## ASYNC TASKS WITH DJANGO CHANNELS

PyCon Canada 2016 Albert O'Connor @amjoconn albertoconnor.ca

#### What is Channels?

The most exciting thing to happen to **Django** since **Django** 



#### I won't be talking about WebSockets

The Channels Docs have a good intro tutorial on WebSockets <a href="https://channels.readthedocs.io/en/stable/getting-started.html">https://channels.readthedocs.io/en/stable/getting-started.html</a>

# I expect you are familiar with Django

#### So, Channels is...

A library which allows **Django** to handle more than just plain HTTP Requests

#### Created by Andrew Godwin



#### Supported by a Mozilla MOSS Grant

### mozilla

#### An "official" Django project

#### Problem: How to support WebSockets in Django

#### Solution: Another level of abstraction



## Receives a HTTP Request, calls a view which returns an HTTP Response

#### Channels

# Receives a message, calls a consumer which can send other messages

#### Django Channels

Receive a HTTP Request message, calls a consumer which calls a view

The view returns a HTTP Response, the consumer send the message to the http.response channel

#### Abstraction!







There are WebSocket specific channels including websocket.connect and websocket.receive But... You can create your own channel! Channels are named queues on a message bus We are going to create a channel to deal with an async background task



### Warning At-most-once delivery

Ordering is also worth thinking about:

https://channels.readthedocs.io/en/stable/getting-started.html#enforcing-ordering

#### **Quick Examples**

Circuits	×							
$\rightarrow$ C (i) circuits.	iqc.uwaterloo.ca/alexparent/seq/				☆ 約	I	<mark>⊙</mark> ≌	
Circuits				F	Functions C	ircuits	Sign Ir	n
alexpa	irent / seq							
Verified 🕕 🗩	Comments 0							
Implements	function seq							
Stats								
Gates	Toffoli Gates	T Gates	Bits	Uploaded				
6508	5149	36043	106	July 29, 2016, 3:52 p.m.				
							)	
					View SVG	a Dow	nload	
.v in0 in	1 in2 in3 in4 in5 in6 in7 in8	in9 in10 in11 in12	in13 in14 i	in15 in16 in17 in18 in	19 in20 in21	in22 i	n23 i	
.i in0 in	1 in2 in3 in4 in5 in6 in7 in8	in9 in10 in11 in12	in13 in14 i	in15 in16 in17 in18 in	19 in20 in21	in22 i	n23 i	
.ol in0 i	n1 in2 in3 in4 in5 in6 in7 in6 n1 in2 in3 in4 in5 in6 in7 in	18 in9 in10 in11 in12	2 in13 in14 i	in15 in16 in17 in18 in in15 in16 in17 in18 i	in19 in20 in21	1 in22 i	in23 ]	
BEGIN								
t2 out0 o	ut1 ut2							
t2 out0 o	ut3							
t2 out0 o	ut4							
t2 out0 o	ut5							
t3 anc0 i	n38 anc1							
t3 anc1 i	n7' anc2							
t3 anc2 i	n32' anc3							



How do you add Channels to your project?

#### Pip install channels and add it to INSTALLED\_APPS

#### Channels "replaces" WSGI with ASGI

Installing Channels includes a ASGI server called Daphne implemented with Twisted **Django** gains a runworker management command For development runserver works by running workers and Daphne in one process using threads For production an ASGI broker is needed between Daphne and the workers

asgi\_redis + redis server is a great option Daphne handles HTTP, WebSockets, and more, enqueuing messages into the right channel Views and consumers can also enqueue messages into channels This means your view and consumer code is written synchronously

### PRIVATE AWE SOME ST STREET AWE SOME 100

https://www.flickr.com/photos/moonlightbulb/3338852116

#### Demo

#### https://github.com/albertoconnor/asyncdemo

Tag: step1 Basic Django app with a view which says hello and simulates sending a notification

```
# In hello/views.py
def delay():
    while True:
        for i in [5, 5, 5, 30]: # Simulate unreliability
            yield i
delay_generator = delay()
def send_notification(message):
    time.sleep(next(delay_generator))
    print(message) # Simulate sending to slack etc.
def hello_view(request, template="hello.html"):
    name = request.GET.get('name', 'World')
    message = 'Hello, {}!'.format(name)
    send_notification(message)
    return render(
        request,
        template,
        dict(message=message),
    )
```

