
Alignak-App Documentation

Release 1.4

Matthieu Estrada

Apr 11, 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	Run	15
4.1	Alignak-App Launcher	15
4.2	Linux Daemon	16
4.3	Windows Installer	16

5	Interface	17
5.1	Login Window	17
5.2	Start	17
5.3	App Dock (Right part)	18
5.3.1	Alignak	18
5.3.2	Livestate part	18
5.3.3	Events part	18
5.4	App Panel (Left part)	18
5.4.1	Dashboard	18
5.4.2	Host Synthesis	19
5.4.2.1	Host View	19
5.4.2.2	Services	19
5.4.2.3	Actions	19
5.4.3	Problems	19
5.4.4	Spied Hosts	19
5.5	External Tools	20
5.5.1	WebUI	20
5.5.2	GLPI	20
6	Architecture	21
6.1	Resume	21
6.2	Architecture	21
7	API	23
7.1	Main Class	23
7.1.1	Alignak App	23
7.2	Backend	24
7.2.1	Backend	24
7.2.2	Data Manager	26
7.3	Items	29
7.3.1	Item	29
7.3.2	Daemon	30
7.3.3	Event	31
7.3.4	History	32
7.3.5	Host	33
7.3.6	Live Synthesis	34
7.3.7	Period	35
7.3.8	Realm	36
7.3.9	Service	37
7.3.10	User	38
7.4	Utils	38
7.4.1	Settings	39
7.4.2	Installer	40
7.4.3	System	40
7.4.4	Logs	41
7.4.5	Time	41
7.5	Locales	42
7.5.1	Locales	42
7.6	QObjects	42
7.6.1	App Main	42
7.6.2	Dock	43
7.6.3	Panel	43
7.6.4	Alignak	44
7.6.4.1	Alignak	44

7.6.4.2	Dashboard	44
7.6.4.3	Livestate	45
7.6.4.4	Problems	46
7.6.4.5	Problems Table	46
7.6.4.6	Status	47
7.6.5	Common	48
7.6.5.1	Actions	48
7.6.5.2	Buttons	49
7.6.5.3	Dialogs	49
7.6.5.4	Frames	51
7.6.5.5	Labels	51
7.6.5.6	Widgets	51
7.6.6	Events	52
7.6.6.1	Events	52
7.6.6.2	Event Item	54
7.6.6.3	Spy	54
7.6.7	Host	55
7.6.7.1	History	55
7.6.7.2	Host	56
7.6.8	Login	57
7.6.8.1	Login	57
7.6.8.2	Server	58
7.6.9	Service	58
7.6.9.1	Service	58
7.6.9.2	Services	59
7.6.9.3	Number Services	59
7.6.9.4	Service Tree Item	59
7.6.10	Systray	60
7.6.10.1	App TrayIcon	60
7.6.11	Threads	60
7.6.11.1	Backend Thread	61
7.6.11.2	Thread Manager	61
7.6.12	User	61
7.6.12.1	User Options	61
7.6.12.2	Password	62
7.6.12.3	Profile	63
7.7	About	64
7.7.1	About	64

8 Indices and tables 65

Python Module Index 67

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.

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`:

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

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 it in `settings.cfg` file, as well as the name and password of the proxy.

In case you use the login window, you will have a button dedicated to the proxy settings.

You can also set 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 alignak Backend.
; Default is %(url)s:5000
backend = %(url)s:5000

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

; Web service
```

(continues on next page)

(continued from previous page)

```

; -----
; -----
; The url of web service
; Default is %(url)s:8888
webservice = %(url)s:8888

; Backend processes
; -----
; -----
; 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

```

(continues on next page)

(continued from previous page)

```

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

; Tab order
; -----
; -----
; Defines order of App tabs: problems, host synthesis and spy.
; p = problems, h = host synthesis, s = spy,
; Default is p,h,s
tab_order = p,h,s

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

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

```

(continues on next page)

(continued from previous page)

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

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

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

; Update panel Service
; -----
↪-----
; 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 Alignak-App Launcher

alignak-app-launcher command line interface:

```
Usage:
  alignak-app-launcher [-h]
  alignak-app-launcher [-s][--start]
  alignak-app-launcher [-i][--install]

Options:
  -h, --help            Show this screen.
  -s, --start            Start Alignak-app in your current shell. (All platforms)
  -i, --install          Check installation folders and files. (All platforms)
                        Install a daemon file and autocompletion. (Linux only)

Daemon [alignak-app]:
  Option "--install" will create a bin folder in your "$HOME", with a daemon file,
  ↳based on
  the environment variables of your current session.

  Then simply run "alignak-app start" to launch application.

Exit codes:
  0 if required operation succeeded

  22 if application detect an SSH session.
  64 if command line parameters are not used correctly
```

4.2 Linux Daemon

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

Here is the available commands for daemon:

```
Usage:
  # Without parameters
  alignak-app           Displays the help message and configured environment_
  ↪variables.

