
Alignak-App Documentation

Release 1.4

Matthieu Estrada

Mar 19, 2018

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
2.5	Windows Setup	7
2.5.1	Requirements	7
2.5.1.1	Python and requirements	7
2.5.1.2	Pyinstaller	7
2.5.1.3	Inno Setup	7
2.5.2	Create Setup	8
3	Configuration	9
3.1	Environment Variables	9
3.1.1	On Linux	9
3.1.2	On Windows	10
3.2	Settings File	10
3.3	Proxy Settings	10
3.4	Connection to Backend	10
3.4.1	File Example	11
4	Use Alignak-App	15
4.1	Run “App”	15
4.1.1	On Linux	15
4.1.1.1	First launch	15

4.1.1.2	Launch daemon	15
4.1.2	On Windows	16
4.1.2.1	With Installer	16
4.1.2.2	From command line	16
4.2	Interface	16
4.2.1	Login Window	16
4.2.2	Start	16
4.2.3	App Dock (Right part)	17
4.2.3.1	Alignak part	17
4.2.3.2	Livestate part	17
4.2.3.3	Events part	17
4.2.3.4	Spy part	17
4.2.4	App Panel (Left part)	17
4.2.4.1	Host Synthesis	18
4.2.4.2	Problems	18
4.2.5	WebUI Integration	18
5	Architecture	19
5.1	Resume	19
5.2	Architecture	19
6	API	21
6.1	Main Class	21
6.1.1	App	21
6.2	Backend	22
6.2.1	Backend	22
6.2.2	Data Manager	24
6.3	Items	25
6.3.1	Item	26
6.3.2	Daemon	27
6.3.3	Event	28
6.3.4	History	29
6.3.5	Host	30
6.3.6	Live Synthesis	31
6.3.7	Period	32
6.3.8	Realm	32
6.3.9	Service	33
6.3.10	User	34
6.4	Utils	35
6.4.1	Settings	35
6.4.2	Installer	36
6.4.3	System	37
6.4.4	Logs	38
6.4.5	Time	38
6.5	Locales	39
6.5.1	Locales	39
6.6	QObjects	39
6.6.1	App Main	39
6.6.2	Dock	39
6.6.3	Panel	40
6.6.4	Alignak	41
6.6.4.1	Alignak	41
6.6.4.2	Dashboard	41
6.6.4.3	Livestate	42

6.6.4.4	Problems	42
6.6.4.5	Problems Table	43
6.6.4.6	Status	44
6.6.5	Common	44
6.6.5.1	Actions	44
6.6.5.2	Buttons	46
6.6.5.3	Dialogs	46
6.6.5.4	Frames	47
6.6.5.5	Labels	48
6.6.5.6	Widgets	49
6.6.6	Events	49
6.6.6.1	Events	50
6.6.6.2	Event Item	51
6.6.6.3	Spy	51
6.6.7	Host	52
6.6.7.1	History	52
6.6.7.2	Host	53
6.6.8	Login	54
6.6.8.1	Login	54
6.6.8.2	Server	55
6.6.9	Service	55
6.6.9.1	Service	55
6.6.9.2	Services	56
6.6.9.3	Number Services	56
6.6.9.4	Service Tree Item	57
6.6.10	Systray	57
6.6.10.1	App TrayIcon	57
6.6.11	Threads	58
6.6.11.1	Backend Thread	58
6.6.11.2	Thread Manager	58
6.6.12	User	59
6.6.12.1	User Options	59
6.6.12.2	Password	59
6.6.12.3	Profile	60
6.7	About	61
6.7.1	About	61

7 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 [Alignak-app Installers](#) 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 *Use 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
git+git://github.com/algorys/alignak-backend-client.git@develop
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.

2.5 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.

2.5.1 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.

2.5.1.1 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*.

2.5.1.2 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.

2.5.1.3 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.

2.5.2 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.

3.1 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.

3.1.1 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/
```

3.1.2 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\
```

3.2 Settings File

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 ! “

3.3 Proxy Settings

If you have a proxy on your network, you can set this proxy in environment variables. [Alignak backend client](#) will manage these variables alone when connecting to the backend.

- For HTTP proxies, set HTTP_PROXY variable.
- For HTTPS proxies, set HTTPS_PROXY variable.

For more informations, please read [Requests documentation](#).

3.4 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](#)

3.4.1 File Example

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 =

; 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
```

(continues on next page)

(continued from previous page)

```

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

; Proxy address
; -----
; -----
; Define proxy address for requests
; Format is: http(s)://proxy_address:port
; Default is empty
proxy =

; Proxy user
; -----
; -----
; Define corresponding proxy user. Only needed if you have a proxy.
; Default is empty
proxy_user =

; Proxy password
; -----
; -----
; Define corresponding proxy password. Only needed if you have a proxy.
; If "proxy_user" is set, a proxy login window will be automatically_
; displayed.
; ! Setting password is less secure !
; Default is empty
proxy_password =

[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

; Problems at start
; -----
; -----
; Defines if "Problems" tab is displayed by default at start
; Default is no. Set to "yes" to display problems by default.
problems = no

; Requests interval

```

(continues on next page)

(continued from previous page)

```

; -----
; -----
; 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
; -----
; -----
; 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 30
spy_interval = 30

; 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

```

(continues on next page)

(continued from previous page)

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

; Update panel Service
; -----
↪-----
; 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
```

4.1 Run “App”

4.1.1 On Linux

4.1.1.1 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 [Configuration](#)).

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

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

4.1.1.2 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.

4.1.2 On Windows

4.1.2.1 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\`.

4.1.2.2 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.

4.2 Interface

4.2.1 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.

4.2.2 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.

4.2.3 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.

4.2.3.1 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.

4.2.3.2 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.

4.2.3.3 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.

4.2.3.4 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.

4.2.4 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.

4.2.4.1 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).

4.2.4.2 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.

4.2.5 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.

5.1 Resume

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.

5.2 Architecture

Alignak-app works as follows:

- 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.
- The *Login* package manage user login if needed.

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**)

6.1 Main Class

6.1.1 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*

static show_login_window()

Show LoginQDialog window for user to login to backend

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

6.2 Backend

The Backend package contains classes to manage alignak data

6.2.1 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, proxies=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
- **proxies** (*dict*) – dictionary for proxy
- **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

6.2.2 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 corresponding to host ID

Parameters **host_id** (*str*) – ‘_id’ of host

Returns services corresponding to host ID

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** – type of item (host, service, ...)
- **item_id** (*str*) – ‘_id’ of item to update
- **data** (*dict*) – the data to be updated

6.3 Items

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

6.3.1 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

static **get_check_text** ()

Return text for check type :param check_type: type of check (active_checks_enabled or passive_checks_enabled) :type check_type: str :return: the corresponding text :rtype: 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

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*

6.3.2 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_check_text()`

Return text for check type :param check_type: type of check (active_checks_enabled or passive_checks_enabled) :type check_type: str :return: the corresponding text :rtype: str

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

6.3.3 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_check_text()`

Return text for check type :param `check_type`: type of check (`active_checks_enabled` or `passive_checks_enabled`):type `check_type`: str :return: the corresponding text :rtype: str

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

6.3.4 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_check_text()
 Return text for check type :param check_type: type of check (active_checks_enabled or passive_checks_enabled) :type check_type: str :return: the corresponding text :rtype: str

static get_history_icon_name(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

6.3.5 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`

static get_check_text()
Return text for check type :param check_type: type of check (active_checks_enabled or passive_checks_enabled) :type check_type: str :return: the corresponding text :rtype: str

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

6.3.6 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_check_text ()
Return text for check type :param check_type: type of check (active_checks_enabled or passive_checks_enabled) :type check_type: str :return: the corresponding text :rtype: str

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

6.3.7 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

static `get_check_text()`

Return text for check type :param *check_type*: type of check (`active_checks_enabled` or `passive_checks_enabled`) :type *check_type*: *str* :return: the corresponding text :rtype: *str*

