
VST Utils Documentation

Release 0.1

VST Consulting

Jan 25, 2021

CONTENTS

1	Quick Start	3
1.1	New application creation	3
1.2	Adding new models to application	6
2	Configuration manual	17
2.1	Introduction	17
2.2	Main settings	17
2.3	Database settings	18
2.4	Cache settings	19
2.5	Locks settings	19
2.6	Session cache settings	19
2.7	Rpc settings	19
2.8	Worker settings	20
2.9	SMTP settings	20
2.10	Web settings	20
2.11	Centrifugo client settings	21
2.12	Production web settings	21
2.13	Configuration options	21
3	Backend API manual	23
3.1	Models	23
3.2	Utils	23
3.3	Web API	23
3.4	Endpoint	24
3.5	Testing Framework	26
4	VST Utils frontend	227
4.1	Frontend src entry points	227
4.2	Association with back-end entities	228
4.3	Model class	228
4.4	QuerySet class	231
4.5	View class	235
4.6	Fields classes	238
5	Redefinition of VST Utils frontend	243
5.1	Signals	243
5.2	Signals available in VST Utils	244
5.3	Class inheritance	250
5.4	Customizing field's Vue component	252
5.5	Customizing field using dynamic format	252

6	VST Utils JavaScript functions	253
6.1	String.format()	253
6.2	String.format_keys()	253
6.3	trim(s)	253
6.4	capitalizeString(string)	253
6.5	sliceLongString(string="", valid_length=100)	254
6.6	isEmptyObject(object)	254
6.7	addCssClassesToElement(element="", title="", type="")	254
6.8	getCookie(name)	254
6.9	deepEqual(x, y)	254
6.10	stringToBoolean(string)	255
6.11	oneCharNumberToTwoChar(n)	255
6.12	allPropertiesIsObjects(obj)	255
6.13	arrayBufferToBase64(buffer)	255
6.14	randomString(length, abc="qwertyuiopasdfghjklzxcvbnm012364489")	255
6.15	hexToRgbA(hex, alpha=1)	256
6.16	getTimeInUptimeFormat(time)	256
6.17	findClosestPath(paths, current_path)	256
6.18	cleanAllCacheAndReloadPage()	256
6.19	cleanOpenApiCacheAndReloadPage()	256
6.20	updateGuiVersionsInLocalStorage()	256
6.21	createDomElement(type, attributes, props)	257
6.22	onLoadingErrorHandler(event)	257
7	Application loading on frontend	259
7.1	Application loading steps	259
7.2	Optimization of application loading	259
7.3	FilesCache class	260
7.4	AppDependenciesLoader class	261
7.5	OpenApiLoader class	263
7.6	StaticFilesLoader class	264
8	Parsing of OpenAPI schema	267
8.1	BaseEntityConstructor class	267
8.2	ModelConstructor class	268
8.3	ViewConstructor class	270
8.4	SubViewWithoutApiPathConstructor class	274
9	Root classes providing application work	277
9.1	App class	277
9.2	ApiConnector class	279
9.3	ErrorHandler class	281
9.4	RouterConstructor class	282
9.5	StoreConstructor class	284
9.6	PopUp class	285
9.7	LocalSettings class	287
9.8	GuiCustomizer class	289
	HTTP Routing Table	293

VST Utils is a small framework for quick generation of single-page applications. The main feature of the VST Utils framework is autogenerated GUI, which is forming based on the OpenAPI schema. OpenAPI schema is a JSON, that contents description of the models used in the REST API and info about all paths of the application.

In this documentation you can find info about QuickStart of new project based on VST Utils, description of base models, views and fields available in the framework, and also you will know how you can redefine some standard models, views and fields in your project.

QUICK START

Starting of new project, based on VST Utils Framework, is rather simple. We recommend creating a virtual environment for each project to avoid conflicts in the system.

Let's learn by example. All you need to do is run several commands. This manual consist of two parts:

1. Description of the process of creating a new application and the main commands for launching and deploying.
2. Description of the process of creating new entities in the application.

1.1 New application creation

Throughout this tutorial, we'll walk you through the creation of a basic poll application.

1. Install VST Utils

```
pip install vstutils
```

2. Create new project, based on VST Utils

If this is your first time using vstutils, you'll have to take care of some initial setup. Namely, you'll need to auto-generate some code that establishes a vstutils application – a collection of settings for an instance of vstutils, including database configuration, Django-specific and vstutils-specific options and application-specific settings. You can execute base command for new project creation.

```
python -m vstutils newproject --name {{app_name}}
```

This command will confirm you such options of new app, as:

- **project name** - name of your new application;
- **project guiname** - name of your new application, that will be used in GUI (web-interface);
- **project directory** - path to directory, where project will be created.

Or you can execute following command, that includes all needed data for new project creation.

```
python -m vstutils newproject --name {{app_name}} --dir {{app_dir}} --  
↳guiname {{app_guiname}} --noinput
```

This command creates new project without confirming any data.

Both these commands create several files in `project` directory, that was mentioned in the new project creation command.

```
/{app_dir}/{app_name}
├── .coveragerc
├── frontend_src
│   ├── app
│   │   └── index
│   ├── .editorconfig
│   ├── .eslintrc.js
│   └── .prettierrc
├── MANIFEST.in
├── package.json
├── .pep8
├── README.rst
├── requirements-test.txt
├── requirements.txt
├── setup.cfg
├── setup.py
├── {{app_name}}
│   ├── __init__.py
│   ├── __main__.py
│   ├── models
│   │   └── __init__.py
│   ├── settings.ini
│   ├── settings.py
│   ├── web.ini
│   └── wsgi.py
├── test.py
├── tox.ini
└── webpack.config.jsdefault
```

where:

- **frontend_src** - directory that contains all sources for frontend;
- **MANIFEST.in** - this file is used for building installation package;
- **{{app_name}}** - directory with files of your application;
- **package.json** - this file contains list of frontend dependencies and commands to build;
- **README.rst** - default README file for your application (this file includes base commands for starting/stopping your application);
- **requirements-test.txt** - file with list of requirements for test environment;
- **requirements.txt** - file with list of requirements for your application;
- **setup.cfg** - this file is used for building installation package;
- **setup.py** - this file is used for building installation package;
- **test.py** - this file is used for tests creation;
- **tox.ini** - this file is used for tests execution;
- **webpack.config.js.default** - this file contain minimal script for webpack (replace 'default' if write smthg in 'app.js').

All following commands you should execute from the `/{app_dir}/{app_name}/` directory.

3. Apply migrations

Let's verify your vstutils project works. Change into the outer `{{app_dir}}/{{app_name}}` directory, if you haven't already, and run the following command:

```
python -m {{app_name}} migrate
```

This command create SQLite (by default) database with default SQL-schema. VSTUTILS supports all databases [Django does](#)¹.

4. Create superuser

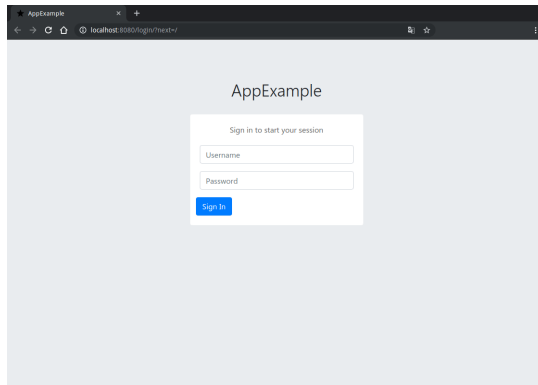
```
python -m {{app_name}} createsuperuser
```

5. Start your application

```
python -m {{app_name}} web
```

Web-interface of your application will be started on the port 8080. You've started the vstutils production server based on [uWSGI](#)².

Warning: Now's a good time to note: if you want to run the web-server under the debugger, then you should run [the standard Django's dev-server](#)³.



If you need to stop your application, use following command:

```
python -m {{app_name}} web stop=/tmp/{{app_name}}_web.pid
```

Current algorithm of new project Quick Start allows you to create the simplest application, based on VST Utils framework. This application will contain only User Model. If you want to create your own models look following section.

¹ <https://docs.djangoproject.com/en/2.2/ref/databases/#databases>

² <https://uwsgi-docs.readthedocs.io/>

³ <https://docs.djangoproject.com/en/2.2/intro/tutorial01/#the-development-server>

1.2 Adding new models to application

If you want to add some new entities to your application, you need to do following on the back-end:

1. Create Model;
2. Create Serializer (optional);
3. Create View (optional);
4. Add created Model or View to the API;
5. Make migrations;
6. Apply migrations;
7. Restart your application.

Let's look how you can do it on the AppExample - application, that has 2 custom models:

- Task (abstraction for some tasks/activities, that user should do);
- Stage (abstraction for some stages, that user should do to complete the task. This model is nested into the Task Model).

1.2.1 Models creation

Firstly, you need to create file `{{model_name}}.py` in the `/{{app_dir}}/{{app_name}}/{{app_name}}/models` directory.

Let make out an example from `BModel`:

More information about Models you can find in [Django Models documentation](#)⁴.

If you don't need to create custom *serializers* or *view sets*, you can go to this *sstage*.

1.2.2 Serializers creation

Note - If you don't need custom serializer you can skip this section

Firstly, you need to create file `serializers.py` in the `/{{app_dir}}/{{app_name}}/{{app_name}}/` directory.

Then you need to add some code like this to `serializers.py`:

```
from datetime import datetime
from vstutils.api import serializers as vst_serializers
from . import models as models

class StageSerializer(models.Stage.generated_view.serializer_class):

    class Meta:
        model = models.Stage
        fields = ('id',
                  'name',
                  'order',)

    def update(self, instance, validated_data):
```

(continues on next page)

⁴ <https://docs.djangoproject.com/en/2.2/topics/db/models/>

(continued from previous page)

```
# Put custom logic to serializer update
instance.last_update = datetime.utcnow()
super().update(instance, validated_data)
```

More information about Serializers you can find in [Django REST Framework documentation for Serializers](#)⁵.

1.2.3 Views creation

Note - If you don't need custom view set you can skip this section

Firstly, you need to create file `views.py` in the `{{ app_dir }}/{{ app_name }}/{{ app_name }}/` directory.

Then you need to add some code like this to `views.py`:

```
from vstutils.api import decorators as deco
from vstutils.api.base import ModelViewSet
from . import serializers as sers
from .models import Stage, Task

class StageViewSet(Stage.generated_view):
    serializer_class_one = sers.StageSerializer

    '''
    Decorator, that allows to put one view into another
    * 'tasks' - suburl for nested view
    * 'methods=["get"]' - allowed methods for this view
    * 'manager_name='hosts' - Name of related QuerySet to the child model instances_
    ↳ (we set it in HostGroup model as "hosts = models.ManyToManyField(Host)")
    * 'view=Task.generated_view' - Nested view, that will be child view for_
    ↳ decorated view
    '''
    @nested_view('stage', view=StageViewSet)
    class TaskViewSet(Task.generated_view):
        '''
        Task operations.
        '''
```

More information about Views and ViewSets you can find in [Django REST Framework documentation for views](#)⁶.

1.2.4 Adding Models to API

To add created Models to the API you need to write something like this at the end of your `settings.py` file:

```
'''
Some code generated by VST Utils
'''

'''
Adds Task view set to the API
Only 'root' (parent) views should be added there.
```

(continues on next page)

⁵ <https://www.django-rest-framework.org/api-guide/serializers/#modelserializer>

⁶ <https://www.django-rest-framework.org/api-guide/viewsets/>

(continued from previous page)

```
Nested views will be added automatically, that's why we add there only Task view.
Stage view will be added automatically, because it is nested to the Task view.
'''
API[VST_API_VERSION][r'task'] = {
    'view': 'newapp2.views.TaskViewSet'
}

'''
You can add model too.
All model generate base ViewSet with data that they have, if dont create custom_
↳ ViewSet or Serializer
'''
API[VST_API_VERSION][r'task'] = dict(
    model='newapp2.models.Task'
)

# Adds link to the task view to the GUI menu
PROJECT_GUI_MENU.insert(0, {
    'name': 'Task',
    # CSS class of font-awesome icon
    'span_class': 'fa fa-list-alt',
    'url': '/task'
})
```

1.2.5 Migrations creation

To make migrations you need to open `{{app_dir}}/{{app_name}}/` directory and execute following command:

```
python -m {{app_name}} makemigrations {{app_name}}
```

More information about Migrations you can find in [Django Migrations documentation](https://docs.djangoproject.com/en/2.2/topics/migrations/)⁷.

1.2.6 Migrations applying

To apply migrations you need to open `{{app_dir}}/{{app_name}}/` directory and execute following command:

```
python -m {{app_name}} migrate
```

1.2.7 Restart of Application

To restart your application, firstly, you need to stop it (if it was started before):

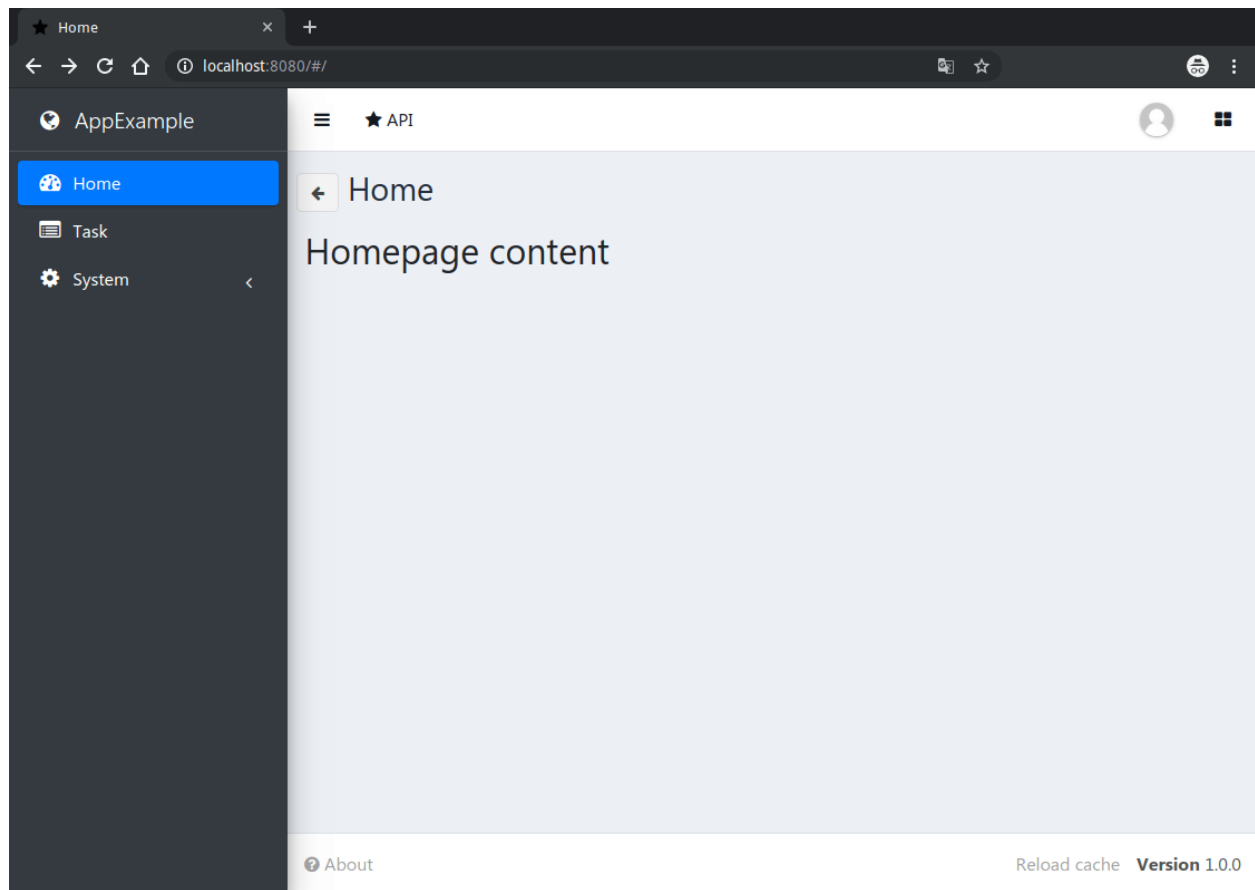
```
python -m {{app_name}} web stop=/tmp/{{app_name}}_web.pid
```

And then start it again:

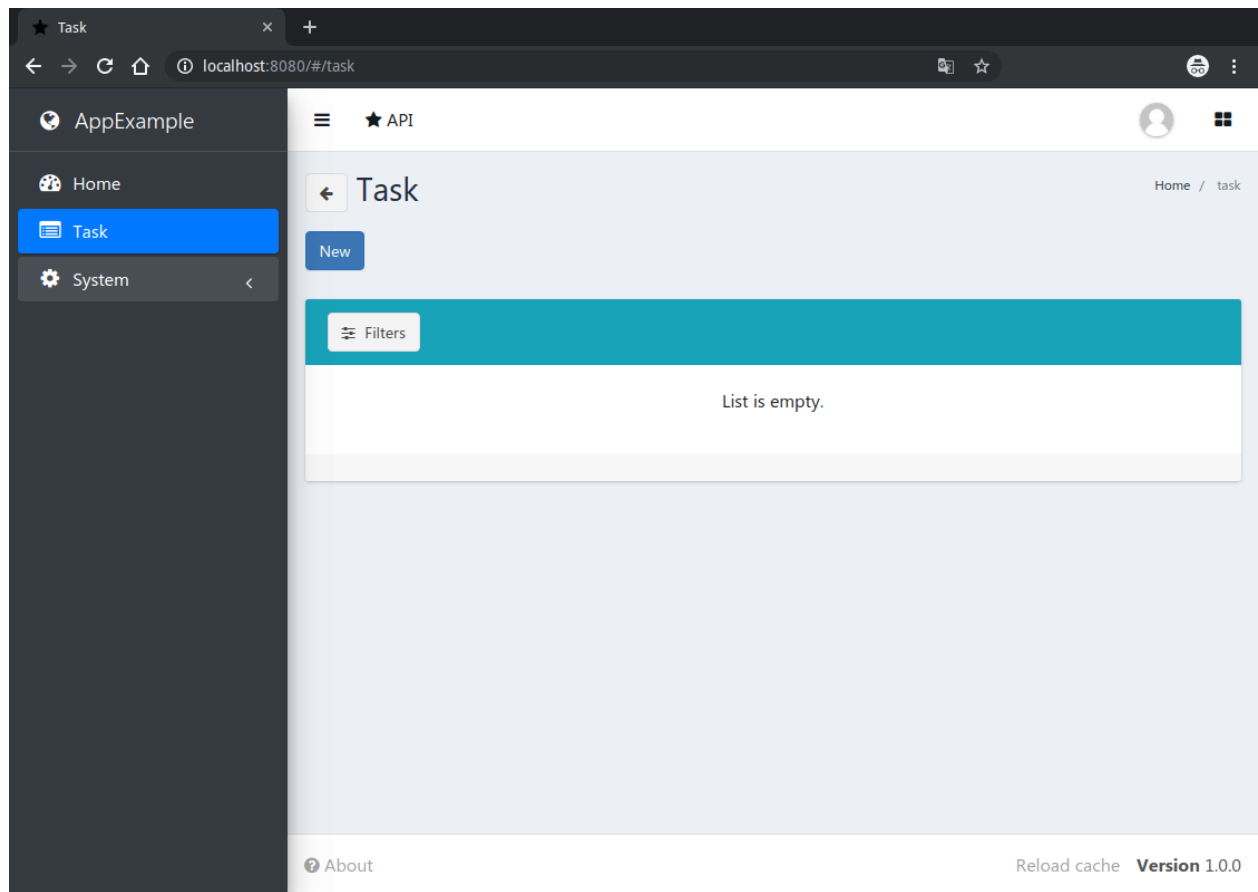
```
python -m {{app_name}} web
```

⁷ <https://docs.djangoproject.com/en/2.2/topics/migrations/>

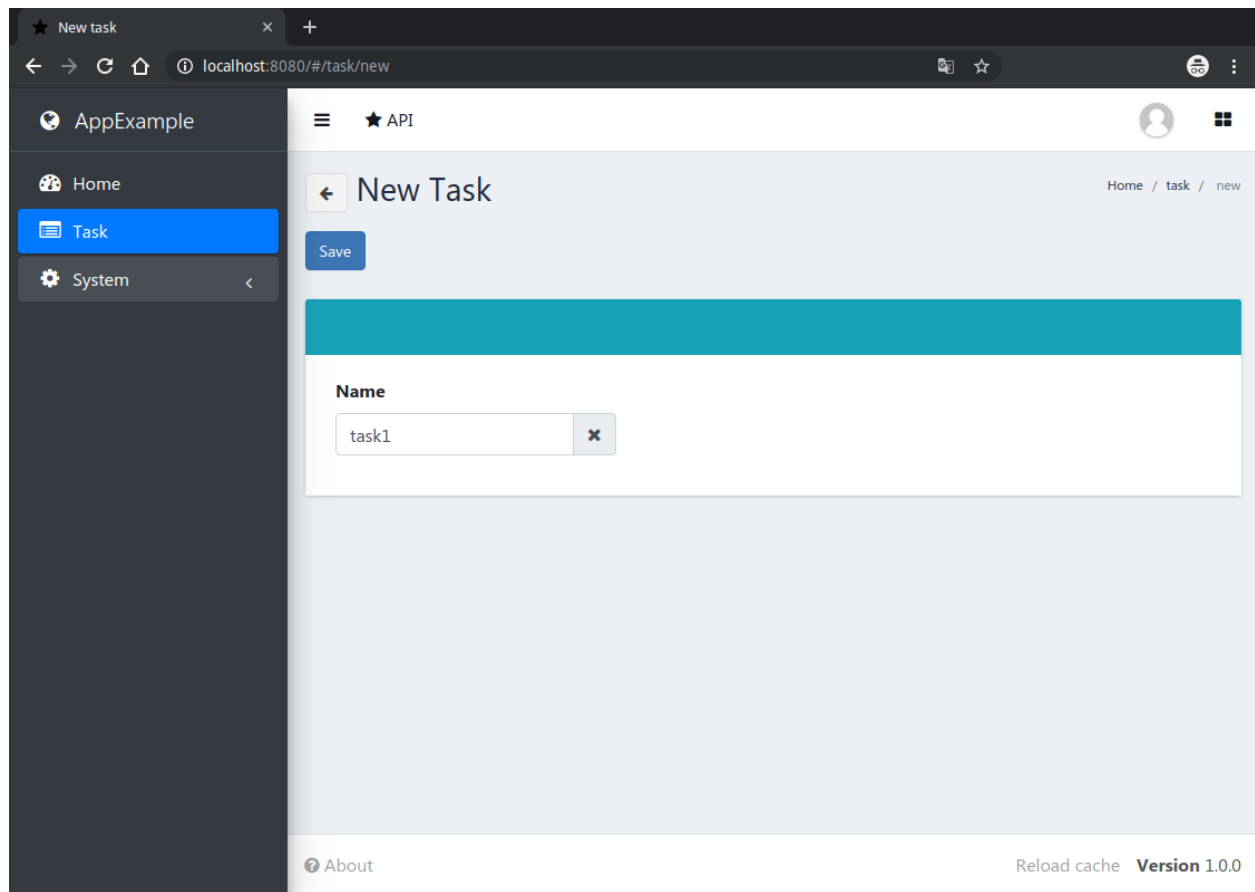
After cache reloading you will see following page:



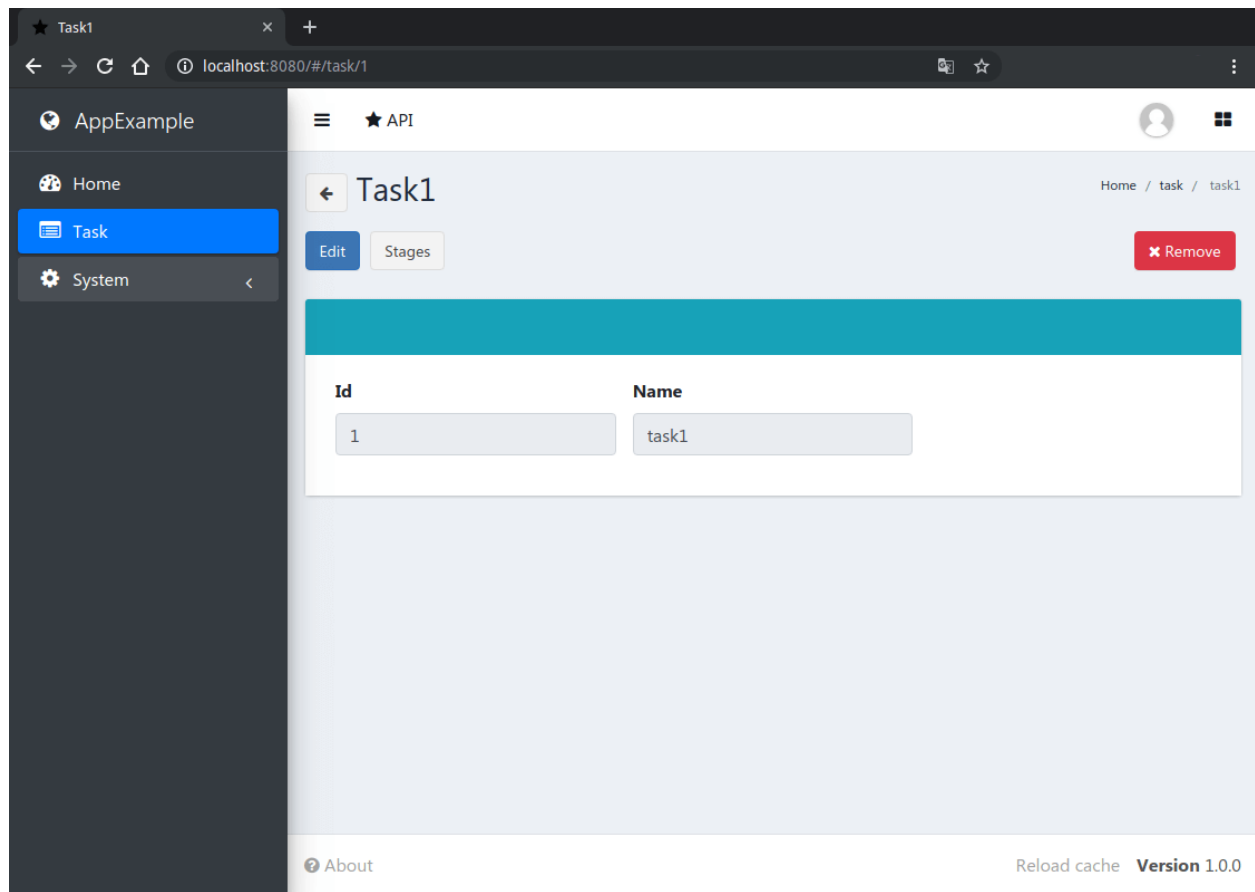
As you can see, link to new Task View was added to the sidebar menu. Let's click on it.



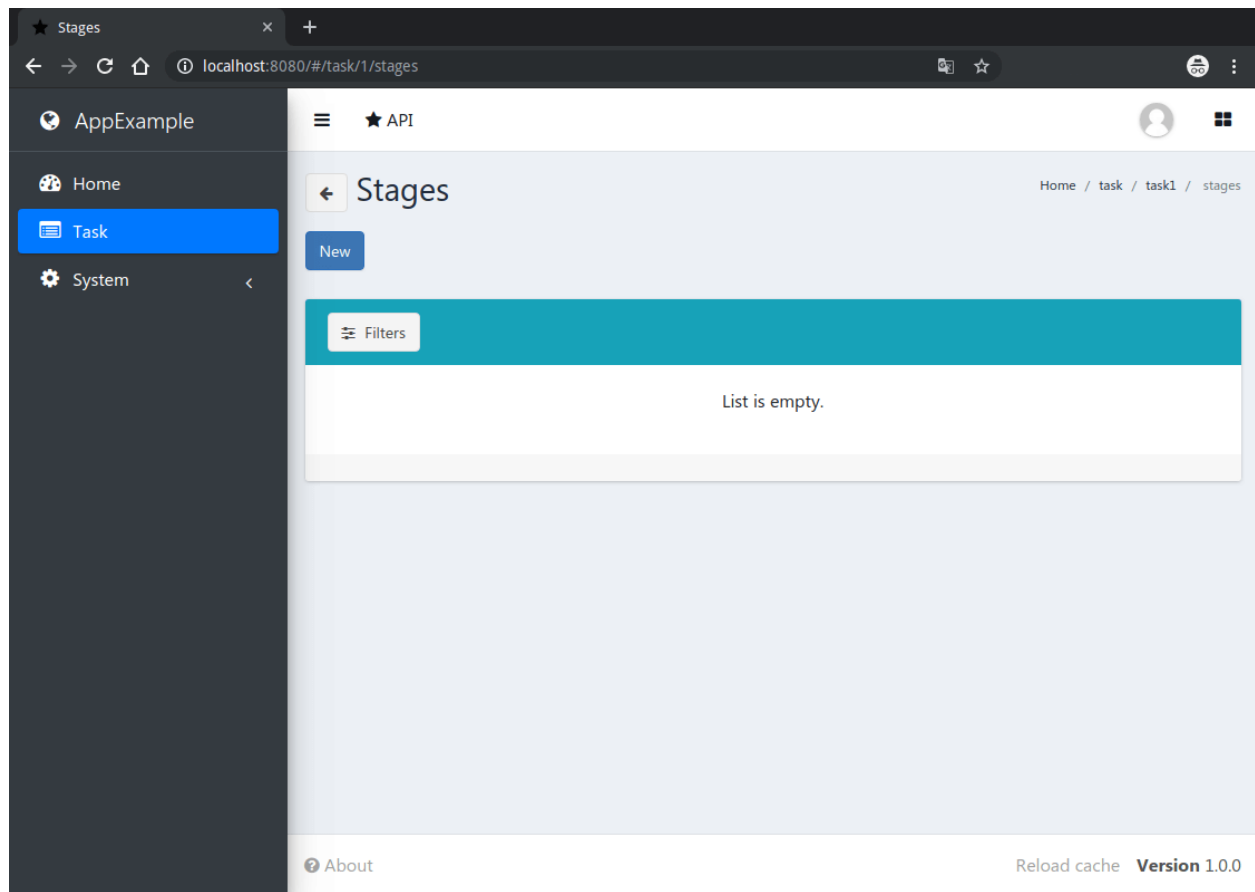
As you can see, there is no task instance in your app. Let's click on 'new' button.



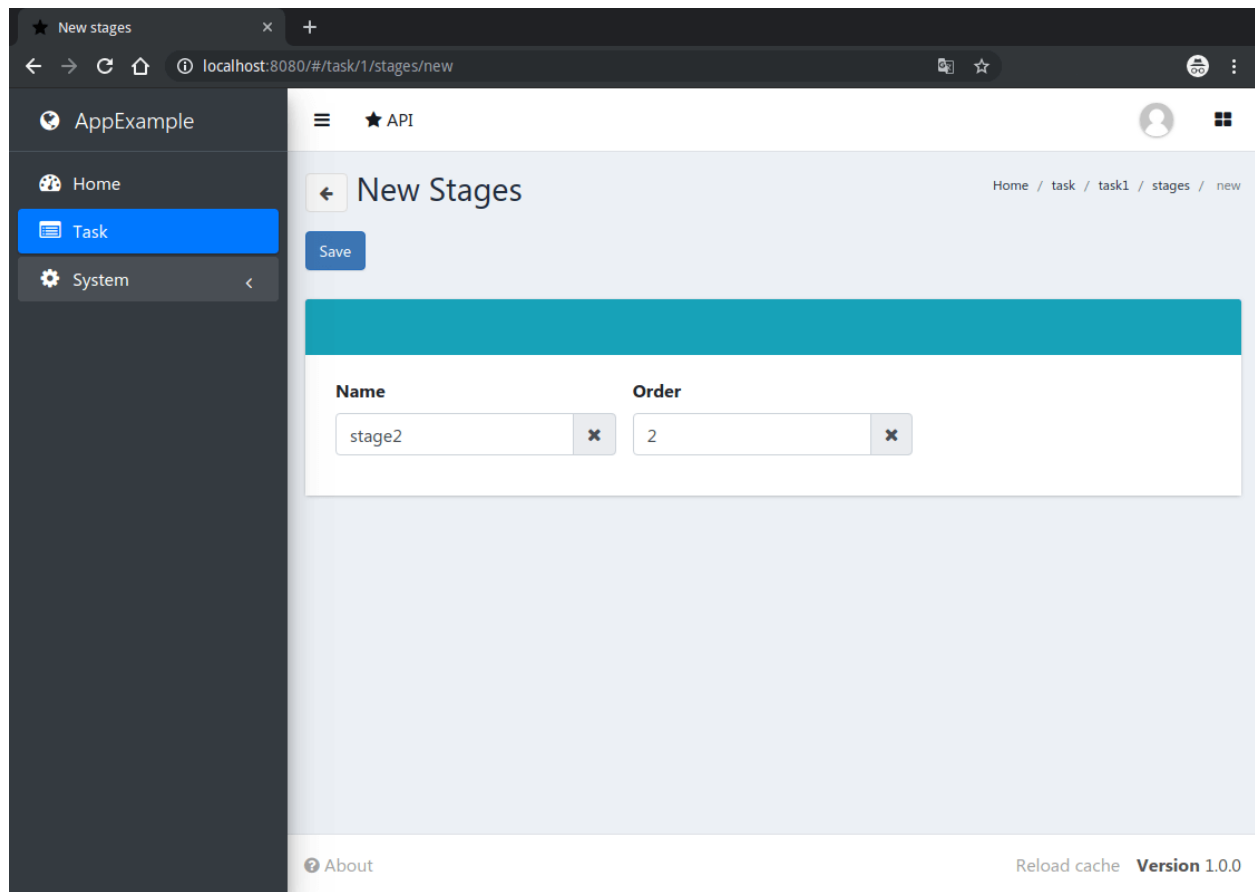
After new task creation you will see following page:

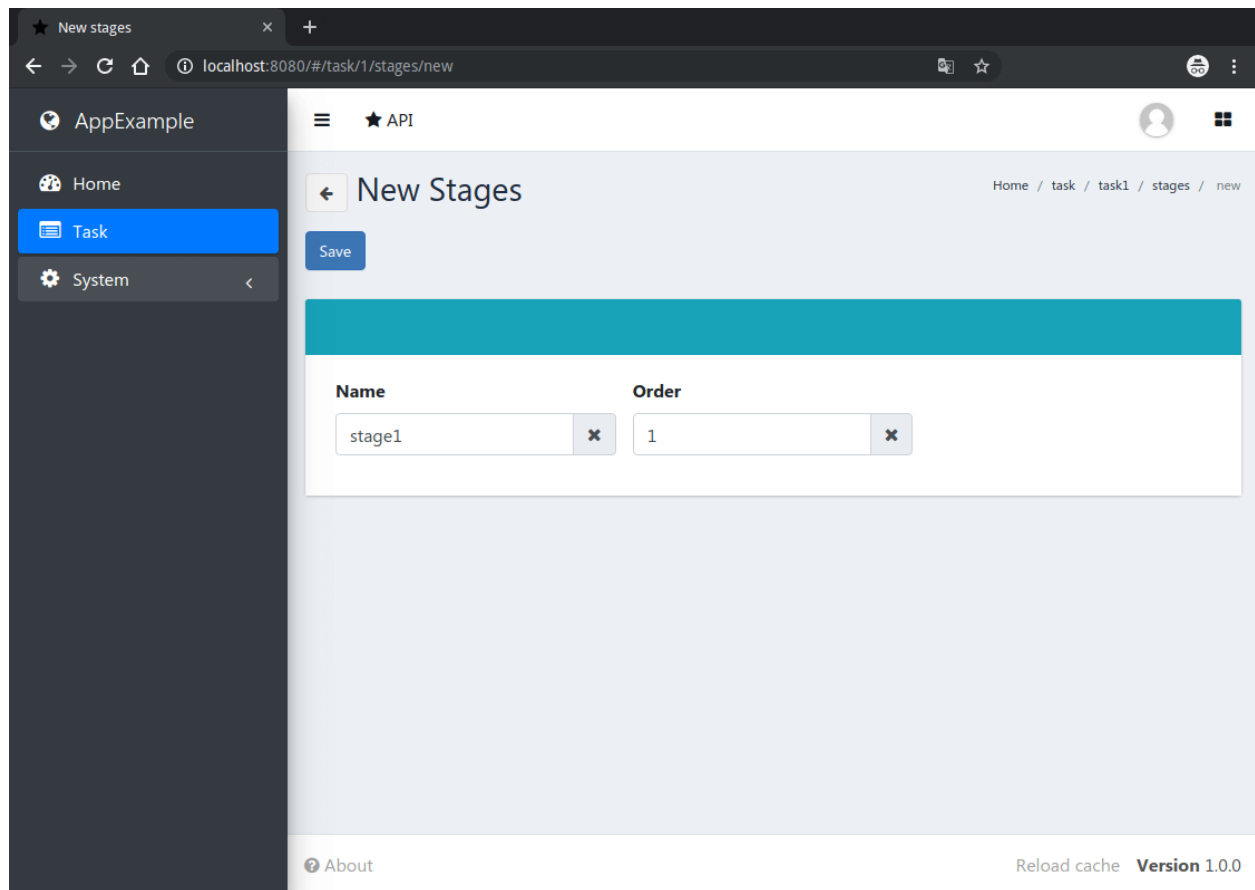


As you can see, there is 'stages' button, that opens page with this task's stages list. Let's click on it.

