
sethpusher Documentation

Release 0.1

Jacek Nosal

July 21, 2015

1	Tutorial	3
1.1	Getting started	3
1.2	Basic usage	3
1.3	Using redis pubsub functionality	3
1.4	Specify host & port	4
1.5	Connecting via websocket or other transport	4
1.6	Connecting, subscribing & unsubscribing	4
1.7	Publishing to channels	4
1.8	Sending message to user	4
1.9	Creating your own app	5
2	Core API	7
3	Websocket API	11
4	Web API	13
5	Indices and tables	15
	Python Module Index	17

Seth-Pusher is tornado backed pub/sub websocket server implementation designed as a set of replaceable components :

```
$ pip install -U sethpusher
```

Seth-Pusher's may also be used with redis pub/sub capabilities

```
$ pip install -U redis && pip install -U hiredis && pip install -U tornado-redis==2.4.18
```

Tutorial A quick **Seth-Pusher** tutorial & overview.

Core API The complete API documentation.

This tutorial introduces **Seth-Pusher** core concepts by example.

1.1 Getting started

Before starting installation, creation of virtualenv is advised. To install server run:

```
$ pip install -U sethpusher
```

1.2 Basic usage

```
$ seth-pusher --debug=true
```

SethPusher relies on sockjs so developing js client make sure to take a look at:

1. <https://github.com/mrjoes/sockjs-tornado>
2. <https://github.com/sockjs/sockjs-client>

When debug flag is set to true we can inspect basic statistics by:

```
$ curl http://localhost:8080/debug/
```

or

```
$ curl curl http://localhost:8080/debug/channels/<channel_name>/
```

1.3 Using redis pubsub functionality

It requires redis-server and additional setup so first:

```
$ sudo apt-get install redis-server  
$ sudo service redis-server start
```

Install additional requirements:

```
$ pip install -U redis  
$ pip install -U hiredis (optionally)  
$ pip install -U tornado-redis==2.4.18
```

Run:

```
$ seth-pusher --debug=true --app='sethpusher.applications.SethPusherOnRedisApplication'
```

1.4 Specify host & port

```
$ seth-pusher --debug=true --host=127.0.0.1 --port=8003
```

1.5 Connecting via websocket or other transport

By default predefined applications expose transport connection via **/connect** endpoint.

1.6 Connecting, subscribing & unsubscribing

Those messages are handled by **/connect** endpoint by default.

1. Connecting:

```
{
  'type': 'connect',
  'subscriber_id': '<unique_user_id>'
}
```

2. Subscribing:

```
{
  'type': 'subscribe',
  'channel': '<channel_name>'
}
```

3. Unsubscribing:

```
{
  'type': 'unsubscribe',
  'channel': '<channel_name>'
}
```

1.7 Publishing to channels

In order to **subscribe/unsubscribe** from channel send proper message to ws handler.

```
$ curl -X POST -H "Content-Type: application/json" -d '{"my": "data"}' http://localhost:8080/api/channel
```

1.8 Sending message to user

By default every user that connects to sethpusher is subscribed to **all** channel which allows us to send him a private message.

```
$ curl -X POST -H "Content-Type: application/json" -d '{"my": "data"}' http://localhost:8080/api/user
```

or

```
$ curl -X POST -H "Content-Type: application/json" -d '{"my": "data", "sender_id": "<author_id>"}' ht
```

1.9 Creating your own app

Do You want your own application class ? Use our **AppRunner** and mixins / components.