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

6.3.8 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

static get_check_text()

Return text for check type :param check_type: type of check (active_checks_enabled or passive_checks_enabled) :type check_type: str :return: the corresponding text :rtype: str

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

6.3.9 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`

static `get_check_text()`

Return text for check type :param check_type: type of check (active_checks_enabled or passive_checks_enabled) :type check_type: str :return: the corresponding text :rtype: str

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

6.3.10 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_check_text()`

Return text for check type :param check_type: type of check (active_checks_enabled or passive_checks_enabled) :type check_type: str :return: the corresponding text :rtype: str

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

6.4 Utils

The Utils package contains classes and functions configurations

6.4.1 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

6.4.2 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

6.4.3 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

6.4.4 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`

6.4.5 Time

Time manage time and date for Alignak-app

`alignak_app.utils.time.get_current_time()`

Return current formatted time

Returns current time

Return type `str`

`alignak_app.utils.time.get_local_datetime(_date)`

Return the local time from a datetime

Parameters **_date** (`datetime.datetime`) – datetime to set to local time

Returns local date time

Return type `datetime.datetime`

`alignak_app.utils.time.get_local_datetime_from_date(_date)`

Return the local timestamp from date

Parameters **_date** (*str*) – date to convert to datetime

Returns corresponding local datetime of date

Return type `datetime.datetime`

`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`

6.5 Locales

The Locales package contains classes to manage translation.

6.5.1 Locales

Locales manage localization of Alignak-app

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

Application localization

Returns gettext translator method

Return type method

6.6 QObjects

The QObjects package contains PyQt5 classes of Alignak-app

6.6.1 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

```
initialize ()
```

Initialize QMainWindow for App

```
mousePressEvent (event)
```

QWidget.mousePressEvent(QMouseEvent)

```
mousePressEvent (event)
```

QWidget.mousePressEvent(QMouseEvent)

6.6.2 Dock

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

```
class alignak_app.qobjects.dock.DockQWidget (parent=None)
```

Bases: PyQt5.QtWidgets.QWidget

Class who create QWidget for dock

```
initialize ()
```

Initialize dock QWidget

6.6.3 Panel

Panel manage creation of QWidget for Panel (Left part)

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

Class who manage Panel with Host and Services QWidget

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

Parameters **hostnames_list** (*list*) – list of host names

```
define_hostname ()  
    Define hostname to display, depends of sender object
```

Returns the hostname to display

Return type *str*

```
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 ()  
    Create the QWidget with its items and layout.
```

```
spy_host ()  
    Spy host who is available in line_search QLineEdit
```

```
update_panel_spytab ()  
    Update text of the psy panel tab
```

6.6.4 Alignak

The Alignak package contains QObjects to display global alignak data, including `alignakdaemon` and `livesynthesis` endpoints

6.6.4.1 Alignak

Alignak manage creation of QWidgets for general Alignak data, like:

- **Daemons status:** status of each daemons (see `StatusQDialog` class)
- **Backend connection:** status of backend connection (see `TrayIcon` class)
- **User:** data of current user (see `User` package)

class `alignak_app.qobjects.alignak.alignak.AlignakQWidget`

Bases: `PyQt5.QtWidgets.QWidget`

Class who display daemons status, backend connection and user informations

initialize()

Initialize QWidget

open_user_widget()

Show ProfileQWidget

show_status_dialog()

Update and show StatusQDialog

update_status()

Update daemons and backend status

6.6.4.2 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.alignak.dashboard.DashboardQWidget` (*parent=None*)

Bases: `PyQt5.QtWidgets.QWidget`

Class who manage Host and Services resume QWidgets 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

6.6.4.3 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.alignak.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

6.6.4.4 Problems

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

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

Class who create Problems QWidget

add_spied_host ()

Add a host to spied hosts

get_btn_widget ()

Return QWidget with spy and host synthesis QPushButton

Returns widget with spy and host button

Return type QWidget

get_curent_user_role_item()
 Return current selected item by `Qt.UserRole`

Returns current selected item or None

Return type *alignak_app.items.item.Item*

get_search_widget()
 Create and return the search QWidget

Returns search QWidget

Return type QWidget

initialize(spy_widget)
 Initialize QWidget and set SpyQWidget

Parameters **spy_widget** (*alignak_app.qobjects.events.spy.SpyQWidget*) – instance of SpyQWidget to manage spy events

update_action_buttons()
 Update status of action buttons and set current item for ActionsQWidget

update_problems_data(item_type=)
 Update data of Problems QTableWidget and problems title

Parameters **item_type** (*str*) – type of item to filter

6.6.4.5 Problems Table

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

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

class *alignak_app.qobjects.alignak.problems_table.ProblemsQTableView* (*parent=None*)
 Bases: *PyQt5.QtWidgets.QTableView*

Class who create Problems QTableView to display each problem

static get_item_text()
 Return item text depends if it's a host or service

Parameters **item** (*alignak_app.items.host.Host | alignak_app.items.service.Service*) – host or service item

Returns text of item

Return type *str*

static get_output_tableitem()
 Return centered QTableWidgetItem with output

Parameters **item** (*alignak_app.items.host.Host | alignak_app.items.service.Service*) – host or service item

Returns table item with text

Return type *QStandardItem*

get_tableitem(item)
 Return centered QTableWidgetItem with problem

Parameters *item* (*alignak_app.items.host.Host* | *alignak_app.items.service.Service*) – host or service item

Returns table item with text

Return type *QStandardItem*

update_view (*problems_data*)

Update *QTableView* model and proxy filter

Parameters *problems_data* (*dict*) – problems found in database

Returns proxy filter to connect with line edit

Return type *QSortFilterProxyModel*

6.6.4.6 Status

Status manage creation of *QDialog* for daemons status

class *alignak_app.qobjects.alignak.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

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

6.6.5 Common

The Common package contains common *QObjects* for Alignak-app, used by several other *QWidgets*

6.6.5.1 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, with ack and downtime buttons

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)

6.6.5.2 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.

```
initialize()
```

Initialize QWidget

```
is_checked()
```

Return “toggle_btn” state

Returns QPushButton state

Return type bool

```
update_btn_state(state)
```

Update QPushButton state, ObjectName and StyleSheet

Parameters *state* (bool) – current state of QPushButton

6.6.5.3 Dialogs

Dialogs manage global QDialogs

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

Class who create Edit QDialog to edit text in Alignak-app

```
accept_text()
```

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

```
get_text_widget()
```

Return text QWidget with QTextEdit

Returns text QWidget

Return type QWidget

```
initialize(title, text)
```

Initialize QDialog for UserNotesQDialog

Parameters

- **title** (*str*) – title of the QDialog
- **text** (*str*) – text to edit

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

Class who create a Message QDialog to display texts in Alignak-app

```
get_message_widget(dialog, title, text)
```

Return colored message QWidget

Parameters

- **dialog** (*str*) – type of dialog (‘text’ or ‘error’)
- **title** (*str*) – title of text to display

- **text** (*str*) – text to display

Returns message QWidget

Return type QWidget

initialize (*widgettitle, dialog, title, text*)

Initialize QDialog for PasswordDialog

Parameters

- **widgettitle** (*str*) – title of the QDialog
- **dialog** (*str*) – type of dialog ('text' or 'error')
- **title** (*str*) – title of text to display
- **text** (*str*) – text to display

class alignak_app.qobjects.common.dialogs.**ValidatorQDialog** (*parent=None*)

Bases: PyQt5.QtWidgets.QDialog

Class who create Validator QDialog to edit text in Alignak-app with regexp to validate

accept_text ()

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

check_text ()

Valid email with QRegExpValidator and inform user

get_text_widget (*regexp*)

Return text QWidget with QTextEdit

Returns text QWidget

Return type QWidget

initialize (*title, text, regexp*)

Initialize QDialog for ValidatorQDialog

Parameters

- **title** (*str*) – title of the QDialog
- **text** (*str*) – text to edit
- **regexp** (*str*) – regular expression to validate

6.6.5.4 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 QWidget

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

6.6.5.5 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

6.6.5.6 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

6.6.6 Events

The Events package contains QObjects to view events collected by the application, like notifications in history, hosts spied, actions performed by the user

6.6.6.1 Events

Events manage creation of QWidget for events

class alignak_app.qobjects.events.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** (*None* | *str*) – data of host. Only necessary if “be_spied” is True

event_exist (*msg*)

Check if event already displayed, move it to top and update tooltip. Only for EventItem who are spied on.

