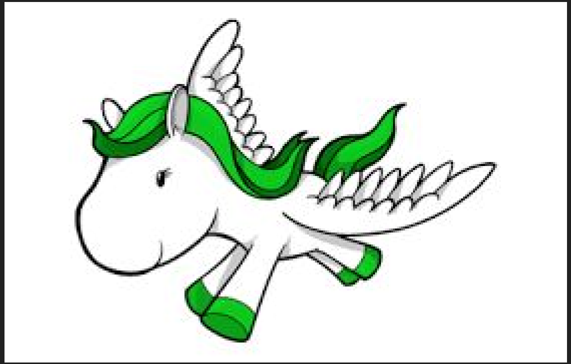
Django Channels

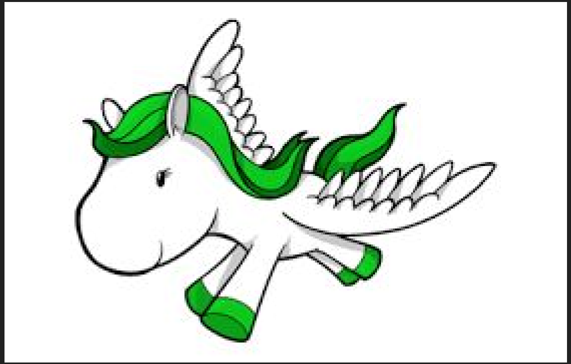
Teaching a mature framework new tricks

@rony_sheer

If you had pick a web framework for a new project, what would you choose?

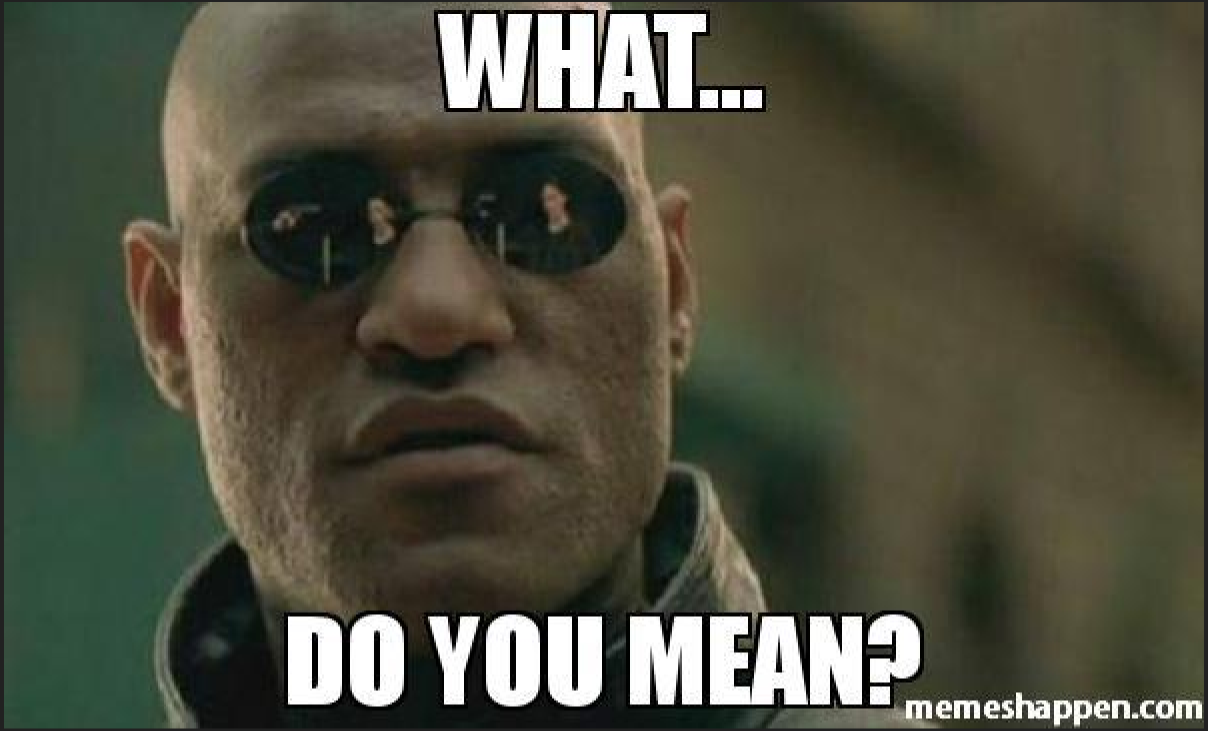


django



But isn’t Django like from 2008… like Web 1.0?