  # With parameters
  alignak-app start      Start Alignak-app daemon.
  alignak-app stop       Stop Alignak-app daemon.
  alignak-app status     Show status of daemon running.
  alignak-app restart    Stop and restart application.
```

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

4.3 Windows Installer

If you used the installer provided in the repository or did you generate your own, just launch **Alignak-app vX.x** shortcut on your desktop.

All files have been installed in `C:\Program Files\Alignak-app\`.

5.1 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.

5.2 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.

5.3 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.

5.3.1 Alignak

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 be notified here.

You can also see each daemon state by clicking on button with Alignak icon, if endpoint is available in backend.

Finally, you can see your profile by clicking on the button.

5.3.2 Livestate part

In the livestate part, you’ll see a resume of number of problems and items monitored in your backend.

5.3.3 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.
- Some events are temporary and remove themselves.
- Some events can be drag and drop (see below)
- Full events text can be see in tooltip when you keep mouse over.

Drag & Drop:

- Events from an host or a service can be drag and dropped in “Spied Hosts” tab.
- Events can also be moved to Host Synthesis view (Panel), to display host instantly.

5.4 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.

5.4.1 Dashboard

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

5.4.2 Host Synthesis

5.4.2.1 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 informations and the list of its services will be displayed.

5.4.2.2 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.

5.4.2.3 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).

5.4.3 Problems

Problems tab will display problems found by Alignak-app in backend. This panel gathers hosts down, critical services or other issues.

- You'll be able to acknowledge problems or trigger downtimes.
- You can choose to spy an host from any item.
- You can display host concerned in Host Synthesis view.

5.4.4 Spied Hosts

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

- You can spy on as many hosts as you want.
- App will keep you informed about host state.
- When you select a spy host, App will show you the information it has collected.
- To stop spying on a host, just double click on it, host will be removed from list.

You can also **drag & drop** spied elements to display host instantly in Host Synthesis.

5.5 External Tools

5.5.1 WebUI

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.

5.5.2 GLPI

Not yet implemented.

6.1 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.

6.2 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**)

7.1 Main Class

7.1.1 Alignak App

Alignak 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.alignakapp.**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.alignakapp.**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.alignakapp.**AppProgressQWidget** (*parent=None*)

Bases: PyQt5.QtWidgets.QWidget

Class who create a small widget for App start progression

initialize ()

Initialize the QWidget

7.2 Backend

The Backend package contains classes to manage alignak data

7.2.1 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.

acknowledge (*item, sticky, notify, comment*)

Prepare data for acknowledge and POST on backend API or WS if available

Parameters

- **item** (*alignak_app.items.host.Host | alignak_app.items.service.Service*) – item to acknowledge: host | service
- **sticky** (*bool*) – define if ack is sticky or not
- **notify** (*bool*) – define if ack should notify user or not
- **comment** (*str*) – comment of ack

Returns request response

Return type *dict*

downtime (*item, fixed, duration, start_stamp, end_stamp, comment*)

Prepare data for downtime and POST on backend API or WS if available

Parameters

- **item** (*alignak_app.items.host.Host | alignak_app.items.service.Service*) – item to downtime: host | service
- **fixed** (*bool*) – define if downtime is fixed or not

- **duration** (*int*) – duration timestamp of downtime
- **start_stamp** (*int*) – start timestamp of downtime
- **end_stamp** (*int*) – end timestamp of downtime
- **comment** (*str*) – comment of downtime

Returns request response

Return type *dict*

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 request response

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_alignakdaemons ()

Launch request on “alignakdaemon” endpoint

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

Launch request on “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 ()

Launch request on “host” endpoint, add hosts in problems if needed

query_livesynthesis ()

Launch request on “livesynthesis” endpoint

query_notifications ()

Launch request on “history” endpoint. Only for ‘type’: ‘monitoring.notification’ and for current App user

query_realms ()

Launch a request on `realm` endpoint

query_services (*host_id=None*)

Launch request for “service” endpoint. If `host_id` is given, only services related to host are added / updated

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

query_services_problems (*state*)

Launch requests on “service” endpoint to get items with “ls_state = state”

Wanted states are: WARNING, CRITICAL and UNKNOWN

Parameters **state** (*str*) – state of service

query_timeperiods ()

Launch a request on `timeperiod` endpoint

query_user ()

Launch request on “user” endpoint. Only for current App user.

7.2.2 Data Manager

DataManager manage alignak data provided by *Client*.

- `database` field contains all data collected by App
- `db_is_ready` field says to App if database has been filled or not (needed on start)
- `old_notifications` field store old notifications from backend to avoid sending them again

```

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()

Update and 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`

static is_problem(*backend_item*)

Return True if *backend_item* is a problem, else return false

Parameters

- **item_type** (*str*) – type of item: “host” or “service”
- **backend_item** (*dict*) – item of backend

Returns if item is a problem or not

Return type `bool`

is_ready()

Check if DataManager is filled and ready

Returns if ready or current status

Return type `str`

remove_item(*item_type*, *key*, *value=None*)

Remove the wanted item in “database[*item_type*]” who contain the “value” or “key”

Parameters

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

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

update_problems ()

Update hosts and services in “problems” database

7.3 Items

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

7.3.1 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_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_overall_state_icon(services, host_overall)`

Return corresponding icon to max of “_overall_state_id”

Parameters

- **services** (*list*) – list of Service() items
- **host_overall** (*int*) – “_overall_state_id” of host

Returns icon corresponding to state

Return type *str*

7.3.2 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

7.3.3 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

7.3.4 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

7.3.5 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`

get_overall_tooltip (*services*)

Return corresponding overall state tooltip depends of `_overall_state_id` of Host and its Services

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

Returns overall tooltip message

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

7.3.6 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

7.3.7 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

7.3.8 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

7.3.9 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

Parameters `host_id` (*str*) – “_id” of host, needed to get only host services

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

7.3.10 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

7.4 Utils

The Utils package contains classes and functions configurations

7.4.1 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

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`

set_config (*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

`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

7.4.2 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

7.4.3 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

7.4.4 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`

7.4.5 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*

7.5 Locales

The Locales package contains classes to manage translation.

7.5.1 Locales

Locales manage localization of Alignak-app

`alignak_app.locales.locales.init_localization()`

Application localization

Returns gettext translator method

Return type *method*

7.6 QObjects

The QObjects package contains PyQt5 classes of Alignak-app

7.6.1 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


```

mouseMoveEvent (event)
    QWidget.mousePressEvent(QMouseEvent)

mousePressEvent (event)
    QWidget.mousePressEvent(QMouseEvent)

showEvent (_)
    showEvent(self, QShowEvent)

```

7.6.2 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 QWidgets for dock

    initialize ()
        Initialize dock QWidget

```

7.6.3 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 QWidgets

    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 have Qt.UserRole

        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_current_host ()
        Return current Host item with name in QLineEdit

        Returns current host

        Return type alignak_app.items.host.Host

    static get_tab_order ()
        Return tab order defined by user, else default order

```

Returns tab order of App

Return type `list`

initialize()

Create the QWidget with its items and layout.

set_host_from_problems()

Set line search if `sender()` is instance of QPushButton from *Problems* QWidget

spy_host()

Spy host who is available in line_search QLineEdit

7.6.4 Alignak

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

7.6.4.1 Alignak

Alignak manage creation of QWidget's 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` (*parent=None*)

Bases: `PyQt5.QtWidgets.QWidget`

Class who display daemons status, backend connection and user informations

initialize()

Initialize QWidget

show_status_dialog()

Update and show StatusQDialog

show_user_widget()

Update and show ProfileQWidget

update_status()

Update daemons and backend status

update_status_btn (*status*)

Update status button, depending if “status_ok” is False or True

Parameters **status** (*bool*) – current status of alignak daemons

7.6.4.2 Dashboard

Dashboard manage the creation of QWidget's 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 QWidget with number of:

        • Hosts: UP, UNREACHABLE, DOWN

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

        • Hosts and services: NOT_MONITORED, ACKNOWLEDGED, DOWNTIMED

    get_host_resume_widget ()
        Return Host resume QWidget

    get_services_resume_widget ()
        Return Services resume QWidget

    initialize ()
        Initialize QWidget

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

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

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

7.6.4.3 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
```

7.6.4.4 Problems

Problems manage creation of QWidgets 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 QTableWidgetItem and problems title

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

7.6.4.5 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`

7.6.4.6 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

init_daemons_labels (`daemons`)

Initialize the daemon QLabels for each daemons

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

initialize ()

Initialize QDialog

mousePressEvent (`event`)

`QWidget.mousePressEvent(QMouseEvent)`

mousePressEvent (*event*)
QWidget.mousePressEvent(QMouseEvent)

update_dialog ()
Update StatusQDialog labels and return if all daemons are ok or not

Returns if status of daemons is ok or not

Return type `bool`

7.6.5 Common

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

7.6.5.1 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

mouseMoveEvent (*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)

7.6.5.2 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

7.6.5.3 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

7.6.5.4 Frames

Frames manage global QFrames for Alignak-app

`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

7.6.5.5 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

7.6.5.6 Widgets

Widgets manage global QWidgets

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

7.6.6 Events

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

7.6.6.1 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*, *host=None*)

Add event to events list

Parameters

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

- **msg** (*str*) – message of event
- **timer** (*bool*) – timer to hide event at end of time
- **host** (*None* / *str*) – data of a host to set `Qt.UserRole`

event_exist (*msg*)

Check if event already displayed, move it to top and update tooltip. Only for `EventItem` who have a `Qt.UserRole`

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, host=None)`

Add event to `Events QWidget` (access function to make event sendable in whole application. Event type define icon 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** (*bool*) – timer to hide event at end of time
- **host** (*None* / *str*) – `_id` of host

7.6.6.2 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_event_type()
        Return event type depending of data content

        Parameters data (dict) – data of backend item

        Returns event type for item

        Return type str

    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, host=None)
        Initialize QListWidgetItem

        Parameters

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

        • msg (str) – message of event

        • timer (bool) – timer to hide event at end of time

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

7.6.6.3 Spy

Spy manage creation of `QWidget` to manage spied items

```
class alignak_app.qobjects.events.spy.SpyQWidget
    Bases: PyQt5.QtWidgets.QWidget

    Class who create QWidget for spied hosts

    static get_hint_item()
        Return an EventItem with a hint text

        Returns event item with hint text

        Return type EventItem

    initialize()
        Intialize Spy QWidget

    manage_host_events (row)
        Manage spy events for a host, defined by current row of “spy_list_widget”

        Parameters row (int) – current row of “spy_list_widget”
```

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

send_spy_events ()
 Send event messages for all hosts who are spied

update_parent_spytab ()
 Update the parent spy tab text

7.6.7 Host

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

7.6.7.1 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

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

mousePressEvent (*event*)
QWidget.mousePressEvent(QMouseEvent)

paintEvent (*_*)
Override to apply “background-color” property of 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

7.6.7.2 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 (*host_item*)

Set data of host and service

Parameters **host_item** (*alignak_app.items.host.Host*) – the Host item

show_history ()

Create and show HistoryQWidget

update_host (*host_item=None*)

Update HostQWidget data and QLabels

Parameters **host_item** (*alignak_app.items.host.Host*) – the Host item

7.6.8 Login

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

7.6.8.1 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)

7.6.8.2 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

    mouseMoveEvent (event)
        QWidget.mousePressEvent(QMouseEvent)

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

7.6.9 Service

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

7.6.9.1 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)

Update ServiceDataQWidget

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

7.6.9.2 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

update_service_data()

Update ServiceDataQWidget

update_widget(host, services)

Update the QTreeWidget and its items

Parameters

- **host** (*alignak_app.items.host.Host*) – the Host item
- **services** (*list*) – list of *Services* items

7.6.9.3 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

7.6.9.4 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
```

7.6.10 Systray

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

7.6.10.1 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
```

7.6.11 Threads

The Threads package contains QObjects who manage application QThreads

7.6.11.1 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
```