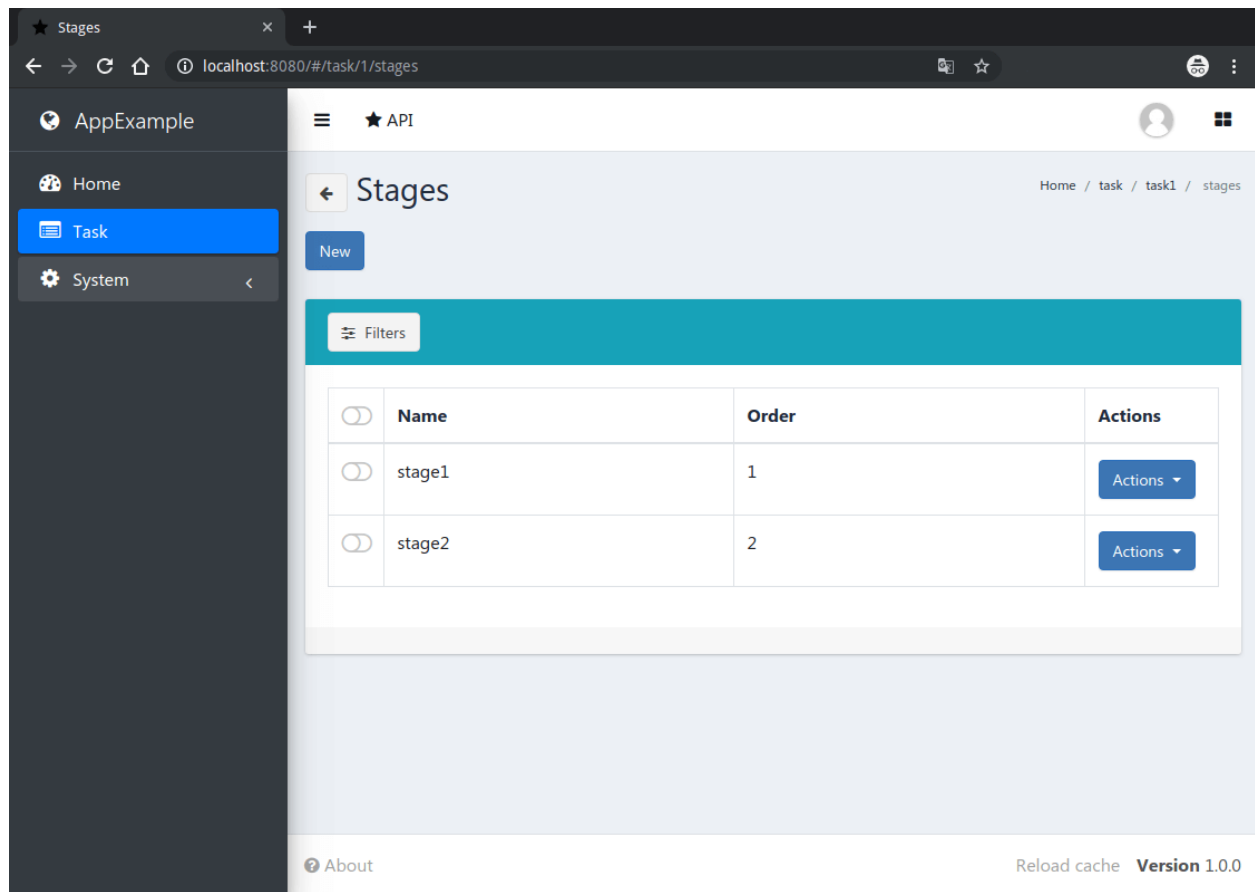


As you can see, there is no stage instance in your app. Let's create 2 new stages.





After stages creation page with stages list will looks like this:



As you can see, sorting by 'order' field is working, as we mentioned in the our `models.py` file for Stage Model.

Additional information about Django and Django REST Framework you can find in [Django documentation](https://docs.djangoproject.com/en/2.2/)⁸ and [Django REST Framework documentation](https://www.django-rest-framework.org/)⁹.

⁸ <https://docs.djangoproject.com/en/2.2/>

⁹ <https://www.django-rest-framework.org/>

CONFIGURATION MANUAL

2.1 Introduction

Though default configuration is suitable for many common cases, vstutils-based applications is highly configurable system. If you need something more advanced (scalability, dedicated DB, custom cache, logging or directories) you can configure vstutils-based application deeply by tweaking `/etc/{{app_name or app_lib_name}}/settings.ini`.

The most important thing to keep in mind when planning your application architecture is that vstutils-based applications have a service-oriented structure. To build a distributed scalable system you only need to connect to a shared *database*, shared *cache*, *locks* and a shared *rpc* service (MQ such as RabbitMQ, Redis, etc.). A shared file storage may be required in some cases, a but it isn't required by vstutils.

Let's consider the main sections of the config and its parameters:

2.2 Main settings

Section `[main]`.

This section is intended for settings related to whole vstutils-based application (both worker and web). Here you can specify verbosity level of vstutils-based application during work, which can be useful for troubleshooting (logging level etc). Also there are settings for changing of timezone for whole app and allowed domains.

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. 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: *.
- **first_day_of_week** - Integer value with first day of week. Default: 0.
- **ldap-server** - LDAP server connection.
- **ldap-default-domain** - Default domain for auth.
- **ldap-auth_format** - Default search request format for auth. Default: cn=<username>, <domain>.
- **timezone** - Timezone of web-application. Default: UTC.
- **log_level** - Logging level. Default: WARNING.
- **enable_admin_panel** - Enable or disable Django Admin panel. Default: false.

2.3 Database settings

Section [database].

Here you can change settings related to database system, which vstutils-based application will use. vstutils-based application 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](#)¹⁰. If you run vstutils-based application 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 work of MariaDB 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]
character-set-server=utf8
collation-server=utf8_unicode_ci
```

¹⁰ <https://docs.djangoproject.com/en/2.22/ref/settings/#databases>

2.4 Cache settings

Section `[cache]`.

This section is for settings related to cache backend used by vstutils-based application. vstutils-based application 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](#)¹¹. 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.

2.5 Locks settings

Section `[locks]`.

Locks is system that vstutils-based application 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 vstutils-based application 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.

2.6 Session cache settings

Section `[session]`.

vstutils-based application store sessions in `database`, 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`.

2.7 Rpc settings

Section `[rpc]`.

vstutils-based application uses Celery for long-running async tasks. 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.

This section actual only with vstutils installed with `rpc` extra dependency.

- **connection** - Celery broker connection. Read more: <http://docs.celeryproject.org/en/latest/userguide/configuration.html#conf-broker-settings> Default: `filesystem:///var/tmp`.
- **concurrency** - Count of celery 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** - Count of retries on project sync operation.

¹¹ <https://docs.djangoproject.com/en/2.2/ref/settings/#caches>

2.8 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/{app_name}_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`.

2.9 SMTP settings

Section `[mail]`.

Django comes with several email sending backends. With the exception of the SMTP backend (which is the default when `host` is set), these backends are only useful during testing and development.

Applications based on `vstutils` uses only `smtp` and `console` backends.

- **host** - IP or domain for smtp-server. If it not set `vstutils` uses `console` backends. Default: `None`.
- **port** - Port for smtp-server connection. Default: `25`.
- **user** - Username for smtp-server connection. Default: `" "`.
- **password** - Auth password for smtp-server connection. Default: `" "`.
- **tls** - Enable/disable `tls` for smtp-server connection. Default: `False`.
- **send_confirmation** - Enable/disable confirmation message after registration. Default: `False`.
- **authenticate_after_registration** - Enable/disable autologin after registration confirmation. Default: `False`.

2.10 Web settings

Section `[web]`.

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

- **allow_cors** - Cross-origin resource sharing enabling. Default: `False`.
- **enable_gravatar** - Enables/disables gravatar service using for users. Default: `True`.
- **rest_swagger_description** - Help string in Swagger schema. Useful for dev-integrations.
- **openapi_cache_timeout** - Cache timeout for storing schema data. Default: `120`.
- **health_throttle_rate** - Count of requests to `/api/health/` endpoint. Default: `60`.
- **bulk_threads** - Threads count for `PATCH /api/endpoint/` endpoint. Default: `3`.
- **session_timeout** - Session life-cycle time. Default: `2w` (two weeks).
- **rest_page_limit** and **page_limit** - Default limit of objects in API list. Default: `1000`.

2.11 Centrifugo client settings

Section `[centrifugo]`.

For installations with centrifugo client, `[centrifugo]` section must be setuped.

- **address** - Centrifugo server address.
- **api_key** - API key for clients.
- **timeout** - Connection timeout.
- **verify** - Connection verification.

2.12 Production web settings

Section `[uwsgi]`.

Here placed settings related to web-server used by vstutils-based application in production (for deb and rpm packages by default). Most of them related to system paths (logging, PID-file and so on). More settings in [uWSGI docs](http://uwsgi-docs.readthedocs.io/en/latest/Configuration.html)¹².

2.13 Configuration options

This section contains additional information for configure additional elements.

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

```
[uwsgi]
addrport = 0.0.0.0:8443,foobar.crt,foobar.key
```

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

¹² <http://uwsgi-docs.readthedocs.io/en/latest/Configuration.html>

BACKEND API MANUAL

VST Utils framework consolidates such frameworks as Django, Django Rest Framework, drf-yasg and Celery. Below are descriptions of some features used in the development of projects based on vstutils.

3.1 Models

A model is the single, definitive source of truth about your data. It contains the essential fields and behaviors of the data you're storing. The goal is to define your data model in one place and automatically derive things from it. You can also define everything you need to get the generated view from the model.

3.1.1 Also you can use custom models without using database:

3.2 Utils

This is some tested set of development utilities. Utilities include a collection of some code that will be useful in one way or another to develop the application. Most of the functions are used by vstutils itself.

3.3 Web API

Web API is based on Django Rest Framework with some nested functions.

3.3.1 Fields

There is additional fields which extends API and GUI. Also, it would be usefull for customizing frontend view without write any JS code.

There is useful validation classes for fields.

3.3.2 Serializers

3.3.3 Views

3.3.4 Filtersets

For greater development convenience, the framework provides additional classes and functions for filtering elements by fields.

3.3.5 Responses

DRF provides a standard set of variables whose names correspond to the human-readable name of the HTTP code. For convenience, we have dynamically wrapped this in a set of classes that have appropriate names and additionally provide the following capabilities:

- String responses are wrapped in json like { "detail": "string response" }.
- Keep attribute timings for further processing in middlewares.
- Set status code from class name (e.g. HTTP_200_OK or Response200 got code 200).

All classes inherit from:

3.3.6 Middlewares

By default, the Django [assumes](#)¹³ that the developer will develop itself Middleware class, but it is not always convenient. The vstutils library offers a convenient request handler class for elegant OOP development. Middlewares is needed to process incoming requests and sent responses before they reach the final destination.

3.4 Endpoint

Endpoint view has two purposes: bulk requests execution and providing openapi schema.

Endpoint url is `/ {API_URL} / endpoint /`, for example value with default settings is `/ api / endpoint /`.

API_URL can be changed in `settings.py`.

3.4.1 Bulk requests

Bulk request allows you send multiple request to api at once, it accepts json list of operations.

Method	Transactional (all operations in one transaction)	Synchronous (operations executed one by one in given order)
PUT <code>/ {API_URL} / endpoint /</code>	NO	YES
POST <code>/ {API_URL} / endpoint /</code>	YES	YES
PATCH <code>/ {API_URL} / endpoint /</code>	NO	NO

¹³ <https://docs.djangoproject.com/en/2.2/topics/http/middleware/#writing-your-own-middleware>

Parameters of one operation (* means that parameter is required):

- `method*` - http method of request
- `path*` - path of request, can be `str` or `list`
- `data` - data that needs to be sent
- `query` - query parameters as `str`
- `headers` - dict with headers which will be sent, names of headers must follow [CGI specification](#)¹⁴ (e.g., `CONTENT_TYPE`, `GATEWAY_INTERFACE`, `HTTP_*`).
- `version` - `str` with specified version of api, if not provided then `VST_API_VERSION` will be used

In any request parameter you can insert result value of previous operations (`<<{OPERATION_NUMBER} [path] [to] [value]>>`), for example:

```
[
  {"method": "post", "path": "user", "data": {"name": "User 1"}},
  {"method": "delete", "version": "v2", "path": ["user", "<<0[data][id]>>"]}
]
```

Result of bulk request is json list of objects for operation:

- `method` - http method
- `path` - path of request, always `str`
- `data` - data that needs to be sent
- `status` - response status code

Transactional bulk request returns 502 `BAG GATEWAY` and make rollback if one of requests is failed.

Warning: If you send non-transactional bulk request, you will get 200 status and must validate statuses on each operation responses.

3.4.2 Openapi schema

Request on `GET /{API_URL}/endpoint/` returns Swagger UI.

Request on `GET /{API_URL}/endpoint/?format=openapi` returns json openapi schema. Also you can specify required version of schema using version query parameter (e.g., `GET /{API_URL}/endpoint/?format=openapi&version=v2`).

Applying hooks to the schema can also be helpful. This functionality will help to change certain data in the schema before it will be send to user. In order to set some hooks, it is enough to specify in `settings.py` the `OPENAPI_HOOKS` which is an array with lines for importing functions. Each function will take 2 named arguments:

- `request` - user request object.
- `schema` - ordered dict with openapi schema.

Note: Sometimes hooks may raise an exception, and in order not to break the chain of data modification, such exceptions are handled. However, the changes made to the schema before the raised exception will be saved.

Example hook:

¹⁴ <https://www.w3.org/CGI/>

```
def hook_add_username_to_guiname(request, schema):
    schema['info']['title'] = f"{request.username} - {schema['info']['title']}"
```

3.5 Testing Framework

VST Utils Framework includes a few helper in base testcase class and improve support for making API requests. That means if you want make bulk request to endpoint you dont need create and init test client, but just need to call:

```
endpoint_results = self.bulk([
    # list of endpoint requests
])
```

3.5.1 Creating test case

After creating new project via `vstutils` you can found `test.py` module, where you see testcase classes based on `vstutils.tests.BaseTestCase`. At the moment, we officially support two styles of writing tests: through classic and simple query wrappers with run check and through runtime optimized bulk queries with manual value checking.

3.5.2 Simple example with classic tests

For example, if you have api endpoint like `/api/v1/project/` and model `Project` you can write testcase like this:

```
from vstutils.tests import BaseTestCase

class ProjectTestCase(BaseTestCase):
    def setUp(self):
        super(ProjectTestCase, self).setUp()
        # init demo project
        self.initial_project = self.get_model_class('project.Test').objects.
        ↪ create(name="Test")

    def tearDown(self):
        super(ProjectTestCase, self).tearDown()
        # remove it after test
        self.initial_project.delete()

    def test_project_endpoint(self):
        # Test checks that api return valid values
        self.list_test('/api/v1/project/', 1)
        self.details_test(
            ["project", self.initial_project.id],
            name=self.initial_project.name
        )
        # Try to create new projects and check list endpoint
        test_data = [
            {"name": f"TestProject{i}"}
            for i in range(2)
        ]
        id_list = self.mass_create("/api/v1/project/", test_data, 'name')
        self.list_test('/api/v1/project/', 1 + len(id_list))
```

This simple example demonstrate functionality of default test case class. Default projects are initialized in such a way that for the fastest and most efficient result it is best to distribute testing of various entities into different classes. This example demonstrate classic style of testing, but you can use bulks in your test cases.

3.5.3 Bulk requests in tests

The bulk query system and its capabilities are very well suited for testing and executing valid queries. Returning to the previous example, it could be rewritten as follows:

```
from vstutils.tests import BaseTestCase

class ProjectTestCase(BaseTestCase):
    def setUp(self):
        super(ProjectTestCase, self).setUp()
        # init demo project
        self.initial_project = self.get_model_class('project.Test').objects.
        ↪ create(name="Test")

    def tearDown(self):
        super(ProjectTestCase, self).tearDown()
        # remove it after test
        self.initial_project.delete()

    def test_project_endpoint(self):
        test_data = [
            {"name": f"TestProject{i}"}
            for i in range(2)
        ]
        bulk_data = [
            {"method": "get", "path": ["project"]},
            {"method": "get", "path": ["project", self.initial_project.id]}
        ]
        bulk_data += [
            {"method": "post", "path": ["project"], "data": i}
            for i in test_data
        ]
        bulk_data.append(
            {"method": "get", "path": ["project"]}
        )
        results = self.bulk_transactional(bulk_data)

        self.assertEqual(results[0]['status'], 200)
        self.assertEqual(results[0]['data']['count'], 1)
        self.assertEqual(results[1]['status'], 200)
        self.assertEqual(results[1]['data']['name'], self.initial_project.name)

        for pos, result in enumerate(results[2:-1]):
            self.assertEqual(result['status'], 201)
            self.assertEqual(result['data']['name'], test_data[pos]['name'])

        self.assertEqual(results[-1]['status'], 200)
        self.assertEqual(results[-1]['data']['count'], 1 + len(test_data))
```

In this case, you have more code rows, but your tests will be closer to GUI workflow, because vstutils-projects uses /api/endpoint/ for requests. Either way, bulk queries are much faster due to some optimizations, so you can reduce testcase execution time.

3.5.4 Test case API

API Paths

GET `/community_template/`

List of community project templates.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/community_template/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "example name",
      "type": "example type"
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*ProjectTemplate Schema*)

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

GET `/community_template/{id}/`

Return a community project template instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "description": "example description",
  "type": "example type",
  "repository": "example repository"
}
```


JSON Parameters

- **id** (*integer*) – Id
- **name** (*string*) – Name(maxLength=1024, minLength=1)
- **description** (*string*) – Description(minLength=1)
- **type** (*string*) – Type(enum=['GIT', 'TAR'])
- **repository** (*string*) – Repository(maxLength=2048, minLength=1)

param integer id(required) A unique value identifying this project template.

POST /community_template/{id}/use_it/

Create project based on this template.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "project_id": 1,
  "name": "example name"
}
```

JSON Parameters

- **project_id** (*integer*) – Project id(readOnly=True, additionalProperties=OrderedDict([('redirect', True)]), x-nullable=True)
- **name** (*string*) – Name(minLength=1)

param integer id(required)

A unique value identifying this project template.

query schema data(required) *ProjectTemplateCreate Schema*

GET /group/

Return all groups.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/group/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "example name",
      "children": true,
      "from_project": true
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```
}
]
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*Group Schema*)

query string id A unique integer value (or comma separated list) identifying this instance.

query string name A name string value (or comma separated list) of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string variables List of variables to filter. Comma separated “key:value” list.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /group/

Create a new group.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "children": true,
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **children** (*boolean*) – Contains groups(default=False)
- **owner** (*object*) – (*User Schema*)

Query Parameters

- **data (required)** (*schema*) – *GroupCreateMaster Schema*

GET /group/{id}/

Return a group instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "children": true,
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **children** (*boolean*) – Children(readOnly=True)
- **owner** (*object*) – (*User Schema*)

param integer id(required) A unique integer value identifying this group.

PUT /group/{id}/

Update a group.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "children": true,
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

(continues on next page)

(continued from previous page)

```
}  
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **children** (*boolean*) – Children(readOnly=True)
- **owner** (*object*) – (*User Schema*)

param integer id(required)

A unique integer value identifying this group.

query schema data(required) *OneGroup Schema*

PATCH /group/{id}/

Update one or more fields on an existing group.

Example Response

```
HTTP/1.1 200 OK  
Vary: OK  
Content-Type: application/json  
  
{  
  "id": 1,  
  "name": "example name",  
  "notes": "example\ntext\narea\n",  
  "children": true,  
  "owner": {  
    "id": 1,  
    "username": "example username",  
    "is_active": true,  
    "email": "example@mail.com"  
  }  
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **children** (*boolean*) – Children(readOnly=True)
- **owner** (*object*) – (*User Schema*)

param integer id(required)

A unique integer value identifying this group.

query schema data(required) *OneGroup Schema*

DELETE /group/{id}/

Remove an existing group.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{}
```

param integer id(required) A unique integer value identifying this group.

POST /group/{id}/copy/

Endpoint which copy instance with deps.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "children": true,
  "from_project": true
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **children** (*boolean*) – Children(readOnly=True)
- **from_project** (*boolean*) – Project Based(readOnly=True)

param integer id(required)

A unique integer value identifying this group.

query schema data(required) *Group Schema*

GET /group/{id}/group/

Return all groups.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/group/1/group/?limit=1&offset=1",
  "previous": null,
  "results": [
```

(continues on next page)

(continued from previous page)

```
{
  {
    "id": 1,
    "name": "example name",
    "children": true,
    "from_project": true
  }
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*Group Schema*)

param integer id(required) A unique integer value identifying this group.

query string id A unique integer value (or comma separated list) identifying this instance.

query string name A name string value (or comma separated list) of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string variables List of variables to filter. Comma separated “key:value” list.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /group/{id}/group/

Create a new group.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  {
    "id": 1,
    "name": "example name",
    "notes": "example\ntext\narea\n",
    "children": true,
    "owner": {
      "id": 1,
      "username": "example username",
      "is_active": true,
      "email": "example@mail.com"
    }
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **children** (*boolean*) – Contains groups(default=False)
- **owner** (*object*) – (*User Schema*)

param integer id(required)

A unique integer value identifying this group.

query schema data(required) *GroupCreateMaster Schema*

GET /group/{id}/group/{group_id}/

Return a group instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "children": true,
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **children** (*boolean*) – Children(readOnly=True)
- **owner** (*object*) – (*User Schema*)

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required) A unique integer value identifying this group.

PUT /group/{id}/group/{group_id}/

Update a group.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json
```

(continues on next page)

(continued from previous page)

```
{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "children": true,
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **children** (*boolean*) – Children(readOnly=True)
- **owner** (*object*) – (*User Schema*)

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required)

A unique integer value identifying this group.

query schema data(required) *OneGroup Schema*

PATCH /group/{id}/group/{group_id}/

Update one or more fields on an existing group.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "children": true,
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)

- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **children** (*boolean*) – Children(readOnly=True)
- **owner** (*object*) – (*User Schema*)

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required)

A unique integer value identifying this group.

query schema data(required) *OneGroup Schema*

DELETE /group/{id}/group/{group_id}/

Remove an existing group.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{ }
```

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required) A unique integer value identifying this group.

POST /group/{id}/group/{group_id}/copy/

Endpoint which copy instance with deps.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "children": true,
  "from_project": true
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **children** (*boolean*) – Children(readOnly=True)
- **from_project** (*boolean*) – Project Based(readOnly=True)

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required)

A unique integer value identifying this group.

query schema data(required) *Group Schema*

POST /group/{id}/group/{group_id}/set_owner/

Change instance owner.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "user_id": 1
}
```

JSON Parameters

- **user_id** (*fk*) – New owner(`additionalProperties=OrderedDict([('model', OrderedDict([('ref', '#/definitions/User')])), ('value_field', 'id'), ('view_field', 'username')]))`)

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required)

A unique integer value identifying this group.

query schema data(required) *SetOwner Schema*

GET /group/{id}/group/{group_id}/variables/

Return all variables of instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/group/1/group/1/variables/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "key": "ansible_host",
      "value": "test_dynamic"
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*InventoryVariable Schema*)

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required) A unique integer value identifying this group.

query string id A unique integer value (or comma separated list) identifying this instance.

query string key A key name string value (or comma separated list) of instance.

query string value A value of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /group/{id}/group/{group_id}/variables/
Create a new variable of instance.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required)

A unique integer value identifying this group.

query schema data(required) *InventoryVariable Schema*

GET /group/{id}/group/{group_id}/variables/{variables_id}/
Return a variable of instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json
```

(continues on next page)

(continued from previous page)

```
{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required) A unique integer value identifying this group.

param integer variables_id(required) A unique integer value identifying instance of this variables sublist.

PUT /group/{id}/group/{group_id}/variables/{variables_id}/
Update variable value.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required) A unique integer value identifying this group.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *InventoryVariable Schema*

PATCH /group/{id}/group/{group_id}/variables/{variables_id}/
Update one or more fields on an existing variable.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required) A unique integer value identifying this group.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *InventoryVariable Schema*

DELETE /group/{id}/group/{group_id}/variables/{variables_id}/
Remove an existing variable.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{}
```

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required) A unique integer value identifying this group.

param integer variables_id(required) A unique integer value identifying instance of this variables sub-list.

GET /group/{id}/host/

Return all hosts.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/group/1/host/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "example name",
      "type": "HOST",
      "from_project": true
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*Host Schema*)

param integer id(required) A unique integer value identifying this group.

query string id A unique integer value (or comma separated list) identifying this instance.

query string name A name string value (or comma separated list) of instance.

query string type Instance type.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string variables List of variables to filter. Comma separated “key:value” list.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /group/{id}/host/

Create a new host.

Example Response

```

HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "type": "HOST",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **owner** (*object*) – (*User Schema*)

param integer id(required)

A unique integer value identifying this group.

query schema data(required) *OneHost Schema*

GET /group/{id}/host/{host_id}/

Return a host instance.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "type": "HOST",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **owner** (*object*) – (*User Schema*)

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required) A unique integer value identifying this group.

PUT /group/{id}/host/{host_id}/
Update a host.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "type": "HOST",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **owner** (*object*) – (*User Schema*)

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required)

A unique integer value identifying this group.

query schema data(required) *OneHost Schema*

PATCH /group/{id}/host/{host_id}/
Update one or more fields on an existing host.

Example Response


```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "type": "HOST",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **owner** (*object*) – (*User Schema*)

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required)

A unique integer value identifying this group.

query schema data(required) *OneHost Schema*

DELETE /group/{id}/host/{host_id}/

Remove an existing host.

Example Response

```

HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{}

```

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required) A unique integer value identifying this group.

POST /group/{id}/host/{host_id}/copy/

Endpoint which copy instance with deps.

Example Response

```

HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

```

(continues on next page)

(continued from previous page)

```
{
  "id": 1,
  "name": "example name",
  "type": "HOST",
  "from_project": true
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **from_project** (*boolean*) – Project Based(readOnly=True)

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required)

A unique integer value identifying this group.

query schema data(required) *Host Schema*

POST /group/{id}/host/{host_id}/set_owner/
Change instance owner.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "user_id": 1
}
```

JSON Parameters

- **user_id** (*fk*) – New owner(additionalProperties=OrderedDict([('model', OrderedDict([('ref', '#/definitions/User')]), ('value_field', 'id'), ('view_field', 'username')]))

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required)

A unique integer value identifying this group.

query schema data(required) *SetOwner Schema*

GET /group/{id}/host/{host_id}/variables/
Return all variables of instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json
```

(continues on next page)

(continued from previous page)

```
{
  "count": 2,
  "next": "http://localhost:8080/group/1/host/1/variables/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "key": "ansible_host",
      "value": "test_dynamic"
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*InventoryVariable Schema*)

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required) A unique integer value identifying this group.

query string id A unique integer value (or comma separated list) identifying this instance.

query string key A key name string value (or comma separated list) of instance.

query string value A value of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /group/{id}/host/{host_id}/variables/

Create a new variable of instance.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)

- **key** (*autocomplete*) – Key(Example values: `ansible_host`, `ansible_port`, `ansible_user`, `ansible_connection`, `ansible_ssh_pass`, `ansible_ssh_private_key_file`, `ansible_ssh_common_args`, `ansible_sftp_extra_args`, `ansible_scp_extra_args`, `ansible_ssh_extra_args`, `ansible_ssh_executable`, `ansible_ssh_pipelining`, `ansible_become`, `ansible_become_method`, `ansible_become_user`, `ansible_become_pass`, `ansible_become_exe`, `ansible_become_flags`, `ansible_shell_type`, `ansible_python_interpreter`, `ansible_ruby_interpreter`, `ansible_perl_interpreter`, `ansible_shell_executable`)
- **value** (*dynamic*) – Value

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required)

A unique integer value identifying this group.

query schema data(required) *InventoryVariable Schema*

GET `/group/{id}/host/{host_id}/variables/{variables_id}/`

Return a variable of instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: `ansible_host`, `ansible_port`, `ansible_user`, `ansible_connection`, `ansible_ssh_pass`, `ansible_ssh_private_key_file`, `ansible_ssh_common_args`, `ansible_sftp_extra_args`, `ansible_scp_extra_args`, `ansible_ssh_extra_args`, `ansible_ssh_executable`, `ansible_ssh_pipelining`, `ansible_become`, `ansible_become_method`, `ansible_become_user`, `ansible_become_pass`, `ansible_become_exe`, `ansible_become_flags`, `ansible_shell_type`, `ansible_python_interpreter`, `ansible_ruby_interpreter`, `ansible_perl_interpreter`, `ansible_shell_executable`)
- **value** (*dynamic*) – Value

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required) A unique integer value identifying this group.

param integer variables_id(required) A unique integer value identifying instance of this variables sublist.

PUT `/group/{id}/host/{host_id}/variables/{variables_id}/`

Update variable value.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required) A unique integer value identifying this group.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *InventoryVariable Schema*

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

Update one or more fields on an existing variable.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)

ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)

- **value** (*dynamic*) – Value

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required) A unique integer value identifying this group.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *InventoryVariable Schema*

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

Remove an existing variable.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{ }
```

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required) A unique integer value identifying this group.

param integer variables_id(required) A unique integer value identifying instance of this variables sublist.

POST /group/{id}/set_owner/

Change instance owner.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "user_id": 1
}
```

JSON Parameters

- **user_id** (*fk*) – New owner(additionalProperties=OrderedDict([('model', OrderedDict([('ref', '#/definitions/User')]), ('value_field', 'id'), ('view_field', 'username')]))

param integer id(required)

A unique integer value identifying this group.

query schema data(required) *SetOwner Schema*

GET /group/{id}/variables/

Return all variables of instance.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/group/1/variables/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "key": "ansible_host",
      "value": "test_dynamic"
    }
  ]
}

```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*InventoryVariable Schema*)

param integer id(required) A unique integer value identifying this group.

query string id A unique integer value (or comma separated list) identifying this instance.

query string key A key name string value (or comma separated list) of instance.

query string value A value of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /group/{id}/variables/

Create a new variable of instance.

Example Response

```

HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: `ansible_host`, `ansible_port`, `ansible_user`, `ansible_connection`, `ansible_ssh_pass`, `ansible_ssh_private_key_file`, `ansible_ssh_common_args`, `ansible_sftp_extra_args`, `ansible_scp_extra_args`, `ansible_ssh_extra_args`, `ansible_ssh_executable`, `ansible_ssh_pipelining`, `ansible_become`, `ansible_become_method`, `ansible_become_user`, `ansible_become_pass`, `ansible_become_exe`, `ansible_become_flags`, `ansible_shell_type`, `ansible_python_interpreter`, `ansible_ruby_interpreter`, `ansible_perl_interpreter`, `ansible_shell_executable`)
- **value** (*dynamic*) – Value

param integer id(required)

A unique integer value identifying this group.

query schema data(required) *InventoryVariable Schema*

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

Return a variable of instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: `ansible_host`, `ansible_port`, `ansible_user`, `ansible_connection`, `ansible_ssh_pass`, `ansible_ssh_private_key_file`, `ansible_ssh_common_args`, `ansible_sftp_extra_args`, `ansible_scp_extra_args`, `ansible_ssh_extra_args`, `ansible_ssh_executable`, `ansible_ssh_pipelining`, `ansible_become`, `ansible_become_method`, `ansible_become_user`, `ansible_become_pass`, `ansible_become_exe`, `ansible_become_flags`, `ansible_shell_type`, `ansible_python_interpreter`, `ansible_ruby_interpreter`, `ansible_perl_interpreter`, `ansible_shell_executable`)
- **value** (*dynamic*) – Value

param integer id(required) A unique integer value identifying this group.

param integer variables_id(required) A unique integer value identifying instance of this variables sub-list.

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

Update variable value.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json
```

(continues on next page)

(continued from previous page)

```
{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer id(required) A unique integer value identifying this group.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *InventoryVariable Schema*

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

Update one or more fields on an existing variable.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer id(required) A unique integer value identifying this group.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *InventoryVariable Schema*

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

Remove an existing variable.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{ }
```

param integer id(required) A unique integer value identifying this group.

param integer variables_id(required) A unique integer value identifying instance of this variables sublist.

GET /history/

Return all history of executions.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/history/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "start_time": "2019-01-07T06:10:31+10:00",
      "executor": 1,
      "initiator": 1,
      "initiator_type": "example initiator_type",
      "project": 1,
      "inventory": 1,
      "kind": "example kind",
      "mode": "example mode",
      "options": "example options",
      "status": "example status",
      "stop_time": "2019-01-07T06:10:31+10:00"
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)

- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*History Schema*)

query string id A unique integer value (or comma separated list) identifying this instance.

query string mode Module or playbook name.

query string kind Kind of execution.

query string status Status of execution.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string name A name string value (or comma separated list) of instance.

query string older Older then this time

query string newer Newer then this time

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

GET /history/{id}/

Return a execution history instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "status": "example status",
  "executor": 1,
  "project": 1,
  "revision": "example revision",
  "inventory": 1,
  "kind": "example kind",
  "mode": "example mode",
  "execute_args": "example execute_args",
  "execution_time": "22:11:34",
  "start_time": "2019-01-07T06:10:31+10:00",
  "stop_time": "2019-01-07T06:10:31+10:00",
  "initiator": 1,
  "initiator_type": "example initiator_type",
  "options": "example options",
  "raw_args": "example raw_args",
  "raw_stdout": "example raw_stdout",
  "raw_inventory": "example raw_inventory"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)

- **status** (*string*) – Status(enum=['DELAY', 'RUN', 'OK', 'ERROR', 'OFFLINE', 'INTERRUPTED'])
- **executor** (*integer*) – Executor(x-nullable=True)
- **project** (*integer*) – Project(x-nullable=True)
- **revision** (*string*) – Revision(maxLength=256, x-nullable=True)
- **inventory** (*integer*) – Inventory(x-nullable=True)
- **kind** (*string*) – Kind(maxLength=50, minLength=1)
- **mode** (*string*) – Mode(maxLength=256, minLength=1)
- **execute_args** (*string*) – Execute args(readOnly=True)
- **execution_time** (*uptime*) – Execution time
- **start_time** (*date-time*) – Start time
- **stop_time** (*date-time*) – Stop time(x-nullable=True)
- **initiator** (*integer*) – Initiator
- **initiator_type** (*string*) – Initiator type(maxLength=50, minLength=1)
- **options** (*string*) – Options(readOnly=True)
- **raw_args** (*string*) – Raw args(minLength=1)
- **raw_stdout** (*string*) – Raw stdout(readOnly=True)
- **raw_inventory** (*string*) – Raw inventory(minLength=1)

param integer id(required) A unique integer value identifying this history.

DELETE /history/{id}/

Remove an existing history record.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{ }
```

param integer id(required) A unique integer value identifying this history.

POST /history/{id}/cancel/

Cancel working task.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "detail": "example detail"
}
```

JSON Parameters

- **detail**(*string*) – Detail(minLength=1)

param integer id(required)

A unique integer value identifying this history.

query schema data(required) *Empty Schema*

DELETE /history/{id}/clear/

Clear history output.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{}
```

param integer id(required) A unique integer value identifying this history.

GET /history/{id}/facts/

Get compiled history facts (only for execution 'module' with module 'setup').

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{}
```

param integer id(required) A unique integer value identifying this history.

GET /hook/

Return all hooks.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/hook/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "example name",
      "type": "example type",
      "when": "example when",
      "enable": true,
      "recipients": "example recipients"
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*Hook Schema*)

query string id A unique integer value (or comma separated list) identifying this instance.

query string name A name string value (or comma separated list) of instance.

query string type Instance type.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /hook/

Create a new hook.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "type": "example type",
  "when": "example when",
  "enable": true,
  "recipients": "example recipients"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **type** (*string*) – Type(enum=['HTTP', 'SCRIPT'])
- **when** (*string*) – When(enum=['on_execution', 'after_execution', 'on_user_add', 'on_user_upd', 'on_user_del', 'on_object_add', 'on_object_upd', 'on_object_del'])
- **enable** (*boolean*) – Enable
- **recipients** (*string*) – Recipients(minLength=1)

Query Parameters

- **data (required)** (*schema*) – *Hook Schema*

GET /hook/{id}/

Return a hook instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "type": "example type",
  "when": "example when",
  "enable": true,
  "recipients": "example recipients"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **type** (*string*) – Type(enum=['HTTP', 'SCRIPT'])
- **when** (*string*) – When(enum=['on_execution', 'after_execution', 'on_user_add', 'on_user_upd', 'on_user_del', 'on_object_add', 'on_object_upd', 'on_object_del'])
- **enable** (*boolean*) – Enable
- **recipients** (*string*) – Recipients(minLength=1)

param integer id(required) A unique integer value identifying this hook.

