

Usando Python y SocketIO para aplicaciones en tiempo real

Sobre mí ;)

Oscar Ramirez

@tuxskar

Python Web Developer

@RavenPack International



Usos de aplicaciones de tiempo real

Chats

Intercambio de mensajería instantánea entre usuarios

Juegos

Actualizaciones de escenarios instantáneos y notificación a todos los usuarios

Apps reactivas

Aplicaciones basadas en eventos

Empecemos con la demo

<https://pycones2017.oramirez.com>

Características principales

Actualizaciones en tiempo real

Interacción de diferentes usuarios

Reacción ante eventos

Obtención del número de usuarios conectados

¿Qué es Socket.IO?

Librería original en JavaScript para aplicaciones en tiempo real

API común cliente y servidor

Usa WebSockets como protocolo primario y long polling como fallback

Permite el uso de canales bidireccionales siempre abiertos

Óptima para aplicaciones orientadas a eventos

¿Quién entiende Socket.IO?

JavaScript, Java, Swift, C, C++, etc:

<https://github.com/socketio>

Python!

<https://github.com/miguelgrinberg/python-socketio>

Flask-socketio

<https://flask-socketio.readthedocs.io/en/latest/>

Socket.io API

Namespaces

Agrupación física de usuarios mediante sockets tcp

Connect – Disconnect

```
on('connect', namespace='/grid-game')  
on('disconnect', namespace='/grid-game')
```

