
Alignak-App Documentation

Release 1.3

Matthieu Estrada

Mar 05, 2018

Contents

1	Introduction	3
1.1	About Alignak	3
1.2	About Alignak-App	3
1.2.1	Features	3
1.2.2	Release cycle	4
1.3	Thanks	4
2	Alignak-app Installers	5
2.1	Requirements	5
2.1.1	Alignak Suite	5
2.1.2	Python 3	5
2.2	Installation (Windows)	5
2.3	Installation (Linux)	6
2.4	Installation (From Sources)	6
2.4.1	Clone and install	6
2.4.2	External Libraries	6
2.4.3	Under Windows	6
3	Environment Variables	7
3.1	On Linux	7
3.2	On Windows	7
4	Configuration	9
4.1	Settings file	9
4.2	Connection to Backend	12
5	Launch Alignak-App	13
5.1	On Linux	13
5.1.1	First launch	13
5.1.2	Launch daemon	13
5.2	On Windows	14
5.2.1	With Installer	14
5.2.2	From command line	14
6	Solve Problems	15
7	Alignak-app Interface	17

7.1	Login Window	17
7.2	Start	17
7.3	App Dock (Right part)	18
7.3.1	Alignak part	18
7.3.2	Livestate part	18
7.3.3	Events part	18
7.3.4	Spy part	18
7.4	App Panel (Left part)	18
7.4.1	Host Synthesis	19
7.4.2	Problems	19
7.5	WebUI Integration	19
8	Windows Setup	21
8.1	Requirements	21
8.1.1	Python and requirements	21
8.1.2	Pyinstaller	21
8.1.3	Inno Setup	22
8.2	Create Setup	22
9	Alignak-app Architecture	23
10	Alignak-App	25
10.1	Main Class	25
10.1.1	App	25
10.2	Backend	26
10.2.1	Backend	26
10.2.2	Data Manager	28
10.3	Items	29
10.3.1	Item	30
10.3.2	Daemon	31
10.3.3	Event	32
10.3.4	History	33
10.3.5	Host	34
10.3.6	Live Synthesis	34
10.3.7	Period	35
10.3.8	Realm	36
10.3.9	Service	37
10.3.10	User	38
10.4	Utils	38
10.4.1	Settings	39
10.4.2	Installer	40
10.4.3	System	40
10.4.4	Logs	41
10.4.5	Time	41
10.5	Locales	42
10.5.1	Locales	42
10.6	QObjects	42
10.6.1	App Main	42
10.6.2	Common	42
10.6.3	Dock (Rigt part)	46
10.6.4	Login	54
10.6.5	Panel (Left part)	54
10.6.6	Systray	60
10.7	QThreads	61

10.7.1	Backend Thread	61
10.7.2	Thread Manager	61
11	Indices and tables	63
	Python Module Index	65

Documentation content:

1.1 About Alignak

Alignak is an open source monitoring framework written in Python under the terms of the [GNU Affero General Public License](#) . It is a fork of the Shinken project.

More information about Alignak is available in the [Alignak documentation](#).

1.2 About Alignak-App

The Alignak-App is an open source Applet in system tray, written in Python. It is available under the terms of the [GNU Affero General Public License](#) .

This application is intended to be used within the Alignak project, it connects to the Alignak backend and allows you to be notified directly on your workstation. You will be able to view the states of the hosts and the services that you have defined in your installation.

1.2.1 Features

Alignak-App has many features:

- Multi-platfom: Linux / Windows
- Receive Alignak notifications inside application
- Spy host: choose to monitor multiple hosts and receive their states
- Search Hosts and Services, acknowledge problems, schedule downtimes and more
- Many configurations
- Links to your [Alignak WebUI](#) from the menus

1.2.2 Release cycle

“App” has no strict schedule for releasing.

Other features will come in the next versions and you can propose new features through [project issues](#). Each feature is discussed in a separate issue.

1.3 Thanks

Thanks to [ddurieux](#), [mohierf](#) and [speal](#) for their help and contribution.

2.1 Requirements

2.1.1 Alignak Suite

You must have Alignak Suite installed before using this app :

- [alignak](#)
- [alignak-backend](#)

Alignak-app need at least a fonctionnal and available **alignak-backend** to connect with it !

2.1.2 Python 3

You must have **Python 3** installed on your system. Only the Windows installer does not need to install Python 3.

2.2 Installation (Windows)

An installer is available in repository of Alignak-app on [releases page](#). It is **recommended** that you use this installer to run Alignak-app on Windows.

You have nothing else to install if you are using the installer !

IMPORTANT: To keep it free, installer is not signed, so **Windows Defender SmartScreen** will warn you about that. Just click on “More Informations” and on “Execute anyway” to run installer.

IMPORTANT: This installer is currently only compatible with **x64** architecture !

You can also buid your own Windows installer (on **develop** branch for example), see [Windows Setup](#) for more information.

However, you can install the application in the same way as under Linux, but this will require you to keep a window open for Python.

2.3 Installation (Linux)

You can install Alignak-app like other python libraries, with `pip3`:

```
pip3 install alignak_app
```

The required Python modules are automatically installed, if not present on your system.

If you install App as a root user (with `sudo`), you have to run `--install` command before launch Alignak-App ! (see [Launch Alignak-App](#))

2.4 Installation (From Sources)

2.4.1 Clone and install

To install from source, clone repos and install with `pip`

```
git clone https://github.com/Alignak-monitoring-contrib/alignak-app
cd alignak-app
# If you're under Windows, use "pip" instead "pip3"
pip3 install .
```

2.4.2 External Libraries

You need to install Python modules that are listed in `requirements.txt` file with `pip`:

```
pyqt5<5.10
configparser
future
alignak_backend_client
python-gettext
```

2.4.3 Under Windows

If you've installed Alignak-app with `pip` under Windows, you must link install directory with your Program Files folder. Otherwise, App won't start.

Open a command Windows console as admin and type the following line:

```
mklink /J "%ProgramFiles%\Alignak-app" c:\Users\<USERNAME>
↪ \AppData\Roaming\Python\alignak_app
```

Replace `<USERNAME>` by your username.

Be sure also that you've install Python3 on your device.

Environment Variables

During installation, Alignak-app creates some folders that contains all the files application need to run. Then she will keep them in **environment variables**.

The application will normally automatically detect the folders used by the application and use environment variables for this purpose.

- ALIGNAKAPP_USR_DIR: this folder is used by application for read the `settings.cfg` file . By default, Alignak-app will log also in this directory.
- ALIGNAKAPP_APP_DIR: this folder is used by application for read `images.ini` file, `style.css` and get images. This variable is normally equal to the ROOT folder described above. **Be careful if you define this variable yourself!**
- ALIGNAKAPP_LOG_DIR: this folder is used to create the log file of Alignak-app.

During Windows installation, if you used **Installer**, these variables are automatically filled.

3.1 On Linux

Alignak-app will try to create ALIGNAKAPP_USR_DIR for later configurations. By default this folder should be:

```
/ $HOME/.local/alignak_app/
```

The ALIGNAKAPP_APP_DIR should be under one of the following folders and is create by pip during installation:

```
/ $HOME/.local/alignak_app/  
/usr/local/alignak_app/
```

3.2 On Windows

On Windows, use **Installer** is recommended (see [Alignak-app Installers](#)) to respect Windows conventions. ALIGNAKAPP_USR_DIR and ALIGNAKAPP_APP_DIR variables are by default set to:

```
C:\Program Files\Alignak-app\
```

If you install by pip under Windows, you may need to **define these variables yourself** ! And ALIGNAKAPP_USR_DIR and ALIGNAKAPP_APP_DIR variables should be by default set to:

```
C:\Users\user\AppData\Roaming\Python\alignak_app\
```

CHAPTER 4

Configuration

Before running application, **you must configure it**.

You will find a `settings.cfg` file located in the `ALIGNAKAPP_USR_DIR` folder cited above.

This file contains Sections who are introduced by a `[section_name]` header. Then, it contains `name = value` entries. All parameters are also explained in file. For the boolean parameters, you can use the following values: `on/off`, `true/false` or `1/0`.

The most significant Section is **[Alignak]**. You'll need set your backend url and ports. Otherwise, Alignak-app proposes to define your server Alignak and its port via the window login.

To Know: Without connection on Backend of Alignak, App won't start ! “

4.1 Settings file

Here is the full `settings.cfg` file. This file contains comments for each setting.

```
[Alignak]
; Backend authentication
; -----
; -----
; Credentials for your backend.
; 1/ Recommended: leave "username" and "password" empty. App will display a
; login Window.
; 2/ Recommended: set your token in "username" option and leave "password"
; empty.
; 3/ Not recommended: set your "username" and your "password".
; Read the alignak-app docs for more details on token and authentication,
; Or visit http://docs.alignak.net/projects/alignak-backend/en/latest/api.html#get-the-authentication-token
; The default values are empty.
username =
password =
```

(continues on next page)

(continued from previous page)

```

; Alignak Url
; -----
↪-----
; Url of alignak server.
; Default is http://127.0.0.1
url = http://127.0.0.1

; Backend url
; -----
↪-----
; The url of your Backend.
; Default is %(url)s:5000
backend = %(url)s:5000

; Webui url
; -----
↪-----
; The url of your Webui
; Default is %(url)s:80
webui = %(url)s:80

; Backend processes
; -----
↪-----
; The number of processes for connection to backend
; -!- Windows users must leave this parameter at 1 -!-
; Default is 1
processes = 1

[Alignak-app]
; Locale
; -----
↪-----
; Define language of App. Currently only "fr_FR", "en_US" are available.
; If empty, "en_US" will be used as default.
; Default is "en_US"
locale = en_US

; Display at start
; -----
↪-----
; Defines whether the application is displayed or not, in full screen or ↪
↪minimized
; Default is "min". Available options are "no", "max", "min".
display = min

; Requests interval
; -----
↪-----
; Define interval of App requests. App make regular requests
; Default is 30. Increase this settings can improve speed of App
; -!- Set this settings less than 5 is not recommended -!-
requests_interval = 30

; Notifications elapsed
; -----
↪-----

```

(continues on next page)

(continued from previous page)

```

; Sets in minutes the time elapsed since notifications were triggered.
; Default is 30
notification_elapsed = 30

; Notifications duration
; -----
↪-----
; Define duration of temporary notifications
; Default is 30
notification_duration = 30

; Spy checks interval
; -----
↪-----
; Define checks interval in seconds for spied hosts. One host check by ↪
↪interval.
; Default is 60
spy_interval = 60

; Update dock "Status"
; -----
↪-----
; Define in seconds the refresh of dock alignak status
; Default is 30
update_status = 30

; Update dock "Buttons"
; -----
↪-----
; Define in seconds the refresh time of dock buttons
; Default is 20
update_buttons = 20

; Update dock "Livestate"
; -----
↪-----
; Define in seconds the refresh time of Dock livestate
; Default is 30
update_livestate = 30

; Update panel Dashboard
; -----
↪-----
; Define in seconds the refresh time of panel Dashboard
; Default is 30
update_dashboard = 30

; Update panel Host
; -----
↪-----
; Define in seconds the refresh time of panel Host view
; Default is 20
update_host = 20

; Update panel Service
; -----
↪-----

```

(continues on next page)

(continued from previous page)

```

; Define in seconds the refresh time of panel Service view
; Default is 20
update_service = 20

[Log]
; Log filename
; -----
↪-----
; Define name of file where logs will be stored.
filename = alignakapp

; Log location
; -----
↪-----
; Set this value to define another directory. Path must be absolute !
; Default is empty. Default values are:
; Linux:   ~/.local/alignak_app
; Windows: C:\ProgramData\Alignak-app
location =

; Application Debug Mode
; -----
↪-----
; Activate Debug mode for Alignak-App to get more logs
; Set to 'yes' or 'true' to activate "debug" mode
; Set to 'no' or 'false' to deactivate "debug" mode
; Default is "no"
debug = no

```

4.2 Connection to Backend

Alignak-app have a login form by default, who let you to connect with the username and password define in backend.

You can also set connection information in the `settings.cfg` file:

- **Recommended:** leave empty “username” and “password”. Alignak-app will display a login Window.
- **Recommended:** set your token in “username” field and leave “password” empty (See below).
- **Not recommended:** set your “username” and your “password”. This method is less secure.

To obtain a token, open a python terminal and type the following commands:

```

import requests
backend_url = 'http://alignak.com:5000'
r = requests.post(
    backend_url + '/login',
    data={
        'username': 'admin',
        'password': 'admin'
    }
)
print(r.text)

```

You can also visit: [Alignak-backend : Get Token](#)

Launch Alignak-App

5.1 On Linux

5.1.1 First launch

Once you have install Alignak-app, just run *alignak-app.py* file. This file should be located in:

```
# If install by pip as user
$HOME/.local/alignak_app/bin/alignak-app.py
# If install by pip as root
/usr/local/alignak_app/bin/alignak-app.py
```

To launch application use `--start`:

```
/usr/local/alignak_app/bin/alignak-app.py --start
```

If you have set environment variables for application, they will be added to the generated daemon script (see *Environment Variables*).