PUT /hook/{id}/

Update a hook.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "type": "example type",
  "when": "example when",
  "enable": true,
  "recipients": "example recipients"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **type** (*string*) – Type(enum=['HTTP', 'SCRIPT'])

- **when** (*string*) – When(enum=['on_execution', 'after_execution', 'on_user_add', 'on_user_upd', 'on_user_del', 'on_object_add', 'on_object_upd', 'on_object_del'])
- **enable** (*boolean*) – Enable
- **recipients** (*string*) – Recipients(minLength=1)

param integer id(required)

A unique integer value identifying this hook.

query schema data(required) *Hook Schema*

PATCH /hook/{id}/

Update one or more fields on an existing hook.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "type": "example type",
  "when": "example when",
  "enable": true,
  "recipients": "example recipients"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **type** (*string*) – Type(enum=['HTTP', 'SCRIPT'])
- **when** (*string*) – When(enum=['on_execution', 'after_execution', 'on_user_add', 'on_user_upd', 'on_user_del', 'on_object_add', 'on_object_upd', 'on_object_del'])
- **enable** (*boolean*) – Enable
- **recipients** (*string*) – Recipients(minLength=1)

param integer id(required)

A unique integer value identifying this hook.

query schema data(required) *Hook Schema*

DELETE /hook/{id}/

Remove an existing hook.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{}
```


param integer id(required) A unique integer value identifying this hook.

GET /host/

Return all hosts.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/host/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "example name",
      "type": "HOST",
      "from_project": true
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*Host Schema*)

query string id A unique integer value (or comma separated list) identifying this instance.

query string name A name string value (or comma separated list) of instance.

query string type Instance type.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string variables List of variables to filter. Comma separated “key:value” list.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /host/

Create a new host.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json
```

(continues on next page)

(continued from previous page)

```
{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "type": "HOST",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **owner** (*object*) – (*User Schema*)

Query Parameters

- **data (required)** (*schema*) – *OneHost Schema*

GET /host/{id}/

Return a host instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "type": "HOST",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes

- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **owner** (*object*) – (*User Schema*)

param integer id(required) A unique integer value identifying this host.

PUT /host/{id}/

Update a host.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "type": "HOST",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **owner** (*object*) – (*User Schema*)

param integer id(required)

A unique integer value identifying this host.

query schema data(required) *OneHost Schema*

PATCH /host/{id}/

Update one or more fields on an existing host.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "type": "HOST",
```

(continues on next page)

(continued from previous page)

```
"owner": {
  "id": 1,
  "username": "example username",
  "is_active": true,
  "email": "example@mail.com"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **owner** (*object*) – (*User Schema*)

param integer id(required)

A unique integer value identifying this host.

query schema data(required) *OneHost Schema*

DELETE /host/{id}/

Remove an existing host.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{}
```

param integer id(required) A unique integer value identifying this host.

POST /host/{id}/copy/

Endpoint which copy instance with deps.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "type": "HOST",
  "from_project": true
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)

- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **from_project** (*boolean*) – Project Based(readOnly=True)

param integer id(required)

A unique integer value identifying this host.

query schema data(required) *Host Schema*

POST /host/{id}/set_owner/

Change instance owner.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "user_id": 1
}
```

JSON Parameters

- **user_id** (*fk*) – New owner(additionalProperties=OrderedDict([('model', OrderedDict([('ref', '#/definitions/User')])), ('value_field', 'id'), ('view_field', 'username')]))

param integer id(required)

A unique integer value identifying this host.

query schema data(required) *SetOwner Schema*

GET /host/{id}/variables/

Return all variables of instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/host/1/variables/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "key": "ansible_host",
      "value": "test_dynamic"
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None

- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*InventoryVariable Schema*)

param integer id(required) A unique integer value identifying this host.

query string id A unique integer value (or comma separated list) identifying this instance.

query string key A key name string value (or comma separated list) of instance.

query string value A value of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /host/{id}/variables/

Create a new variable of instance.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer id(required)

A unique integer value identifying this host.

query schema data(required) *InventoryVariable Schema*

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

Return a variable of instance.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer id(required) A unique integer value identifying this host.

param integer variables_id(required) A unique integer value identifying instance of this variables sub-list.

PUT /host/{id}/variables/{variables_id}/
Update variable value.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer id(required) A unique integer value identifying this host.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *InventoryVariable Schema*

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

Update one or more fields on an existing variable.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer id(required) A unique integer value identifying this host.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *InventoryVariable Schema*

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

Remove an existing variable.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{ }
```

param integer id(required) A unique integer value identifying this host.

param integer variables_id(required) A unique integer value identifying instance of this variables sublist.

GET /inventory/

Return all inventories.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/inventory/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "example name",
      "from_project": true
    }
  ]
}

```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*Inventory Schema*)

query string id A unique integer value (or comma separated list) identifying this instance.

query string name A name string value (or comma separated list) of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string variables List of variables to filter. Comma separated “key:value” list.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /inventory/

Create a new inventory.

Example Response

```

HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",

```

(continues on next page)

(continued from previous page)

```
"owner": {
  "id": 1,
  "username": "example username",
  "is_active": true,
  "email": "example@mail.com"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **owner** (*object*) – (*User Schema*)

Query Parameters

- **data (required)** (*schema*) – *OneInventory Schema*

POST /inventory/import_inventory/
Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "inventory_id": 1,
  "name": "example name",
  "raw_data": "example raw_data"
}
```

JSON Parameters

- **inventory_id** (*integer*) – Inventory id(readOnly=True, additionalProperties=OrderedDict([('redirect', True)]), x-nullable=True)
- **name** (*string*) – Name(minLength=1)
- **raw_data** (*string*) – Raw data(minLength=1)

Query Parameters

- **data (required)** (*schema*) – *InventoryImport Schema*

GET /inventory/{id}/
Return a inventory instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json
```

(continues on next page)

(continued from previous page)

```
{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **owner** (*object*) – (*User Schema*)

param integer id(required) A unique integer value identifying this inventory.

PUT /inventory/{id}/

Update a inventory.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **owner** (*object*) – (*User Schema*)

param integer id(required)

A unique integer value identifying this inventory.

query schema data(required) *OneInventory Schema*

PATCH /inventory/{id}/

Update one or more fields on an existing inventory.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **owner** (*object*) – (*User Schema*)

param integer id(required)

A unique integer value identifying this inventory.

query schema data(required) *OneInventory Schema*

DELETE /inventory/{id}/

Remove an existing inventory.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{ }
```

param integer id(required) A unique integer value identifying this inventory.

GET /inventory/{id}/all_groups/

Return all groups.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json
```

(continues on next page)

(continued from previous page)

```
{
  "count": 2,
  "next": "http://localhost:8080/inventory/1/all_groups/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "example name",
      "children": true,
      "from_project": true
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*Group Schema*)

param integer id(required) A unique integer value identifying this inventory.

query string id A unique integer value (or comma separated list) identifying this instance.

query string name A name string value (or comma separated list) of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string variables List of variables to filter. Comma separated “key:value” list.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

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

Return a group instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "children": true,
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,

```

(continues on next page)

(continued from previous page)

```
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **children** (*boolean*) – Children(readOnly=True)
- **owner** (*object*) – (*User Schema*)

param string all_groups_id(required)

param integer id(required) A unique integer value identifying this inventory.

GET /inventory/{id}/all_hosts/
Return all hosts.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/inventory/1/all_hosts/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "example name",
      "type": "HOST",
      "from_project": true
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*Host Schema*)

param integer id(required) A unique integer value identifying this inventory.

query string id A unique integer value (or comma separated list) identifying this instance.

query string name A name string value (or comma separated list) of instance.

query string type Instance type.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string variables List of variables to filter. Comma separated "key:value" list.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

GET /inventory/{id}/all_hosts/{all_hosts_id}/
Return a host instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "type": "HOST",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **owner** (*object*) – (*User Schema*)

param string all_hosts_id(required)

param integer id(required) A unique integer value identifying this inventory.

POST /inventory/{id}/copy/
Endpoint which copy instance with deps.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
```

(continues on next page)

(continued from previous page)

```
"from_project": true
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **from_project** (*boolean*) – Project Based(readOnly=True)

param integer id(required)

A unique integer value identifying this inventory.

query schema data(required) *Inventory Schema*

GET /inventory/{id}/group/

Return all groups.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/inventory/1/group/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "example name",
      "children": true,
      "from_project": true
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*Group Schema*)

param integer id(required) A unique integer value identifying this inventory.

query string id A unique integer value (or comma separated list) identifying this instance.

query string name A name string value (or comma separated list) of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string variables List of variables to filter. Comma separated “key:value” list.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /inventory/{id}/group/

Create a new group.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "children": true,
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **children** (*boolean*) – Contains groups(default=False)
- **owner** (*object*) – (*User Schema*)

param integer id(required)

A unique integer value identifying this inventory.

query schema data(required) *GroupCreateMaster Schema*

GET /inventory/{id}/group/{group_id}/

Return a group instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "children": true,
```

(continues on next page)

(continued from previous page)

```
"owner": {
  "id": 1,
  "username": "example username",
  "is_active": true,
  "email": "example@mail.com"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **children** (*boolean*) – Children(readOnly=True)
- **owner** (*object*) – (*User Schema*)

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required) A unique integer value identifying this inventory.

PUT /inventory/{id}/group/{group_id}/
Update a group.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "children": true,
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **children** (*boolean*) – Children(readOnly=True)
- **owner** (*object*) – (*User Schema*)

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required)

A unique integer value identifying this inventory.

query schema data(required) *OneGroup Schema*

PATCH /inventory/{id}/group/{group_id}/

Update one or more fields on an existing group.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "children": true,
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **children** (*boolean*) – Children(readOnly=True)
- **owner** (*object*) – (*User Schema*)

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required)

A unique integer value identifying this inventory.

query schema data(required) *OneGroup Schema*

DELETE /inventory/{id}/group/{group_id}/

Remove an existing group.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{}
```

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required) A unique integer value identifying this inventory.

POST `/inventory/{id}/group/{group_id}/copy/`

Endpoint which copy instance with deps.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "children": true,
  "from_project": true
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **children** (*boolean*) – Children(readOnly=True)
- **from_project** (*boolean*) – Project Based(readOnly=True)

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required)

A unique integer value identifying this inventory.

query schema data(required) *Group Schema*

POST `/inventory/{id}/group/{group_id}/set_owner/`

Change instance owner.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "user_id": 1
}
```

JSON Parameters

- **user_id** (*fk*) – New owner(additionalProperties=OrderedDict([('model', OrderedDict([('ref', '#/definitions/User')])), ('value_field', 'id'), ('view_field', 'username')]))

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required)

A unique integer value identifying this inventory.

query schema data(required) *SetOwner Schema*

GET `/inventory/{id}/group/{group_id}/variables/`
Return all variables of instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/inventory/1/group/1/variables/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "key": "ansible_host",
      "value": "test_dynamic"
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*InventoryVariable Schema*)

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required) A unique integer value identifying this inventory.

query string id A unique integer value (or comma separated list) identifying this instance.

query string key A key name string value (or comma separated list) of instance.

query string value A value of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST `/inventory/{id}/group/{group_id}/variables/`
Create a new variable of instance.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
```

(continues on next page)

(continued from previous page)

```

    "id": 1,
    "key": "ansible_host",
    "value": "test_dynamic"
  }

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required)

A unique integer value identifying this inventory.

query schema data(required) *InventoryVariable Schema*

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

Return a variable of instance.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required) A unique integer value identifying this inventory.

param integer variables_id(required) A unique integer value identifying instance of this variables sublist.

PUT /inventory/{id}/group/{group_id}/variables/{variables_id}/

Update variable value.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required) A unique integer value identifying this inventory.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *InventoryVariable Schema*

PATCH /inventory/{id}/group/{group_id}/variables/{variables_id}/

Update one or more fields on an existing variable.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: `ansible_host`, `ansible_port`, `ansible_user`, `ansible_connection`, `ansible_ssh_pass`, `ansible_ssh_private_key_file`, `ansible_ssh_common_args`, `ansible_sftp_extra_args`, `ansible_scp_extra_args`, `ansible_ssh_extra_args`, `ansible_ssh_executable`, `ansible_ssh_pipelining`, `ansible_become`, `ansible_become_method`, `ansible_become_user`, `ansible_become_pass`, `ansible_become_exe`, `ansible_become_flags`, `ansible_shell_type`, `ansible_python_interpreter`, `ansible_ruby_interpreter`, `ansible_perl_interpreter`, `ansible_shell_executable`)
- **value** (*dynamic*) – Value

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required) A unique integer value identifying this inventory.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *InventoryVariable Schema*

DELETE /inventory/{id}/group/{group_id}/variables/{variables_id}/
Remove an existing variable.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{ }
```

param integer group_id(required) A unique integer value identifying instance of this groups sublist.

param integer id(required) A unique integer value identifying this inventory.

param integer variables_id(required) A unique integer value identifying instance of this variables sublist.

GET /inventory/{id}/host/
Return all hosts.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/inventory/1/host/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "example name",
      "type": "HOST",
      "from_project": true
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```
    ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*Host Schema*)

param integer id(required) A unique integer value identifying this inventory.

query string id A unique integer value (or comma separated list) identifying this instance.

query string name A name string value (or comma separated list) of instance.

query string type Instance type.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string variables List of variables to filter. Comma separated “key:value” list.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /inventory/{id}/host/

Create a new host.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "type": "HOST",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes

- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **owner** (*object*) – (*User Schema*)

param integer id(required)

A unique integer value identifying this inventory.

query schema data(required) *OneHost Schema*

GET /inventory/{id}/host/{host_id}/

Return a host instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "type": "HOST",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **owner** (*object*) – (*User Schema*)

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required) A unique integer value identifying this inventory.

PUT /inventory/{id}/host/{host_id}/

Update a host.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
```

(continues on next page)

(continued from previous page)

```

    "type": "HOST",
    "owner": {
      "id": 1,
      "username": "example username",
      "is_active": true,
      "email": "example@mail.com"
    }
  }
}

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **owner** (*object*) – (*User Schema*)

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required)

A unique integer value identifying this inventory.

query schema data(required) *OneHost Schema*

PATCH /inventory/{id}/host/{host_id}/

Update one or more fields on an existing host.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "type": "HOST",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)

- **owner** (*object*) – (*User Schema*)

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required)

A unique integer value identifying this inventory.

query schema data(required) *OneHost Schema*

DELETE /inventory/{id}/host/{host_id}/

Remove an existing host.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{ }
```

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required) A unique integer value identifying this inventory.

POST /inventory/{id}/host/{host_id}/copy/

Endpoint which copy instance with deps.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "type": "HOST",
  "from_project": true
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **from_project** (*boolean*) – Project Based(readOnly=True)

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required)

A unique integer value identifying this inventory.

query schema data(required) *Host Schema*

POST /inventory/{id}/host/{host_id}/set_owner/

Change instance owner.

Example Response

```

HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "user_id": 1
}

```

JSON Parameters

- **user_id** (*fk*) – New owner(`additionalProperties=OrderedDict([(‘model’, OrderedDict([(‘$ref’, ‘#/definitions/User’)])), (‘value_field’, ‘id’), (‘view_field’, ‘username’)]))`)

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required)

A unique integer value identifying this inventory.

query schema data(required) *SetOwner Schema*

GET /inventory/{id}/host/{host_id}/variables/
Return all variables of instance.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/inventory/1/host/1/variables/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "key": "ansible_host",
      "value": "test_dynamic"
    }
  ]
}

```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*InventoryVariable Schema*)

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required) A unique integer value identifying this inventory.

query string id A unique integer value (or comma separated list) identifying this instance.

query string key A key name string value (or comma separated list) of instance.

query string value A value of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /inventory/{id}/host/{host_id}/variables/

Create a new variable of instance.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required)

A unique integer value identifying this inventory.

query schema data(required) *InventoryVariable Schema*

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

Return a variable of instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
```

(continues on next page)

(continued from previous page)

```

    "key": "ansible_host",
    "value": "test_dynamic"
}

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required) A unique integer value identifying this inventory.

param integer variables_id(required) A unique integer value identifying instance of this variables sublist.

PUT /inventory/{id}/host/{host_id}/variables/{variables_id}/
Update variable value.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required) A unique integer value identifying this inventory.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *InventoryVariable Schema*

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

Update one or more fields on an existing variable.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required) A unique integer value identifying this inventory.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *InventoryVariable Schema*

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

Remove an existing variable.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{ }
```

param integer host_id(required) A unique integer value identifying instance of this hosts sublist.

param integer id(required) A unique integer value identifying this inventory.

param integer variables_id(required) A unique integer value identifying instance of this variables sublist.

POST `/inventory/{id}/set_owner/`

Change instance owner.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "user_id": 1
}
```

JSON Parameters

- **user_id** (*fk*) – New owner(`additionalProperties=OrderedDict([('model', OrderedDict([('ref', '#/definitions/User')]), ('value_field', 'id'), ('view_field', 'username')])`)

param integer id(required)

A unique integer value identifying this inventory.

query schema data(required) *SetOwner Schema*

GET `/inventory/{id}/variables/`

Return all variables of instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/inventory/1/variables/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "key": "ansible_host",
      "value": "test_dynamic"
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*InventoryVariable Schema*)

param integer id(required) A unique integer value identifying this inventory.

query string id A unique integer value (or comma separated list) identifying this instance.

query string key A key name string value (or comma separated list) of instance.

query string value A value of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /inventory/{id}/variables/

Create a new variable of instance.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: `ansible_host`, `ansible_port`, `ansible_user`, `ansible_connection`, `ansible_ssh_pass`, `ansible_ssh_private_key_file`, `ansible_ssh_common_args`, `ansible_sftp_extra_args`, `ansible_scp_extra_args`, `ansible_ssh_extra_args`, `ansible_ssh_executable`, `ansible_ssh_pipelining`, `ansible_become`, `ansible_become_method`, `ansible_become_user`, `ansible_become_pass`, `ansible_become_exe`, `ansible_become_flags`, `ansible_shell_type`, `ansible_python_interpreter`, `ansible_ruby_interpreter`, `ansible_perl_interpreter`, `ansible_shell_executable`)
- **value** (*dynamic*) – Value

param integer id(required)

A unique integer value identifying this inventory.

query schema data(required) *InventoryVariable Schema*

GET /inventory/{id}/variables/{variables_id}/

Return a variable of instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: `ansible_host`, `ansible_port`, `ansible_user`, `ansible_connection`, `ansible_ssh_pass`, `ansible_ssh_private_key_file`, `ansible_ssh_common_args`, `ansible_sftp_extra_args`, `ansible_scp_extra_args`, `ansible_ssh_extra_args`, `ansible_ssh_executable`, `ansible_ssh_pipelining`, `ansible_become`, `ansible_become_method`, `ansible_become_user`, `ansible_become_pass`, `ansible_become_exe`, `ansible_become_flags`, `ansible_shell_type`, `ansible_python_interpreter`, `ansible_ruby_interpreter`, `ansible_perl_interpreter`, `ansible_shell_executable`)
- **value** (*dynamic*) – Value

param integer id(required) A unique integer value identifying this inventory.

param integer variables_id(required) A unique integer value identifying instance of this variables sublist.

PUT /inventory/{id}/variables/{variables_id}/
Update variable value.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: `ansible_host`, `ansible_port`, `ansible_user`, `ansible_connection`, `ansible_ssh_pass`, `ansible_ssh_private_key_file`, `ansible_ssh_common_args`, `ansible_sftp_extra_args`, `ansible_scp_extra_args`, `ansible_ssh_extra_args`, `ansible_ssh_executable`, `ansible_ssh_pipelining`, `ansible_become`, `ansible_become_method`, `ansible_become_user`, `ansible_become_pass`, `ansible_become_exe`, `ansible_become_flags`, `ansible_shell_type`, `ansible_python_interpreter`, `ansible_ruby_interpreter`, `ansible_perl_interpreter`, `ansible_shell_executable`)
- **value** (*dynamic*) – Value

param integer id(required) A unique integer value identifying this inventory.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *InventoryVariable Schema*

PATCH /inventory/{id}/variables/{variables_id}/
Update one or more fields on an existing variable.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer id(required) A unique integer value identifying this inventory.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *InventoryVariable Schema*

DELETE /inventory/{id}/variables/{variables_id}/

Remove an existing variable.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{}
```

param integer id(required) A unique integer value identifying this inventory.

param integer variables_id(required) A unique integer value identifying instance of this variables sublist.

GET /project/

Return all projects.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
```

(continues on next page)

(continued from previous page)

```

"count": 2,
"next": "http://localhost:8080/project/?limit=1&offset=1",
"previous": null,
"results": [
  {
    "id": 1,
    "name": "example name",
    "type": "example type",
    "status": "example status"
  }
]
}

```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*Project Schema*)

query string id A unique integer value (or comma separated list) identifying this instance.

query string name A name string value (or comma separated list) of instance.

query string status Project sync status.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string variables List of variables to filter. Comma separated “key:value” list.

query string status__not Project sync status.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /project/

Create a new project.

Example Response

```

HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "status": "example status",
  "type": "MANUAL",
  "repository": "MANUAL",
  "repo_auth": "NONE",
  "auth_data": "test_dynamic",
  "branch": "test_dynamic",

```

(continues on next page)

(continued from previous page)

```

"additional_playbook_path": "example additional_playbook_path"
}

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **status** (*string*) – Status(readOnly=True, minLength=1)
- **type** (*string*) – Repo type(enum=['MANUAL', 'GIT', 'TAR'], default=MANUAL)
- **repository** (*string*) – Repo url(default=MANUAL, minLength=1)
- **repo_auth** (*dynamic*) – Repo auth type
- **auth_data** (*dynamic*) – Repo auth data
- **branch** (*dynamic*) – Branch for GIT(branch/tag/SHA) or TAR(subdir)
- **additional_playbook_path** (*string*) – Directory with playbooks(minLength=1, x-nullable=True)

Query Parameters

- **data (required)** (*schema*) – *ProjectCreateMaster Schema*

GET /project/{id}/

Return a project instance.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "repository": "MANUAL",
  "status": "example status",
  "revision": "example revision",
  "branch": "example branch",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  },
  "notes": "example\ntext\narea\n",
  "readme_content": "test_html",
  "execute_view_data": {}
}

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)

- **repository** (*string*) – Repository(default=MANUAL, minLength=1)
- **status** (*string*) – Status(enum=['NEW', 'ERROR', 'OK', 'WAIT_SYNC', 'SYNC'], readOnly=True)
- **revision** (*string*) – Revision(readOnly=True)
- **branch** (*string*) – Branch(readOnly=True)
- **owner** (*object*) – (*User Schema*)
- **notes** (*textarea*) – Notes
- **readme_content** (*html*) – Information(readOnly=True)
- **execute_view_data** (*object*) – (*Data Schema*)

param integer id(required) A unique integer value identifying this project.

PUT /project/{id}/
Update a project.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "repository": "MANUAL",
  "status": "example status",
  "revision": "example revision",
  "branch": "example branch",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  },
  "notes": "example\ntext\narea\n",
  "readme_content": "test_html",
  "execute_view_data": {}
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **repository** (*string*) – Repository(default=MANUAL, minLength=1)
- **status** (*string*) – Status(enum=['NEW', 'ERROR', 'OK', 'WAIT_SYNC', 'SYNC'], readOnly=True)
- **revision** (*string*) – Revision(readOnly=True)
- **branch** (*string*) – Branch(readOnly=True)
- **owner** (*object*) – (*User Schema*)

- **notes** (*textarea*) – Notes
- **readme_content** (*html*) – Information(readOnly=True)
- **execute_view_data** (*object*) – (*Data Schema*)

param integer id(required)

A unique integer value identifying this project.

query schema data(required) *OneProject Schema*

PATCH /project/{id}/

Update one or more fields on an existing project.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "repository": "MANUAL",
  "status": "example status",
  "revision": "example revision",
  "branch": "example branch",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  },
  "notes": "example\ntext\narea\n",
  "readme_content": "test_html",
  "execute_view_data": {}
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **repository** (*string*) – Repository(default=MANUAL, minLength=1)
- **status** (*string*) – Status(enum=['NEW', 'ERROR', 'OK', 'WAIT_SYNC', 'SYNC'], readOnly=True)
- **revision** (*string*) – Revision(readOnly=True)
- **branch** (*string*) – Branch(readOnly=True)
- **owner** (*object*) – (*User Schema*)
- **notes** (*textarea*) – Notes
- **readme_content** (*html*) – Information(readOnly=True)
- **execute_view_data** (*object*) – (*Data Schema*)

param integer id(required)

A unique integer value identifying this project.

query schema data(required) *OneProject Schema*

DELETE /project/{id}/

Remove an existing project.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{ }
```

param integer id(required) A unique integer value identifying this project.

POST /project/{id}/copy/

Endpoint which copy instance with deps.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "type": "example type",
  "status": "example status"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **type** (*string*) – Type(readOnly=True, minLength=1)
- **status** (*string*) – Status(enum=['NEW', 'ERROR', 'OK', 'WAIT_SYNC', 'SYNC'], readOnly=True)

param integer id(required)

A unique integer value identifying this project.

query schema data(required) *Project Schema*

POST /project/{id}/execute_module/

Execute *ansible -m [module]* with arguments.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
```

(continues on next page)

(continued from previous page)

```
"detail": "example detail",
"history_id": 1,
"executor": 1
}
```

JSON Parameters

- **detail** (*string*) – Detail(minLength=1)
- **history_id** (*integer*) – History id(additionalProperties=OrderedDict([('redirect', True)]), x-nullable=True)
- **executor** (*integer*) – Executor(x-nullable=True)

param integer id(required)

A unique integer value identifying this project.

query schema data(required) *AnsibleModule Schema*

POST /project/{id}/execute_playbook/

Execute *ansible-playbook* with arguments.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "detail": "example detail",
  "history_id": 1,
  "executor": 1
}
```

JSON Parameters

- **detail** (*string*) – Detail(minLength=1)
- **history_id** (*integer*) – History id(additionalProperties=OrderedDict([('redirect', True)]), x-nullable=True)
- **executor** (*integer*) – Executor(x-nullable=True)

param integer id(required)

A unique integer value identifying this project.

query schema data(required) *AnsiblePlaybook Schema*

GET /project/{id}/history/

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
```

(continues on next page)

(continued from previous page)

```

"count": 2,
"next": "http://localhost:8080/project/1/history/?limit=1&offset=1",
"previous": null,
"results": [
  {
    "id": 1,
    "start_time": "2019-01-07T06:10:31+10:00",
    "executor": 1,
    "initiator": 1,
    "initiator_type": "example initiator_type",
    "revision": "example revision",
    "inventory": 1,
    "kind": "example kind",
    "mode": "example mode",
    "options": "example options",
    "status": "example status",
    "stop_time": "2019-01-07T06:10:31+10:00"
  }
]
}

```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*ProjectHistory Schema*)

param integer id(required) A unique integer value identifying this project.

query string id A unique integer value (or comma separated list) identifying this instance.

query string mode Module or playbook name.

query string kind Kind of execution.

query string status Status of execution.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string name A name string value (or comma separated list) of instance.

query string older Older then this time

query string newer Newer then this time

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

GET /project/{id}/history/{history_id}/
Example Response

```

HTTP/1.1 200 OK
Vary: OK

```

(continues on next page)

(continued from previous page)

```
Content-Type: application/json

{
  "id": 1,
  "status": "example status",
  "executor": 1,
  "project": 1,
  "revision": "example revision",
  "inventory": 1,
  "kind": "example kind",
  "mode": "example mode",
  "execute_args": "example execute_args",
  "execution_time": "22:11:34",
  "start_time": "2019-01-07T06:10:31+10:00",
  "stop_time": "2019-01-07T06:10:31+10:00",
  "initiator": 1,
  "initiator_type": "example initiator_type",
  "options": "example options",
  "raw_args": "example raw_args",
  "raw_stdout": "example raw_stdout",
  "raw_inventory": "example raw_inventory"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **status** (*string*) – Status(enum=['DELAY', 'RUN', 'OK', 'ERROR', 'OFFLINE', 'INTERRUPTED'])
- **executor** (*integer*) – Executor(x-nullable=True)
- **project** (*integer*) – Project(x-nullable=True)
- **revision** (*string*) – Revision(maxLength=256, x-nullable=True)
- **inventory** (*integer*) – Inventory(x-nullable=True)
- **kind** (*string*) – Kind(maxLength=50, minLength=1)
- **mode** (*string*) – Mode(maxLength=256, minLength=1)
- **execute_args** (*string*) – Execute args(readOnly=True)
- **execution_time** (*uptime*) – Execution time
- **start_time** (*date-time*) – Start time
- **stop_time** (*date-time*) – Stop time(x-nullable=True)
- **initiator** (*integer*) – Initiator
- **initiator_type** (*string*) – Initiator type(maxLength=50, minLength=1)
- **options** (*string*) – Options(readOnly=True)
- **raw_args** (*string*) – Raw args(minLength=1)
- **raw_stdout** (*string*) – Raw stdout(readOnly=True)
- **raw_inventory** (*string*) – Raw inventory(minLength=1)

param integer history_id(required) A unique integer value identifying instance of this history sublist.

param integer id(required) A unique integer value identifying this project.

DELETE /project/{id}/history/{history_id}/

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{ }
```

param integer history_id(required) A unique integer value identifying instance of this history sublist.

param integer id(required) A unique integer value identifying this project.

POST /project/{id}/history/{history_id}/cancel/

Cancel working task.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "detail": "example detail"
}
```

JSON Parameters

- **detail** (*string*) – Detail(minLength=1)

param integer history_id(required) A unique integer value identifying instance of this history sublist.

param integer id(required)

A unique integer value identifying this project.

query schema data(required) *Empty Schema*

DELETE /project/{id}/history/{history_id}/clear/

Clear history output.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{ }
```

param integer history_id(required) A unique integer value identifying instance of this history sublist.

param integer id(required) A unique integer value identifying this project.

GET /project/{id}/history/{history_id}/facts/

Get compiled history facts (only for execution ‘module’ with module ‘setup’).

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{}
```

param integer history_id(required) A unique integer value identifying instance of this history sublist.

param integer id(required) A unique integer value identifying this project.

GET `/project/{id}/inventory/`
Return all inventories.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/project/1/inventory/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "example name",
      "from_project": true
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*Inventory Schema*)

param integer id(required) A unique integer value identifying this project.

query string id A unique integer value (or comma separated list) identifying this instance.

query string name A name string value (or comma separated list) of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string variables List of variables to filter. Comma separated “key:value” list.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /project/{id}/inventory/

Create a new inventory.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **owner** (*object*) – (*User Schema*)

param integer id(required)

A unique integer value identifying this project.

query schema data(required) *OneInventory Schema*

POST /project/{id}/inventory/file_import_inventory/

Create a new inventory.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "inventory_id": 1,
  "name": "example name",
  "raw_data": "example raw_data"
}
```

JSON Parameters

- **inventory_id** (*integer*) – Inventory id(readOnly=True, additionalProperties=OrderedDict([('redirect', True)]), x-nullable=True)
- **name** (*string*) – Name(pattern=([^][^.]⁺?\$)([^]([.][V/])?[^]([.]*[wd-[^]]+?[^][.]*[wd-[^]]*[V/][^])+?\$), minLength=1)

- **raw_data** (*string*) – Raw data(readOnly=True, minLength=1)

param integer id(required)

A unique integer value identifying this project.

query schema data(required) *InventoryFileImport Schema*

POST /project/{id}/inventory/import_inventory/

Create a new inventory.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "inventory_id": 1,
  "name": "example name",
  "raw_data": "example raw_data"
}
```

JSON Parameters

- **inventory_id** (*integer*) – Inventory id(readOnly=True, additionalProperties=OrderedDict([('redirect', True)]), x-nullable=True)
- **name** (*string*) – Name(minLength=1)
- **raw_data** (*string*) – Raw data(minLength=1)

param integer id(required)

A unique integer value identifying this project.

query schema data(required) *InventoryImport Schema*

GET /project/{id}/inventory/{inventory_id}/

Return a inventory instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **owner** (*object*) – (*User Schema*)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

PUT /project/{id}/inventory/{inventory_id}/
Update a inventory.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **owner** (*object*) – (*User Schema*)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required)

A unique integer value identifying instance of this inventories sublist.

query schema data(required) *OneInventory Schema*

PATCH /project/{id}/inventory/{inventory_id}/
Update one or more fields on an existing inventory.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json
```

(continues on next page)

(continued from previous page)

```
{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **owner** (*object*) – (*User Schema*)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required)

A unique integer value identifying instance of this inventories sublist.

query schema data(required) *OneInventory Schema*

DELETE /project/{id}/inventory/{inventory_id}/

Remove an existing inventory.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{}
```

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

GET /project/{id}/inventory/{inventory_id}/all_groups/

Return all groups.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/project/1/inventory/1/all_groups/?limit=1&
  ↪offset=1",
```

(continues on next page)

(continued from previous page)

```

    "previous": null,
    "results": [
      {
        "id": 1,
        "name": "example name",
        "children": true,
        "from_project": true
      }
    ]
  }
}

```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*Group Schema*)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

query string id A unique integer value (or comma separated list) identifying this instance.

query string name A name string value (or comma separated list) of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string variables List of variables to filter. Comma separated “key:value” list.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

GET /project/{id}/inventory/{inventory_id}/all_groups/{all_groups_id}/
Return a group instance.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "children": true,
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}

```

(continues on next page)

(continued from previous page)

```
}
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **children** (*boolean*) – Children(readOnly=True)
- **owner** (*object*) – (*User Schema*)

param string all_groups_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

GET /project/{id}/inventory/{inventory_id}/all_hosts/
Return all hosts.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/project/1/inventory/1/all_hosts/?limit=1&
↪offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "example name",
      "type": "HOST",
      "from_project": true
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*Host Schema*)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

query string id A unique integer value (or comma separated list) identifying this instance.

query string name A name string value (or comma separated list) of instance.

query string type Instance type.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string variables List of variables to filter. Comma separated "key:value" list.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

GET /project/{id}/inventory/{inventory_id}/all_hosts/{all_hosts_id}/
Return a host instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "type": "HOST",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **owner** (*object*) – (*User Schema*)

param string all_hosts_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

POST /project/{id}/inventory/{inventory_id}/copy/
Endpoint which copy instance with deps.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "from_project": true
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **from_project** (*boolean*) – Project Based(readOnly=True)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required)

A unique integer value identifying instance of this inventories sublist.

query schema data(required) *Inventory Schema*

GET /project/{id}/inventory/{inventory_id}/group/
Return all groups.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/project/1/inventory/1/group/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "example name",
      "children": true,
      "from_project": true
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*Group Schema*)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

query string id A unique integer value (or comma separated list) identifying this instance.

query string name A name string value (or comma separated list) of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string variables List of variables to filter. Comma separated "key:value" list.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /project/{id}/inventory/{inventory_id}/group/
Create a new group.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "children": true,
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **children** (*boolean*) – Contains groups(default=False)
- **owner** (*object*) – (*User Schema*)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required)

A unique integer value identifying instance of this inventories sublist.

query schema data(required) *GroupCreateMaster Schema*

GET /project/{id}/inventory/{inventory_id}/group/{group_id}/
Return a group instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "children": true,
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **children** (*boolean*) – Children(readOnly=True)
- **owner** (*object*) – (*User Schema*)

param string group_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

PUT /project/{id}/inventory/{inventory_id}/group/{group_id}/
Update a group.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "children": true,
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```


JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **children** (*boolean*) – Children(readOnly=True)
- **owner** (*object*) – (*User Schema*)

param string group_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required)

A unique integer value identifying instance of this inventories sublist.

query schema data(required) *OneGroup Schema*

PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/

Update one or more fields on an existing group.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "children": true,
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **children** (*boolean*) – Children(readOnly=True)
- **owner** (*object*) – (*User Schema*)

param string group_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required)

A unique integer value identifying instance of this inventories sublist.

query schema data(required) *OneGroup Schema*

DELETE /project/{id}/inventory/{inventory_id}/group/{group_id}/

Remove an existing group.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{ }
```

param string group_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

POST /project/{id}/inventory/{inventory_id}/group/{group_id}/copy/

Endpoint which copy instance with deps.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "children": true,
  "from_project": true
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **children** (*boolean*) – Children(readOnly=True)
- **from_project** (*boolean*) – Project Based(readOnly=True)

param string group_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required)

A unique integer value identifying instance of this inventories sublist.

query schema data(required) *Group Schema*

POST /project/{id}/inventory/{inventory_id}/group/{group_id}/set_owner/

Change instance owner.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json
```

(continues on next page)

(continued from previous page)

```
{
  "user_id": 1
}
```

JSON Parameters

- **user_id** (*fk*) – New owner(`additionalProperties=OrderedDict([(‘model’, OrderedDict([(‘$ref’, ‘#/definitions/User’)])), (‘value_field’, ‘id’), (‘view_field’, ‘username’)]))`)

param string group_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required)

A unique integer value identifying instance of this inventories sublist.

query schema data(required) *SetOwner Schema*

GET /project/{id}/inventory/{inventory_id}/group/{group_id}/variables/

Return all variables of instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/project/1/inventory/1/group/1/variables/?
↪limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "key": "ansible_host",
      "value": "test_dynamic"
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*InventoryVariable Schema*)

param string group_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

query string id A unique integer value (or comma separated list) identifying this instance.

query string key A key name string value (or comma separated list) of instance.

query string value A value of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST `/project/{id}/inventory/{inventory_id}/group/{group_id}/variables/`

Create a new variable of instance.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: `ansible_host`, `ansible_port`, `ansible_user`, `ansible_connection`, `ansible_ssh_pass`, `ansible_ssh_private_key_file`, `ansible_ssh_common_args`, `ansible_sftp_extra_args`, `ansible_scp_extra_args`, `ansible_ssh_extra_args`, `ansible_ssh_executable`, `ansible_ssh_pipelining`, `ansible_become`, `ansible_become_method`, `ansible_become_user`, `ansible_become_pass`, `ansible_become_exe`, `ansible_become_flags`, `ansible_shell_type`, `ansible_python_interpreter`, `ansible_ruby_interpreter`, `ansible_perl_interpreter`, `ansible_shell_executable`)
- **value** (*dynamic*) – Value

param string group_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required)

A unique integer value identifying instance of this inventories sublist.

query schema data(required) *InventoryVariable Schema*

GET `/project/{id}/inventory/{inventory_id}/group/{group_id}/variables/{variables_id}/`

Return a variable of instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json
```

(continues on next page)

(continued from previous page)

```
{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param string group_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

param integer variables_id(required) A unique integer value identifying instance of this variables sublist.

PUT /project/{id}/inventory/{inventory_id}/group/{group_id}/variables/{variables_id}/
Update variable value.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)

- **value** (*dynamic*) – Value

param string group_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *InventoryVariable Schema*

PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/variables/{variables_id}/
Update one or more fields on an existing variable.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param string group_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *InventoryVariable Schema*

DELETE /project/{id}/inventory/{inventory_id}/group/{group_id}/variables/{variables_id}/
Remove an existing variable.

Example Response

```

HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{}

```

param string group_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

param integer variables_id(required) A unique integer value identifying instance of this variables sublist.

GET /project/{id}/inventory/{inventory_id}/host/
Return all hosts.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/project/1/inventory/1/host/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "example name",
      "type": "HOST",
      "from_project": true
    }
  ]
}

```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*Host Schema*)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

query string id A unique integer value (or comma separated list) identifying this instance.

query string name A name string value (or comma separated list) of instance.

query string type Instance type.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string variables List of variables to filter. Comma separated "key:value" list.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /project/{id}/inventory/{inventory_id}/host/
Create a new host.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "type": "HOST",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **owner** (*object*) – (*User Schema*)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required)

A unique integer value identifying instance of this inventories sublist.

query schema data(required) *OneHost Schema*

GET /project/{id}/inventory/{inventory_id}/host/{host_id}/
Return a host instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json
```

(continues on next page)

(continued from previous page)

```
{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "type": "HOST",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **owner** (*object*) – (*User Schema*)

param string host_id(required)**param integer id(required)** A unique integer value identifying this project.**param integer inventory_id(required)** A unique integer value identifying instance of this inventories sublist.

