whistle Documentation

Release 1.0.1

Romain Dorgueil

Jun 11, 2018

Contents

Contents:

CHAPTER 1

Quick start

1.1 Installation

```
$ pip install whistle
```

1.2 Quick start

```
from whistle import Event, EventDispatcher

class HelloWorldEvent(Event):
    pass

dispatcher = EventDispatcher()
```

CHAPTER 2

Reference

2.1 Event

class whistle.**Event**

Base class to represent whistle's events. You can subclass this if you want to embed special data and associated logic with your events, or just let the event dispatcher create instances for you

The event handlers will have *Event* instances passed, so you can bundle any data required by your handlers there.

propagation_stopped = False

Has the event propagation ended?

stop_propagation()

Stop event propagation, meaning that the remaining handlers won't be called after this one.

2.2 EventDispatcher

class whistle.**EventDispatcher**

A logical event dispatcher. All events can only be dispatched in the context of an *EventDispatcher*, and event dispatchers are fully independant and isolated.

add_listener(*event_id*, *listener*, *priority=0*)

dispatch(*event_id*, *event=None*)

do_dispatch(*listeners*, *event*)

get_listeners(*event_id=None*)

has_listeners(*event_id=None*)

listen(*event_id*, *priority=0*)

Decorator that add the decorated functions as one of this instance listeners, for the given event name.

Parameters

- **event_id** –
- **priority** –

Returns

remove_listener(*event_id*, *listener*)

Remove a fiven listener from this event dispatcher. For now, if the listener is not registered for this event, this method has no effect, but we may raise ValueError in the future if the given listener is not found, you should catch it if you're not certain the listener is registered.

TODO raise ValueError if not found.

Parameters

- **event_id** –
- **listener** –

Returns

sort_listeners(*event_id*)

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Index