But isn't django like from 2008… Like web 1.0


WHAT...
DO YOU MEAN?
memeshappen.com

- Slow
- What about
- Rest-api
- Async
- Slowww

OH HELL NO
YOU DID NOT JUST GO THERE
memegenerator.net

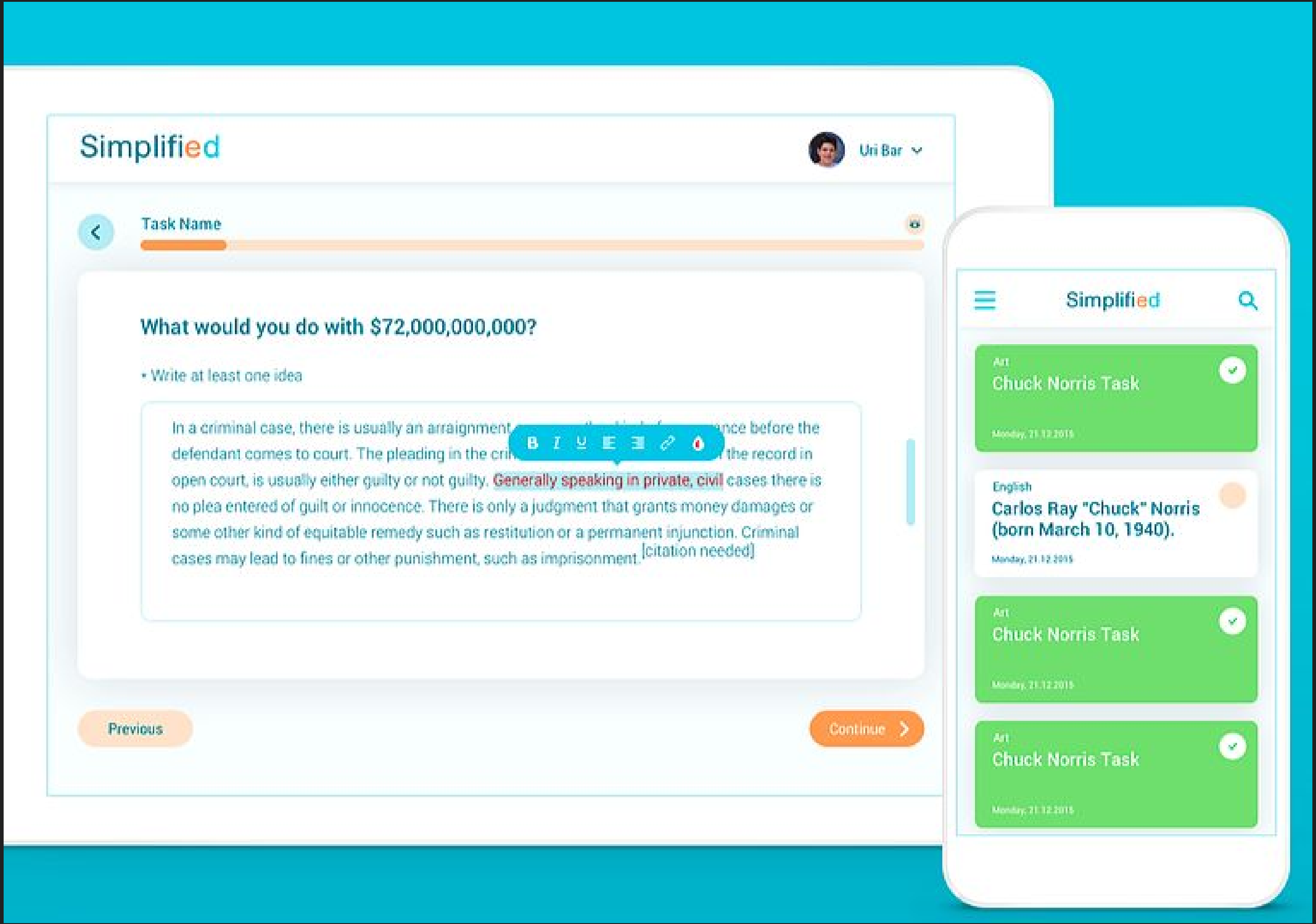
Common workarounds

- Use Tornado
- Use NodeJS
- SAAS like Pusher

Until...

Who is this guy?

- Teacher
- I coded a few apps to help teachers
    - *(Then I went on to code a little platform to help teachers)*
- Python makes me happy
- @rony_sheer on twitter
- simplifiedlms.com



What would you do with $72,000,000,000?

