SUDHAUS7

ConFoo.CA
DEVELOPER CONFERENCE

INTRODUCTION TO REACTPHP AND WEBSOCKETS

CONFOO MONTREAL 2024

FRANK BERGER

A BIT ABOUT ME

- Frank Berger
- Head of Engineering Sudhaus7, a label of B-Factor GmbH, Stuttgart
- Started as an Unix System administrator who also codes in 1996
- Web development since 2000
- Does TYPO3 since 2005

WEBSOCKETS

Protocol ws:// and wss://

Standardized in 2011 as RFC 6455

Located in layer 7 of the OSI model (on top of layer 4 TCP)

Upgrade protocol on top of HTTP

Allows full-duplex communication

Support for all major browsers

Support in all major languages

REACTPHP (& RATCHET)

The solution in PHP

REACTPHP

ReactPHP is an event driven, non-blocking I/O framework for webservices in PHP

(think NodeJS on the server, but with PHP)

Runs on a separate port on the server, needs to be proxied through the webserver

Handles thousands of connections and events in a single PHP process

Does not need any special extensions in PHP, but leverages new features in PHP 8.1 like fibers

RATCHET

A websocket implementation with ReactPHP

BASIC IMPLEMENTATION (SERVER SIDE)

```
composer require cboden/ratchet
```

```
use Ratchet\MessageComponentInterface;
use Ratchet\ConnectionInterface;

require_once __DIR__ . "/vendor/autoload.php";

$app = new Ratchet\App('0.0.0.0', 8080);
$app->route('/echo', new Ratchet\Server\EchoServer, array('*'));
$app->run();
```

CLIENT SIDE

```html
<ul id="log">

</ul>
```

```javascript
var conn = new WebSocket('ws://0.0.0.0:8080/echo');
conn.onopen = function(ev) {
  conn.send('hello');
}
conn.onmessage = function(ev) {
  let li=document.createElement('li');
  li.innerText = ev.data;
  document.getElementById('log').append(li);
}
```

CLIENT SIDE AVAILABLE EVENTS

```
conn.onopen
```

```
conn.onmessage
```

```
conn.onclose
```

```
conn.onerror
```

```
conn.readyState === 1
```

