

# HTTP messages

Modern way of dealing with HTTP in PHP

 @dpcon #dpc16

@hannesvdvreken

**Hi, my name is Hannes.**





**madewithlove.be**

20 × 🥰





# TCS AMSTERDAM MARATHON

**PSR-7**

# **PSR-7**

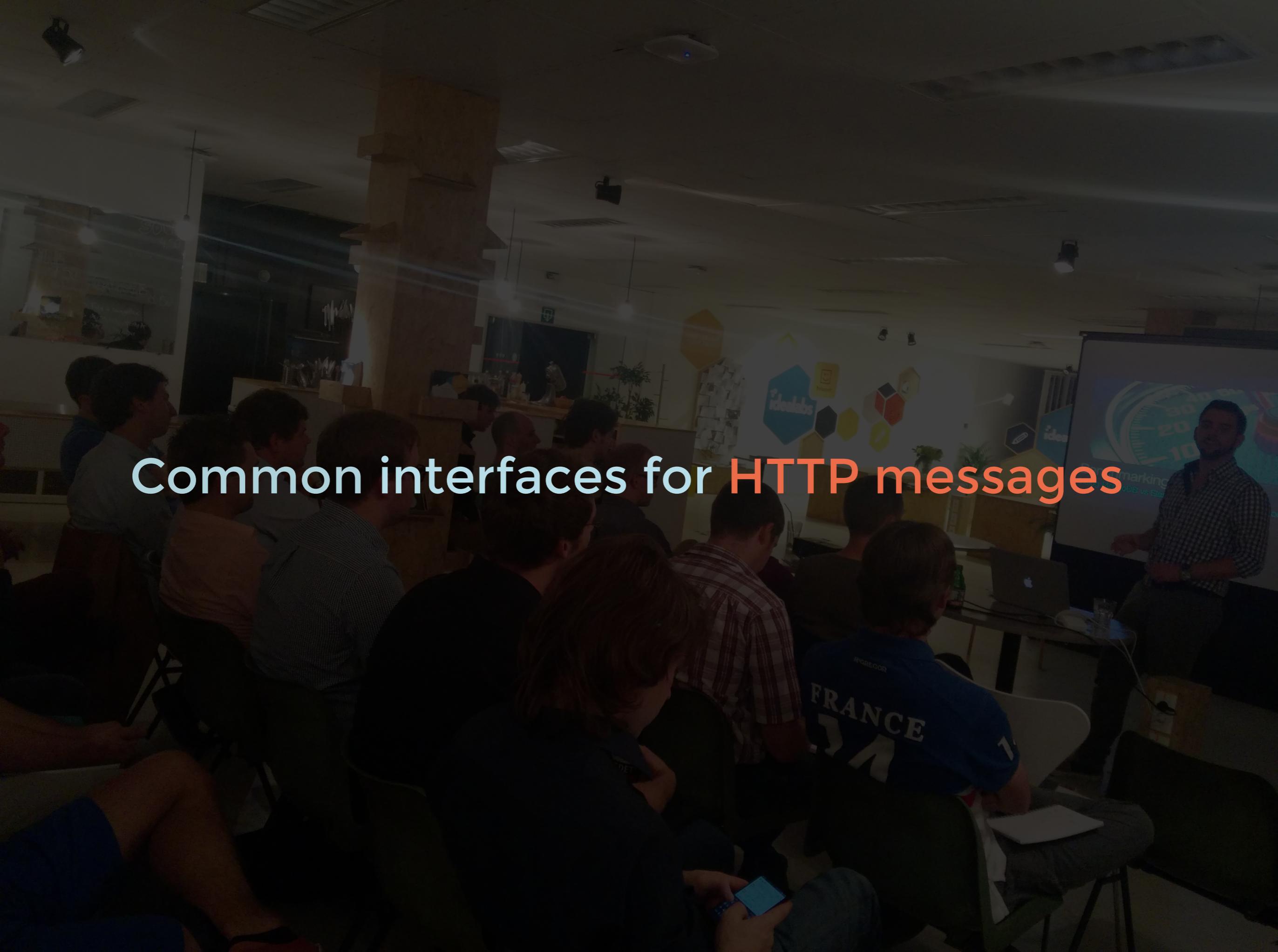
**Common interfaces for HTTP messages**

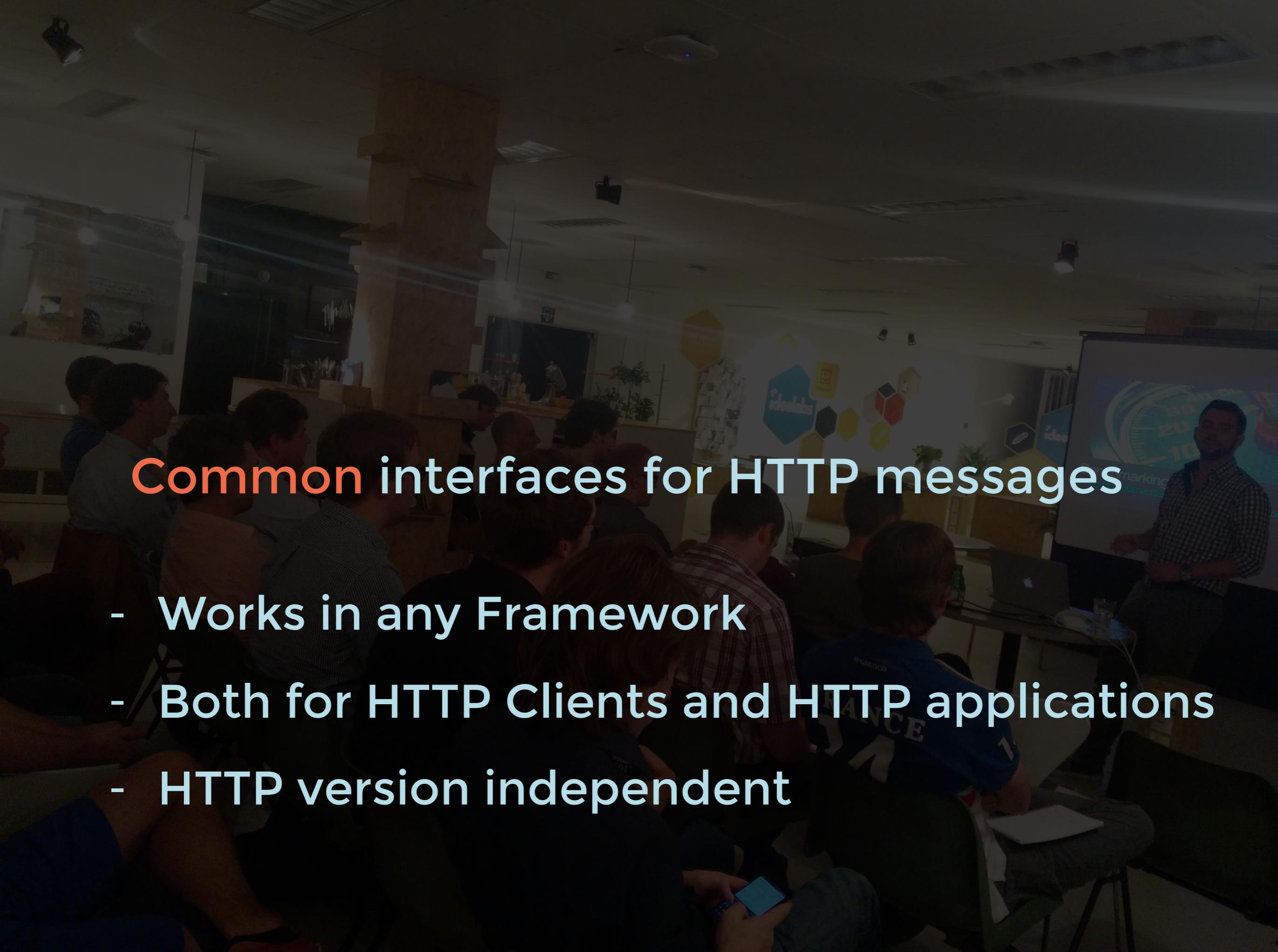
1. [psr/http-message](#)
2. [guzzlehttp/guzzle](#)
3. More PSR-7 usages