To generate an Alignak-app daemon file, use `--install`:

```
/usr/local/alignak_app/bin/alignak-app.py --install
```

5.1.2 Launch daemon

Once installed, just run:

```
alignak-app start
```

Here is the available commands:

```
alignak-app {start|stop|status|restart}
```

With no parameters, this will display help and your environment variables configuration.

Note: To generate a new daemon file with other environment, just run `alignak-app.py --start` again with your new environment variables.

5.2 On Windows

5.2.1 With Installer

Just launch `Alignak-app vX.x` shortcut on your desktop or run the `alignak-app.exe` located in `C:\Program Files\Alignak-app\`.

5.2.2 From command line

If you install from sources or with `pip`, you just have to launch the `alignak-app.py` in `bin` folder:

```
python c:\Users\user\AppData\Roaming\Python\alignak_app\bin\alignak-app.py --start
```

Obviously, it requires to keep an open command prompt on Windows.

Note: There is no daemon available on Windows, please use provided setup.

Solve Problems

If you're having trouble getting started and running the app, here are some things to check:

1. Be sure you have correctly:
 - Install application: see *Alignak-app Installers*.
 - Set your configuration file: see *Environment Variables*.
2. The application generates a log file in folder defined in `ALIGNAKAPP_LOG_DIR` (see *Environment Variables*). Search for `ERROR` messages in this file.
3. For more help you can set `debug` to **yes** to have more informations.
4. That your backend is reachable from your computer.

Alignak-app Interface

7.1 Login Window

If you have not set your configuration file, Alignak-app will display a login window by default. In this window, you'll be able to:

- Configure your Alignak server (by clicking on **server** icon).
- Type your **username** and **password** to login.

7.2 Start

Once you're connected, Alignak-app will start and show you a progress bar. Time to start will depend on your installation (a backend with a lot of host and services will take more time).

When Alignak-app has started, you'll have:

- The **Main Window** of App, separated into two parts:
 - The “App Dock” (Right part)
 - The “App Panel” (Left part)
- A **Tray Icon** in your task bar. This icon gives you access to a menu with:
 - Alignak-app menu, to view the app if it has been reduced
 - About menu, to show informations of Alignak-app
 - Reload menu, to reload your configuration.

Other menu will come in future versions.

7.3 App Dock (Right part)

When App is launched, the right part (called “dock”) will contains different buttons and some container who will receive different notifications.

7.3.1 Alignak part

This part of dock contains informations about your connection to backend and the states of Alignak daemons. If you’ve some connection problems or if backend is restart, you’ll see icons change.

You can also see each daemon state by clicking on button with Alignak icon.

7.3.2 Livestate part

In the livestate part, you’ll have many buttons who let you see: hosts, user configuration, problems view (available soon) and a button to reach your WebUI if available.

You’ll also see a resume of number of problems for hosts and services monitored in your backend.

7.3.3 Events part

This is one of the most interesting features of the application. You’ll receive many informations in this part like alignak notifications, acknowledge or modifications you do in App.

Events:

- If you want to remove an event, simply double click on it.
- Full events text can be see in tooltip when you keep mouse over.
- Some events are temporary and remove themselves.
- Some events can be drag and drop (see below)

Drag & Drop:

- Events from a host or a service can be drag to in “Spy Hosts”. Then App will send you regular notifications about that host.
- Events can also be move to host synthesis view (Panel), to display host instantly.

7.3.4 Spy part

This is where you can find the list of hosts that you spy on.

- You can spy on as many hosts as you want.
- To stop spying on a host, just double click on it, host will be removed from list.

7.4 App Panel (Left part)

The left part of application (called “Panel”), will display a **Hosts Synthesis View**. This is where you can see your monitored hosts and services.

7.4.1 Host Synthesis

Dashboard:

At the top of this window, you will find a dashboard with the number of items, for each state (OK, CRITICAL, DOWN...).

Host View:

Start typing the name of a host in the search bar and App will propose to you different corresponding names. When you select a host, its information and the list of its services will be displayed.

Services:

When a host is displayed, you will have the list of services, classified by aggregation. Click on one of them to display them.

If you click on a service, a summary of its status will be displayed on the right.

Actions: You will then be able to perform various actions such as seeing the details of each of its services, acknowledging a problem or planning a downtime for an item (service or host).

You will also have access to the host's history. It may take a while to be available.

Another tab called "Problems" is used to display the problems listed by the backend (like a DOWN host or a CRITICAL service).

7.4.2 Problems

Other tab will display problems found by Alignak-app in backend.

This will bring together down hosts, critical services or any other worries, such as an unreachable host.

You'll be able to acknowledge problems or trigger downtimes.

7.5 WebUI Integration

App also have lot of buttons which bring you to WebUI. You've to set WebUI url in configuration file to make this buttons available.

App will be aware also on events trigger in WebUI in host history.

Windows Setup

For releases, a setup is generated for Windows and is available for [download](#). For the version under development, you have to do it yourself.

8.1 Requirements

Obviously, you must clone the Alignak-app repository, on the develop branch before. Like that you'll have the last fixes. Normally, we try to have a branch `develop` as stable as possible.

8.1.1 Python and requirements

You have to install [Python 3.5](#) in any case.

Then the requirements of Alignak-app . Otherwise, *pyinstaller* will not have the required *.dll* for compilation. All is available on **Pypi**:

```
# In repository folder
pip install -r requirements.txt --user
```

Once done, you'll normally have your python modules installed in:

```
"%APPDATA%\Python\Python35\"
```

Then install *pyinstaller*.

8.1.2 Pyinstaller

The module *pyinstaller* is also available on **Pypi**. So just run the following command:

```
pip install pyinstaller --user
```

Normally, *pyinstaller.exe* command will be available under:

```
"%APPDATA%\Python\Python35\Scripts\"
```

And will be added to your *PATH* variable. If it is not the case, you can add this folder to your *PATH* without problem, you will definitely need it for other python libraries.

8.1.3 Inno Setup

Inno Setup is a free installer for Windows. It is very powerful and allows to create and customize installers quite easily.

To install Inno Setup, just download the last **unicode** version on [Official download](#) page.

Be sure to choose unicode version !

And simply run it with values as default.

8.2 Create Setup

To create your own setup, you'll find scripts in `bin\win` folder of repository. There is also images, a redistribuable for Windows (needed for old versions of Windows) and 2 script files.

The first one is `pyinstaller_app.bat`.

Before running it, check the `--paths` arguments. Normally, you'll have just to change the repository folder (line 13).

Be sure to put absolute paths !

The others are normally the same on your device. If `pyinstaller` does not find the `PyQt` dll, check these paths.

Then run the `.bat`. This script will generate an `alignak-app.exe` in **dist** folder. Don't move it !

After, simply open the Inno Setup file `alignak-app-win-setup.iss`. You can change *ShortVersion* if you want, but normally these digits are same as current develop. And then, compile the file with `CTRL+F9` or from menu `Build->Compile`.

This will generate an installer inside the `dist\setup` folder.

Your installer is ready !

You can then uninstall the python libraries if necessary, your Setup will no longer use them. All the libraries you need are compressed into the executable.

Alignak-app Architecture

This Python module is a desktop application, with a system tray icon, for Alignak solution.

Application notify you when you have changes in your monitoring. You can trigger actions inside application, see status of monitored items, problems to solved and many other things.

Alignak-app have the following architecture:

- The *BackendClient* manage requests with Alignak backend.
- The *ThreadManager* will launch *BackendQThread(s)* to trigger requests in *BackendClient*.
- The *DataManager* will store data provided by *BackendClient* in *Items*.
- The *QObjects* package display/update the data stored in *DataManager*.
- The *Utils* package contains settings, logs, installation,...
- The *Locales* package contains translations.

Alignak-app will use a system of *Installer* who use **environment variables** to run:

- ALIGNAKAPP_USR_DIR: contains settings of user (**write rights**)
- ALIGNAKAPP_LOG_DIR: contains log files of App (**write rights**)
- ALIGNAKAPP_APP_DIR: contains binaries of App (images, css, languages) (**read rights**)

The *Login* manage user login if needed.

10.1 Main Class

10.1.1 App

App manages the creation of all objects and QObjects for the whole application:

- Creation of *AppProgressBar* until the Data Manager is ready
- Creation of *AppQMainWindow*
- Creation of standard python objects (settings, css, localization)

class alignak_app.app.**AlignakApp**

Bases: PyQt5.QtCore.QObject

Class who build Alignak-app QObjects, initialize configurations, systray icon and Thread Manager.

static check_threads ()

Launch periodically threads

static quit_launched_threads ()

Exit the threads that were started when the application started

Parameters **launched_threads** (*list*) – list of threads that have been launched

Returns empty list if all the threads have been left or current list

Return type *list*

start (*username=None, password=None*)

Start Alignak-app

class alignak_app.app.**AppProgressBar**

Bases: PyQt5.QtWidgets.QProgressBar

AppProgressBar in busy mode with text displayed at the center.

set_text (*text*)
Set text of QProgressBar

Parameters **text** (*str*) – text of progress bar

text ()
Overload: text(self) -> str

Returns text of progress bar

Return type *str*

class alignak_app.app.AppProgressQWidget (*parent=None*)
Bases: PyQt5.QtWidgets.QWidget

Class who create a small widget for App start progression

initialize ()
Initialize the QWidget

10.2 Backend

The Backend package contains classes to manage alignak data

10.2.1 Backend

Backend manage connection and access to Alignak backend. It also fill *DataManager*.

class alignak_app.backend.backend.BackendClient
Bases: *object*

Class who collect informations with Backend-Client and returns data for Alignak-App.

get (*endpoint, params=None, projection=None, all_items=False*)
GET on alignak Backend REST API.

Parameters

- **endpoint** (*str*) – endpoint (API URL)
- **params** (*dict / None*) – dict of parameters for the app_backend API
- **projection** (*list / None*) – list of field to get, if None, get all
- **all_items** (*bool*) – make GET on all items

Returns desired request of app_backend

Return type *dict*

get_backend_status_icon ()
Return backend status icon name

Returns daemon status icon name

Return type *str*

login (*username=None, password=None, check=False*)
Connect to alignak backend

Parameters

- **username** (*str*) – name or token of user

- **password** (*str*) – password of user. If token given, this parameter is useless
- **check** (*bool*) – define if login is a check or a first login

Returns True if connected or False if not

Return type `bool`

patch (*endpoint, data, headers*)

PATCH on alignak Backend REST API

Parameters

- **endpoint** (*str*) – endpoint (API URL)
- **data** (*dict*) – properties of item to update
- **headers** (*dict*) – headers (example: Content-Type). ‘If-Match’ required

Returns dictionary containing patch response from the backend

Return type `dict`

post (*endpoint, data, headers=None*)

POST on alignak Backend REST API

Parameters

- **endpoint** (*str*) – endpoint (API URL)
- **data** (*dict*) – properties of item to create | add
- **headers** (*dict | None*) – headers (example: Content-Type)

Returns response (creation information)

Return type `dict`

query_daemons_data ()

Launch request for “alignakdaemon” endpoint

query_history_data (*hostname=None, host_id=None*)

Launch request for “history” endpoint but only for hosts in “data_manager”

Parameters

- **hostname** (*str*) – name of host we want history
- **host_id** (*str*) – id of host for history

query_hosts_data ()

Launch request for “host” endpoint

query_livesynthesis_data ()

Launch request for “livesynthesis” endpoint

query_notifications_data ()

Launch request for “history” endpoint but only for notifications of current user

query_period_data ()

Launch a request on timeperiod endpoint

query_realms_data ()

Launch a request on realm endpoint

query_services_data ()

Launch request for “service” endpoint

query_user_data()
Launch request for “user” endpoint

10.2.2 Data Manager

DataManager manage alignak data provided by *Client*.

```
class alignak_app.backend.datamanager.DataManager
    Bases: object

    Class who store alignak data

get_all_hostnames()
    Collect and return all names of all hosts

    Returns all names of all hosts

    Return type list

get_events()
    Get the last events

    Returns events formatted for App to send

    Return type list

get_host_services(host_id)
    Return services of wanted host

    Parameters host_id (str) – ‘_id’ of host

    Returns services of host

    Return type list

get_host_with_services(host_field)
    Returns the desired host and all its services

    Parameters host_field (str) – field of wanted host: host_id | name

    Returns dict with host data and its associated services

    Return type dict

get_item(item_type, key, value=None)
    Return the wanted item for item type who contain the value

    Parameters

    • item_type (str) – type of wanted item
    • key (str) – key contained in item
    • value (str) – value of the key if needed

    Returns wanted item

    Return type alignak_app.items.item.*

get_items_and_problems()
    Return total of items and problems

    Returns dict of problem and total number for each item

    Return type dict
```

get_period_name (*period*)

Return the period name or alias

Parameters **period** (*str*) – wanted period `_id`

Returns the wanted realm alias or name if available

Return type *str*

get_problems ()

Return items who are in problem: hosts and services

Returns dict of items in problem, and number for each type of item

Return type *dict*

get_realm_name (*realm*)

Return the realm name or alias

Parameters **realm** (*str*) – wanted realm `_id`

Returns the wanted realm alias or name if available

Return type *str*

get_synthesis_count ()

Get on “synthesis” endpoint and return the states of hosts and services

Returns states of hosts and services.

