
ClockAlarm Documentation

Release 1.4.0

Loïc Charrière, Samuel Gauthier

Jun 16, 2017

Contents

1	Features	3
2	Installation	5
2.1	Requirements	5
3	Development	7
3.1	Requirements	7
4	Bugs/Requests	9
5	ClockAlarm API Documentation	11
5.1	_clockalarm	11
5.1.1	_clockalarm package	11
5.1.1.1	Subpackages	11
5.1.1.2	Submodules	15
5.1.1.3	_clockalarm.Alert module	15
5.1.1.4	_clockalarm.AlertCollection module	16
5.1.1.5	_clockalarm.Clock module	17
5.1.1.6	_clockalarm.Notification module	17
5.1.1.7	_clockalarm.NotificationCenter module	18
5.1.1.8	_clockalarm.SimpleAlert module	19
5.1.1.9	_clockalarm.main module	19
5.1.1.10	Module contents	20
6	Indices and tables	21
	Python Module Index	23

ClockAlarm is a cross-platform (linux, Windows, macOS) alarm manager that will help you improve your life organization. Define alerts and never miss events again.

CHAPTER 1

Features

- Simple / Periodic alerts
- Works on Windows, macOS and Linux
- Safe, corrupt-free and humanly readable database
- Fully customizable alerts (sound, color, font, etc)

You will need a working python environment.

2.1 Requirements

- python => 3.6
- pygame => 1.9.3
- PyQt5 => 5.8.2
- sip => 4.19.2
- tinydb => 3.2.2

You can install these packages using the following command:

```
$ pip3 install pygame pyqt5 sip tinydb
```

Clone the repository:

```
$ git clone https://github.com/BFH-BTI7301-project1/ClockAlarm.git
```

Launch ClockAlarm:

```
$ python3 bin/clockalarm&
```


3.1 Requirements

- coverage >= 4.4.1
- pytest >= 3.0.7
- pytest-cov >= 2.5.1
- pytest-qt >= 2.1.0
- pytest-catchlog >= 1.2.2
- coveralls >= 1.1

You can install these packages using the following command:

```
$ pip3 install -r stable-req.txt
```

Run the tests:

```
$ py.test --cov-report term --cov=._clockalarm/_tests --no-xvfb
```

[GitHub project page](#)

CHAPTER 4

Bugs/Requests

Please use the [GitHub issue tracker](#) to submit bugs or request features.

5.1 `_clockalarm`

5.1.1 `_clockalarm` package

5.1.1.1 Subpackages

`_clockalarm.UI` package

Submodules

`_clockalarm.UI.AlertListWidget` module

class `_clockalarm.UI.AlertListWidget.AlertListWidget`

Bases: `PyQt5.QtWidgets.QTableWidget`

Visual list displaying the Alerts from the database.

actualize (*alert_list*)

Actualizes the graphical alert list based on the list given in argument.

alert_list

The list containing all the alerts.

init_ui ()

Initialization helper method.

Creates a visual empty grid with column headers ID, Time, Periodicity and Message. If alerts are in the db, fills the list with alerts and sorts them.

`_clockalarm.UI.ColorSelectorWidget` module

class `_clockalarm.UI.ColorSelectorWidget.ColorSelectorWidget` (*hex_color=None, parent=None*)

Bases: `PyQt5.QtWidgets.QWidget`

Custom widget to selected a color for the notification

button_click ()

Called when the `color_select_button` is clicked.

Choose a color from a palette.

change_event ()

Called every times the `QLineEdit` field is updated.

Update the `hex_color` value and the color of the `QPushButton`.

init_ui ()

Initialize the GUI of the `QWidget`

set_hex_color (*hex_color*)

Set a new hexadecimal color

hex_color

str – The new color.

`_clockalarm.UI.MainWindow` module

class `_clockalarm.UI.MainWindow.MainWindow` (*application, *args*)

Bases: `PyQt5.QtWidgets.QMainWindow`

Main window interface extending `PyQt5.QtWidgets.QMainWindow`.

Shows a list of all the alerts retrieved from a JSON file.

application

The main application

***args**

The parent pointer or window flags

add_simple_alert ()

Creates a `SimpleAlertEditWidget` and shows it to the user.

closeEvent (*event*)

Overrides `closeEvent` () method.