7.6.11.2 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
```

7.6.12 User

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

7.6.12.1 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

7.6.12.2 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

7.6.12.3 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

mousePressEvent (*event*)

QWidget.mousePressEvent(QMouseEvent)

mousePressEvent (*event*)

QWidget.mousePressEvent(QMouseEvent)

paintEvent (_)

Override to apply “background-color” property of 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

7.7 About

7.7.1 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 8

Indices and tables

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

a

- `alignak_app`, 21
- `alignak_app.alignakapp`, 23
- `alignak_app.app`, 15
- `alignak_app.backend`, 24
- `alignak_app.backend.backend`, 24
- `alignak_app.backend.datamanager`, 26
- `alignak_app.items`, 29
 - `alignak_app.items.daemon`, 30
 - `alignak_app.items.event`, 31
 - `alignak_app.items.history`, 32
 - `alignak_app.items.host`, 33
 - `alignak_app.items.item`, 29
 - `alignak_app.items.livesynthesis`, 34
 - `alignak_app.items.period`, 35
 - `alignak_app.items.realm`, 36
 - `alignak_app.items.service`, 36
 - `alignak_app.items.user`, 38
- `alignak_app.locales`, 42
 - `alignak_app.locales.locales`, 42
- `alignak_app.qobjects`, 42
 - `alignak_app.qobjects.about`, 64
 - `alignak_app.qobjects.alignak`, 44
 - `alignak_app.qobjects.alignak.alignak`, 44
 - `alignak_app.qobjects.alignak.dashboard`, 44
 - `alignak_app.qobjects.alignak.livestate`, 45
 - `alignak_app.qobjects.alignak.problems`, 45
 - `alignak_app.qobjects.alignak.problems_table`, 46
 - `alignak_app.qobjects.alignak.status`, 47
 - `alignak_app.qobjects.app_main`, 42
 - `alignak_app.qobjects.common`, 48
 - `alignak_app.qobjects.common.actions`, 48
 - `alignak_app.qobjects.common.buttons`, 49
 - `alignak_app.qobjects.common.dialogs`, 49
 - `alignak_app.qobjects.common.frames`, 51
 - `alignak_app.qobjects.common.labels`, 51
 - `alignak_app.qobjects.common.widgets`, 51
 - `alignak_app.qobjects.dock`, 43
 - `alignak_app.qobjects.events`, 52
 - `alignak_app.qobjects.events.events`, 52
 - `alignak_app.qobjects.events.item`, 54
 - `alignak_app.qobjects.events.spy`, 54
 - `alignak_app.qobjects.host`, 55
 - `alignak_app.qobjects.host.history`, 55
 - `alignak_app.qobjects.host.host`, 56
 - `alignak_app.qobjects.login`, 57
 - `alignak_app.qobjects.login.login`, 57
 - `alignak_app.qobjects.login.server`, 57
 - `alignak_app.qobjects.panel`, 43
 - `alignak_app.qobjects.service`, 58
 - `alignak_app.qobjects.service.service`, 58
 - `alignak_app.qobjects.service.services`, 59
 - `alignak_app.qobjects.service.services_dashboard`, 59
 - `alignak_app.qobjects.service.tree_item`, 59
 - `alignak_app.qobjects.systray`, 60
 - `alignak_app.qobjects.systray.tray_icon`, 60
 - `alignak_app.qobjects.threads`, 60
 - `alignak_app.qobjects.threads.thread`, 60
 - `alignak_app.qobjects.threads.threadmanager`, 61
 - `alignak_app.qobjects.user`, 61
 - `alignak_app.qobjects.user.options`, 61
 - `alignak_app.qobjects.user.password`, 62
 - `alignak_app.qobjects.user.profile`, 62
- `alignak_app.utils`, 38
 - `alignak_app.utils.config`, 38
 - `alignak_app.utils.installer`, 39
 - `alignak_app.utils.logs`, 41
 - `alignak_app.utils.system`, 40

`alignak_app.utils.time`, [41](#)

A

- AboutQDialog (class in alignak_app.qobjects.about), 64
- accept_login() (alignak_app.qobjects.login.login.LoginQDialog method), 57
- accept_text() (alignak_app.qobjects.common.dialogs.EditQDialog method), 49
- accept_text() (alignak_app.qobjects.common.dialogs.ValidatorQDialog method), 50
- acknowledge() (alignak_app.backend.backend.BackendClient method), 24
- AckQDialog (class in alignak_app.qobjects.common.actions), 48
- ActionsQWidget (class in alignak_app.qobjects.common.actions), 48
- add_about_menu() (alignak_app.qobjects.systray.tray_icon.AppTrayIcon method), 60
- add_acknowledge() (alignak_app.qobjects.common.actions.ActionsQWidget method), 48
- add_alignak_menu() (alignak_app.qobjects.systray.tray_icon.AppTrayIcon method), 60
- add_daemon_labels() (alignak_app.qobjects.alignak.status.StatusQDialog method), 47
- add_daemon_titles_labels() (alignak_app.qobjects.alignak.status.StatusQDialog method), 47
- add_downtime() (alignak_app.qobjects.common.actions.ActionsQWidget method), 48
- add_event() (alignak_app.qobjects.events.events.EventsQWidget method), 52
- add_priority_thread() (alignak_app.qobjects.threads.threadmanager.ThreadManager method), 61
- add_quit_menu() (alignak_app.qobjects.systray.tray_icon.AppTrayIcon method), 60
- add_reload_menu() (alignak_app.qobjects.systray.tray_icon.AppTrayIcon method), 60
- add_spied_host() (alignak_app.qobjects.alignak.problems.ProblemsQWidget method), 46
- add_webui_menu() (alignak_app.qobjects.systray.tray_icon.AppTrayIcon method), 60
- alignak_app (module), 21
- alignak_app.alignakapp (module), 23
- alignak_app.app (module), 15
- alignak_app.backend (module), 24
- alignak_app.backend.backend (module), 24
- alignak_app.backend.datamanager (module), 26
- alignak_app.items (module), 29
- alignak_app.items.daemon (module), 30
- alignak_app.items.event (module), 31
- alignak_app.items.history (module), 32
- alignak_app.items.host (module), 33
- alignak_app.items.item (module), 29
- alignak_app.items.livesynthesis (module), 34
- alignak_app.items.period (module), 35
- alignak_app.items.realm (module), 36
- alignak_app.items.service (module), 36
- alignak_app.items.user (module), 38
- alignak_app.locales (module), 42
- alignak_app.locales.locales (module), 42
- alignak_app.qobjects (module), 42
- alignak_app.qobjects.about (module), 64
- alignak_app.qobjects.alignak (module), 44
- alignak_app.qobjects.alignak.alignak (module), 44
- alignak_app.qobjects.alignak.dashboard (module), 44
- alignak_app.qobjects.alignak.livestate (module), 45
- alignak_app.qobjects.alignak.problems (module), 45
- alignak_app.qobjects.alignak.problems_table (module), 46
- alignak_app.qobjects.alignak.status (module), 47
- alignak_app.qobjects.app_main (module), 42
- alignak_app.qobjects.common (module), 48

alignak_app.qobjects.common.actions (module), 48
 alignak_app.qobjects.common.buttons (module), 49
 alignak_app.qobjects.common.dialogs (module), 49
 alignak_app.qobjects.common.frames (module), 51
 alignak_app.qobjects.common.labels (module), 51
 alignak_app.qobjects.common.widgets (module), 51
 alignak_app.qobjects.dock (module), 43
 alignak_app.qobjects.events (module), 52
 alignak_app.qobjects.events.events (module), 52
 alignak_app.qobjects.events.item (module), 54
 alignak_app.qobjects.events.spy (module), 54
 alignak_app.qobjects.host (module), 55
 alignak_app.qobjects.host.history (module), 55
 alignak_app.qobjects.host.host (module), 56
 alignak_app.qobjects.login (module), 57
