evenz Documentation

Release 0.1.0

patdaburu

Oct 13, 2019

CONTENTS:

Simplify event-driven python!

CHAPTER

ONE

GETTING STARTED

1.1 Installing the Library

You can use pip to install *evenz*.

```
pip install evenz
```

CHAPTER

TWO

API DOCUMENTATION

Import this module to make event handling a little simpler.

class evenz.events.**Args**
Bases: tuple

Extend this named tuple to provide easy-to-understand arguments for your events.

property sender
the originator of the event

class evenz.events.**Event**(*f: Callable*, *sender: Any = None*)
Bases: object

An event object wraps a function and notifies a set of handlers when the function is called.

__init__(*f: Callable*, *sender: Any = None*)

Parameters

- **f** – the function that triggers the event
- **sender** – the sender of the event

property handlers
Get the handlers for this function.

Returns an iteration of the handlers.

subscribe(*handler: Callable*)
Subscribe a handler function to this event.

Parameters **handler** – the handler

Note: You can also use the += operator.

trigger(**args*, ***kwargs*)
Trigger the event.

unsubscribe(*handler: Callable*)
Unsubscribe a handler function from this event.

Parameters **handler** – the handler

Note: You can also use the -= operator.

evenz.events.**event**(*f: Callable*) → evenz.events.Event
Decorate a function or method to create an *Event*.

Parameters **f** – the function.

Returns the event

See also:

If you are decorating a method within a class, you'll need to use the *observable()* class decorator on the class as well.

evenz.events.**observable**(*cls*)
Use this decorator to mark a class that exposes events.

Parameters **cls** – the class

Returns the class

See also:

If you are using this decorator, you probably also want to use *event()* on some of the methods.

CHAPTER

THREE

DEVELOPMENT

3.1 Getting Started

This section provides instructions for setting up your development environment. If you follow the steps from top to bottom you should be ready to roll by the end.

3.1.1 Get the Source

The source code for the *evenz* project lives at github. You can use *git clone* to get it.

```
git clone https://github.com/patdaburu/bnrml
```

3.1.2 Create the Virtual Environment

You can create a virtual environment and install the project's dependencies using *make*.

```
make venv
make install
source venv/bin/activate
```

3.1.3 Try It Out

One way to test out the environment is to run the tests. You can do this with the *make test* target.

```
make test
```

If the tests run and pass, you're ready to roll.

3.1.4 Getting Answers

Once the environment is set up, you can perform a quick build of this project documentation using the *make answers* target.

```
make answers
```

3.2 Using the *Makefile*

This project includes a Makefile that you can use to perform common tasks such as running tests and building documentation.

3.2.1 Targets

This section contains a brief description of the targets defined in the `Makefile`.

clean

Remove generated packages, documentation, temporary files, *etc*.

lint

Run pylint against the project files.

test

Run the unit tests.

quicktest

Run the unit tests without performing pre-test validations (like *linting*).

docs

Build the documentation for production.

Note: You can also build the documents directly, bypassing validations like *linting* and *testing* using Sphinx Makefile directly.

```
cd docs
make clean && make html
make latexpdf
```

---

answers

Perform a quick build of the documentation and open it in your browser.

package

Build the package for publishing.

publish

Publish the package to your repository.

build

Install the current project locally so that you may run the command-line application.

venv

Create a virtual environment.

install

Install (or update) project dependencies.

licenses

Generate a report of the projects dependencies and respective licenses.

Note: If project dependencies change, please update this documentation.

3.3 Publishing the Package

As you make changes to the project, you'll probably want to publish new version of the package. *(That's the point, right?)*

3.3.1 Publishing

The actual process of publishing the project is just a matter of running the *publish* target.

```
make publish
```

3.3.2 Installing

If you just need to install the library in your project, have a look at the *general tutorial* article.

3.4 Building the Documentation

3.4.1 Sphinx

The documentation in this project is generated by Sphinx from reStructuredTex.

3.4.2 Ubuntu/Debian

This project started on Ubuntu Linux 18.04. That doesn't mean you can't use another distribution, or even another operating system, but you may need to perform some additional setup steps to get your builds working. (If you do get it working under another system, please consider adding an article to let others know how you did it!)

Prerequisites

The project uses the Sphinx LatexBuilder to generate a PDF document. If you're using Ubuntu (or Debian) you'll need to install texlive and latexmk.

```
sudo apt-get install texlive-latex-recommended \
    texlive-latex-extra \
    texlive-fonts-recommended \
    latexmk
```

3.4.3 make

Once everything is in place, you can build the documentation using the *make docs* the target defined in the project's *Makefile*.

```
make docs
```

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

CHAPTER

FIVE

DEPENDENCIES

The `requirements.txt` file contains this project's module dependencies. You can install these dependencies using `pip`.

```
pip install -r requirements.txt
```

5.1 requirements.txt

```
click>=7.0,<8
pip-check-reqs>=2.0.1,<3
pip-licenses>=1.7.1,<2
pylint>=1.8.4,<2
pytest>=3.4.0,<4
pytest-cov>=2.5.1,<3
pytest-pythonpath>=0.7.2,<1
setuptools>=38.4.0
Sphinx>=2.2.0

tox>=3.0.0,<4
twine>=1.11.0,<2
```

5.2 Runtime Dependencies and Licenses

Name	Version	License	URL
Click	7.0	BSD	https://palletsprojects.com/p/click/
bleach	3.1.0	Apache Software License	https://github.com/mozilla/bleach
filelock	3.0.12	Public Domain <http://unlicense.org>	https://github.com/benediktschmitt/py-filelock
frozenordered-dict	1.2.0	MIT	https://github.com/wsmith323/frozenordereddict
importlib-metadata	0.18	Apache Software License	http://importlib-metadata.readthedocs.io/
readme-renderer	24.0	Apache License, Version 2.0	https://github.com/pypa/readme_renderer
toml	0.10.0	MIT	https://github.com/uiri/toml
webencodings	0.5.1	BSD	https://github.com/SimonSapin/python-webencodings
zipp	0.5.1	UNKNOWN	https://github.com/jaraco/zipp

CHAPTER

SIX

INTRODUCTION

6.1 Overview

evenz is a simplified implementation of the observer pattern in python. It uses some friendly syntax conventions from other languages.

6.2 Create a Class with Events

For this simple example we'll create an observable object with just one event.

```
from evenz import observable, event

@observable
class Dog(object):
    """
    This is a dog that can bark.  We can also listen for a 'barked' event.

    """
    def bark(self, count: int):
        """
        Call this method to make the dog bark.

        :param count: How many times will the dog bark?
        """
        for i in range(0, count):
            print('Woof!')
        self.barked(count)

    @event
    def barked(self, count: int):
        """
        This event is raised when the dog barks.

        :param count: how many times did the dog bark?
        """
```

6.3 Handle Events

Now let's respond to the observable object's event by subscribing handler methods.

```
# Create our observable dog.
dog = Dog()

# Create a handler function for the dog's 'barked' event.
def on_bark(sender: Dog, count: int):
    for i in range(0, count):
        print('Hush, puppy!')

# When the dog barks, we'll respond.
dog.barked += on_bark

# Have the dog bark a few times.
dog.bark(5)

# At this point, we're fed up and no longer listening for barks.
dog.barked -= on_bark

# Now the dog's barks should go without a response.
print('OK.  We are no longer listening.')
dog.bark(5)
```

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

e

INDEX

Symbols

A

E

H

O

S

T

U