PUT /project/{id}/inventory/{inventory_id}/host/{host_id}/
Update a host.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "type": "HOST",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)

- **notes** (*textarea*) – Notes
- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **owner** (*object*) – (*User Schema*)

param string host_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required)

A unique integer value identifying instance of this inventories sublist.

query schema data(required) *OneHost Schema*

PATCH /project/{id}/inventory/{inventory_id}/host/{host_id}/

Update one or more fields on an existing host.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "type": "HOST",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **owner** (*object*) – (*User Schema*)

param string host_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required)

A unique integer value identifying instance of this inventories sublist.

query schema data(required) *OneHost Schema*

DELETE /project/{id}/inventory/{inventory_id}/host/{host_id}/

Remove an existing host.

Example Response

```

HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{}

```

param string host_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

POST /project/{id}/inventory/{inventory_id}/host/{host_id}/copy/

Endpoint which copy instance with deps.

Example Response

```

HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "type": "HOST",
  "from_project": true
}

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **type** (*string*) – Type(enum=['HOST', 'RANGE'], default=HOST)
- **from_project** (*boolean*) – Project Based(readOnly=True)

param string host_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required)

A unique integer value identifying instance of this inventories sublist.

query schema data(required) *Host Schema*

POST /project/{id}/inventory/{inventory_id}/host/{host_id}/set_owner/

Change instance owner.

Example Response

```

HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{

```

(continues on next page)

(continued from previous page)

```

    "user_id": 1
  }

```

JSON Parameters

- **user_id** (*fk*) – New owner(`additionalProperties=OrderedDict([('model', OrderedDict([('ref', '#/definitions/User')])), ('value_field', 'id'), ('view_field', 'username')]))`)

param string host_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required)

A unique integer value identifying instance of this inventories sublist.

query schema data(required) *SetOwner Schema*

GET `/project/{id}/inventory/{inventory_id}/host/{host_id}/variables/`
Return all variables of instance.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/project/1/inventory/1/host/1/variables/?
↪limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "key": "ansible_host",
      "value": "test_dynamic"
    }
  ]
}

```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*InventoryVariable Schema*)

param string host_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

query string id A unique integer value (or comma separated list) identifying this instance.

query string key A key name string value (or comma separated list) of instance.

query string value A value of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /project/{id}/inventory/{inventory_id}/host/{host_id}/variables/
Create a new variable of instance.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param string host_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required)

A unique integer value identifying instance of this inventories sublist.

query schema data(required) *InventoryVariable Schema*

GET /project/{id}/inventory/{inventory_id}/host/{host_id}/variables/{variables_id}/
Return a variable of instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
```

(continues on next page)

(continued from previous page)

```
"key": "ansible_host",
"value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param string host_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

param integer variables_id(required) A unique integer value identifying instance of this variables sublist.

PUT /project/{id}/inventory/{inventory_id}/host/{host_id}/variables/{variables_id}/
Update variable value.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param string host_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *InventoryVariable Schema*

PATCH /project/{id}/inventory/{inventory_id}/host/{host_id}/variables/{variables_id}/
Update one or more fields on an existing variable.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param string host_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *InventoryVariable Schema*

DELETE /project/{id}/inventory/{inventory_id}/host/{host_id}/variables/{variables_id}/
Remove an existing variable.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json
```

(continues on next page)

(continued from previous page)

```
{ }
```

param string host_id(required)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

param integer variables_id(required) A unique integer value identifying instance of this variables sublist.

POST /project/{id}/inventory/{inventory_id}/set_owner/
Change instance owner.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "user_id": 1
}
```

JSON Parameters

- **user_id** (*fk*) – New owner(`additionalProperties=OrderedDict([('model', OrderedDict([('ref', '#/definitions/User')])), ('value_field', 'id'), ('view_field', 'username')]))`)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required)

A unique integer value identifying instance of this inventories sublist.

query schema data(required) *SetOwner Schema*

GET /project/{id}/inventory/{inventory_id}/variables/
Return all variables of instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/project/1/inventory/1/variables/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "key": "ansible_host",
      "value": "test_dynamic"
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```

    }
  ]
}

```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*InventoryVariable Schema*)

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

query string id A unique integer value (or comma separated list) identifying this instance.

query string key A key name string value (or comma separated list) of instance.

query string value A value of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /project/{id}/inventory/{inventory_id}/variables/
Create a new variable of instance.

Example Response

```

HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required)

A unique integer value identifying instance of this inventories sublist.

query schema data(required) *InventoryVariable Schema*

GET `/project/{id}/inventory/{inventory_id}/variables/{variables_id}/`
Return a variable of instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

param integer variables_id(required) A unique integer value identifying instance of this variables sublist.

PUT `/project/{id}/inventory/{inventory_id}/variables/{variables_id}/`
Update variable value.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *InventoryVariable Schema*

PATCH /project/{id}/inventory/{inventory_id}/variables/{variables_id}/
Update one or more fields on an existing variable.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "ansible_host",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: ansible_host, ansible_port, ansible_user, ansible_connection, ansible_ssh_pass, ansible_ssh_private_key_file, ansible_ssh_common_args, ansible_sftp_extra_args, ansible_scp_extra_args, ansible_ssh_extra_args, ansible_ssh_executable, ansible_ssh_pipelining, ansible_become, ansible_become_method, ansible_become_user, ansible_become_pass, ansible_become_exe, ansible_become_flags, ansible_shell_type, ansible_python_interpreter, ansible_ruby_interpreter, ansible_perl_interpreter, ansible_shell_executable)
- **value** (*dynamic*) – Value

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *InventoryVariable Schema*

DELETE `/project/{id}/inventory/{inventory_id}/variables/{variables_id}/`
Remove an existing variable.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{ }
```

param integer id(required) A unique integer value identifying this project.

param integer inventory_id(required) A unique integer value identifying instance of this inventories sublist.

param integer variables_id(required) A unique integer value identifying instance of this variables sublist.

GET `/project/{id}/module/`
Return all available modules of project.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/project/1/module/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "path": "example path",
      "name": "example name"
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*Module Schema*)

param integer id(required) A unique integer value identifying this project.

query string path Full path to module.

query string name Name of module.

query string path__not Full path to module.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

GET /project/{id}/module/{module_id}/
Return a module details of project instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "path": "example path",
  "name": "example name",
  "data": {}
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **path** (*string*) – Path(maxLength=1024, minLength=1)
- **name** (*string*) – Name(readOnly=True)
- **data** (*object*) – Data(readOnly=True)

param integer id(required) A unique integer value identifying this project.

param string module_id(required)

GET /project/{id}/periodic_task/
Return all periodic tasks in project.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/project/1/periodic_task/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "example name",
      "kind": "PLAYBOOK",
      "mode": "test_dynamic",
      "inventory": "test_dynamic",
      "save_result": true,
      "template": 1,
      "template_opt": "test_dynamic",
      "enabled": true,
      "type": "CRONTAB",
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```
        "schedule": "test_dynamic"
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*Periodictask Schema*)

param integer id(required) A unique integer value identifying this project.

query string id A unique integer value (or comma separated list) identifying this instance.

query string mode Periodic task module or playbook name.

query string kind Kind of periodic task.

query string type Instance type.

query number template A unique integer id of template used in periodic task.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string name A name string value (or comma separated list) of instance.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /project/{id}/periodic_task/

Create a new periodic task.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "kind": "PLAYBOOK",
  "mode": "test_dynamic",
  "inventory": "test_dynamic",
  "save_result": true,
  "template": 1,
  "template_opt": "test_dynamic",
  "enabled": true,
  "type": "CRONTAB",
  "schedule": "test_dynamic",
  "notes": "example\ntext\narea\n"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **kind** (*string*) – Task type(enum=['PLAYBOOK', 'MODULE', 'TEMPLATE'], default=PLAYBOOK)
- **mode** (*dynamic*) – Mode
- **inventory** (*dynamic*) – Inventory
- **save_result** (*boolean*) – Save result
- **template** (*integer*) – Template(x-nullable=True)
- **template_opt** (*dynamic*) – Template opt
- **enabled** (*boolean*) – Enabled
- **type** (*string*) – Interval type(enum=['CRONTAB', 'INTERVAL'], default=CRONTAB)
- **schedule** (*dynamic*) – Schedule
- **notes** (*textarea*) – Notes

param integer id(required)

A unique integer value identifying this project.

query schema data(required) *OnePeriodictask Schema*

GET /project/{id}/periodic_task/{periodic_task_id}/

Return a perodic task instance.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "kind": "PLAYBOOK",
  "mode": "test_dynamic",
  "inventory": "test_dynamic",
  "save_result": true,
  "template": 1,
  "template_opt": "test_dynamic",
  "enabled": true,
  "type": "CRONTAB",
  "schedule": "test_dynamic",
  "notes": "example\ntext\narea\n"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)

- **kind** (*string*) – Task type(enum=['PLAYBOOK', 'MODULE', 'TEMPLATE'], default=PLAYBOOK)
- **mode** (*dynamic*) – Mode
- **inventory** (*dynamic*) – Inventory
- **save_result** (*boolean*) – Save result
- **template** (*integer*) – Template(x-nullable=True)
- **template_opt** (*dynamic*) – Template opt
- **enabled** (*boolean*) – Enabled
- **type** (*string*) – Interval type(enum=['CRONTAB', 'INTERVAL'], default=CRONTAB)
- **schedule** (*dynamic*) – Schedule
- **notes** (*textarea*) – Notes

param integer id(required) A unique integer value identifying this project.

param integer periodic_task_id(required) A unique integer value identifying instance of this periodic_task sublist.

PUT /project/{id}/periodic_task/{periodic_task_id}/
Update a periodic task.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "kind": "PLAYBOOK",
  "mode": "test_dynamic",
  "inventory": "test_dynamic",
  "save_result": true,
  "template": 1,
  "template_opt": "test_dynamic",
  "enabled": true,
  "type": "CRONTAB",
  "schedule": "test_dynamic",
  "notes": "example\ntext\narea\n"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **kind** (*string*) – Task type(enum=['PLAYBOOK', 'MODULE', 'TEMPLATE'], default=PLAYBOOK)
- **mode** (*dynamic*) – Mode
- **inventory** (*dynamic*) – Inventory
- **save_result** (*boolean*) – Save result

- **template** (*integer*) – Template(x-nullable=True)
- **template_opt** (*dynamic*) – Template opt
- **enabled** (*boolean*) – Enabled
- **type** (*string*) – Interval type(enum=['CRONTAB', 'INTERVAL'], default=CRONTAB)
- **schedule** (*dynamic*) – Schedule
- **notes** (*textarea*) – Notes

param integer id(required) A unique integer value identifying this project.

param integer periodic_task_id(required)

A unique integer value identifying instance of this periodic_task sublist.

query schema data(required) *OnePeriodictask Schema*

PATCH /project/{id}/periodic_task/{periodic_task_id}/

Update one or more fields on an existing periodic task.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "kind": "PLAYBOOK",
  "mode": "test_dynamic",
  "inventory": "test_dynamic",
  "save_result": true,
  "template": 1,
  "template_opt": "test_dynamic",
  "enabled": true,
  "type": "CRONTAB",
  "schedule": "test_dynamic",
  "notes": "example\ntext\narea\n"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **kind** (*string*) – Task type(enum=['PLAYBOOK', 'MODULE', 'TEMPLATE'], default=PLAYBOOK)
- **mode** (*dynamic*) – Mode
- **inventory** (*dynamic*) – Inventory
- **save_result** (*boolean*) – Save result
- **template** (*integer*) – Template(x-nullable=True)
- **template_opt** (*dynamic*) – Template opt
- **enabled** (*boolean*) – Enabled

- **type** (*string*) – Interval type(enum=['CRONTAB', 'INTERVAL'], default=CRONTAB)
- **schedule** (*dynamic*) – Schedule
- **notes** (*textarea*) – Notes

param integer id(required) A unique integer value identifying this project.

param integer periodic_task_id(required)

A unique integer value identifying instance of this periodic_task sublist.

query schema data(required) *OnePeriodictask Schema*

DELETE /project/{id}/periodic_task/{periodic_task_id}/

Remove an existing periodic task.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{ }
```

param integer id(required) A unique integer value identifying this project.

param integer periodic_task_id(required) A unique integer value identifying instance of this periodic_task sublist.

POST /project/{id}/periodic_task/{periodic_task_id}/execute/

Ad-hoc execute periodic task.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "detail": "example detail",
  "history_id": 1,
  "executor": 1
}
```

JSON Parameters

- **detail** (*string*) – Detail(minLength=1)
- **history_id** (*integer*) – History id(additionalProperties=OrderedDict([('redirect', True)]), x-nullable=True)
- **executor** (*integer*) – Executor(x-nullable=True)

param integer id(required) A unique integer value identifying this project.

param integer periodic_task_id(required)

A unique integer value identifying instance of this periodic_task sublist.

query schema data(required) *Empty Schema*

GET /project/{id}/periodic_task/{periodic_task_id}/variables/
Return all variables of periodic task.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/project/1/periodic_task/1/variables/?limit=1&
  ↪offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "key": "example key",
      "value": "example value"
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*PeriodicTaskVariable Schema*)

param integer id(required) A unique integer value identifying this project.

param integer periodic_task_id(required) A unique integer value identifying instance of this periodic_task sublist.

query string id A unique integer value (or comma separated list) identifying this instance.

query string key A key name string value (or comma separated list) of instance.

query string value A value of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /project/{id}/periodic_task/{periodic_task_id}/variables/
Create a new variable of periodic task.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json
```

(continues on next page)

(continued from previous page)

```
{
  "id": 1,
  "key": "example key",
  "value": "example value"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*string*) – Key(maxLength=512, minLength=1)
- **value** (*string*) – Value(default=)

param integer id(required) A unique integer value identifying this project.

param integer periodic_task_id(required)

A unique integer value identifying instance of this periodic_task sublist.

query schema data(required) *PeriodicTaskVariable Schema*

GET /project/{id}/periodic_task/{periodic_task_id}/variables/{variables_id}/
Return a variable of periodic task.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "example key",
  "value": "example value"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*string*) – Key(maxLength=512, minLength=1)
- **value** (*string*) – Value(default=)

param integer id(required) A unique integer value identifying this project.

param integer periodic_task_id(required) A unique integer value identifying instance of this periodic_task sublist.

param integer variables_id(required) A unique integer value identifying instance of this variables sublist.

PUT /project/{id}/periodic_task/{periodic_task_id}/variables/{variables_id}/
Update variable value.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "example key",
  "value": "example value"
}

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*string*) – Key(maxLength=512, minLength=1)
- **value** (*string*) – Value(default=)

param integer id(required) A unique integer value identifying this project.

param integer periodic_task_id(required) A unique integer value identifying instance of this periodic_task sublist.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *PeriodicTaskVariable Schema*

PATCH /project/{id}/periodic_task/{periodic_task_id}/variables/{variables_id}/
Update one or more fields on an existing variable.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "example key",
  "value": "example value"
}

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*string*) – Key(maxLength=512, minLength=1)
- **value** (*string*) – Value(default=)

param integer id(required) A unique integer value identifying this project.

param integer periodic_task_id(required) A unique integer value identifying instance of this periodic_task sublist.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *PeriodicTaskVariable Schema*

DELETE `/project/{id}/periodic_task/{periodic_task_id}/variables/{variables_id}/`
Remove an existing variable.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{ }
```

param integer id(required) A unique integer value identifying this project.

param integer periodic_task_id(required) A unique integer value identifying instance of this periodic_task sublist.

param integer variables_id(required) A unique integer value identifying instance of this variables sublist.

GET `/project/{id}/playbook/`
Return all playbooks of project.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/project/1/playbook/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "example name",
      "playbook": "example playbook"
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*Playbook Schema*)

param integer id(required) A unique integer value identifying this project.

query string id A unique integer value (or comma separated list) identifying this instance.

query string name A name string value (or comma separated list) of instance.

query string playbook Playbook filename.

query string pb_filter Playbook filename - filter for prefetch.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string playbook__not Playbook filename.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

GET /project/{id}/playbook/{playbook_id}/
Return a playbook of project instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "playbook": "example playbook"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=251, minLength=1)
- **playbook** (*string*) – Playbook(readOnly=True, minLength=1)

param integer id(required) A unique integer value identifying this project.

param integer playbook_id(required) A unique integer value identifying instance of this playbook sub-list.

POST /project/{id}/set_owner/
Change instance owner.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "user_id": 1
}
```

JSON Parameters

- **user_id** (*fk*) – New owner(additionalProperties=OrderedDict([('model', OrderedDict([('ref', '#/definitions/User')])), ('value_field', 'id'), ('view_field', 'username')]))

param integer id(required)

A unique integer value identifying this project.

query schema data(required) *SetOwner Schema*

POST `/project/{id}/sync/`
Sync project with repository.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "detail": "example detail"
}
```

JSON Parameters

- **detail** (*string*) – Detail(minLength=1)

param integer id(required)

A unique integer value identifying this project.

query schema data(required) *Empty Schema*

GET `/project/{id}/template/`
Return all execute templates in project.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/project/1/template/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "example name",
      "kind": "Task",
      "data": {},
      "options": {},
      "options_list": [
        "array_example"
      ]
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)

- **results** (*array*) – (*Template Schema*)

param integer id(required) A unique integer value identifying this project.

query string id A unique integer value (or comma separated list) identifying this instance.

query string name A name string value (or comma separated list) of instance.

query string kind A kind of template.

query string inventory The inventory id or path in project.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /project/{id}/template/

Create a new execute template.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "kind": "Task",
  "data": {},
  "options": {},
  "options_list": [
    "array_example"
  ]
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **kind** (*string*) – Type(enum=['Task', 'Module'], default=Task)
- **data** (*object*) – (*Data Schema*)
- **options** (*object*) – (*Data Schema*)
- **options_list** (*array*) – (items=OrderedDict([('type', 'string')]), readOnly=True)

param integer id(required)

A unique integer value identifying this project.

query schema data(required) *OneTemplate Schema*

GET /project/{id}/template/{template_id}/

Return a execute template instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "kind": "Task",
  "data": {},
  "options": {},
  "options_list": [
    "array_example"
  ]
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **kind** (*string*) – Type(enum=[‘Task’, ‘Module’], default=Task)
- **data** (*object*) – (*Data Schema*)
- **options** (*object*) – (*Data Schema*)
- **options_list** (*array*) – (items=OrderedDict([‘type’, ‘string’])), readOnly=True)

param integer id(required) A unique integer value identifying this project.

param integer template_id(required) A unique integer value identifying instance of this template sub-list.

PUT /project/{id}/template/{template_id}/

Update a execute template.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "kind": "Task",
  "data": {},
  "options": {},
  "options_list": [
```

(continues on next page)

(continued from previous page)

```

    "array_example"
  ]
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **kind** (*string*) – Type(enum=['Task', 'Module'], default=Task)
- **data** (*object*) – (*Data Schema*)
- **options** (*object*) – (*Data Schema*)
- **options_list** (*array*) – (items=OrderedDict([('type', 'string')]), readOnly=True)

param integer id(required) A unique integer value identifying this project.

param integer template_id(required)

A unique integer value identifying instance of this template sublist.

query schema data(required) *OneTemplate Schema*

PATCH /project/{id}/template/{template_id}/

Update one or more fields on an existing execute template.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "kind": "Task",
  "data": {},
  "options": {},
  "options_list": [
    "array_example"
  ]
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **name** (*string*) – Name(maxLength=512, minLength=1)
- **notes** (*textarea*) – Notes
- **kind** (*string*) – Type(enum=['Task', 'Module'], default=Task)
- **data** (*object*) – (*Data Schema*)
- **options** (*object*) – (*Data Schema*)

- **options_list** (*array*) – (items=OrderedDict([('type', 'string')]), readOnly=True)

param integer id(required) A unique integer value identifying this project.

param integer template_id(required)

A unique integer value identifying instance of this template sublist.

query schema data(required) *OneTemplate Schema*

DELETE /project/{id}/template/{template_id}/

Remove an existing execute template.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{ }
```

param integer id(required) A unique integer value identifying this project.

param integer template_id(required) A unique integer value identifying instance of this template sublist.

POST /project/{id}/template/{template_id}/execute/

Execute template with option.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "detail": "example detail",
  "history_id": 1,
  "executor": 1
}
```

JSON Parameters

- **detail** (*string*) – Detail(minLength=1)
- **history_id** (*integer*) – History id(additionalProperties=OrderedDict([('redirect', True)]), x-nullable=True)
- **executor** (*integer*) – Executor(x-nullable=True)

param integer id(required) A unique integer value identifying this project.

param integer template_id(required)

A unique integer value identifying instance of this template sublist.

query schema data(required) *TemplateExec Schema*

GET /project/{id}/variables/

Return all variables of instance.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/project/1/variables/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "key": "repo_type",
      "value": "test_dynamic"
    }
  ]
}

```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*ProjectVariable Schema*)

param integer id(required) A unique integer value identifying this project.

query string id A unique integer value (or comma separated list) identifying this instance.

query string key A key name string value (or comma separated list) of instance.

query string value A value of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /project/{id}/variables/
Create a new variable of instance.

Example Response

```

HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "key": "repo_type",
  "value": "test_dynamic"
}

```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: repo_type, repo_sync_on_run, repo_sync_on_run_timeout, repo_branch, repo_password, repo_key, playbook_path, ci_template)
- **value** (*dynamic*) – Value

param integer id(required)

A unique integer value identifying this project.

query schema data(required) *ProjectVariable Schema*

GET /project/{id}/variables/{variables_id}/

Return a variable of instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "repo_type",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: repo_type, repo_sync_on_run, repo_sync_on_run_timeout, repo_branch, repo_password, repo_key, playbook_path, ci_template)
- **value** (*dynamic*) – Value

param integer id(required) A unique integer value identifying this project.

param integer variables_id(required) A unique integer value identifying instance of this variables sub-list.

PUT /project/{id}/variables/{variables_id}/

Update variable value.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "repo_type",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: repo_type, repo_sync_on_run, repo_sync_on_run_timeout, repo_branch, repo_password, repo_key, playbook_path, ci_template)
- **value** (*dynamic*) – Value

param integer id(required) A unique integer value identifying this project.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *ProjectVariable Schema*

PATCH /project/{id}/variables/{variables_id}/

Update one or more fields on an existing variable.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "key": "repo_type",
  "value": "test_dynamic"
}
```

JSON Parameters

- **id** (*integer*) – Id(readOnly=True)
- **key** (*autocomplete*) – Key(Example values: repo_type, repo_sync_on_run, repo_sync_on_run_timeout, repo_branch, repo_password, repo_key, playbook_path, ci_template)
- **value** (*dynamic*) – Value

param integer id(required) A unique integer value identifying this project.

param integer variables_id(required)

A unique integer value identifying instance of this variables sublist.

query schema data(required) *ProjectVariable Schema*

DELETE /project/{id}/variables/{variables_id}/

Remove an existing variable.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{}
```

param integer id(required) A unique integer value identifying this project.

param integer variables_id(required) A unique integer value identifying instance of this variables sub-list.

GET /team/

Return all teams.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/team/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "name": "example name"
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*Team Schema*)

query string id A unique integer value (or comma separated list) identifying this instance.

query string name A name string value (or comma separated list) of instance.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string name__not A name string value (or comma separated list) of instance.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /team/

Create a new team.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
}
```

(continues on next page)

(continued from previous page)

```

    "owner": {
      "id": 1,
      "username": "example username",
      "is_active": true,
      "email": "example@mail.com"
    }
  }
}

```

JSON Parameters

- **id** (*integer*) – ID(readOnly=True)
- **name** (*string*) – Name(maxLength=150, minLength=1)
- **notes** (*textarea*) – Notes
- **owner** (*object*) – (*User Schema*)

Query Parameters

- **data (required)** (*schema*) – *OneTeam Schema*

GET /team/{id}/

Return a team instance.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}

```

JSON Parameters

- **id** (*integer*) – ID(readOnly=True)
- **name** (*string*) – Name(maxLength=150, minLength=1)
- **notes** (*textarea*) – Notes
- **owner** (*object*) – (*User Schema*)

param integer id(required) A unique integer value identifying this user group.**PUT** /team/{id}/

Update a team.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – ID(readOnly=True)
- **name** (*string*) – Name(maxLength=150, minLength=1)
- **notes** (*textarea*) – Notes
- **owner** (*object*) – (*User Schema*)

param integer id(required)

A unique integer value identifying this user group.

query schema data(required) *OneTeam Schema*

PATCH /team/{id}/

Update one or more fields on an existing team.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "name": "example name",
  "notes": "example\ntext\narea\n",
  "owner": {
    "id": 1,
    "username": "example username",
    "is_active": true,
    "email": "example@mail.com"
  }
}
```

JSON Parameters

- **id** (*integer*) – ID(readOnly=True)
- **name** (*string*) – Name(maxLength=150, minLength=1)

- **notes** (*textarea*) – Notes
- **owner** (*object*) – (*User Schema*)

param integer id(required)

A unique integer value identifying this user group.

query schema data(required) *OneTeam Schema*

DELETE /team/{id}/

Remove an existing team.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{ }
```

param integer id(required) A unique integer value identifying this user group.

POST /team/{id}/copy/

Endpoint which copy instance with deps.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "name": "example name"
}
```

JSON Parameters

- **id** (*integer*) – ID(readOnly=True)
- **name** (*string*) – Name(maxLength=150, minLength=1)

param integer id(required)

A unique integer value identifying this user group.

query schema data(required) *Team Schema*

POST /team/{id}/set_owner/

Change instance owner.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "user_id": 1
}
```

JSON Parameters

- **user_id** (*fk*) – New owner(`additionalProperties=OrderedDict([('model', OrderedDict([('ref', '#/definitions/User')])), ('value_field', 'id'), ('view_field', 'username')]))`)

param integer id(required)

A unique integer value identifying this user group.

query schema data(required) *SetOwner Schema*

GET /team/{id}/user/

Return all users.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/team/1/user/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "username": "example username",
      "is_active": true,
      "email": "example@mail.com"
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*User Schema*)

param integer id(required) A unique integer value identifying this user group.

query string id A unique integer value (or comma separated list) identifying this instance.

query string username A name string value (or comma separated list) of instance.

query string is_active Boolean value meaning status of user.

query string first_name Users first name.

query string last_name Users last name.

query string email Users e-mail value.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string username__not A name string value (or comma separated list) of instance.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /team/{id}/user/

Create a new user.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "username": "example username",
  "is_active": true,
  "is_staff": true,
  "first_name": "example first_name",
  "last_name": "example last_name",
  "email": "example@mail.com",
  "password": "example password",
  "password2": "example password2"
}
```

JSON Parameters

- **id** (*integer*) – ID(readOnly=True)
- **username** (*string*) – Username(description=Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only., pattern=^[w. @+-]+\$, maxLength=150, minLength=1)
- **is_active** (*boolean*) – Is active(default=True)
- **is_staff** (*boolean*) – Is staff(default=False)
- **first_name** (*string*) – First name(maxLength=30)
- **last_name** (*string*) – Last name(maxLength=150)
- **email** (*email*) – Email(minLength=1)
- **password** (*string*) – Password(minLength=1)
- **password2** (*string*) – Repeat password(minLength=1)

param integer id(required)

A unique integer value identifying this user group.

query schema data(required) *CreateUser Schema*

GET /team/{id}/user/{user_id}/

Return a user instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
```

(continues on next page)

(continued from previous page)

```
"username": "example username",
"is_active": true,
"first_name": "example first_name",
"last_name": "example last_name",
"email": "example@mail.com"
}
```

JSON Parameters

- **id** (*integer*) – ID(readOnly=True)
- **username** (*string*) – Username(description=Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only., pattern=^[w.@+-]+\$, maxLength=150, minLength=1)
- **is_active** (*boolean*) – Is active(default=True)
- **first_name** (*string*) – First name(maxLength=30)
- **last_name** (*string*) – Last name(maxLength=150)
- **email** (*email*) – Email(minLength=1)

param integer id(required) A unique integer value identifying this user group.

param integer user_id(required) A unique integer value identifying instance of this users sublist.

PUT /team/{id}/user/{user_id}/
Update a user.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "username": "example username",
  "is_active": true,
  "first_name": "example first_name",
  "last_name": "example last_name",
  "email": "example@mail.com"
}
```

JSON Parameters

- **id** (*integer*) – ID(readOnly=True)
- **username** (*string*) – Username(description=Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only., pattern=^[w.@+-]+\$, maxLength=150, minLength=1)
- **is_active** (*boolean*) – Is active(default=True)
- **first_name** (*string*) – First name(maxLength=30)
- **last_name** (*string*) – Last name(maxLength=150)
- **email** (*email*) – Email(minLength=1)

param integer id(required) A unique integer value identifying this user group.

param integer user_id(required)

A unique integer value identifying instance of this users sublist.

query schema data(required) *OneUser Schema*

PATCH /team/{id}/user/{user_id}/

Update one or more fields on an existing user.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "username": "example username",
  "is_active": true,
  "first_name": "example first_name",
  "last_name": "example last_name",
  "email": "example@mail.com"
}
```

JSON Parameters

- **id** (*integer*) – ID(readOnly=True)
- **username** (*string*) – Username(description=Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only., pattern=^[w.@+-]+\$, maxLength=150, minLength=1)
- **is_active** (*boolean*) – Is active(default=True)
- **first_name** (*string*) – First name(maxLength=30)
- **last_name** (*string*) – Last name(maxLength=150)
- **email** (*email*) – Email(minLength=1)

param integer id(required) A unique integer value identifying this user group.

param integer user_id(required)

A unique integer value identifying instance of this users sublist.

query schema data(required) *OneUser Schema*

DELETE /team/{id}/user/{user_id}/

Remove an existing user.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{ }
```

param integer id(required) A unique integer value identifying this user group.

param integer user_id(required) A unique integer value identifying instance of this users sublist.

POST /team/{id}/user/{user_id}/change_password/

Create a new user.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "old_password": "example old_password",
  "password": "example password",
  "password2": "example password2"
}
```

JSON Parameters

- **old_password** (*string*) – Old password(minLength=1)
- **password** (*string*) – New password(minLength=1)
- **password2** (*string*) – Confirm new password(minLength=1)

param integer id(required) A unique integer value identifying this user group.

param integer user_id(required)

A unique integer value identifying instance of this users sublist.

query schema data(required) *ChangePassword Schema*

POST /team/{id}/user/{user_id}/copy/

Endpoint which copy instance with deps.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "username": "example username",
  "is_active": true,
  "email": "example@mail.com"
}
```

JSON Parameters

- **id** (*integer*) – ID(readOnly=True)
- **username** (*string*) – Username(description=Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only., pattern=^[w.@+-]+\$, maxLength=150, minLength=1)
- **is_active** (*boolean*) – Is active(default=True)
- **email** (*email*) – Email(minLength=1)

param integer id(required) A unique integer value identifying this user group.

param integer user_id(required)

A unique integer value identifying instance of this users sublist.

query schema data(required) *User Schema*

GET /team/{id}/user/{user_id}/settings/

```
from rest_framework.settings import api_settings print(api_settings.DEFAULT_RENDERER_CLASSES)
```

Any setting with string import paths will be automatically resolved and return the class, rather than the string literal.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "lang": "en",
  "autoupdateInterval": 15000,
  "chartLineSettings": {
    "all_tasks": {
      "active": true
    },
    "delay": {
      "active": true
    },
    "ok": {
      "active": true
    },
    "error": {
      "active": true
    },
    "interrupted": {
      "active": true
    },
    "offline": {
      "active": true
    }
  },
  "widgetSettings": {
    "pmwUsersCounter": {
      "active": true,
      "collapse": true,
      "sort": 1
    },
    "pmwProjectsCounter": {
      "active": true,
      "collapse": true,
      "sort": 1
    },
    "pmwTemplatesCounter": {
      "active": true,
      "collapse": true,
      "sort": 1
    },
    "pmwInventoriesCounter": {
      "active": true,
      "collapse": true,
```

(continues on next page)

(continued from previous page)

```

        "sort": 1
    },
    "pmwGroupsCounter": {
        "active": true,
        "collapse": true,
        "sort": 1
    },
    "pmwHostsCounter": {
        "active": true,
        "collapse": true,
        "sort": 1
    },
    "pmwChartWidget": {
        "active": true,
        "collapse": true,
        "sort": 1
    }
},
"selectedSkin": "example selectedSkin",
"skinsSettings": {}
}

```

JSON Parameters

- **lang** (*string*) – Lang(enum=['en', 'ru'], default=en)
- **autoupdateInterval** (*integer*) – Autoupdateinterval(default=15000)
- **chartLineSettings** (*object*) – ([ChartLineSettings Schema](#))
- **widgetSettings** (*object*) – ([WidgetSettings Schema](#))
- **selectedSkin** (*string*) – Selectedskin(minLength=1)
- **skinsSettings** (*object*) – ([Data Schema](#))

param integer id(required) A unique integer value identifying this user group.

param integer user_id(required) A unique integer value identifying instance of this users sublist.

POST /team/{id}/user/{user_id}/settings/

from rest_framework.settings import api_settings print(api_settings.DEFAULT_RENDERER_CLASSES)

Any setting with string import paths will be automatically resolved and return the class, rather than the string literal.

Example Response

```

HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "lang": "en",
  "autoupdateInterval": 15000,
  "chartLineSettings": {
    "all_tasks": {
      "active": true
    },

```

(continues on next page)

(continued from previous page)

```

    "delay": {
        "active": true
    },
    "ok": {
        "active": true
    },
    "error": {
        "active": true
    },
    "interrupted": {
        "active": true
    },
    "offline": {
        "active": true
    }
},
"widgetSettings": {
    "pmwUsersCounter": {
        "active": true,
        "collapse": true,
        "sort": 1
    },
    "pmwProjectsCounter": {
        "active": true,
        "collapse": true,
        "sort": 1
    },
    "pmwTemplatesCounter": {
        "active": true,
        "collapse": true,
        "sort": 1
    },
    "pmwInventoriesCounter": {
        "active": true,
        "collapse": true,
        "sort": 1
    },
    "pmwGroupsCounter": {
        "active": true,
        "collapse": true,
        "sort": 1
    },
    "pmwHostsCounter": {
        "active": true,
        "collapse": true,
        "sort": 1
    },
    "pmwChartWidget": {
        "active": true,
        "collapse": true,
        "sort": 1
    }
},
"selectedSkin": "example selectedSkin",
"skinsSettings": {}
}

```

JSON Parameters

- **lang** (*string*) – Lang(enum=['en', 'ru'], default=en)
- **autoupdateInterval** (*integer*) – Autoupdateinterval(default=15000)
- **chartLineSettings** (*object*) – (*ChartLineSettings Schema*)
- **widgetSettings** (*object*) – (*WidgetSettings Schema*)
- **selectedSkin** (*string*) – Selectedskin(minLength=1)
- **skinsSettings** (*object*) – (*Data Schema*)

param integer id(required) A unique integer value identifying this user group.

param integer user_id(required)

A unique integer value identifying instance of this users sublist.

query schema data(required) *UserSettings Schema*

DELETE /team/{id}/user/{user_id}/settings/

from rest_framework.settings import api_settings print(api_settings.DEFAULT_RENDERER_CLASSES)

Any setting with string import paths will be automatically resolved and return the class, rather than the string literal.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{ }
```

param integer id(required) A unique integer value identifying this user group.

param integer user_id(required) A unique integer value identifying instance of this users sublist.