• Write at least one idea

In a criminal case, there is usually an arraignment [illegible] ence before the defendant comes to court. The pleading in the cri[illegible] the record in open court, is usually either guilty or not guilty. Generally speaking in private, civil cases there is no plea entered of guilt or innocence. There is only a judgment that grants money damages or some other kind of equitable remedy such as restitution or a permanent injunction. Criminal cases may lead to fines or other punishment, such as imprisonment.[citation needed]

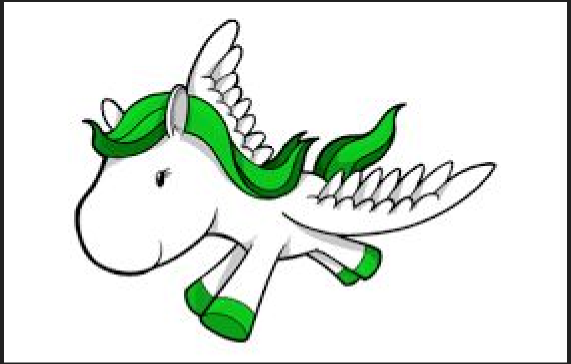
Art
Chuck Norris Task
Monday, 21.12.2015

English
Carlos Ray "Chuck" Norris (born March 10, 1940).
Monday, 21.12.2015

Art
Chuck Norris Task
Monday, 21.12.2015

Art
Chuck Norris Task
Monday, 21.12.2015

Our stack



Why Django?

- Documentation is second to none
- Very sensible defaults
- Abstracts away the right stuff
- Gives you control of level of abstraction

```
import …

def hey_view(request):
    return HttpResponse(‘hey’)


urlpatterns = (
    url(r’^$’, hey_view),
)
```

```
class HeyView(TemplateView):
    …
```

Or

```
urlpatterns = [
    url(r'^$',
        TemplateView.as_view(template_name='index.html'))
]
```

Testing

```
assertEqual(response.status_code, 200)
```

Pythonic

```
import this
```

Djangonic

- Superbly documented
- Let’s you find abstraction sweetspot
- Very testable
- Sensible defaults
- Nudges towards time-tested architectural patterns

Djangonic + WebSockets?

- Superbly documented
- Let’s you find abstraction sweetspot
- Very testable
- Sensible defaults
- Nudges towards time-tested architectural patterns

WebSockets

1. Connect via HTTPS or HTTP
2. Request WebSockets connection
3. Request is accepted or rejected
4. Two-way send/receive interaction
5. Disconnect

Use cases

- Messaging
- Instant updates
- Gaming
- Art

WSGI is very much about Request and Response

- Great for HTTP
- Not so great for WebSockets

ASGI

- New interface
- Suitable for HTTP, HTTP 2, WebSockets

So how do we make WebSockets Djangonic?

Channels

A channel

“an ordered, first-in first-out queue with message expiry and at-most-once delivery to only one listener at a time.” — channels docs

A channel

“an ordered, first-in first-out queue with message expiry and at-most-once delivery to only one listener at a time.” — channels docs

Huh? Show me the code!

View vs. consumer

```
def  post_list(request):
    ...
    return render(
        request, "list.html",
        {"posts": posts})
```

```
def  ws_message(message):
    ...
    message.reply_channel.send({
        “text”: message.context[‘text’],
    })
```

Channel = FIFO queue

Consumer: Listens to a channel

```
def  ws_message(message):
    ...
    message.reply_channel.send({
        “text”: message.context[‘text’],
    })
```

Consumer: Listens to a channel

```
class ChatConsumer(JsonWebsocketConsumer):
    strict_ordering = False
    slight_ordering = True

    def connection_groups(self, **kwargs):
        return [‘friends’]

    def receive(self,content):
        self.group_send(content)
```

Routing: Look familiar?

```
channel_routing = [
    route_class(ChatConsumer, path=r’^/chat’),
]
```

You know more
than you’d think

Channel layers

```
CHANNEL_LAYERS = {
    "default": {
        "BACKEND": "asgiref.inmemory.ChannelLayer",
        "ROUTING": "myproject.routing.channel_routing",
    },
}
```

Channel layers(production)

```
CHANNEL_LAYERS = {
    "default": {
        "BACKEND": "asgi_redis.RedisChannelLayer",
        "CONFIG": {
            "hosts": [("localhost", 6379)],
        },
        "ROUTING": "myproject.routing.channel_routing",
    },
}
```

Daphne

```python
class SketchConsumer(JsonWebsocketConsumer):
    strict_ordering = False
    slight_ordering = True

    def connection_groups(self, **kwargs):
        return ['draw']

    def receive(self, content, **kwargs):
        self.group_send('draw', content)
```

Javascript

```
var socket = new WebSocket("ws://" + window.location.host + "/draw");
```

```
// es 6
let socket = new WebSocket(`ws://${ window.location.host }/draw`);
```

```
data = JSON.stringify({
    'x': mouseX,
    'y': mouseY
});

socket.send(data);
```

Resources

- Channels documentation
- ASGI documentation
- Andrew Godwin's Channels talk (Django Under the Hood)
- Get started with Channels by Jacob Kaplan-Moss
- P5js + Daniel Shiffman tuorials

Thank you