zroya

Release 0.1.2

Apr 19, 2018

Contents

ZROYA is a Python wrapper of win32 API for creating Windows notifications.

CHAPTER 1

Features:

- Creating Windows 10 notification.
- No Sound support.
- Events (click, hidden, shown).

CHAPTER 2

Requires:

Zroya requires pypiwin32 library to work properly. See: *Installation*.

CHAPTER 3

User guide

3.1 Installation

Zroya is being developed with `pypiwin32` as only requirement.

```
pip install pypiwin32
```

If you install zroya from PyPi, all requirements should be installed with just one line:

```
pip install zroya
```

Or you may close github repository and install zroya locally:

```
git clone https://github.com/malja/zroya.git
```

That is all. See *Basic Usage* for basic examples of usage.

3.2 Basic Usage

Install zroya package as described in *Installation*.

After successful installation, import zroya into your project.

```
from zroya import TrayIcon

nc = TrayIcon()
nc.create("My Nofification", "Longer text with\nmultiline support")
```

This example creates simple notification with title, text and info icon. This may be sufficient, however what if you would like to react to user clicking on your notification?

```
from zroya import TrayIcon

exit = False

def click_callback(data):
    global exit
    print("User clicked on notification")
    exit = True

nc = TrayIcon()
nc.create("My Nofification", "Longer text with\nmultiline support", on_click=click_
↪callback)

while nc.update():
    if exit:
        nc.quit()
```

3.2.1 Events

First of all, we defined `click_callback` function. It will be called when user clicks on notification box. The only parameter is dictionary with all data you used when creating this notification.

You have to **register** this callback function to notification when you call *`zroya.TrayIcon.create()`* method as `on_click` parameter.

You may register four different events: `on_click`, `on_show`, `on_hide`. We already discussed `on_click`.

`on_show` is called whenever notification is shown to user. `on_hide` should be called when user hide your notification.

3.2.2 Loop

Finally we move to *`zroya.TrayIcon.update()`* method. You should call it from your application's event loop. It basically pulls all waiting events, runs callbacks and returns True. It is **non-blocking** however it needs to be called periodically for callback to be called as soon as possible.

Without this function, events will never be fired!

3.3 Code examples

This page contains few examples of usage.

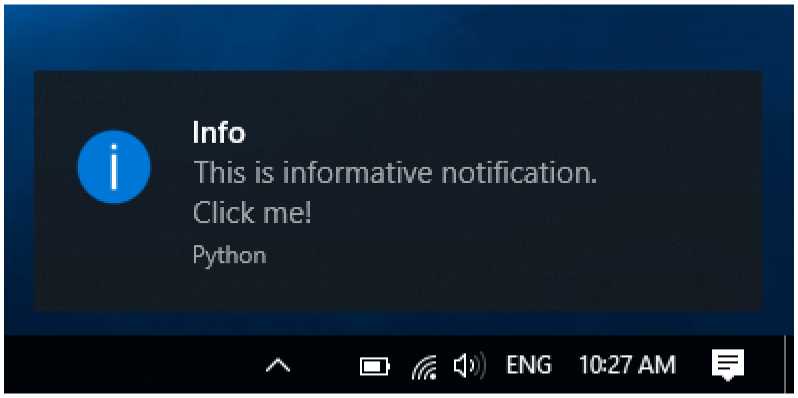
3.3.1 Basic notification

Following example show basic usage of zroya. First, *`zroya.TrayIcon`* instance is created. It is the main center for all notification manipulation. As long as this object exists, you may create notifications, receive events and run callbacks.

Second line creates simple notification with title and text. Since icon and callback parameters are omnited, info icon will be used with no events attached to this notification.

```
from zroya import TrayIcon

nc = TrayIcon() # 1
nc.create("Info", "This is informative notification.\nClick me!") # 2
```


Info
This is informative notification.
Click me!
Python

3.3.2 Notification

Because of Windows limitations, you can create only one notification at the time. If you have one notification running, creating another one will result in TrayIcon.create method returning False.

```
# Zroya is imported, nc contains TrayIcon instance

nc.create("Info", "This is informative notification.\nClick me!")
```

3.3.3 Events

Zroya supports three event types. Each one corresponds with some notification action.

