Alignak Backend Documentation

Release 1.0

David Durieux

October 11, 2015

Contents

Contents:

CHAPTER 1

Introduction

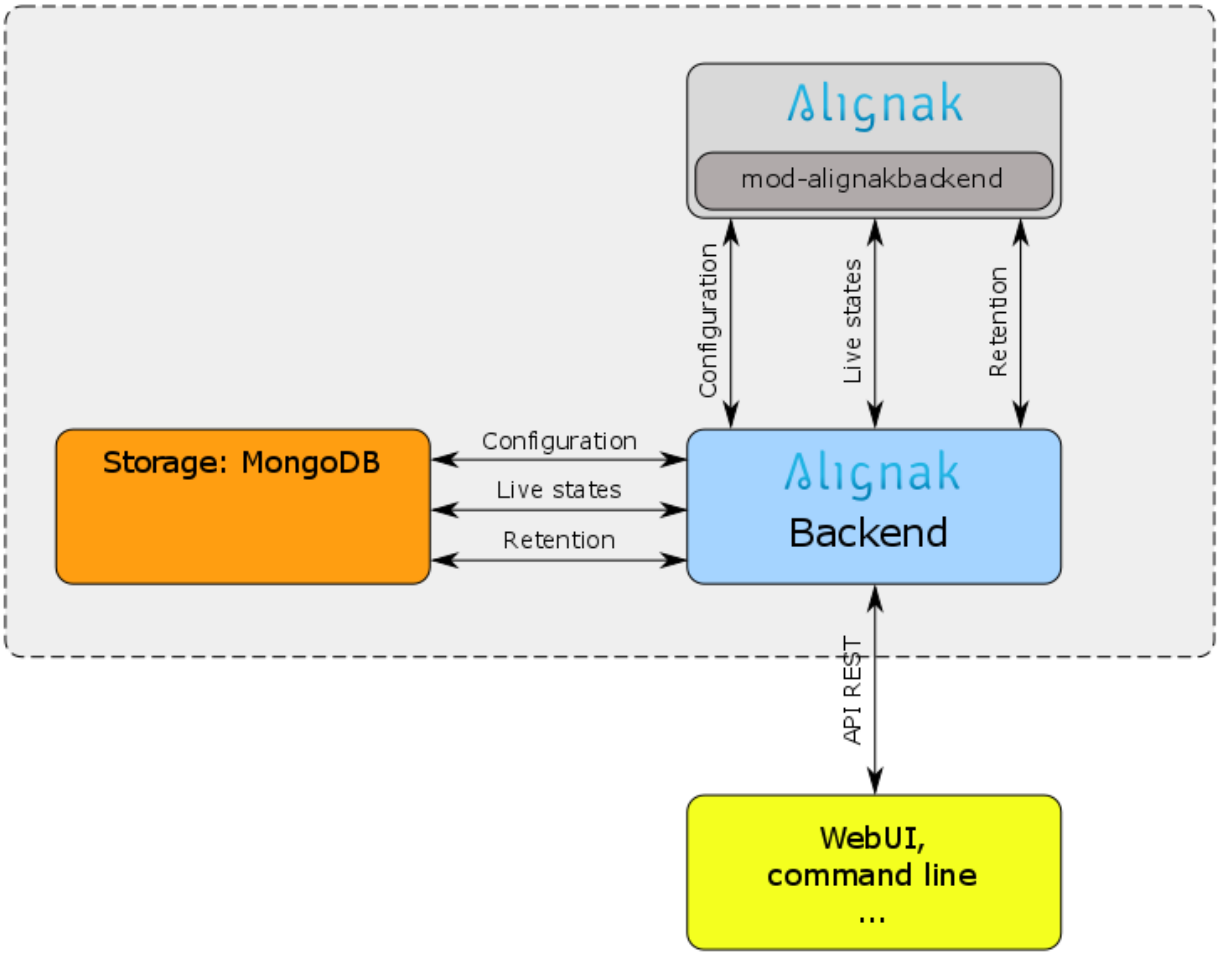
This project is a Alignak Backend. It is used to:

- manage configuration (hosts, services, contacts, timeperiods...)
  - end user (webui, command line...) can get and add configurations elements
  - Alignak get this configuration when start arbiter module
- manage retention
  - Alignak load and save retention information about checks/hosts/services
- manage live states
  - Alignak add/update states of hosts and services
  - end user (webui, command line...) can get these information

CHAPTER 2

Architecture

This is the architecture schema about the backend behaviors:



CHAPTER 3

Installation

3.1 Requirements

Tu use this backend, you need first install and run MongoDB

3.2 Install with pip

3.2.1 With pip

You can install with pip:

```
pip install alignak_backend
```

This not work for the moment because not yet released

3.2.2 From source

You can install it from source:

```
git clone https://github.com/Alignak-monitoring/alignak-backend
cd alignak-backend
pip install .
```

3.2.3 For contributors

If you want to hack into the codebase (e.g for future contribution), just install like this:

```
pip install -e .
```

3.3 Install from source without pip

If you are on Debian:

```
apt-get -y install python python-dev python-pip git
```

Get the project sources:

```
git clone https://github.com/Alignak-monitoring/alignak-backend
```

Install python prerequisites:

```
pip install -r alignak-backend/requirements.txt
```

And install:

```
cd alignak-backend
python setup.py install
```

CHAPTER 4

Run

4.1 Production mode

First create a file anywhere on your system, with name 'alignakbackend.py' and with the content:

```
from alignak_backend.app import app
```

You can use many possibilities, we suggest you with uwsgi and start it in sme directory of file created previously.

With socket (+ nginx / apache in frontal):

```
uwsgi -s /tmp/uwsgi.sock -w alignakbackend:app --enable-threads
```

With http port directly:

```
uwsgi -w alignakbackend:app --socket 0.0.0.0:80 --protocol=http --enable-threads
```

Alignak-backend run on port 80 like specified in arguments, so use:

```
http://ip/
```

4.2 Developer mode

To run in developper mode (mean with few connections), you can start with command:

```
alignak_backend
```

Alignak-backend run on port 5000, so use:

```
http://ip:5000/
```

CHAPTER 5

Developer Interface

This part of documentation is related of API REST for interact with this backend. The examples in this part of documentation use :

- IP as 127.0.0.1
- a resource name as service

5.1 Get all resources available

All resources available in backend is available on root endpoint of backend:

```
http://127.0.0.1:5000
```

5.2 Authentication in the backend

The is an authentication system in the backend.

There are user accounts defined with *username*, *password* and *token*

To access to backend enpoints, you need the *token* associated to your account.

5.2.1 Get the token

Send POST method to *http://127.0.0.1:5000/login* with fields:

- *username*: xxx
- *password*: xxx

Example:

```
curl -H "Content-Type: application/json" -X POST -d '{"username":"admin","password":"admin"}' http:/
```

It will get for you the token.

Example of answer:

```
{
    "token": "1442583814636-bed32565-2ff7-4023-87fb-34a3ac93d34c"
}
```

5.2.2 Generate new token (so revoke old)

If you want to generate a new token (mean revoke old token), add this field to the request made when you get the token:

- *action*: *generate*

Example:

```
curl -H "Content-Type: application/json" -X POST -d '{"username":"admin","password":"admin","action"
```

5.2.3 How to use the token

For all method you request to endpoints, you need to pass the token. Do pass this token, you can use *basic auth*. Pass token as username and set password empty.

5.3 GET method (get)

5.3.1 All items

The endpoint to get all items of a resource is:

```
http://127.0.0.1:5000/service
```

The items will be in response in section *_items*.

All items + filtering

We can filter items to get with these syntax:

```
http://127.0.0.1:5000/service?where={"service_description": "ping"}
```

All items + sorting

We can sorting items to get with these syntax:

```
http://127.0.0.1:5000/service?sort=service_description
```

If you want to sort by descending:

```
http://127.0.0.1:5000/service?sort=-service_description
```

All items + embedded

In this example, service resource has data relation with host resource with field *host_name*. If you get items, you will have for this field an _id like *55d113976376e9835e1b2feb*

It's possible to have all fields of this host in same time with:

```
http://127.0.0.1:5000/service?embedded={"host_name":1}
```