**1. psr/http-message**

# Common interfaces for HTTP messages

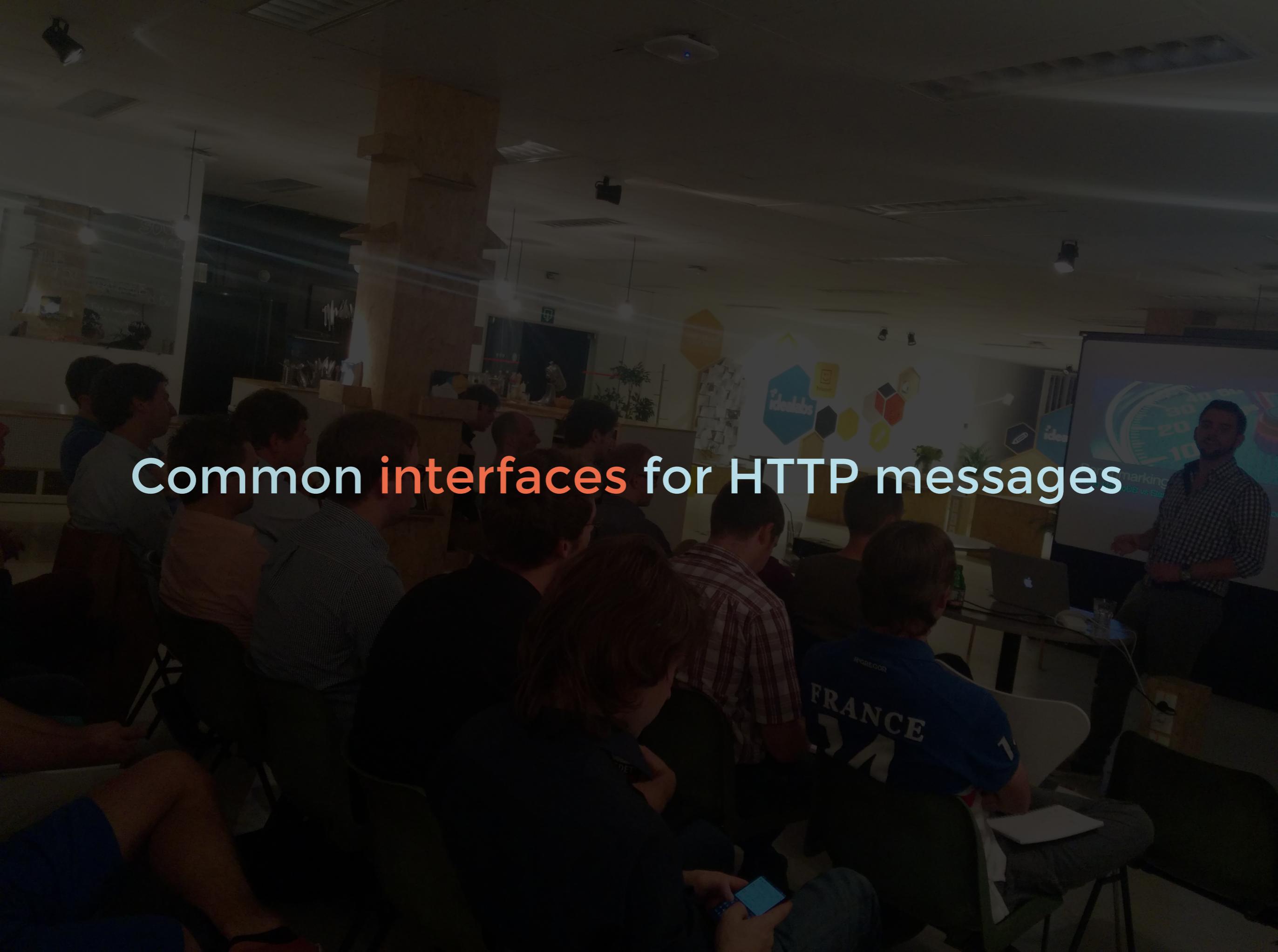




## Common interfaces for HTTP messages

- Works in any Framework
- Both for HTTP Clients and HTTP applications
- HTTP version independent