First of them is *on_click*. This event is fired when user clicks on notification. It does not depend whether it is during timeout (notification is shown over the tray bar) or after it (notification is in Windows notification center).

on_show is event fired right after notification is created.

on_hide is the last of supported notification event. It is fired when notification is closed by user.

You may register one or all of above mentioned events for each notification. When calling `zroya.TrayIcon.create()`, pass function name as value to one of *on_** parameters.

```
# zroya is imported, nc contains TrayIcon instance

nc.create("Info", "Text does not matter now",
    on_click = on_click_callback,
    on_hide = on_hide_callback,
    on_show = on_show_callback
)
```

Callbacks

Each callback should be function or method of following form:

```python
def callback(data):
    """
    This is general callback for zroya.
    :param dict data:   All data used when creating corresponding notification.
    """

    # Your code here
    pass
```

Data parameter consist of following keys:

```python
data = {
    "title": "Notification title",
    "message": "Original notification text",
    "icon": "Path to notification icon, or one of TrayIcon.ICON_*",
    "sound": "True/False value. Should sound be played when notification appears?"
}
```

Event loop

This may be the most important part of events example. Because zroya depends on win32 library, it shares bit of its mechanics. On of them is application event loop. This loop takes care of running corresponding actions for each event.

Zroya uses method *zroya.TrayIcon.update()*. It polls all waiting events, runs user defined callbacks and then pass the execution to default Windows event handler.

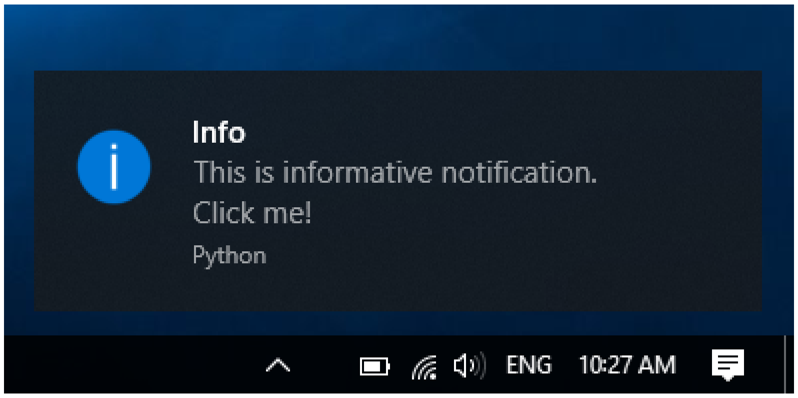
You should call it periodically in your application main loop. Without is, event callbacks won't work.

3.3.4 Icons

For most cases, using system default *info* icon is enought. This icon will be used as default.

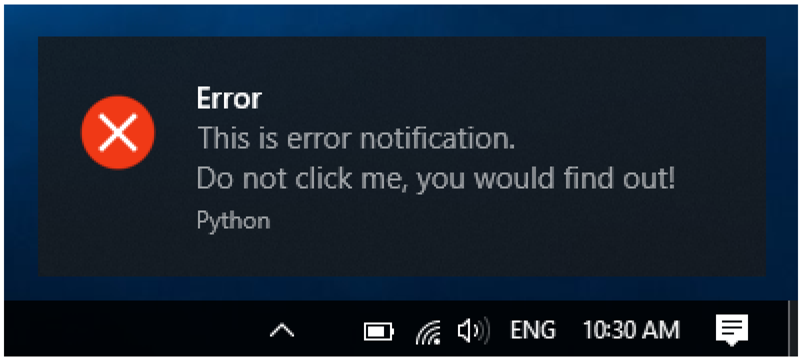
In other cases, you may specify, which icon should be displayed next to notification text. Zroya offers three basic types:

`zroya.TrayIcon.ICON_INFO` is default icon.