 alignak_app.qobjects.login.login (module), 57
 alignak_app.qobjects.login.server (module), 57
 alignak_app.qobjects.panel (module), 43
 alignak_app.qobjects.service (module), 58
 alignak_app.qobjects.service.service (module), 58
 alignak_app.qobjects.service.services (module), 59
 alignak_app.qobjects.service.services_dashboard (module), 59
 alignak_app.qobjects.service.tree_item (module), 59
 alignak_app.qobjects.systray (module), 60
 alignak_app.qobjects.systray.tray_icon (module), 60
 alignak_app.qobjects.threads (module), 60
 alignak_app.qobjects.threads.thread (module), 60
 alignak_app.qobjects.threads.threadmanager (module), 61
 alignak_app.qobjects.user (module), 61
 alignak_app.qobjects.user.options (module), 61
 alignak_app.qobjects.user.password (module), 62
 alignak_app.qobjects.user.profile (module), 62
 alignak_app.utils (module), 38
 alignak_app.utils.config (module), 38
 alignak_app.utils.installer (module), 39
 alignak_app.utils.logs (module), 41
 alignak_app.utils.system (module), 40
 alignak_app.utils.time (module), 41
 AlignakApp (class in alignak_app.alignakapp), 23
 AlignakQWidget (class in alignak_app.qobjects.alignak.alignak), 44
 AppProgressBar (class in alignak_app.alignakapp), 23
 AppProgressQWidget (class in alignak_app.alignakapp), 24
 AppQMainWindow (class in alignak_app.qobjects.app_main), 42
 AppTrayIcon (class in alignak_app.qobjects.systray.tray_icon), 60

B

BackendClient (class in alignak_app.backend.backend), 24

BackendQThread (class in alignak_app.qobjects.threads.thread), 61
 build_menu() (alignak_app.qobjects.systray.tray_icon.AppTrayIcon method), 60

C

center_widget() (in module alignak_app.qobjects.common.widgets), 52
 check_connection() (alignak_app.qobjects.systray.tray_icon.AppTrayIcon method), 60
 check_environment() (alignak_app.utils.installer.Installer method), 40
 check_install_files() (alignak_app.utils.installer.Installer method), 40
 check_install_folders() (alignak_app.utils.installer.Installer method), 40
 check_installation() (alignak_app.utils.installer.Installer method), 40
 check_text() (alignak_app.qobjects.common.dialogs.ValidatorQDialog method), 50
 check_threads() (alignak_app.alignakapp.AlignakApp static method), 23
 check_user_installation() (alignak_app.utils.installer.Installer method), 40
 clean_threads() (alignak_app.qobjects.threads.threadmanager.ThreadManager method), 61
 close_item() (alignak_app.qobjects.events.item.EventItem method), 54
 create() (alignak_app.items.daemon.Daemon method), 30
 create() (alignak_app.items.event.Event method), 31
 create() (alignak_app.items.history.History method), 32
 create() (alignak_app.items.host.Host method), 33
 create() (alignak_app.items.item.Item method), 29
 create() (alignak_app.items.livesynthesis.LiveSynthesis method), 34
 create() (alignak_app.items.period.Period method), 35
 create() (alignak_app.items.realm.Realm method), 36
 create() (alignak_app.items.service.Service method), 37
 create() (alignak_app.items.user.User method), 38
 create_logger() (in module alignak_app.utils.logs), 41
 create_widget() (alignak_app.qobjects.login.login.LoginQDialog method), 57

D

Daemon (class in alignak_app.items.daemon), 30
 DashboardQWidget (class in alignak_app.qobjects.alignak.dashboard), 44
 DataManager (class in alignak_app.backend.datamanager), 26
 display_host() (alignak_app.qobjects.panel.PanelQWidget method), 43

- DockQWidget (class in alignak_app.qobjects.dock), 43
- DownQDialog (class in alignak_app.qobjects.common.actions), 48
- downtime() (alignak_app.backend.backend.BackendClient method), 24
- dragEnterEvent() (alignak_app.qobjects.panel.PanelQWidget method), 43
- dragMoveEvent() (alignak_app.qobjects.panel.PanelQWidget method), 43
- dropEvent() (alignak_app.qobjects.panel.PanelQWidget method), 43
- duration_to_seconds() (alignak_app.qobjects.common.actions.DownQDialog method), 48
- ## E
- EditQDialog (class in alignak_app.qobjects.common.dialogs), 49
- enable_notifications() (alignak_app.qobjects.user.profile.ProfileQWidget method), 63
- Event (class in alignak_app.items.event), 31
- event_exist() (alignak_app.qobjects.events.events.EventsQWidget method), 53
- EventItem (class in alignak_app.qobjects.events.item), 54
- EventsQWidget (class in alignak_app.qobjects.events.events), 52
- ## F
- file_executable() (in module alignak_app.utils.system), 40
- ## G
- get() (alignak_app.backend.backend.BackendClient method), 25
- get_actions_widget() (alignak_app.qobjects.host.host.HostQWidget method), 56
- get_actions_widget() (alignak_app.qobjects.service.service.ServiceDataQWidget method), 58
- get_all_hostnames() (alignak_app.backend.datamanager.DataManager method), 27
- get_available_icons() (alignak_app.items.host.Host static method), 33
- get_available_icons() (alignak_app.items.service.Service static method), 37
- get_backend_status_icon() (alignak_app.backend.backend.BackendClient method), 25
- get_btn_widget() (alignak_app.qobjects.alignak.problems.ProblemsQWidget method), 46
- get_check_text() (alignak_app.items.daemon.Daemon static method), 31
- get_check_text() (alignak_app.items.event.Event static method), 32
- get_check_text() (alignak_app.items.history.History static method), 32
- get_check_text() (alignak_app.items.host.Host static method), 33
- get_check_text() (alignak_app.items.item.Item static method), 29
- get_check_text() (alignak_app.items.livesynthesis.LiveSynthesis static method), 34
- get_check_text() (alignak_app.items.period.Period static method), 35
- get_check_text() (alignak_app.items.realm.Realm static method), 36
- get_check_text() (alignak_app.items.service.Service static method), 37
- get_check_text() (alignak_app.items.user.User static method), 38
- get_config() (alignak_app.utils.config.Settings method), 39
- get_curent_user_role_item() (alignak_app.qobjects.alignak.problems.ProblemsQWidget method), 46
- get_current_host() (alignak_app.qobjects.panel.PanelQWidget method), 43
- get_current_time() (in module alignak_app.utils.time), 41
- get_daemons_names() (alignak_app.items.daemon.Daemon static method), 31
- get_daemons_status_icon() (alignak_app.items.daemon.Daemon static method), 31
- get_display_name() (alignak_app.items.host.Host static method), 33
- get_display_name() (alignak_app.items.period.Period static method), 35
- get_display_name() (alignak_app.items.realm.Realm static method), 36
- get_display_name() (alignak_app.items.service.Service static method), 37
- get_event_type() (alignak_app.qobjects.events.item.EventItem static method), 54
- get_event_type() (alignak_app.qobjects.host.history.HistoryQWidget static method), 55
- get_event_widget() (alignak_app.qobjects.host.history.HistoryQWidget method), 55
- get_events() (alignak_app.backend.datamanager.DataManager

method), 27

get_events_widget() (in module alignak_app.qobjects.events.events), 53

