups/,	157		
GET /inventory/{id}/group/{group_id}/groups/{groups_id}/,	158		

POST /inventory/{id}/group/{group_id}/variables/{variables_id}/, 177	PATCH /inventory/{id}/group/{group_id}/groups/{groups_id}/, 163
POST /inventory/{id}/hosts/, 180	PATCH /inventory/{id}/group/{group_id}/groups/{groups_id}/, 163
POST /inventory/{id}/hosts/{hosts_id}/copy/, 183	PATCH /inventory/{id}/group/{group_id}/groups/{groups_id}/, 164
POST /inventory/{id}/hosts/{hosts_id}/variables/, 185	PATCH /inventory/{id}/group/{group_id}/groups/{groups_id}/, 166
POST /inventory/{id}/variables/, 190	PATCH /inventory/{id}/group/{group_id}/groups/{groups_id}/, 167
PUT /inventory/{id}/, 150	PATCH /inventory/{id}/group/{group_id}/groups/{groups_id}/, 169
PUT /inventory/{id}/group/{group_id}/, 156	PATCH /inventory/{id}/group/{group_id}/hosts/{hosts_id}/, 172
PUT /inventory/{id}/group/{group_id}/groups/{groups_id}/, 159	PATCH /inventory/{id}/group/{group_id}/hosts/{hosts_id}/, 173
PUT /inventory/{id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/, 162	PATCH /inventory/{id}/group/{group_id}/variables/{variables_id}/, 175
PUT /inventory/{id}/group/{group_id}/groups/{groups_id}/variables/{variables_id}/, 166	PATCH /inventory/{id}/group/{group_id}/variables/{variables_id}/set_owner/, 176
PUT /inventory/{id}/group/{group_id}/groups/{groups_id}/variables/{variables_id}/set_owner/, 169	PATCH /inventory/{id}/group/{group_id}/variables/{variables_id}/, 178
PUT /inventory/{id}/group/{group_id}/hosts/{hosts_id}/, 172	PATCH /inventory/{id}/group/{group_id}/variables/{variables_id}/, 182
PUT /inventory/{id}/group/{group_id}/hosts/{hosts_id}/variables/{variables_id}/, 175	PATCH /inventory/{id}/hosts/{hosts_id}/variables/{variables_id}/, 184
PUT /inventory/{id}/group/{group_id}/variables/{variables_id}/, 178	PATCH /inventory/{id}/set_owner/, 188
PUT /inventory/{id}/hosts/{hosts_id}/, 181	PATCH /inventory/{id}/variables/{variables_id}/, 192
PUT /inventory/{id}/hosts/{hosts_id}/variables/{variables_id}/, 186	
PUT /inventory/{id}/state/, 189	
PUT /inventory/{id}/variables/{variables_id}/, 191	
DELETE /inventory/{id}/, 151	/project
DELETE /inventory/{id}/group/{group_id}/, 157	GET /project/, 193
DELETE /inventory/{id}/group/{group_id}/groups/{groups_id}/, 160	GET /project/{id}/, 194
DELETE /inventory/{id}/group/{group_id}/groups/{groups_id}/_project_template_options/, 163	GET /project/{id}/_project_template_options/, 196
DELETE /inventory/{id}/group/{group_id}/groups/{groups_id}/_project_template_options/{_project_template_options_id}/, 167	GET /project/{id}/_project_template_options/{_project_template_options_id}/, 197
DELETE /inventory/{id}/group/{group_id}/groups/{groups_id}/ansible_modules/{ansible_modules_id}/, 170	GET /project/{id}/ansible_modules/{ansible_modules_id}/, 198
DELETE /inventory/{id}/group/{group_id}/groups/{groups_id}/variables/{variables_id}/, 173	GET /project/{id}/ansible_playbooks/, 199
DELETE /inventory/{id}/group/{group_id}/hosts/{hosts_id}/, 176	GET /project/{id}/ansible_playbooks/{ansible_playbooks_id}/, 202
DELETE /inventory/{id}/group/{group_id}/hosts/{hosts_id}/variables/{variables_id}/, 178	GET /project/{id}/execution_templates/{execution_templates_id}/, 203
DELETE /inventory/{id}/hosts/{hosts_id}/, 183	GET /project/{id}/execution_templates/{execution_templates_id}/, 206
DELETE /inventory/{id}/hosts/{hosts_id}/variables/{variables_id}/, 188	GET /project/{id}/execution_templates/{execution_templates_id}/, 208
DELETE /inventory/{id}/variables/{variables_id}/, 192	GET /project/{id}/execution_templates/{execution_templates_id}/, 210
PATCH /inventory/{id}/, 151	GET /project/{id}/execution_templates/{execution_templates_id}/, 210
PATCH /inventory/{id}/group/{group_id}/, 156	GET /project/{id}/execution_templates/{execution_templates_id}/, 212
PATCH /inventory/{id}/group/{group_id}/groups/{groups_id}/, 159	GET /project/{id}/execution_templates/{execution_templates_id}/, 215