Parameters **msg** (*str*) – message of event

Returns if message exist or not in events QWidgetList

Return type *bool*

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.events.events.get_events_widget ()

Return EventsQWidget instance

Returns EventsQWidget instance

Return type *alignak_app.qobjects.events.events.EventsQWidget*

alignak_app.qobjects.events.events.init_event_widget ()

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

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

Add event to Events QWidget (access function to make event sendable in whole application. Event type define icon and color of *EventItem*.

- **green**: [OK, UP]
- **blue**: [UNKNOWN, INFO, SPY]

- **orange**: [WARNING, UNREACHABLE, WARN]
- **red**: [DOWN, CRITICAL, ALERT]
- **yellow**: [ACK]
- **yellow**: [DOWNTIME]

Parameters

- **event_type** (*str*) – type of event to send
- **msg** (*str*) – message of event
- **timer** – timer to hide event at end of time
- **spied_on** (*bool*) – make event spy able
- **host** (*None* | *str*) – _id of host. Only necessary if “spied_on” is True

6.6.6.2 Event Item

Event item manage creation of QListWidgetItem for events

class alignak_app.qobjects.events.item.**EventItem**

Bases: PyQt5.QtWidgets.QListWidgetItem

Class who create an event QListWidgetItem

close_item()

Hide items when timer is finished

static get_icon()

Return name of icon event

Parameters **event_type** (*str*) – type of event

Returns name of icon

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** (*None* | *str*) – _id of host. Only necessary if “be_spied” is True

6.6.6.3 Spy

Spy manage creation of QWidget for host items who are spied

class alignak_app.qobjects.events.spy.**SpyQWidget**

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.events.spy.**SpyQWidget**

Bases: PyQt5.QtWidgets.QWidget

Class who create QWidget for spied hosts

initialize ()

Intialize Spy QWidget

remove_event ()

Remove item when user double click on an item, update parent tab text

send_spy_events ()

Send event for one host spied

update_parent_spytab ()

Update the parent spy tab text

6.6.7 Host

The Host package contains QObjects to display data of the backend `host`

6.6.7.1 History

History manage creation of QWidget to display history of a host

class alignak_app.qobjects.host.history.**HistoryQWidget** (*parent=None*)

Bases: PyQt5.QtWidgets.QWidget

Class who create the History QWidget for host

static **get_event_type** (*hostname*)

Return event type for history

Parameters

- **event** (*dict*) – event of history
- **hostname** (*str*) – name of host attached to event

Returns the event type

Return type *str*

get_event_widget (*hostname, event*)

Return event QWidget with icon, event text and event message

Parameters

- **hostname** (*str*) – name of host attached to event
- **event** (*dict*) – data of an event

Returns widget of event

Return type QWidget

static get_icon_label ()

Return QWidget with corresponding icon to item state

Parameters **event** (*dict*) – data of an event

Returns icon QWidget

Return type QWidget

initialize ()

Initialize History QWidget

update_history_data (*hostname, host_history*)

Update data of history QTableWidgetItem

Parameters

- **hostname** (*str*) – name of the host
- **host_history** (*History*) – history of host

6.6.7.2 Host

Host manage creation of QWidget to display host data

class alignak_app.qobjects.host.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_data()

Display QDialog for patch

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

6.6.8 Login

The Login package contains QObjects to display connection windows, for login and server settings

6.6.8.1 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

handle_proxy()

Handle Proxy QDialog display and set proxies for login

```

static handle_server ()
    Handle for Server QDialog and set alignak backend server settings

mousePressEvent (event)
    QWidget.mousePressEvent(QMouseEvent)

mousePressEvent (event)
    QWidget.mousePressEvent(QMouseEvent)

set_proxy_settings (proxy_password=None)
    Set the proxy settings, with password if given

    Parameters proxy_password (str) – the password of proxy

showEvent (_)
    QDialog.showEvent(QShowEvent)

```

6.6.8.2 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.

    get_settings_widget ()
        Return the alignak settings QWidget

        Returns settings QWidget

        Return type QWidget

    initialize_dialog ()
        Initialize Server QDialog

    mousePressEvent (event)
        QWidget.mousePressEvent(QMouseEvent)

    mousePressEvent (event)
        QWidget.mousePressEvent(QMouseEvent)

```

6.6.9 Service

The Service package contains QObjects to display data of the backend service

6.6.9.1 Service

Service manage creation of QWidget to display service data

```

class alignak_app.qobjects.service.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

6.6.9.2 Services

Services manage creation of QWidget to display the services

```
class alignak_app.qobjects.service.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
```

6.6.9.3 Number Services

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

```
class alignak_app.qobjects.service.services_dashboard.ServicesDashboardQWidget (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

6.6.9.4 Service Tree Item

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

```
class alignak_app.qobjects.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
```

6.6.10 Systray

The Service package contains `QObjects` to display system tray icon (taskbar)

6.6.10.1 App TrayIcon

App TrayIcon manage the creation of `QSystemTrayIcon` for Alignak-app menus

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

    Class who create QMenu and QActions for QSystemTrayIcon (displayed in task bar)

    add_about_menu ()
        Create and add to menu “about” QAction

    add_alignak_menu ()
        Create and add to menu “app” QAction

    add_quit_menu ()
        Create and add to menu “exit” QAction

    add_reload_menu ()
        Create and add to menu “reload” QAction

    add_webui_menu ()
        Create and add to menu “webui” QAction

    build_menu ()
        Initialize and create each QAction of QMenu

    check_connection ()
        Check periodically connection to Alignak backend

    static quit_app ()
        Quit application and close all widgets handle by application
```

```
refresh_menus()  
    Refresh menu if needed  
  
static reload_configuration()  
    Reload configuration
```

6.6.11 Threads

The Threads package contains QObjects who manage application QThreads

6.6.11.1 Backend Thread

Backend Thread manage creation of QThreads for backend requests

```
class alignak_app.qobjects.threads.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
```

6.6.11.2 Thread Manager

Thread Manager manage BackendQThreads creations and their priority

```
class alignak_app.qobjects.threads.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
```


6.6.12 User

The User package contains QObjects to display data of the backend `user`

6.6.12.1 User Options

User Options manage creation options of QDialogs for user notifications options

class `alignak_app.qobjects.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.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

6.6.12.2 Password

Password manage Qdialog for user password

```
class alignak_app.qobjects.user.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
```

