



# Building PyCon Guide

WebSockets and Django Channels

	Oregon Ballroom 201–202	Oregon Ballroom 203–204	Portland Ballroom 251 & 258	Portland Ballroom 252–253	Portland Ballroom 254–255
08:00AM	Breakfast				
09:00AM	Welcome to PyCon				
09:30AM	Keynote — Lorena Barba				
10:10AM	Break				
10:50AM	Machete-mode debugging: Hacking your way out of a tight spot Ned Batchelder Intermediate S	File descriptors, Unix sockets and other POSIX wizardry Christian Heimes Intermediate S	Thinking In Coroutines Lukasz Langa Intermediate S	Seriously Strong Security on a Shoestring (CW) Kelsey Gilmore-Innis Intermediate S	Python as a configuration language Guillermo Pérez Novice S
11:30AM	Refactoring Python: Why and how to restructure your code Brett Statkin Novice S	Pyjion: who doesn't want faster for free? Brett Cannon, Dino Viehland Experienced S	A tale of concurrency through creativity in Python: a deep dive into how event works. Kavya Joshi Intermediate S	When the going gets tough, get TUF going Ying Li, David Lawrence Novice S	Git: A Peek Under the Hood Clara Bennett Intermediate S
12:10PM	Code Unto Others Nathaniel Manista, Augie Fackler Intermediate S	Here be Dragons - Writing Safe C Extensions Paul Ross Experienced S	Building a Quantitative Trading Strategy To Beat the S&P500 Karen Rubin Intermediate S	Designing secure systems with Object-Capabilities, Python, and Cap'n Proto Drew Fisher Intermediate S	Building An Interpreter In RPython Juozas Kazukenas Experienced S
12:40PM	Lunch	Lunch	Lunch	Lunch	Lunch
12:55PM					
01:40PM					
01:55PM	Awaken your home: Python and the Internet of Things Paulus Schoutsen Intermediate S	Usable Ops: How to make web infrastructure management easier. Kate Heddleston, Joyce Tang Novice S	Education Summit wrap-up: teaching Python — how hard can it be? Nicholas Tollervey Novice S	Networking without an OS Josh Triplett Novice S	SQLite: Gotchas and Gimmes Dave Sawyer Intermediate S
02:35PM	Keeping cool, using a Raspberry PI to create a networked temperature sensor Dwight Hubbard Novice S	The cobbler's children have no shoes, or building better tools for ourselves Alex Gaynor Intermediate S	Finding closure with closures Thomas Ballinger Intermediate S	Wrestling Python into LLVM Intermediate Representation Anna Herlihy Intermediate S	Postgres present and future Craig Kerstiens Intermediate S
03:15PM	Laser Cutters, 3D Printers, and Python Giles Hall Novice S	Pythons in A Container - Lessons Learned Dockerizing Python Micro-Services Dorian Pula Intermediate S	The Life Cycle of a Python Class Mike Graham Experienced S	Memory Management in Python - The Basics Nina Zakharenko Novice S	Django 1.8/1.9 and PostgreSQL: An Ever-Closer Union Christophe Pettus Intermediate S
03:45PM	Break	Break	Break	Break	Break
04:00PM					
04:15PM					
04:30PM	A Beginner's Guide to Deep Learning Irene Chen Intermediate S	Reinventing Django for the Real-Time Web Andrew Godwin Intermediate S	To mock, or not to mock, that is the question Ana Balica Intermediate S	Playing with Python Bytecode Scott Sanderson, Joe Jevnik Experienced S	Build a Better Hat Rack: All Contributions Welcome Katie McLaughlin Novice S
05:10PM	Visual Diagnostics for More Informed Machine Learning: Within and Beyond Scikit-Learn Rebecca Blinbro Intermediate	Building Protocol Libraries The Right Way Cory Benfield Intermediate S	Better Integration Testing with Cucumber Jay Gao Novice S	Let's read code: the requests library Susan Tan Intermediate S	Bake the Cookies, Wear the Dress: Connecting with Confidential Authenticity Adrienne Lowe Novice S

```
import lxml
```

```
# ugly scraper code
```



Select presentation to change

ADD PRESENTATION +

Q

Search

Action: 

-----

 Go 0 of 95 selected

<input type="checkbox"/>	YEAR	TITLE	2	START TIME	1	DURATION
<input type="checkbox"/>	2016	File descriptors, Unix sockets and other POSIX wizardry		May 30, 2016, 10:50 a.m.		30
<input type="checkbox"/>	2016	Machete-mode debugging: Hacking your way out of a tight spot		May 30, 2016, 10:50 a.m.		30
<input type="checkbox"/>	2016	Python as a configuration language		May 30, 2016, 10:50 a.m.		30
<input type="checkbox"/>	2016	Seriously Strong Security on a Shoestring (CW)		May 30, 2016, 10:50 a.m.		30
<input type="checkbox"/>	2016	Thinking In Coroutines		May 30, 2016, 10:50 a.m.		30
<input type="checkbox"/>	2016	A tale of concurrency through creativity in Python: a deep dive into how gevent works.		May 30, 2016, 11:30 a.m.		30
<input type="checkbox"/>	2016	Git: A Peek Under the Hood		May 30, 2016, 11:30 a.m.		30
<input type="checkbox"/>	2016	Pyjion: who doesn't want faster for free?		May 30, 2016, 11:30 a.m.		30
<input type="checkbox"/>	2016	Refactoring Python: Why and how to restructure your code		May 30, 2016, 11:30 a.m.		30
<input type="checkbox"/>	2016	When the going gets tough, get TUF going		May 30, 2016, 11:30 a.m.		30
<input type="checkbox"/>	2016	Designing secure systems with Object-Capabilities, Python, and Cap'n Proto		May 30, 2016, 12:10 p.m.		30
<input type="checkbox"/>	2016	Here be Dragons - Writing Safe C Extensions		May 30, 2016, 12:10 p.m.		30
<input type="checkbox"/>	2016	Building An Interpreter In RPython		May 30, 2016, 12:10 p.m.		45
<input type="checkbox"/>	2016	Building a Quantitative Trading Strategy To Beat the S&P500		May 30, 2016, 12:10 p.m.		45
<input type="checkbox"/>	2016	Code Unto Others		May 30, 2016, 12:10 p.m.		45
<input type="checkbox"/>	2016	Networking without an OS		May 30, 2016, 1:40 p.m.		45
<input type="checkbox"/>	2016	Usable Ops: How to make web infrastructure management easier.		May 30, 2016, 1:40 p.m.		45
<input type="checkbox"/>	2016	Awaken your home: Python and the Internet of Things		May 30, 2016, 1:55 p.m.		30
<input type="checkbox"/>	2016	Education Summit wrap-up: teaching Python — how hard can it be?		May 30, 2016, 1:55 p.m.		30
<input type="checkbox"/>	2016	SQLite: Gotchas and Gimmes		May 30, 2016, 1:55 p.m.		30
<input type="checkbox"/>	2016	Finding closure with closures		May 30, 2016, 2:35 p.m.		30