HTTP Routing Table

POST /project/{id}/inventory/{inventory_id}/hosts/{hosts_id}/variables/, 259	DELETE /project/{id}/inventory/{inventory_id}/group/{group_id}/, 234
POST /project/{id}/inventory/{inventory_id}/variables/, 262	DELETE /project/{id}/inventory/{inventory_id}/group/{group_id}/, 234
POST /project/{id}/variables/, 265	238
PUT /project/{id}/, 194	DELETE /project/{id}/inventory/{inventory_id}/group/{group_id}/, 234
PUT /project/{id}/execution_templates/{execution_templates_id}/, 204	DELETE /project/{id}/inventory/{inventory_id}/group/{group_id}/, 234
PUT /project/{id}/execution_templates/{execution_templates_id}/options/{options_id}/, 212	DELETE /project/{id}/inventory/{inventory_id}/group/{group_id}/, 234
PUT /project/{id}/execution_templates/{execution_templates_id}/options/{options_id}/periodic_tasks/{periodic_tasks_id}/, 218	DELETE /project/{id}/inventory/{inventory_id}/group/{group_id}/, 234
PUT /project/{id}/inventory/{inventory_id}/, 225	252
PUT /project/{id}/inventory/{inventory_id}/group/{group_id}/, 230	DELETE /project/{id}/inventory/{inventory_id}/group/{group_id}/, 234
PUT /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/, 233	DELETE /project/{id}/inventory/{inventory_id}/hosts/{hosts_id}/, 234
PUT /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/, 237	DELETE /project/{id}/inventory/{inventory_id}/variables/{variables_id}/, 234
PUT /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/hosts/{hosts_id}/variables/{variables_id}/, 241	DELETE /project/{id}/variables/{variables_id}/, 234
PUT /project/{id}/inventory/{inventory_id}/group/{group_id}/groups/{groups_id}/variables/{variables_id}/, 244	PATCH /project/{id}/, 195
PUT /project/{id}/inventory/{inventory_id}/group/{group_id}/execution_templates/{execution_templates_id}/, 247	PATCH /project/{id}/, 195
PUT /project/{id}/inventory/{inventory_id}/group/{group_id}/execution_templates/{execution_templates_id}/variables/{variables_id}/, 251	PATCH /project/{id}/, 195
PUT /project/{id}/inventory/{inventory_id}/group/{group_id}/variables/{variables_id}/, 254	PATCH /project/{id}/, 195
PUT /project/{id}/inventory/{inventory_id}/hosts/{hosts_id}/, 256	PATCH /project/{id}/, 195
PUT /project/{id}/inventory/{inventory_id}/hosts/{hosts_id}/cancel/, 260	PATCH /project/{id}/, 195
PUT /project/{id}/inventory/{inventory_id}/status/{status_id}/, 261	PATCH /project/{id}/, 195
PUT /project/{id}/inventory/{inventory_id}/variables/{variables_id}/, 263	PATCH /project/{id}/, 195
PUT /project/{id}/variables/{variables_id}/, 266	PATCH /project/{id}/, 195
DELETE /project/{id}/, 196	PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/, 234
DELETE /project/{id}/execution_templates/{execution_templates_id}/, 205	PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/, 234
DELETE /project/{id}/execution_templates/{execution_templates_id}/history/{history_id}/, 209	PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/, 234
DELETE /project/{id}/execution_templates/{execution_templates_id}/history/{history_id}/clear/, 209	PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/, 234
DELETE /project/{id}/execution_templates/{execution_templates_id}/options/{options_id}/, 214	PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/, 234
DELETE /project/{id}/execution_templates/{execution_templates_id}/options/{options_id}/periodic_tasks/{periodic_tasks_id}/, 219	PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/, 234
DELETE /project/{id}/history/{history_id}/, 222	248
DELETE /project/{id}/history/{history_id}/clear/, 223	249
DELETE /project/{id}/inventory/{inventory_id}/, 226	251
DELETE /project/{id}/inventory/{inventory_id}/group/{group_id}/, 230	PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/, 234

```
PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/variables/{variables_id}/,
254
PATCH /project/{id}/inventory/{inventory_id}/hosts/{hosts_id}/,
257
PATCH /project/{id}/inventory/{inventory_id}/hosts/{hosts_id}/set_owner/,
258
PATCH /project/{id}/inventory/{inventory_id}/hosts/{hosts_id}/variables/{variables_id}/,
260
PATCH /project/{id}/inventory/{inventory_id}/set_owner/,
261
PATCH /project/{id}/inventory/{inventory_id}/variables/{variables_id}/,
263
PATCH /project/{id}/set_owner/, 264
PATCH /project/{id}/sync/, 264
PATCH /project/{id}/variables/{variables_id}/,
266
```