Allows to intercept the window closing event.

delete_alerts ()

Deletes the selected alerts.

edit_simple_alert ()

Edits the selected alert and shows an `SimpleAlertEditWidget` to the user.

static export_json_db ()

Exports all the Alerts into a JSON file.

static import_json_db ()

Imports Alerts from a JSON file into ClockAlarm.


```

init_ui ()
    Init helper method to set up the main window.

mute_button_click ()
    Mutes the alerts and updates the mute button.

tray_icon_click (reason)
    Displays the main window when clicking on the app icon in the tray.

```

`_clockalarm.UI.NotificationWidget` module

```

class _clockalarm.UI.NotificationWidget.NotificationWidget (geometry, notification,
                                                         parent=None)

    Bases: PyQt5.QtWidgets.QWidget

    Notification widget

    geometry
        The position and size of the widget on the screen

    notification
        The notification

    init_ui (geom)
        Helper method that sets the style of the NotificationWidget.

    geom
        The position and size of the widget on the screen

    mousePressEvent (event)
        Override of :class:`~PyQt5.QtWidgets.QWidget.mousePressEvent` method

    popup_close

```

`_clockalarm.UI.SimpleAlertEditWidget` module

```

class _clockalarm.UI.SimpleAlertEditWidget.SimpleAlertEditWidget (alert: _clock-
                                                                    alarm.SimpleAlert.SimpleAlert
                                                                    = None)

    Bases: PyQt5.QtWidgets.QWidget

    Widget allowing to create or edit a SimpleAlert.

    alert
        A SimpleAlert to edit, default None

    init_ui (alert)
        Initialization helper method.

    alert
        The SimpleAlert to edit

```

`_clockalarm.UI.SoundSelectorWidget` module

```

class _clockalarm.UI.SoundSelectorWidget.SoundSelectorWidget (sound_name: str =
                                                                None, parent=None)

    Bases: PyQt5.QtWidgets.QWidget

    Costum widget to selected a sound

```

Select a wave file and import it in the application sound folder.

button_click()

Called when the sound_select_button is clicked. Get a wave sound file path from the file explorer.

init_ui()

Initialize the GUI of the QWidget

load_sound(sound_path: str)

Load the given .wave sound file in the application folder

If the file already exist in the sound folder of the app, nothing append. If a file with the same name already exist in the sound folder, he is overwritten.

sound_path

the full path of the sound file.

Exceptions: ValueError: If the given sound_path isn't a valid wave sound.

set_sound(sound: str)

Set a new sound

Modify the name and update the edit field.

sound

str – The new sound basename.

Exceptions: ValueError: If the sound_name argument isn't a wave file.

Module contents

[_clockalarm.resources package](#)

Module contents

[_clockalarm.utils package](#)

Submodules

[_clockalarm.utils.importExportUtils module](#)

`_clockalarm.utils.importExportUtils.export_alerts_file(dest)`

Copy the file at ALERT_DB_PATH into the .json location specified the argument

If existing, the destination file is erased.

Parameters **dest** (str) – full path of the destination location

Exceptions: FileNotFoundError: If the given destination file doesn't exist or haven't a json extension
shutil.SameFileError: If the src and destination files are the same

`_clockalarm.utils.importExportUtils.get_default_config(key, cmd='str')`

Read the configuration file and return the value for the given key

The cmd attribute defines the type of the value. Default is str.

`_clockalarm.utils.importExportUtils.key`

The key of the key-value pair in the config file

`_clockalarm.utils.importExportUtils.cmd`

Type of the desired value. Default is str.

Exceptions: `KeyError`: Is the key doesn't match any key in the config file. `configparser.NoSectionError`: If the config file or the section doesn't exist.

`_clockalarm.utils.importExportUtils.import_alerts_file(src)`

Copy the .json file given as argument into the location specified by `ALERT_DB_PATH`

If existing, the destination file is erased. All the application is restarted after an import.

Parameters `src (str)` – full path of the source json database

Exceptions: `FileNotFoundError`: If the given source file doesn't exist or isn't a json file `SameFileError`: If the src and destination files are the same

`_clockalarm.utils.importExportUtils.set_default_config(key, value)`

Set the given (key,value) pair in the config file.

Preserves the Caps

`_clockalarm.utils.importExportUtils.key`

the key

`_clockalarm.utils.importExportUtils.value`

the value

Exceptions: `configparser.NoSectionError`: If the config file or the section doesn't exist.