Return type *dict*

is_ready ()

Check if DataManager is filled and ready

Returns if ready or current status

Return type *str*

update_database (*item_type*, *items_list*)

Update an item type in database

Parameters

- **item_type** (*str*) – type of item to update
- **items_list** (*list* | *dict*) – list of items for the wanted type

update_item_data (*item_type*, *item_id*, *data*)

Update a single item in database

Parameters

- **item_type** –
- **item_id** –
- **data** –

Returns

10.3 Items

The Items package contains classes who create items who are stored in *DataManager*.

10.3.1 Item

Item is parent class for all items objects:

- *Daemon*,
- *Event*,
- *History*,
- *Host*,
- *LiveSynthesis*,
- *Period*,
- *Realm*,
- *Service*,
- *User*,

class alignak_app.items.item.Item

Bases: `object`

Class who create an item

create (*_id*, *data*, *name=None*)

Create wanted item

Parameters

- **_id** (*str*) – id of the item. Often equal to id in alignak backend
- **data** (*dict* / *list*) – data of the item
- **name** (*str*) – name of the item if available

get_tooltip ()

Return the tooltip message depending state and actions

Returns tooltip message

Return type `str`

update_data (*key*, *new_value*)

Update data of the wanted key

Parameters

- **key** (*str*) – key to update
- **new_value** – new value of the key

alignak_app.items.item.**get_host_msg_and_event_type** (*host_and_services*)

Return corresponding event icon to number of services who are in alert

Parameters **host_and_services** (*dict*) – Host() item and its Service() items

Returns event type and message

Return type `dict`

alignak_app.items.item.**get_icon_name** (*item_type*, *state*, *acknowledge*, *downtime*, *monitored*)

Return icon for a host or a service item

Parameters

- **item_type** (*str*) – type of item: host | service

- **state** (*str*) – state of item
- **acknowledge** (*bool*) – if item is acknowledged or not
- **downtime** (*bool*) – if item is downtimed
- **monitored** (*int*) – define if host is monitored or not (0 is not monitored, 1 or 2 is monitored)

Returns icon name for icon

Return type *str*

`alignak_app.items.item.get_icon_name_from_state(item_type, state)`

Return icon name from state for host or service

Parameters

- **item_type** (*str*) – type of item: host or service
- **state** (*str*) – state of item

Returns

`alignak_app.items.item.get_real_host_state_icon(services)`

Return corresponding icon to number of services who are in alert

Parameters **services** (*list*) – list of Service() items

Returns icon corresponding to state

Return type *str*

10.3.2 Daemon

Daemon manage creation of daemon item for backend alignakdaemon endpoint

class `alignak_app.items.daemon.Daemon`

Bases: `alignak_app.items.item.Item`

Class who create a daemon item

create (*_id*, *data*, *name=None*)

Create wanted item

Parameters

- **_id** (*str*) – id of the item. Often equal to id in alignak backend
- **data** (*dict* / *list*) – data of the item
- **name** (*str*) – name of the item if available

static `get_daemons_names()`

Returns all the names of daemons

Returns all the names of daemons

Return type *list*

static `get_daemons_status_icon()`

Return daemons status icon name

Returns daemons status icon name

Return type *str*

static get_request_model()
Return the request model for alignakdaemon requests
Returns request model for alignakdaemon endpoint
Return type `dict`

static get_states()
Return states of daemons or backend
Parameters **status** (`str`) – status of item
Returns the status string
Return type `str`

get_tooltip()
Return the tooltip message depending state and actions
Returns tooltip message
Return type `str`

update_data (`key`, `new_value`)
Update data of the wanted key
Parameters

- **key** (`str`) – key to update
- **new_value** – new value of the key

10.3.3 Event

Event manage creation of event item (Notifications collected from `history` endpoint)

class `alignak_app.items.event.Event`
Bases: `alignak_app.items.item.Item`
Class who create an event item

create (`_id`, `data`, `name=None`)
Create wanted item
Parameters

- **_id** (`str`) – id of the item. Often equal to id in alignak backend
- **data** (`dict` | `list`) – data of the item
- **name** (`str`) – name of the item if available

static get_request_model()
Return the request model for notification requests
Returns request model for history endpoint (only for notifications)
Return type `dict`

get_tooltip()
Return the tooltip message depending state and actions
Returns tooltip message
Return type `str`

update_data (*key*, *new_value*)
Update data of the wanted key

Parameters

- **key** (*str*) – key to update
- **new_value** – new value of the key

10.3.4 History

History manage creation of history item for backend history endpoint

class alignak_app.items.history.**History**

Bases: *alignak_app.items.item.Item*

Class who create a history item

create (*_id*, *data*, *name=None*)
Create wanted item

Parameters

- **_id** (*str*) – id of the item. Often equal to id in alignak backend
- **data** (*dict* / *list*) – data of the item
- **name** (*str*) – name of the item if available

static get_history_icon_name_from_message (*event_type*)
Return icon name related to message or event type

Parameters

- **message** (*str*) – message of an history event
- **event_type** (*str*) – type of history event

Returns icon name

Return type *str*

static get_request_model ()
Return the request model for history requests

Returns request model for history endpoint

Return type *dict*

get_tooltip ()
Return the tooltip message depending state and actions

Returns tooltip message

Return type *str*

update_data (*key*, *new_value*)
Update data of the wanted key

Parameters

- **key** (*str*) – key to update
- **new_value** – new value of the key

10.3.5 Host

Host manage creation of host item for backend `host` endpoint

class `alignak_app.items.host.Host`

Bases: `alignak_app.items.item.Item`

Class who create a host item

create (*_id*, *data*, *name=None*)

Create wanted item

Parameters

- **_id** (*str*) – id of the item. Often equal to id in alignak backend
- **data** (*dict* / *list*) – data of the item
- **name** (*str*) – name of the item if available

static `get_available_icons()`

Return list of available icons for a Host item

Returns list of available icons for Host

Return type `list`

get_display_name ()

Return alias or name if available

Returns name or alias

Return type `str`

static `get_request_model()`

Return the request model for host requests

Returns request model for host endpoint

Return type `dict`

get_tooltip ()

Return the tooltip message depending state and actions

Returns tooltip message

Return type `str`

update_data (*key*, *new_value*)

Update data of the wanted key

Parameters

- **key** (*str*) – key to update
- **new_value** – new value of the key

10.3.6 Live Synthesis

Live Synthesis manage creation of livesynthesis item for backend `livesynthesis` endpoint

class `alignak_app.items.livesynthesis.LiveSynthesis`

Bases: `alignak_app.items.item.Item`

Class who create livesynthesis item

create (*_id*, *data*, *name=None*)

Create wanted item

Parameters

- **_id** (*str*) – id of the item. Often equal to id in alignak backend
- **data** (*dict* / *list*) – data of the item
- **name** (*str*) – name of the item if available

static get_request_model ()

Return the request model for livesynthesis requests

Returns request model for livesynthesis endpoint

Return type *dict*

static get_synthesis_count_model ()

Return the synthesis count model

Returns synthesis count model

Return type *dict*

get_tooltip ()

Return the tooltip message depending state and actions

Returns tooltip message

Return type *str*

update_data (*key*, *new_value*)

Update data of the wanted key

Parameters

- **key** (*str*) – key to update
- **new_value** – new value of the key

10.3.7 Period

Period manage creation of period item for backend timeperiod endpoint

class alignak_app.items.period.**Period**

Bases: *alignak_app.items.item.Item*

Class who create a period item

create (*_id*, *data*, *name=None*)

Create wanted item

Parameters

- **_id** (*str*) – id of the item. Often equal to id in alignak backend
- **data** (*dict* / *list*) – data of the item
- **name** (*str*) – name of the item if available

get_display_name ()

Return alias or name if available

Returns name or alias

Return type *str*

static get_request_model()
Return the request model for timeperiod requests
Returns request model for timeperiod endpoint
Return type `dict`

get_tooltip()
Return the tooltip message depending state and actions
Returns tooltip message
Return type `str`

update_data(key, new_value)
Update data of the wanted key
Parameters

- **key** (`str`) – key to update
- **new_value** – new value of the key

10.3.8 Realm

Realm manage creation of realm item for backend realm endpoint

class `alignak_app.items.realm.Realm`
Bases: `alignak_app.items.item.Item`
Class who create a realm item

create(_id, data, name=None)
Create wanted item
Parameters

- **_id** (`str`) – id of the item. Often equal to id in alignak backend
- **data** (`dict` / `list`) – data of the item
- **name** (`str`) – name of the item if available

get_display_name()
Return alias or name if available
Returns name or alias
Return type `str`

static get_request_model()
Return the request model for realm requests
Returns request model for realm endpoint
Return type `dict`

get_tooltip()
Return the tooltip message depending state and actions
Returns tooltip message
Return type `str`

update_data(key, new_value)
Update data of the wanted key

Parameters

- **key** (*str*) – key to update
- **new_value** – new value of the key

10.3.9 Service

Service manage creation of service item for backend `service` endpoint

class `alignak_app.items.service.Service`

Bases: `alignak_app.items.item.Item`

Class who create a service item

create (*_id*, *data*, *name=None*)

Create wanted item

Parameters

- **_id** (*str*) – id of the item. Often equal to id in alignak backend
- **data** (*dict* / *list*) – data of the item
- **name** (*str*) – name of the item if available

static `get_available_icons()`

Return list of available icons for a Service item

Returns list of available icons for Service

Return type `list`

get_display_name ()

Return alias or name if available

Returns name or alias

Return type `str`

static `get_request_model()`

Return the request model for service requests

Returns request model for service endpoint

Return type `dict`

static `get_service_states_nb()`

Return all service state in a dict with int() as zero

Returns all service state with int() as zero

Return type `dict`

get_tooltip ()

Return the tooltip message depending state and actions

Returns tooltip message

Return type `str`

update_data (*key*, *new_value*)

Update data of the wanted key

Parameters

- **key** (*str*) – key to update

- **new_value** – new value of the key

10.3.10 User

User manage creation of user item

class `alignak_app.items.user.User`

Bases: `alignak_app.items.item.Item`

Class who create user item for backend `user` endpoint

create (`_id`, `data`, `name=None`)

Create wanted item

Parameters

- **_id** (`str`) – id of the item. Often equal to id in alignak backend
- **data** (`dict` / `list`) – data of the item
- **name** (`str`) – name of the item if available

static `get_request_model()`

Return the request model for user requests

Parameters **token** (`str`) – token of user

Returns request model for user endpoint

Return type `dict`

get_role ()

Get user role

Returns role of user

Return type `str`

get_tooltip ()

Return the tooltip message depending state and actions

Returns tooltip message

Return type `str`

update_data (`key`, `new_value`)

Update data of the wanted key

Parameters

- **key** (`str`) – key to update
- **new_value** – new value of the key

10.4 Utils

The Utils package contains classes and functions configurations

10.4.1 Settings

Settings manage configurations of Alignak-app

- `settings.cfg`: contains configurations of Alignak-app (stored in `ALIGNAKAPP_USR_DIR`)
- `images.ini`: contains all images names (stored in `ALIGNAKAPP_APP_DIR`)
- `style.css`: contains css of Alignak-app (stored in `ALIGNAKAPP_APP_DIR`)

class `alignak_app.utils.config.Settings`

Bases: `object`

Class who read and create configuration for Alignak-app

edit_setting_value (*section, option, new_value*)

Set an option in configuration file

Parameters

- **section** (*str*) – section to edit
- **option** (*str*) – option to edit, corresponding to the wanted section
- **new_value** (*str*) – new value to set in place of old

get_config (*section, option, boolean=False*)

Return global application configuration values

Parameters

- **section** (*str*) – wanted configuration section
- **option** (*str*) – wanted configuration option
- **boolean** (*bool*) – define if value is boolean or not

Returns configuration value

Return type `str | bool`

get_image (*name*)

Return the path of wanted image

Parameters **name** (*str*) – name of image

Returns full path of image

Return type `str`

init_config ()

Initialize configurations

init_css ()

Init the css file and fill app_css

`alignak_app.utils.config.get_url_endpoint_from_icon_name` (*icon_name*)

Return endpoint depending of “icon_name”

Parameters **icon_name** (*str*) – naem of icon

`alignak_app.utils.config.open_url` (*endpoint='login'*)

Open web browser on wanted endpoint

Parameters **endpoint** (*str*) – endpoint of webui

10.4.2 Installer

Installer check and manage installation of Alignak-app

- Check ALIGNAKAPP_APP_DIR, ALIGNAKAPP_USR_DIR and ALIGNAKAPP_LOG_DIR folders
- Check required files for application and user
- Install configuration and daemon files for user

class alignak_app.utils.installer.Installer

Bases: `object`

Class who create and check: folders, files and environment variables for Alignak-App

check_environment (*mode='start'*)

Assign fields to environment variables if they don't exist

check_install_files ()

Check Alignak-app files

Returns if all files are present

Return type `bool`

check_install_folders ()

Check Alignak-app folders

Returns if all folders are present

Return type `bool`

check_installation (*mode='start'*)

Check Alignak-app installation

check_user_installation ()

Check user installation files

init_environment ()

Assign environment variables to fields if they exists

install ()

Install Alignak-app user files on system

10.4.3 System

System manage interactions with folders and files for Alignak-app

alignak_app.utils.system.**file_executable** (*filename*)

Make filename executable

Parameters **filename** (*str*) – file to make executable

alignak_app.utils.system.**install_file** (*origin_dir, dest_dir, filename*)

Install (copy) filename from an origin folder to a destination folder

Parameters

- **origin_dir** (*str*) – origin folder where file to copy is located
- **dest_dir** (*str*) – destination folder where file will be copied
- **filename** (*str*) – name of file to copy

`alignak_app.utils.system.mkdir(folder)`

Make wanted folder

Parameters `folder` (*str*) – folder to create

Returns if creation is success or not

Return type `bool`

`alignak_app.utils.system.read_config_file(cfg_parser, filename)`

Read configuration file and assign it to configParser object

Parameters

- **cfg_parser** (*configparser.ConfigParser*) – configparser object
- **filename** (*str*) – name of file to read

Returns corresponding filename if read is success

Return type `str`

`alignak_app.utils.system.write_file(origin_dir, dest_dir, filename, formatted_var=None)`

Write a file from an origin to a destination, with formatted variables if needed

Parameters

- **origin_dir** (*str*) – origin folder where file is located
- **dest_dir** (*str*) – destination folder where file will be located
- **filename** (*str*) – name of file to write
- **formatted_var** (*tuple*) – tuple of variable to format origin file

`alignak_app.utils.system.write_rc_file(filename)`

Write RC file to add autocompletion for Alignak-app

Parameters **filename** (*str*) – name of file for autocompletion

10.4.4 Logs

Logs manage application logs directory and file who are created inside `ALIGNAKAPP_LOG_DIR`.