get_external_link_label() (alignak_app.qobjects.about.AboutQDialog static method), 64

get_frame_separator() (in module alignak_app.qobjects.common.frames), 51

get_hint_item() (alignak_app.qobjects.events.spy.SpyQWidget static method), 54

get_history_icon_name() (alignak_app.items.history.History static method), 32

get_host_icon_widget() (alignak_app.qobjects.host.host.HostQWidget method), 56

get_host_resume_widget() (alignak_app.qobjects.alignak.dashboard.DashboardQWidget method), 45

get_host_services() (alignak_app.backend.datamanager.DataManager method), 27

get_host_with_services() (alignak_app.backend.datamanager.DataManager method), 27

get_hosts_notif_widget() (alignak_app.qobjects.user.profile.ProfileQWidget method), 63

get_icon() (alignak_app.qobjects.events.item.EventItem static method), 54

get_icon_item() (in module alignak_app.qobjects.common.labels), 51

get_icon_label() (alignak_app.qobjects.host.history.HistoryQWidget static method), 55

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

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

get_icon_pixmap() (in module alignak_app.qobjects.common.labels), 51

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

get_informations_widget() (alignak_app.qobjects.user.profile.ProfileQWidget method), 63

get_item() (alignak_app.backend.datamanager.DataManager method), 27

get_item_text() (alignak_app.qobjects.alignak.problems_table.ProblemsQTableView static method), 46

get_item_type_widget() (alignak_app.qobjects.alignak.livestate.LivestateQWidget method), 45

get_items_and_problems() (alignak_app.backend.datamanager.DataManager method), 27

get_last_check_widget() (alignak_app.qobjects.host.host.HostQWidget method), 56

get_last_check_widget() (alignak_app.qobjects.service.service.ServiceDataQWidget method), 58

get_local_datetime() (in module alignak_app.utils.time), 41

get_local_datetime_from_date() (in module alignak_app.utils.time), 42

get_logo_widget() (in module alignak_app.qobjects.common.widgets), 52

get_message_widget() (alignak_app.qobjects.common.dialogs.MessageQDialog method), 50

get_notes_mail_widget() (alignak_app.qobjects.user.profile.ProfileQWidget method), 63

get_notes_output_widget() (alignak_app.qobjects.host.host.HostQWidget method), 56

get_notifications_widget() (alignak_app.qobjects.user.options.UserOptionsQDialog method), 61

get_notifications_widget() (alignak_app.qobjects.user.profile.ProfileQWidget method), 63

get_output_tableitem() (alignak_app.qobjects.alignak.problems_table.ProblemsQTableView static method), 47

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

get_overall_tooltip() (alignak_app.items.host.Host static method), 33

get_period_name() (alignak_app.backend.datamanager.DataManager method), 27

get_problems() (alignak_app.backend.datamanager.DataManager method), 27

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

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

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

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

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

get_request_model() (alignak_app.items.livesynthesis.LiveSynthesis static method), 34

[get_request_model\(\)](#) (alignak_app.items.period.Period static method), 35
[get_request_model\(\)](#) (alignak_app.items.realm.Realm static method), 36
[get_request_model\(\)](#) (alignak_app.items.service.Service static method), 37
[get_request_model\(\)](#) (alignak_app.items.user.User static method), 38
[get_role\(\)](#) (alignak_app.items.user.User method), 38
[get_search_widget\(\)](#) (alignak_app.qobjects.alignak.problems.ProblemsQWidget method), 46
[get_selected_options\(\)](#) (alignak_app.qobjects.user.options.UserOptionsQDialog static method), 62
[get_service_icon_widget\(\)](#) (alignak_app.qobjects.service.service.ServiceDataQWidget method), 58
[get_service_states_nb\(\)](#) (alignak_app.items.service.Service static method), 37
[get_services_notif_widget\(\)](#) (alignak_app.qobjects.user.profile.ProfileQWidget method), 63
[get_services_resume_widget\(\)](#) (alignak_app.qobjects.alignak.dashboard.DashboardQWidget method), 45
[get_settings_widget\(\)](#) (alignak_app.qobjects.login.server.ServerQDialog method), 58
[get_states\(\)](#) (alignak_app.items.daemon.Daemon static method), 31
[get_synthesis_count\(\)](#) (alignak_app.backend.datamanager.DataManager method), 28
[get_synthesis_count_model\(\)](#) (alignak_app.items.livesynthesis.LiveSynthesis static method), 34
[get_tab_order\(\)](#) (alignak_app.qobjects.panel.PanelQWidget static method), 43
[get_tableitem\(\)](#) (alignak_app.qobjects.alignak.problems_table.ProblemsQTableView method), 47
[get_text_widget\(\)](#) (alignak_app.qobjects.common.dialogs.EditQDialog method), 49
[get_text_widget\(\)](#) (alignak_app.qobjects.common.dialogs.ValidatorQDialog method), 50
[get_threads_to_launch\(\)](#) (alignak_app.qobjects.threads.threadmanager.ThreadManager method), 61
[get_time_diff_since_last_timestamp\(\)](#) (in module alignak_app.utils.time), 42
[get_tooltip\(\)](#) (alignak_app.items.daemon.Daemon method), 31
[get_tooltip\(\)](#) (alignak_app.items.event.Event method), 32
[get_tooltip\(\)](#) (alignak_app.items.history.History method), 33
[get_tooltip\(\)](#) (alignak_app.items.host.Host method), 34
[get_tooltip\(\)](#) (alignak_app.items.item.Item method), 29
[get_tooltip\(\)](#) (alignak_app.items.livesynthesis.LiveSynthesis method), 35
[get_tooltip\(\)](#) (alignak_app.items.period.Period method), 35
[get_tooltip\(\)](#) (alignak_app.items.realm.Realm method), 36
[get_tooltip\(\)](#) (alignak_app.items.service.Service method), 37
[get_tooltip\(\)](#) (alignak_app.items.user.User method), 38
[get_url_endpoint_from_icon_name\(\)](#) (in module alignak_app.utils.config), 39
[get_variables_widget\(\)](#) (alignak_app.qobjects.host.host.HostQWidget method), 56

H

[handle_accept\(\)](#) (alignak_app.qobjects.common.actions.DownQDialog method), 49
[handle_confirm\(\)](#) (alignak_app.qobjects.user.password.PasswordQDialog method), 62
[handle_proxy\(\)](#) (alignak_app.qobjects.login.login.LoginQDialog method), 57
[handle_server\(\)](#) (alignak_app.qobjects.login.login.LoginQDialog static method), 57
[History](#) (class in alignak_app.items.history), 32
[HistoryQWidget](#) (class in alignak_app.qobjects.host.history), 55
[Host](#) (class in alignak_app.items.host), 33
[HostQWidget](#) (class in alignak_app.qobjects.host.host), 56

I

[init_config\(\)](#) (alignak_app.utils.config.Settings method), 35
[init_css\(\)](#) (alignak_app.utils.config.Settings method), 39
[init_daemons_labels\(\)](#) (alignak_app.qobjects.alignak.status.StatusQDialog method), 47
[init_environment\(\)](#) (alignak_app.utils.installer.Installer method), 40
[init_event_widget\(\)](#) (in module alignak_app.qobjects.events.events), 53
[init_localization\(\)](#) (in module alignak_app.locales.locales), 42