Module contents

5.1.1.2 Submodules

5.1.1.3 `_clockalarm.Alert` module

`class _clockalarm.Alert.Alert(trigger_time)`

Bases: `PyQt5.QtCore.QObject`

Abstract Alert class representing alerts.

trigger_time

The trigger time of the alert.

id

The identification number of the alert in the alert.

Set up when added to the database.

get_id()

Get the Alert identifier.

Returns The identifier of the alert.

get_trigger_time()

Get the Alert trigger time in seconds since epoch.

Returns The trigger time

timeout

triggered()

Triggers the alert action.

class `_clockalarm.Alert.AlertMeta`

Bases: `sip.wrappertype, abc.ABC`

Abstract Metaclass inheriting QObject's metaclass

5.1.1.4 `_clockalarm.AlertCollection` module

class `_clockalarm.AlertCollection.AlertCollection` (*parent=None*)

Bases: `object`

Collection of all the Alert objects running on the program.

This class contains utilities to load, update and save the alerts in a TinyDB database, and maintain the correct state of all the alerts

add (*alert: _clockalarm.SimpleAlert.SimpleAlert*)

Add the Alert given in argument to the collection of alerts

If parent is set, the new Alert is connected to the notification center and the list of displayed alerts is updated.

alert

SimpleAlert – The alert to add to the collection

Exceptions: `ValueError`: If the alert argument is `None` or incorrect.

check_timers (*trig_time*)

Check all the alerts to see if on should be triggered

This function is triggered periodically by the clock.

trig_time

Actual time, in seconds.

clean_db ()

Make the TinyDB database consistent

All the outdated alerts without periodicity are removed. New trigger time is calculated for outdated alerts with periodicity.

If the db is corrupted, nothing append.

delete (*id_alert: int*)

Remove an alert from the collection

id_alert

The id number of the alert to remove.

Exceptions: `KeyError`: If the alert to delete doesn't exist in the database.

display ()

Actualize the UI alert list display

If parent is unset, the `AlertCollection` object isn't link to any `QWindow` and nothing append

edit (*id_alert: int, notification: _clockalarm.Notification.Notification = None, trigger_time: int = None, periodicity: int = None*)

Update an alert with the given modifications

If the trigger_time is in the past, he won't re updated.

id_alert

int – The id number of the alert to modify

notification

Notification, optional – Default is None. The new notification

trigger_time

int, optional – Default is None. The new trigger time

periodicity(int, optional

Default is None. The new periodicity

Exceptions: `KeyError`: If the alert to edit doesn't exist in the database. `ValueError`: If the periodicity argument is under zero.

load_db()

Fill the AlertCollection with Alerts form the TinyDB database

Exceptions: `IOError`: If a required parameter can't be found in the database. The database is probably corrupted.

5.1.1.5 _clockalarm.Clock module

class _clockalarm.Clock.**Clock** (*frequency, parent=None*)

Bases: `PyQt5.QtCore.QThread`

Clock emitting ticks with a given periodicity.

The class extends `QThread` to allow threading.

frequency

The frequency at which the clock emits a tick.

parent

The parent. Default value is None.

run()

Start the clock.

stop()

Stop the clock.

tick

5.1.1.6 _clockalarm.Notification module

class _clockalarm.Notification.**Notification** (*message: str, color_hex: str = None, font_family: str = None, font_size: int = None, sound: str = None*)

Bases: `object`

Encapsulate the parameters of a Notification in order to pass it to the NotificationCenter

get_color()

Build a QColor from color_hex parameter

If color_hex parameter is missing, replace it with de default configuration.

Returns The QColor of the notification

get_font()

Build a QFont from font_family and font_size

If font_family or font_size parameter is missing, replace it with de default configuration.

Returns The QFont of the notification

get_message()

Getter for message

Returns The notification text

get_sound()

Build a mixer.Sound object from sound parameter

If the sound path parameter is missing, replace it with de default configuration.