FILTER

By Duration

All

30

45

By Category

All

Best Practices & Patterns

Community

Databases

Education

Embedded Systems

Gaming

Other

Python Core (language, stdlib, etc.)

Python Internals

Python Libraries

Science

Security

Systems Administration

Testing

Web Frameworks

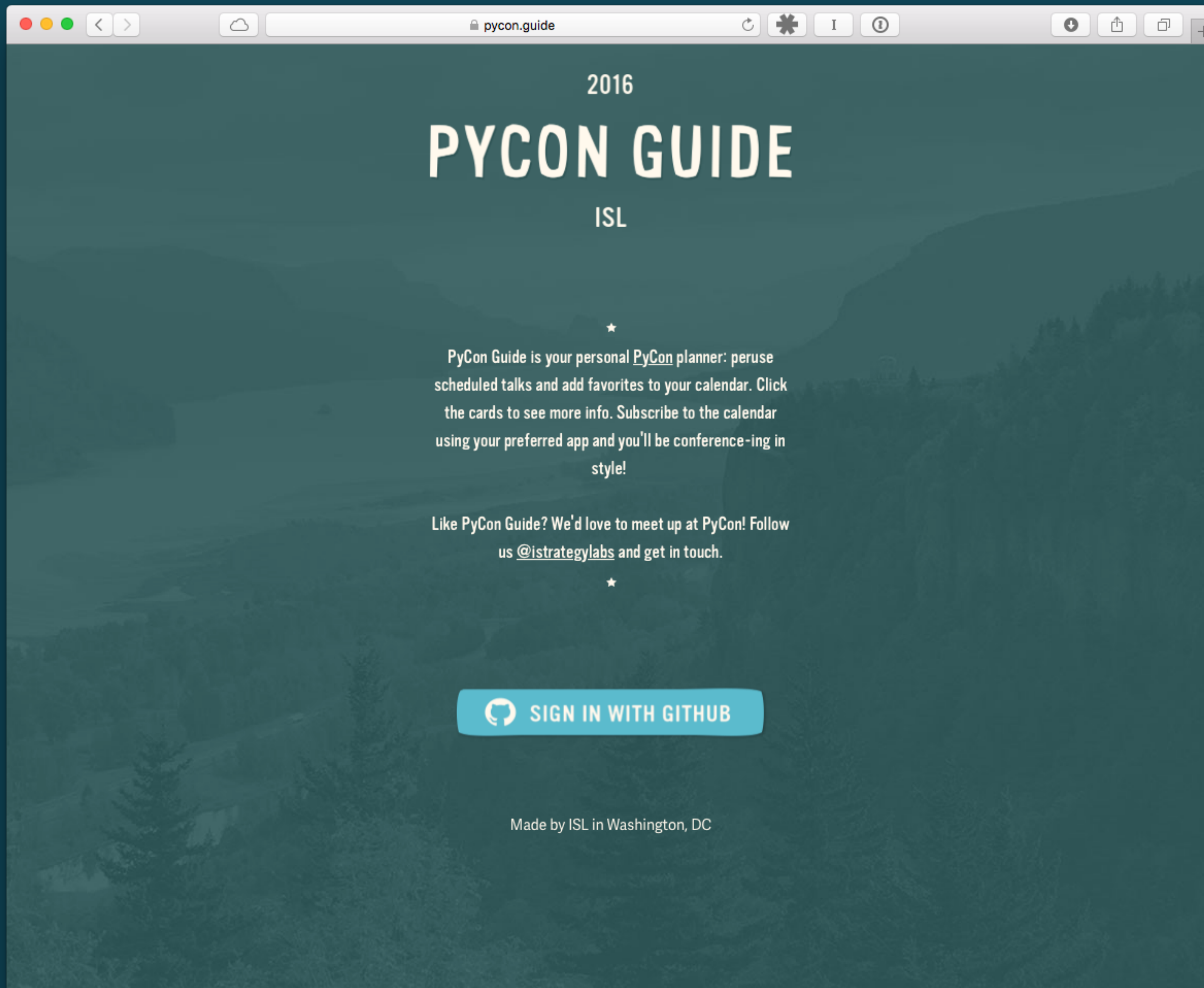
By Audience

All

Experienced

Intermediate

Novice



2016

# PYCON GUIDE

ISL



PyCon Guide is your personal [PyCon](#) planner: peruse scheduled talks and add favorites to your calendar. Click the cards to see more info. Subscribe to the calendar using your preferred app and you'll be conference-ing in style!