`alignak_app.utils.logs.create_logger()`

Create the logger for Alignak-App

Returns the RootLogger of App

Return type `logging.RootLogger`

10.4.5 Time

Time manage time and date for Alignak-app

`alignak_app.utils.time.get_time_diff_since_last_timestamp(timestamp)`

Return the diff between the last time stamp

Parameters **timestamp** (*float*) – timestamp of the last check

Returns time difference formatted

Return type `str`

10.5 Locales

The Locales package contains classes to manage translation.

10.5.1 Locales

Locales manage localization of Alignak-app

```
alignak_app.locales.locales.init_localization()
```

Application localization

Returns gettext translator method

Return type method

10.6 QObjects

The QObjects package contains PyQt5 classes of Alignak-app

10.6.1 App Main

App Main manage creation of QMainWindow for:

- *Dock* (Right part)
- *Panel* (Left part)

```
class alignak_app.qobjects.app_main.AppQMainWindow (parent=None)
```

Bases: PyQt5.QtWidgets.QMainWindow

Class who create QMainWindow of Alignak-app

```
connect_dock_buttons ()
```

Connect dock QWidget buttons to host and problems tab

```
initialize ()
```

Initialize QMainWindow for App

```
mouseMoveEvent (event)
```

QWidget.mousePressEvent(QMouseEvent)

```
mousePressEvent (event)
```

QWidget.mousePressEvent(QMouseEvent)

```
open_host_widget ()
```

Show HostQWidget

```
open_problems_widget ()
```

Show ProblemsQWidget

10.6.2 Common

The Common package contains common QObjects for Alignak-app

Actions

Actions manage global QDialog's for actions : Acknowledge and Downtime

```
class alignak_app.qobjects.common.actions.AckQDialog (parent=None)
    Bases: PyQt5.QtWidgets.QDialog
```

Class who create Acknowledge QDialog for hosts/services

```
initialize (item_type, item_name, comment)
    Initialize Acknowledge QDialog
```

Parameters

- **item_type** (*str*) – type of item to acknowledge : host | service
- **item_name** (*str*) – name of the item to acknowledge
- **comment** (*str*) – the default comment of action

```
mousePressEvent (event)
    QWidget.mousePressEvent(QMouseEvent)
```

```
mousePressEvent (event)
    QWidget.mousePressEvent(QMouseEvent)
```

```
class alignak_app.qobjects.common.actions.ActionsQWidget (parent=None)
    Bases: PyQt5.QtWidgets.QWidget
```

Class who create Actions QWidget

```
add_acknowledge ()
    Create AckQDialog and manage acknowledge
```

```
add_downtime ()
    Create DownQDialog and manage downtime
```

```
initialize (item)
    Initialize Actions QWidget
```

```
update_widget ()
    Update QWidget
```

```
class alignak_app.qobjects.common.actions.DownQDialog (parent=None)
    Bases: PyQt5.QtWidgets.QDialog
```

Class who create Downtime QDialog for hosts/services

```
duration_to_seconds ()
    Return “duration” QTimeEdit value in seconds
```

Returns “duration” in seconds

Return type `int`

```
handle_accept ()
    Check if end_time timestamp is not lower than start_time
```

```
initialize (item_type, item_name, comment)
    Initialize Downtime QDialog
```

Parameters

- **item_type** (*str*) – type of item to acknowledge : host | service
- **item_name** (*str*) – name of the item to acknowledge

- **comment** (*str*) – the default comment of action

```
mousePressEvent (event)  
    QWidget.mousePressEvent(QMouseEvent)  
  
mousePressEvent (event)  
    QWidget.mousePressEvent(QMouseEvent)
```

Buttons

Buttons manage global QPushButton

```
class alignak_app.qobjects.common.buttons.ToggleQWidgetButton (parent=None)  
    Bases: PyQt5.QtWidgets.QWidget  
  
    Class who create QWidget with “toggle” QPushButton inside.  
  
    get_btn_state ()  
        Return “toggle_btn” state  
  
        Returns QPushButton state  
  
        Return type bool  
  
    initialize ()  
        Initialize QWidget  
  
    update_btn_state (state)  
        Update QPushButton state, ObjectName and StyleSheet  
  
        Parameters state (bool) – current state of QPushButton
```

Frames

Frames manage global QFrames for Alignak-app

```
class alignak_app.qobjects.common.frames.AppQFrame (parent=None)  
    Bases: PyQt5.QtWidgets.QFrame  
  
    Class who create a QFrame container for App QWidgets  
  
    add_widget (widget)  
        Add the child QWidget of AppQWidget  
  
        Parameters widget (QWidget) – QWidget to add  
  
    close_widget ()  
        Close QFrame and child widget  
  
    get_title_widget (title)  
        Return the title QWidget for App windows  
  
        Parameters title (str) – title of frame  
  
        Returns a title QWidget  
  
        Return type QWidget  
  
    initialize (title)  
        Initialize the QFrame, with its “title”  
  
        Parameters title (str) – title of frame
```

```

minimize()
    Minimize QFrame

minimize_maximize()
    Minimize / Maximize QFrame

mousePressEvent (event)
    QWidget.mousePressEvent(QMouseEvent)

mousePressEvent (event)
    QWidget.mousePressEvent(QMouseEvent)

show_widget()
    Show and center QFrame and child widget

```

```

alignak_app.qobjects.common.frames.get_frame_separator (vertical=False)
    Return a frame separator

    Parameters vertical (bool) – define if separator is vertical or horizontal

    Returns frame separator

    Return type QFrame

```

Labels

Labels manage global QLabels and QPixmap

```

alignak_app.qobjects.common.labels.get_icon_item (item_type, problem_nb)
    Return QPixmap with the corresponding image

    Parameters
        • item_type (str) – type of item: host, service or problem
        • problem_nb (int) – problem number

    Returns QPixmap with corresponding image

    Return type QPixmap

```

```

alignak_app.qobjects.common.labels.get_icon_pixmap (value, icons)
    Return corresponding icon name in icons

    Parameters
        • value (bool) – value of attribute
        • icons (list) – pair of icons

    Returns pixmap icon corresponding to value

    Return type QPixmap

```

Widgets

Widgets manage global QWidget

```

class alignak_app.qobjects.common.widgets.LogoQWidget
    Bases: PyQt5.QtWidgets.QWidget

    Class who manage creation of Logo QWidget

```

initialize (*child_widget*, *title*, *exitapp*)

Initialize QWidget

Parameters

- **child_widget** (*QWidget*) – widget child of LogoQWidget, needed for action button and layout
- **title** (*str*) – title of widget
- **exitapp** (*bool*) – define if close button close application or just child QWidget

minimize ()

Minimize QWidget

minimize_maximize ()

Minimize / Maximize QWidget

paintEvent (_)

Override to apply “background-color” property of QWidget

`alignak_app.qobjects.common.widgets.center_widget` (*widget*)

Center QWidget

Parameters **widget** (*QWidget*) – widget to center

`alignak_app.qobjects.common.widgets.get_logo_widget` (*child_widget*, *title*, *exitapp=False*)

Return LogoQWidget with alignak logo

Parameters

- **child_widget** (*QWidget*) – widget child of LogoQWidget, needed for action button and layout
- **title** (*str*) – title of widget
- **exitapp** (*bool*) – define if close button close application or just child QWidget

Returns Logo QWidget with buttons and child QWidget

Return type QWidget

10.6.3 Dock (Rigt part)

The Dock package contains classes who manage Dock UI (Right part)

Password

Password manage Qdialog for user password

class `alignak_app.qobjects.dock.password.PasswordQDialog` (*parent=None*)

Bases: `PyQt5.QtWidgets.QDialog`

Class who create PasswordDialog QDialog

handle_confirm ()

Handle accept_btn for password

initialize ()

Initialize QDialog for PasswordDialog

Status

Status manage creation of QWidget and QDialog for Alignak status:

- Alignak daemons status: status of each daemons
- Alignak backend status: status of backend connection

class alignak_app.qobjects.dock.status.**StatusQDialog** (*parent=None*)

Bases: PyQt5.QtWidgets.QDialog

Class who create QWidget for Daemons status.

add_daemon_labels (*daemon_item, line*)

Add daemon QLabels to layout

Parameters

- **daemon_item** (*alignak_app.items.item_daemon.Daemon*) – daemon item
- **line** (*int*) – current line of layout

add_daemon_titles_labels (*line*)

Add QLabels titles for daemons to layout

Parameters **line** (*int*) – current line of layout

get_buttons_widget ()

Return QWidget with buttons

Returns widget with ok and refresh buttons

Return type QWidget

initialize ()

Initialize QDialog

set_daemons_labels (*daemons*)

Initialize the daemon QLabels for each daemons

Parameters **daemons** (*list*) – list of daemon items

update_dialog ()

Update StatusQDialog labels

class alignak_app.qobjects.dock.status.**StatusQWidget**

Bases: PyQt5.QtWidgets.QWidget

Class who display daemons and backend status

initialize ()

Initialize QWidget

paintEvent (_)

Override to paint background

show_status_dialog ()

Update and show StatusQDialog

update_status ()

Update daemons and backend status

Token

Token manage creation of QDialog for user token

```
class alignak_app.qobjects.dock.token.TokenQDialog (parent=None)
    Bases: PyQt5.QtWidgets.QDialog

    Class who create Token QDialog

    get_token_widget ()
        Return token QWidget

        Returns token QWidget

        Return type QWidget

    initialize ()
        Initialize QDialog for PasswordDialog
```

User Notes

User Notes manage creation of QDialog for user notes

```
class alignak_app.qobjects.dock.user_notes.UserNotesQDialog (parent=None)
    Bases: PyQt5.QtWidgets.QDialog

    Class who create UserNotes QDialog

    accept_notes ()
        Set QDialog notes to Rejected or Accepted (prevent to patch for nothing)

    get_user_notes_widget ()
        Return notes QWidget

        Parameters notes (str) – notes of the user

        Returns notes QWidget

        Return type QWidget

    initialize (notes)
        Initialize QDialog for UserNotesQDialog
```

User Options

User Options manage creation options of QDialogs for user notifications options

```
class alignak_app.qobjects.dock.user_options.UserOptionsQDialog (parent=None)
    Bases: PyQt5.QtWidgets.QDialog

    Class who create options QDialog

    get_notifications_widget (item_type, options)
        Create QWidget with options and their icons

        Parameters

        • item_type (str) – define item type for options: host or service

        • options (list) – list of notification options

        Returns QWidget with options and icons
```

Return type QWidget

static `get_selected_options` (*options*)

Return the options who are selected or not

Parameters

- **item_type** (*str*) – type of item we want options
- **options** (*list*) – options for the wanted item type

Returns dict of options selected

Return type dict

initialize (*item_type*, *options*)

Initialize QDialog with App widget logo and options QWidget

Parameters

- **item_type** (*str*) – define item type for options: host or service
- **options** (*list*) – list of notification options

`alignak_app.qobjects.dock.user_options.show_options_dialog` (*item_type*, *notification_options*)

Show the UserOptionsQDialog

Parameters

- **item_type** (*str*) – type of item we want options
- **notification_options** (*list*) – options for the wanted item type

Buttons

Buttons manage creation of QWidget with QPush buttons for:

- Hosts Synthesys View
- Problems view
- User widget
- Webui to reach Alignak-Webui

class `alignak_app.qobjects.dock.buttons.ButtonsQWidget` (*parent=None*)