1. Create file **my_app.py**

2. Run:

```
$ python my_app.py --logging=debug
```

2. Test it with:

```
$ curl http://localhost:6666/debug/\?batman\=bruce_wayne
```

Code:

```
from tornado.options import parse_command_line

from sethpusher.default_server import AppRunner
from sethpusher.applications import SethPusherInMemoryApplication
from sethpusher.authenticators import ApiKeyAuthenticationPolicy
from sethpusher.persistors import ExternalApiPersistencePolicy

class MyPersistence(ExternalApiPersistencePolicy):
    api_url = 'http://www.my_super_api.com'

    def persist(self, data, **kwargs):
        print data
        print kwargs
        print "KABOOM !"

class MyAuth(ApiKeyAuthenticationPolicy):
    key_name = u'batman'
    key_secret_value = u'bruce_wayne'

class MyVeryOwnApp(SethPusherInMemoryApplication):

    def get_authentication_policy(self, **kwargs):
        return MyAuth()

    def get_persistence_policy(self, **kwargs):
        return MyPersistence()

if __name__ == '__main__':
    parse_command_line()
    runner = AppRunner(port=6666, debug=True)
    runner.start(app=MyVeryOwnApp)
```



```
class sethpusher.applications.DefaultSethApplication (**kwargs)
    Bases: tornado.web.Application

    authentication_policy = None

    get_authentication_policy (**kwargs)

    get_dispatcher ()

    get_persistence_policy (**kwargs)

    persist (data, **kwargs)

    persistence_policy = None

class sethpusher.applications.SethPusherInMemoryApplication (**kwargs)
    Bases: sethpusher.applications.DefaultSethApplication,
           sethpusher.mixins.InMemoryPubSubMixin

class sethpusher.applications.SethPusherOnRedisApplication (**kwargs)
    Bases: sethpusher.applications.DefaultSethApplication,
           sethpusher.mixins.RedisPubSubMixin

class sethpusher.authenticators.ApiKeyAuthenticationPolicy
    Bases: sethpusher.authenticators.BaseAuthenticationPolicy

    authenticate (request, **kwargs)

    key_name = u'api_key'

    key_secret_value = u'secret'

class sethpusher.authenticators.BaseAuthenticationPolicy
    Bases: object

    authenticate (request, **kwargs)

class sethpusher.authenticators.EchoAuthenticationPolicy
    Bases: sethpusher.authenticators.BaseAuthenticationPolicy

    authenticate (request, **kwargs)

class sethpusher.authenticators.HeaderAuthenticationPolicy
    Bases: sethpusher.authenticators.BaseAuthenticationPolicy

    authenticate (request, **kwargs)

    secret_header = u'Seth-Authentication-Header'
```

```
    secret_header_value = u'Seth-Value'

class sethpusher.default_server.AppRunner(*args, **kwargs)
    Bases: object

    host = '127.0.0.1'

    port = 8080

    start(app)

sethpusher.default_server.main()

sethpusher.default_server.shutdown(server_instance)

class sethpusher.dispatchers.DefaultMessageDispatcher
    Bases: object

    CONNECT = 'connect'

    DISCONNECT = 'disconnect'

    SUBSCRIBE = 'subscribe'

    UNSUBSCRIBE = 'unsubscribe'

    channels

    dispatch(msg)

    handle_connect(msg)

    handle_subscribe(msg)

    handle_unsubscribe(msg)

    is_connected

    register_handler(handler)

    validate_message(msg)

class sethpusher.mixins.InMemoryPubSubMixin
    Bases: sethpusher.mixins.PubSubMixin

    get_publisher(**kwargs)

    get_subscriber(**kwargs)

    get_subscriber_count(channel=None)

    get_subscribers(channel=None)

    get_user(user_id)

    pub_sub

class sethpusher.mixins.PubSubMixin
    Bases: object

    add_subscriber(channel, subscriber)

    broadcast(message, channel=None)

    get_publisher()

    get_subscriber()

    get_subscriber_count(channel=None)
```

```
    get_subscribers (channel=None)
    get_user (user)
    remove_subscriber (channel, subscriber)
    send_to_channel (msg, channel)
    send_to_user (msg, user)
class sethpusher.mixins.RedisPubSubMixin
    Bases: sethpusher.mixins.PubSubMixin
    add_subscriber (channel, subscriber)
    get_publisher (**kwargs)
    get_subscriber (**kwargs)
    get_subscriber_count (channel=None)
    get_subscribers (channel=None)
    get_user (user_id)
    is_subscribed_to_channel (subscriber, channel)
    redis_host = 'localhost'
    redis_password = None
    redis_port = 6379
    redis_selected_db = None
class sethpusher.persistors.DefaultPersistencePolicy
    Bases: object
    Provides interface to persistence policies.
    persist (data, **kwargs)
class sethpusher.persistors.ExternalApiPersistencePolicy
    Bases: sethpusher.persistors.DefaultPersistencePolicy
    Posts message to selected API endpoint using AsyncHTTPClient
    api_url = ''
    get_api_url (**kwargs)
    persist (data, **kwargs)
class sethpusher.persistors.RedisPersistencePolicy
    Bases: sethpusher.persistors.DefaultPersistencePolicy
    Stores messages in redis
    persist (data, **kwargs)
class sethpusher.utils.BasicPubSubManager
    Bases: object
    publish (channel, message)
    subscribe (channels, subscriber)
    unsubscribe (channel, subscriber)
sethpusher.utils.is_request_body_json (f)
```