6.6.12.3 Profile

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

```
class alignak_app.qobjects.user.profile.ProfileQWidget (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

        Parameters

        • notification_type (str) – type of notifications (host or service)

        • btn_state (bool) – state of sender button

    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_mail_widget()
        Return QWidget with notes and email areas and edition buttons

        Returns notes and email QWidget

        Return type QWidget

    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 (*patch_type*)

Display QDialogs for patches

Parameters **patch_type** (*str*) – type of patch (“notes” or “password”)

static show_token_dialog ()

Show Token QDialog

update_widget ()

Update UserQWidget

6.7 About

6.7.1 About

About manage creation of QDialog to display data about Alignak-app

class alignak_app.qobjects.about.**AboutQDialog** (*parent=None*)

Bases: PyQt5.QtWidgets.QDialog

Class who create QDialog to display about data

static get_external_link_label (*title=""*)

Return QLabel with clickable text

Parameters

- **link** (*str*) – link to make clickable
- **title** (*str*) – title of link

Returns QLabel with clickable link

Return type QLabel

initialize ()

Initialize QDialog

show_about ()

Show about dialog

CHAPTER 7

Indices and tables

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

a

`alignak_app`, 19
`alignak_app.app`, 21
`alignak_app.backend`, 22
`alignak_app.backend.backend`, 22
`alignak_app.backend.datamanager`, 24
`alignak_app.items`, 25
`alignak_app.items.daemon`, 27
`alignak_app.items.event`, 28
`alignak_app.items.history`, 29
`alignak_app.items.host`, 30
`alignak_app.items.item`, 25
`alignak_app.items.livesynthesis`, 31
`alignak_app.items.period`, 31
`alignak_app.items.realm`, 32
`alignak_app.items.service`, 33
`alignak_app.items.user`, 34
`alignak_app.locales`, 39
`alignak_app.locales.locales`, 39
`alignak_app.qobjects`, 39
`alignak_app.qobjects.about`, 61
`alignak_app.qobjects.alignak`, 41
`alignak_app.qobjects.alignak.alignak`, 41
`alignak_app.qobjects.alignak.dashboard`, 41
`alignak_app.qobjects.alignak.livestate`, 42
`alignak_app.qobjects.alignak.problems`, 42
`alignak_app.qobjects.alignak.problems_table`, 43
`alignak_app.qobjects.alignak.status`, 44
`alignak_app.qobjects.app_main`, 39
`alignak_app.qobjects.common`, 44
`alignak_app.qobjects.common.actions`, 44
`alignak_app.qobjects.common.buttons`, 45
`alignak_app.qobjects.common.dialogs`, 46
`alignak_app.qobjects.common.frames`, 47
`alignak_app.qobjects.common.labels`, 48
`alignak_app.qobjects.common.widgets`, 49
`alignak_app.qobjects.dock`, 39
`alignak_app.qobjects.events`, 49
`alignak_app.qobjects.events.events`, 49
`alignak_app.qobjects.events.item`, 51
`alignak_app.qobjects.events.spy`, 51
`alignak_app.qobjects.host`, 52
`alignak_app.qobjects.host.history`, 52
`alignak_app.qobjects.host.host`, 53
`alignak_app.qobjects.login`, 54
`alignak_app.qobjects.login.login`, 54
`alignak_app.qobjects.login.server`, 55
`alignak_app.qobjects.panel`, 39
`alignak_app.qobjects.service`, 55
`alignak_app.qobjects.service.service`, 55
`alignak_app.qobjects.service.services`, 56
`alignak_app.qobjects.service.services_dashboard`, 56
`alignak_app.qobjects.service.tree_item`, 57
`alignak_app.qobjects.systray`, 57
`alignak_app.qobjects.systray.tray_icon`, 57
`alignak_app.qobjects.threads`, 58
`alignak_app.qobjects.threads.thread`, 58
`alignak_app.qobjects.threads.threadmanager`, 58
`alignak_app.qobjects.user`, 59
`alignak_app.qobjects.user.options`, 59
`alignak_app.qobjects.user.password`, 59
`alignak_app.qobjects.user.profile`, 60
`alignak_app.utils`, 35
`alignak_app.utils.config`, 35
`alignak_app.utils.installer`, 36
`alignak_app.utils.logs`, 38
`alignak_app.utils.system`, 37
`alignak_app.utils.time`, 38

A

- AboutQDialog (class in alignak_app.qobjects.about), 61
- accept_login() (alignak_app.qobjects.login.login.LoginQDialog method), 54
- accept_text() (alignak_app.qobjects.common.dialogs.EditQDialog method), 46
- accept_text() (alignak_app.qobjects.common.dialogs.ValidatorQDialog method), 47
- AckQDialog (class in alignak_app.qobjects.common.actions), 44
- ActionsQWidget (class in alignak_app.qobjects.common.actions), 45
- add_about_menu() (alignak_app.qobjects.systay.tray_icon.AppTrayIcon method), 57
- add_acknowledge() (alignak_app.qobjects.common.actions.ActionsQWidget method), 45
- add_alignak_menu() (alignak_app.qobjects.systay.tray_icon.AppTrayIcon method), 57
- add_daemon_labels() (alignak_app.qobjects.alignak.status.StatusQDialog method), 44
- add_daemon_titles_labels() (alignak_app.qobjects.alignak.status.StatusQDialog method), 44
- add_downtime() (alignak_app.qobjects.common.actions.ActionsQWidget method), 45
- add_event() (alignak_app.qobjects.events.events.EventsQWidget method), 50
- add_priority_thread() (alignak_app.qobjects.threads.threadmanager.ThreadManager method), 58
- add_quit_menu() (alignak_app.qobjects.systay.tray_icon.AppTrayIcon method), 57
- add_reload_menu() (alignak_app.qobjects.systay.tray_icon.AppTrayIcon method), 57
- add_spied_host() (alignak_app.qobjects.alignak.problems.ProblemsQWidget method), 42
- add_spy_host() (alignak_app.qobjects.events.spy.SpyQWidget method), 51
- add_webui_menu() (alignak_app.qobjects.systay.tray_icon.AppTrayIcon method), 57
- add_widget() (alignak_app.qobjects.common.frames.AppQFrame method), 47
- alignak_app (module), 19
- alignak_app.app (module), 21
- alignak_app.backend (module), 22
- alignak_app.backend.backend (module), 22
- alignak_app.backend.datamanager (module), 24
- alignak_app.items (module), 25
- alignak_app.items.daemon (module), 27
- alignak_app.items.event (module), 28
- alignak_app.items.history (module), 29
- alignak_app.items.host (module), 30
- alignak_app.items.item (module), 25
- alignak_app.items.livesynthesis (module), 31
- alignak_app.items.period (module), 31
- alignak_app.items.realm (module), 32
- alignak_app.items.service (module), 33
- alignak_app.items.user (module), 34
- alignak_app.locales (module), 39
- alignak_app.locales.locales (module), 39
- alignak_app.qobjects (module), 39
- alignak_app.qobjects.about (module), 61
- alignak_app.qobjects.alignak (module), 41
- alignak_app.qobjects.alignak.alignak (module), 41
- alignak_app.qobjects.alignak.dashboard (module), 41
- alignak_app.qobjects.alignak.livestate (module), 42
- alignak_app.qobjects.alignak.problems (module), 42
- alignak_app.qobjects.alignak.problems_table (module), 43
- alignak_app.qobjects.alignak.status (module), 44
- alignak_app.qobjects.app_main (module), 39

[alignak_app.qobjects.common \(module\), 44](#)
[alignak_app.qobjects.common.actions \(module\), 44](#)
[alignak_app.qobjects.common.buttons \(module\), 45](#)
[alignak_app.qobjects.common.dialogs \(module\), 46](#)
[alignak_app.qobjects.common.frames \(module\), 47](#)
[alignak_app.qobjects.common.labels \(module\), 48](#)
[alignak_app.qobjects.common.widgets \(module\), 49](#)