Returns The mixer.Sound file of the notification

5.1.1.7 _clockalarm.NotificationCenter module

class _clockalarm.NotificationCenter.NotificationCenter (*screen_geometry*, *parent=None*)

Bases: `object`

Class handling the display of the Notification Widgets

Receives the notifications to display and add it to a queue. If there is free slot on the UI display area, pop the queue and display the notification. If the user clicks on the widget, closes it and compacts the remaining widgets.

add_to_queue (*notification*)

Add a new notification to the queue

The notification will wait till there is a free spot in the display zone

notification

Notification – The notification to add to enqueue

Exceptions: ValueError: If the user try to add None or incorrect Notification object to the queue

close_popup (*popup: _clockalarm.UI.NotificationWidget.NotificationWidget*)

Triggered by a NotificationWidget when closed

Remove the popup from the list and refresh the display zone.

popup

The NotificationWidget to remove from the display zone

Exceptions: ValueError: If the user pass a None or incorrect popup object as argument. KeyError: If the popup isn't in the list and can't be removed

display_popup (*geom: PyQt5.QtCore.QRect, notification*)

Display a QWidget popup

Play a sound if the program isn't muted

geom

QRect – position and size of the widget on the screen

notification

Notification – the notification to display

Exceptions: `ValueError`: In case of None or invalid `QRect` object as `geom` argument `ValueError`: In case of None or invalid `Notification` object as `notification` argument `IndexError`: If the number of displayed popups is greater or equal to the max number of displayed popups

refresh()

Refresh the display off notifications in the display zone

Compact the remaining notifications in the display zone, if there is free slots, pop the notification queue and display the popup.

5.1.1.8 `_clockalarm.SimpleAlert` module

class `_clockalarm.SimpleAlert.SimpleAlert` (*trigger_time, notification, periodicity=None*)

Bases: `_clockalarm.Alert.Alert`

Simple Alert implementation with text message

trigger_time

The time at which the alert is triggered.

notification

The notification to display when the alert is triggered.

periodicity

frequency in which the alert is displayed, in seconds.

get_notification()

Get the Notification

Returns The Notification of the Alert

get_periodicity()

Get the SimpleAlert periodicity

Returns The periodicity

triggered()

This method does the same as `triggered()`

5.1.1.9 `_clockalarm.main` module

class `_clockalarm.main.App` (*default_config_path, alert_db_path, *argv*)

Bases: `PyQt5.QtWidgets.QApplication`

Main Application extending `QApplication`

CLOCK_FREQUENCY = `None`

MUTE = `None`

init_alert_collection()

Init `AlertCollection` object and connect the clock to it

Note: The alerts will be loaded from the default database

init_clock (*freq*)

Init and start the Clock with the given frequency

freq

frequency of the clock's ticks

init_config ()

Open the config_test.cfg file and use a parser to retrieve default values

init_ui ()

Initialisation of the main window GUI

`_clockalarm.main.main` (*argv*)

Main function called when application starts

`_clockalarm.main.argv`

optional – path of the config file and the alertDB file usage: program configFile alrtDBFile If no argument entered, default is config_test.cfg and alertDB.json

Returns the system exit code

5.1.1.10 Module contents

CHAPTER 6

Indices and tables

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

—
_clockalarm, 20
_clockalarm.Alert, 15
_clockalarm.AlertCollection, 16
_clockalarm.Clock, 17
_clockalarm.main, 19
_clockalarm.Notification, 17
_clockalarm.NotificationCenter, 18
_clockalarm.resources, 14
_clockalarm.SimpleAlert, 19
_clockalarm.UI, 14
_clockalarm.UI.AlertListWidget, 11
_clockalarm.UI.ColorSelectorWidget, 12
_clockalarm.UI.MainWindow, 12
_clockalarm.UI.NotificationWidget, 13
_clockalarm.UI.SimpleAlertEditWidget,
13
_clockalarm.UI.SoundSelectorWidget, 13
_clockalarm.utils, 15
_clockalarm.utils.importExportUtils, 14

Symbols

[_clockalarm \(module\), 20](#)
[_clockalarm.Alert \(module\), 15](#)
[_clockalarm.AlertCollection \(module\), 16](#)
[_clockalarm.Clock \(module\), 17](#)
[_clockalarm.Notification \(module\), 17](#)
[_clockalarm.NotificationCenter \(module\), 18](#)
[_clockalarm.SimpleAlert \(module\), 19](#)
[_clockalarm.UI \(module\), 14](#)
[_clockalarm.UI.AlertListWidget \(module\), 11](#)
[_clockalarm.UI.ColorSelectorWidget \(module\), 12](#)
[_clockalarm.UI.MainWindow \(module\), 12](#)
[_clockalarm.UI.NotificationWidget \(module\), 13](#)
[_clockalarm.UI.SimpleAlertEditWidget \(module\), 13](#)
[_clockalarm.UI.SoundSelectorWidget \(module\), 13](#)
[_clockalarm.main \(module\), 19](#)
[_clockalarm.resources \(module\), 14](#)
[_clockalarm.utils \(module\), 15](#)
[_clockalarm.utils.importExportUtils \(module\), 14](#)