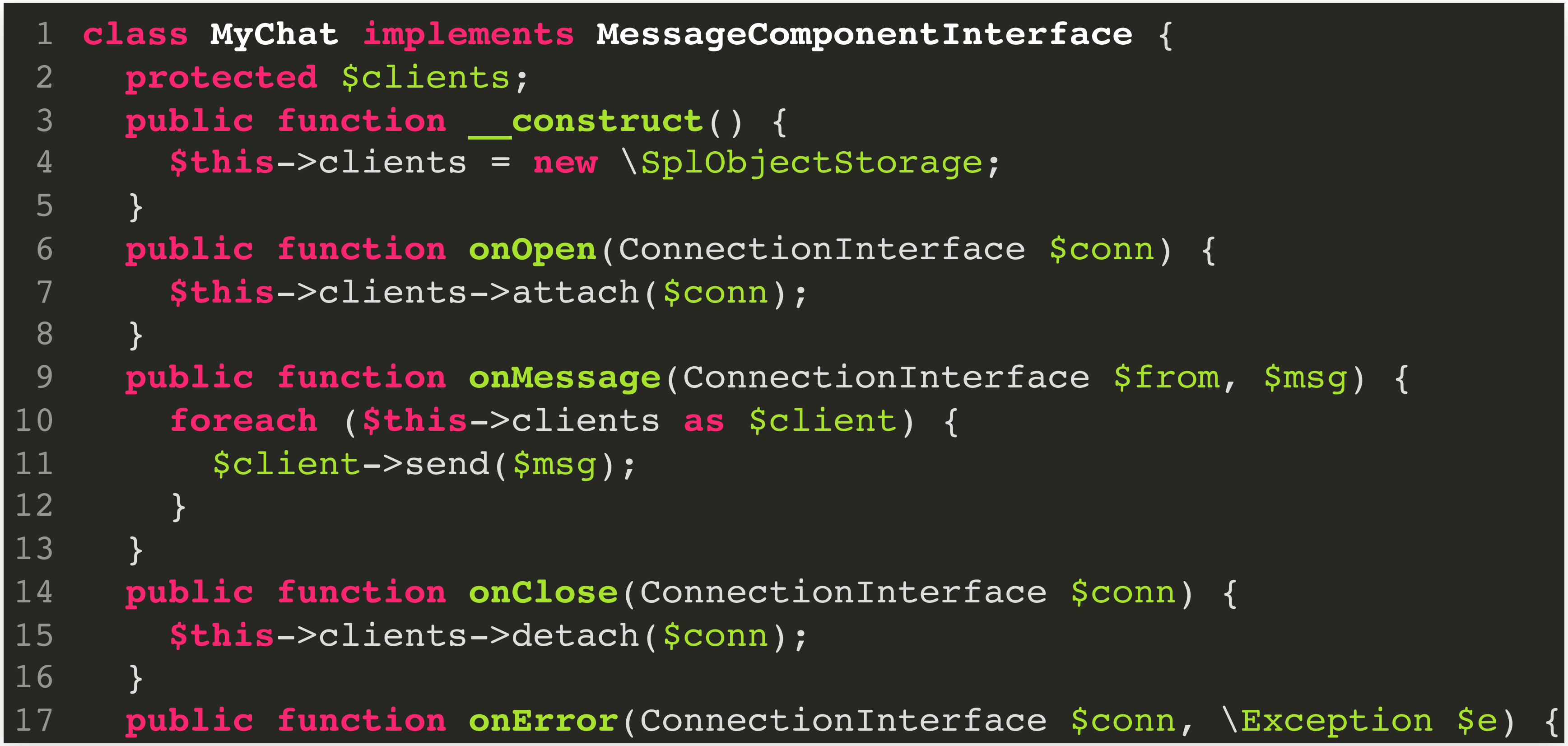
manage the open state

use custom JS events

Nothing happening much yet, how about we try this in two browsers?

DISTRIBUTING THE MESSAGE

CHAT SERVER SIDE



```php
class MyChat implements MessageComponentInterface {
  protected $clients;
  public function __construct() {
    $this->clients = new \SplObjectStorage;
  }
  public function onOpen(ConnectionInterface $conn) {
    $this->clients->attach($conn);
  }
  public function onMessage(ConnectionInterface $from, $msg) {
    foreach ($this->clients as $client) {
      $client->send($msg);
    }
  }
  public function onClose(ConnectionInterface $conn) {
    $this->clients->detach($conn);
  }
  public function onError(ConnectionInterface $conn, \Exception $e) {
```

```php
$app = new Ratchet\App('0.0.0.0', 8080);
$app->route('/chat', new MyChat, array('*'));
$app->run();
```

LETS TRY IT

LETS IMPLEMENT A SIMPLE CHAT

FURTHER IDEAS

Using redis for backlog

Adding user profiles

Direct messages

.. and so on

BUT IT IS RATCHET, NOT REACTPHP!!!

Don't worry - you have options..