[alignak_app.qobjects.dock \(module\), 39](#)
[alignak_app.qobjects.events \(module\), 49](#)
[alignak_app.qobjects.events.events \(module\), 49](#)
[alignak_app.qobjects.events.item \(module\), 51](#)
[alignak_app.qobjects.events.spy \(module\), 51](#)
[alignak_app.qobjects.host \(module\), 52](#)
[alignak_app.qobjects.host.history \(module\), 52](#)
[alignak_app.qobjects.host.host \(module\), 53](#)
[alignak_app.qobjects.login \(module\), 54](#)
[alignak_app.qobjects.login.login \(module\), 54](#)
[alignak_app.qobjects.login.server \(module\), 55](#)
[alignak_app.qobjects.panel \(module\), 39](#)
[alignak_app.qobjects.service \(module\), 55](#)
[alignak_app.qobjects.service.service \(module\), 55](#)
[alignak_app.qobjects.service.services \(module\), 56](#)
[alignak_app.qobjects.service.services_dashboard \(module\), 56](#)
[alignak_app.qobjects.service.tree_item \(module\), 57](#)
[alignak_app.qobjects.systray \(module\), 57](#)
[alignak_app.qobjects.systray.tray_icon \(module\), 57](#)
[alignak_app.qobjects.threads \(module\), 58](#)
[alignak_app.qobjects.threads.thread \(module\), 58](#)
[alignak_app.qobjects.threads.threadmanager \(module\), 58](#)
[alignak_app.qobjects.user \(module\), 59](#)
[alignak_app.qobjects.user.options \(module\), 59](#)
[alignak_app.qobjects.user.password \(module\), 59](#)
[alignak_app.qobjects.user.profile \(module\), 60](#)
[alignak_app.utils \(module\), 35](#)
[alignak_app.utils.config \(module\), 35](#)
[alignak_app.utils.installer \(module\), 36](#)
[alignak_app.utils.logs \(module\), 38](#)
[alignak_app.utils.system \(module\), 37](#)
[alignak_app.utils.time \(module\), 38](#)
[AlignakApp \(class in alignak_app.app\), 21](#)
[AlignakQWidget \(class in alignak_app.qobjects.alignak.alignak\), 41](#)
[AppProgressBar \(class in alignak_app.app\), 21](#)
[AppProgressQWidget \(class in alignak_app.app\), 22](#)
[AppQFrame \(class in alignak_app.qobjects.common.frames\), 47](#)
[AppQMainWindow \(class in alignak_app.qobjects.app_main\), 39](#)
[AppTrayIcon \(class in alignak_app.qobjects.systray.tray_icon\), 57](#)

B

[BackendClient \(class in alignak_app.backend.backend\), 22](#)
[BackendQThread \(class in alignak_app.qobjects.threads.thread\), 58](#)
[build_menu\(\) \(alignak_app.qobjects.systray.tray_icon.AppTrayIcon method\), 57](#)

C

[center_widget\(\) \(in module alignak_app.qobjects.common.widgets\), 49](#)
[check_connection\(\) \(alignak_app.qobjects.systray.tray_icon.AppTrayIcon method\), 57](#)
[check_environment\(\) \(alignak_app.utils.installer.Installer method\), 36](#)
[check_install_files\(\) \(alignak_app.utils.installer.Installer method\), 36](#)
[check_install_folders\(\) \(alignak_app.utils.installer.Installer method\), 36](#)
[check_installation\(\) \(alignak_app.utils.installer.Installer method\), 37](#)
[check_text\(\) \(alignak_app.qobjects.common.dialogs.ValidatorQDialog method\), 47](#)
[check_threads\(\) \(alignak_app.app.AlignakApp static method\), 21](#)
[check_user_installation\(\) \(alignak_app.utils.installer.Installer method\), 37](#)
[clean_threads\(\) \(alignak_app.qobjects.threads.threadmanager.ThreadManager method\), 58](#)
[close_item\(\) \(alignak_app.qobjects.events.item.EventItem method\), 51](#)
[close_widget\(\) \(alignak_app.qobjects.common.frames.AppQFrame method\), 47](#)
[create\(\) \(alignak_app.items.daemon.Daemon method\), 27](#)
[create\(\) \(alignak_app.items.event.Event method\), 28](#)
[create\(\) \(alignak_app.items.history.History method\), 29](#)
[create\(\) \(alignak_app.items.host.Host method\), 30](#)
[create\(\) \(alignak_app.items.item.Item method\), 26](#)
[create\(\) \(alignak_app.items.livesynthesis.LiveSynthesis method\), 31](#)
[create\(\) \(alignak_app.items.period.Period method\), 32](#)
[create\(\) \(alignak_app.items.realm.Realm method\), 32](#)
[create\(\) \(alignak_app.items.service.Service method\), 33](#)
[create\(\) \(alignak_app.items.user.User method\), 34](#)
[create_line_search\(\) \(alignak_app.qobjects.panel.PanelQWidget method\), 40](#)
[create_logger\(\) \(in module alignak_app.utils.logs\), 38](#)
[create_widget\(\) \(alignak_app.qobjects.login.login.LoginQDialog method\), 54](#)

D

Daemon (class in alignak_app.items.daemon), 27
 DashboardQWidget (class in alignak_app.qobjects.alignak.dashboard), 41
 DataManager (class in alignak_app.backend.datamaneger), 24
 define_hostname() (alignak_app.qobjects.panel.PanelQWidget method), 40
 display_host() (alignak_app.qobjects.panel.PanelQWidget method), 40
 DockQWidget (class in alignak_app.qobjects.dock), 39
 DownQDialog (class in alignak_app.qobjects.common.actions), 45
 dragEnterEvent() (alignak_app.qobjects.panel.PanelQWidget method), 40
 dragMoveEvent() (alignak_app.qobjects.events.spy.SpyQListWidget method), 52
 dragMoveEvent() (alignak_app.qobjects.panel.PanelQWidget method), 40
 dropEvent() (alignak_app.qobjects.events.spy.SpyQListWidget method), 52
 dropEvent() (alignak_app.qobjects.panel.PanelQWidget method), 40
 duration_to_seconds() (alignak_app.qobjects.common.actions.DownQDialog method), 45

E

edit_setting_value() (alignak_app.utils.config.Settings method), 35
 EditQDialog (class in alignak_app.qobjects.common.dialogs), 46
 enable_notifications() (alignak_app.qobjects.user.profile.ProfileQWidget method), 60
 Event (class in alignak_app.items.event), 28
 event_exist() (alignak_app.qobjects.events.events.EventsQWidget method), 50
 EventItem (class in alignak_app.qobjects.events.item), 51
 EventsQWidget (class in alignak_app.qobjects.events.events), 50

F

file_executable() (in module alignak_app.utils.system), 37

G

get() (alignak_app.backend.backend.BackendClient method), 22
 get_actions_widget() (alignak_app.qobjects.host.host.HostQWidget method), 53
 get_actions_widget() (alignak_app.qobjects.service.service.ServiceDataQWidget method), 55
 get_all_hostnames() (alignak_app.backend.datamaneger.DataManager method), 24
 get_available_icons() (alignak_app.items.host.Host static method), 30
 get_available_icons() (alignak_app.items.service.Service static method), 33
 get_backend_status_icon() (alignak_app.backend.backend.BackendClient method), 22
 get_btn_widget() (alignak_app.qobjects.alignak.problems.ProblemsQWidget method), 42
 get_check_text() (alignak_app.items.daemon.Daemon static method), 27
 get_check_text() (alignak_app.items.event.Event static method), 28
 get_check_text() (alignak_app.items.history.History static method), 29
 get_check_text() (alignak_app.items.host.Host static method), 30
 get_check_text() (alignak_app.items.item.Item static method), 26
 get_check_text() (alignak_app.items.livesynthesis.LiveSynthesis static method), 31
 get_check_text() (alignak_app.items.period.Period static method), 32
 get_check_text() (alignak_app.items.realm.Realm static method), 33
 get_check_text() (alignak_app.items.service.Service static method), 34
 get_check_text() (alignak_app.items.user.User static method), 34
 get_config() (alignak_app.utils.config.Settings method), 35