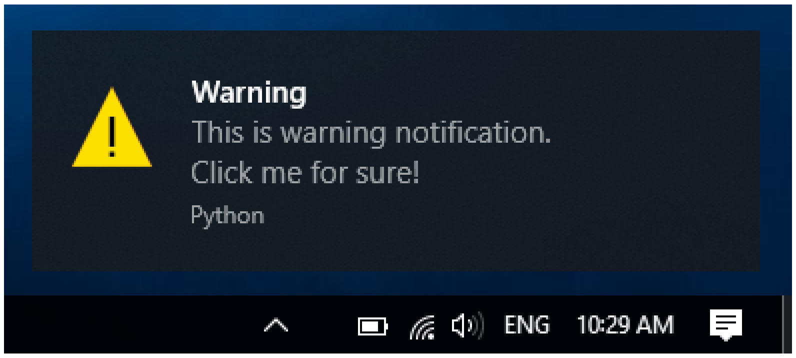


Info
This is informative notification.
Click me!
Python

`zroya.TrayIcon.ICON_ERROR` shows red cross:


Error
This is error notification.
Do not click me, you would find out!
Python

`zroya.TrayIcon.ICON_WARNING` shows yellow warning sign:


Warning
This is warning notification.
Click me for sure!
Python

Following code generates warning notification:

```
# zroya is imported, nc contains TrayIcon instance

nc.create("Test", "Test", icon=TrayIcon.ICON_WARNING) #1
```

In additional to those icons, zroya supports user defined icons. Just pass absolute path to icon file (.ico, .bmp) as *icon* parameter of `zroya.TrayIcon.create()` method.

```
# zroya is imported, nc contains TrayIcon instance

nc.create("Test", "Test", icon="C:\\Path\To\\My\\Image.ico") #1
```

3.3.5 Sound effects

Last feature of zroya is probably muting sound for notification. When creating a new one, passing *False* to `zroya.TrayIcon.create()` parameter *sound* will mute its sound.

```
# zroya is imported, nc contains TrayIcon instance

nc.create("No Sound", "No sound for this notification", sound=False) #1
```

3.4 API reference

This page contains documentation of all methods available in zroya.

zroya.TrayIcon	TrayIcon represents an application icon.
zroya.TrayIcon.create	Shows notification.
zroya.TrayIcon.update	Polls all waiting events.
zroya.TrayIcon.quit	Ends notification event loop.

3.4.1 Methods

class zroya.**TrayIcon**

TrayIcon represents an application icon. You may add one bubble notification.

create(*title*, *message*, *icon=None*, *sound=True*, *on_click=None*, *on_show=None*, *on_hide=None*)

Shows notification. All callback functions take two parameters. First one is integer with notification ID, second is dict with parameters used for creation (title, message, icon).

Parameters

- **title** (*str*) – Notification title.
- **message** (*str*) – Short notification text.
- **icon** (*str/int*) – Path to the icon file or one of TrayIcon.ICON_*. Use None for info icon.
- **sound** (*bool*) – Should sound be played when notification appears?
- **on_click** (*callable*) – On click callback. Called when user clicks on notification.
- **on_show** (*callable*) – On show callback. Called when notification is shown.
- **on_hide** (*callable*) – On hide callback. Called when notification is hidden by user.

Returns Notification ID.

hide()

Remove notification. If notification was not shown yet, return false. :return: True if notification was hidden, False otherwise.

quit()

Ends notification event loop.

Returns Nothing.

update()

Polls all waiting events. This function should be called periodically in event loop. It polls all waiting events from queue and calls corresponding callbacks.

Returns False when notification center was shut down. True otherwise.

CHAPTER 4

Reference

zroya.TrayIcon	TrayIcon represents an application icon.
zroya.TrayIcon.create	Shows notification.
zroya.TrayIcon.quit	Ends notification event loop.
zroya.TrayIcon.update	Polls all waiting events.

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

Index