Either - generate the event loop beforehand

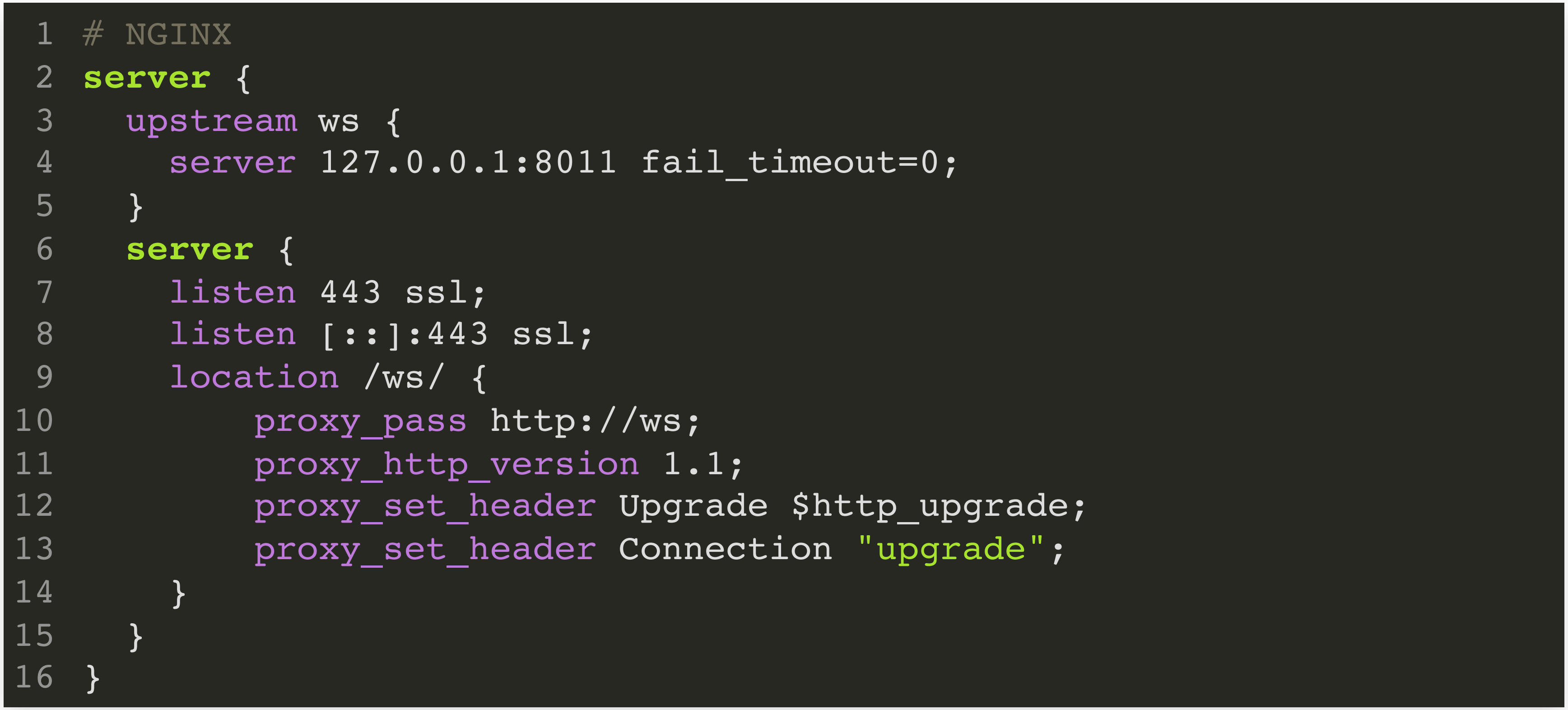
```php
$loop = new StreamSelectLoop();
$redisevents = new OtherReactComponentTalkingToRedisForExample($loop);
```

and add it to the Ratchet Server

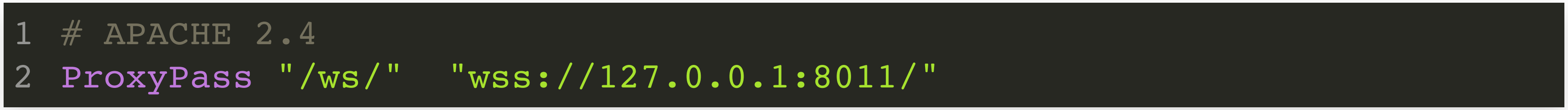
```php
$ws = new \Ratchet\App($hostname, $port, '0.0.0.0', $loop);
```

Or - do manually what \Ratchet\App is doing in registering their part onto the event loop, adding it as a normal event-loop handler

PROXY IT!



```
# NGINX
server {
  upstream ws {
    server 127.0.0.1:8011 fail_timeout=0;
  }
  server {
    listen 443 ssl;
    listen [::]:443 ssl;
    location /ws/ {
        proxy_pass http://ws;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }
  }
}
```



```
# APACHE 2.4
ProxyPass "/ws/"  "wss://127.0.0.1:8011/"
```

KEEP IT RUNNING

forever - from the nodejs world

docker - running the "server.php" as an entrypoint (and mark it for restart if it fails)

SUMMARY

Working with websockets is not so hard

So go ahead and start

https://github.com/codeseveneleven/talk-websockets

SUDHAUS7

ConFoo.CA
DEVELOPER CONFERENCE

THANK YOU
I AM HERE FOR THE REST OF THE WEEK

X: @FoppelFB

Fediverse: @foppel@phpc.social

https://sudhaus7.de/

fberger@sudhaus7.de