Like PyCon Guide? We'd love to meet up at PyCon! Follow us [@istrategylabs](#) and get in touch.



SIGN IN WITH GITHUB

Made by ISL in Washington, DC



ADMIN

PYCON GUIDE

SUBSCRIBE TO CALENDAR

LOG OUT

MAY 30, 2016

10:50 a.m.

File descriptors, Unix sockets and other POSIX wizardry

🕒 30 minutes

📖 Python Internals

📊 Intermediate

📍 Oregon 203–204

Add to Calendar

Machete-mode debugging: Hacking your way out of a tight spot

🕒 30 minutes

📖 Best Practices & Patterns

📊 Intermediate

📍 Oregon 201–202

Add to Calendar

Python as a configuration language

🕒 30 minutes

📖 Systems Administration

📊 Novice

📍 Portland 254–255

Add to Calendar

Seriously Strong Security on a Shoestring (CW)

🕒 30 minutes

📖 Security

📊 Intermediate

📍 Portland 252–253

Add to Calendar

May 2016

	Sun 29	Mon 30	Tue 31	Wed 1	
all-day					
noon					
1 PM					
2 PM		Thinking In... File descrip...	Click: A Ple...	The New Mo...	
3 PM		Refactoring... A tale of co...	Python Typ...	HTTP/2 and...	
4 PM		3:10 PM Code Unto Others Oregon 20...			
5 PM		Awaken you...	4:40 PM Usable Ops:...	Comp... Write... More...	
6 PM		Postgr... Findin... Wrestl...	Remote Calls != Local Calls:...	Python's Infamous GIL Portla...	
7 PM		6:15 PM The Life Cy...	6:15 PM Django 1.8/1...	6:15 PM From Devel...	6:15 PM The Report...
8 PM		7:15 PM Reinv...	7:15 PM Playin... To mo...	Why can't we be friends: do c...	
9 PM		Let's read code: the requests...	Web D... A tale... Dispel...		

Reinventing Django for the Real-Time Web

Oregon 203–204

May 30, 2016 7:15 PM to 8 PM

while retaining Django's nice view model and not making developers write low-level asynchronous code. As a bonus, it also enables offloading of tasks outside the request-response cycle; for example, thumbnailing images after they're uploaded and a successful response is returned.

The talk will cover:

- The landscape today and the basic WSGI/view abstraction
- The challenges WebSockets and similar protocols bring
- How Channels maps these concepts into a familiar view-like layer
- Examples of using Channels to do things like chat rooms, live blogs, and delayed thumbnailing
- Deploying Channels at scale and how it

ADMIN

PYCON GUIDE

SUBSCRIBE TO CALENDAR

LOG OUT

MAY 30, 2016

10:50 a.m.

File descriptors, Unix sockets and other POSIX wizardry

🕒 30 minutes

📖 Python Internals

📊 Intermediate

📍 Oregon 203–204

Remove from Calendar

Machete-mode debugging: Hacking your way out of a tight spot

🕒 30 minutes

📖 Best Practices & Patterns

📊 Intermediate

📍 Oregon 201–202

Add to Calendar

Python as a configuration language

🕒 30 minutes

📖 Systems Administration

📊 Novice

📍 Portland 254–255

Add to Calendar

Seriously Strong Security on a Shoestring (CW)

🕒 30 minutes

📖 Security

📊 Intermediate

📍 Portland 252–253

Add to Calendar



# WebSockets







- **Single TCP connection**
- **Low latency**
- **Full-duplex**
- **Nearly universal support**

# Web Sockets - LS

Bidirectional communication technology for web apps

Global	90.53%	+	0.46%	=	90.98%
unprefixed:	90.53%	+	0.38%	=	90.9%
U.S.A.	96.47%	+	0.49%	=	96.96%
unprefixed:	96.47%	+	0.47%	=	96.94%

Current aligned

Usage relative

Show all

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Chrome for Android
			29						
			45					4.3	
			49					4.4	
8		46	50			9.2		4.4.4	
11	13	47	51	9.1	38	9.3	all	50	51
	14	48	52	10	39				
		49	53	TP	40				
		50	54						



**Welcome to JavaScript Frederick**



```
var ws = new WebSocket(  
    'wss://pycon.guide/presentations')  
  
ws.onmessage = function(msg) {  
    console.log('The server said: ' + msg.data);  
}  
  
ws.send('Hello, friend.')
```

```
var ws = new ReconnectingWebSocket(  
    'wss://pycon.guide/presentations')  
  
ws.onmessage = function(msg) {  
    console.log('The server said: ' + msg.data);  
}  
  
ws.send('Hello, friend.')
```