WebSocket API

```
class sethpusher.ws.urls.SethSockJSRouter (connection,      prefix='',      user_settings={},
                                           io_loop=None, **kwargs)
    Bases: sockjs.tornado.router.SockJSRouter

sethpusher.ws.urls.get_ws_handlers (application_instance)

class sethpusher.ws.views.SethSocketHandler (session)
    Bases: sockjs.tornado.conn.SockJSConnection

    application
    check_origin (origin)
    close (code=3000, message='')
    handle_connect (source_msg, subscriber_id)
    handle_subscribe (source_msg)
    handle_unsubscribe (source_msg)
    on_close ()
    on_message (raw_msg)
    on_open (request)
```



```
sethpusher.api.urls.get_api_handlers(application_instance)

class sethpusher.api.views.ChannelDebugHandler(application, request, **kwargs)
    Bases: sethpusher.api.views.SethHandler

    get(*args, **kwargs)

class sethpusher.api.views.ChannelHandler(application, request, **kwargs)
    Bases: sethpusher.api.views.SethHandler

    post(*args, **kwargs)

class sethpusher.api.views.OverallDebugHandler(application, request, **kwargs)
    Bases: sethpusher.api.views.SethHandler

    get(*args, **kwargs)

class sethpusher.api.views.SethHandler(application, request, **kwargs)
    Bases: tornado.web.RequestHandler

    error(message, code=500)

    json_response(data, code=200)

    prepare()

class sethpusher.api.views.UserHandler(application, request, **kwargs)
    Bases: sethpusher.api.views.SethHandler

    post(*args, **kwargs)
```

Indices and tables

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

S

- `sethpusher.api.urls`, [13](#)
- `sethpusher.api.views`, [13](#)
- `sethpusher.applications`, [7](#)
- `sethpusher.authenticators`, [7](#)
- `sethpusher.default_server`, [8](#)
- `sethpusher.dispatchers`, [8](#)
- `sethpusher.mixins`, [8](#)
- `sethpusher.persistors`, [9](#)
- `sethpusher.utils`, [9](#)
- `sethpusher.ws.urls`, [11](#)
- `sethpusher.ws.views`, [11](#)

A

`add_subscriber()` (sethpusher.mixins.PubSubMixin method), 8
`add_subscriber()` (sethpusher.mixins.RedisPubSubMixin method), 9
`api_url` (sethpusher.persistors.ExternalApiPersistencePolicy attribute), 9
`ApiKeyAuthenticationPolicy` (class in sethpusher.authenticators), 7
`application` (sethpusher.ws.views.SethSocketHandler attribute), 11
`AppRunner` (class in sethpusher.default_server), 8
`authenticate()` (sethpusher.authenticators.ApiKeyAuthenticationPolicy method), 7
`authenticate()` (sethpusher.authenticators.BaseAuthenticationPolicy method), 7
`authenticate()` (sethpusher.authenticators.EchoAuthenticationPolicy method), 7
`authenticate()` (sethpusher.authenticators.HeaderAuthenticationPolicy method), 7
`authentication_policy` (sethpusher.applications.DefaultSethApplication attribute), 7

B

`BaseAuthenticationPolicy` (class in sethpusher.authenticators), 7
`BasicPubSubManager` (class in sethpusher.utils), 9
`broadcast()` (sethpusher.mixins.PubSubMixin method), 8