[initialize\(\)](#) (alignak_app.alignakapp.AppProgressQWidget method), 24

[initialize\(\) \(alignak_app.qobjects.about.AboutQDialog method\), 64](#)
[initialize\(\) \(alignak_app.qobjects.alignak.alignak.AlignakQWidget method\), 44](#)
[initialize\(\) \(alignak_app.qobjects.alignak.dashboard.DashboardQDialog method\), 45](#)
[initialize\(\) \(alignak_app.qobjects.alignak.livestate.LivestateQWidget method\), 45](#)
[initialize\(\) \(alignak_app.qobjects.alignak.problems.ProblemsQWidget method\), 46](#)
[initialize\(\) \(alignak_app.qobjects.alignak.status.StatusQDialog method\), 47](#)
[initialize\(\) \(alignak_app.qobjects.app_main.AppQMainWindow method\), 42](#)
[initialize\(\) \(alignak_app.qobjects.common.actions.AckQDialog method\), 48](#)
[initialize\(\) \(alignak_app.qobjects.common.actions.ActionsQWidget method\), 48](#)
[initialize\(\) \(alignak_app.qobjects.common.actions.DownQDialog method\), 49](#)
[initialize\(\) \(alignak_app.qobjects.common.buttons.ToggleQWidgetButton method\), 49](#)
[initialize\(\) \(alignak_app.qobjects.common.dialogs.EditQDialog method\), 50](#)
[initialize\(\) \(alignak_app.qobjects.common.dialogs.MessageQDialog method\), 50](#)
[initialize\(\) \(alignak_app.qobjects.common.dialogs.ValidatorQDialog method\), 50](#)
[initialize\(\) \(alignak_app.qobjects.common.widgets.LogoQWidget method\), 51](#)
[initialize\(\) \(alignak_app.qobjects.dock.DockQWidget method\), 43](#)
[initialize\(\) \(alignak_app.qobjects.events.events.EventsQWidget method\), 53](#)
[initialize\(\) \(alignak_app.qobjects.events.item.EventItem method\), 54](#)
[initialize\(\) \(alignak_app.qobjects.events.spy.SpyQWidget method\), 54](#)
[initialize\(\) \(alignak_app.qobjects.host.history.HistoryQWidget method\), 55](#)
[initialize\(\) \(alignak_app.qobjects.host.host.HostQWidget method\), 56](#)
[initialize\(\) \(alignak_app.qobjects.panel.PanelQWidget method\), 44](#)
[initialize\(\) \(alignak_app.qobjects.service.service.ServiceDataQWidget method\), 58](#)
[initialize\(\) \(alignak_app.qobjects.service.services.ServicesQWidget method\), 59](#)
[initialize\(\) \(alignak_app.qobjects.service.services_dashboard.ServicesDashboardQWidget method\), 59](#)
[initialize\(\) \(alignak_app.qobjects.service.tree_item.ServiceTreeItem method\), 60](#)
[initialize\(\) \(alignak_app.qobjects.user.options.UserOptionsQDialog method\), 62](#)
[initialize\(\) \(alignak_app.qobjects.user.password.PasswordQDialog method\), 62](#)
[initialize\(\) \(alignak_app.qobjects.user.profile.ProfileQWidget method\), 63](#)
[initialize\(\) \(alignak_app.qobjects.login.server.ServerQDialog method\), 58](#)
[install\(\) \(alignak_app.utils.installer.Installer method\), 40](#)
[is_widget\(\) \(in module alignak_app.utils.system\), 40](#)
[Installer \(class in alignak_app.utils.installer\), 40](#)
[is_checked\(\) \(alignak_app.qobjects.common.buttons.ToggleQWidgetButton method\), 49](#)
[is_problem\(\) \(alignak_app.backend.datamanager.DataManager static method\), 28](#)
[is_ready\(\) \(alignak_app.backend.datamanager.DataManager method\), 28](#)
[Item class in alignak_app.items.item\), 29](#)
[launch_threads\(\) \(alignak_app.qobjects.threads.threadmanager.ThreadManager method\), 61](#)
[LivestateQWidget \(class in alignak_app.qobjects.alignak.livestate\), 45](#)
[LiveSynthesis \(class in alignak_app.items.livesynthesis\), 34](#)
[login\(\) \(alignak_app.backend.backend.BackendClient method\), 25](#)
[LoginQDialog \(class in alignak_app.qobjects.login.login\), 57](#)
[LogoQWidget \(class in alignak_app.qobjects.common.widgets\), 51](#)
[manage_host_events\(\) \(alignak_app.qobjects.events.spy.SpyQWidget method\), 54](#)
[MessageQDialog \(class in alignak_app.qobjects.common.dialogs\), 50](#)
[minimize\(\) \(alignak_app.qobjects.common.widgets.LogoQWidget method\), 52](#)
[minimize_maximize\(\) \(alignak_app.qobjects.common.widgets.LogoQWidget method\), 52](#)
[mkdir\(\) \(in module alignak_app.utils.system\), 40](#)
[mouseMoveEvent\(\) \(alignak_app.qobjects.alignak.status.StatusQDialog method\), 47](#)
[mouseMoveEvent\(\) \(alignak_app.qobjects.app_main.AppQMainWindow method\), 42](#)
[mouseMoveEvent\(\) \(alignak_app.qobjects.common.actions.AckQDialog method\), 48](#)

- [mouseMoveEvent\(\)](#) (alignak_app.qobjects.common.actions.DownQDialog method), [49](#)
[mouseMoveEvent\(\)](#) (alignak_app.qobjects.host.history.HistoryQWidget method), [55](#)
[mouseMoveEvent\(\)](#) (alignak_app.qobjects.login.login.LoginQDialog method), [57](#)
[mouseMoveEvent\(\)](#) (alignak_app.qobjects.login.server.ServerQDialog method), [58](#)
[mouseMoveEvent\(\)](#) (alignak_app.qobjects.user.profile.ProfileQWidget method), [63](#)
[mousePressEvent\(\)](#) (alignak_app.qobjects.alignak.status.StatusQDialog method), [48](#)
[mousePressEvent\(\)](#) (alignak_app.qobjects.app_main.AppQMainWindow method), [43](#)
[mousePressEvent\(\)](#) (alignak_app.qobjects.common.actions.AckQDialog method), [48](#)
[mousePressEvent\(\)](#) (alignak_app.qobjects.common.actions.DownQDialog method), [49](#)
[mousePressEvent\(\)](#) (alignak_app.qobjects.host.history.HistoryQWidget method), [55](#)
[mousePressEvent\(\)](#) (alignak_app.qobjects.login.login.LoginQDialog method), [57](#)
[mousePressEvent\(\)](#) (alignak_app.qobjects.login.server.ServerQDialog method), [58](#)
[mousePressEvent\(\)](#) (alignak_app.qobjects.user.profile.ProfileQWidget method), [63](#)
- ## O
- [open_item_type_url\(\)](#) (alignak_app.qobjects.alignak.dashboard.DashboardQWidget method), [45](#)
[open_url\(\)](#) (in module alignak_app.utils.config), [39](#)
- ## P
- [paintEvent\(\)](#) (alignak_app.qobjects.alignak.livestate.LivestateQWidget method), [45](#)
[paintEvent\(\)](#) (alignak_app.qobjects.common.widgets.LogoQWidget method), [52](#)
[paintEvent\(\)](#) (alignak_app.qobjects.host.history.HistoryQWidget method), [56](#)
[paintEvent\(\)](#) (alignak_app.qobjects.user.profile.ProfileQWidget method), [63](#)
[PanelQWidget](#) (class in alignak_app.qobjects.panel), [43](#)
[PasswordQDialog](#) (class in alignak_app.qobjects.user.password), [62](#)
[patch\(\)](#) (alignak_app.backend.backend.BackendClient method), [25](#)
[patch_data\(\)](#) (alignak_app.qobjects.host.host.HostQWidget method), [56](#)
[patch_data\(\)](#) (alignak_app.qobjects.user.profile.ProfileQWidget method), [63](#)
[patch_host_checks\(\)](#) (alignak_app.qobjects.host.host.HostQWidget method), [56](#)
[Period](#) (class in alignak_app.items.period), [35](#)
[periodic_refresh\(\)](#) (alignak_app.qobjects.service.service.ServiceDataQWidget method), [58](#)
[post\(\)](#) (alignak_app.backend.backend.BackendClient method), [25](#)
[ProblemsQTableView](#) (class in alignak_app.qobjects.alignak.problems_table), [46](#)
[ProblemsQWidget](#) (class in alignak_app.qobjects.alignak.problems), [46](#)