- **open**
- **error**
- **message**
- **close**



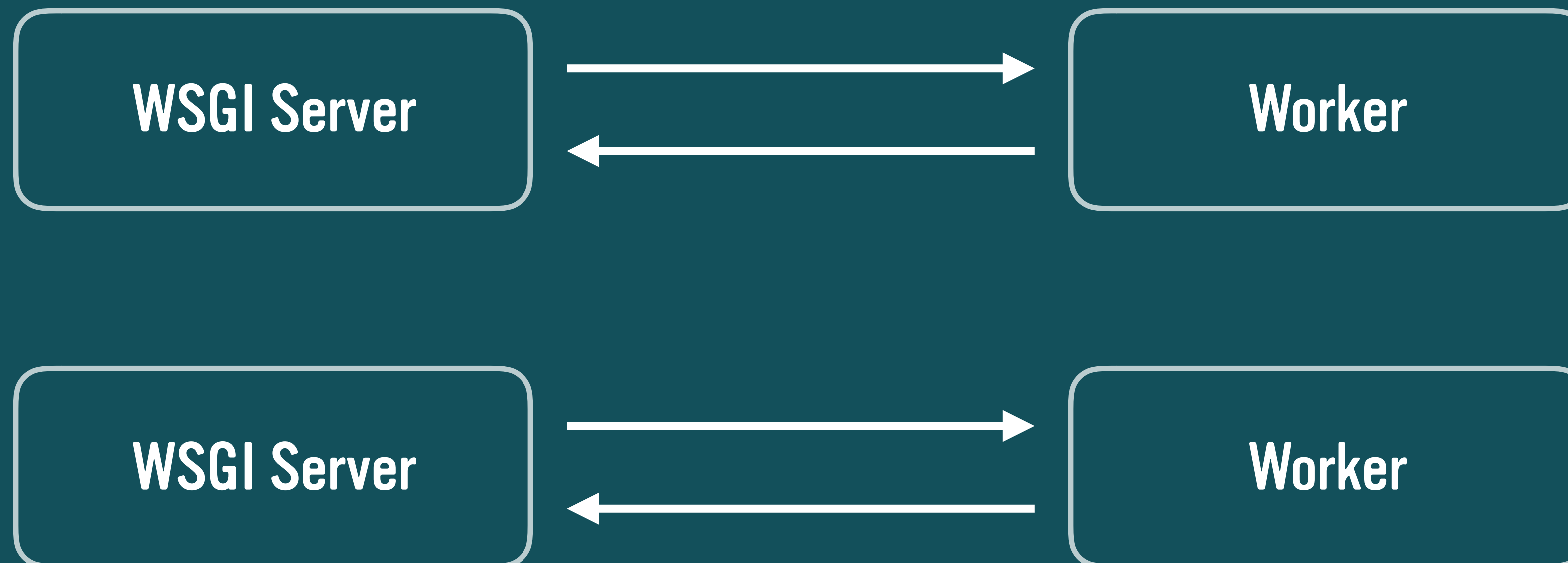
Payload? ヽ\_(ツ)\_ノ

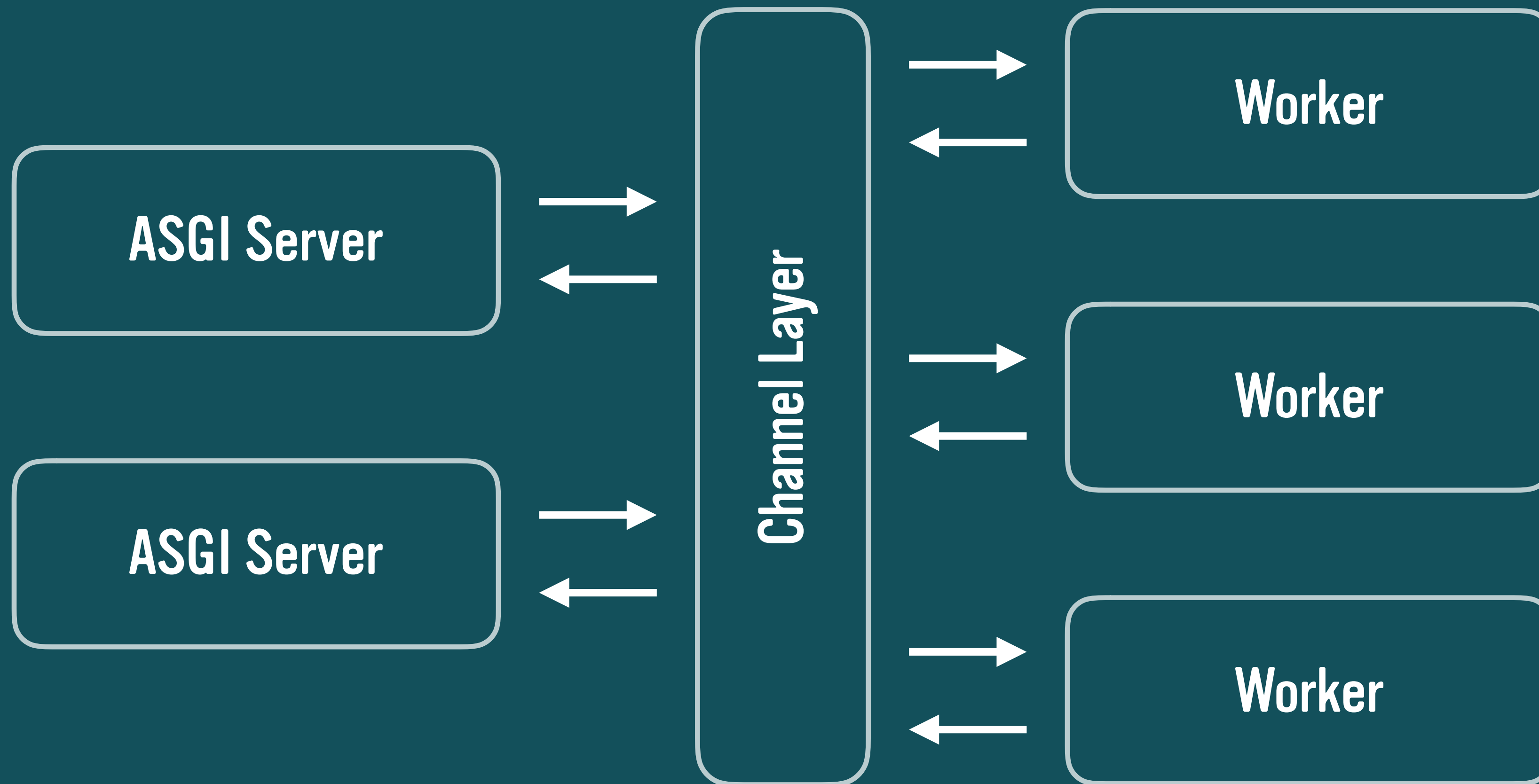
```