Bases: `PyQt5.QtWidgets.QWidget`

Class who create buttons for Dock QWidget

initialize ()

Initialize QWidget

open_user_widget ()

Show UserQWidget

update_widget ()

Update the QWidget buttons

Dock

Dock manage creation of QWidget for **Dock** (Right part)

```
class alignak_app.qobjects.dock.dock.DockQWidget (parent=None)
    Bases: PyQt5.QtWidgets.QWidget

    Class who create QWidget for dock

    initialize()
        Initialize dock QWidget

    send_spy_events()
        Send event for one host spied
```

Events

Events manage creation QWidget for events

```
class alignak_app.qobjects.dock.events.EventItem
    Bases: PyQt5.QtWidgets.QListWidgetItem

    Class who create an event QListWidgetItem

    close_item()
        Hide items when timer is finished

    static get_color_event()
        Return corresponding color of event type

        Parameters event_type (str) – the type of event

        Returns the associated color with the event

        Return type str

    initialize (event_type, msg, timer=False, spied_on=False, host=None)
        Initialize QListWidgetItem

        Parameters

        • event_type (str) – the type of event: OK, DOWN, ACK, ...

        • msg (str) – message of event

        • timer – timer to hide event at end of time

        • spied_on (bool) – make event spy able

        • host (str) – _id of host. Only necessary if “be_spied” is True

class alignak_app.qobjects.dock.events.EventsQWidget
    Bases: PyQt5.QtWidgets.QWidget

    Class who create QWidget for events

    add_event (event_type, msg, timer=False, spied_on=False, host=None)
        Add event to events list

        Parameters

        • event_type (str) – the type of event: OK, DOWN, ACK, ...

        • msg (str) – message of event

        • timer – timer to hide event at end of time
```


- **spied_on** (*bool*) – make event spy able
- **host** (*str*) – data of host. Only necessary if “be_spied” is True

initialize ()

Intialize QWidget

remove_event (*item=None*)

Remove item when user double click on an item

Parameters **item** (*EventItem*) – item to remove, else remove the current row

remove_timer_event (*event*)

Remove EventItem with timer

Parameters **event** (*EventItem*) – EventItem with timer

send_datamanager_events ()

Add events stored in DataManager

`alignak_app.qobjects.dock.events.get_events_widget ()`

Return EventsQWidget instance

Returns EventsQWidget instance

Return type *EventsQWidget*

`alignak_app.qobjects.dock.events.init_event_widget ()`

Initialize the global EventsQWidget. Used for drag & drop and send events

`alignak_app.qobjects.dock.events.send_event (event_type, msg, timer=False, spied_on=False, host=None)`

Access function to simplify code in rest of application

Parameters **event_type** – type of event, define color of EventItem()

- **green**: [OK, UP]
- **blue**: [UNKNOWN, INFO]
- **orange**: [WARNING, UNREACHABLE, WARN]
- **red**?: [DOWN, CRITICAL, ALERT]
- **yellow**: [ACK]
- **yellow**: [DOWNTIME]
- **yellow**: [TODO]

Parameters

- **msg** (*str*) – message of event
- **timer** – timer to hide event at end of time
- **spied_on** (*bool*) – make event spy able
- **host** (*str*) – _id of host. Only necessary if “spied_on” is True

Livestate

Livestate manage creation fo QWidget to resume number of:

- Number of hosts monitored and in problem

- Number of services monitored and in problem
- Number of item monitored and in problems

class alignak_app.qobjects.dock.livestate.LivestateQWidget
Bases: PyQt5.QtWidgets.QWidget

Class who display items livestate: hosts, services and number of problems

get_item_type_widget (*item_type*, *problem_nb*, *total_nb*)
Create and return QWidget with backend data

Parameters

- **item_type** (*str*) – type of item: host, service, problem
- **problem_nb** (*int*) – number of problems for item type
- **total_nb** (*int*) – total number of item type

Returns widget with its data

Return type QWidget

initialize ()
Initialize QWidget

paintEvent (_)
Override paintEvent to paint background

update_labels ()
Update QLabels of QWidget

Spy

Spy manage creation of QWidget for host items who are spied

class alignak_app.qobjects.dock.spy.SpyQListWidget
Bases: PyQt5.QtWidgets.QListWidget

Class who create QListWidget for spied hosts

add_spy_host (*host_id*)
Add a host to spied list and create corresponding EventItem()

Parameters **host_id** (*str*) – “_id” of host to spy

dragMoveEvent (*event*)

Override dragMoveEvent. Only accept EventItem() objects who are “spied_on” and not already spied

Parameters **event** – event triggered when something move

dropEvent (*event*)

Override dropEvent. Get dropped item data, create a new one, and delete the one who is in EventsQWidget

Parameters **event** – event triggered when something is dropped

class alignak_app.qobjects.dock.spy.SpyQWidget
Bases: PyQt5.QtWidgets.QWidget

Class who create QWidget for spied hosts

```
initialize()
    Intialize QWidget

remove_event()
    Remove item when user double click on an item
```

User

User manage creation of QWidget who display user profile who use Alignak-app

```
class alignak_app.qobjects.dock.user.UserQWidget (parent=None)
    Bases: PyQt5.QtWidgets.QWidget
```

Class who create QWidget for User profile.

```
enable_notifications (notification_type, btn_state)
    Enable notification for the wanted type: hosts or services
```

```
get_hosts_notif_widget ()
    Create and return notification QWidget for hosts
```

Returns hosts notification QWidget

Return type QWidget

```
get_informations_widget ()
    Create and return QWidget with user informations
```

Returns main QWidget

Return type QWidget

```
get_notes_scrollarea ()
    Return QScrollArea widget for user notes
```

Returns user notes QScrollArea

Return type QScrollArea

```
get_notifications_widget ()
    Create and return notification QWidget for hosts and services
```

Returns notifications QWidget

Return type QWidget

```
get_services_notif_widget ()
    Create and return notification QWidget for services
```

Returns services notification QWidget

Return type QWidget

```
initialize()
    Initialize User QWidget
```

```
patch_data ()
    Hide and show QLabel for notes or PATCH password
```

```
static show_token_dialog ()
    Show TokenQDialog
```

```
update_widget ()
    Update UserQWidget
```

10.6.4 Login

The Login package contains classes who manage Alignak-app login

Login

Login manage creation of QDialog for user login

```
class alignak_app.qobjects.login.login.LoginQDialog (parent=None)
    Bases: PyQt5.QtWidgets.QDialog

    Class who create login QDialog.

    accept_login ()
        Accept Login or not if backend is connected

    create_widget ()
        Create widget login

    static handle_server ()
        Handle for server button

    mouseMoveEvent (event)
        QWidget.mousePressEvent(QMouseEvent)

    mousePressEvent (event)
        QWidget.mousePressEvent(QMouseEvent)

    showEvent (_)
        QDialog.showEvent(QShowEvent)
```

Server

Server manage creation of QDialog for Alignak backend server settings

```
class alignak_app.qobjects.login.server.ServerQDialog (parent=None)
    Bases: PyQt5.QtWidgets.QDialog

    Class who create Server QDialog.

    initialize_dialog ()
        Initialize Server QDialog

    mouseMoveEvent (event)
        QWidget.mousePressEvent(QMouseEvent)

    mousePressEvent (event)
        QWidget.mousePressEvent(QMouseEvent)
```

10.6.5 Panel (Left part)

The Panel package contains classes who create Panel UI

Panel

Panel manage creation of QWidget for Panel (Left part)

```

class alignak_app.qobjects.panel.panel.PanelQWidget (parent=None)
    Bases: PyQt5.QtWidgets.QWidget

    Class who manage Panel with Host and Services QWidgets

    create_line_search (hostnames_list=None)
        Add all hosts to QLineEdit and set QCompleter

        Parameters hostnames_list (list) – list of host names

    display_host ()
        Display and update HostQWidget

    dragEnterEvent (event)
        Override dragEnterEvent.

        Parameters event – event triggered when something enter

    dragMoveEvent (event)
        Override dragMoveEvent. Only accept EventItem() objects who are “spied_on” and not already spied

        Parameters event – event triggered when something move

    dropEvent (event)
        Override dropEvent. Get dropped item data, create a new one, and delete the one who is in EventsQWidget

        Parameters event – event triggered when something is dropped

    get_search_widget ()
        Create and return the search QWidget

        Returns search QWidget

        Return type QWidget

    get_synthesis_widget ()
        Return synthesis QWidget()

        Returns synthesis QWidget()

        Return type QWidget

    initialize (spy_widget)
        Create the QWidget with its items and layout.

        Parameters spy_widget (alignak_app.pyqt.dock.widgets.spy.SpyQWidget) – SpyQWidget to allow HostQWidget add spied host

    spy_host ()
        Spy host who is available in line_search QLineEdit

```

Dashboard

Dashboard manage the creation of QWidgets to display number of:

- Each states of Hosts items
- Each states of Services items

```
class alignak_app.qobjects.panel.dashboard.DashboardQWidget (parent=None)
    Bases: PyQt5.QtWidgets.QWidget

    Class who manage Host and Services resume QWidget with number of:

        • Hosts: UP, UNREACHABLE, DOWN

        • Services: OK, WARNING, CRITICAL, UNKNWON, UNREACHABLE

        • Hosts and services: NOT_MONITORED, ACKNOWLEDGED, DOWNTIMED

    get_host_resume_widget ()
        Return Host resume QWidget

    get_services_resume_widget ()
        Return Services resume QWidget

    initialize ()
        Initialize QWidget

    open_item_type_url (item_type)
        Retrieve sender to send right endpoint to open_url() function for item type

        Parameters item_type (str) – type of item: hosts | services

    update_dashboard ()
        Update number of items in dashboard
```

History

History manage creation of QWidget to display history of a host

```
class alignak_app.qobjects.panel.history.HistoryQWidget (parent=None)
    Bases: PyQt5.QtWidgets.QWidget

    Class who create the History QWidget for host

    static get_history_widget_model (hostname)
        Create and return event QWidgets

        Parameters

            • history_event (dict) – event of history given by backend

            • hostname (str) – name of host

        Returns history Qwidget model with data

        Return type QWidget

    initialize (hostname, host_history)
        Initialize History QWidget

        Parameters

            • hostname (str) – name of the host

            • host_history (History) – history of host
```

Host

Host manage creation of QWidget to display host data

```

class alignak_app.qobjects.panel.host.HostQWidget (parent=None)
    Bases: PyQt5.QtWidgets.QWidget

    Class who create QWidget to display host data

    get_actions_widget ()
        Return QWidget with actions buttons

        Returns widget with buttons

        Return type QWidget

    get_host_icon_widget ()
        Return QWidget with overall icon state and host name

        Returns widget with host name and icon

        Return type QWidget

    get_last_check_widget ()
        Return QWidget with last check data

        Returns widget with last check data

        Return type QWidget

    get_notes_output_widget ()
        Return QWidget with output and notes data

        Returns widget with host output and notes

        Return type QWidget

    get_variables_widget ()
        Return QWidget with host variables

        Returns widget with host variables

        Return type QWidget

    initialize ()
        Initialize QWidget

    patch_host_checks (check_type, state)
        Patch the host check: 'active_checks_enabled' | 'passive_checks_enabled'

        Parameters

        • check_type (str) – type of check: 'active_checks_enabled' | 'passive_checks_enabled'

        • state (bool) – state of Toggle button

    set_data (hostname)
        Set data of host and service

        Parameters hostname (str) – name of host to display

    show_history ()
        Create and show HistoryQWidget

    update_host (hostname=None)
        Update HostQWidget data and QLabels

        Parameters hostname (str) – name of host who is display

```

Number Services

Number Services manage creation of QWidget to display number of host services, for each state

```
class alignak_app.qobjects.panel.number_services.NumberServicesQWidget (parent=None)  
    Bases: PyQt5.QtWidgets.QWidget
```

Class who create QWidget number of services of a host

```
initialize()  
    Initialize QWidget
```

```
update_widget(services_items, host_name)  
    Update QWidget
```

Parameters

- **services_items** (*list*) – list of Service items
- **host_name** (*str*) – name of host attached to services

Problems

Problems manage creation of QWidget to display problems found in Alignak backend:

- **Hosts:** DOWN
- **Services:** WARNING, CRITICAL, UNKNOWN

```
class alignak_app.qobjects.panel.problems.ProblemsQWidget (parent=None)  
    Bases: PyQt5.QtWidgets.QWidget
```

Class who create Problems QWidget

```
static get_icon_widget(state)  
    Return QWidget with corresponding icon to item state
```

Parameters

- **item_type** (*str*) – type of item: host | service
- **state** (*str*) – state of item

Returns QWidget with corresponding icon

Return type QWidget

```
get_problems_widget_title()  
    Return QWidget title with number of problems and refresh QPushButton
```

Returns QWidget with number of problems

Return type QWidget

```
get_table_items_problem(item)  
    Return list of QWidgets and QTableWidgetItem
```

Parameters **item** (*alignak_app.core.models.host.Host | alignak_app.core.models.service.Service*) – ItemModel instance with its data