A

[actualize\(\) \(_clockalarm.UI.AlertListWidget.AlertListWidget method\), 11](#)
[add\(\) \(_clockalarm.AlertCollection.AlertCollection method\), 16](#)
[add_simple_alert\(\) \(_clockalarm.UI.MainWindow.MainWindow method\), 12](#)
[add_to_queue\(\) \(_clockalarm.NotificationCenter.NotificationCenter method\), 18](#)
[alert \(_clockalarm.AlertCollection.AlertCollection attribute\), 16](#)
[alert \(_clockalarm.UI.SimpleAlertEditWidget.SimpleAlertEditWidget attribute\), 13](#)
[Alert \(class in _clockalarm.Alert\), 15](#)
[alert_list \(_clockalarm.UI.AlertListWidget.AlertListWidget attribute\), 11](#)

[AlertCollection \(class in _clockalarm.AlertCollection\), 16](#)
[AlertListWidget \(class in _clockalarm.UI.AlertListWidget\), 11](#)
[AlertMeta \(class in _clockalarm.Alert\), 16](#)
[App \(class in _clockalarm.main\), 19](#)
[application \(_clockalarm.UI.MainWindow.MainWindow attribute\), 12](#)
[argv \(in module _clockalarm.main\), 20](#)

B

[button_click\(\) \(_clockalarm.UI.ColorSelectorWidget.ColorSelectorWidget method\), 12](#)
[button_click\(\) \(_clockalarm.UI.SoundSelectorWidget.SoundSelectorWidget method\), 14](#)

C

[change_event\(\) \(_clockalarm.UI.ColorSelectorWidget.ColorSelectorWidget method\), 12](#)
[check_timers\(\) \(_clockalarm.AlertCollection.AlertCollection method\), 16](#)
[clean_db\(\) \(_clockalarm.AlertCollection.AlertCollection method\), 16](#)
[Clock \(class in _clockalarm.Clock\), 17](#)
[CLOCK_FREQUENCY \(_clockalarm.main.App attribute\), 19](#)
[close_popup\(\) \(_clockalarm.NotificationCenter.NotificationCenter method\), 18](#)
[closeEvent\(\) \(_clockalarm.UI.MainWindow.MainWindow method\), 12](#)
[cmd \(in module _clockalarm.utils.importExportUtils\), 15](#)
[ColorSelectorWidget \(class in _clockalarm.UI.ColorSelectorWidget\), 12](#)

D

[delete\(\) \(_clockalarm.AlertCollection.AlertCollection method\), 16](#)