# Common **interfaces** for HTTP messages



Pre-PSR-7



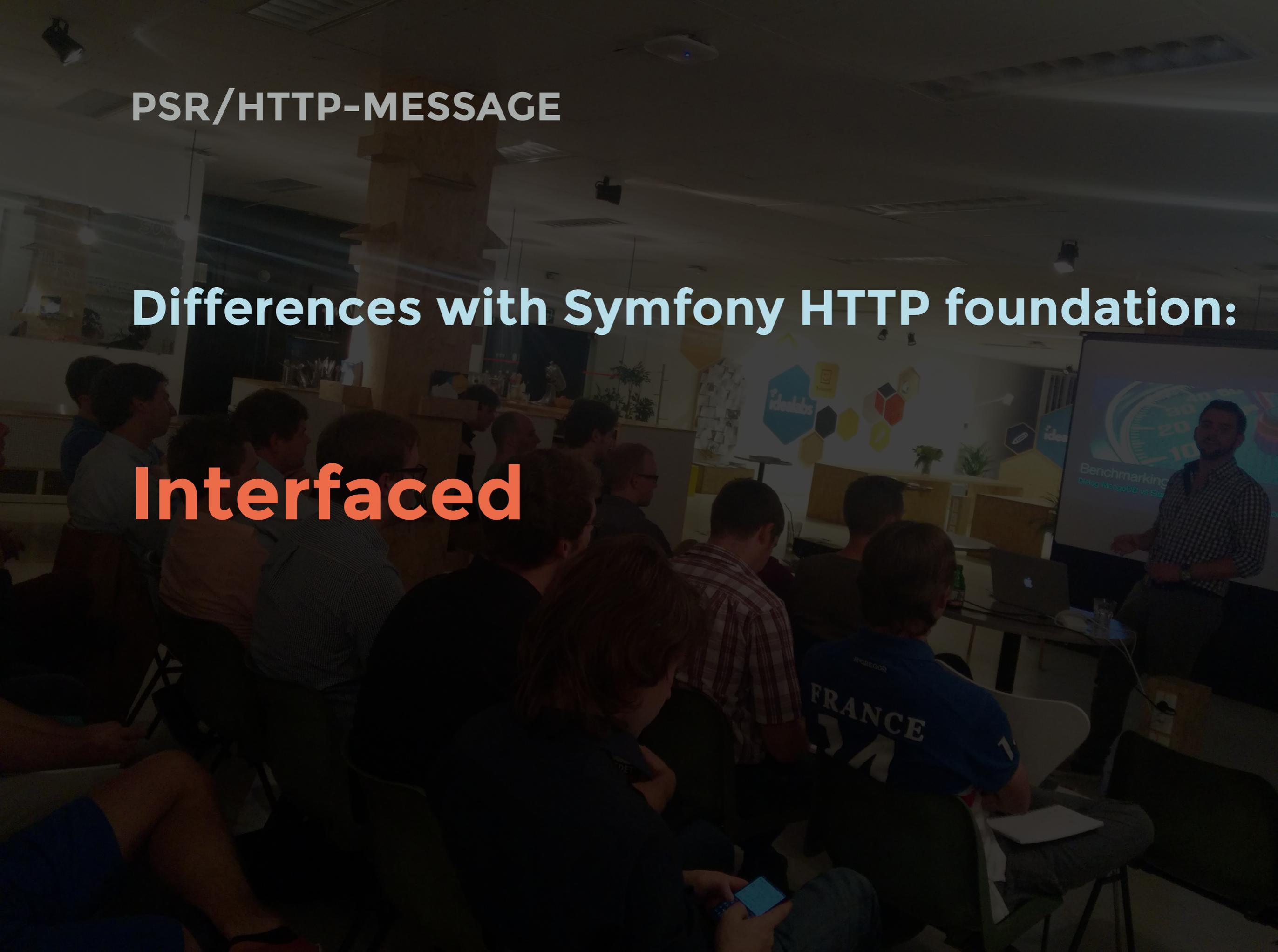
composer req symfony/http-foundation



**PSR/HTTP-MESSAGE**

**Differences with Symfony HTTP foundation:**

**Interfaced**



**PSR/HTTP-MESSAGE**

**Differences with Symfony HTTP foundation:**

**Interfaced**

**Immutable**

## PSR/HTTP-MESSAGE

```
$newRequest = $request  
->withHeader($name, $value);
```

```
var_dump($newRequest === $request);
```

```
> false
```

PSR/HTTP-MESSAGE

let's take a look



# PSR/HTTP-MESSAGE - RECAP

MessageInterface

RequestInterface

ResponseInterface

StreamInterface !