Tag: step2 Install Channels and update settings # In requirements.txt
django==1.10.2
channels==0.17.3

Successfully installed asgiref-0.14.0 autobahn-0.16.0 channels-0.17.3 daphne-0.15.0 six-1.10.0 twisted-16.4.1 txaio-2.5.1 zope.interface-4.3.2

```
# In asyncdemo/settings.py
INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'channels',
)
CHANNEL_LAYERS = {
    "default": {
        "BACKEND": "asgiref.inmemory.ChannelLayer",
        "ROUTING": "asyncdemo.routing.channel_routing",
    },
}
# In routing.py
from channels.routing import route
channel_routing = []
```

Performing system checks...

System check identified no issues (0 silenced).
November 10, 2016 - 11:43:12
Django version 1.10.2, using settings 'asyncdemo.settings'
Starting Channels development server at http://127.0.0.1:8001/
Channel layer default (asgiref.inmemory.ChannelLayer)
Quit the server with CONTROL-C.
2016-11-10 11:43:12,340 - INFO - worker - Listening on channels http.request, websocket.connect, websocket.receive
2016-11-10 11:43:12,340 - INFO - worker - Listening on channels http.request, websocket.connect, websocket.receive
2016-11-10 11:43:12,341 - INFO - worker - Listening on channels http.request, websocket.connect, websocket.receive
2016-11-10 11:43:12,341 - INFO - worker - Listening on channels http.request, websocket.connect, websocket.receive
2016-11-10 11:43:12,341 - INFO - worker - Listening on channels http.request, websocket.connect, websocket.receive
2016-11-10 11:43:12,341 - INFO - worker - Listening on channels http.request, websocket.connect, websocket.receive
2016-11-10 11:43:12,341 - INFO - worker - Listening on channels http.request, websocket.connect, websocket.receive
2016-11-10 11:43:12,341 - INFO - worker - Listening on channels http.request, websocket.connect, websocket.receive
2016-11-10 11:43:12,341 - INFO - worker - Listening on channels http.request, websocket.connect, websocket.receive
2016-11-10 11:43:12,341 - INFO - worker - Listening on channels http.request, websocket.connect, websocket.receive
2016-11-10 11:43:12,347 - INFO - server - Using busy-loop synchronous mode on channel layer

#### Tag: step3 Create channel and use it

```
# In hello/views.py
from django.shortcuts import render
from channels import Channel
def hello_view(request, template="hello.html"):
    name = request.GET.get('name', 'World')
    message = 'Hello, {}!'.format(name)
    Channel('notify').send(
        dict(
            message=message,
    return render(
        request,
        template,
        dict(message=message),
    )
```

```
# In asyncdemo/routing.py
from channels.routing import route
from hello import consumers
channel_routing = [
    route('notify', consumers.notify),
]
# In hello/consumers.py
import time
def delay():
    while True:
        for i in [5, 5, 5, 30]:
            yield i
delay_generator = delay()
def notify(message):
    time.sleep(next(delay_generator))
    print(message['message'])
```

Now the website is responsive until it gets backed up Tag: bonusround Use redis, Daphne and run separate processes # In requirements.txt
django==1.10.2
channels==0.17.3
asgi\_redis==0.14.1 # Also need redis running

```
# In asyncdemo/settings.py
CHANNEL_LAYERS = {
    "default": {
        "BACKEND": "asgi_redis.RedisChannelLayer",
        "CONFIG": {
            "hosts": ['redis://localhost:6379'],
        "ROUTING": "asyncdemo.routing.channel_routing",
    },
}
```

This should be enough to get runserver working again

To use Daphne we need to create asgi.py similar to wsgi.py

```
# In asyncdemo/asgi.py
import os
import channels.asgi
```

```
os.environ.setdefault(
    "DJANGO_SETTINGS_MODULE",
    "asyncdemo.settings"
```

)

channel\_layer = channels.asgi.get\_channel\_layer()

#### daphne asyncdemo.asgi:channel\_layer --port 8000

#### Now we need some workers

#### python manage.py runworker

#### python manage.py runworker --exclude-channels=notify

#### That's it!

# Now go forth and write readable async code

### This was... ASYNC TASKS WITH DJANGO CHANNELS

Thanks! Questions?

PyCon Canada 2016 Albert O'Connor @amjoconn albertoconnor.ca