Returns list of QWidgets and QTableWidgetItem

Return type *list*

```
initialize()  
    Initialize QWidget
```



```
update_problems_data()
```

Update data of QTableWidget and problems title

Service

Service manage creation of QWidget to display service data

```
class alignak_app.qobjects.panel.service.ServiceDataQWidget (parent=None)
    Bases: PyQt5.QtWidgets.QWidget

    Class who create QWidget with service data

    get_actions_widget()
        Return QWidget with actions buttons

        Returns widget with buttons

        Return type QWidget

    get_last_check_widget()
        Return QWidget with last check data

        Returns widget with last check data

        Return type QWidget

    get_service_icon_widget()
        Return QWidget with its icon and name

        Returns widget with icon and name

        Return type QWidget

    initialize()
        Initialize QWidget

    periodic_refresh()
        Refresh QWidget periodically

    update_widget (service=None)
        Update ServiceDataQWidget

        Parameters service (alignak_app.core.models.service.Service) – Service
        item with its data
```

Service Tree Item

Service Tree Item manage creation of QTreeWidgetItem(s) to store and display Service item data

```
class alignak_app.qobjects.panel.service_tree_item.ServiceTreeItem (parent=None)
    Bases: PyQt5.QtWidgets.QTreeWidgetItem

    Class who create QTreeWidgetItem with service data

    initialize (service_item)
        Initialize the QTreeWidgetItem

        Parameters service_item (alignak_app.core.models.service.Service) –
        service item with its data

    update_item()
        Update QIcon of service
```

Services

Services manage creation of QWidget to display the services

```
class alignak_app.qobjects.panel.services.ServicesQWidget (parent=None)
    Bases: PyQt5.QtWidgets.QWidget

    Class wo create services QWidget

    initialize()
        Initialize QWidget

    set_data(hostname)
        Set data of host and service

        Parameters hostname (str) – name of host to display

    update_service_data()
        Update ServiceDataqWidget

    update_widget()
        Update the service QWidget
```

10.6.6 Systray

The Systray package contains classes who create QSystemTrayIcon UI.

QActions Factory

QAction Factory manage creation of QActions for TrayIcon.

```
class alignak_app.qobjects.systray.qactions_factory.QActionFactory
    Bases: object

    Create QActions with its icon and content

    create(name, content, parent)
        Create QAction

    get_action(name)
        Return QAction

        Parameters name (str) – name of the QAction

        Returns wanted QAction

        Return type QAction
```

Tray Icon

Tray Icon manage the creation of QSystemTrayIcon for Alignak-app menus

```
class alignak_app.qobjects.systray.tray_icon.TrayIcon (icon, parent=None)
    Bases: PyQt5.QtWidgets.QSystemTrayIcon

    Class who create QMenu and QAction.

    add_about_menu()
        Create AppAbout QWidget and “about” action.
```

```
add_alignak_menu()  
    Create “dock” action  
  
add_reload_menu()  
    Create “reload” action  
  
build_menu()  
    Initialize and create each action of menu.  
  
check_connection()  
    Check periodically connection for App  
  
create_quit_action()  
    Create quit action.  
  
static quit_app()  
    Quit application and close all widgets handle by application  
  
static reload_configuration()  
    Reload configuration
```

10.7 QThreads

The QThreads package contains classes who manage application threads

10.7.1 Backend Thread

Backend Thread manage creation of QThread for backend requests

```
class alignak_app.qthreads.thread.BackendQThread(thread, data=None)  
    Bases: PyQt5.QtCore.QThread  
  
    Class who create a QThread to trigger requests  
  
    run()  
        Run the QThread. Trigger actions depending on the selected thread_name
```

10.7.2 Thread Manager

Thread Manager manage creation of QObject for launched threads

```
class alignak_app.qthreads.threadmanager.ThreadManager(parent=None)  
    Bases: PyQt5.QtCore.QObject  
  
    Class who create BackendQThreads periodically, to request on Alignak Backend endpoints  
  
    add_priority_thread(thread_name, data)  
        Launch a thread with higher priority (doesn't wait launch_threads() function)  
  
        Parameters  
        • thread_name (str) – name of priority thread  
        • data (dict) – data to give to thread for request  
  
    clean_threads()  
        Clean current BackendQThreads
```

get_threads_to_launch()

Return the threads_to_launch to run in BackendQRunnable

Returns threads_to_launch to run

Return type `list`

launch_threads()

Launch periodically threads

stop_priority_threads()

Stop priority threads

stop_threads()

Stop ThreadManager and close all running BackendQThreads

CHAPTER 11

Indices and tables

- `genindex`
- `modindex`
- `search`

a

alignak_app, 23
alignak_app.app, 25
alignak_app.backend, 26
alignak_app.backend.backend, 26
alignak_app.backend.datamanager, 28
alignak_app.items, 29
alignak_app.items.daemon, 31
alignak_app.items.event, 32
alignak_app.items.history, 33
alignak_app.items.host, 33
alignak_app.items.item, 29
alignak_app.items.livesynthesis, 34
alignak_app.items.period, 35
alignak_app.items.realm, 36
alignak_app.items.service, 37
alignak_app.items.user, 38
alignak_app.locales, 42
alignak_app.locales.locales, 42
alignak_app.qobjects, 42
alignak_app.qobjects.app_main, 42
alignak_app.qobjects.common, 42
alignak_app.qobjects.common.actions, 42
alignak_app.qobjects.common.buttons, 44
alignak_app.qobjects.common.frames, 44
alignak_app.qobjects.common.labels, 45
alignak_app.qobjects.common.widgets, 45
alignak_app.qobjects.dock, 46
alignak_app.qobjects.dock.buttons, 49
alignak_app.qobjects.dock.dock, 49
alignak_app.qobjects.dock.events, 50
alignak_app.qobjects.dock.livestate, 51
alignak_app.qobjects.dock.password, 46
alignak_app.qobjects.dock.spy, 52
alignak_app.qobjects.dock.status, 46
alignak_app.qobjects.dock.token, 47
alignak_app.qobjects.dock.user, 53
alignak_app.qobjects.dock.user_notes, 48
alignak_app.qobjects.dock.user_options, 48
alignak_app.qobjects.login, 54
alignak_app.qobjects.login.login, 54
alignak_app.qobjects.login.server, 54
alignak_app.qobjects.panel, 54
alignak_app.qobjects.panel.dashboard, 55
alignak_app.qobjects.panel.history, 56
alignak_app.qobjects.panel.host, 56
alignak_app.qobjects.panel.number_services, 57
alignak_app.qobjects.panel.panel, 54
alignak_app.qobjects.panel.problems, 58
alignak_app.qobjects.panel.service, 59
alignak_app.qobjects.panel.service_tree_item, 59
alignak_app.qobjects.panel.services, 59
alignak_app.qobjects.systray, 60
alignak_app.qobjects.systray.qactions_factory, 60
alignak_app.qobjects.systray.tray_icon, 60
alignak_app.qthreads, 61
alignak_app.qthreads.thread, 61
alignak_app.qthreads.threadmanager, 61
alignak_app.utils, 38
alignak_app.utils.config, 38
alignak_app.utils.installer, 39
alignak_app.utils.logs, 41
alignak_app.utils.system, 40
alignak_app.utils.time, 41

A

accept_login() (alignak_app.qobjects.login.login.LoginQDialog
 method), 54
 accept_notes() (alignak_app.qobjects.dock.user_notes.UserNotesQDialog
 method), 48
 AckQDialog (class in alignak_app.qobjects.common.actions), 43
 ActionsQWidget (class in alignak_app.qobjects.common.actions), 43
 add_about_menu() (alignak_app.qobjects.systray.tray_icon.TrayIcon
 method), 60
 add_acknowledge() (alignak_app.qobjects.common.actions.ActionsQWidget
 method), 43
 add_alignak_menu() (alignak_app.qobjects.systray.tray_icon.TrayIcon
 method), 60
 add_daemon_labels() (alignak_app.qobjects.dock.status.StatusQDialog
 method), 47
 add_daemon_titles_labels() (alignak_app.qobjects.dock.status.StatusQDialog
 method), 47
 add_downtime() (alignak_app.qobjects.common.actions.ActionsQWidget
 method), 43
 add_event() (alignak_app.qobjects.dock.events.EventsQWidget
 method), 50
 add_priority_thread() (alignak_app.qthreads.threadmanager.ThreadManager
 method), 61
 add_reload_menu() (alignak_app.qobjects.systray.tray_icon.TrayIcon
 method), 61
 add_spy_host() (alignak_app.qobjects.dock.spy.SpyQWidget
 method), 52
 add_widget() (alignak_app.qobjects.common.frames.AppQFrame
 method), 44
 alignak_app (module), 23
 alignak_app.app (module), 25
 alignak_app.backend (module), 26
 alignak_app.backend.backend (module), 26
 alignak_app.backend.datamanager (module), 28
 alignak_app.items (module), 29
 alignak_app.items.daemon (module), 31
 alignak_app.items.event (module), 32
 alignak_app.items.history (module), 33
 alignak_app.items.host (module), 33
 alignak_app.items.item (module), 29
 alignak_app.items.livesynthesis (module), 34
 alignak_app.items.period (module), 35
 alignak_app.items.realm (module), 36
 alignak_app.items.service (module), 37
 alignak_app.items.user (module), 38
 alignak_app.locales (module), 42
 alignak_app.locales.locales (module), 42
 alignak_app.qobjects (module), 42
 alignak_app.qobjects.app_main (module), 42
 alignak_app.qobjects.common (module), 42
 alignak_app.qobjects.common.actions (module), 42
 alignak_app.qobjects.common.buttons (module), 44
 alignak_app.qobjects.common.frames (module), 44
 alignak_app.qobjects.common.labels (module), 45
 alignak_app.qobjects.common.widgets (module), 45
 alignak_app.qobjects.dock (module), 46
 alignak_app.qobjects.dock.buttons (module), 49
 alignak_app.qobjects.dock.dock (module), 49
 alignak_app.qobjects.dock.events (module), 50
 alignak_app.qobjects.dock.livestate (module), 51
 alignak_app.qobjects.dock.password (module), 46
 alignak_app.qobjects.dock.spy (module), 52
 alignak_app.qobjects.dock.status (module), 46
 alignak_app.qobjects.dock.token (module), 47
 alignak_app.qobjects.dock.user (module), 53
 alignak_app.qobjects.dock.user_notes (module), 48
 alignak_app.qobjects.dock.user_options (module), 48
 alignak_app.qobjects.login (module), 54
 alignak_app.qobjects.login.login (module), 54
 alignak_app.qobjects.login.server (module), 54

`alignak_app.qobjects.panel` (module), 54
`alignak_app.qobjects.panel.dashboard` (module), 55
`alignak_app.qobjects.panel.history` (module), 56
`alignak_app.qobjects.panel.host` (module), 56
`alignak_app.qobjects.panel.number_services` (module), 57
`alignak_app.qobjects.panel.panel` (module), 54
`alignak_app.qobjects.panel.problems` (module), 58
`alignak_app.qobjects.panel.service` (module), 59
`alignak_app.qobjects.panel.service_tree_item` (module), 59
`alignak_app.qobjects.panel.services` (module), 59
`alignak_app.qobjects.systray` (module), 60
`alignak_app.qobjects.systray.qactions_factory` (module), 60
`alignak_app.qobjects.systray.tray_icon` (module), 60
`alignak_app.qthreads` (module), 61
`alignak_app.qthreads.thread` (module), 61
`alignak_app.qthreads.threadmanager` (module), 61
`alignak_app.utils` (module), 38
`alignak_app.utils.config` (module), 38
`alignak_app.utils.installer` (module), 39
`alignak_app.utils.logs` (module), 41
`alignak_app.utils.system` (module), 40
`alignak_app.utils.time` (module), 41
`AlignakApp` (class in `alignak_app.app`), 25
`AppProgressBar` (class in `alignak_app.app`), 25
`AppProgressQWidget` (class in `alignak_app.app`), 26
`AppQFrame` (class in `alignak_app.qobjects.common.frames`), 44
`AppQMainWindow` (class in `alignak_app.qobjects.app_main`), 42

B

`BackendClient` (class in `alignak_app.backend.backend`), 26
`BackendQThread` (class in `alignak_app.qthreads.thread`), 61
`build_menu()` (`alignak_app.qobjects.systray.tray_icon.TrayIcon` method), 61
`ButtonsQWidget` (class in `alignak_app.qobjects.dock.buttons`), 49

C

`center_widget()` (in module `alignak_app.qobjects.common.widgets`), 46
`check_connection()` (`alignak_app.qobjects.systray.tray_icon.TrayIcon` method), 61
`check_environment()` (`alignak_app.utils.installer.Installer` method), 40
`check_install_files()` (`alignak_app.utils.installer.Installer` method), 40