UriInterface

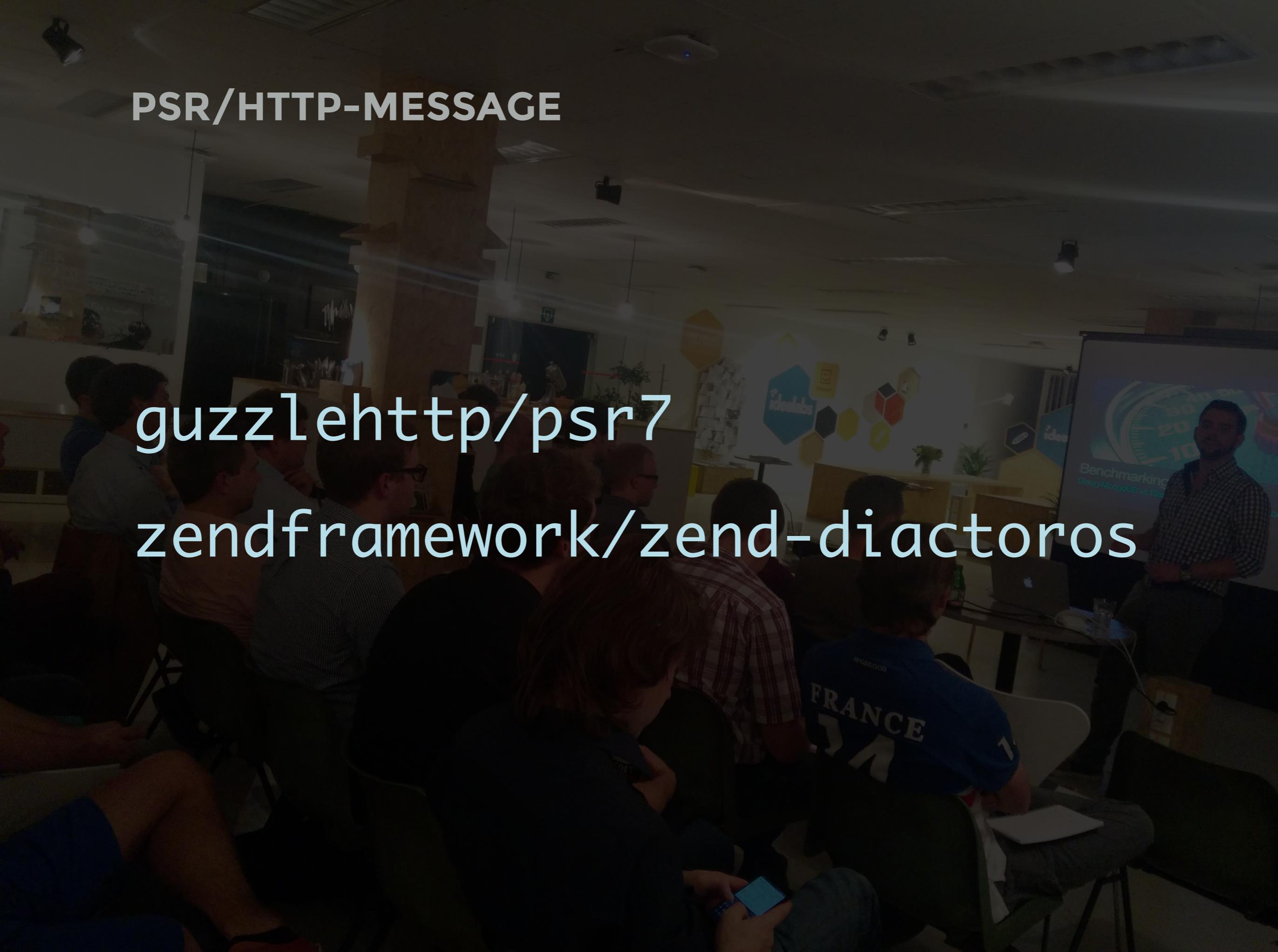
ServerRequestInterface

(`$_GET`, `$_POST`, `$_SERVER`, `$_COOKIE`, `$_FILES`)