GET /user/

Return all users.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "count": 2,
  "next": "http://localhost:8080/user/?limit=1&offset=1",
  "previous": null,
  "results": [
    {
      "id": 1,
      "username": "example username",
      "is_active": true,
      "email": "example@mail.com"
    }
  ]
}
```

JSON Parameters

- **count** (*integer*) – None
- **next** (*uri*) – (x-nullable=True)
- **previous** (*uri*) – (x-nullable=True)
- **results** (*array*) – (*User Schema*)

query string id A unique integer value (or comma separated list) identifying this instance.

query string username A name string value (or comma separated list) of instance.

query string is_active Boolean value meaning status of user.

query string first_name Users first name.

query string last_name Users last name.

query string email Users e-mail value.

query string id__not A unique integer value (or comma separated list) identifying this instance.

query string username__not A name string value (or comma separated list) of instance.

query string ordering Which field to use when ordering the results.

query integer limit Number of results to return per page.

query integer offset The initial index from which to return the results.

POST /user/

Create a new user.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "username": "example username",
  "is_active": true,
  "is_staff": true,
  "first_name": "example first_name",
  "last_name": "example last_name",
  "email": "example@mail.com",
  "password": "example password",
  "password2": "example password2"
}
```

JSON Parameters

- **id** (*integer*) – ID(readOnly=True)
- **username** (*string*) – Username(description=Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only., pattern=^[w.@+-]+\$, maxLength=150, minLength=1)
- **is_active** (*boolean*) – Is active(default=True)
- **is_staff** (*boolean*) – Is staff(default=False)
- **first_name** (*string*) – First name(maxLength=30)

- **last_name** (*string*) – Last name(maxLength=150)
- **email** (*email*) – Email(minLength=1)
- **password** (*string*) – Password(minLength=1)
- **password2** (*string*) – Repeat password(minLength=1)

Query Parameters

- **data(required)** (*schema*) – *CreateUser Schema*

GET /user/{id}/

Return a user instance.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "username": "example username",
  "is_active": true,
  "first_name": "example first_name",
  "last_name": "example last_name",
  "email": "example@mail.com"
}
```

JSON Parameters

- **id** (*integer*) – ID(readOnly=True)
- **username** (*string*) – Username(description=Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only., pattern=^[w.@+-]+\$, maxLength=150, minLength=1)
- **is_active** (*boolean*) – Is active(default=True)
- **first_name** (*string*) – First name(maxLength=30)
- **last_name** (*string*) – Last name(maxLength=150)
- **email** (*email*) – Email(minLength=1)

param integer id(required) A unique integer value identifying this user.

PUT /user/{id}/

Update a user.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "username": "example username",
  "is_active": true,
```

(continues on next page)

(continued from previous page)

```

"first_name": "example first_name",
"last_name": "example last_name",
"email": "example@mail.com"
}

```

JSON Parameters

- **id** (*integer*) – ID(readOnly=True)
- **username** (*string*) – Username(description=Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only., pattern=^[w.@+-]+\$, maxLength=150, minLength=1)
- **is_active** (*boolean*) – Is active(default=True)
- **first_name** (*string*) – First name(maxLength=30)
- **last_name** (*string*) – Last name(maxLength=150)
- **email** (*email*) – Email(minLength=1)

param integer id(required)

A unique integer value identifying this user.

query schema data(required) *OneUser Schema*

PATCH /user/{id}/

Update one or more fields on an existing user.

Example Response

```

HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "id": 1,
  "username": "example username",
  "is_active": true,
  "first_name": "example first_name",
  "last_name": "example last_name",
  "email": "example@mail.com"
}

```

JSON Parameters

- **id** (*integer*) – ID(readOnly=True)
- **username** (*string*) – Username(description=Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only., pattern=^[w.@+-]+\$, maxLength=150, minLength=1)
- **is_active** (*boolean*) – Is active(default=True)
- **first_name** (*string*) – First name(maxLength=30)
- **last_name** (*string*) – Last name(maxLength=150)
- **email** (*email*) – Email(minLength=1)

param integer id(required)

A unique integer value identifying this user.

query schema data(required) *OneUser Schema*

DELETE /user/{id}/

Remove an existing user.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{}
```

param integer id(required) A unique integer value identifying this user.

POST /user/{id}/change_password/

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "old_password": "example old_password",
  "password": "example password",
  "password2": "example password2"
}
```

JSON Parameters

- **old_password** (*string*) – Old password(minLength=1)
- **password** (*string*) – New password(minLength=1)
- **password2** (*string*) – Confirm new password(minLength=1)

param integer id(required)

A unique integer value identifying this user.

query schema data(required) *ChangePassword Schema*

POST /user/{id}/copy/

Endpoint which copy instance with deps.

Example Response

```
HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "id": 1,
  "username": "example username",
  "is_active": true,
  "email": "example@mail.com"
}
```

JSON Parameters

- **id** (*integer*) – ID(readOnly=True)
- **username** (*string*) – Username(description=Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only., pattern=^[w.@+-]+\$, maxLength=150, minLength=1)
- **is_active** (*boolean*) – Is active(default=True)
- **email** (*email*) – Email(minLength=1)

param integer id(required)

A unique integer value identifying this user.

query schema data(required) *User Schema*

GET /user/{id}/settings/

Return user settings.

Example Response

```
HTTP/1.1 200 OK
Vary: OK
Content-Type: application/json

{
  "lang": "en",
  "autoupdateInterval": 15000,
  "chartLineSettings": {
    "all_tasks": {
      "active": true
    },
    "delay": {
      "active": true
    },
    "ok": {
      "active": true
    },
    "error": {
      "active": true
    },
    "interrupted": {
      "active": true
    },
    "offline": {
      "active": true
    }
  },
  "widgetSettings": {
    "pmwUsersCounter": {
      "active": true,
      "collapse": true,
      "sort": 1
    },
    "pmwProjectsCounter": {
      "active": true,
      "collapse": true,
      "sort": 1
    },
    "pmwTemplatesCounter": {
      "active": true,
```

(continues on next page)

(continued from previous page)

```

        "collapse": true,
        "sort": 1
    },
    "pmwInventoriesCounter": {
        "active": true,
        "collapse": true,
        "sort": 1
    },
    "pmwGroupsCounter": {
        "active": true,
        "collapse": true,
        "sort": 1
    },
    "pmwHostsCounter": {
        "active": true,
        "collapse": true,
        "sort": 1
    },
    "pmwChartWidget": {
        "active": true,
        "collapse": true,
        "sort": 1
    }
},
"selectedSkin": "example selectedSkin",
"skinsSettings": {}
}

```

JSON Parameters

- **lang** (*string*) – Lang(enum=['en', 'ru'], default=en)
- **autoupdateInterval** (*integer*) – Autoupdateinterval(default=15000)
- **chartLineSettings** (*object*) – (*ChartLineSettings Schema*)
- **widgetSettings** (*object*) – (*WidgetSettings Schema*)
- **selectedSkin** (*string*) – Selectedskin(minLength=1)
- **skinsSettings** (*object*) – (*Data Schema*)

param integer id(required) A unique integer value identifying this user.

POST /user/{id}/settings/

Return user settings.

Example Response

```

HTTP/1.1 201 Created
Vary: CREATED
Content-Type: application/json

{
  "lang": "en",
  "autoupdateInterval": 15000,
  "chartLineSettings": {
    "all_tasks": {

```

(continues on next page)

(continued from previous page)

```

        "active": true
    },
    "delay": {
        "active": true
    },
    "ok": {
        "active": true
    },
    "error": {
        "active": true
    },
    "interrupted": {
        "active": true
    },
    "offline": {
        "active": true
    }
},
"widgetSettings": {
    "pmwUsersCounter": {
        "active": true,
        "collapse": true,
        "sort": 1
    },
    "pmwProjectsCounter": {
        "active": true,
        "collapse": true,
        "sort": 1
    },
    "pmwTemplatesCounter": {
        "active": true,
        "collapse": true,
        "sort": 1
    },
    "pmwInventoriesCounter": {
        "active": true,
        "collapse": true,
        "sort": 1
    },
    "pmwGroupsCounter": {
        "active": true,
        "collapse": true,
        "sort": 1
    },
    "pmwHostsCounter": {
        "active": true,
        "collapse": true,
        "sort": 1
    },
    "pmwChartWidget": {
        "active": true,
        "collapse": true,
        "sort": 1
    }
},
"selectedSkin": "example selectedSkin",
"skinsSettings": {}

```

(continues on next page)

(continued from previous page)

}

JSON Parameters

- **lang** (*string*) – Lang(enum=['en', 'ru'], default=en)
- **autoupdateInterval** (*integer*) – Autoupdateinterval(default=15000)
- **chartLineSettings** (*object*) – (*ChartLineSettings Schema*)
- **widgetSettings** (*object*) – (*WidgetSettings Schema*)
- **selectedSkin** (*string*) – Selectedskin(minLength=1)
- **skinsSettings** (*object*) – (*Data Schema*)

param integer id(required)

A unique integer value identifying this user.

query schema data(required) *UserSettings Schema***DELETE /user/{id}/settings/**

Return user settings.

Example Response

```
HTTP/1.1 204 No Content
Vary: NO CONTENT
Content-Type: application/json

{}
```

param integer id(required) A unique integer value identifying this user.**Schemas Description****ProjectTemplate Schema**

```
{
  "id": {
    "title": "Id",
    "type": "integer"
  },
  "name": {
    "title": "Name",
    "type": "string",
    "maxLength": 1024,
    "minLength": 1
  },
  "type": {
    "title": "Type",
    "type": "string",
    "enum": [
      "GIT",
      "TAR"
    ]
  }
}
```

(continues on next page)

(continued from previous page)

```

    }
}

```

OneProjectTemplate Schema

```

{
  "id": {
    "title": "Id",
    "type": "integer"
  },
  "name": {
    "title": "Name",
    "type": "string",
    "maxLength": 1024,
    "minLength": 1
  },
  "description": {
    "title": "Description",
    "type": "string",
    "minLength": 1
  },
  "type": {
    "title": "Type",
    "type": "string",
    "enum": [
      "GIT",
      "TAR"
    ]
  },
  "repository": {
    "title": "Repository",
    "type": "string",
    "maxLength": 2048,
    "minLength": 1
  }
}

```

ProjectTemplateCreate Schema

```

{
  "project_id": {
    "title": "Project id",
    "type": "integer",
    "readOnly": true,
    "additionalProperties": {
      "redirect": true
    },
    "x-nullable": true
  },
  "name": {
    "title": "Name",
    "type": "string",

```

(continues on next page)

(continued from previous page)

```
        "minLength": 1
    }
}
```

Group Schema

```
{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "name": {
    "title": "Name",
    "type": "string",
    "maxLength": 512,
    "minLength": 1
  },
  "children": {
    "title": "Children",
    "type": "boolean",
    "readOnly": true
  },
  "from_project": {
    "title": "Project Based",
    "type": "boolean",
    "readOnly": true
  }
}
```

User Schema

```
{
  "id": {
    "title": "ID",
    "type": "integer",
    "readOnly": true
  },
  "username": {
    "title": "Username",
    "description": "Required. 150 characters or fewer. Letters, digits, ↵
↳and @/./+/-/_ only.",
    "type": "string",
    "pattern": "^[\\w.\\@+-]+$",
    "maxLength": 150,
    "minLength": 1
  },
  "is_active": {
    "title": "Is active",
    "type": "boolean",
    "default": true
  },
}
```

(continues on next page)

(continued from previous page)

```

    "email": {
      "title": "Email",
      "type": "string",
      "format": "email",
      "minLength": 1
    }
  }
}

```

GroupCreateMaster Schema

```

{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "name": {
    "title": "Name",
    "type": "string",
    "maxLength": 512,
    "minLength": 1
  },
  "notes": {
    "title": "Notes",
    "type": "string",
    "format": "textarea"
  },
  "children": {
    "title": "Contains groups",
    "type": "boolean",
    "default": false
  },
  "owner": {
    "id": {
      "title": "ID",
      "type": "integer",
      "readOnly": true
    },
    "username": {
      "title": "Username",
      "description": "Required. 150 characters or fewer. Letters, ↵
↵digits and @/./+/-/_ only.",
      "type": "string",
      "pattern": "^[\\w. @+-]+$",
      "maxLength": 150,
      "minLength": 1
    },
    "is_active": {
      "title": "Is active",
      "type": "boolean",
      "default": true
    },
    "email": {
      "title": "Email",

```

(continues on next page)

(continued from previous page)

```

        "type": "string",
        "format": "email",
        "minLength": 1
    }
}
}

```

OneGroup Schema

```

{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "name": {
    "title": "Name",
    "type": "string",
    "maxLength": 512,
    "minLength": 1
  },
  "notes": {
    "title": "Notes",
    "type": "string",
    "format": "textarea"
  },
  "children": {
    "title": "Children",
    "type": "boolean",
    "readOnly": true
  },
  "owner": {
    "id": {
      "title": "ID",
      "type": "integer",
      "readOnly": true
    },
    "username": {
      "title": "Username",
      "description": "Required. 150 characters or fewer. Letters, ↵
↳digits and @/./+/-/_ only.",
      "type": "string",
      "pattern": "^[\\w.@+-]+$",
      "maxLength": 150,
      "minLength": 1
    },
    "is_active": {
      "title": "Is active",
      "type": "boolean",
      "default": true
    },
    "email": {
      "title": "Email",
      "type": "string",

```

(continues on next page)

(continued from previous page)

```

        "format": "email",
        "minLength": 1
    }
}
}

```

SetOwner Schema

```

{
  "user_id": {
    "title": "New owner",
    "type": "integer",
    "format": "fk",
    "additionalProperties": {
      "model": {
        "id": {
          "title": "ID",
          "type": "integer",
          "readOnly": true
        },
        "username": {
          "title": "Username",
          "description": "Required. 150 characters or fewer.␣
↪Letters, digits and @/./+/-/_ only.",
          "type": "string",
          "pattern": "^[\\w.\\@+-]+$",
          "maxLength": 150,
          "minLength": 1
        },
        "is_active": {
          "title": "Is active",
          "type": "boolean",
          "default": true
        },
        "email": {
          "title": "Email",
          "type": "string",
          "format": "email",
          "minLength": 1
        }
      },
      "value_field": "id",
      "view_field": "username"
    }
  }
}

```

InventoryVariable Schema

```
{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "key": {
    "title": "Key",
    "type": "string",
    "format": "autocomplete",
    "enum": [
      "ansible_host",
      "ansible_port",
      "ansible_user",
      "ansible_connection",
      "ansible_ssh_pass",
      "ansible_ssh_private_key_file",
      "ansible_ssh_common_args",
      "ansible_sftp_extra_args",
      "ansible_scp_extra_args",
      "ansible_ssh_extra_args",
      "ansible_ssh_executable",
      "ansible_ssh_pipelining",
      "ansible_become",
      "ansible_become_method",
      "ansible_become_user",
      "ansible_become_pass",
      "ansible_become_exe",
      "ansible_become_flags",
      "ansible_shell_type",
      "ansible_python_interpreter",
      "ansible_ruby_interpreter",
      "ansible_perl_interpreter",
      "ansible_shell_executable"
    ]
  },
  "value": {
    "title": "Value",
    "type": "string",
    "format": "dynamic",
    "additionalProperties": {
      "choices": {},
      "field": "key",
      "types": {
        "ansible_become": "boolean",
        "ansible_become_pass": "password",
        "ansible_port": "integer",
        "ansible_ssh_pass": "password",
        "ansible_ssh_private_key_file": "secretfile"
      }
    }
  }
}
```

Host Schema

```
{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "name": {
    "title": "Name",
    "type": "string",
    "maxLength": 512,
    "minLength": 1
  },
  "type": {
    "title": "Type",
    "type": "string",
    "enum": [
      "HOST",
      "RANGE"
    ],
    "default": "HOST"
  },
  "from_project": {
    "title": "Project Based",
    "type": "boolean",
    "readOnly": true
  }
}
```

OneHost Schema

```
{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "name": {
    "title": "Name",
    "type": "string",
    "maxLength": 512,
    "minLength": 1
  },
  "notes": {
    "title": "Notes",
    "type": "string",
    "format": "textarea"
  },
  "type": {
    "title": "Type",
    "type": "string",
    "enum": [
      "HOST",
      "RANGE"
    ]
  }
}
```

(continues on next page)

(continued from previous page)

```

    ],
    "default": "HOST"
  },
  "owner": {
    "id": {
      "title": "ID",
      "type": "integer",
      "readOnly": true
    },
    "username": {
      "title": "Username",
      "description": "Required. 150 characters or fewer. Letters, ↵
↳ digits and @/./+/-/_ only.",
      "type": "string",
      "pattern": "^[\\w. @+-]+$",
      "maxLength": 150,
      "minLength": 1
    },
    "is_active": {
      "title": "Is active",
      "type": "boolean",
      "default": true
    },
    "email": {
      "title": "Email",
      "type": "string",
      "format": "email",
      "minLength": 1
    }
  }
}

```

History Schema

```

{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "start_time": {
    "title": "Start time",
    "type": "string",
    "format": "date-time"
  },
  "executor": {
    "title": "Executor",
    "type": "integer",
    "x-nullable": true
  },
  "initiator": {
    "title": "Initiator",
    "type": "integer"
  },
}

```

(continues on next page)

(continued from previous page)

```
"initiator_type": {
  "title": "Initiator type",
  "type": "string",
  "maxLength": 50,
  "minLength": 1
},
"project": {
  "title": "Project",
  "type": "integer",
  "x-nullable": true
},
"inventory": {
  "title": "Inventory",
  "type": "integer",
  "x-nullable": true
},
"kind": {
  "title": "Kind",
  "type": "string",
  "maxLength": 50,
  "minLength": 1
},
"mode": {
  "title": "Mode",
  "type": "string",
  "maxLength": 256,
  "minLength": 1
},
"options": {
  "title": "Options",
  "type": "string",
  "readOnly": true
},
"status": {
  "title": "Status",
  "type": "string",
  "enum": [
    "DELAY",
    "RUN",
    "OK",
    "ERROR",
    "OFFLINE",
    "INTERRUPTED"
  ]
},
"stop_time": {
  "title": "Stop time",
  "type": "string",
  "format": "date-time",
  "x-nullable": true
}
}
```

OneHistory Schema

```
{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "status": {
    "title": "Status",
    "type": "string",
    "enum": [
      "DELAY",
      "RUN",
      "OK",
      "ERROR",
      "OFFLINE",
      "INTERRUPTED"
    ]
  },
  "executor": {
    "title": "Executor",
    "type": "integer",
    "x-nullable": true
  },
  "project": {
    "title": "Project",
    "type": "integer",
    "x-nullable": true
  },
  "revision": {
    "title": "Revision",
    "type": "string",
    "maxLength": 256,
    "x-nullable": true
  },
  "inventory": {
    "title": "Inventory",
    "type": "integer",
    "x-nullable": true
  },
  "kind": {
    "title": "Kind",
    "type": "string",
    "maxLength": 50,
    "minLength": 1
  },
  "mode": {
    "title": "Mode",
    "type": "string",
    "maxLength": 256,
    "minLength": 1
  },
  "execute_args": {
    "title": "Execute args",
    "type": "string",
    "readOnly": true
  }
}
```

(continues on next page)

(continued from previous page)

```
},
"execution_time": {
  "title": "Execution time",
  "type": "integer",
  "format": "uptime"
},
"start_time": {
  "title": "Start time",
  "type": "string",
  "format": "date-time"
},
"stop_time": {
  "title": "Stop time",
  "type": "string",
  "format": "date-time",
  "x-nullable": true
},
"initiator": {
  "title": "Initiator",
  "type": "integer"
},
"initiator_type": {
  "title": "Initiator type",
  "type": "string",
  "maxLength": 50,
  "minLength": 1
},
"options": {
  "title": "Options",
  "type": "string",
  "readOnly": true
},
"raw_args": {
  "title": "Raw args",
  "type": "string",
  "minLength": 1
},
"raw_stdout": {
  "title": "Raw stdout",
  "type": "string",
  "readOnly": true
},
"raw_inventory": {
  "title": "Raw inventory",
  "type": "string",
  "minLength": 1
}
}
```

Empty Schema

```
{}
```

ActionResponse Schema

```
{
  "detail": {
    "title": "Detail",
    "type": "string",
    "minLength": 1
  }
}
```

Data Schema

```
{}
```

Hook Schema

```
{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "name": {
    "title": "Name",
    "type": "string",
    "maxLength": 512,
    "minLength": 1
  },
  "type": {
    "title": "Type",
    "type": "string",
    "enum": [
      "HTTP",
      "SCRIPT"
    ]
  },
  "when": {
    "title": "When",
    "type": "string",
    "enum": [
      "on_execution",
      "after_execution",
      "on_user_add",
      "on_user_upd",
      "on_user_del",
      "on_object_add",

```

(continues on next page)

(continued from previous page)

```

        "on_object_upd",
        "on_object_del"
    ],
    "enable": {
        "title": "Enable",
        "type": "boolean"
    },
    "recipients": {
        "title": "Recipients",
        "type": "string",
        "minLength": 1
    }
}

```

Inventory Schema

```

{
    "id": {
        "title": "Id",
        "type": "integer",
        "readOnly": true
    },
    "name": {
        "title": "Name",
        "type": "string",
        "maxLength": 512,
        "minLength": 1
    },
    "from_project": {
        "title": "Project Based",
        "type": "boolean",
        "readOnly": true
    }
}

```

OneInventory Schema

```

{
    "id": {
        "title": "Id",
        "type": "integer",
        "readOnly": true
    },
    "name": {
        "title": "Name",
        "type": "string",
        "maxLength": 512,
        "minLength": 1
    },
    "notes": {
        "title": "Notes",

```

(continues on next page)

(continued from previous page)

```

        "type": "string",
        "format": "textarea"
    },
    "owner": {
        "id": {
            "title": "ID",
            "type": "integer",
            "readOnly": true
        },
        "username": {
            "title": "Username",
            "description": "Required. 150 characters or fewer. Letters, ↵
↵digits and @/./+/-/_ only.",
            "type": "string",
            "pattern": "^[\\w. @+-]+$",
            "maxLength": 150,
            "minLength": 1
        },
        "is_active": {
            "title": "Is active",
            "type": "boolean",
            "default": true
        },
        "email": {
            "title": "Email",
            "type": "string",
            "format": "email",
            "minLength": 1
        }
    }
}

```

InventoryImport Schema

```

{
    "inventory_id": {
        "title": "Inventory id",
        "type": "integer",
        "readOnly": true,
        "additionalProperties": {
            "redirect": true
        },
        "x-nullable": true
    },
    "name": {
        "title": "Name",
        "type": "string",
        "minLength": 1
    },
    "raw_data": {
        "title": "Raw data",
        "type": "string",
        "minLength": 1
    }
}

```

Project Schema

```
{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "name": {
    "title": "Name",
    "type": "string",
    "maxLength": 512,
    "minLength": 1
  },
  "type": {
    "title": "Type",
    "type": "string",
    "readOnly": true,
    "minLength": 1
  },
  "status": {
    "title": "Status",
    "type": "string",
    "enum": [
      "NEW",
      "ERROR",
      "OK",
      "WAIT_SYNC",
      "SYNC"
    ],
    "readOnly": true
  }
}
```

ProjectCreateMaster Schema

```
{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "name": {
    "title": "Name",
    "type": "string",
    "maxLength": 512,
    "minLength": 1
  },
  "status": {
    "title": "Status",
    "type": "string",
    "readOnly": true,
    "minLength": 1
  },
  "type": {
```

(continues on next page)

(continued from previous page)

```

        "title": "Repo type",
        "type": "string",
        "enum": [
            "MANUAL",
            "GIT",
            "TAR"
        ],
        "default": "MANUAL"
    },
    "repository": {
        "title": "Repo url",
        "type": "string",
        "default": "MANUAL",
        "minLength": 1
    },
    "repo_auth": {
        "title": "Repo auth type",
        "type": "string",
        "format": "dynamic",
        "default": "NONE",
        "additionalProperties": {
            "choices": {
                "GIT": [
                    "NONE",
                    "KEY",
                    "PASSWORD"
                ]
            },
            "field": "type",
            "types": {
                "GIT": "string",
                "MANUAL": "hidden",
                "TAR": "hidden"
            }
        }
    },
    "auth_data": {
        "title": "Repo auth data",
        "type": "string",
        "format": "dynamic",
        "default": "",
        "additionalProperties": {
            "choices": {},
            "field": "repo_auth",
            "types": {
                "KEY": "secretfile",
                "NONE": "hidden",
                "PASSWORD": "password"
            }
        }
    },
    "branch": {
        "title": "Branch for GIT(branch/tag/SHA) or TAR(subdir)",
        "type": "string",
        "format": "dynamic",
        "additionalProperties": {
            "choices": {},

```

(continues on next page)

(continued from previous page)

```

        "field": "type",
        "types": {
            "GIT": "string",
            "MANUAL": "hidden",
            "TAR": "string"
        },
        "x-nullable": true
    },
    "additional_playbook_path": {
        "title": "Directory with playbooks",
        "type": "string",
        "minLength": 1,
        "x-nullable": true
    }
}

```

OneProject Schema

```

{
    "id": {
        "title": "Id",
        "type": "integer",
        "readOnly": true
    },
    "name": {
        "title": "Name",
        "type": "string",
        "maxLength": 512,
        "minLength": 1
    },
    "repository": {
        "title": "Repository",
        "type": "string",
        "default": "MANUAL",
        "minLength": 1
    },
    "status": {
        "title": "Status",
        "type": "string",
        "enum": [
            "NEW",
            "ERROR",
            "OK",
            "WAIT_SYNC",
            "SYNC"
        ],
        "readOnly": true
    },
    "revision": {
        "title": "Revision",
        "type": "string",
        "readOnly": true
    },
}

```

(continues on next page)

(continued from previous page)

```

    "branch": {
        "title": "Branch",
        "type": "string",
        "readOnly": true
    },
    "owner": {
        "id": {
            "title": "ID",
            "type": "integer",
            "readOnly": true
        },
        "username": {
            "title": "Username",
            "description": "Required. 150 characters or fewer. Letters, ↵
↵digits and @/./+/-/_ only.",
            "type": "string",
            "pattern": "^[\\w. @+-]+$",
            "maxLength": 150,
            "minLength": 1
        },
        "is_active": {
            "title": "Is active",
            "type": "boolean",
            "default": true
        },
        "email": {
            "title": "Email",
            "type": "string",
            "format": "email",
            "minLength": 1
        }
    },
    "notes": {
        "title": "Notes",
        "type": "string",
        "format": "textarea"
    },
    "readme_content": {
        "title": "Information",
        "type": "string",
        "format": "html",
        "readOnly": true
    },
    "execute_view_data": {}
}

```

AnsibleModule Schema

```

{
  "module": {
    "title": "Module",
    "type": "string",
    "format": "autocomplete",
    "additionalProperties": {
      "model": {
        "id": {
          "title": "Id",
          "type": "integer",
          "readOnly": true
        },
        "path": {
          "title": "Path",
          "type": "string",
          "maxLength": 1024,
          "minLength": 1
        },
        "name": {
          "title": "Name",
          "type": "string",
          "readOnly": true
        }
      },
      "value_field": "name",
      "view_field": "path"
    }
  },
  "args": {
    "title": "Args",
    "description": "host pattern",
    "type": "string"
  },
  "background": {
    "title": "Background",
    "description": "run asynchronously, failing after X seconds",
    ↪(default=N/A)",
    "type": "integer"
  },
  "become": {
    "title": "Become",
    "description": "run operations with become (does not imply password",
    ↪prompting)",
    "type": "boolean",
    "default": false
  },
  "become_method": {
    "title": "Become method",
    "description": "privilege escalation method to use (default=sudo),",
    ↪use `ansible-doc -t become -l` to list valid choices.",
    "type": "string"
  },
  "become_user": {
    "title": "Become user",
    "description": "run operations as this user (default=root)",

```

(continues on next page)

(continued from previous page)

```

        "type": "string"
    },
    "check": {
        "title": "Check",
        "description": "don't make any changes; instead, try to predict some_
↪of the changes that may occur",
        "type": "boolean",
        "default": false
    },
    "connection": {
        "title": "Connection",
        "description": "connection type to use (default=smart)",
        "type": "string"
    },
    "diff": {
        "title": "Diff",
        "description": "when changing (small) files and templates, show the_
↪differences in those files; works great with --check",
        "type": "boolean",
        "default": false
    },
    "extra_vars": {
        "title": "Extra vars",
        "description": "set additional variables as key=value or YAML/JSON,
↪if filename prepend with @",
        "type": "string"
    },
    "forks": {
        "title": "Forks",
        "description": "specify number of parallel processes to use_
↪(default=5)",
        "type": "integer"
    },
    "inventory": {
        "title": "Inventory",
        "description": "specify inventory host path or comma separated host_
↪list. --inventory-file is deprecated",
        "type": "string",
        "format": "autocomplete",
        "additionalProperties": {
            "model": {
                "id": {
                    "title": "Id",
                    "type": "integer",
                    "readOnly": true
                },
                "name": {
                    "title": "Name",
                    "type": "string",
                    "maxLength": 512,
                    "minLength": 1
                },
                "from_project": {
                    "title": "Project Based",
                    "type": "boolean",
                    "readOnly": true
                }
            }
        }
    }

```

(continues on next page)

(continued from previous page)

```

        },
        "value_field": "id",
        "view_field": "name"
    },
    },
    "limit": {
        "title": "Limit",
        "description": "further limit selected hosts to an additional pattern
↪",
        "type": "string"
    },
    "list_hosts": {
        "title": "List hosts",
        "description": "outputs a list of matching hosts; does not execute_
↪anything else",
        "type": "boolean",
        "default": false
    },
    "module_path": {
        "title": "Module path",
        "description": "prepend colon-separated path(s) to module library_
↪(default=~/.ansible/plugins/modules:/usr/share/ansible/plugins/modules)",
        "type": "string"
    },
    "one_line": {
        "title": "One line",
        "description": "condense output",
        "type": "boolean",
        "default": false
    },
    "playbook_dir": {
        "title": "Playbook dir",
        "description": "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.",
        "type": "string"
    },
    "poll": {
        "title": "Poll",
        "description": "set the poll interval if using -B (default=15)",
        "type": "integer"
    },
    "private_key": {
        "title": "Private key",
        "description": "use this file to authenticate the connection",
        "type": "string",
        "format": "secretfile"
    },
    "scp_extra_args": {
        "title": "Scp extra args",
        "description": "specify extra arguments to pass to scp only (e.g. -l)
↪",
        "type": "string"
    },
    "sftp_extra_args": {
        "title": "Sftp extra args",
        "description": "specify extra arguments to pass to sftp only (e.g. -
↪f, -l)",

```

(continues on next page)

(continued from previous page)

```

        "type": "string"
    },
    "ssh_common_args": {
        "title": "Ssh common args",
        "description": "specify common arguments to pass to sftp/scp/ssh (e.
↪g. ProxyCommand)",
        "type": "string"
    },
    "ssh_extra_args": {
        "title": "Ssh extra args",
        "description": "specify extra arguments to pass to ssh only (e.g. -R)
↪",
        "type": "string"
    },
    "syntax_check": {
        "title": "Syntax check",
        "description": "perform a syntax check on the playbook, but do not_
↪execute it",
        "type": "boolean",
        "default": false
    },
    "timeout": {
        "title": "Timeout",
        "description": "override the connection timeout in seconds_
↪(default=10)",
        "type": "integer"
    },
    "tree": {
        "title": "Tree",
        "description": "log output to this directory",
        "type": "string"
    },
    "user": {
        "title": "User",
        "description": "connect as this user (default=None)",
        "type": "string"
    },
    "vault_id": {
        "title": "Vault id",
        "description": "the vault identity to use",
        "type": "string"
    },
    "vault_password_file": {
        "title": "Vault password file",
        "description": "vault password file",
        "type": "string",
        "format": "secretfile"
    },
    "verbose": {
        "title": "Verbose",
        "description": "verbose mode (-vvv for more, -vvvv to enable_
↪connection debugging)",
        "type": "integer",
        "default": 0,
        "maximum": 4
    },
    "group": {

```

(continues on next page)

(continued from previous page)

```
    "title": "Group",
    "type": "string",
    "default": "all"
  }
}
```

ExecuteResponse Schema

```
{
  "detail": {
    "title": "Detail",
    "type": "string",
    "minLength": 1
  },
  "history_id": {
    "title": "History id",
    "type": "integer",
    "additionalProperties": {
      "redirect": true
    },
    "x-nullable": true
  },
  "executor": {
    "title": "Executor",
    "type": "integer",
    "x-nullable": true
  }
}
```

AnsiblePlaybook Schema

```
{
  "playbook": {
    "title": "Playbook",
    "type": "string",
    "format": "autocomplete",
    "additionalProperties": {
      "model": {
        "id": {
          "title": "Id",
          "type": "integer",
          "readOnly": true
        },
        "name": {
          "title": "Name",
          "type": "string",
          "maxLength": 251,
          "minLength": 1
        },
        "playbook": {
          "title": "Playbook",
          "type": "string",
```

(continues on next page)

(continued from previous page)

```

        "maxLength": 256,
        "minLength": 1
    },
    },
    "value_field": "playbook",
    "view_field": "name"
}
},
"args": {
    "title": "Args",
    "description": "Playbook(s)",
    "type": "string"
},
"become": {
    "title": "Become",
    "description": "run operations with become (does not imply password_
↪prompting)",
    "type": "boolean",
    "default": false
},
"become_method": {
    "title": "Become method",
    "description": "privilege escalation method to use (default=sudo),
↪use `ansible-doc -t become -l` to list valid choices.",
    "type": "string"
},
"become_user": {
    "title": "Become user",
    "description": "run operations as this user (default=root)",
    "type": "string"
},
"check": {
    "title": "Check",
    "description": "don't make any changes; instead, try to predict some_
↪of the changes that may occur",
    "type": "boolean",
    "default": false
},
"connection": {
    "title": "Connection",
    "description": "connection type to use (default=smart)",
    "type": "string"
},
"diff": {
    "title": "Diff",
    "description": "when changing (small) files and templates, show the_
↪differences in those files; works great with --check",
    "type": "boolean",
    "default": false
},
"extra_vars": {
    "title": "Extra vars",
    "description": "set additional variables as key=value or YAML/JSON,
↪if filename prepend with @",
    "type": "string"
},
"flush_cache": {

```

(continues on next page)

(continued from previous page)

```

        "title": "Flush cache",
        "description": "clear the fact cache for every host in inventory",
        "type": "boolean",
        "default": false
    },
    "force_handlers": {
        "title": "Force handlers",
        "description": "run handlers even if a task fails",
        "type": "boolean",
        "default": false
    },
    "forks": {
        "title": "Forks",
        "description": "specify number of parallel processes to use_
↪ (default=5)",
        "type": "integer"
    },
    "inventory": {
        "title": "Inventory",
        "description": "specify inventory host path or comma separated host_
↪ list. --inventory-file is deprecated",
        "type": "string",
        "format": "autocomplete",
        "additionalProperties": {
            "model": {
                "id": {
                    "title": "Id",
                    "type": "integer",
                    "readOnly": true
                },
                "name": {
                    "title": "Name",
                    "type": "string",
                    "maxLength": 512,
                    "minLength": 1
                },
                "from_project": {
                    "title": "Project Based",
                    "type": "boolean",
                    "readOnly": true
                }
            },
            "value_field": "id",
            "view_field": "name"
        }
    },
    "limit": {
        "title": "Limit",
        "description": "further limit selected hosts to an additional pattern
↪",
        "type": "string"
    },
    "list_hosts": {
        "title": "List hosts",
        "description": "outputs a list of matching hosts; does not execute_
↪ anything else",
        "type": "boolean",

```

(continues on next page)

(continued from previous page)

```

        "default": false
    },
    "list_tags": {
        "title": "List tags",
        "description": "list all available tags",
        "type": "boolean",
        "default": false
    },
    "list_tasks": {
        "title": "List tasks",
        "description": "list all tasks that would be executed",
        "type": "boolean",
        "default": false
    },
    "module_path": {
        "title": "Module path",
        "description": "prepend colon-separated path(s) to module library_
↪ (default=~/.ansible/plugins/modules:/usr/share/ansible/plugins/modules)",
        "type": "string"
    },
    "private_key": {
        "title": "Private key",
        "description": "use this file to authenticate the connection",
        "type": "string",
        "format": "secretfile"
    },
    "scp_extra_args": {
        "title": "Scp extra args",
        "description": "specify extra arguments to pass to scp only (e.g. -l)
↪ ",
        "type": "string"
    },
    "sftp_extra_args": {
        "title": "Sftp extra args",
        "description": "specify extra arguments to pass to sftp only (e.g. -
↪ f, -l)",
        "type": "string"
    },
    "skip_tags": {
        "title": "Skip tags",
        "description": "only run plays and tasks whose tags do not match_
↪ these values",
        "type": "string"
    },
    "ssh_common_args": {
        "title": "Ssh common args",
        "description": "specify common arguments to pass to sftp/scp/ssh (e.
↪ g. ProxyCommand)",
        "type": "string"
    },
    "ssh_extra_args": {
        "title": "Ssh extra args",
        "description": "specify extra arguments to pass to ssh only (e.g. -R)
↪ ",
        "type": "string"
    },
    "start_at_task": {

```

(continues on next page)

(continued from previous page)

```

        "title": "Start at task",
        "description": "start the playbook at the task matching this name",
        "type": "string"
    },
    "step": {
        "title": "Step",
        "description": "one-step-at-a-time: confirm each task before running
↪",
        "type": "boolean",
        "default": false
    },
    "syntax_check": {
        "title": "Syntax check",
        "description": "perform a syntax check on the playbook, but do not
↪execute it",
        "type": "boolean",
        "default": false
    },
    "tags": {
        "title": "Tags",
        "description": "only run plays and tasks tagged with these values",
        "type": "string"
    },
    "timeout": {
        "title": "Timeout",
        "description": "override the connection timeout in seconds
↪(default=10)",
        "type": "integer"
    },
    "user": {
        "title": "User",
        "description": "connect as this user (default=None)",
        "type": "string"
    },
    "vault_id": {
        "title": "Vault id",
        "description": "the vault identity to use",
        "type": "string"
    },
    "vault_password_file": {
        "title": "Vault password file",
        "description": "vault password file",
        "type": "string",
        "format": "secretfile"
    },
    "verbose": {
        "title": "Verbose",
        "description": "verbose mode (-vvv for more, -vvvv to enable
↪connection debugging)",
        "type": "integer",
        "default": 0,
        "maximum": 4
    }
}