All items + projection

Projection is used to get only some fields of each items. For example, to get only *service_description* of services:

```
http://127.0.0.1:5000/service?projection={"service_description":1}
```

Pagination

The pagination is by default configured to 25 per request/page. It's possible to increase it to the limit of 50 with:

```
http://127.0.0.1:5000/service?max_results=50
```

In case of have many pages, in the items got, you have section:

```
_links: {
    self: {
        href: "service",
        title: "service"
    },
    last: {
        href: "service?page=13",
        title: "last page"
    },
    parent: {
        href: "/",
        title: "home"
    },
    next: {
        href: "service?page=2",
        title: "next page"
    }
},
```

So if you have _links/next, there is a next page.

Meta information

In the answer, you have a meta section:

```
_meta: {
    max_results: 25,
    total: 309,
    page: 1
}
```

5.3.2 One item

To get only one item, we query with the *_id* in endpoint, like:

```
http://127.0.0.1:5000/service/55d113976376e9835e1b3fee
```

It's possible in this case to use:

- *projection*
- *embedded*

5.4 POST method (add)

This method is used to *create new item*. It's required to use *POST* method for HTTP

You need to point to the endpoint of the resource like:

```
http://127.0.0.1:5000/service
```

and send a JSON of data like:

```
{"service_description":"ping","notification_interval":60}
```

If you want to add a relation with another resource, you must add the id of the resource, like:

```
{"service_description":"ping","notification_interval":60,"host_name":"55d113976376e9835e1b2feb"}
```

You will receive a response with the new *_id* and the *_etag* like:

```
{"_updated": "Tue, 25 Aug 2015 14:10:02 GMT", "_links": {"self": {"href": "service/55dc773a6376e90ac
```

5.5 PATCH method (update)

This method is used to *update fields* of an item. It's required to use *PATCH* method for HTTP

You need to point to the item endpoint of the resource like:

```
http://127.0.0.1:5000/service/55dc773a6376e90ac95f836f
```

You need to add in headers the *_etag* you have got when add or when you get data of this item:

```
"If-Match: 3c996dc10cb86173fa79f807e0d84e88c2f3a28f"
```

and send a JSON of data like:

```
{"service_description":"pong"}
```

5.6 DELETE method (delete)

It's required to use *DELETE* method for HTTP

5.6.1 All items

The endpoint to delete all items of a resource is:

```
http://127.0.0.1:5000/service
```

5.6.2 One item

The endpoint to delete an item of a resource is:

```
http://127.0.0.1:5000/service/55dc773a6376e90ac95f836f
```

5.7 More info about API

When run the Alignak Backend, it exist an endpoint with API documentation:

```
http://127.0.0.1:5000/docs
```

5.8 List of resources

List of resources and information.

5.8.1 Configuration part

List of resources used for configuration:

command

Parameter	Type	Required	Default	Data relation
definition_order	integer		100	
reactionner_tag	string		None	
module_type	string		fork	
command_name	**string**	**True**		
use	objectid			*command*
name	string			
register	boolean		True	
command_line	**string**	**True**		
poller_tag	string		None	
timeout	integer		-1	
enable_environment_macros	boolean		False	

contact

Parameter	Type	Required	Default	Data relation
host_notification_commands	list of objectid			*command*
definition_order	integer		100	
address1	string			
service_notification_options	list		[]	
address3	string			
address4	string			
address5	string			
address6	string			
contactgroups	list of objectid			*contactgroup*
is_admin	boolean		False	
service_notifications_enabled	boolean		True	
can_submit_commands	boolean		False	
contact_name	**string**	**True**		
service_notification_commands	list of objectid			*command*
pager	string			