Rooms

Agrupación lógica de usuarios dentro de grupos

Join – Leave

```
join_room(room)  
leave_room(room)
```

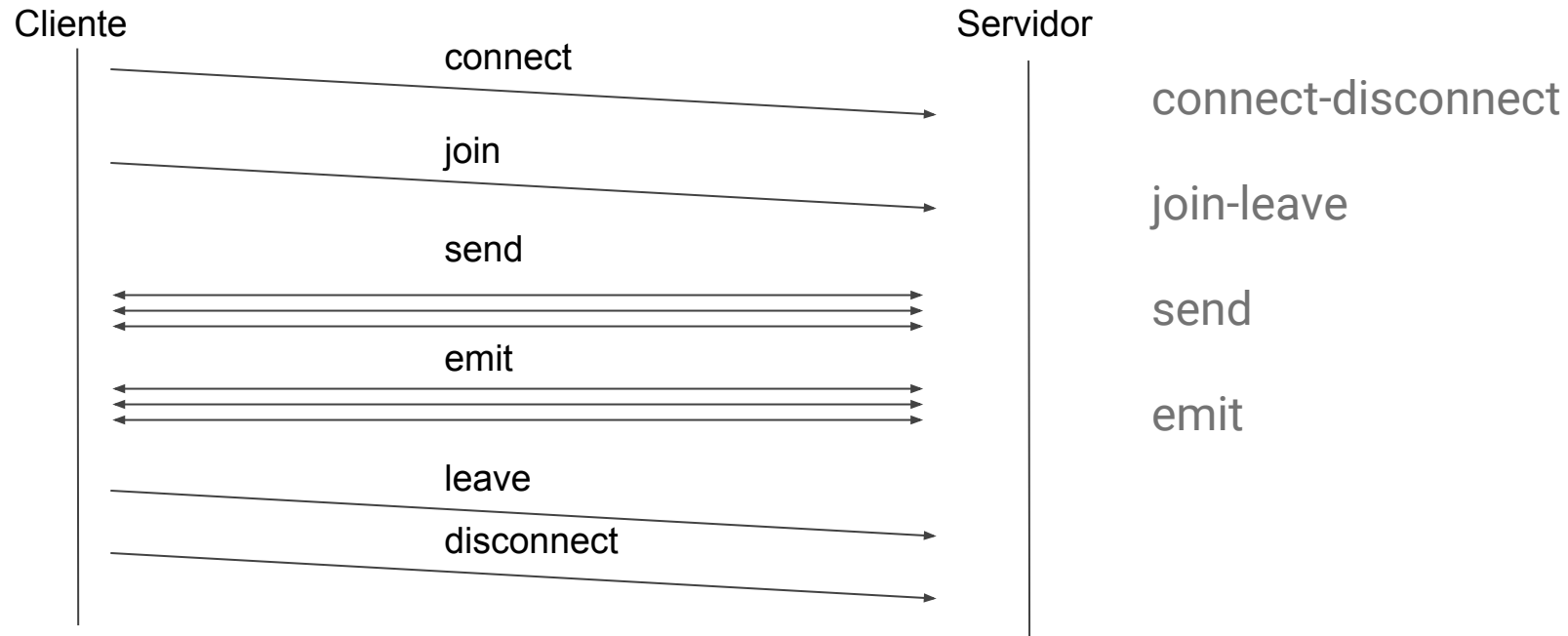
Mensajes

Envío de mensajes bidireccional con o sin etiqueta para el mensaje

Emit – Send

```
send(payload)  
emit('grid_status', payload)  
emit('status', payload, room=room)  
emit('std', payload, broadcast=True)
```


Socket.IO workflow



Flask SocketIO

```
>>> pip install flask-socketio
```

eventlet

Soporta long polling y
WebSockets directamente

gevent +
gevent-websocket

Necesitas instalar ambos
para soportar websocket y
long-polling

Connect - disconnect

```
app = Flask(__name__)  
socketio = SocketIO(app)
```

```
@socketio.on('connect', namespace='/grid-game')
```

```
def on_connect(msg):
```

```
    ...
```

```
@socketio.on('disconnect', namespace='/grid-game')
```

```
def on_disconnect():
```

```
    ...
```

```
if __name__ == '__main__':
```

```
    socketio.run(app)
```

Servidor

```
var namespace = '/grid-game',  
url = location.protocol + '//' + document.domain + ':' +  
location.port + namespace
```

```
var socket = io.connect(url);
```

Cliente

Namespace - rooms

Send - emit

```
@socketio.on('join', namespace='/grid-game')
def on_join(msg):
    join_room(msg.room)
    send(dict(user_id='super-id'))

@socketio.on('leave', namespace='/grid-game')
def on_leave(msg):
    leave_room(msg.room)
    user_connected = len(users_connected)
    emit('l_users', user_connected, room=msg.room)

@socketio.on('message', namespace='/grid-game')
def on_message(msg):
    send(dict(userId=request.sid))
```

Servidor

```
socket.emit('join', {room: 'awesome-room'});

socket.emit('leave', {room: 'awesome-room'});

socket.send(payload);
```

Cliente

Demo automatizada

<https://pycones2017.oramirez.com>

Testing with Flask-socketIO

```
def test_connect():
    nmsp = '/grid-game'
    client = socketio.test_client(app, namespace=nmsp)
    received = client.get_received(nmsp)

    g = received[0].get
    assert g('args').get('symbol') is not None
    client.disconnect()
```

```
def test_user_move():
    ...
    user_position = position_msg.get('args')[-1].get('pos')
    y_pos = user_position[1]
    client.emit('movement', {'direction': 'up'},
namespace=namespace)
    received = client.get_received(namespace)
    ...
    new_y_pos = new_position[1]
    assert new_y_pos == (y_pos - 1)
    client.disconnect()
```

Desplegando Flask-socketIO

Desarrollo

```
socketio.run(app)
```

Producción

Nginx + Gunicorn + (Redis)

```
gunicorn --worker-class eventlet -w 1 module:app
```

Escalando con Flask-socketIO

Redis

pip install redis

```
socketio = SocketIO(app, message_queue='redis://')
```

RabbitMQ

pip install kombu

```
socketio = SocketIO(app, message_queue='amqp://')
```


¿Preguntas?

Código: <https://github.com/tuxskar/flask-socketio-grid-game>

Oscar Ramirez
tuxskar(at)gmail.com

RavenPack International S.L.
WE ARE HIRING!
www.ravenpack.com/careers

Gracias!