/stats

```
GET /stats/, 267
```

/user

```
GET /user/, 267
GET /user/{id}/, 268
GET /user/{id}/_settings/, 270
GET /user/{id}/twofa/, 271
POST /user/, 268
POST /user/{id}/change_password/, 271
POST /user/{id}/copy/, 271
PUT /user/{id}/, 269
PUT /user/{id}/_settings/, 270
PUT /user/{id}/twofa/, 272
DELETE /user/{id}/, 270
PATCH /user/{id}/, 269
```


Symbols

`_get_serializer_fields()` (polemarch.plugins.execution.base.BasePlugin method), 101

`_get_serializer_fields()` (polemarch.plugins.inventory.base.BasePlugin method), 103

`_get_serializer_import_fields()` (polemarch.plugins.inventory.base.BasePlugin method), 103

`_get_serializer_import_metaclass()` (polemarch.plugins.inventory.base.BasePlugin method), 104

`_get_serializer_metaclass()` (polemarch.plugins.execution.base.BasePlugin method), 101

`_get_serializer_metaclass()` (polemarch.plugins.inventory.base.BasePlugin method), 104

`_process_arg()` (polemarch.plugins.execution.base.BasePlugin method), 101

A

`arg_shown_on_history_as_inventory` (polemarch.plugins.execution.base.BasePlugin attribute), 101

`arg_shown_on_history_as_mode` (polemarch.plugins.execution.base.BasePlugin attribute), 101

B

`base_command` (polemarch.plugins.execution.base.BasePlugin attribute), 101

`BasePlugin` (class in polemarch.plugins.execution.base), 101

`BasePlugin` (class in polemarch.plugins.inventory.base), 103

D

`defaults` (polemarch.plugins.inventory.base.BasePlugin attribute), 104

E

`error_codes` (polemarch.plugins.execution.base.BasePlugin

attribute), 101

G

`get_args()` (polemarch.plugins.execution.base.BasePlugin method), 101

`get_env_vars()` (polemarch.plugins.execution.base.BasePlugin method), 102

`get_execution_data()` (polemarch.plugins.execution.base.BasePlugin method), 102

`get_pre_commands()` (polemarch.plugins.execution.base.BasePlugin method), 102

`get_raw_inventory()` (polemarch.plugins.execution.base.BasePlugin method), 102

`get_raw_inventory()` (polemarch.plugins.inventory.base.BasePlugin class method), 104

`get_serializer_class()` (polemarch.plugins.execution.base.BasePlugin method), 102

`get_serializer_class()` (polemarch.plugins.inventory.base.BasePlugin method), 104

`get_serializer_import_class()` (polemarch.plugins.inventory.base.BasePlugin method), 104

`get_verbose_level()` (polemarch.plugins.execution.base.BasePlugin method), 102

I

`import_inventory()` (polemarch.plugins.inventory.base.BasePlugin class method), 104

N

`name` (polemarch.plugins.execution.base.BasePlugin property), 103

P

`post_execute_hook()` (polemarch.plugins.execution.base.BasePlugin method), 103

`prepare_execution_dir()` (*pole-
march.plugins.execution.base.BasePlugin
method*), [103](#)

R

`render_inventory()` (*pole-
march.plugins.inventory.base.BasePlugin
method*), [104](#)

S

`serializer_fields` (*pole-
march.plugins.execution.base.BasePlugin
attribute*), [103](#)

`serializer_fields` (*pole-
march.plugins.inventory.base.BasePlugin
attribute*), [104](#)

`serializer_import_fields` (*pole-
march.plugins.inventory.base.BasePlugin
attribute*), [105](#)

`state_managed` (*pole-
march.plugins.inventory.base.BasePlugin
attribute*), [105](#)

`supports_import` (*pole-
march.plugins.inventory.base.BasePlugin
attribute*), [105](#)

V

`verbose_output()` (*pole-
march.plugins.execution.base.BasePlugin
method*), [103](#)