ws.send('interested:2397');
```



**Welcome Back to Python Frederick**







# Redis

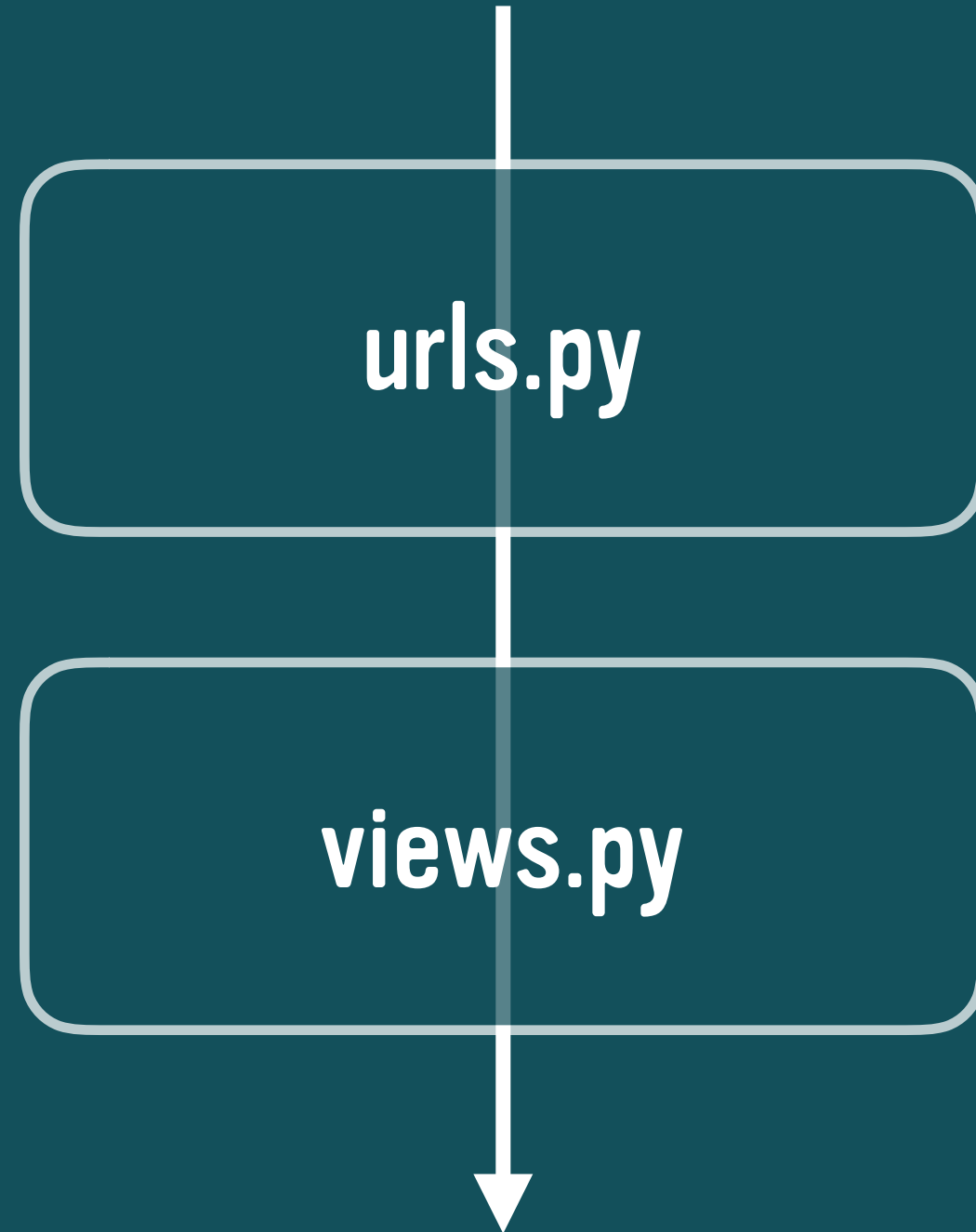
Oh, POSIX IPC and in-memory too

Request