**PSR/HTTP-MESSAGE**

**guzzlehttp/psr7**

**zendframework/zend-diactoros**



## **2. guzzlehttp/guzzle**

GUZZLEHTTP/GUZZLE

composer req guzzlehttp/guzzle



**GUZZLEHTTP/GUZZLE**

**composer.json**

**^6.2**



# Packagist Semver Checker

[Learn more about version constraints](#)

guzzlehttp/guzzle

Search

^6.2

stable

## Results for [guzzlehttp/guzzle:^6.2](#)

6.2-dev **6.2.0** 6.1.1 6.1.0 6.0.2 6.0.1 6.0.0 5.3.x-dev 5.3.0 5.2.0 5.1.0  
5.0.3 5.0.2 5.0.1 5.0.0 4.2.3 4.2.2 4.2.1 4.2.0 4.1.8 4.1.7 4.1.6  
...  
v2.6.6 v2.6.5 v2.6.4 v2.6.3 v2.6.2 v2.6.1 v2.6.0 v2.5.0 v2.4.1 v2.4.0  
v2.3.2 v2.2.4 v2.2.3 v2.2.2 v2.2.1 v2.2.0 v2.1.4 v2.1.3 v2.1.2 v2.1.1  
v2.1.0 v2.0.5 v2.0.4 v2.0.3 v2.0.2 v2.0.1 v2.0.0 v1.0.4 v1.0.3



madewithlove

semver.mwl.be

[GUZZLEHTTP/GUZZLE](http://GUZZLEHTTP/GUZZLE)