```

ProjectHistory Schema

```
{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "start_time": {
    "title": "Start time",
    "type": "string",
    "format": "date-time"
  },
  "executor": {
    "title": "Executor",
    "type": "integer",
    "x-nullable": true
  },
  "initiator": {
    "title": "Initiator",
    "type": "integer"
  },
  "initiator_type": {
    "title": "Initiator type",
    "type": "string",
    "maxLength": 50,
    "minLength": 1
  },
  "revision": {
    "title": "Revision",
    "type": "string",
    "maxLength": 256,
    "x-nullable": true
  },
  "inventory": {
    "title": "Inventory",
    "type": "integer",
    "x-nullable": true
  },
  "kind": {
    "title": "Kind",
    "type": "string",
    "maxLength": 50,
    "minLength": 1
  },
  "mode": {
    "title": "Mode",
    "type": "string",
    "maxLength": 256,
    "minLength": 1
  },
  "options": {
    "title": "Options",
    "type": "string",
    "readOnly": true
  },
  "status": {
```

(continues on next page)

(continued from previous page)

```

        "title": "Status",
        "type": "string",
        "enum": [
            "DELAY",
            "RUN",
            "OK",
            "ERROR",
            "OFFLINE",
            "INTERRUPTED"
        ]
    },
    "stop_time": {
        "title": "Stop time",
        "type": "string",
        "format": "date-time",
        "x-nullable": true
    }
}

```

InventoryFileImport Schema

```

{
    "inventory_id": {
        "title": "Inventory id",
        "type": "integer",
        "readOnly": true,
        "additionalProperties": {
            "redirect": true
        },
        "x-nullable": true
    },
    "name": {
        "title": "Name",
        "type": "string",
        "pattern": "([^\w\d\.-]+?)|(^([\\w\d\\-\\.])?([\\w\d\\-\\.])*(\\w\\d\\-\\.)+?[\\w\d\\-\\.])?*",
        "minLength": 1
    },
    "raw_data": {
        "title": "Raw data",
        "type": "string",
        "readOnly": true,
        "minLength": 1
    }
}

```

Module Schema

```
{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "path": {
    "title": "Path",
    "type": "string",
    "maxLength": 1024,
    "minLength": 1
  },
  "name": {
    "title": "Name",
    "type": "string",
    "readOnly": true
  }
}
```

OneModule Schema

```
{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "path": {
    "title": "Path",
    "type": "string",
    "maxLength": 1024,
    "minLength": 1
  },
  "name": {
    "title": "Name",
    "type": "string",
    "readOnly": true
  },
  "data": {
    "title": "Data",
    "type": "object",
    "readOnly": true
  }
}
```

Periodictask Schema

```

{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "name": {
    "title": "Name",
    "type": "string",
    "maxLength": 512,
    "minLength": 1
  },
  "kind": {
    "title": "Task type",
    "type": "string",
    "enum": [
      "PLAYBOOK",
      "MODULE",
      "TEMPLATE"
    ],
    "default": "PLAYBOOK"
  },
  "mode": {
    "title": "Mode",
    "type": "string",
    "format": "dynamic",
    "additionalProperties": {
      "choices": {},
      "field": "kind",
      "types": {
        "MODULE": "fk_autocomplete",
        "PLAYBOOK": "fk_autocomplete",
        "TEMPLATE": "hidden"
      }
    }
  },
  "inventory": {
    "title": "Inventory",
    "type": "string",
    "format": "dynamic",
    "additionalProperties": {
      "choices": {},
      "field": "kind",
      "types": {
        "MODULE": "fk_autocomplete",
        "PLAYBOOK": "fk_autocomplete",
        "TEMPLATE": "hidden"
      }
    }
  },
  "save_result": {
    "title": "Save result",
    "type": "boolean"
  },
  "template": {

```

(continues on next page)

(continued from previous page)

```
        "title": "Template",
        "type": "integer",
        "x-nullable": true
    },
    "template_opt": {
        "title": "Template opt",
        "type": "string",
        "format": "dynamic",
        "additionalProperties": {
            "choices": {},
            "field": "kind",
            "types": {
                "MODULE": "hidden",
                "PLAYBOOK": "hidden",
                "TEMPLATE": "autocomplete"
            }
        },
        "x-nullable": true
    },
    "enabled": {
        "title": "Enabled",
        "type": "boolean"
    },
    "type": {
        "title": "Interval type",
        "type": "string",
        "enum": [
            "CRONTAB",
            "INTERVAL"
        ],
        "default": "CRONTAB"
    },
    "schedule": {
        "title": "Schedule",
        "type": "string",
        "format": "dynamic",
        "additionalProperties": {
            "choices": {},
            "field": "type",
            "types": {
                "CRONTAB": "crontab",
                "INTERVAL": "integer"
            }
        }
    }
}
```

OnePeriodictask Schema

```

{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "name": {
    "title": "Name",
    "type": "string",
    "maxLength": 512,
    "minLength": 1
  },
  "kind": {
    "title": "Task type",
    "type": "string",
    "enum": [
      "PLAYBOOK",
      "MODULE",
      "TEMPLATE"
    ],
    "default": "PLAYBOOK"
  },
  "mode": {
    "title": "Mode",
    "type": "string",
    "format": "dynamic",
    "additionalProperties": {
      "choices": {},
      "field": "kind",
      "types": {
        "MODULE": "fk_autocomplete",
        "PLAYBOOK": "fk_autocomplete",
        "TEMPLATE": "hidden"
      }
    }
  },
  "inventory": {
    "title": "Inventory",
    "type": "string",
    "format": "dynamic",
    "additionalProperties": {
      "choices": {},
      "field": "kind",
      "types": {
        "MODULE": "fk_autocomplete",
        "PLAYBOOK": "fk_autocomplete",
        "TEMPLATE": "hidden"
      }
    }
  },
  "save_result": {
    "title": "Save result",
    "type": "boolean"
  },
  "template": {

```

(continues on next page)

(continued from previous page)

```
    "title": "Template",
    "type": "integer",
    "x-nullable": true
  },
  "template_opt": {
    "title": "Template opt",
    "type": "string",
    "format": "dynamic",
    "additionalProperties": {
      "choices": {},
      "field": "kind",
      "types": {
        "MODULE": "hidden",
        "PLAYBOOK": "hidden",
        "TEMPLATE": "autocomplete"
      }
    }
  },
  "x-nullable": true
},
"enabled": {
  "title": "Enabled",
  "type": "boolean"
},
"type": {
  "title": "Interval type",
  "type": "string",
  "enum": [
    "CRONTAB",
    "INTERVAL"
  ],
  "default": "CRONTAB"
},
"schedule": {
  "title": "Schedule",
  "type": "string",
  "format": "dynamic",
  "additionalProperties": {
    "choices": {},
    "field": "type",
    "types": {
      "CRONTAB": "crontab",
      "INTERVAL": "integer"
    }
  }
},
"notes": {
  "title": "Notes",
  "type": "string",
  "format": "textarea"
}
}
```

PeriodicTaskVariable Schema

```
{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "key": {
    "title": "Key",
    "type": "string",
    "maxLength": 512,
    "minLength": 1
  },
  "value": {
    "title": "Value",
    "type": "string",
    "default": ""
  }
}
```

Playbook Schema

```
{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "name": {
    "title": "Name",
    "type": "string",
    "maxLength": 251,
    "minLength": 1
  },
  "playbook": {
    "title": "Playbook",
    "type": "string",
    "maxLength": 256,
    "minLength": 1
  }
}
```

OnePlaybook Schema

```
{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "name": {
    "title": "Name",
```

(continues on next page)

(continued from previous page)

```
    "type": "string",
    "maxLength": 251,
    "minLength": 1
  },
  "playbook": {
    "title": "Playbook",
    "type": "string",
    "readOnly": true,
    "minLength": 1
  }
}
```

Template Schema

```
{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "name": {
    "title": "Name",
    "type": "string",
    "maxLength": 512,
    "minLength": 1
  },
  "kind": {
    "title": "Type",
    "type": "string",
    "enum": [
      "Task",
      "Module"
    ],
    "default": "Task"
  },
  "data": {},
  "options": {},
  "options_list": {
    "type": "array",
    "items": {
      "type": "string"
    },
    "readOnly": true
  }
}
```


OneTemplate Schema

```
{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "name": {
    "title": "Name",
    "type": "string",
    "maxLength": 512,
    "minLength": 1
  },
  "notes": {
    "title": "Notes",
    "type": "string",
    "format": "textarea"
  },
  "kind": {
    "title": "Type",
    "type": "string",
    "enum": [
      "Task",
      "Module"
    ],
    "default": "Task"
  },
  "data": {},
  "options": {},
  "options_list": {
    "type": "array",
    "items": {
      "type": "string"
    },
    "readOnly": true
  }
}
```

TemplateExec Schema

```
{
  "option": {
    "title": "Option",
    "description": "Option name from template options.",
    "type": "string",
    "minLength": 0
  }
}
```

ProjectVariable Schema

```
{
  "id": {
    "title": "Id",
    "type": "integer",
    "readOnly": true
  },
  "key": {
    "title": "Key",
    "type": "string",
    "format": "autocomplete",
    "enum": [
      "repo_type",
      "repo_sync_on_run",
      "repo_sync_on_run_timeout",
      "repo_branch",
      "repo_password",
      "repo_key",
      "playbook_path",
      "ci_template"
    ]
  },
  "value": {
    "title": "Value",
    "type": "string",
    "format": "dynamic",
    "additionalProperties": {
      "choices": {
        "repo_sync_on_run": [
          true,
          false
        ],
        "repo_type": [
          "MANUAL",
          "GIT",
          "TAR"
        ]
      }
    },
    "field": "key",
    "types": {
      "ci_template": "fk",
      "repo_key": "secretfile",
      "repo_password": "password",
      "repo_sync_on_run_timeout": "uptime"
    }
  }
}
```

Team Schema

```
{
  "id": {
    "title": "ID",
    "type": "integer",
    "readOnly": true
  },
  "name": {
    "title": "Name",
    "type": "string",
    "maxLength": 150,
    "minLength": 1
  }
}
```

OneTeam Schema

```
{
  "id": {
    "title": "ID",
    "type": "integer",
    "readOnly": true
  },
  "name": {
    "title": "Name",
    "type": "string",
    "maxLength": 150,
    "minLength": 1
  },
  "notes": {
    "title": "Notes",
    "type": "string",
    "format": "textarea"
  },
  "owner": {
    "id": {
      "title": "ID",
      "type": "integer",
      "readOnly": true
    },
    "username": {
      "title": "Username",
      "description": "Required. 150 characters or fewer. Letters, ↵
↵digits and @/./+/-/_ only.",
      "type": "string",
      "pattern": "^[\\w. @+-]+$",
      "maxLength": 150,
      "minLength": 1
    },
    "is_active": {
      "title": "Is active",
      "type": "boolean",
      "default": true
    }
  },
}
```

(continues on next page)

(continued from previous page)

```
        "email": {
            "title": "Email",
            "type": "string",
            "format": "email",
            "minLength": 1
        }
    }
}
```

CreateUser Schema

```
{
  "id": {
    "title": "ID",
    "type": "integer",
    "readOnly": true
  },
  "username": {
    "title": "Username",
    "description": "Required. 150 characters or fewer. Letters, digits,
↪and @/./+/-/_ only.",
    "type": "string",
    "pattern": "^[\\w.\\@+-]+$",
    "maxLength": 150,
    "minLength": 1
  },
  "is_active": {
    "title": "Is active",
    "type": "boolean",
    "default": true
  },
  "is_staff": {
    "title": "Is staff",
    "type": "boolean",
    "default": false
  },
  "first_name": {
    "title": "First name",
    "type": "string",
    "maxLength": 30
  },
  "last_name": {
    "title": "Last name",
    "type": "string",
    "maxLength": 150
  },
  "email": {
    "title": "Email",
    "type": "string",
    "format": "email",
    "minLength": 1
  },
  "password": {
    "title": "Password",
```

(continues on next page)

(continued from previous page)

```

        "type": "string",
        "minLength": 1
    },
    "password2": {
        "title": "Repeat password",
        "type": "string",
        "minLength": 1
    }
}

```

OneUser Schema

```

{
  "id": {
    "title": "ID",
    "type": "integer",
    "readOnly": true
  },
  "username": {
    "title": "Username",
    "description": "Required. 150 characters or fewer. Letters, digits,
↪and @/./+/-/_ only.",
    "type": "string",
    "pattern": "^[\\w. @+-]+$",
    "maxLength": 150,
    "minLength": 1
  },
  "is_active": {
    "title": "Is active",
    "type": "boolean",
    "default": true
  },
  "first_name": {
    "title": "First name",
    "type": "string",
    "maxLength": 30
  },
  "last_name": {
    "title": "Last name",
    "type": "string",
    "maxLength": 150
  },
  "email": {
    "title": "Email",
    "type": "string",
    "format": "email",
    "minLength": 1
  }
}

```

ChangePassword Schema

```
{
  "old_password": {
    "title": "Old password",
    "type": "string",
    "minLength": 1
  },
  "password": {
    "title": "New password",
    "type": "string",
    "minLength": 1
  },
  "password2": {
    "title": "Confirm new password",
    "type": "string",
    "minLength": 1
  }
}
```

ChartLineSetting Schema

```
{
  "active": {
    "title": "Active",
    "type": "boolean",
    "default": true
  }
}
```

ChartLineSettings Schema

```
{
  "all_tasks": {
    "active": {
      "title": "Active",
      "type": "boolean",
      "default": true
    }
  },
  "delay": {
    "active": {
      "title": "Active",
      "type": "boolean",
      "default": true
    }
  },
  "ok": {
    "active": {
      "title": "Active",
      "type": "boolean",
      "default": true
    }
  }
}
```

(continues on next page)

(continued from previous page)

```
    },
    "error": {
      "active": {
        "title": "Active",
        "type": "boolean",
        "default": true
      }
    },
    "interrupted": {
      "active": {
        "title": "Active",
        "type": "boolean",
        "default": true
      }
    },
    "offline": {
      "active": {
        "title": "Active",
        "type": "boolean",
        "default": true
      }
    }
  }
}
```

CounterWidgetSetting Schema

```
{
  "active": {
    "title": "Active",
    "type": "boolean",
    "default": true
  },
  "collapse": {
    "title": "Collapse",
    "type": "boolean",
    "readOnly": true,
    "default": false
  },
  "sort": {
    "title": "Sort",
    "type": "integer",
    "default": 0
  }
}
```

WidgetSetting Schema

```
{
  "active": {
    "title": "Active",
    "type": "boolean",
    "default": true
  },
  "collapse": {
    "title": "Collapse",
    "type": "boolean",
    "default": false
  },
  "sort": {
    "title": "Sort",
    "type": "integer",
    "default": 0
  }
}
```

WidgetSettings Schema

```
{
  "pmwUsersCounter": {
    "active": {
      "title": "Active",
      "type": "boolean",
      "default": true
    },
    "collapse": {
      "title": "Collapse",
      "type": "boolean",
      "readOnly": true,
      "default": false
    },
    "sort": {
      "title": "Sort",
      "type": "integer",
      "default": 0
    }
  },
  "pmwProjectsCounter": {
    "active": {
      "title": "Active",
      "type": "boolean",
      "default": true
    },
    "collapse": {
      "title": "Collapse",
      "type": "boolean",
      "readOnly": true,
      "default": false
    },
    "sort": {
      "title": "Sort",
```

(continues on next page)

(continued from previous page)

```

        "type": "integer",
        "default": 0
    },
    },
    "pmwTemplatesCounter": {
        "active": {
            "title": "Active",
            "type": "boolean",
            "default": true
        },
        "collapse": {
            "title": "Collapse",
            "type": "boolean",
            "readOnly": true,
            "default": false
        },
        "sort": {
            "title": "Sort",
            "type": "integer",
            "default": 0
        }
    },
    "pmwInventoriesCounter": {
        "active": {
            "title": "Active",
            "type": "boolean",
            "default": true
        },
        "collapse": {
            "title": "Collapse",
            "type": "boolean",
            "readOnly": true,
            "default": false
        },
        "sort": {
            "title": "Sort",
            "type": "integer",
            "default": 0
        }
    },
    "pmwGroupsCounter": {
        "active": {
            "title": "Active",
            "type": "boolean",
            "default": true
        },
        "collapse": {
            "title": "Collapse",
            "type": "boolean",
            "readOnly": true,
            "default": false
        },
        "sort": {
            "title": "Sort",
            "type": "integer",
            "default": 0
        }
    }
}

```

(continues on next page)

(continued from previous page)

```
    },
    "pmwHostsCounter": {
      "active": {
        "title": "Active",
        "type": "boolean",
        "default": true
      },
      "collapse": {
        "title": "Collapse",
        "type": "boolean",
        "readOnly": true,
        "default": false
      },
      "sort": {
        "title": "Sort",
        "type": "integer",
        "default": 0
      }
    },
    "pmwChartWidget": {
      "active": {
        "title": "Active",
        "type": "boolean",
        "default": true
      },
      "collapse": {
        "title": "Collapse",
        "type": "boolean",
        "default": false
      },
      "sort": {
        "title": "Sort",
        "type": "integer",
        "default": 0
      }
    }
  }
}
```

UserSettings Schema

```
{
  "lang": {
    "title": "Lang",
    "type": "string",
    "enum": [
      "en",
      "ru"
    ],
    "default": "en"
  },
  "autoupdateInterval": {
    "title": "Autoupdateinterval",
    "type": "integer",
    "default": 15000
  }
}
```

(continues on next page)

(continued from previous page)

```

    },
    "chartLineSettings": {
      "all_tasks": {
        "active": {
          "title": "Active",
          "type": "boolean",
          "default": true
        }
      },
      "delay": {
        "active": {
          "title": "Active",
          "type": "boolean",
          "default": true
        }
      },
      "ok": {
        "active": {
          "title": "Active",
          "type": "boolean",
          "default": true
        }
      },
      "error": {
        "active": {
          "title": "Active",
          "type": "boolean",
          "default": true
        }
      },
      "interrupted": {
        "active": {
          "title": "Active",
          "type": "boolean",
          "default": true
        }
      },
      "offline": {
        "active": {
          "title": "Active",
          "type": "boolean",
          "default": true
        }
      }
    },
    "widgetSettings": {
      "pmwUsersCounter": {
        "active": {
          "title": "Active",
          "type": "boolean",
          "default": true
        },
        "collapse": {
          "title": "Collapse",
          "type": "boolean",
          "readOnly": true,
          "default": false
        }
      }
    }
  }

```

(continues on next page)

(continued from previous page)

```
    },
    "sort": {
      "title": "Sort",
      "type": "integer",
      "default": 0
    }
  },
  "pmwProjectsCounter": {
    "active": {
      "title": "Active",
      "type": "boolean",
      "default": true
    },
    "collapse": {
      "title": "Collapse",
      "type": "boolean",
      "readOnly": true,
      "default": false
    },
    "sort": {
      "title": "Sort",
      "type": "integer",
      "default": 0
    }
  },
  "pmwTemplatesCounter": {
    "active": {
      "title": "Active",
      "type": "boolean",
      "default": true
    },
    "collapse": {
      "title": "Collapse",
      "type": "boolean",
      "readOnly": true,
      "default": false
    },
    "sort": {
      "title": "Sort",
      "type": "integer",
      "default": 0
    }
  },
  "pmwInventoriesCounter": {
    "active": {
      "title": "Active",
      "type": "boolean",
      "default": true
    },
    "collapse": {
      "title": "Collapse",
      "type": "boolean",
      "readOnly": true,
      "default": false
    },
    "sort": {
      "title": "Sort",
```

(continues on next page)

(continued from previous page)

```

        "type": "integer",
        "default": 0
    },
    },
    "pmwGroupsCounter": {
        "active": {
            "title": "Active",
            "type": "boolean",
            "default": true
        },
        "collapse": {
            "title": "Collapse",
            "type": "boolean",
            "readOnly": true,
            "default": false
        },
        "sort": {
            "title": "Sort",
            "type": "integer",
            "default": 0
        }
    },
    "pmwHostsCounter": {
        "active": {
            "title": "Active",
            "type": "boolean",
            "default": true
        },
        "collapse": {
            "title": "Collapse",
            "type": "boolean",
            "readOnly": true,
            "default": false
        },
        "sort": {
            "title": "Sort",
            "type": "integer",
            "default": 0
        }
    },
    "pmwChartWidget": {
        "active": {
            "title": "Active",
            "type": "boolean",
            "default": true
        },
        "collapse": {
            "title": "Collapse",
            "type": "boolean",
            "default": false
        },
        "sort": {
            "title": "Sort",
            "type": "integer",
            "default": 0
        }
    }
}

```

(continues on next page)

(continued from previous page)

```
    },  
    "selectedSkin": {  
        "title": "Selectedskin",  
        "type": "string",  
        "minLength": 1  
    },  
    "skinsSettings": {}  
}
```

VST UTILS FRONTEND

The main feature of the VST Utils framework is autogenerated GUI, which is forming based on the [OpenAPI schema¹⁵](#). Autogenerated GUI is available because of some frontend mechanisms work.

VST Utils frontend consists of 2 blocks:

1. Block, that is responsible for *parsing of OpenAPI schema* and generating entities necessary for application work:
 - *Objects of Model class* - store model's name, fields and so on;
 - *Objects of QuerySet class* - operate with data of Model instances;
 - *Objects of View class* - have info about representation of Model instance.
2. Block, that is responsible for rendering and routing. This block works based on:
 - [Vue¹⁶](#) - views rendering;
 - [Vue Router¹⁷](#) - routing between application views;
 - [Vuex¹⁸](#) - application state store.

Let's look on the structure of those blocks' elements.

4.1 Frontend src entry points

`vstutils` frontend has several webpack entry points:

- `spa.js` - main application
- `doc.js` - bundle for documentation
- `auth.js` - libraries for login and register pages

All entry points contains at least:

- `jquery`
- `bootstrap`
- `fontawesome`
- `admin-lte`

¹⁵ <https://swagger.io/docs/specification/about/>

¹⁶ <https://vuejs.org/>

¹⁷ <https://router.vuejs.org/>

¹⁸ <https://vuex.vuejs.org/>

4.2 Association with back-end entities

VST Utils's back-end is realised on the base of Django Framework, which operates by some entities in it's work:

- Django Model¹⁹;
- Django Queryset²⁰;
- Django View²¹.

So, we decided, that it will be great, if we will use similar entities in the frontend, because it will be easier for backend developer understand how frontend works and vice versa.

Model class and *View class* are used for generating of frontend Models and View on the final step of *parsing of OpenAPI schema*.

4.3 Model class

Model class is an abstraction, that is supposed to be similar to the Django Model abstraction. This class has all necessary data about Model structure: name of the Model, Model's fields, which field is a primary key, which field is should be used as view field (in some autocomplete lists, for example). Also this class has some methods, for manipulations with Model instances.

Objects of this class are Models, so with the help of this class VST Utils creates all necessary Models based on data from OpenAPI schema.

To get Model Instance you should use `getInstance()` method of Model object.

Let's look on properties of this class.

4.3.1 Structure of Model

Let's look on structure of Model object on the example of User Model:

```
{
  name: "User",
  fiends: {
    id: IntegerField,
    username: StringField,
    email: StringField,
    is_active: BooleanField,
  },
  non_instance_attr: ["non_instance_attr", "constructor", "getInstance"],
  pk_name: "id",
  view_name: "username",
}
```

As you can see from the example Model has several properties:

- **name** - string, that stores name of Model;
- **fields** - object, that stores Model fields - instances of guiFields;

¹⁹ <https://docs.djangoproject.com/en/2.2/topics/db/models/>

²⁰ <https://docs.djangoproject.com/en/2.2/ref/models/querysets/>

²¹ <https://docs.djangoproject.com/en/2.2/topics/class-based-views/>

- **non_instance_attr** - array, that stores names of properties and methods, that should not be added to the Model Instance (during Model instance creation in the `getInstance()` method);
- **pk_name** - string, that stores name of Model field, that is Primary Key for this model.
- **view_name** - string, that stores name of Model field, that is supposed to be used for Model Instance representation in some autocomplete lists and so on.

Let's look on properties of Model Instance.

4.3.2 Structure of Model Instance

Let's look on structure of Model Instance on the example of User Model Instance:

```
{
  name: "User",
  fiends: {
    id: IntegerField,
    username: StringField,
    email: StringField,
    is_active: BooleanField,
  },
  non_instance_attr: ["non_instance_attr", "constructor", "getInstance"],
  pk_name: "id",
  view_name: "username",
  data: {
    id: 1,
    username: "admin",
    email: "admin@mail.com"
    is_active: true,
  },
  queryset: QuerySet,
}
```

As you can see from the example Model Instance has the same properties as Model plus 2 new properties:

- **name** - string, that stores name of Model;
- **fields** - object, that stores Model fields - instances of `guiFields`;
- **non_instance_attr** - array, that stores names of properties and methods, that should not be added to the Model Instance (during Model instance creation in the `getInstance()` method);
- **pk_name** - string, that stores name of Model field, that is Primary Key for this model.
- **view_name** - string, that stores name of Model field, that is supposed to be used for Model Instance representation in some autocomplete lists and so on.
- **data** - object with Model Instance data (values of Model Instance fields);
- **queryset** - instance of `QuerySet` class - `queryset`, connected with the current Model Instance.

Let look on the methods of Model and Model Instance.

4.3.3 constructor(name, fields)

Arguments:

- name: {string} - Model's name.
- fields: {object} - Object with Model fields. Those fields are supposed to be instances of guiFields.

Description: Standard constructor of JS class. This method creates new Model with current name and fields. Also this method sets which field is PK and which field is view field.

4.3.4 getPrefetchFields()

Description: Method loops through Model fields, tries to find fields of FK type (and fields, inherited from FK field) and returns matched fields. This method is supposed to be called, when it is necessary to load prefetched data of Model Instance (name of instance, to which FK field is linked, for example).

4.3.5 getInstance(data, queryset)

Arguments:

- data: {object} - Object with Model Instance data.
- queryset: {object} - Instance of QuerySet class. This QuerySet is aimed to load and set all necessary data for Model instance in API.

Description: This method returns Model Instance.

4.3.6 toInner(form_data=this.data)

Arguments:

- form_data: {object} - Object with Model Instance data, that was modified by user. Default value is `ModelInstance.data` property.

Description: This method converts data from 'GUI view' format into format, appropriate for API.

4.3.7 toRepresent(api_data=this.data)

Arguments:

- api_data: {object} - Object with Model Instance data, that was get from API. Default value is `ModelInstance.data` property.

Description: This method converts data from API format into format, appropriate for 'GUI view'.

4.3.8 getPkValue()

Description: Method, that returns Model Instance's value of PK field.

4.3.9 getViewFieldValue()

Description: Method, that returns Model Instance's value of view field.

4.3.10 save(method="patch")

Arguments:

- **method:** {string} - Name of method (POST, PATCH, PUT), that should be used during saving.

Description: Method, that saves Model Instance's changes.

4.3.11 delete()

Description: Method, that deletes Model Instance.

4.4 QuerySet class

QuerySet class is an abstraction, that is supposed to be similar to the Django QuerySet abstraction. This class has methods for filtering, getting, creating Model Instances. Those methods form appropriate API requests and send them.

Let's look on properties of this class.

4.4.1 Structure of QuerySet Instance

Let's look on structure of QuerySet Instance on the example of QuerySet for User Model Instance, available by `/user/1` path:

```
{
  model: Model,
  query: {},
  url: "/user/1",
  use_prefetch: true,
  cache: ModelInstance,
}
```

As you can see from the example QuerySet Instance has following properties:

- **model** - User Model;
- **query** - object with filters values - (key, value) pairs.
- **url** - string, that stores URL by which connected Model Instance and it's data are available;
- **use_prefetch** - boolean/array - if boolean - means - use prefetch or not, otherwise, means array with names of model fields, that should be used as prefetch fields.
- **cache** - object/array - if current QuerySet Instance is connected with `list` view, this property will be array, storing list of Model Instances. Otherwise, it will store only one Model Instance and type of this property will be 'object'.

Let's look on methods of this class.

4.4.2 constructor(model, url, query={})

Arguments:

- model: {object} - Model for which this QuerySet will be created.
- url: {string} - Current url of view. For example, if current URL is `/user/1`, QuerySet for Model Instance, that should be represented on the view with current url, will have property url equal to `/user/1`.
- query: {object} - Object, that stores current QuerySet filters (pairs of key, value). Default value: empty object.

Description: Standard constructor of JS class. This method creates new QuerySet with current arguments. Also it sets property `use_prefetch` equal to `false`. This property means load prefetch data or not.

4.4.3 makeQueryString(query=this.query)

Arguments:

- query: {object} - Object with pairs of key, value for QuerySet filters. Default value: `this.query`.

Description: Method, that converts 'query' object into 'filters' string, appropriate for bulk query.

4.4.4 getDataType()

Description: Method, that converts 'this.url' string into 'data_type' array, appropriate for bulk query.

4.4.5 formBulkQuery(method, data)

Arguments:

- method: {string} - Method(get/delete/post/put/patch) of bulk query.
- data: {object} - 'data' property for body of bulk query, data of Model Instance.

Description: Method, that forms body of bulk query.

4.4.6 formQueryAndSend(method, data)

Arguments:

- method: {string} - Method(get/delete/post/put/patch) of bulk query.
- data: {object} - 'data' property for body of bulk query, data of Model Instance.

Description: Method, that forms bulk query and sends it to API.

4.4.7 sendQuery(bulk)

Arguments:

- bulk: {object} - bulk Object with properties of bulk data.

Description: Method, that sends bulk query to API.

4.4.8 clone(props={}, save_cache=false)

Arguments:

- props: {object} - Object with properties, that should be rewritten in clone. Default value: empty object.
- save_cache: {boolean} If true, cache of current QuerySet will be saved in clone. Default value: false.

Description: Method, that returns clone (new QuerySet instance) of current QuerySet, without cache, by default.

4.4.9 copy(props={})

Arguments:

- props: {object} - Object with properties, that should be rewritten in clone. Default value: empty object.

Description: Method, that returns copy (new QuerySet instance) of current QuerySet, with cache of current QuerySet.

4.4.10 all()

Description: Method, that returns clone of current QuerySet Instance with current value of 'this.query' property.

4.4.11 filter(filters)

Arguments:

- filters: {object} - Object with filters(key, value), according to which Model Instances list should be sorted.

Description: Method, that returns clone of current QuerySet Instance with new filters, that will be saved in 'query' property.

4.4.12 exclude(filters)

Arguments:

- filters: {object} - Object with filters(key, value), according to which some instances should be excluded from Model instances list.

Description: Method, that returns clone of current QuerySet Instance with new filters, that will be saved in 'query' property.

4.4.13 prefetch(instances=true)

Arguments:

- instances: {boolean | array} If boolean - means - Use prefetch or not, otherwise, means array with names of model fields, that should be used as prefetch field.

Description: Method, that returns clone of current QuerySet Instance with new value of 'use_prefetch' property.

4.4.14 items()

Description: Method, that sends to API get request for getting list of Model Instances, appropriate for filters from 'this.query' property. Method, returns promise, that it will return list of Model instances, if API request be successful.

4.4.15 create(data)

Arguments:

- data: {object} Data of new Model Instance.

Description: Method, that sends query to API for creation of new Model Instance and returns promise, that it will return Model Instance, if query response be successful.

4.4.16 delete()

Description: Method, that deletes all Model Instances, that this.items() returns. It means, that this method deletes all instances, that were filtered before it's execution. This method is expected to be called after instance filtering. This method is only for querysets, that have 'url' of 'list' type. This method should not be applied for querysets with 'page' type url.

4.4.17 get()

Description: Method, that returns promise, that it will return Model Instance with 'this.url' URI, if API query be successful.

4.4.18 clearCache()

Description: Method, that cleans QuerySet cache.

4.4.19 _getPrefetchFields()

Description: Method, that returns array with names of prefetch fields.

4.4.20 `_getBulkDataForPrefetch(prefetch_fields, instances)`

Arguments:

- `prefetch_fields` {array} - Array with names of prefetch fields.
- `instances` {object} - Object with loaded Model Instances.

Description: Method, that forms `bulk_data` for prefetch Bulk.

4.4.21 `_getBulkDataForPrefetchForInstance(prefetch_fields, instance, bulk_data)`

Arguments:

- `prefetch_fields`: {array} - Array with names of prefetch fields.
- `instance`: {object} - Model instance.
- `bulk_data`: {object} - Object with `bulk_data`.

Description: Method, that forms prefetch `bulk_data` for one instance.

4.4.22 `_loadPrefetchData(prefetch_fields, instances)`

Arguments:

- `prefetch_fields`: {array} - Array with names of prefetch fields.
- `instances`: {object} - Object with loaded model instances.

Description: Method, that loads prefetch info for instances, which were loaded by current queryset.

4.4.23 `_setPrefetchValue(res, bulk_data, instances, field_name)`

Arguments:

- `res`: {object} - Prefetch API response.
- `bulk_data_item`: {object} - Object bulk data for prefetch request.
- `instances`: {array} - Array with instances.
- `field_name`: {string} - Name of model field.

Description: Method, that adds loaded prefetch data to instances.

4.5 View class

View class is an abstraction, that stores info about representation of some Model Instance available by some path (URL). This class is used for creation of views objects, that stores info about template, that should be used for current view, about views, to which current view can be linked and so on.

There are 5 types of views in VST Utils:

- **list** - view, that is responsible for representation of Model Instances list;
- **page_new** - view, that is responsible for representation of the page for new Model Instance creation;
- **page** - view, that is responsible for representation of Model Instance;

- **page_edit** - view, that is responsible for representation of the page for Model Instance editing;
- **action** - view, that is responsible for representation of the page for some Model Instance action execution.

Those views have common structure, but values of several properties are different for each type of view.

Let's look on the structure of the `list` view on the example of `User list` view.

```
{
  objects: QuerySet,
  schema: {
    actions: {},
    autoupdate: true,
    child_links: {
      change_password: {
        name: "change_password",
        path: "/user/{id}/change_password/",
      },
      copy: {
        name: "copy",
        path: "/user/{id}/copy/",
      },
      edit: {
        name: "edit",
        path: "/user/{id}/edit/",
      },
      remove: {
        name: "remove",
      },
    },
    filters: {
      id: IntegerField,
      id__not: IntegerField,
      is_active: StringField,
      ordering: StringField,
      username: StringField,
      username__not: StringField,
    },
    level: 2,
    multi_actions: {
      remove: {
        name: "remove",
        title: "Remove",
        multi_action: true,
      },
    },
    name: "user",
    operation_id: "user_list",
    operations: {
      new: {
        name: "new",
        path: "/user/new/",
      },
    },
    page_path: "/user/{id}/",
    path: "/user/",
    query_type: "get",
    sublinks: {},
    type: "list",
  },
}
```

(continues on next page)

(continued from previous page)

```

    },
    template: "#template_view_entity",
    mixins: [],
  }

```

As you can see from the example View Instance has following properties:

- **objects** - QuerySet instance, connected with current view;
- **schema** - object, that stores options of current view and links to connected views;
- **template** - string, that contents either template string of current view or 'id' of element, that stores template string;
- **mixins** - array, that stores additional mixins for Vue component of current view. These mixins will be used during *creation of route* for current view. These mixins can redefine some base functionality of view Vue component or add some additional ones. *RouterConstructor class* will combine these mixins with base ones (that will be defined based on the view type).

Let's look closely on the properties of the `schema` property:

- **actions** - object, that stores pairs of key, value, where key - name of action, that can be executed/opened from this view, value - object with options of current action (name, path and so on). Action is an activity, that can be executed for some Model Instance (for example, for User Model Instance it could be 'change password', 'copy');
- **autoupdate** - boolean, that means make automatic requests for data updating from current view or not;
- **child_links** - object, that stores pairs of key, value, where key - name of child link, that can be executed/opened from this view, value - object with options of current child link (name, path and so on). Child link is an action / operation / sublink of a page view, that could be executed/opened from the `list` view;
- **filters** - object, that stores pairs of key, value, where key - name of filter, which can be used on current view, value - instance of `guiField`, that is used for setting filter value. Filter is some option of Model Instance with the help of which you can categorize you search on Model Instance list;
- **level** - number, that tells about nesting depth of current view path;
- **multi_actions** - object, that stores pairs of key, value, where key - name of multi action, that can be executed/opened from this view, value - object with options of current multi action (name, path and so on). Multi action is an action / operation, that can be executed from `list` view for several Model Instance at the same time;
- **name** - string, name of Model or Action, connected with current view;
- **operation_id** - string, that stores name of Model and type of view. This property is used in OpenAPI schema;
- **operations** - object, that stores pairs of key, value, where key - name of operation, that can be executed/opened from this view, value - object with options of current operation (name, path and so on). Operation is some basic activity, that be done with the Model Instance (create, edit, save, delete);
- **page_path** - string, path of page view connected with current `list` view;
- **path** - string, path of current view (template for URL);
- **query_type** - string, name of HTTP method, that should be used for API requests from current view;
- **sublinks** - object, that stores pairs of key, value, where key - name of sublink, that can be opened from this view, value - object with options of current sublink (name, path and so on). Sublink is an `list` view, that is nested in current. For example, in path `/foo/{id}/bar/` bar will be sublink for `foo`;
- **type** - string, type of view (`list` / `page_new` / `page` / `page_edit` / `action`);

Let's look on methods of this class.

4.5.1 constructor(model, schema, template)

Arguments:

- model: {object} - Model, with which this view is connected.
- schema: {object} - Options of current view, that include settings for a view (internal links, view type and so on).
- template: {string} - Template content or id of script with template content.

Description: Standard constructor of JS class. This method creates new View with current arguments.

4.5.2 getViewSublinkButtons(type, buttons, instance)

Arguments:

- type: {string} - Buttons type - actions / operations / sublinks / child_links.
- buttons: {object} - Object with buttons options.
- instance: {object} - Model Instance connected with current view.