urls.py

views.py

Response





**Request**

**urls.py**

**views.py**

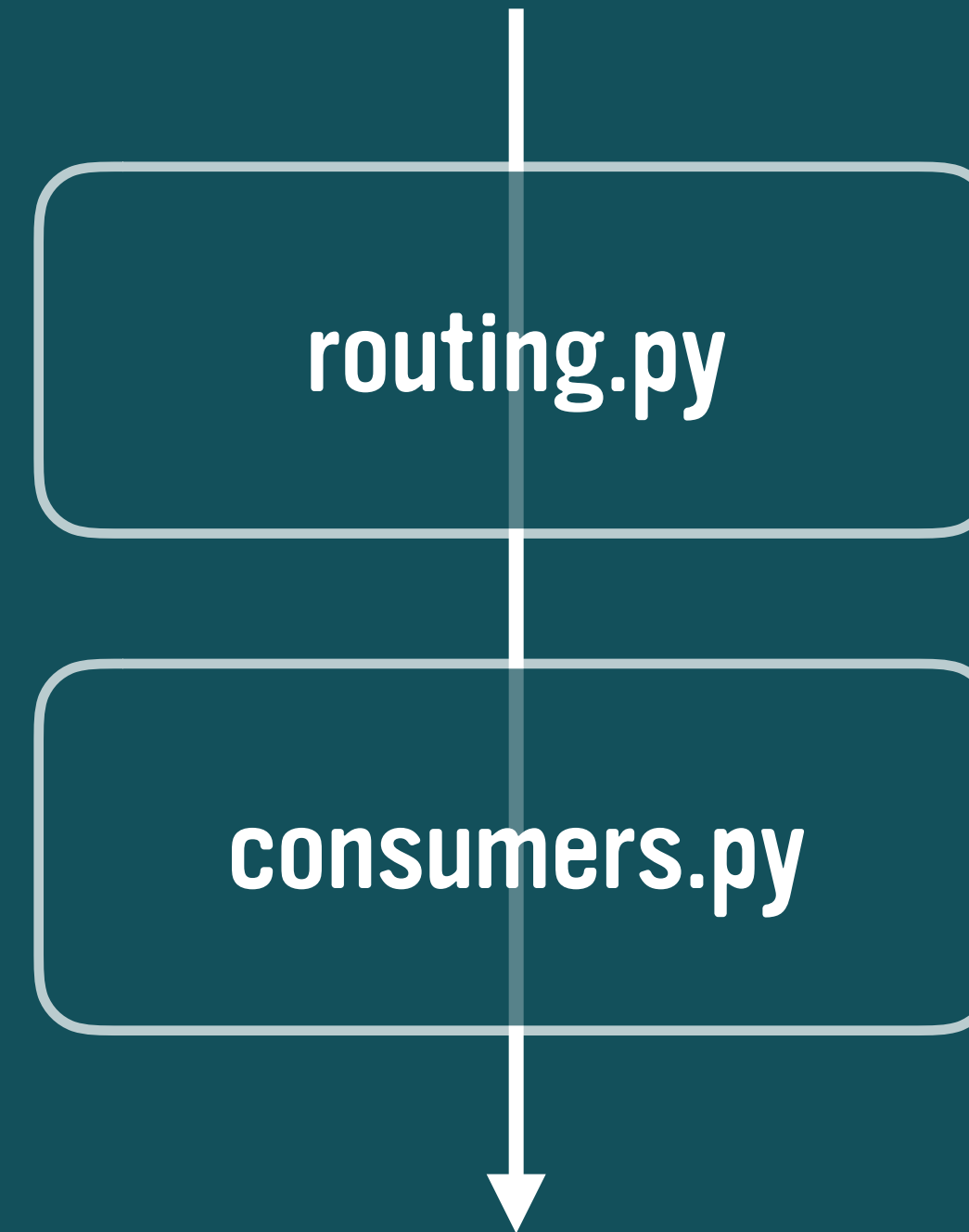
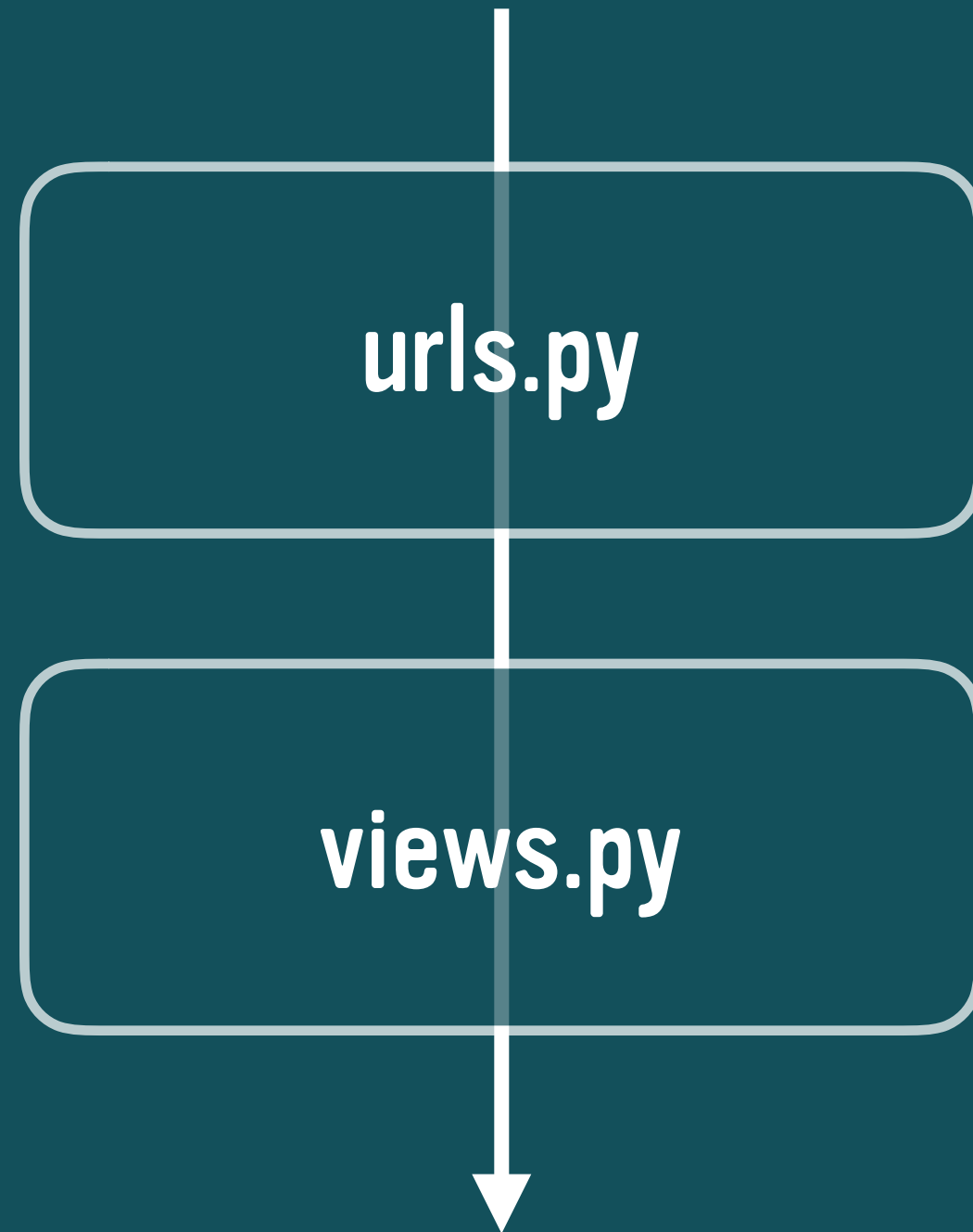
**Response**