 get_curent_user_role_item() (alignak_app.qobjects.alignak.problems.ProblemsQWidget method), 42
 get_current_time() (in module alignak_app.utils.time), 38
 get_daemons_names() (alignak_app.items.daemon.Daemon static method), 27
 get_daemons_status_icon() (alignak_app.items.daemon.Daemon static method), 28
 get_display_name() (alignak_app.items.host.Host static method), 30
 get_display_name() (alignak_app.items.period.Period static method), 31

method), 32

get_display_name() (alignak_app.items.realm.Realm method), 33

get_display_name() (alignak_app.items.service.Service method), 34

get_event_type() (alignak_app.qobjects.host.history.HistoryQWidget static method), 52

get_event_widget() (alignak_app.qobjects.host.history.HistoryQWidget method), 52

get_events() (alignak_app.backend.datamanager.DataManager method), 24

get_events_widget() (in module alignak_app.qobjects.events.events), 50

get_external_link_label() (alignak_app.qobjects.about.AboutQDialog static method), 61

get_frame_separator() (in module alignak_app.qobjects.common.frames), 48

get_history_icon_name() (alignak_app.items.history.History static method), 29

get_host_icon_widget() (alignak_app.qobjects.host.host.HostQWidget method), 53

get_host_msg_and_event_type() (in module alignak_app.items.item), 26

get_host_resume_widget() (alignak_app.qobjects.alignak.dashboard.DashboardQWidget method), 41

get_host_services() (alignak_app.backend.datamanager.DataManager method), 24

get_host_with_services() (alignak_app.backend.datamanager.DataManager method), 24

get_hosts_notif_widget() (alignak_app.qobjects.user.profile.ProfileQWidget method), 60

get_icon() (alignak_app.qobjects.events.item.EventItem static method), 51

get_icon_item() (in module alignak_app.qobjects.common.labels), 48

get_icon_label() (alignak_app.qobjects.host.history.HistoryQWidget static method), 53

get_icon_name() (in module alignak_app.items.item), 26

get_icon_name_from_state() (in module alignak_app.items.item), 27

get_icon_pixmap() (in module alignak_app.qobjects.common.labels), 48

get_image() (alignak_app.utils.config.Settings method), 36

get_informations_widget() (alignak_app.qobjects.user.profile.ProfileQWidget method), 60

get_item() (alignak_app.backend.datamanager.DataManager method), 24

get_item_text() (alignak_app.qobjects.alignak.problems_table.ProblemsQT static method), 43

get_item_type_widget() (alignak_app.qobjects.alignak.livestate.LivestateQWidget method), 42

get_items_and_problems() (alignak_app.backend.datamanager.DataManager method), 24

get_last_check_widget() (alignak_app.qobjects.host.host.HostQWidget method), 53

get_last_check_widget() (alignak_app.qobjects.service.service.ServiceDataQWidget method), 56

get_local_datetime() (in module alignak_app.utils.time), 38

get_local_datetime_from_date() (in module alignak_app.utils.time), 38

get_logo_widget() (in module alignak_app.qobjects.common.widgets), 49

get_message_widget() (alignak_app.qobjects.common.dialogs.MessageQDialog method), 46

get_notes_mail_widget() (alignak_app.qobjects.user.profile.ProfileQWidget method), 60

get_notes_output_widget() (alignak_app.qobjects.host.host.HostQWidget method), 53

get_notifications_widget() (alignak_app.qobjects.user.options.UserOptionsQDialog method), 59

get_notifications_widget() (alignak_app.qobjects.user.profile.ProfileQWidget method), 60

get_output_tableitem() (alignak_app.qobjects.alignak.problems_table.ProblemsQT static method), 43

get_period_name() (alignak_app.backend.datamanager.DataManager method), 25

get_problems() (alignak_app.backend.datamanager.DataManager method), 25

get_real_host_state_icon() (in module alignak_app.items.item), 27

get_realm_name() (alignak_app.backend.datamanager.DataManager method), 25

get_request_model() (alignak_app.items.daemon.Daemon static method), 28

[get_request_model\(\)](#) (alignak_app.items.event.Event static method), 28
[get_request_model\(\)](#) (alignak_app.items.history.History static method), 29
[get_request_model\(\)](#) (alignak_app.items.host.Host static method), 30
[get_request_model\(\)](#) (alignak_app.items.livesynthesis.LiveSynthesis static method), 31
[get_request_model\(\)](#) (alignak_app.items.period.Period static method), 32
[get_request_model\(\)](#) (alignak_app.items.realm.Realm static method), 33
[get_request_model\(\)](#) (alignak_app.items.service.Service static method), 34
[get_request_model\(\)](#) (alignak_app.items.user.User static method), 35
[get_role\(\)](#) (alignak_app.items.user.User method), 35
[get_search_widget\(\)](#) (alignak_app.qobjects.alignak.problems.ProblemsQWidget method), 43
[get_search_widget\(\)](#) (alignak_app.qobjects.panel.PanelQWidget method), 40
[get_selected_options\(\)](#) (alignak_app.qobjects.user.options.UserOptionsQDialog static method), 59
[get_service_icon_widget\(\)](#) (alignak_app.qobjects.service.service.ServiceDataQWidget method), 56
[get_service_states_nb\(\)](#) (alignak_app.items.service.Service static method), 34
[get_services_notif_widget\(\)](#) (alignak_app.qobjects.user.profile.ProfileQWidget method), 60
[get_services_resume_widget\(\)](#) (alignak_app.qobjects.alignak.dashboard.DashboardQWidget method), 41
[get_settings_widget\(\)](#) (alignak_app.qobjects.login.server.ServerQDialog method), 55
[get_states\(\)](#) (alignak_app.items.daemon.Daemon static method), 28
[get_synthesis_count\(\)](#) (alignak_app.backend.datamanager.DataManager method), 25
[get_synthesis_count_model\(\)](#) (alignak_app.items.livesynthesis.LiveSynthesis static method), 31
[get_synthesis_widget\(\)](#) (alignak_app.qobjects.panel.PanelQWidget method), 40
[get_tableitem\(\)](#) (alignak_app.qobjects.alignak.problems_table.ProblemsQTableView method), 43
[get_text_widget\(\)](#) (alignak_app.qobjects.common.dialogs.EditQDialog method), 46
[get_text_widget\(\)](#) (alignak_app.qobjects.common.dialogs.ValidatorQDialog method), 47
[get_threads_to_launch\(\)](#) (alignak_app.qobjects.threads.threadmanager.ThreadManager method), 58
[get_time_diff_since_last_timestamp\(\)](#) (in module alignak_app.utils.time), 38
[get_title_widget\(\)](#) (alignak_app.qobjects.common.frames.AppQFrame method), 47
[get_tooltip\(\)](#) (alignak_app.items.daemon.Daemon method), 28
[get_tooltip\(\)](#) (alignak_app.items.event.Event method), 29
[get_tooltip\(\)](#) (alignak_app.items.history.History method), 29
[get_tooltip\(\)](#) (alignak_app.items.host.Host method), 30
[get_tooltip\(\)](#) (alignak_app.items.item.Item method), 26
[get_tooltip\(\)](#) (alignak_app.items.livesynthesis.LiveSynthesis method), 31
[get_tooltip\(\)](#) (alignak_app.items.period.Period method), 32
[get_tooltip\(\)](#) (alignak_app.items.realm.Realm method), 33
[get_tooltip\(\)](#) (alignak_app.items.service.Service method), 34
[get_tooltip\(\)](#) (alignak_app.items.user.User method), 35
[get_url_endpoint_from_icon_name\(\)](#) (in module alignak_app.utils.config), 36
[get_variables_widget\(\)](#) (alignak_app.qobjects.host.host.HostQWidget method), 54