Description: Method, that handles view buttons (actions, operations, sublinks, child_links) and returns them.

4.5.3 getPathTemplateForRouter(path="")

Arguments:

- path: {string} - View path.

Description: Method returns string with template of route path for current view.

4.5.4 static getQuerySetConstructor(model)

Arguments:

- model: {object} - Model object.

Description: Method, that returns QuerySet constructor for view.

4.6 Fields classes

Very often during creation of some new app developers need to make common fields of some base types and formats (string, boolean, number and so on). Create everytime similar functionality is rather boring and ineffective, so we tried to solve this problem with the help of VST Utils.

VST Utils has set of built-in fields of the most common types and formats, that can be used for different cases. For example, when you need to add some field to you web form, that should hide value of inserted password, all you need to do - just set appropriate format `password` instead of `string` and VST Utils make all work for you.

Field classes are used in Model Instances as fields and also are used in Views Instances of `list` type as filters.

All available fields classes are stored in the `guiFields` variable. There are 44 fields formats in VST Utils:

- **base** - base field, from which the most other fields are inherited;

- **string** - string field, for inserting and representation of some short 'string' values;
- **textarea** - string field, for inserting and representation of some long 'string' values;
- **number** - number field, for inserting and representation of 'number' values;
- **integer** - number field, for inserting and representation of values of 'integer' format;
- **int32** - number field, for inserting and representation of values of 'int32' format;
- **int64** - number field, for inserting and representation of values of 'int64' format;
- **double** - number field, for inserting and representation of values of 'double' format;
- **float** - number field, for inserting and representation of values of 'float' format;;
- **boolean** - boolean field, for inserting and representation of 'boolean' values;
- **choices** - string field, with strict set of preset values, user can only choose one of the available value variants;
- **autocomplete** - string field, with set of preset values, user can either choose one of the available value variants or insert his own value;
- **password** - string field, that hides inserted value by '*' symbols;
- **file** - string field, that can read content of the file;
- **secretfile** - string field, that can read content of the file and then hide it from representation;
- **binfile** - string field, that can read content of the file and convert it to the 'base64' format;
- **namedbinfile** - field of JSON format, that takes and returns JSON with 2 properties: name (string) - name of file and content(base64 string) - content of file;
- **namedbinimage** - field of JSON format, that takes and returns JSON with 2 properties: name (string) - name of image and content(base64 string) - content of image;
- **multiplenamedbinfile** - field of JSON format, that takes and returns array with objects, consisting of 2 properties: name (string) - name of file and content(base64 string) - content of file;
- **multiplenamedbinimage** - field of JSON format, that takes and returns array with objects, consisting of 2 properties: name (string) - name of image and content(base64 string) - content of image;
- **text_paragraph** - string field, that is represented as text paragraph (without any inputs);
- **plain_text** - string field, that saves all non-printing characters during representation;
- **html** - string field, that contents different html tags and that renders them during representation;
- **date** - date field, for inserting and representation of 'date' values in 'YYYY-MM-DD' format;
- **date_time** - date field, for inserting and representation of 'date' values in 'YYYY-MM-DD HH:mm' format;
- **uptime** - string field, that converts time duration (amount of seconds) into one of the most appropriate variants (23:59:59 / 01d 00:00:00 / 01m 30d 00:00:00 / 99y 11m 30d 22:23:24) due to the it's value size;
- **time_interval** - number field, that converts time from milliseconds into seconds;
- **crontab** - string field, that has additional form for creation schedule in 'crontab' format;
- **json** - field of JSON format, during representation it uses another guiFields for representation of current field properties;
- **api_object** - field, that is used for representation of some Model Instance from API (value of this field is the whole Model Instance data). This is read only field;
- **fk** - field, that is used for representation of some Model Instance from API (value of this field is the Model Instance Primary Key). During edit mode this field has strict set of preset values to choose;

- **fk_autocomplete** - field, that is used for representation of some Model Instance from API (value of this field is the Model Instance Primary Key or some string). During edit mode user can either choose of the preset values from autocomplete list or insert his own value;
- **fk_multi_autocomplete** - field, that is used for representation of some Model Instance from API (value of this field is the Model Instance Primary Key or some string). During edit mode user can either choose of the preset values from modal window or insert his own value;
- **color** - string field, that stores HEX code of selected color;
- **inner_api_object** - field, that is linked to the fields of another model;
- **api_data** - field for representing some data from API;
- **dynamic** - field, that can change its format depending on the values of surrounding fields;
- **hidden** - field, that is hidden from representation;
- **form** - field, that combines several other fields and stores those values as one JSON, where key - name of form field, value - value of form field;
- **button** - special field for form field, imitates button in form;
- **string_array** - field, that converts array with strings into one string;
- **string_id** - string field, that is supposed to be used in URLs as 'id' key. It has additional validation, that checks, that field's value is not equal to some other URL keys (new/ edit/ remove).

4.6.1 guiFields.base structure

All guiFields have common structure, so let's look on fields properties on the example of guiFields.base:

- **mixins** - array with mixin objects, that will be used during rendering of field component by Vue.js;
- **options** - object with field's options (settings), like name, title, required, readOnly, type, format, description and so on.

Let's look on the guiFields.base methods.

4.6.2 constructor(options={})

Arguments:

- options: {object} - Options of field instance.

Description: Standard constructor of JS class. This method creates new guiFields.base instance with current arguments.

4.6.3 toInner(data={})

Arguments:

- data: {object} - Object with values of current field and fields from the same fields wrapper. For example, from the same Model Instance.

Description: Method, that converts field value from representation format into format appropriate for API.

4.6.4 toRepresent(data={})

Arguments:

- data: {object} - Object with values of current field and fields from the same fields wrapper. For example, from the same Model Instance.

Description: Method, that converts field value from format appropriate for API into representation format.

4.6.5 validateValue(data={})

Arguments:

- data: {object} - Object with values of current field and fields from the same fields wrapper. For example, from the same Model Instance.

Description: Method, that validates values. Method checks that value satisfies field's options.

4.6.6 static get mixins()

Description: Static property for storing field mixins. Content of those mixins for each field is storing in `gui_fields_mixins` variable.

Fields of other formats have the same structure and the same methods, but realisation of this methods can vary. Also some fields have some additional properties in options and some additional methods.

REDEFINITION OF VST UTILS FRONTEND

VST Utils is rather flexible framework, so if you need to redefine some base frontend functionality in your project you can do it in 2 ways:

- *Signals* - appropriate for situations, when you need to add some additional functionality to the base one or when you need to redefine some parts of entity's structure. For example:
 - add some additional field to *Model object*;
 - change *Model object's* field format;
 - change template of some *View object*;
 - change something in *View object's* schema;
 - extend or redefine behavior of the *View object's* Vue component via mixins;
 - and so on.
- *Class inheritance* - appropriate for situations, when you need to redefine some part of functionality (methods, properties) of such base entities as *Model class*, *QuerySet class*, *Fields classes*.

Let's look closely on both ways of functionality redefinition.

5.1 Signals

System of signals is a mechanism, that VST Utils uses for app customization.

Let's look how it works.

Very often you need to modify something after some event has occurred. But how can you know about this event? And what if you need to know about this event in several blocks of code?

To solve this problem VST Utils uses system of signals, where:

- you can emit some signal, which tells all subscribers, that some event has occurred, and pass some data/variables from the context, where this event has occurred;
- you can subscribe to some signal, that notifies you about some event, and also you can pass some callback (handler) that can do something with data/variables, that were passed from the context, where event had occurred.

5.1.1 Emit signal

To emit some signal you need to write following in you code:

```
tabSignal.emit(name_of_signal, context);
```

where:

- **name_of_signal** - string, which stores name of signal (event);
- **context** - some variable of any type, that will be passed to the callback (handler) during connection to this signal.

Example of signal emitting:

```
let app = {  
  name: 'example of app';  
};  
  
tabSignal.emit('app.created', app);
```

5.1.2 Connect to signal

To connect to some signal you need to write following in you code:

```
tabSignal.connect(name_of_signal, callback);
```

where:

- **name_of_signal** - string, which stores name of signal (event);
- **callback** - function, that can do something with variables, which will be passed from event's context to this callback as arguments.

Example of connecting to signal:

```
/* ... */  
function callback(app) {  
  app.title = 'example of app title';  
}  
  
tabSignal.connect('app.created', callback);  
/* ... */
```

5.2 Signals available in VST Utils

VST Utils has some signals, that are emitting during application work. If you need to customize/redefine something in you project you can subscribe to these signals and do some code with those context. Also you can emit you own signals in your project, if you need.

Let's look what signals are used in VST Utils.

5.2.1 openapi.loaded

Signal name: “openapi.loaded”.

Context argument: openapi - {object} - OpenAPI schema loaded from API.

Description: This signal will be emitted after OpenAPI schema was loaded. You can use this signal if you need to redefine something in the OpenAPI schema, before it was parsed.

5.2.2 resource.loaded

Signal name: “resource.loaded”.

Context argument: None.

Description: This signal will be emitted after all static files were successfully loaded and added to the page.

5.2.3 app.version.updated

Signal name: “app.version.updated”.

Context argument: None.

Description: This signal will be emitted during app loading if VST Utils detects, that version of your project was updated.

5.2.4 app.beforeInitStore

Signal name: “app.beforeInitStore”.

Context argument: obj - {object} - Object with following structure: {storeConstructor: StoreConstructor}, where storeConstructor is an instance of StoreConstructor.

Description: This signal will be emitted after creation of StoreConstructor instance and before app creation

5.2.5 app.beforeInitRouter

Signal name: “app.beforeInitRouter”.

Context argument: obj - {object} - Object with following structure: {routerConstructor: RouterConstructor}, where routerConstructor is an instance of RouterConstructor.

Description: This signal will be emitted after creation of RouterConstructor instance and before app creation

5.2.6 app.beforeInit

Signal name: “app.beforeInit”.

Context argument: obj - {object} - Object with following structure: {app: app}, where app is an instance of App class.

Description: This signal will be emitted after app variable initialization (OpenAPI schema was parsed, models and views were created), but before app was mounted to the page.

5.2.7 app.afterInit

Signal name: “app.afterInit”.

Context argument: obj - {object} - Object with following structure: {app: app}, where app is an instance of App class.

Description: This signal will be emitted after app was mounted to the page.

5.2.8 app.language.changed

Signal name: “app.language.changed”.

Context argument: obj - {object} - Object with following structure: {lang: lang}, where lang is an code of applied language.

Description: This signal will be emitted after app interface language was changed.

5.2.9 models[model_name].fields.beforeInit

Signal name: “models[” + model_name + “].fields.beforeInit”. For example, for User model: “models[User].fields.beforeInit”.

Context argument: fields - {object} - Object with pairs of key, value, where key - name of field, value - object with it options. On this moment, field - is just object with options, it is not guiFields instance.

Description: This signal will be emitted before creation of guiFields instances for Model fields.

5.2.10 models[model_name].fields.afterInit

Signal name: “models[” + model_name + “].fields.afterInit”. For example, for User model: “models[User].fields.afterInit”.

Context argument: fields - {object} - Object with pairs of key, value, where key - name of field, value - guiFields instance.

Description: This signal will be emitted after creation of guiFields instances for Model fields.

5.2.11 models[model_name].created

Signal name: “models[” + model_name + “].created”. For example, for User model: “models[User].created”.

Context argument: obj - {object} - Object with following structure: {model: model}, where model is the created Model.

Description: This signal will be emitted after creation of Model object.

5.2.12 allModels.created

Signal name: “allModels.created”.

Context argument: obj - {object} - Object with following structure: {models: models}, where models is the object, storing Models objects.

Description: This signal will be emitted after all models were created.

5.2.13 views.schema[path].beforeInit

Signal name: “views.schema[” + path + “.beforeInit”.

Context argument: obj - {object} - Object with following structure: {schema, editStyle}: **schema** - object with opanpi schema that will be used for generating operations; **editStyle** - boolean that tells is edit only view should be generated, by default editStyle equals info[‘x-edit-style’] of openapi schema.

Description: This signal will be emitted before generations of operations from opanpi view. Allows you to change default behavior of editOnly views.

5.2.14 views[path].filters.beforeInit

Signal name: “views[” + path + “.filters.beforeInit”. For example, for /user/ view: “views[/user/].filters.beforeInit”.

Context argument: filters - {object} - Object with pairs of key, value, where key - name of filter, value - object with it options. On this moment, filter - is just object with options, it is not guiFields instance.

Description: This signal will be emitted before creation of guiFields instances for View filters.

5.2.15 views[path].filters.afterInit

Signal name: “views[” + path + “.filters.afterInit”. For example, for /user/ view: “views[/user/].filters.afterInit”.

Context argument: filters - {object} - Object with pairs of key, value, where key - name of filter, value - guiFields instance.

Description: This signal will be emitted after creation of guiFields instances for View filters.

5.2.16 views[path].beforeInit

Signal name: “views[” + path + “.beforeInit”. For example, for /user/ view: “views[/user/].beforeInit”.

Context argument: obj - {object} - Object with following structure: {schema: schema, model: model, template: template,}, where schema - object with view schema options, model - model for which current view is created, template - template of view component.

Description: This signal will be emitted before creation of View Instance.

5.2.17 views[path].afterInit

Signal name: “views[” + path + “].afterInit”. For example, for /user/ view: “views[/user/].afterInit”.

Context argument: obj - {object} - Object with following structure: {view: view}, where view - created View Instance.

Description: This signal will be emitted after creation of View Instance, but before setting actions / child_links / multi_actions / operations / sublinks properties.

5.2.18 views.afterInitEach

Signal name: “views.afterInitEach”.

Context argument: obj - {object} - Object with following structure: { views, path }, where views - object with initialized View Instances, path - path of view that was currently initialized.

Description: This signal will be emitted after initialization of each view.

5.2.19 allViews.inited

Signal name: “allViews.inited”.

Context argument: obj - {object} - Object with following structure: {views: views}, where views - object with created View Instances.

Description: This signal will be emitted after creation of all View Instances, but before setting actions / child_links / multi_actions / operations / sublinks properties.

5.2.20 views[path].created

Signal name: “views[” + path + “].created”. For example, for /user/ view: “views[/user/].created”.

Context argument: obj - {object} - Object with following structure: {view: view}, where view - fully created View Instance.

Description: This signal will be emitted after full creation of View Instance, with set actions / child_links / multi_actions / operations / sublinks properties.

5.2.21 allViews.created

Signal name: “allViews.created”.

Context argument: obj - {object} - Object with following structure: {views: views}, where views - object with fully created View Instances.

Description: This signal will be emitted after creation of all fully View Instances, with set actions / child_links / multi_actions / operations / sublinks properties.

5.2.22 routes[name].created

Signal name: “routes[” + name + “].created”. For example, for /user/ view: “routes[/user/].created”.

Context argument: route - {object} - Object with following structure: {name: name, path: path, component: component}, where name - name of route, path - template of route’s path, component - component, that will be rendered for current route.

Description: This signal will be emitted after route was formed and added to routes list.

5.2.23 allRoutes.created

Signal name: “allRoutes.created”.

Context argument: routes - {array} - Array with route objects with following structure: {name: name, path: path, component: component}, where name - name of route, path - template of route’s path, component - component, that will be rendered for current route.

Description: This signal will be emitted after all routes was formed and added to routes list.

5.2.24 LocalSettings.property

Signal name: LocalSettings.name + “.” + property. For example, when we set property “skin” to instance of LocalSettings “guiLocalSettings”, signal name will be “guiLocalSettings.skin”.

Context argument: obj - {object} - Object with following structure {type: ‘set’, name: property_name, value: property_value}.

Description: This signal will be executed, when some property will be set to the Instance of LocalSettings class.

5.2.25 GuiCustomizer.beforeInit

Signal name: “GuiCustomizer.beforeInit”.

Context argument: obj - {object} - Instance of GuiCustomizer class.

Description: This signal will be executed before initialization of GuiCustomizer Instance.

5.2.26 GuiCustomizer.afterInit

Signal name: “GuiCustomizer.afterInit”.

Context argument: obj - {object} - Instance of GuiCustomizer class.

Description: This signal will be executed after initialization of GuiCustomizer Instance.

5.2.27 GuiCustomizer.skins_custom_settings.reseted

Signal name: “GuiCustomizer.skins_custom_settings.reseted”.

Context argument: obj - {object} - Instance of GuiCustomizer class.

Description: This signal will be executed after custom settings of GuiCustomizer skin were reset.

5.2.28 GuiCustomizer.skins_custom_settings.saved

Signal name: “GuiCustomizer.skins_custom_settings.saved”.

Context argument: obj - {object} - Instance of GuiCustomizer class.

Description: This signal will be executed after custom settings of GuiCustomizer skin were saved.

5.2.29 GuiCustomizer.skin.name.changed

Signal name: “GuiCustomizer.skin.name.changed”.

Context argument: obj - {object} - Instance of GuiCustomizer class.

Description: This signal will be executed after changing of current GuiCustomizer skin.

5.2.30 GuiCustomizer.skin.settings.changed

Signal name: “GuiCustomizer.skin.settings.changed”.

Context argument: obj - {object} - Instance of GuiCustomizer class.

Description: This signal will be executed after changing of current GuiCustomizer skin’s settings.

5.3 Class inheritance

If you need to redefine some functionality of such base entities as *Model class*, *QuerySet class*, *Fields classes*, you can use JavaScript class inheritance for this purposes. You can create new class, that will be inherited from some base VST Utils class and contain some new functionality.

Let’s look on the examples.

5.3.1 Creation of custom Model or QuerySet class

There is only one base class for models (Class Model) and one base class for querysets (Class QuerySet) in VST Utils. These classes are stored in following variables:

- **guiModels** - {object} - variable for storing model classes (base Class Model available as guiModels.Model);
- **guiQuerySets** - {object} - variable for storing queryset classes (base Class QuerySet available as guiQuerySets.QuerySet).

So if you want to add some custom model class, for example, for User Model, you need to write something like this:

```
guiModels.UserModel = class UserModel extends guiModels.Model {
  /*
    some code here
  */
}
```

Name of new custom model, that would be a key in guiModels object, is extremely important. This key should be formed as model_name + “Model”:

```
/**
 * Simple example of model options from OpenAPI Schema.
 * For creation of new custom Model you should not create variable like model_
 * → options.
 */
const model_options = {
  /* ... */
  name: "User";
  /* ... */
};

const user_guiModel_name = model_options.name + "Model";

guiModels[user_guiModel_name] = class UserModel extends guiModels.Model {
  /*
    some code here
  */
}
```

So if you model name is “User”, then key for guiModels should be equal to the “UserModel”. If model name is “Abc”, key - “AbcModel” and so on.

Names of those keys are so important, because during parsing of OpenAPI schema and creation of base entities (models and querysets) VST Utils will automatically checks is there some custom class for this entity in classes store objects (guiModels and guiQuerySets). If there is some relative class for this entity, VST Utils will use it for creation of new model/queryset, otherwise, it will use base class (guiModels.Model and guiQuerySets.QuerySet).

The same principle works and for custom QuerySets. For example, if you want to create custom QuerySet, that will be used for User Model, name (key in guiQuerySets) of this should be equal to “UserQuerySet”. If model name is “Abc”, name (key) - “AbcQuerySet”.

```
guiQuerySets.UserQuerySet = class UserQuerySet extends guiQuerySets.QuerySet {
  /*
    some code here
  */
}
```

5.3.2 Creation of custom Field class

If you need to create some custom field, you also can use mechanism of JavaScript class inheritance. All Fields classes are stored in **guiFields** object, so if you want to add some new field, you should write something like this:

```
guiFields.custom_field = class CustomField extends guiFields.base {
  /*
    some code here
  */
}
```

Here, in JavaScript class, you can redefine some base properties and methods, that will be available for your new `custom_field`. But if you want to change some view properties of you field, you should write some Vue mixin:

```
const custom_field_mixin = {
  /*
    some code here
  */
};

guiFields.custom_field = class CustomField extends guiFields.base {
  /*
    some code here
  */

  /**
   * Redefinition of base guiField static property 'mixins'.
   * Here we just add some additional features to Vue component for guiFeilds.base.
   * If you want to add some definitely new Vue component for your custom_field_
   ↪ (without mixing to the guiFields.base component)
   * you can write something like this: 'return [custom_field_mixin];'.
   */
  static get mixins() {
    return super.mixins.concat(custom_field_mixin);
  }
}
```

5.4 Customizing field's Vue component

Fields' components are being rendered using `Field.mixins()` static method. You can append any mixins (Vue components) to returning array. To change field appearance mixin must define `render` method.

5.5 Customizing field using dynamic format

Setting `field.format` to `dynamic` you can provide `field.additionalProperties.callback` and customize field on every rendering, for example:

```
function field1_callback(parent_values = {}) {
  return {
    format: someCondition(parent_values) ? 'string' : 'integer'
  };
}

tabSignal.connect('models[Model1].fields.beforeInit', (fields) => {
  if (fields.hasOwnProperty('field1')) {
    fields.field1.format = 'dynamic';
    fields.field1.additionalProperties.callback = field1_callback;
  }
})
```

Real field that will be displayed will receive options from `DynamicField.options` and data returned from `field.options.additionalProperties.callback`.

VST UTILS JAVASCRIPT FUNCTIONS

List of common JS functions that you can use in your application.

6.1 String.format()

Description: This function gets list or associative array with values, that should change some keys in the string.

6.2 String.format_keys()

Description: Function searches and returns all {keys} as list.

6.3 trim(s)

Arguments:

- s: {string} - string to trim.

Description: Function trims (deletes non-printing characters from the beginning and from the end of string).

6.4 capitalizeString(string)

Arguments:

- string: {string} - string to edit.

Description: Returns string with first letter in upper case and others in lower case.

6.5 sliceLongString(string="", valid_length=100)

Arguments:

- string: {string} - String for slicing. Default value: "".
- valid_length: {number} - Amount of letters. Default value: 100.

Description: Function compares string's length with the 'valid_length'. If 'string.length' > 'valid_length', then function slices string to the 'valid_length' and adds '...' at the end of sliced string.

6.6 isEmptyObject(object)

Arguments:

- object: {object} - object to check.

Description: Checks object is empty (object has no properties). Returns true if object is empty, otherwise, returns false.

6.7 addCssClassesToElement(element="", title="", type="")

Arguments:

- element: {string} - Name of element. Default value: "";
- title: {string} - Title of element. Default value: "";
- type: {string} - Type of element. Default value: "".

Description: Returns string contains css class(-es), that were formed based on different combinations of arguments.

6.8 getCookie(name)

Arguments:

- name: {string} - Name of cookie.

Description: Returns cookie value if it exists. Otherwise, returns null.

6.9 deepEqual(x, y)

Arguments:

- x: {object} - First object to compare;
- y: {object} - Second object to compare.

Description: Function checks types of x and y. If both are objects, then function checks that all properties of those objects are equal. Otherwise, function compares x and y with the === operator.

6.10 stringToBoolean(string)

Arguments:

- string: {string} - String to check on boolean value.

Description: Function checks string for keeping boolean value and return this value.

6.11 oneCharNumberToTwoChar(n)

Arguments:

- n: {number} - Number to transform.

Description: Function converts numbers from 0 to 9 into 00 to 09.

6.12 allPropertiesIsObjects(obj)

Arguments:

- obj: {object} - Object to check.

Description: Function checks that all properties of object are also objects. Returns boolean value.

6.13 arrayBufferToBase64(buffer)

Arguments:

- buffer: {array} - Instance of ArrayBuffer.

Description: Function converts instance of ArrayBuffer to Base64.

6.14 randomString(length, abc="qwertyuiopasdfghjklzxcvbnm012364489")

Arguments:

- length: {number} - Random string's length;
- abc: {string} - String with chars, that can be used in random string. Default value: "qwertyuiopasdfghjklzxcvbnm012364489".

Description: Function forms random string and returns it.

6.15 hexToRgbA(hex, alpha=1)

Arguments:

- hex: {string} - Color in HEX format (#fefefe);
- alpha: {number} - Opacity value of RGBA color (0-1). Default value: 1.

Description: Function converts color from hex to rgba.

6.16 getTimeInUptimeFormat(time)

Arguments:

- time: {number} - Time in seconds.

Description: Function returns time in uptime format.

6.17 findClosestPath(paths, current_path)

Arguments:

- paths: {array} - Array with paths({string});
- current_path: {string} - Path, based on which function makes search.

Description: Function, that finds the most appropriate (closest) path from path array to current_path. It's supposed, that values in 'paths' array were previously sorted. It's supposed, that 'paths' array does not contain all application paths.

6.18 cleanAllCacheAndReloadPage()

Description: Function, that cleans files cache, unregisters current Service Worker instance and reloads page.

6.19 cleanOpenApiCacheAndReloadPage()

Description: Function, that removes OpenAPI schema from cache and reloads page.

6.20 updateGuiVersionsInLocalStorage()

Description: Function saves to the Local Storage values of global gui versions variables.

6.21 createDomElement(type, attributes, props)

Arguments:

- type: {string} - Type (tag) of DOM element;
- attributes: {array} - Array of objects - DOM element attributes(key, value);
- props: {object} - Object with properties of DOM element.

Description: Function creates DOM element and sets it attributes and props.

6.22 onLoadingErrorHandler(event)

Arguments:

- event: {object} - Error event.

Description: Handler for window.onerror event, that should be called during app dependencies loading, if some error occurred in content of loaded files.

APPLICATION LOADING ON FRONTEND

Loading of VST Utils application on client is done by **app-loader.js** script.

7.1 Application loading steps

To load application **app-loader.js** is going through following steps:

1. loading of app's dependencies (via *AppDependenciesLoader class*):
 - OpenAPI schema;
 - static files (JS, CSS, TPL);
2. creating of app instance (*App class*) and saving it to the global 'app' variable;
3. invoking of *app.start()* method. This method initiates *parsing of OpenAPI schema and generating of frontend Models and Views* and mounts root Vue instance to the DOM, after all *Models objects* and *Views objects* was generated.

7.2 Optimization of application loading

VST Utils tries to load your application as fast as possible, that's why VST Utils uses Service Worker and IndexedDB during its work.

Service Worker is used for caching of:

- static files (CSS, JS, fonts, images);
- offline fallback page (offline.html).

Service worker is a great tool, that does all dirty things with caching for us, but, unfortunately, it works only if application is working on the host with HTTPS.

We can't be enough sure, that all applications, based on VST Utils, will be always working under HTTPS, that why we also use IndexedDB for caching the most heavy API requests.

IndexedDB is used for caching of:

- OpenAPI schema;
- list of available interface languages (interface localization);
- dicts with translations for different languages.

Service Worker and IndexedDB will store this cache while application version is staying the same. Once application version changed, Service Worker and IndexedBD caches will automatically clean and application will load new OpenAPI schema, new static files and so on. That's why the longest application loading will be, when user loads application of new version. Each time user tries to load application of the same version as he loaded before, application will be loading much faster, than at the first time, because of using Service Worker and IndexedDB caches.

7.3 FilesCache class

FilesCache is an abstraction, that is responsible for setting/getting files' content (strings, json - transformed to string) to/from cache. In current realization of VST Utils cache is stored in the [IndexedDB²²](#).

7.3.1 Properties:

- **indexed_db** - object - realization of IndexedDB in current browser.
- **db_name** - string - name of database.
- **db_version** - number - version of database.
- **store_name** - string - name of database's store.

7.3.2 Methods:

constructor(opt={})

Arguments:

- **opt: {object}** - object with options for FilesCache instance, that can redefine some base properties of instance.

Description: Standard constructor of JS class. This method creates new FilesCache instance with current arguments.

connectDB()

Description: Method, that returns promise of getting connection to current instance's database.

getFile(file_name)

Arguments:

- **file_name: {string}** - Name of file.

Description: Method, that returns promise to get file from FilesCache instance's database.

²² https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API/Basic_Concepts_Behind_IndexedDB

setFile(file_name, file_data)**Arguments:**

- file_name: {string} - Name of file.
- file_data: {string} - Content of file.

Description: Method, that returns promise to save file in FilesCache instance's database.

delFile(file_name)**Arguments:**

- file_name: {string} - Name of file.

Description: Method, that returns promise to delete file from FilesCache instance's database.

deleteAllCache()

Description: Method, that returns promise to delete all files from FilesCache instance's database (delete FilesCache instance's database).

7.4 AppDependenciesLoader class

AppDependenciesLoader is an abstraction, that is responsible for loading of App dependencies (OpenAPI schema, static files) and appending them to the page. Also this class has methods for creating and adding to the page 'Loader block' - DOM element, that collects loading logs and shows status of app dependencies loading. This class loads only app's dependencies, it does not create/load app instance.

7.4.1 Properties:

- **openApiLoader** - Object, that has methods for loading of OpenAPI schema. Instance of *OpenApiLoader class*.
- **filesLoader** - Object, that has methods for loading of static files (js, css, tpl) and appending them to the page. Instance of *StaticFilesLoader class*.
- **loading** - Boolean property, that means is loader loading something from outer resources right now or not.
- **errors** - Array, that collects errors occurred during app's dependencies loading.

7.4.2 Methods:

constructor(resource_list, cache)**Arguments:**

- resource_list: {array} - array, with objects, containing info about static files, that should be loaded (name(url), type, priority).
- cache: {object} - Object, that has methods for manipulating with cache. It is supposed to be instance of *FilesCache class*.

Description: Standard constructor of JS class. This method creates new AppDependenciesLoader instance with current arguments.

addLoaderBlockToPage()

Description: Method creates and adds to the page root DOM element, that will show loading status and collect loading logs. This DOM element has children elements:

- loading status;
- loading progress bar;
- reload cache button;
- project info table (will be shown only if some error occurs);
- loading logs wrapper (will be shown only if some error occurs).

formLogsWrapper()

Description: Method, that forms DOM elements, which will store loading logs.

formProjectInfoTable()

Description: Method forms DOM element - table, that stores info about project.

setLoadingOperation(operation)

Arguments:

- operation: {string} - String with name of loading operation.

Description: Method sets current loading operation to one of the children DOM elements of Loader block.

showLoadingAnimation()

Description: Method shows loading animation while some dependencies are loading from server.

setLoadingProgress(width)

Arguments:

- width: {number} - Value of loading progress bar width.

Description: Method, that changes loading progress bar value.

hideLoaderBlock()

Description: Method, that firstly hides loader block and then removes it from the DOM.

appendLog(data, extendData)**Arguments:**

- data: {object | string} - Logging message.
- extendData: {object} - Additional logging message.

Description: Method, that adds logs of files loading.

appendError(exception, extendData)**Arguments:**

- exception: {object | string} - Error object or string.
- extendData: {object} - Additional logging message.

Description: Method, that adds to the html document info about file loading error.

showUpdatingAppVersionMessage()

Description: Method shows message about updating of app version.

loadAndAppendDependencies()

Description: Method returns promise to load all dependencies and append them to the page. Main method of current class. This method creates and add to the page DOM element, that shows loading status and collects loading logs, loads app dependencies(OpenAPI schema, static files) and appends them to the page.

loadDependencies()

Description: Method returns promise to load all app's dependencies.

appendDependencies(dependencies)**Arguments:**

- dependencies: {array} - Response array, connecting loaded OpenAPI schema and files.

Description: Method returns promise to append dependencies(static files) to the page.

7.5 OpenApiLoader class

OpenApiLoader is an abstraction, that is responsible for loading of OpenAPI schema. OpenApiLoader has methods for loading of OpenAPI schema from API as well as from cache.

7.5.1 Properties:

- **cache** - object, that manages operations connected with caching of API responses. It is supposed to be instance of *FilesCache class*.

7.5.2 Methods:

constructor(cache)

Arguments:

- cache: {object} - object, that manages operations connected with caching of API responses. It is supposed to be instance of *FilesCache class*.

Description: Standard constructor of JS class. This method creates new OpenApiLoader instance with current arguments.

loadSchema()

Description: Method, that promises to load OpenApi schema. According to the situation it loads OpenAPI schema from API or from cache.

loadSchemaFromApi()

Description: Method, that promises to load OpenApi schema from API.

loadSchemaFromCache()

Description: Method, that promises to load OpenApi schema from cache.

7.6 StaticFilesLoader class

StaticFilesLoader is an abstraction, that is responsible for the loading of app's static files (js, css, tpl) and appending them to the DOM.

7.6.1 Properties:

- **resource_list** - array, with objects, containing info about static files, that should be loaded (name(url), type, priority).

7.6.2 Methods:

constructor(resource_list)

Arguments:

- resource_list: {array} - array, with objects, containing info about static files, that should be loaded (name(url), type, priority).

Description: Standard constructor of JS class. This method creates new StaticFilesLoader instance with current arguments.

loadAllFiles()

Description: Method, that loads all files form resource_list. Method returns promise of files loading.

checkAllFilesLoaded(response)

Arguments:

- response: {array} - List of responses on files loading requests.

Description: Method checks, that all files were loaded with 200 status.

appendFilesSync(response, index, callbacks)

Arguments:

- response: {array} - List of responses on files loading requests.
- index: {number} - List index of element from resource list and response arrays.
- callbacks: {object} - Dict with callbacks.

Description: Method, that appends files synchronously (in 'priority' order) to the page. Firstly, current method adds to the page file with '0' index, then it appends file with '1' index and so on.

appendFile_js(file, content)

Arguments:

- file: {object} - Object with file properties (type, name(url)).
- content: {string} - File's content.

Description: Method, that appends JS type file to the page.

appendFile_css(file, content)

Arguments:

- file: {object} - Object with file properties (type, name(url)).
- content: {string} - File's content.

Description: Method, that appends CSS type file to the page.

appendFile_tpl(file, content)

Arguments:

- file: {object} - Object with file properties (type, name(url)).
- content: {string} - File's content.

Description: Method, that appends TPL type file to the page.

PARSING OF OPENAPI SCHEMA

OpenAPI schema stores info about API Models and Paths, where these models can be available (views). VST Utils parsers OpenAPI schema and generates based on it Models objects and Views objects, that can be used on client (frontend) for SPA generating.

Parsing of OpenAPI schema for getting Models data is done via *ModelConstructor class*. Generation of Models objects, on the final step of parsing, is done via *Model class* (or one of its *children*).

Parsing of OpenAPI schema for getting Views data is done via *ViewConstructor class*. Generation of Views objects, on the final step of parsing, is done via *View class*.

8.1 BaseEntityConstructor class

BaseEntityConstructor is a parent class for *ModelConstructor class* and *ViewConstructor class*. BaseEntityConstructor class stores common methods for these both classes.

8.1.1 Properties:

- **dictionary** - object, storing info about properties names in OpenAPI Schema and some settings for views of different types. It knows how some properties are named in current swagger version.

8.1.2 Methods:

constructor(openapi_dictionary)

Arguments:

- openapi_dictionary: {object} - object, storing info about properties names in OpenAPI Schema and some settings for views of different types. It knows how some properties are named in current swagger version.

Description: Standard constructor of JS class. This method creates new BaseEntityConstructor instance with current arguments.

getModelRefsProps()

Description: Method, that returns array with properties names, that store reference to model.

getFieldFormat(field)**Arguments:**

- field: {object} - Object with field options.

Description: Method, that defines format of current field.

8.2 ModelConstructor class

ModelConstructor is a class, that have methods for parsing of OpenAPI schema and generating of Models objects based on the result of parsing. ModelConstructor is a child class of *BaseEntityConstructor class*.

8.2.1 Properties:

- **dictionary** - object, storing info about properties names in OpenAPI Schema and some settings for views of different types. It knows how some properties are named in current swagger version.
- **pk_names** - array, with names of fields, that can have role of PK/ID field.
- **classes** - object, with available Model classes. For example, we can have some base *Model class* and *some CustomModel class*.

8.2.2 Methods:

constructor(openapi_dictionary, models_classes)**Arguments:**

- openapi_dictionary: {object} - object, storing info about properties names in OpenAPI Schema and some settings for views of different types. It knows how some properties are named in current swagger version.
- models_classes: {object} - object, with available Model classes. For example, we can have some base *Model class* and *some CustomModel class*.

Description: Standard constructor of JS class. This method creates new ModelConstructor instance with current arguments.

getModelsList(openapi_schema)**Arguments:**

- openapi_schema: {object} - Object, storing OpenAPI Schema.

Description: Method, that returns Models list, from OpenAPI schema.

getModelFieldsList(model)**Arguments:**

- model: {object} - Schema of model from OpenAPI schema.

Description: Method, that returns list of fields for current model.

getModelRequiredFieldsList(model)**Arguments:**

- model: {object} - Schema of model from OpenAPI schema.

Description: Method, that returns list of required fields' names for current model.

getModelFieldFormat(field)**Arguments:**

- field: {object} - Field from OpenAPI's Model schema.

Description: Method, that defines format of current field.

generateModelFields(model, model_name)**Arguments:**

- model: {object} - Schema of model from OpenAPI schema.
- model_name: {string} - Name of model.

Description: Method, that returns object with guiFields for current Model. Method defines appropriate guiField for every field from OpenAPI's Model schema.

getModelsConstructor(model)**Arguments:**

- model: {string} - Name of model.

Description: Method, that returns Model class (class, that will be used for creating of Model object based on the OpenAPI's Model schema), appropriate for current model name.

generateModels(openapi_schema)**Arguments:**

- openapi_schema: {object} - Object, storing OpenAPI Schema.

Description: Method, that generates Models objects based on OpenAPI schema. Method returns dict with generating models.

8.3 ViewConstructor class

ModelConstructor is a class, that have methods for parsing of OpenAPI schema and generating of Views objects based on the result of parsing. ViewConstructor is a child class of *BaseEntityConstructor* class.

8.3.1 Properties:

- **dictionary** - object, storing info about properties names in OpenAPI Schema and some settings for views of different types. It knows how some properties are named in current swagger version.
- **models** - Object with Models objects, generated based on OpenAPI Schema.