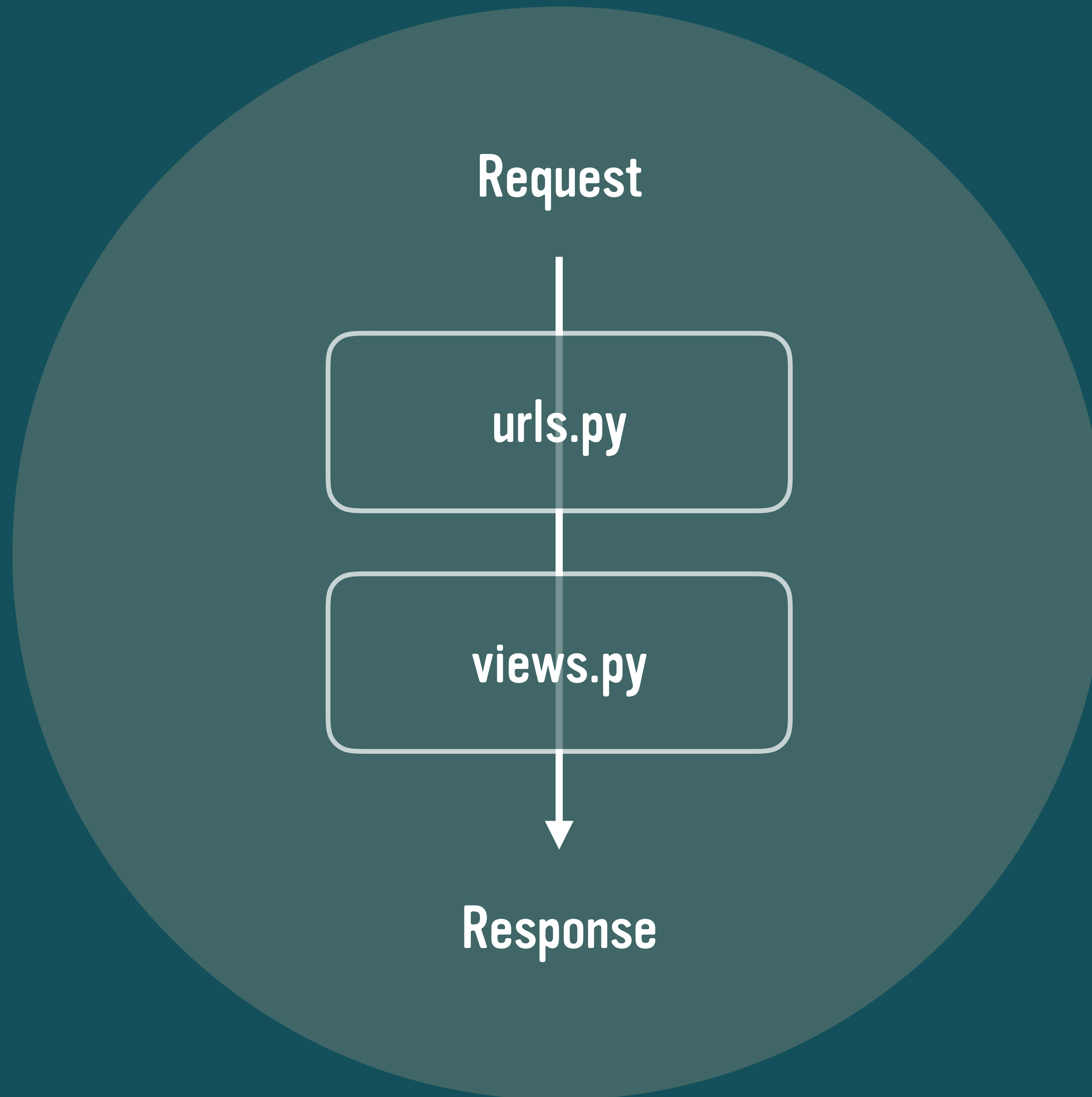
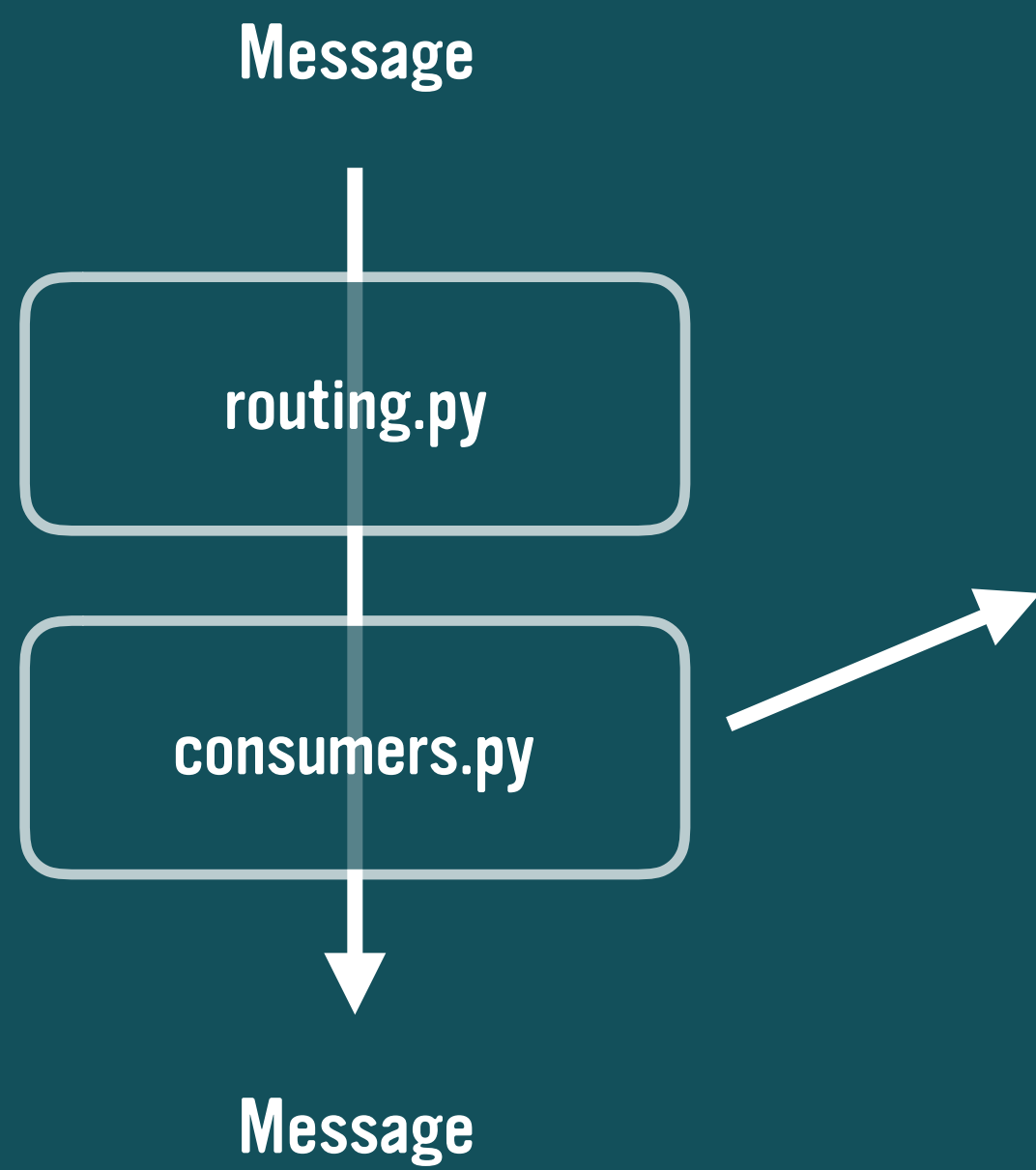
**Message**