Continued on next page

Table 5.1 – continued from previous page

Parameter	Type	Required	Default	Data relation
notificationways	list		[]	
note	string			
use	objectid			*contact*
password	string		NOPASSWORDSET	
host_notification_period	objectid			*timeperiod*
name	string			
expert	boolean		False	
host_notifications_enabled	boolean		True	
retain_nonstatus_information	boolean		True	
register	boolean		True	
service_notification_period	objectid			*timeperiod*
min_business_impact	integer		0	
retain_status_information	boolean		True	
address2	string			
alias	string			
email	string			
host_notification_options	list		[]	

contactgroup

Parameter	Type	Required	Default	Data relation
contactgroup_name	**string**	**True**		
alias	string			
contactgroup_members	list of objectid			*contactgroup*
members	list of objectid			*contact*

escalation

Parameter	Type	Required	Default	Data relation
definition_order	integer		100	
escalation_name	string			
contact_groups	list of objectid			*contactgroup*
escalation_period	string			
last_notification	integer		0	
imported_from	string			
use	objectid			*escalation*
name	string			
notification_interval	integer		-1	
contacts	list of objectid			*contact*
last_notification_time	integer		0	
escalation_options	list		['d', 'u', 'r', 'w', 'c']	
register	boolean		True	
first_notification_time	integer		0	
first_notification	integer		0	

host

Parameter	Type	Required	Default	Data relation
check_command	objectid			*command*
active_checks_enabled	boolean		True	
snapshot_interval	integer		5	
icon_image_alt	string			
business_impact_modulations	list		[]	
flap_detection_options	list		['o', 'd', 'u']	
service_includes	list		[]	
hostgroups	list of objectid			*hostgroup*
checkmodulations	list		[]	
action_url	string			
notes_url	string			
escalations	list of objectid			*escalation*
low_flap_threshold	integer		50	
process_perf_data	boolean		True	
business_rule_downtime_as_ack	boolean		False	
trigger_name	string			
service_overrides	list		[]	
statusmap_image	string			
check_period	objectid			*timeperiod*
maintenance_period	string			
use	objectid			*host*
poller_tag	string		None	
display_name	string			
notification_interval	integer		60	
contacts	list of objectid			*contact*
notification_period	objectid			*timeperiod*
trigger	string			
failure_prediction_enabled	boolean		False	
retry_interval	integer		0	
resultmodulations	list		[]	
retain_status_information	boolean		True	
icon_image	string			
stalking_options	list		[]	
event_handler_enabled	boolean		False	
3d_coords	string			
parents	list of objectid			*host*
snapshot_criteria	list		['d', 'u']	
initial_state	string		u	
first_notification_delay	integer		0	
flap_detection_enabled	boolean		True	
notification_options	list		['d', 'u', 'r', 'f']	
passive_checks_enabled	boolean		True	
labels	list		[]	
icon_set	string			
definition_order	integer		100	
snapshot_period	string			
macromodulations	list		[]	
retain_nonstatus_information	boolean		True	
notifications_enabled	boolean		True	
event_handler	string			

Continued on next page

Table 5.2 – continued from previous page

Parameter	Type	Required	Default	Data relation
contact_groups	list of objectid			*contactgroup*
vrml_image	string			
snapshot_command	string			
freshness_threshold	integer		0	
check_command_args	string			
snapshot_enabled	boolean		False	
address	string			
realm	string		None	
business_rule_smart_notifications	boolean		False	
service_excludes	list		[]	
imported_from	string			
business_rule_host_notification_options	list		[]	
custom_views	list		[]	
time_to_orphanage	integer		300	
trigger_broker_raise_enabled	boolean		False	
name	string			
notes	string			
register	boolean		True	
reactionner_tag	string		None	
alias	string			
host_name	**string**	**True**		
check_interval	integer		5	
trending_policies	list		[]	
business_impact	integer		2	
max_check_attempts	integer		1	
business_rule_output_template	string			
2d_coords	string			
business_rule_service_notification_options	list		[]	
check_freshness	boolean		False	

hostdependency

Parameter	Type	Required	Default	Data relation
inherits_parent	boolean		False	
notification_failure_criteria	list		['n']	
definition_order	integer		100	
dependent_host_name	string			
dependent_hostgroup_name	string			
imported_from	string			
use	objectid			*hostdependency*
name	string			
dependency_period	string			
execution_failure_criteria	list		['n']	
register	boolean		True	
hostgroup_name	string		unknown	
host_name	string			