`check_install_folders()` (`alignak_app.utils.installer.Installer` method), 40
`check_installation()` (`alignak_app.utils.installer.Installer` method), 40
`check_threads()` (`alignak_app.app.AlignakApp` static method), 25
`check_user_installation()` (`alignak_app.utils.installer.Installer` method), 40
`clean_threads()` (`alignak_app.qthreads.threadmanager.ThreadManager` method), 61
`close_item()` (`alignak_app.qobjects.dock.events.EventItem` method), 50
`close_widget()` (`alignak_app.qobjects.common.frames.AppQFrame` method), 44
`connect_dock_buttons()` (`alignak_app.qobjects.app_main.AppQMainWindow` method), 42
`create()` (`alignak_app.items.daemon.Daemon` method), 31
`create()` (`alignak_app.items.event.Event` method), 32
`create()` (`alignak_app.items.history.History` method), 33
`create()` (`alignak_app.items.host.Host` method), 34
`create()` (`alignak_app.items.item.Item` method), 30
`create()` (`alignak_app.items.livesynthesis.LiveSynthesis` method), 34
`create()` (`alignak_app.items.period.Period` method), 35
`create()` (`alignak_app.items.realm.Realm` method), 36
`create()` (`alignak_app.items.service.Service` method), 37
`create()` (`alignak_app.items.user.User` method), 38
`create()` (`alignak_app.qobjects.systray.qactions_factory.QActionFactory` method), 60
`create_line_search()` (`alignak_app.qobjects.panel.panel.PanelQWidget` method), 55
`create_logger()` (in module `alignak_app.utils.logs`), 41
`create_quit_action()` (`alignak_app.qobjects.systray.tray_icon.TrayIcon` method), 61
`create_widget()` (`alignak_app.qobjects.login.login.LoginQDialog` method), 54

D

`Daemon` (class in `alignak_app.items.daemon`), 31
`DashboardQWidget` (class in `alignak_app.qobjects.panel.dashboard`), 55
`DataManager` (class in `alignak_app.backend.datamanager`), 28
`display_host()` (`alignak_app.qobjects.panel.panel.PanelQWidget` method), 55
`DockQWidget` (class in `alignak_app.qobjects.dock.dock`), 50
`DownQDialog` (class in `alignak_app.qobjects.common.actions`), 43

[dragEnterEvent\(\)](#) (alignak_app.qobjects.panel.panel.PanelQWidget method), [55](#)
[dragMoveEvent\(\)](#) (alignak_app.qobjects.dock.spy.SpyQListWidget method), [52](#)
[dragMoveEvent\(\)](#) (alignak_app.qobjects.panel.panel.PanelQWidget method), [55](#)
[dropEvent\(\)](#) (alignak_app.qobjects.dock.spy.SpyQListWidget method), [52](#)
[dropEvent\(\)](#) (alignak_app.qobjects.panel.panel.PanelQWidget method), [55](#)
[duration_to_seconds\(\)](#) (alignak_app.qobjects.common.actions.DownQDialog method), [43](#)

E

[edit_setting_value\(\)](#) (alignak_app.utils.config.Settings method), [39](#)
[enable_notifications\(\)](#) (alignak_app.qobjects.dock.user.UserQWidget method), [53](#)
[Event](#) (class in alignak_app.items.event), [32](#)
[EventItem](#) (class in alignak_app.qobjects.dock.events), [50](#)
[EventsQWidget](#) (class in alignak_app.qobjects.dock.events), [50](#)

F

[file_executable\(\)](#) (in module alignak_app.utils.system), [40](#)

G

[get\(\)](#) (alignak_app.backend.backend.BackendClient method), [26](#)
[get_action\(\)](#) (alignak_app.qobjects.systray.qactions_factory.QActionFactory method), [60](#)
[get_actions_widget\(\)](#) (alignak_app.qobjects.panel.host.HostQWidget method), [57](#)
[get_actions_widget\(\)](#) (alignak_app.qobjects.panel.service.ServiceDataQWidget method), [59](#)
[get_all_hostnames\(\)](#) (alignak_app.backend.datamanager.DataManager method), [28](#)
[get_available_icons\(\)](#) (alignak_app.items.host.Host static method), [34](#)
[get_available_icons\(\)](#) (alignak_app.items.service.Service static method), [37](#)
[get_backend_status_icon\(\)](#) (alignak_app.backend.backend.BackendClient method), [26](#)
[get_btn_state\(\)](#) (alignak_app.qobjects.common.buttons.ToggleQWidgetButton method), [44](#)
[get_buttons_widget\(\)](#) (alignak_app.qobjects.dock.status.StatusQDialog method), [47](#)
[get_color_event\(\)](#) (alignak_app.qobjects.dock.events.EventItem static method), [50](#)
[get_config\(\)](#) (alignak_app.utils.config.Settings method), [39](#)
[get_daemons_names\(\)](#) (alignak_app.items.daemon.Daemon static method), [31](#)
[get_daemons_status_icon\(\)](#) (alignak_app.items.daemon.Daemon static method), [31](#)
[get_display_name\(\)](#) (alignak_app.items.host.Host method), [34](#)
[get_display_name\(\)](#) (alignak_app.items.period.Period method), [35](#)
[get_display_name\(\)](#) (alignak_app.items.realm.Realm method), [36](#)
[get_display_name\(\)](#) (alignak_app.items.service.Service method), [37](#)
[get_events\(\)](#) (alignak_app.backend.datamanager.DataManager method), [28](#)
[get_events_widget\(\)](#) (in module alignak_app.qobjects.dock.events), [51](#)
[get_frame_separator\(\)](#) (in module alignak_app.qobjects.common.frames), [45](#)
[get_history_icon_name_from_message\(\)](#) (alignak_app.items.history.History static method), [33](#)
[get_history_widget_model\(\)](#) (alignak_app.qobjects.panel.history.HistoryQWidget static method), [56](#)
[get_host_icon_widget\(\)](#) (alignak_app.qobjects.panel.host.HostQWidget method), [57](#)
[get_host_msg_and_event_type\(\)](#) (in module alignak_app.items.item), [30](#)
[get_host_resume_widget\(\)](#) (alignak_app.qobjects.panel.dashboard.DashboardQWidget method), [56](#)
[get_host_services\(\)](#) (alignak_app.backend.datamanager.DataManager method), [28](#)
[get_host_with_services\(\)](#) (alignak_app.backend.datamanager.DataManager method), [28](#)
[get_hosts_notif_widget\(\)](#) (alignak_app.qobjects.dock.user.UserQWidget method), [53](#)
[get_icon_item\(\)](#) (in module alignak_app.qobjects.common.icons), [44](#)

nak_app.qobjects.common.labels), 45

get_icon_name() (in module alignak_app.items.item), 30

get_icon_name_from_state() (in module alignak_app.items.item), 31

get_icon_pixmap() (in module alignak_app.qobjects.common.labels), 45

get_icon_widget() (alignak_app.qobjects.panel.problems.ProblemsQWidget static method), 58

get_image() (alignak_app.utils.config.Settings method), 39

get_informations_widget() (alignak_app.qobjects.dock.user.UserQWidget method), 53

get_item() (alignak_app.backend.datamanager.DataManager method), 28

get_item_type_widget() (alignak_app.qobjects.dock.livestate.LivestateQWidget method), 52

get_items_and_problems() (alignak_app.backend.datamanager.DataManager method), 28

get_last_check_widget() (alignak_app.qobjects.panel.host.HostQWidget method), 57

get_last_check_widget() (alignak_app.qobjects.panel.service.ServiceDataQWidget method), 59

get_logo_widget() (in module alignak_app.qobjects.common.widgets), 46

get_notes_output_widget() (alignak_app.qobjects.panel.host.HostQWidget method), 57

get_notes_scrollarea() (alignak_app.qobjects.dock.user.UserQWidget method), 53

get_notifications_widget() (alignak_app.qobjects.dock.user.UserQWidget method), 53

get_notifications_widget() (alignak_app.qobjects.dock.user_options.UserOptionsQDialog method), 48

get_period_name() (alignak_app.backend.datamanager.DataManager method), 28

get_problems() (alignak_app.backend.datamanager.DataManager method), 29

get_problems_widget_title() (alignak_app.qobjects.panel.problems.ProblemsQWidget method), 58

get_real_host_state_icon() (in module alignak_app.items.item), 31

get_realm_name() (alignak_app.backend.datamanager.DataManager method), 29

get_request_model() (alignak_app.items.daemon.Daemon static method), 31

get_request_model() (alignak_app.items.event.Event static method), 32

get_request_model() (alignak_app.items.history.History static method), 33

get_request_model() (alignak_app.items.host.Host static method), 34

get_request_model() (alignak_app.items.livesynthesis.LiveSynthesis static method), 35

get_request_model() (alignak_app.items.period.Period static method), 36

get_request_model() (alignak_app.items.realm.Realm static method), 36

get_request_model() (alignak_app.items.service.Service static method), 37

get_request_model() (alignak_app.items.user.User static method), 38

get_role() (alignak_app.items.user.User method), 38

get_search_widget() (alignak_app.qobjects.panel.panel.PanelQWidget method), 55

get_selected_options() (alignak_app.qobjects.dock.user_options.UserOptionsQDialog static method), 49

get_service_icon_widget() (alignak_app.qobjects.panel.service.ServiceDataQWidget method), 59

get_service_states_nb() (alignak_app.items.service.Service static method), 37

get_services_notif_widget() (alignak_app.qobjects.dock.user.UserQWidget method), 53

get_services_resume_widget() (alignak_app.qobjects.panel.dashboard.DashboardQWidget method), 56

get_synthesis_count() (alignak_app.items.daemon.Daemon static method), 32

get_synthesis_count() (alignak_app.backend.datamanager.DataManager method), 29

get_synthesis_count_model() (alignak_app.items.livesynthesis.LiveSynthesis static method), 35

get_synthesis_widget() (alignak_app.qobjects.panel.panel.PanelQWidget method), 55

get_table_items_problem() (alignak_app.qobjects.panel.problems.ProblemsQWidget method), 58

- [get_threads_to_launch\(\)](#) (alignak_app.qthreads.threadmanager.ThreadManager method), 61
[get_time_diff_since_last_timestamp\(\)](#) (in module alignak_app.utils.time), 41
[get_title_widget\(\)](#) (alignak_app.qobjects.common.frames.AppQFrame method), 44
[get_token_widget\(\)](#) (alignak_app.qobjects.dock.token.TokenQDialog method), 48
[get_tooltip\(\)](#) (alignak_app.items.daemon.Daemon method), 32
[get_tooltip\(\)](#) (alignak_app.items.event.Event method), 32
[get_tooltip\(\)](#) (alignak_app.items.history.History method), 33
[get_tooltip\(\)](#) (alignak_app.items.host.Host method), 34
[get_tooltip\(\)](#) (alignak_app.items.item.Item method), 30
[get_tooltip\(\)](#) (alignak_app.items.livesynthesis.LiveSynthesis method), 35
[get_tooltip\(\)](#) (alignak_app.items.period.Period method), 36
[get_tooltip\(\)](#) (alignak_app.items.realm.Realm method), 36
[get_tooltip\(\)](#) (alignak_app.items.service.Service method), 37
[get_tooltip\(\)](#) (alignak_app.items.user.User method), 38
[get_url_endpoint_from_icon_name\(\)](#) (in module alignak_app.utils.config), 39
[get_user_notes_widget\(\)](#) (alignak_app.qobjects.dock.user_notes.UserNotesQDialog method), 48
[get_variables_widget\(\)](#) (alignak_app.qobjects.panel.host.HostQWidget method), 57
- ## H
- [handle_accept\(\)](#) (alignak_app.qobjects.common.actions.DownQDialog method), 43
[handle_confirm\(\)](#) (alignak_app.qobjects.dock.password.PasswordQDialog method), 46
[handle_server\(\)](#) (alignak_app.qobjects.login.login.LoginQDialog static method), 54
[History](#) (class in alignak_app.items.history), 33
[HistoryQWidget](#) (class in alignak_app.qobjects.panel.history), 56
[Host](#) (class in alignak_app.items.host), 34
[HostQWidget](#) (class in alignak_app.qobjects.panel.host), 56
- ## I
- [init_config\(\)](#) (alignak_app.utils.config.Settings method), 39
[init_css\(\)](#) (alignak_app.utils.config.Settings method), 39
[init_environment\(\)](#) (alignak_app.utils.installer.Installer method), 40
[init_event_widget\(\)](#) (in module alignak_app.qobjects.dock.events), 51
[init_localization\(\)](#) (in module alignak_app.locales.locales), 42
[initialize\(\)](#) (alignak_app.app.AppProgressQWidget method), 26
[initialize\(\)](#) (alignak_app.qobjects.app_main.AppQMainWindow method), 42
[initialize\(\)](#) (alignak_app.qobjects.common.actions.AckQDialog method), 43
[initialize\(\)](#) (alignak_app.qobjects.common.actions.ActionsQWidget method), 43
[initialize\(\)](#) (alignak_app.qobjects.common.actions.DownQDialog method), 43
[initialize\(\)](#) (alignak_app.qobjects.common.buttons.ToggleQWidgetButton method), 44
[initialize\(\)](#) (alignak_app.qobjects.common.frames.AppQFrame method), 44
[initialize\(\)](#) (alignak_app.qobjects.common.widgets.LogoQWidget method), 45
[initialize\(\)](#) (alignak_app.qobjects.dock.buttons.ButtonsQWidget method), 49
[initialize\(\)](#) (alignak_app.qobjects.dock.dock.DockQWidget method), 50
[initialize\(\)](#) (alignak_app.qobjects.dock.events.EventItem method), 50
[initialize\(\)](#) (alignak_app.qobjects.dock.events.EventsQWidget method), 51
[initialize\(\)](#) (alignak_app.qobjects.dock.livestate.LivestateQWidget method), 52
[initialize\(\)](#) (alignak_app.qobjects.dock.password.PasswordQDialog method), 46
[initialize\(\)](#) (alignak_app.qobjects.dock.spy.SpyQWidget method), 52
[initialize\(\)](#) (alignak_app.qobjects.dock.status.StatusQDialog method), 47
[initialize\(\)](#) (alignak_app.qobjects.dock.status.StatusQWidget method), 47
[initialize\(\)](#) (alignak_app.qobjects.dock.token.TokenQDialog method), 48
[initialize\(\)](#) (alignak_app.qobjects.dock.user.UserQWidget method), 53
[initialize\(\)](#) (alignak_app.qobjects.dock.user_notes.UserNotesQDialog method), 48
[initialize\(\)](#) (alignak_app.qobjects.dock.user_options.UserOptionsQDialog method), 49
[initialize\(\)](#) (alignak_app.qobjects.panel.dashboard.DashboardQWidget method), 56
[initialize\(\)](#) (alignak_app.qobjects.panel.history.HistoryQWidget method), 56
[initialize\(\)](#) (alignak_app.qobjects.panel.host.HostQWidget method), 57