**routing.py**

**consumers.py**

**Message**





```
$ pip install channels
```

```
CHANNEL_LAYERS = {
    'default': {
        'BACKEND':
            'asgi_redis.RedisChannelLayer',
        'CONFIG': {
            'hosts':
                ['redis://localhost:6379/0'],
        },
        'ROUTING':
            'pyconguide.routing.channel_routing',
    },
}
```



```
from channels.routing import route
from pyconguide.consumers import \
    ws_connect, ws_message, ws_disconnect

channel_routing = [
    route('websocket.connect', ws_connect),
    route('websocket.receive', ws_message),
    route('websocket.disconnect', ws_disconnect),
]
```

```
def ws_connect(message):  
    print('Client connected')  
  
def ws_message(message):  
    data = message.content.get('text')  
    cmd, pk = data.split(':')
```

```
def ws_message(message):  
    message.reply_channel.send(  
        {'text': 'OK, I got it. Stop.'})
```

```
from channels.auth import (channel_session_user,  
                           channel_session_user_from_http)
```

```
@channel_session_user_from_http
```

```
def ws_connect(message):  
    pass
```

```
@channel_session_user
```

```
def ws_message(message):  
    print('message from {}'.format(  
        message.user.username))
```





# TODO

```
@channel_session_user_from_http
```

```
def ws_connect(message):
```

```
    Group(message.user.username).add(  
        message.reply_channel)
```


```
@receiver(post_save, sender=Interest)
```

```
def send_update(sender, instance, **kwargs):
```

```
    Group(instance.user.username).send({  
        'text': 'interested:{}'.format(instance.pk),  
    })
```

# Deployment





# WSGI → ASGI

+ dual mode



```
import os
```

```
os.environ.setdefault(  
    "DJANGO_SETTINGS_MODULE", "pyconguide.settings")
```

```
from django.core.wsgi import get_wsgi_application
```

```
application = get_wsgi_application()
```

```
import os
```

```
import channels.asgi
```

```
os.environ.setdefault(
```

```
    "DJANGO_SETTINGS_MODULE", "pyconguide.settings")
```

```
channel_layer = channels.asgi.get_channel_layer()
```

# The Future

# More About Channels

Django Channels

<https://channels.readthedocs.io/>

Django Channels Examples

<https://github.com/andrewgodwin/channels-examples>

Reinventing Django for the Real-Time Web

<https://www.youtube.com/watch?v=2sEPipctTxw>



# More About PyCon Guide

<https://pycon.guide>

GitHub

<https://github.com/istrategylabs/pycon-guide>

# Wonderful People

Olivia Cheng

Ben Schrijnemakers

Sarah-Jaine Szekeresh



# Thanks!

@jcarbaugh