hostescalation

Parameter	Type	Required	Default	Data relation
use	objectid			*hostescalation*
name	string			
definition_order	integer		100	
contacts	objectid			*contact*
last_notification_time	integer			
escalation_options	list		['d', 'u', 'r', 'w', 'c']	
register	boolean		True	
contact_groups	objectid			*contactgroup*
notification_interval	integer		30	
hostgroup_name	string			
escalation_period	string			
host_name	string			
first_notification_time	integer			
first_notification	integer			
last_notification	integer			
imported_from	string		unknown	

hostextinfo

Parameter	Type	Required	Default	Data relation
statusmap_image	string			
definition_order	integer		100	
icon_image_alt	string			
vrml_image	string			
notes_url	string			
imported_from	string			
use	objectid			*hostextinfo*
name	string			
notes	string			
register	boolean		True	
icon_image	string			
3d_coords	string			
host_name	**string**	**True**		
2d_coords	string			

hostgroup

Parameter	Type	Required	Default	Data relation
hostgroup_name	**string**	**True**		
action_url	string			
notes_url	string			
members	list of objectid			*host*
alias	string			
realm	string		None	
hostgroup_members	list of objectid			*hostgroup*
notes	string			

service

Parameter	Type	Required	Default	Data relation
check_command	objectid			*command*
active_checks_enabled	boolean		True	
icon_image_alt	string			
business_impact_modulations	list		[]	
flap_detection_options	list		['o', 'w', 'c', 'u']	
labels	list		[]	
snapshot_command	string			
duplicate_foreach	string			
action_url	string			
is_volatile	boolean		False	
escalations	list of objectid			*escalation*
low_flap_threshold	integer		-1	
process_perf_data	boolean		True	
business_rule_downtime_as_ack	boolean		False	
snapshot_interval	integer		5	
check_period	objectid			*timeperiod*
maintenance_period	objectid			*timeperiod*
use	objectid			*service*
poller_tag	string		None	
display_name	string			
notification_interval	integer		60	
contacts	list of objectid			*contact*
notification_period	objectid			*timeperiod*
trigger	string			
failure_prediction_enabled	boolean		False	
snapshot_period	string			
retry_interval	integer		0	
resultmodulations	list		[]	
retain_status_information	boolean		True	
icon_image	string			
stalking_options	list		[]	
event_handler_enabled	boolean		False	
host_dependency_enabled	boolean		True	
aggregation	string			
snapshot_criteria	list		['w', 'c', 'u']	
initial_state	string		o	
first_notification_delay	integer		0	
flap_detection_enabled	boolean		True	
notification_options	list		['w', 'u', 'c', 'r', 'f', 's']	
passive_checks_enabled	boolean		True	
default_value	string		[]	
icon_set	string			
definition_order	integer		100	
high_flap_threshold	integer		-1	
macromodulations	list		[]	
retain_nonstatus_information	boolean		True	
notifications_enabled	boolean		True	
event_handler	string			

Continued on next page

Table 5.3 – continued from previous page

Parameter	Type	Required	Default	Data relation
contact_groups	list of objectid			*contactgroup*
business_rule_smart_notifications	boolean		False	
business_impact	integer		2	
freshness_threshold	integer		0	
check_command_args	string			
snapshot_enabled	boolean		False	
service_dependencies	list of objectid			*service*
parallelize_check	boolean		True	
service_description	string			
trigger_name	string			
checkmodulations	list		[]	
imported_from	string			
business_rule_host_notification_options	list		[]	
time_to_orphanage	integer		300	
trigger_broker_raise_enabled	boolean		False	
name	string			
notes	string			
register	boolean		True	
reactionner_tag	string		None	
hostgroup_name	string			
servicegroups	list of objectid			*servicegroup*
host_name	objectid			*host*
check_interval	integer		5	
trending_policies	list		[]	
merge_host_contacts	boolean		False	
notes_url	string			
max_check_attempts	integer		1	
business_rule_output_template	string			
custom_views	list		[]	
business_rule_service_notification_options	list		[]	
check_freshness	boolean		False	

servicedependency

Parameter	Type	Required	Default	Data relation
inherits_parent	boolean		False	
notification_failure_criteria	list		['n']	
definition_order	integer		100	
explode_hostgroup	boolean		False	
dependent_host_name	string			
dependent_hostgroup_name	string			
imported_from	string		unknown	
use	objectid			*servicedependency*
name	string			
dependency_period	string			
execution_failure_criteria	list		['n']	
register	boolean		True	
hostgroup_name	string		unknown	
host_name	string			
dependent_service_description	string			