C

`ChannelDebugHandler` (class in sethpusher.api.views), 13
`ChannelHandler` (class in sethpusher.api.views), 13
`channels` (sethpusher.dispatchers.DefaultMessageDispatcher attribute), 8
`check_origin()` (sethpusher.ws.views.SethSocketHandler method), 11
`close()` (sethpusher.ws.views.SethSocketHandler method), 11

`CONNECT` (sethpusher.dispatchers.DefaultMessageDispatcher attribute), 8

D

`DefaultMessageDispatcher` (class in sethpusher.dispatchers), 8
`DefaultPersistencePolicy` (class in sethpusher.persistors), 9
`DefaultSethApplication` (class in sethpusher.applications), 7
`DISCONNECT` (sethpusher.dispatchers.DefaultMessageDispatcher attribute), 8
`dispatch()` (sethpusher.dispatchers.DefaultMessageDispatcher method), 8

E

`EchoAuthenticationPolicy` (class in sethpusher.authenticators), 7
`ExternalApiPersistencePolicy` (class in sethpusher.persistors), 9
`get()` (sethpusher.api.views.SethHandler method), 13

G

`get()` (sethpusher.api.views.ChannelDebugHandler method), 13
`get()` (sethpusher.api.views.OverallDebugHandler method), 13
`get_api_handlers()` (in module sethpusher.api.urls), 13
`get_api_url()` (sethpusher.persistors.ExternalApiPersistencePolicy method), 9
`get_authentication_policy()` (sethpusher.applications.DefaultSethApplication method), 7
`get_dispatcher()` (sethpusher.applications.DefaultSethApplication method), 7
`get_persistence_policy()` (sethpusher.applications.DefaultSethApplication method), 7
`get_publisher()` (sethpusher.mixins.InMemoryPubSubMixin method), 8

`get_publisher()` (sethpusher.mixins.PubSubMixin method), 8

`get_publisher()` (sethpusher.mixins.RedisPubSubMixin method), 9

`get_subscriber()` (sethpusher.mixins.InMemoryPubSubMixin method), 8

`get_subscriber()` (sethpusher.mixins.PubSubMixin method), 8

`get_subscriber()` (sethpusher.mixins.RedisPubSubMixin method), 9

`get_subscriber_count()` (sethpusher.mixins.InMemoryPubSubMixin method), 8

`get_subscriber_count()` (sethpusher.mixins.PubSubMixin method), 8

`get_subscriber_count()` (sethpusher.mixins.RedisPubSubMixin method), 9

`get_subscribers()` (sethpusher.mixins.InMemoryPubSubMixin method), 8

`get_subscribers()` (sethpusher.mixins.PubSubMixin method), 8

`get_subscribers()` (sethpusher.mixins.RedisPubSubMixin method), 9

`get_user()` (sethpusher.mixins.InMemoryPubSubMixin method), 8

`get_user()` (sethpusher.mixins.PubSubMixin method), 9

`get_user()` (sethpusher.mixins.RedisPubSubMixin method), 9

`get_ws_handlers()` (in module sethpusher.ws.urls), 11

H

`handle_connect()` (sethpusher.dispatchers.DefaultMessageDispatcher method), 8

`handle_connect()` (sethpusher.ws.views.SethSocketHandler method), 11

`handle_subscribe()` (sethpusher.dispatchers.DefaultMessageDispatcher method), 8

`handle_subscribe()` (sethpusher.ws.views.SethSocketHandler method), 11

`handle_unsubscribe()` (sethpusher.dispatchers.DefaultMessageDispatcher method), 8