[ProfileQWidget](#) (class in alignak_app.qobjects.user.profile), [63](#)
- ## Q
- [query_alignakdaemons\(\)](#) (alignak_app.backend.backend.BackendClient method), [26](#)
[query_history\(\)](#) (alignak_app.backend.backend.BackendClient method), [26](#)
[query_hosts\(\)](#) (alignak_app.backend.backend.BackendClient method), [26](#)
[query_livesynthesis\(\)](#) (alignak_app.backend.backend.BackendClient method), [26](#)
[query_notifications\(\)](#) (alignak_app.backend.backend.BackendClient method), [26](#)
[query_realms\(\)](#) (alignak_app.backend.backend.BackendClient method), [26](#)
[query_services\(\)](#) (alignak_app.backend.backend.BackendClient method), [26](#)
[query_services_problems\(\)](#) (alignak_app.backend.backend.BackendClient method), [26](#)
[query_timeperiods\(\)](#) (alignak_app.backend.backend.BackendClient method), [26](#)
[query_user\(\)](#) (alignak_app.backend.backend.BackendClient method), [26](#)

quit_app() (alignak_app.qobjects.systray.tray_icon.AppTrayIcon method), 44
 static method), 60
 quit_launched_threads() (alignak_app.qobjects.login.login.LoginQDialog static method), 57
 set_proxy_settings() (alignak_app.qobjects.login.login.LoginQDialog method), 57
 set_text() (alignak_app.alignakapp.AppProgressBar method), 24
 Settings (class in alignak_app.utils.config), 39
 show_about() (alignak_app.qobjects.about.AboutQDialog method), 64
 show_history() (alignak_app.qobjects.host.host.HostQWidget method), 57
 show_login_window() (alignak_app.alignakapp.AlignakApp static method), 23
 show_options_dialog() (in module alignak_app.qobjects.user.options), 62
 show_status_dialog() (alignak_app.qobjects.alignak.alignak.AlignakQWidget method), 44
 show_token_dialog() (alignak_app.qobjects.user.profile.ProfileQWidget static method), 64
 show_user_widget() (alignak_app.qobjects.alignak.alignak.AlignakQWidget method), 44
 showEvent() (alignak_app.qobjects.app_main.AppQMainWindow method), 43
 showEvent() (alignak_app.qobjects.login.login.LoginQDialog method), 57
 spy_host() (alignak_app.qobjects.panel.PanelQWidget method), 44
 SpyQWidget (class in alignak_app.qobjects.events.spy), 54
 start() (alignak_app.alignakapp.AlignakApp method), 23
 StatusQDialog (class in alignak_app.qobjects.alignak.status), 47
 stop_priority_threads() (alignak_app.qobjects.threads.threadmanager.ThreadManager method), 61
 stop_threads() (alignak_app.qobjects.threads.threadmanager.ThreadManager method), 61
 text() (alignak_app.alignakapp.AppProgressBar method), 24
 ThreadManager (class in alignak_app.qobjects.threads.threadmanager), 61
 ToggleQWidgetButton (class in alignak_app.qobjects.common.buttons), 49
 update_action_buttons() (alignak_app.qobjects.alignak.problems.ProblemsQWidget method), 49

R

read_config_file() (in module alignak_app.utils.system), 41
 Realm (class in alignak_app.items.realm), 36
 refresh_menus() (alignak_app.qobjects.systray.tray_icon.AppTrayIcon method), 60
 reload_configuration() (alignak_app.qobjects.systray.tray_icon.AppTrayIcon static method), 60
 remove_event() (alignak_app.qobjects.events.events.EventsQWidget method), 53
 remove_event() (alignak_app.qobjects.events.spy.SpyQWidget method), 55
 remove_item() (alignak_app.backend.datamanager.DataManager method), 28
 remove_timer_event() (alignak_app.qobjects.events.events.EventsQWidget method), 53
 run() (alignak_app.qobjects.threads.thread.BackendQThread method), 61

S

send_datamanager_events() (alignak_app.qobjects.events.events.EventsQWidget method), 53
 send_event() (in module alignak_app.qobjects.events.events), 53
 send_spy_events() (alignak_app.qobjects.events.spy.SpyQWidget method), 55
 ServerQDialog (class in alignak_app.qobjects.login.server), 58
 Service (class in alignak_app.items.service), 37
 ServiceDataQWidget (class in alignak_app.qobjects.service.service), 58
 ServicesDashboardQWidget (class in alignak_app.qobjects.service.services_dashboard), 59
 ServicesQWidget (class in alignak_app.qobjects.service.services), 59
 ServiceTreeWidgetItem (class in alignak_app.qobjects.service.tree_item), 59
 set_config() (alignak_app.utils.config.Settings method), 39
 set_data() (alignak_app.qobjects.host.host.HostQWidget method), 57
 set_host_from_problems() (alignak_app.qobjects.panel.PanelQWidget method), 49

T

U

method), 46

update_btn_state() (alignak_app.qobjects.common.buttons.ToggleQWidgetButton method), 49

update_dashboard() (alignak_app.qobjects.alignak.dashboard.DashboardQWidget method), 45

update_data() (alignak_app.items.daemon.Daemon method), 31

update_data() (alignak_app.items.event.Event method), 32

update_data() (alignak_app.items.history.History method), 33

update_data() (alignak_app.items.host.Host method), 34

update_data() (alignak_app.items.item.Item method), 29

update_data() (alignak_app.items.livesynthesis.LiveSynthesis method), 35

update_data() (alignak_app.items.period.Period method), 35

update_data() (alignak_app.items.realm.Realm method), 36

update_data() (alignak_app.items.service.Service method), 37

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

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

update_dialog() (alignak_app.qobjects.alignak.status.StatusQDialog method), 48

update_history_data() (alignak_app.qobjects.host.history.HistoryQWidget method), 56

update_host() (alignak_app.qobjects.host.host.HostQWidget method), 57

update_item() (alignak_app.qobjects.service.tree_item.ServiceTreeItem method), 60

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

update_labels() (alignak_app.qobjects.alignak.livestate.LivestateQWidget method), 45

update_parent_spytab() (alignak_app.qobjects.events.spy.SpyQWidget method), 55

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

update_problems_data() (alignak_app.qobjects.alignak.problems.ProblemsQWidget method), 46

update_service_data() (alignak_app.qobjects.service.services.ServicesQWidget method), 59

update_status() (alignak_app.qobjects.alignak.alignak.AlignakQWidget method), 44

update_status_btn() (alignak_app.qobjects.alignak.alignak.AlignakQWidget method), 44

update_view() (alignak_app.qobjects.alignak.problems_table.ProblemsQTable method), 47

update_widget() (alignak_app.qobjects.common.actions.ActionsQWidget method), 48

update_widget() (alignak_app.qobjects.service.service.ServiceDataQWidget method), 59

update_widget() (alignak_app.qobjects.service.services.ServicesQWidget method), 59

update_widget() (alignak_app.qobjects.service.services_dashboard.ServicesDashboardQWidget method), 59

update_widget() (alignak_app.qobjects.user.profile.ProfileQWidget method), 64

User (class in alignak_app.items.user), 38

UserOptionsQDialog (class in alignak_app.qobjects.user.options), 61

V

ValidatorQDialog (class in alignak_app.qobjects.common.dialogs), 50

W

write_file() (in module alignak_app.utils.system), 41

write_rc_file() (in module alignak_app.utils.system), 41