method), 57
 initialize() (alignak_app.qobjects.panel.number_services.NumberServicesQWidget (alignak_app.qobjects.common.actions.DownQDialog method), 58
 initialize() (alignak_app.qobjects.panel.panel.PanelQWidget method), 55
 initialize() (alignak_app.qobjects.panel.problems.ProblemsQWidget method), 58
 initialize() (alignak_app.qobjects.panel.service.ServiceDataQWidget method), 59
 initialize() (alignak_app.qobjects.panel.service_tree_item.ServiceTreeItem method), 59
 initialize() (alignak_app.qobjects.panel.services.ServicesQWidget method), 60
 initialize_dialog() (alignak_app.qobjects.login.server.ServerQDialog method), 54
 install() (alignak_app.utils.installer.Installer method), 40
 install_file() (in module alignak_app.utils.system), 40
 Installer (class in alignak_app.utils.installer), 40
 is_ready() (alignak_app.backend.datamanager.DataManager method), 29
 Item (class in alignak_app.items.item), 30

L

launch_threads() (alignak_app.qthreads.threadmanager.ThreadManager method), 62
 LivestateQWidget (class in alignak_app.qobjects.dock.livestate), 52
 LiveSynthesis (class in alignak_app.items.livesynthesis), 34
 login() (alignak_app.backend.backend.BackendClient method), 26
 LoginQDialog (class in alignak_app.qobjects.login.login), 54
 LogoQWidget (class in alignak_app.qobjects.common.widgets), 45

M

minimize() (alignak_app.qobjects.common.frames.AppQFrame method), 44
 minimize() (alignak_app.qobjects.common.widgets.LogoQWidget method), 46
 minimize_maximize() (alignak_app.qobjects.common.frames.AppQFrame method), 45
 minimize_maximize() (alignak_app.qobjects.common.widgets.LogoQWidget method), 46
 mkdir() (in module alignak_app.utils.system), 40
 mouseMoveEvent() (alignak_app.qobjects.app_main.AppQMainWindow method), 42
 mouseMoveEvent() (alignak_app.qobjects.common.actions.AckQDialog

method), 43
 mouseMoveEvent() (alignak_app.qobjects.common.frames.AppQFrame method), 45
 mouseMoveEvent() (alignak_app.qobjects.login.login.LoginQDialog method), 54
 mouseMoveEvent() (alignak_app.qobjects.login.server.ServerQDialog method), 54
 mousePressEvent() (alignak_app.qobjects.app_main.AppQMainWindow method), 42
 mousePressEvent() (alignak_app.qobjects.common.actions.AckQDialog method), 43
 mousePressEvent() (alignak_app.qobjects.common.actions.DownQDialog method), 44
 mousePressEvent() (alignak_app.qobjects.common.frames.AppQFrame method), 45
 mousePressEvent() (alignak_app.qobjects.login.login.LoginQDialog method), 54
 mousePressEvent() (alignak_app.qobjects.login.server.ServerQDialog method), 54

N

NumberServicesQWidget (class in alignak_app.qobjects.panel.number_services), 58

O

open_host_widget() (alignak_app.qobjects.app_main.AppQMainWindow method), 42
 open_item_type_url() (alignak_app.qobjects.panel.dashboard.DashboardQWidget method), 56
 open_problems_widget() (alignak_app.qobjects.app_main.AppQMainWindow method), 42
 open_url() (in module alignak_app.utils.config), 39
 open_user_widget() (alignak_app.qobjects.dock.buttons.ButtonsQWidget method), 49

P

paintEvent() (alignak_app.qobjects.common.widgets.LogoQWidget

- method), 46
- paintEvent() (alignak_app.qobjects.dock.livestate.LivestateQWidget method), 52
- paintEvent() (alignak_app.qobjects.dock.status.StatusQWidget method), 47
- PanelQWidget (class in alignak_app.qobjects.panel.panel), 54
- PasswordQDialog (class in alignak_app.qobjects.dock.password), 46
- patch() (alignak_app.backend.backend.BackendClient method), 27
- patch_data() (alignak_app.qobjects.dock.user.UserQWidget method), 53
- patch_host_checks() (alignak_app.qobjects.panel.host.HostQWidget method), 57
- Period (class in alignak_app.items.period), 35
- periodic_refresh() (alignak_app.qobjects.panel.service.ServiceDataQWidget method), 59
- post() (alignak_app.backend.backend.BackendClient method), 27
- ProblemsQWidget (class in alignak_app.qobjects.panel.problems), 58
- query_daemons_data() (alignak_app.backend.backend.BackendClient method), 27
- query_history_data() (alignak_app.backend.backend.BackendClient method), 27
- query_hosts_data() (alignak_app.backend.backend.BackendClient method), 27
- query_livesynthesis_data() (alignak_app.backend.backend.BackendClient method), 27
- query_notifications_data() (alignak_app.backend.backend.BackendClient method), 27
- query_period_data() (alignak_app.backend.backend.BackendClient method), 27
- query_realms_data() (alignak_app.backend.backend.BackendClient method), 27
- query_services_data() (alignak_app.backend.backend.BackendClient method), 27
- query_user_data() (alignak_app.backend.backend.BackendClient method), 27
- quit_app() (alignak_app.qobjects.systray.tray_icon.TrayIcon static method), 61
- quit_launched_threads() (alignak_app.app.AlignakApp static method), 25
- ## R
- read_config_file() (in module alignak_app.utils.system), 41
- Realm (class in alignak_app.items.realm), 36
- reload_configuration() (alignak_app.qobjects.systray.tray_icon.TrayIcon static method), 61
- remove_event() (alignak_app.qobjects.dock.events.EventsQWidget method), 51
- remove_event() (alignak_app.qobjects.dock.spy.SpyQWidget method), 53
- remove_timer_event() (alignak_app.qobjects.dock.events.EventsQWidget method), 51
- run() (alignak_app.qthreads.thread.BackendQThread method), 61
- ## S
- send_datamanager_events() (alignak_app.qobjects.dock.events.EventsQWidget method), 51
- send_event() (in module alignak_app.qobjects.dock.events), 51
- send_spy_events() (alignak_app.qobjects.dock.dock.DockQWidget method), 50
- ServerQDialog (class in alignak_app.qobjects.login.server), 54
- Service (class in alignak_app.items.service), 37
- ServiceDataQWidget (class in alignak_app.qobjects.panel.service), 59
- ServicesQWidget (class in alignak_app.qobjects.panel.services), 60
- ServiceTreeWidgetItem (class in alignak_app.qobjects.panel.service_tree_item), 59
- set_daemons_labels() (alignak_app.qobjects.dock.status.StatusQDialog method), 47
- set_data() (alignak_app.qobjects.panel.host.HostQWidget method), 57
- set_data() (alignak_app.qobjects.panel.services.ServicesQWidget method), 60
- set_text() (alignak_app.app.AppProgressBar method), 25
- Settings (class in alignak_app.utils.config), 39

[show_history\(\)](#) (alignak_app.qobjects.panel.host.HostQWidget method), 57
[show_options_dialog\(\)](#) (in module alignak_app.qobjects.dock.user_options), 49
[show_status_dialog\(\)](#) (alignak_app.qobjects.dock.status.StatusQWidget method), 47
[show_token_dialog\(\)](#) (alignak_app.qobjects.dock.user.UserQWidget static method), 53
[show_widget\(\)](#) (alignak_app.qobjects.common.frames.AppQWidget method), 45
[showEvent\(\)](#) (alignak_app.qobjects.login.login.LoginQDialog method), 54
[spy_host\(\)](#) (alignak_app.qobjects.panel.panel.PanelQWidget method), 55
[SpyQListWidget](#) (class in alignak_app.qobjects.dock.spy), 52
[SpyQWidget](#) (class in alignak_app.qobjects.dock.spy), 52
[start\(\)](#) (alignak_app.app.AlignakApp method), 25
[StatusQDialog](#) (class in alignak_app.qobjects.dock.status), 47
[StatusQWidget](#) (class in alignak_app.qobjects.dock.status), 47
[stop_priority_threads\(\)](#) (alignak_app.qthreads.threadmanager.ThreadManager method), 62
[stop_threads\(\)](#) (alignak_app.qthreads.threadmanager.ThreadManager method), 62
T
[text\(\)](#) (alignak_app.app.AppProgressBar method), 26
[ThreadManager](#) (class in alignak_app.qthreads.threadmanager), 61
[ToggleQWidgetButton](#) (class in alignak_app.qobjects.common.buttons), 44
[TokenQDialog](#) (class in alignak_app.qobjects.dock.token), 48
[TrayIcon](#) (class in alignak_app.qobjects.systray.tray_icon), 60
U
[update_btn_state\(\)](#) (alignak_app.qobjects.common.buttons.ToggleQWidgetButton method), 44
[update_dashboard\(\)](#) (alignak_app.qobjects.panel.dashboard.DashboardQWidget method), 56
[update_data\(\)](#) (alignak_app.items.daemon.Daemon method), 32
[update_data\(\)](#) (alignak_app.items.event.Event method), 32
[update_data\(\)](#) (alignak_app.items.history.History method), 33
[update_data\(\)](#) (alignak_app.items.host.Host method), 34
[update_data\(\)](#) (alignak_app.items.item.Item method), 30
[update_data\(\)](#) (alignak_app.items.livesynthesis.LiveSynthesis method), 35
[update_data\(\)](#) (alignak_app.items.period.Period method), 36
[update_data\(\)](#) (alignak_app.items.realm.Realm method), 36
[update_data\(\)](#) (alignak_app.items.service.Service method), 37
[update_data\(\)](#) (alignak_app.items.user.User method), 38
[update_database\(\)](#) (alignak_app.backend.datamanager.DataManager method), 29
[update_dialog\(\)](#) (alignak_app.qobjects.dock.status.StatusQDialog method), 47
[update_host\(\)](#) (alignak_app.qobjects.panel.host.HostQWidget method), 57
[update_item\(\)](#) (alignak_app.qobjects.panel.service_tree_item.ServiceTreeItem method), 59
[update_item_data\(\)](#) (alignak_app.backend.datamanager.DataManager method), 29
[update_labels\(\)](#) (alignak_app.qobjects.dock.livestate.LivestateQWidget method), 52
[update_problems_data\(\)](#) (alignak_app.qobjects.panel.problems.ProblemsQWidget method), 59
[update_service_data\(\)](#) (alignak_app.qobjects.panel.services.ServicesQWidget method), 60
[update_status\(\)](#) (alignak_app.qobjects.dock.status.StatusQWidget method), 47
[update_widget\(\)](#) (alignak_app.qobjects.common.actions.ActionsQWidget method), 43
[update_widget\(\)](#) (alignak_app.qobjects.dock.buttons.ButtonsQWidget method), 49
[update_widget\(\)](#) (alignak_app.qobjects.dock.user.UserQWidget method), 53
[update_widget\(\)](#) (alignak_app.qobjects.panel.number_services.NumberServicesQWidget method), 58
[update_widget\(\)](#) (alignak_app.qobjects.panel.service.ServiceDataQWidget method), 59
[update_widget\(\)](#) (alignak_app.qobjects.panel.services.ServicesQWidget method), 60
[User](#) (class in alignak_app.items.user), 38
[UserNotesQDialog](#) (class in alignak_app.qobjects.dock.user_notes), 48
[UserOptionsQDialog](#) (class in alignak_app.qobjects.dock.user_options), 48
[UserQWidget](#) (class in alignak_app.qobjects.dock.user), 53

W

`write_file()` (in module `alignak_app.utils.system`), [41](#)

`write_rc_file()` (in module `alignak_app.utils.system`), [41](#)