8.3.2 Methods:

constructor(openapi_dictionary, models)

Arguments:

- openapi_dictionary: {object} - object, storing info about properties names in OpenAPI Schema and some settings for views of different types. It knows how some properties are named in current swagger version.
- models: {object} - Object with Models objects, generated based on OpenAPI Schema.

Description: Standard constructor of JS class. This method creates new ViewConstructor instance with current arguments.

getPaths(openapi_schema)

Arguments:

- openapi_schema: {object} - Object, storing OpenAPI Schema.

Description: Method, that returns paths list from OpenAPI Schema.

getPathOperationId(path_obj_prop)

Arguments:

- path_obj_prop: {object} - Property of path object, from OpenAPI's path dict.

Description: Method, that returns 'operation_id' property of current path type object (path_obj_prop).

getTypesOperationAlwaysToAdd()

Description: Method, that returns Array with views types, to which ViewConstructor should always add operations from dictionary.

getViewSchema_name(path)**Arguments:**

- path: {string} - Key of path object, from OpenAPI's path dict.

Description: Method, that returns path's name.

getViewSchema_baseOptions(path)**Arguments:**

- path: {string} - Key of path object, from OpenAPI's path dict.

Description: Method, that returns base options of view schema.

getViewSchema_filters(operation_id_filters, path_obj_prop)**Arguments:**

- operation_id_filters: {object} - Filters property from operation_id_options.
- path_obj_prop: {object} - Property of path object, from OpenAPI's path dict.

Description: Method, that returns object with filters for current path.

generateViewSchemaFilters(operation_id_filters, path_obj_prop, path)**Arguments:**

- operation_id_filters: {object} - Filters property from operation_id_options.
- path_obj_prop: {object} - Property of path object, from OpenAPI's path dict.
- path: {string} - Path of view.

Description: Method, that generates new guiField objects for View filters.

getFilterFormat(filter)**Arguments:**

- filter: {object} - Object with filter options (object from View schema).

Description: Method, that defined format for filter's guiField object.

getViewSchema_operationIdOptions(operation_id, path, path_obj_prop)**Arguments:**

- operation_id: {string} - 'operation_id' value.
- path: {string} - Key of path object, from OpenAPI's path dict.
- path_obj_prop: {object} - Property of path object, from OpenAPI's path dict.

Description: Method, that return operation_id options for view schema. It gets 'operation_id' options from ope-napi_dictionary and sets them.

getModelNameLink(obj, max_level=0, level=0)

Arguments:

- obj: {object} - Property of path object, from OpenAPI's path dict, for which method should find Model name.
- max_level: {number} - Max level of inner recursion.
- level: {number} - Current level of recursion.

getModelName(path_obj_prop)

Arguments:

- path_obj_prop: {object} - Property of path object, from OpenAPI's path dict.

Description: Method, that returns name of Model, connected with current path type object (path_obj_prop).

getViewSchema_model(path_obj_prop)

Arguments:

- path_obj_prop: {object} - Property of path object, from OpenAPI's path dict.

Description: Method, that returns Model, connected with current path type object (path_obj_prop).

getViewTemplate(schema)

Arguments:

- schema: {object} - View schema.

Description: Method, that returns template for a current view schema.

getViews(constructor, openapi_schema)

Arguments:

- constructor: {class} - *View class* - constructor, that returns View object.
- openapi_schema: {object} - Object, storing OpenAPI Schema.

Description: Method, that creates views based on OpenAPI Schema.

internalLinkIsOperation(name, path_obj)

Arguments:

- name: {string} - Name of a link obj.
- path_obj: {object} - View object of a path, for which internal links are setting.

Description: Method, that checks: is current link an operation for this path_obj.

getInternalLinkObj_extension(link_name, link_type, path_obj)**Arguments:**

- link_name: {string} - Name of a link.
- link_type: {string} - Type of link object (child_links, actions, operations, sublinks).
- path_obj: {object} - View object for a path (object FROM which link will be formed).

Description: Method, that returns extension from opeanapi_dictionary for current link obj.

isPathObjSchemaEmpty(path_obj)**Arguments:**

- path_obj: {object} - View object for a link path.

Description: Method, that defines emptiness of path_obj.

getInternalLinkObj(link_name, link_type, link, link_obj, path_obj)**Arguments:**

- link_name: {string} - Name of a link.
- link_type: {string} - Type of link object (child_links, actions, operations, sublinks).
- link: {string} - Real path of link_obj.
- link_obj: {object} - View object for a link (object TO which link will be formed).
- path_obj: {object} - View object for a path (object FROM which link will be formed).

Description: Method, that returns object for a current link.

getViewInternalLinks(views, path)**Arguments:**

- views: {object} - Dict with view objects.
- path: {string} - Path of current view.

Description: Method, that finds and returns internal links(links for another views) for a current view.

getViewMultiActions(views, path)**Arguments:**

- views: {object} - Dict with view objects..
- path: {string} - Path of current view.

Description: Method, that finds and returns multi_actions for a current view. Multi_actions - actions/operations, that can be called for a list of instances.

connectPageAndListViews(*views*, *page_path*)**Arguments:**

- *views*: {object} - Dict with view objects..
- *page_path*: {string} - Path of page view.

generateViews(*constructor*, *openapi_schema*)**Arguments:**

- *constructor*: {class} - *View class* - constructor, that returns View object.
- *openapi_schema*: {object} - Object, storing OpenAPI Schema.

Description: Method, that returns dict with views, ready to use.

8.4 SubViewWithoutApiPathConstructor class

SubViewWithoutApiPathConstructor is a class, that have methods for creation of SubViews of guiViews - views, paths of which do not exist in API, but they should be in GUI.

For example, we have some paths in API:

- /foo/;
- /foo/{pk}/;
- /bar/;
- /bar/{pk}/.

And we do not have following paths in API:

- /foo/{pk}/bar/;
- /foo/{pk}/bar/{bar_id}/.

But we want them exist in GUI.

Current class creates views for following paths. All API requests from /foo/{pk}/bar/{bar_id}/ view will be send to the /bar/{pk}/ API path.

8.4.1 Properties:

- **view_constr** - object, that has methods for parsing of OpenAPI Schema and Views generator. Instance of *ViewConstructor class*.
- **path_prefix** - string, containing path prefix, that will be added to the SubView. For example, for to get SubView /foo/{pk}/bar/, we need to add prefix /foo/{pk}/ to the /bar/ path.

8.4.2 Methods:

constructor(openapi_dictionary, models, opt={})

Arguments:

- openapi_dictionary: {object} - object, storing info about properties names in OpenAPI Schema and some settings for views of different types. It knows how some properties are named in current swagger version.
- models: {object} - Object with Models objects, generated based on OpenAPI Schema.
- opt: {object} - Object, with some setting (prefix, for example).

Description: Standard constructor of JS class. This method creates new SubViewWithoutApiPathConstructor instance with current arguments.

generateSubView(views, path, new_path)

Arguments:

- views: {object} - Dict with view objects..
- path: {string} - Path of view, that should be cloned as SubView.
- new_path: {string} - Path of SubView.

Description: Method, that returns new SubView object.

getSubViewMixin()

Description: Method, that returns mixin for SubView Vue component.

ROOT CLASSES PROVIDING APPLICATION WORK

VST Utils has several JavaScript classes, instances of which VST Utils uses for application work.

9.1 App class

App class is the main class for application. This class is an abstraction for the whole application. Let's look what properties and methods instances of this class have.

9.1.1 Properties:

- **api** - object, that manages connection with API (sends API requests). Instance of *ApiConnector class*;
- **models** - dict, that stores all models, generated from OpenAPI schema;
- **views** - dict, that stores all views, generated for all paths from OpenAPI schema;
- **error_handler** - object, that handles errors. Instance of *ErrorHandler class*;
- **languages** - array, that stores objects, containing language's name and code;
- **translation** - dict, that stores translations for each language;
- **user** - object, that stores data of authorized user;
- **application** - main(root) Vue instance for current application, that has access to the app router, created by *RouterConstructor class*, and to the app store, created by *StoreConstructor class*.

9.1.2 Methods:

constructor(api_config, openapi, cache)

Arguments:

- **api_config**: {object} - Dict with options for ApiConnector constructor.
- **openapi**: {object} - Object with OpenAPI schema.
- **cache**: {object} - Cache instance (is supposed to be instance of FilesCache class).

Description: Standard constructor of JS class. This method creates new App instance with current arguments.

start()

Description: Method starts application work: loads necessary initial data from API, registers all models and views, create root Vue instance and mounts it to the DOM.

initModels()

Description: Method initiates generating of app models and saves them to the 'models' property of App instance (this.models).

generateModels()

Description: Method generates app Models based on the OpenAPI Schema.

initViews()

Description: Method initiates generating of app Views and saves them to the 'views' property of App instance (this.views).

generateViews()

Description: Method generates app Views based on the OpenAPI Schema.

prepareViewsModelsFields()

Description: Method runs through all views and handles all fields with additionalProperties.

setLanguage(lang)

Arguments:

- lang: {string} - Code of language, that should be set as current App language.

Description: Method returns a promise of applying some language to app interface. This method is supposed to be called after app was mounted.

_prefetchTranslation(lang)

Arguments:

- lang: {string} - Code of language, that should be set as current App language.

Description: Method returns a promise of checking that current language exists and translations for language is loaded.

mountApplication()

Description: Method, that creates application's root Vue instance, containing router (created by *RouterConstructor class*), store (created by *StoreConstructor class*), and mounts it to the DOM.

9.2 ApiConnector class

ApiConnector is an abstraction, that manages all requests to the API. It forms, sends API requests and returns API responses. Let's look what properties and methods instances of this class have.

9.2.1 Properties:

- **config** - object with config properties for Api connector;
- **openapi** - object, containing OpenAPI Schema;
- **cache** - object, that manages operations connected with caching of api responses;
- **api** - object with methods for providing Api connection;
- **bulk_collector** - object for collecting several bulk requests into one.

9.2.2 Methods:

constructor(config, openapi, cache)

Arguments:

- config: {object} - Dict with options for Api connector.
- openapi: {object} - Object with OpenAPI schema.
- cache: {object} - Cache instance (is supposed to be instance of FilesCache class).

Description: Standard constructor of JS class. This method creates new ApiConnector instance with current arguments.

query(method, url="", data={})

Arguments:

- method: {string} - Method of HTTP request.
- url: {string} - Relative part of link, to which send API requests.
- data: {object} - Query body.

Description: Method, that sends API request.

bulkQuery(data)

Arguments:

- data: {object} - Body of bulk request.

Description: Method, that collects several bulk requests into one and then sends it to API. It's a kind of debouncer under sendBulk() method.

sendBulk()

Description: Method, that sends one big (collected) bulk request to API.

getHostUrl()

Description: Method returns URL of API host (server).

getTimeZone()

Description: Method returns string, containing time zone of API host.

getStaticPath()

Description: Method returns relative path (from host url) to the directory with static path.

getUserId()

Description: Method returns id of user, that is now authorized and uses application.

loadUser()

Description: Method, that loads data of authorized user.

loadLanguages()

Description: Method, that returns promise to load list of App languages from API.

getLanguagesFromCache()

Description: Method, that returns promise to get list of App languages from cache.

getLanguages()

Description: Method, that returns promise to get list of App languages.

loadTranslations(lang)

Arguments:

- lang: {string} - Code of language, translations of which to load.

Description: Method, that returns promise to load translations for some language from API.

getTranslationsFromCache(lang)

Arguments:

- lang: {string} - Code of language, translations of which to load.

Description: Method, that returns promise to get translations for some language from cache.

getTranslations(lang)

Arguments:

- lang: {string} - Code of language, translations of which to load.

Description: Method, that returns promise to get translations for some language.

9.3 ErrorHandler class

ErrorHandler is an abstraction, that handles errors. Instances of class can transform error to string add show error message to user.

9.3.1 Methods:

constructor()

Description: Standard constructor of JS class. This method creates new ErrorHandler instance.

errorToString(error)

Arguments:

- error: {string | object} - String or object with error message.

Description: Method, that transforms error to string.

showError(to_pop_up, to_console)**Arguments:**

- to_pop_up: {string} - String, that will be shown in pop up notification.
- to_console: {string} - String, that will be logged into console.

Description: Method, that shows error to user.

defineErrorAndShow(error)**Arguments:**

- error: {string | object} - String or object with error message.

Description: Method, that transforms error into string and shows ot to user.

9.4 RouterConstructor class

RouterConstructor is an abstraction, that is responsible for generating of application's router. In current realization of VST Utils, application's router is [Vue Router Instance](#)²³. RouterConstructor forms routes and Vue components (based on views) for these routes.

9.4.1 Properties:

- **views** - object with generated Views objects (instances of *View class*).
- **components_templates** - object with mixins of Views' Vue components of different types (list, page_new, page, page_edit, action). This mixins is supposed to be used for Views, that have description in OpenAPI Schema and they should send some requests to API.
- **custom_components_templates** - object with mixins for Views' Vue components of custom pages (home page, 404 error page). This mixins is supposed to be used for Views, that have no description in OpenAPI Schema and they should not send some requests to API.
- **routes** - array, that stores route objects, containing info about route path and mixins, that should be used for generating of component for current route.

9.4.2 Methods:

constructor(views, components_templates, custom_components_templates)**Arguments:**

- views: {object} - object with generated Views objects (instances of *View class*).
- components_templates: {object} - object with mixins of Views' Vue components of different types (list, page_new, page, page_edit, action). This mixins is supposed to be used for Views, that have description in OpenAPI Schema and they should send some requests to API.
- custom_components_templates: {object} - object with mixins for Views' Vue components of custom pages (home page, 404 error page). This mixins is supposed to be used for Views, that have no description in OpenAPI Schema and they should not send some requests to API.

²³ <https://router.vuejs.org/>

Description: Standard constructor of JS class. This method creates new RouterConstructor instance with current arguments.

getRouteComponent(view)

Arguments:

- view: {object} - View object (instance of *View class*).

Description: Method, that returns mixin for Vue component for a route, connected with current view.

getRouteComponentMixins(view)

Arguments:

- view: {object} - View object (instance of *View class*).

Description: Method, that collects appropriate mixins for a View Vue component into one array and returns it.

getRoutes()

Description: Method, that returns array of routes objects, existing in current App.

formAllRoutes()

Description: Method, that forms array of all routes objects, existing in current App.

formRoutesBasedOnViews()

Description: Method, that forms array of routes objects, existing in current App, and based on App Views, that were created via *View class* and have description in OpenAPI Schema (this.views).

formRoutesBasedOnCustomComponents()

Description: Method, that forms array of routes objects, existing in current App, and based on App custom views components (this.custom_components_templates).

emitSignalAboutRouteCreation(route)

Arguments:

- route: {object} - Object with route properties (name, path, component).

Description: Method emits signal: “route was created”.

getRouter()

Description: Method, that returns new instance of [Vue Router](#)²⁴, containing info about all routes, available in the application. This Router Instance is ready for usage in application.

9.5 StoreConstructor class

StoreConstructor is an abstraction, that is responsible for generating of application's state store. In current realization of VST Utils, application's state store is [Vuex Instance](#)²⁵.

9.5.1 Properties:

- **views** - object with generated Views objects (instances of [View class](#)).
- **store** - object, that stores following properties: [state](#)²⁶, [getters](#)²⁷, [mutations](#)²⁸, [actions](#)²⁹.

9.5.2 Methods:

constructor(views)

Arguments:

- **views**: {object} - object with generated Views objects (instances of [View class](#)).

Description: Standard constructor of JS class. This method creates new StoreConstructor instance with current arguments.

getStore_state()

Description: Method, that forms [state](#)³⁰ property of Store object.

getStore_getters()

Description: Method, that forms store [getters](#)³¹ - properties/methods, that return data from store.

²⁴ <https://router.vuejs.org/>

²⁵ <https://vuex.vuejs.org/>

²⁶ <https://vuex.vuejs.org/guide/state.html>

²⁷ <https://vuex.vuejs.org/guide/getters.html>

²⁸ <https://vuex.vuejs.org/guide/mutations.html>

²⁹ <https://vuex.vuejs.org/guide/actions.html>

³⁰ <https://vuex.vuejs.org/guide/state.html>

³¹ <https://vuex.vuejs.org/guide/getters.html>

getStore_mutations()

Description: Method, that forms store [mutations](#)³² - single way of state changing in Vuex store.

getStore_actions()

Description: Method, that forms store [actions](#)³³ - asynchronous operations.

getStore()

Description: Method, that returns application's state store - new instance of [Vuex Store](#)³⁴, containing info about [state properties](#)³⁵, [getters](#)³⁶, [mutations](#)³⁷, [actions](#)³⁸, needed for the work of application's state store. This Store Instance is ready for usage in application.

9.6 PopUp class

PopUp is an abstraction, that is responsible for displaying pop up messages (notifications) for user. In current realization of VST Utils, PopUp class is working based on the [iziToast](#)³⁹.

9.6.1 Properties:

- **options** - object, storing pop up settings.

9.6.2 Methods:

constructor(options)

Arguments:

- **options:** {object} - object with custom options, that should be used instead of default PopUp class's options.

Description: Standard constructor of JS class. This method creates new PopUp instance with current arguments.

³² <https://vuex.vuejs.org/guide/mutations.html>

³³ <https://vuex.vuejs.org/guide/actions.html>

³⁴ <https://vuex.vuejs.org/>

³⁵ <https://vuex.vuejs.org/guide/state.html>

³⁶ <https://vuex.vuejs.org/guide/getters.html>

³⁷ <https://vuex.vuejs.org/guide/mutations.html>

³⁸ <https://vuex.vuejs.org/guide/actions.html>

³⁹ <https://izitoast.marcelodolza.com/>

`_getPopUpSettings(key, opt={})`

Arguments:

- key: {string} - Type of pop up notification.
- opt: {object} - Object with custom settings of new pop up notification.

Description: Method, that forms settings for new pop up notification.

`_showPopUp(type, opt)`

Arguments:

- type: {string} - Type of pop up notification.
- opt: {object} - Object with settings of pop up notification.

Description: Method, that shows new pop up notification.

`_generatePopUp(type="show", message="", opt={})`

Arguments:

- type: {string} - Type of pop up notification.
- message: {string} - Text of pop up notification's body.
- opt: {object} - Object with custom settings for pop up notification.

Description: Method, that forms settings for new pop up notification and shows it.

`default(message="", opt={})`

Arguments:

- message: {string} - Body text of pop up notification.
- opt: {object} - Object with custom settings for pop up notification.

Description: Method, that generates default pop up notification.

`info(message="", opt={})`

Arguments:

- message: {string} - Body text of pop up notification.
- opt: {object} - Object with custom settings for pop up notification.

Description: Method, that generates info pop up notification.

success(message="", opt={})

Arguments:

- message: {string} - Body text of pop up notification.
- opt: {object} - Object with custom settings for pop up notification.

Description: Method, that generates success pop up notification.

warning(message="", opt={})

Arguments:

- message: {string} - Body text of pop up notification.
- opt: {object} - Object with custom settings for pop up notification.

Description: Method, that generates warning pop up notification.

error(message="", opt={})

Arguments:

- message: {string} - Body text of pop up notification.
- opt: {object} - Object with custom settings for pop up notification.

Description: Method, that generates error pop up notification.

question(message="", answer_buttons = [], opt={})

Arguments:

- message: {string} - Question text.
- answer_buttons: {array} - Array of strings - titles for answer buttons.
- opt: {object} - Object with custom settings for question pop up.

Description: Method, that generates question pop up. Returns promise of getting user's answer.

9.7 LocalSettings class

LocalSettings is an abstraction, that is responsible for manipulating by settings saved to the [Local Storage](#)⁴⁰. It is used for saving some user's local settings to the one property(object) of [Local Storage](#)⁴¹.

For example:

```
window.localStorage.localSettings = {
  hideMenu: true,
  lang: "en",
  skin: "default"
}
```

⁴⁰ <https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage>

⁴¹ <https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage>

9.7.1 Properties:

- **name** - string - Key name of Local Storage's property to which Local Settings will be saved.
- **__settings** - object - Property for storing current settings (including tmpSettings).
- **__tmpSettings** - object - Property for storing temporary settings.
- **__beforeAsTmpSettings** - object - Property for storing setting value, as it was before user set tmpSettings value.

9.7.2 Methods:

constructor(name)

Arguments:

- name: {string} - Key name of Local Storage's property to which Local Settings will be saved.

Description: Standard constructor of JS class. This method creates new LocalSettings instance with current arguments.

sync()

Description: Method syncs this.__settings property with data from window.localStorage[this.name].

get(name)

Arguments:

- name: {string} - Key of property from local settings.

Description: Method returns property, that is stored in local settings at 'name' key.

set(name, value)

Arguments:

- name: {string} - Key of property from local settings.
- value: {any} - Value of property from local settings.

Description: Method sets property value in local settings.

delete(name)

Arguments:

- name: {string} - Key of property from local settings.

Description: Method deletes property, that is stored in local settings at 'name' key.

setIfNotExists(name, value)**Arguments:**

- name: {string} - Key of property from local settings.
- value: {any} - Value of property from local settings.

Description: Method sets property value in local settings, if it was not set before.

setAsTmp(name, value)**Arguments:**

- name: {string} - Key of property from local settings.
- value: {any} - Value of property from local settings.

Description: Method sets temporary property value in local settings.

__removeTmpSettings()

Description: Method removes tmpSettings from current settings and add previous values (if they were).

9.8 GuiCustomizer class

GuiCustomizer is an abstraction, that is responsible for changing GUI skins (themes) and for changing skins' settings.

9.8.1 Properties:

- **skin** - object - Object should have 2 properties:
 - name: {string} - name of current skin(theme);
 - settings: {object} - object with settings of current skin (theme).
- **skins** - object - Object, that stores setting of available skins (themes).
- **skins_custom_settings** - object - Object, that stores user's custom settings for skins.
- **form** - object - Object, that stores options, that will be used in formField.options.form. This object has options for formField real elements. This object mixes skin options with form_base options.
- **form_base** - object - Object, that stores base options for formField.options.form. These options should be available in every skin.
- **formField** - object - Property, that stores instance of *guiField.form* - this field is responsible for representing skins setting to user.
- **skinField** - object - Property, that stores instance of *guiField.choices* - this field is responsible for representing name of selected skin to user.

9.8.2 Methods:

constructor(skin={}, skins={}, custom_setting={})

Arguments:

- skin: {object} - Object should have 2 properties:
 - name: {string} - name of current skin(theme);
 - settings: {object} - object with settings of current skin (theme).
- skins: {object} - Object, that stores setting of available skins (themes).
- custom_setting: {object} - Object, that stores user's custom settings for skins.

Description: Standard constructor of JS class. This method creates new LocalSettings instance with current arguments.

init()

Description: Method, that initiates work of guiCustomizer. It creates skin and form fields, forms guiCustomizer form options.

formCss()

Description: Method, that forms string, which contains values of CSS variables, based of skin.name and skin.settings.

replaceCss(css)

Arguments:

- css: {string} - New CSS.

Description: Method, that deletes previous <style></style> DOM element with CSS variables for previous skin and appends new <style></style> DOM element with new styles.

loadSkinCustomSettings()

Description: Method, that returns custom settings of current skin.

applySkinCustomSettings()

Description: Method, that adds current skin's custom_settings to original settings.

applySkinDefaultSettings()

Description: Method, that adds default settings of current skin to skin.settings.

updateCssVariables()

Description: Method, that updates CSS variables, based on current skin settings and changes skin class of 'body' - DOM element. Method, activates current skin settings.

resetSkinSettings()

Description: Method, that resets custom skin settings to default.

saveSkinSettings()

Description: Method, that saves custom skin settings.

resetSkinSettingsAndUpdateCss()

Description: Method, that resets custom skin settings to default and updates skin settings.

updateSkinSettings()

Description: Method, that updates current skin settings.

setSkin(skin)

Arguments:

- skin: {string} - Name of selected skin.

Description: Method, that changes selected skin.

setSkinSettings(settings)

Arguments:

- settings: {object} - Object with new settings.

Description: Method, that changes current skin settings.

HTTP ROUTING TABLE

/community_template

GET /community_template/, 28
 GET /community_template/{id}/, 28
 POST /community_template/{id}/use_it/, 29

/group

GET /group/, 29
 GET /group/{id}/, 31
 GET /group/{id}/group/, 33
 GET /group/{id}/group/{group_id}/, 35
 GET /group/{id}/group/{group_id}/variables/, 38
 GET /group/{id}/group/{group_id}/variables/{variables_id}/, 39
 GET /group/{id}/host/, 42
 GET /group/{id}/host/{host_id}/, 43
 GET /group/{id}/host/{host_id}/variables/, 46
 GET /group/{id}/host/{host_id}/variables/{variables_id}/, 48
 GET /group/{id}/variables/, 50
 GET /group/{id}/variables/{variables_id}/, 52
 POST /group/, 30
 POST /group/{id}/copy/, 33
 POST /group/{id}/group/, 34
 POST /group/{id}/group/{group_id}/copy/, 37

POST /group/{id}/group/{group_id}/set_owner/, 37
 POST /group/{id}/group/{group_id}/variables/, 39
 POST /group/{id}/host/, 42
 POST /group/{id}/host/{host_id}/copy/, 45

POST /group/{id}/host/{host_id}/set_owner/, 46

POST /group/{id}/host/{host_id}/variables/, 47

POST /group/{id}/set_owner/, 50
 POST /group/{id}/variables/, 51

PUT /group/{id}/, 31
 PUT /group/{id}/group/{group_id}/, 35
 PUT /group/{id}/group/{group_id}/variables/{variables_id}/, 40
 PUT /group/{id}/host/{host_id}/, 44
 PUT /group/{id}/host/{host_id}/variables/{variables_id}/, 48
 PUT /group/{id}/variables/{variables_id}/, 52
 DELETE /group/{id}/, 32
 DELETE /group/{id}/group/{group_id}/, 37
 DELETE /group/{id}/group/{group_id}/variables/{variables_id}/, 41
 DELETE /group/{id}/host/{host_id}/, 45
 DELETE /group/{id}/host/{host_id}/variables/{variables_id}/, 50
 DELETE /group/{id}/variables/{variables_id}/, 54
 PATCH /group/{id}/, 32
 PATCH /group/{id}/group/{group_id}/, 36
 PATCH /group/{id}/group/{group_id}/variables/{variables_id}/, 41
 PATCH /group/{id}/host/{host_id}/, 44
 PATCH /group/{id}/host/{host_id}/variables/{variables_id}/, 49
 PATCH /group/{id}/variables/{variables_id}/, 53

/history

GET /history/, 54
 GET /history/{id}/, 55
 GET /history/{id}/facts/, 57
 POST /history/{id}/cancel/, 56
 DELETE /history/{id}/, 56
 DELETE /history/{id}/clear/, 57

/hook

GET /hook/, 57
 GET /hook/{id}/, 58
 POST /hook/, 58
 PUT /hook/{id}/, 59

DELETE /hook/{id}/, 60
PATCH /hook/{id}/, 60

/host

GET /host/, 61
GET /host/{id}/, 62
GET /host/{id}/variables/, 65
GET /host/{id}/variables/{variables_id}/, 66
POST /host/, 61
POST /host/{id}/copy/, 64
POST /host/{id}/set_owner/, 65
POST /host/{id}/variables/, 66
PUT /host/{id}/, 63
PUT /host/{id}/variables/{variables_id}/, 67
DELETE /host/{id}/, 64
DELETE /host/{id}/variables/{variables_id}/, 68
PATCH /host/{id}/, 63
PATCH /host/{id}/variables/{variables_id}/, 68

/inventory

GET /inventory/, 68
GET /inventory/{id}/, 70
GET /inventory/{id}/all_groups/, 72
GET /inventory/{id}/all_groups/{all_groups_id}/, 73
GET /inventory/{id}/all_hosts/, 74
GET /inventory/{id}/all_hosts/{all_hosts_id}/, 75
GET /inventory/{id}/group/, 76
GET /inventory/{id}/group/{group_id}/, 77
GET /inventory/{id}/group/{group_id}/variables/, 80
GET /inventory/{id}/group/{group_id}/variables/{variables_id}/, 82
GET /inventory/{id}/host/, 84
GET /inventory/{id}/host/{host_id}/, 86
GET /inventory/{id}/host/{host_id}/variables/, 89
GET /inventory/{id}/host/{host_id}/variables/{variables_id}/, 90
GET /inventory/{id}/variables/, 93
GET /inventory/{id}/variables/{variables_id}/, 94
POST /inventory/, 69
POST /inventory/import_inventory/, 70
POST /inventory/{id}/copy/, 75
POST /inventory/{id}/group/, 77
POST /inventory/{id}/group/{group_id}/copy/, 79
POST /inventory/{id}/group/{group_id}/set_owner/, 80
POST /inventory/{id}/group/{group_id}/variables/, 81
POST /inventory/{id}/host/, 85
POST /inventory/{id}/host/{host_id}/copy/, 88
POST /inventory/{id}/host/{host_id}/set_owner/, 88
POST /inventory/{id}/host/{host_id}/variables/, 90
POST /inventory/{id}/set_owner/, 92
POST /inventory/{id}/variables/, 94
PUT /inventory/{id}/, 71
PUT /inventory/{id}/group/{group_id}/, 78
PUT /inventory/{id}/group/{group_id}/variables/{variables_id}/, 83
PUT /inventory/{id}/host/{host_id}/, 86
PUT /inventory/{id}/host/{host_id}/variables/{variables_id}/, 91
PUT /inventory/{id}/variables/{variables_id}/, 95
DELETE /inventory/{id}/, 72
DELETE /inventory/{id}/group/{group_id}/, 79
DELETE /inventory/{id}/group/{group_id}/variables/{variables_id}/, 84
DELETE /inventory/{id}/host/{host_id}/, 88
DELETE /inventory/{id}/host/{host_id}/variables/{variables_id}/, 92
DELETE /inventory/{id}/variables/{variables_id}/, 96
PATCH /inventory/{id}/, 71
PATCH /inventory/{id}/group/{group_id}/, 79
PATCH /inventory/{id}/group/{group_id}/variables/{variables_id}/, 83
PATCH /inventory/{id}/host/{host_id}/, 87
PATCH /inventory/{id}/host/{host_id}/variables/{variables_id}/, 92
PATCH /inventory/{id}/variables/{variables_id}/, 95

/project

GET /project/, 96
GET /project/{id}/, 98
GET /project/{id}/history/, 102
GET /project/{id}/history/{history_id}/, 103
GET /project/{id}/history/{history_id}/facts/, 105

GET /project/{id}/inventory/, 106	102
GET /project/{id}/inventory/{inventory_id}/, 108	POST /project/{id}/history/{history_id}/cancel/, 105
GET /project/{id}/inventory/{inventory_id}/all_projects/, 110	POST /project/{id}/inventory/file_import_inventory/, 106
GET /project/{id}/inventory/{inventory_id}/all_groups/{all_groups_id}/, 111	POST /project/{id}/inventory/import_inventory/, 107
GET /project/{id}/inventory/{inventory_id}/all_hosts/, 112	POST /project/{id}/inventory/{inventory_id}/copy/, 108
GET /project/{id}/inventory/{inventory_id}/all_hosts/{all_hosts_id}/, 113	POST /project/{id}/inventory/{inventory_id}/group/, 109
GET /project/{id}/inventory/{inventory_id}/group/{group_id}/, 114	POST /project/{id}/inventory/{inventory_id}/group/{group_id}/, 110
GET /project/{id}/inventory/{inventory_id}/group/{group_id}/variables/, 115	POST /project/{id}/inventory/{inventory_id}/group/{group_id}/variables/{variables_id}/, 111
GET /project/{id}/inventory/{inventory_id}/group/{group_id}/variables/{variables_id}/, 119	POST /project/{id}/inventory/{inventory_id}/host/, 112
GET /project/{id}/inventory/{inventory_id}/group/{group_id}/variables/{variables_id}/, 120	POST /project/{id}/inventory/{inventory_id}/host/{host_id}/, 113
GET /project/{id}/inventory/{inventory_id}/host/, 123	POST /project/{id}/inventory/{inventory_id}/host/{host_id}/variables/, 114
GET /project/{id}/inventory/{inventory_id}/host/{host_id}/, 124	POST /project/{id}/inventory/{inventory_id}/host/{host_id}/variables/{variables_id}/, 115
GET /project/{id}/inventory/{inventory_id}/host/{host_id}/variables/, 128	POST /project/{id}/inventory/{inventory_id}/set_owner/, 116
GET /project/{id}/inventory/{inventory_id}/host/{host_id}/variables/{variables_id}/, 129	POST /project/{id}/inventory/{inventory_id}/variables/, 117
GET /project/{id}/inventory/{inventory_id}/variables/, 132	POST /project/{id}/inventory/{inventory_id}/variables/{variables_id}/, 118
GET /project/{id}/inventory/{inventory_id}/variables/{variables_id}/, 134	POST /project/{id}/periodic_task/, 138
GET /project/{id}/module/, 136	POST /project/{id}/periodic_task/{periodic_task_id}/, 142
GET /project/{id}/module/{module_id}/, 137	POST /project/{id}/periodic_task/{periodic_task_id}/set_owner/, 147
GET /project/{id}/periodic_task/, 137	POST /project/{id}/sync/, 148
GET /project/{id}/periodic_task/{periodic_task_id}/, 139	POST /project/{id}/template/{template_id}/execute/, 149
GET /project/{id}/periodic_task/{periodic_task_id}/variables/{variables_id}/, 144	POST /project/{id}/template/{template_id}/execute/, 151
GET /project/{id}/periodic_task/{periodic_task_id}/variables/{variables_id}/, 144	POST /project/{id}/variables/, 153
GET /project/{id}/playbook/, 146	PUT /project/{id}/, 99
GET /project/{id}/playbook/{playbook_id}/, 147	PUT /project/{id}/inventory/{inventory_id}/, 109
GET /project/{id}/template/, 148	PUT /project/{id}/inventory/{inventory_id}/group/{group_id}/, 116
GET /project/{id}/template/{template_id}/, 149	PUT /project/{id}/inventory/{inventory_id}/group/{group_id}/variables/{variables_id}/, 121
GET /project/{id}/variables/, 152	PUT /project/{id}/inventory/{inventory_id}/host/{host_id}/, 125
GET /project/{id}/variables/{variables_id}/, 154	PUT /project/{id}/inventory/{inventory_id}/host/{host_id}/variables/{variables_id}/, 130
POST /project/, 97	PUT /project/{id}/inventory/{inventory_id}/variables/{variables_id}/, 134
POST /project/{id}/copy/, 101	
POST /project/{id}/execute_module/, 101	
POST /project/{id}/execute_playbook/, 101	

```

PUT /project/{id}/periodic_task/{periodic_task_id}/team/{team_id}/,
140 GET /team/{id}/, 156
PUT /project/{id}/periodic_task/{periodic_task_id}/variables/{variables_id}/,
144 GET /team/{id}/user/, 160
PUT /project/{id}/template/{template_id}/, 150 GET /team/{id}/user/{user_id}/, 161
PUT /project/{id}/variables/{variables_id}/, 154 GET /team/{id}/user/{user_id}/settings/,
154 POST /team/{id}/, 156
DELETE /project/{id}/, 101 POST /team/{id}/copy/, 159
DELETE /project/{id}/history/{history_id}/, 105 POST /team/{id}/set_owner/, 159
DELETE /project/{id}/history/{history_id}/clear/, 105 POST /team/{id}/user/{user_id}/change_password/,
105 163
DELETE /project/{id}/inventory/{inventory_id}/, 110 POST /team/{id}/user/{user_id}/copy/,
110 164
DELETE /project/{id}/inventory/{inventory_id}/group/{group_id}/, 117 POST /team/{id}/user/{user_id}/settings/,
117 166
DELETE /project/{id}/inventory/{inventory_id}/group/{group_id}/variables/{variables_id}/,
122 PUT /team/{id}/user/{user_id}/, 162
DELETE /project/{id}/inventory/{inventory_id}/host/{host_id}/, 126 DELETE /team/{id}/, 159
DELETE /project/{id}/inventory/{inventory_id}/host/{host_id}/variables/{variables_id}/,
131 DELETE /team/{id}/user/{user_id}/settings/,
131 168
DELETE /project/{id}/inventory/{inventory_id}/variables/{variables_id}/,
136 PATCH /team/{id}/, 158
DELETE /project/{id}/periodic_task/{periodic_task_id}/, 142 /user
DELETE /project/{id}/periodic_task/{periodic_task_id}/variables/{variables_id}/,
146 GET /user/{id}/, 168
DELETE /project/{id}/template/{template_id}/, 152 GET /user/{id}/settings/, 173
DELETE /project/{id}/variables/{variables_id}/, 155 POST /user/, 169
PATCH /project/{id}/, 100 POST /user/{id}/change_password/, 172
PATCH /project/{id}/inventory/{inventory_id}/, 109 POST /user/{id}/copy/, 172
PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/, 117 DELETE /user/{id}/settings/, 176
PATCH /project/{id}/inventory/{inventory_id}/group/{group_id}/variables/{variables_id}/,
122 PATCH /user/{id}/, 171
PATCH /project/{id}/inventory/{inventory_id}/host/{host_id}/,
126 PATCH /project/{id}/inventory/{inventory_id}/host/{host_id}/variables/{variables_id}/,
131 PATCH /project/{id}/inventory/{inventory_id}/variables/{variables_id}/,
135 PATCH /project/{id}/periodic_task/{periodic_task_id}/,
141 PATCH /project/{id}/periodic_task/{periodic_task_id}/variables/{variables_id}/,
145 PATCH /project/{id}/template/{template_id}/,
151 PATCH /project/{id}/variables/{variables_id}/,
155

```