`handle_unsubscribe()` (sethpusher.ws.views.SethSocketHandler method), 11

`HeaderAuthenticationPolicy` (class in sethpusher.authenticators), 7

`host` (sethpusher.default_server.AppRunner attribute), 8

I

`InMemoryPubSubMixin` (class in sethpusher.mixins), 8

`is_connected` (sethpusher.dispatchers.DefaultMessageDispatcher attribute), 8

`is_request_body_json()` (in module sethpusher.utils), 9

`is_subscribed_to_channel()` (sethpusher.mixins.RedisPubSubMixin method), 9

J

`json_response()` (sethpusher.api.views.SethHandler method), 13

K

`key_name` (sethpusher.authenticators.ApiKeyAuthenticationPolicy attribute), 7

`key_secret_value` (sethpusher.authenticators.ApiKeyAuthenticationPolicy attribute), 7

M

`main()` (in module sethpusher.default_server), 8

O

`on_close()` (sethpusher.ws.views.SethSocketHandler method), 11

`on_message()` (sethpusher.ws.views.SethSocketHandler method), 11

`on_open()` (sethpusher.ws.views.SethSocketHandler method), 11

`OverallDebugHandler` (class in sethpusher.api.views), 13

P

`persist()` (sethpusher.applications.DefaultSethApplication method), 7

`persist()` (sethpusher.persistors.DefaultPersistencePolicy method), 9

`persist()` (sethpusher.persistors.ExternalApiPersistencePolicy method), 9

`persist()` (sethpusher.persistors.RedisPersistencePolicy method), 9

`persistence_policy` (sethpusher.applications.DefaultSethApplication attribute), 7

`port` (sethpusher.default_server.AppRunner attribute), 8

`post()` (sethpusher.api.views.ChannelHandler method), 13

`post()` (sethpusher.api.views.UserHandler method), 13

`prepare()` (sethpusher.api.views.SethHandler method), 13

`pub_sub` (sethpusher.mixins.InMemoryPubSubMixin attribute), 8

`publish()` (sethpusher.utils.BasicPubSubManager method), 9

`PubSubMixin` (class in sethpusher.mixins), 8

R

redis_host (sethpusher.mixins.RedisPubSubMixin attribute), 9

redis_password (sethpusher.mixins.RedisPubSubMixin attribute), 9

redis_port (sethpusher.mixins.RedisPubSubMixin attribute), 9

redis_selected_db (sethpusher.mixins.RedisPubSubMixin attribute), 9

RedisPersistencePolicy (class in sethpusher.persistors), 9

RedisPubSubMixin (class in sethpusher.mixins), 9

register_handler() (sethpusher.dispatchers.DefaultMessageDispatcher method), 8

remove_subscriber() (sethpusher.mixins.PubSubMixin method), 9

S

secret_header (sethpusher.authenticators.HeaderAuthenticationPolicy attribute), 7

secret_header_value (sethpusher.authenticators.HeaderAuthenticationPolicy attribute), 7

send_to_channel() (sethpusher.mixins.PubSubMixin method), 9

send_to_user() (sethpusher.mixins.PubSubMixin method), 9

SethHandler (class in sethpusher.api.views), 13

sethpusher.api.urls (module), 13

sethpusher.api.views (module), 13

sethpusher.applications (module), 7

sethpusher.authenticators (module), 7

sethpusher.default_server (module), 8

sethpusher.dispatchers (module), 8

sethpusher.mixins (module), 8

sethpusher.persistors (module), 9

sethpusher.utils (module), 9

sethpusher.ws.urls (module), 11

sethpusher.ws.views (module), 11

SethPusherInMemoryApplication (class in sethpusher.applications), 7

SethPusherOnRedisApplication (class in sethpusher.applications), 7

SethSocketHandler (class in sethpusher.ws.views), 11

SethSockJSRouter (class in sethpusher.ws.urls), 11

shutdown() (in module sethpusher.default_server), 8

start() (sethpusher.default_server.AppRunner method), 8

SUBSCRIBE (sethpusher.dispatchers.DefaultMessageDispatcher attribute), 8

subscribe() (sethpusher.utils.BasicPubSubManager method), 9

U

UNSUBSCRIBE (sethpusher.dispatchers.DefaultMessageDispatcher attribute), 8

unsubscribe() (sethpusher.utils.BasicPubSubManager method), 9

UserHandler (class in sethpusher.api.views), 13

V

validate_message() (sethpusher.dispatchers.DefaultMessageDispatcher method), 8