[handle_accept\(\)](#) (alignak_app.qobjects.common.actions.DownQDialog method), 45
[handle_confirm\(\)](#) (alignak_app.qobjects.user.password.PasswordQDialog method), 60
[handle_proxy\(\)](#) (alignak_app.qobjects.login.login.LoginQDialog method), 54
[handle_server\(\)](#) (alignak_app.qobjects.login.login.LoginQDialog static method), 54
[History](#) (class in alignak_app.items.history), 29
[HistoryQWidget](#) (class in alignak_app.qobjects.host.history), 52
[Host](#) (class in alignak_app.items.host), 30
[HostQWidget](#) (class in alignak_app.qobjects.host.host), 53

I

- `init_config()` (`alignak_app.utils.config.Settings` method), 36
- `init_css()` (`alignak_app.utils.config.Settings` method), 36
- `init_environment()` (`alignak_app.utils.installer.Installer` method), 37
- `init_event_widget()` (in module `alignak_app.qobjects.events.events`), 50
- `init_localization()` (in module `alignak_app.locales.locales`), 39
- `initialize()` (`alignak_app.app.AppProgressQWidget` method), 22
- `initialize()` (`alignak_app.qobjects.about.AboutQDialog` method), 61
- `initialize()` (`alignak_app.qobjects.alignak.alignak.AlignakQWidget` method), 41
- `initialize()` (`alignak_app.qobjects.alignak.dashboard.DashboardQWidget` method), 41
- `initialize()` (`alignak_app.qobjects.alignak.livestate.LivestateQWidget` method), 42
- `initialize()` (`alignak_app.qobjects.alignak.problems.ProblemsQWidget` method), 43
- `initialize()` (`alignak_app.qobjects.alignak.status.StatusQDialog` method), 44
- `initialize()` (`alignak_app.qobjects.app_main.AppQMainWindow` method), 39
- `initialize()` (`alignak_app.qobjects.common.actions.AckQDialog` method), 44
- `initialize()` (`alignak_app.qobjects.common.actions.ActionsQWidget` method), 45
- `initialize()` (`alignak_app.qobjects.common.actions.DownQDialog` method), 45
- `initialize()` (`alignak_app.qobjects.common.buttons.ToggleQWidgetButton` method), 46
- `initialize()` (`alignak_app.qobjects.common.dialogs.EditQDialog` method), 46
- `initialize()` (`alignak_app.qobjects.common.dialogs.MessageQDialog` method), 47
- `initialize()` (`alignak_app.qobjects.common.dialogs.ValidatorQDialog` method), 47
- `initialize()` (`alignak_app.qobjects.common.frames.AppQFrame` method), 48
- `initialize()` (`alignak_app.qobjects.common.widgets.LogoQWidget` method), 49
- `initialize()` (`alignak_app.qobjects.dock.DockQWidget` method), 39
- `initialize()` (`alignak_app.qobjects.events.events.EventsQWidget` method), 50
- `initialize()` (`alignak_app.qobjects.events.item.EventItem` method), 51
- `initialize()` (`alignak_app.qobjects.events.spy.SpyQWidget` method), 52
- `initialize()` (`alignak_app.qobjects.host.history.HistoryQWidget` method), 53
- `initialize()` (`alignak_app.qobjects.host.host.HostQWidget` method), 54
- `initialize()` (`alignak_app.qobjects.panel.PanelQWidget` method), 40
- `initialize()` (`alignak_app.qobjects.service.service.ServiceDataQWidget` method), 56
- `initialize()` (`alignak_app.qobjects.service.services.ServicesQWidget` method), 56
- `initialize()` (`alignak_app.qobjects.service.services_dashboard.ServicesDashboardQWidget` method), 56
- `initialize()` (`alignak_app.qobjects.service.tree_item.ServiceTreeItem` method), 57
- `initialize()` (`alignak_app.qobjects.user.options.UserOptionsQDialog` method), 59
- `initialize()` (`alignak_app.qobjects.user.password.PasswordQDialog` method), 60
- `initialize()` (`alignak_app.qobjects.user.profile.ProfileQWidget` method), 60
- `initialize_dialog()` (`alignak_app.qobjects.login.server.ServerQDialog` method), 55
- `install()` (`alignak_app.utils.installer.Installer` method), 37
- `install_file()` (in module `alignak_app.utils.system`), 37
- `Installer` (class in `alignak_app.utils.installer`), 36
- `is_checked()` (`alignak_app.qobjects.common.buttons.ToggleQWidgetButton` method), 46
- `is_ready()` (`alignak_app.backend.datamanager.DataManager` method), 25
- `Item` (class in `alignak_app.items.item`), 26

M

- `MessageQDialog` (class in `alignak_app.qobjects.common.dialogs`), 46
- `minimize()` (`alignak_app.qobjects.common.frames.AppQFrame` method), 48
- `minimize()` (`alignak_app.qobjects.common.widgets.LogoQWidget` method), 49
- `minimize_maximize()` (`alignak_app.qobjects.common.frames.AppQFrame` method), 48

[minimize_maximize\(\)](#) (alignak_app.qobjects.common.widgets.LogoQWidget method), [49](#)
[mkdir\(\)](#) (in module alignak_app.utils.system), [37](#)
[mousemoveEvent\(\)](#) (alignak_app.qobjects.app_main.AppQMainWindow method), [39](#)
[mousemoveEvent\(\)](#) (alignak_app.qobjects.common.actions.AckQDialog method), [45](#)
[mousemoveEvent\(\)](#) (alignak_app.qobjects.common.actions.DownQDialog method), [45](#)
[mousemoveEvent\(\)](#) (alignak_app.qobjects.common.frames.AppQFrame method), [48](#)
[mousemoveEvent\(\)](#) (alignak_app.qobjects.login.login.LoginQDialog method), [55](#)
[mousemoveEvent\(\)](#) (alignak_app.qobjects.login.server.ServerQDialog method), [55](#)
[mousePressEvent\(\)](#) (alignak_app.qobjects.app_main.AppQMainWindow method), [39](#)
[mousePressEvent\(\)](#) (alignak_app.qobjects.common.actions.AckQDialog method), [45](#)
[mousePressEvent\(\)](#) (alignak_app.qobjects.common.actions.DownQDialog method), [45](#)
[mousePressEvent\(\)](#) (alignak_app.qobjects.common.frames.AppQFrame method), [48](#)
[mousePressEvent\(\)](#) (alignak_app.qobjects.login.login.LoginQDialog method), [55](#)
[mousePressEvent\(\)](#) (alignak_app.qobjects.login.server.ServerQDialog method), [55](#)
O
[open_item_type_url\(\)](#) (alignak_app.qobjects.alignak.dashboard.DashboardQWidget method), [41](#)
[open_url\(\)](#) (in module alignak_app.utils.config), [36](#)
[open_user_widget\(\)](#) (alignak_app.qobjects.alignak.alignak.AlignakQWidget method), [41](#)
P
[paintEvent\(\)](#) (alignak_app.qobjects.alignak.livestate.LivestateQWidget method), [42](#)
[paintEvent\(\)](#) (alignak_app.qobjects.common.widgets.LogoQWidget method), [49](#)
[PanelQWidget](#) (class in alignak_app.qobjects.panel), [40](#)
[PasswordQDialog](#) (class in alignak_app.qobjects.user.password), [59](#)
[patch\(\)](#) (alignak_app.backend.backend.BackendClient method), [23](#)
[patch_data\(\)](#) (alignak_app.qobjects.host.host.HostQWidget method), [54](#)
[patch_data\(\)](#) (alignak_app.qobjects.user.profile.ProfileQWidget method), [61](#)
[patch_host_checks\(\)](#) (alignak_app.qobjects.host.host.HostQWidget method), [54](#)
[Period](#) (class in alignak_app.items.period), [32](#)
[periodic_refresh\(\)](#) (alignak_app.qobjects.service.service.ServiceDataQWidget method), [56](#)
[post\(\)](#) (alignak_app.backend.backend.BackendClient method), [23](#)
[ProblemsQTableView](#) (class in alignak_app.qobjects.alignak.problems_table), [43](#)
[ProblemsQWidget](#) (class in alignak_app.qobjects.alignak.problems), [42](#)
[ProfileQWidget](#) (class in alignak_app.qobjects.user.profile), [60](#)
Q
[query_daemons_data\(\)](#) (alignak_app.backend.backend.BackendClient method), [23](#)