delete_alerts() (_clockalarm.UI.MainWindow.MainWindow method), 12

display() (_clockalarm.AlertCollection.AlertCollection method), 16

display_popup() (_clockalarm.NotificationCenter.NotificationCenter method), 18

E

edit() (_clockalarm.AlertCollection.AlertCollection method), 16

edit_simple_alert() (_clockalarm.UI.MainWindow.MainWindow method), 12

export_alerts_file() (in module _clockalarm.utils.importExportUtils), 14

export_json_db() (_clockalarm.UI.MainWindow.MainWindow static method), 12

F

freq (_clockalarm.main.App attribute), 20

frequency (_clockalarm.Clock.Clock attribute), 17

G

geom (_clockalarm.NotificationCenter.NotificationCenter attribute), 18

geom (_clockalarm.UI.NotificationWidget.NotificationWidget attribute), 13

geometry (_clockalarm.UI.NotificationWidget.NotificationWidget attribute), 13

get_color() (_clockalarm.Notification.Notification method), 17

get_default_config() (in module _clockalarm.utils.importExportUtils), 14

get_font() (_clockalarm.Notification.Notification method), 18

get_id() (_clockalarm.Alert.Alert method), 15

get_message() (_clockalarm.Notification.Notification method), 18

get_notification() (_clockalarm.SimpleAlert.SimpleAlert method), 19

get_periodicity() (_clockalarm.SimpleAlert.SimpleAlert method), 19

get_sound() (_clockalarm.Notification.Notification method), 18

get_trigger_time() (_clockalarm.Alert.Alert method), 15

H

hex_color (_clockalarm.UI.ColorSelectorWidget.ColorSelectorWidget attribute), 12

I

id (_clockalarm.Alert.Alert attribute), 15

id_alert (_clockalarm.AlertCollection.AlertCollection attribute), 16, 17

import_alerts_file() (in module _clockalarm.utils.importExportUtils), 15

import_json_db() (_clockalarm.UI.MainWindow.MainWindow static method), 12

init_alert_collection() (_clockalarm.main.App method), 19

init_clock() (_clockalarm.main.App method), 20

init_config() (_clockalarm.main.App method), 20

init_ui() (_clockalarm.main.App method), 20

init_ui() (_clockalarm.UI.AlertListWidget.AlertListWidget method), 11

init_ui() (_clockalarm.UI.ColorSelectorWidget.ColorSelectorWidget method), 12

init_ui() (_clockalarm.UI.MainWindow.MainWindow method), 12

init_ui() (_clockalarm.UI.NotificationWidget.NotificationWidget method), 13

init_ui() (_clockalarm.UI.SimpleAlertEditWidget.SimpleAlertEditWidget method), 13

init_ui() (_clockalarm.UI.SoundSelectorWidget.SoundSelectorWidget method), 14

K

key (in module _clockalarm.utils.importExportUtils), 14, 15

L

load_db() (_clockalarm.AlertCollection.AlertCollection method), 17

load_sound() (_clockalarm.UI.SoundSelectorWidget.SoundSelectorWidget method), 14

M

main() (in module _clockalarm.main), 20

MainWindow (class in _clockalarm.UI.MainWindow), 12

mousePressEvent() (_clockalarm.UI.NotificationWidget.NotificationWidget method), 13

MUTE (_clockalarm.main.App attribute), 19

mute_button_click() (_clockalarm.UI.MainWindow.MainWindow method), 13

N

notification (_clockalarm.AlertCollection.AlertCollection attribute), 17

notification (_clockalarm.NotificationCenter.NotificationCenter attribute), 18, 19

notification (`_clockalarm.SimpleAlert.SimpleAlert` attribute), 19

notification (`_clockalarm.UI.NotificationWidget.NotificationWidget` attribute), 13

Notification (class in `_clockalarm.Notification`), 17

NotificationCenter (class in `_clockalarm.NotificationCenter`), 18

NotificationWidget (class in `_clockalarm.UI.NotificationWidget`), 13

trigger_time (`_clockalarm.SimpleAlert.SimpleAlert` attribute), 19

triggered() (`_clockalarm.Alert.Alert` method), 16

triggered() (`_clockalarm.SimpleAlert.SimpleAlert` method), 19

V

value (in module `_clockalarm.utils.importExportUtils`), 15

P

parent (`_clockalarm.Clock.Clock` attribute), 17

periodicity (`_clockalarm.SimpleAlert.SimpleAlert` attribute), 19

popup (`_clockalarm.NotificationCenter.NotificationCenter` attribute), 18

popup_close (`_clockalarm.UI.NotificationWidget.NotificationWidget` attribute), 13

R

refresh() (`_clockalarm.NotificationCenter.NotificationCenter` method), 19

run() (`_clockalarm.Clock.Clock` method), 17

S

set_default_config() (in module `_clockalarm.utils.importExportUtils`), 15

set_hex_color() (`_clockalarm.UI.ColorSelectorWidget.ColorSelectorWidget` method), 12

set_sound() (`_clockalarm.UI.SoundSelectorWidget.SoundSelectorWidget` method), 14

SimpleAlert (class in `_clockalarm.SimpleAlert`), 19

SimpleAlertEditWidget (class in `_clockalarm.UI.SimpleAlertEditWidget`), 13

sound (`_clockalarm.UI.SoundSelectorWidget.SoundSelectorWidget` attribute), 14

sound_path (`_clockalarm.UI.SoundSelectorWidget.SoundSelectorWidget` attribute), 14

SoundSelectorWidget (class in `_clockalarm.UI.SoundSelectorWidget`), 13

stop() (`_clockalarm.Clock.Clock` method), 17

T

tick (`_clockalarm.Clock.Clock` attribute), 17

timeout (`_clockalarm.Alert.Alert` attribute), 15

tray_icon_click() (`_clockalarm.UI.MainWindow.MainWindow` method), 13

trig_time (`_clockalarm.AlertCollection.AlertCollection` attribute), 16

trigger_time (`_clockalarm.Alert.Alert` attribute), 15

trigger_time (`_clockalarm.AlertCollection.AlertCollection` attribute), 17