
Async Dispatch Documentation

Release 0.1

Toby Patterson

July 03, 2016

1	Installation	3
2	Quick Start	5
3	API	7

Python Asyncio-enabled Event Dispatcher Based on Publishers and Subscribers.

Installation

Async Dispatch uses the keywords `async` and `await`, and thus requires [Python 3.5](#) or greater. This version of Python is distributed in the most recent release of [Ubuntu 16.04](#), and can be easily installed on Mac OS X using [brew](#) .

The preferred way to install Async Dispatch for Python 3.5 is using *pip*:

```
pip install async-dispatch
```

Quick Start

Simply create a producer, create a consumer, and start the server.

```
1  #!/usr/bin/env python3.5
2
3  import asyncio
4  import json
5  import time
6
7  from async_dispatch import Dispatcher, SubscriberInterface, PublisherInterface, Event
8
9
10 class BasicPublisher(PublisherInterface):
11
12     count = 0
13
14     async def produce(self):
15         cls = self.__class__
16         cls.count += 1
17         v = cls.count
18
19         return Event('test', time.time(), json.dumps({'value':v}))
20
21
22 class BasicSubscriber(SubscriberInterface):
23
24     listen_to_events = ['test']
25
26     def __init__(self):
27
28         self.total = 0
29
30     async def consume(self, event):
31
32         event_name, timestamp, payload = event
33         payload = json.loads(payload)
34         self.total += int(payload['value'])
35
36     def get_total(self):
37
38         return self.total
39
40 if __name__ == '__main__':
41
```

```
42 publisher = BasicPublisher()
43 subscriber = BasicSubscriber()
44
45 loop = asyncio.get_event_loop()
46
47 server = Dispatcher(publisher, subscriber, loop=loop)
48 loop.run_until_complete(server.start(max_events=5))
49
50 total = subscriber.get_total()
51 print("Total is %d" % total)
52
53 assert total == 15
54
```

API