[query_history_data\(\)](#) (alignak_app.backend.backend.BackendClient method), [23](#)
[query_hosts_data\(\)](#) (alignak_app.backend.backend.BackendClient method), [23](#)
[query_livesynthesis_data\(\)](#) (alignak_app.backend.backend.BackendClient method), [23](#)
[query_notifications_data\(\)](#) (alignak_app.backend.backend.BackendClient method), [23](#)
[query_period_data\(\)](#) (alignak_app.backend.backend.BackendClient method), [23](#)
[query_realms_data\(\)](#) (alignak_app.backend.backend.BackendClient method), [23](#)
[query_services_data\(\)](#) (alignak_app.backend.backend.BackendClient method), [23](#)

[query_user_data\(\)](#) (alignak_app.backend.backend.BackendClient method), [24](#)
[quit_app\(\)](#) (alignak_app.qobjects.systray.tray_icon.AppTrayIcon static method), [57](#)
[quit_launched_threads\(\)](#) (alignak_app.app.AlignakApp static method), [21](#)

R

[read_config_file\(\)](#) (in module alignak_app.utils.system), [37](#)
[Realm](#) (class in alignak_app.items.realm), [32](#)
[refresh_menus\(\)](#) (alignak_app.qobjects.systray.tray_icon.AppTrayIcon method), [57](#)
[reload_configuration\(\)](#) (alignak_app.qobjects.systray.tray_icon.AppTrayIcon static method), [58](#)
[remove_event\(\)](#) (alignak_app.qobjects.events.events.EventsQWidget method), [50](#)
[remove_event\(\)](#) (alignak_app.qobjects.events.spy.SpyQWidget method), [52](#)
[remove_timer_event\(\)](#) (alignak_app.qobjects.events.events.EventsQWidget method), [50](#)
[run\(\)](#) (alignak_app.qobjects.threads.thread.BackendQThread method), [58](#)

S

[send_datamanager_events\(\)](#) (alignak_app.qobjects.events.events.EventsQWidget method), [50](#)
[send_event\(\)](#) (in module alignak_app.qobjects.events.events), [50](#)
[send_spy_events\(\)](#) (alignak_app.qobjects.events.spy.SpyQWidget method), [52](#)
[ServerQDialog](#) (class in alignak_app.qobjects.login.server), [55](#)
[Service](#) (class in alignak_app.items.service), [33](#)
[ServiceDataQWidget](#) (class in alignak_app.qobjects.service.service), [55](#)
[ServicesDashboardQWidget](#) (class in alignak_app.qobjects.service.services_dashboard), [56](#)
[ServicesQWidget](#) (class in alignak_app.qobjects.service.services), [56](#)
[ServiceTreeWidgetItem](#) (class in alignak_app.qobjects.service.tree_item), [57](#)
[set_daemons_labels\(\)](#) (alignak_app.qobjects.alignak.status.StatusQDialog method), [44](#)
[set_data\(\)](#) (alignak_app.qobjects.host.host.HostQWidget method), [54](#)
[set_data\(\)](#) (alignak_app.qobjects.service.services.ServicesQWidget method), [56](#)
[set_proxy_settings\(\)](#) (alignak_app.qobjects.login.login.LoginQDialog method), [55](#)
[set_text\(\)](#) (alignak_app.app.AppProgressBar method), [22](#)
[Settings](#) (class in alignak_app.utils.config), [35](#)
[show_about\(\)](#) (alignak_app.qobjects.about.AboutQDialog method), [61](#)
[show_history\(\)](#) (alignak_app.qobjects.host.host.HostQWidget method), [54](#)
[show_login_window\(\)](#) (alignak_app.app.AlignakApp static method), [21](#)
[show_options_dialog\(\)](#) (in module alignak_app.qobjects.user.options), [59](#)
[show_status_dialog\(\)](#) (alignak_app.qobjects.alignak.alignak.AlignakQWidget method), [41](#)
[show_token_dialog\(\)](#) (alignak_app.qobjects.user.profile.ProfileQWidget static method), [61](#)
[show_widget\(\)](#) (alignak_app.qobjects.common.frames.AppQFrame method), [48](#)
[showEvent\(\)](#) (alignak_app.qobjects.login.login.LoginQDialog method), [55](#)
[spy_host\(\)](#) (alignak_app.qobjects.panel.PanelQWidget method), [40](#)
[SpyQListWidget](#) (class in alignak_app.qobjects.events.spy), [51](#)
[SpyQWidget](#) (class in alignak_app.qobjects.events.spy), [52](#)
[start\(\)](#) (alignak_app.app.AlignakApp method), [21](#)
[StatusQDialog](#) (class in alignak_app.qobjects.alignak.status), [44](#)
[stop_priority_threads\(\)](#) (alignak_app.qobjects.threads.threadmanager.ThreadManager method), [58](#)
[stop_threads\(\)](#) (alignak_app.qobjects.threads.threadmanager.ThreadManager method), [58](#)

T

[text\(\)](#) (alignak_app.app.AppProgressBar method), [22](#)
[ThreadManager](#) (class in alignak_app.qobjects.threads.threadmanager), [58](#)
[ToggleQWidgetButton](#) (class in alignak_app.qobjects.common.buttons), [46](#)

U

[update_action_buttons\(\)](#) (alignak_app.qobjects.alignak.problems.ProblemsQWidget method), [43](#)
[update_btn_state\(\)](#) (alignak_app.qobjects.common.buttons.ToggleQWidgetButton method), [46](#)

method), 46

update_dashboard() (alignak_app.qobjects.alignak.dashboard.DashboardQWidget method), 42

update_data() (alignak_app.items.daemon.Daemon method), 28

update_data() (alignak_app.items.event.Event method), 29

update_data() (alignak_app.items.history.History method), 30

update_data() (alignak_app.items.host.Host method), 31

update_data() (alignak_app.items.item.Item method), 26

update_data() (alignak_app.items.livesynthesis.LiveSynthesis method), 31

update_data() (alignak_app.items.period.Period method), 32

update_data() (alignak_app.items.realm.Realm method), 33

update_data() (alignak_app.items.service.Service method), 34

update_data() (alignak_app.items.user.User method), 35

update_database() (alignak_app.backend.datamanager.DataManager method), 25

update_dialog() (alignak_app.qobjects.alignak.status.StatusQDialog method), 44

update_history_data() (alignak_app.qobjects.host.history.HistoryQWidget method), 53

update_host() (alignak_app.qobjects.host.host.HostQWidget method), 54

update_item() (alignak_app.qobjects.service.tree_item.ServiceTreeItem method), 57

update_item_data() (alignak_app.backend.datamanager.DataManager method), 25

update_labels() (alignak_app.qobjects.alignak.livestate.LivestateQWidget method), 42

update_panel_spytab() (alignak_app.qobjects.panel.PanelQWidget method), 40

update_parent_spytab() (alignak_app.qobjects.events.spy.SpyQWidget method), 52

update_problems_data() (alignak_app.qobjects.alignak.problems.ProblemsQWidget method), 43

update_service_data() (alignak_app.qobjects.service.services.ServicesQWidget method), 56

update_status() (alignak_app.qobjects.alignak.alignak.AlignakQWidget method), 41

update_view() (alignak_app.qobjects.alignak.problems_table.ProblemsQTableView method), 44

update_widget() (alignak_app.qobjects.common.actions.ActionsQWidget method), 45

update_widget() (alignak_app.qobjects.service.service.ServiceDataQWidget method), 56

update_widget() (alignak_app.qobjects.service.services.ServicesQWidget method), 56

update_widget() (alignak_app.qobjects.service.services_dashboard.Services method), 56

update_widget() (alignak_app.qobjects.user.profile.ProfileQWidget method), 61

User (class in alignak_app.items.user), 34

UserOptionsQDialog (class in alignak_app.qobjects.user.options), 59

V

ValidatorQDialog (class in alignak_app.qobjects.common.dialogs), 47

W

write_file() (in module alignak_app.utils.system), 37

write_rc_file() (in module alignak_app.utils.system), 38