serviceescalation

Parameter	Type	Required	Default	Data relation
use	objectid			*serviceescalation*
name	string			
definition_order	integer		100	
contacts	objectid			*contact*
last_notification_time	integer			
escalation_options	list		['d', 'u', 'r', 'w', 'c']	
register	boolean		True	
contact_groups	objectid			*contactgroup*
notification_interval	integer		30	
hostgroup_name	string			
escalation_period	string			
host_name	string			
first_notification_time	integer			
service_description	string			
first_notification	integer			
last_notification	integer			
imported_from	string		unknown	

serviceextinfo

Parameter	Type	Required	Default	Data relation
definition_order	integer		100	
icon_image_alt	string			
notes_url	string			
service_description	string			
imported_from	string			
use	objectid			*serviceextinfo*
name	string			
notes	string			
register	boolean		True	
icon_image	string			
host_name	**string**	**True**		

servicegroup

Parameter	Type	Required	Default	Data relation
servicegroup_name	**string**	**True**		
notes	string			
alias	string			
action_url	string			
notes_url	string			
members	objectid			*contact*
servicegroup_members	list of objectid			*servicegroup*

timeperiod

Parameter	Type	Required	Default	Data relation
dateranges	list		[]	
alias	string			
use	objectid			*timeperiod*
name	string			
definition_order	integer		100	
exclude	list		[]	
register	boolean		True	
is_active	boolean		False	
imported_from	string			
timeperiod_name	**string**	**True**		

trigger

Parameter	Type	Required	Default	Data relation
use	objectid			*trigger*
name	string			
definition_order	integer		100	
register	boolean		True	
code_src	string			
trigger_name	string			
imported_from	string		unknown	

5.8.2 Live state part

List of live states (last check date, current state...):

livehost

Parameter	Type	Required	Default	Data relation
state_type	string		HARD	
long_output	string		None	
state	string		UP	
host_name	**objectid**	**True**		**:ref:‘host <resource-host>‘**
output	string		None	
acknowledged	boolean		False	
perf_data	string		None	
last_check	integer		None	

liveservice

Parameter	Type	Required	Default	Data relation
state_type	string		HARD	
long_output	string		None	
state	string		OK	
output	string		None	
service_description	**objectid**	**True**		**:ref:'service <resource-service>'**
acknowledged	boolean		False	
perf_data	string		None	
last_check	integer		None	

5.8.3 Retention part

List of retentions resources:

retentionhost

Parameter	Type	Required	Default	Data relation
host	string			

retentionservice

Parameter	Type	Required	Default	Data relation
service	list			

5.8.4 Log part

List of log resources:

loghost

Parameter	Type	Required	Default	Data relation
state_type	string		HARD	
long_output	string		None	
state	string		UP	
host_name	**objectid**	**True**		**:ref:'host <resource-host>'**
output	string		None	
acknowledged	boolean		False	
perf_data	string		None	
last_check	integer		None	

logservice

Parameter	Type	Required	Default	Data relation
state_type	string		HARD	
long_output	string		None	
state	string		UP	
output	string		None	
service_description	**objectid**	**True**		**:ref:'service <resource-service>'**
acknowledged	boolean		False	
perf_data	string		None	
last_check	integer		None	

CHAPTER 6

Tutorial of solution

6.1 Description of the solution

This is a tutorial for install and use the complete solution:

- pure monitoring tool: Alignak
- backend alignak: this tool
- modules in Alignak to have communication between alignak and the backend

A schema of the solution is available in page architecture.

6.2 Documentation links

So there are the steps to install all:

- install Alignak
- install the alignak-backend
- install the mod-alignakbackend (documentation not yet writen, so link to repository)

CHAPTER 7

Tools

With the backend, some tools has been added and can help you in some cases.

This is the list:

- cfg_to_backend.py: script to open cfg files (alignak, nagios, shinken) and send config to the backend

CHAPTER 8

Indices and tables

- genindex
- modindex
- search