# What's new?



FRANCE

# GUZZLEHTTP/GUZZLE - WHAT'S NEW

# PSR-7



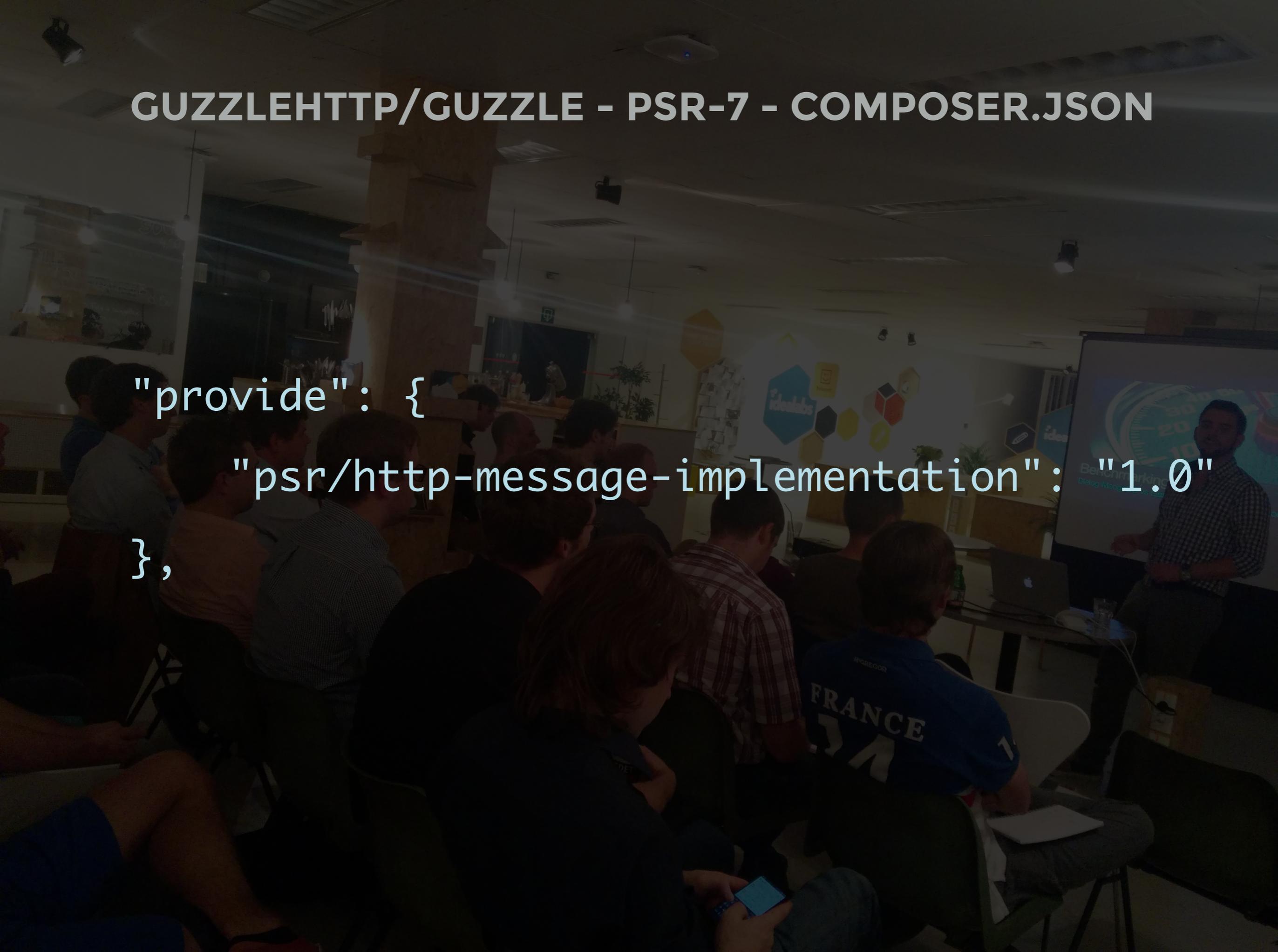
GUZZLEHTTP/GUZZLE - PSR-7

[guzzlehttp/psr7](https://guzzlehttp.org/psr7)



# GUZZLEHTTP/GUZZLE - PSR-7 - COMPOSER.JSON

```
"provide": {  
  "psr/http-message-implementation": "1.0"  
},
```





Packagist is the main [Composer](#) repository. It aggregates public PHP packages installable with Composer.

[psr/http-message-implementation](#) (Virtual Package)

↓ 0

★ 0

[psr/http-message](#)

PHP

↓ 1 207 983

Common interface for HTTP messages

★ 191



Packagist is the main [Composer](#) repository. It aggregates public PHP packages installable with Composer.

The following packages **provide** psr/http-message-implementation

[guzzlehttp/psr7](#)

PSR-7 message implementation

PHP

↓ 1 110 065

★ 58

[zendframework/zend-diactoros](#)

PSR HTTP Message implementations

PHP

↓ 54 940

★ 187

## GUZZLEHTTP/GUZZLE - PSR-7

```
use GuzzleHttp\Psr7\Request;
```

```
$psrRequest = new Request(  
    'GET', $uri, $headers, $body  
);
```

# GUZZLEHTTP/GUZZLE - PSR-7

```
$client = new GuzzleHttp\Client();  
$client->send($psrRequest);
```



GUZZLEHTTP/GUZZLE - WHAT'S NEW

What else?



# GUZZLEHTTP/GUZZLE - PROMISES

# Promises



# GUZZLEHTTP/GUZZLE - PROMISES

Not so new: ( $\geq 5.0.0$ )

# Promises



# GUZZLEHTTP/GUZZLE - PSR-7

guzzlehttp/promise



# GUZZLEHTTP/GUZZLE - PROMISES

```
use Psr\Http\Message\ResponseInterface as Res;
```

```
$promise = $client->sendAsync($request);
```

```
$promise->then(function (Res $response) {  
    $response->getBody();  
});
```

```
$promise->wait();
```

**GUZZLEHTTP/GUZZLE**

# Demo

<https://github.com/hannesvdvreken/psr7-demo/tree/master/1-guzzle>

<https://github.com/hannesvdvreken/psr7-demo/tree/master/2-promise>

<https://github.com/hannesvdvreken/psr7-demo/tree/master/3-multi>

GUZZLEHTTP/GUZZLE - WHAT'S NEW

What else?



# GUZZLEHTTP/GUZZLE - EVENTING

( $\geq 4.0.0$ ,  $< 6.0.0$ )

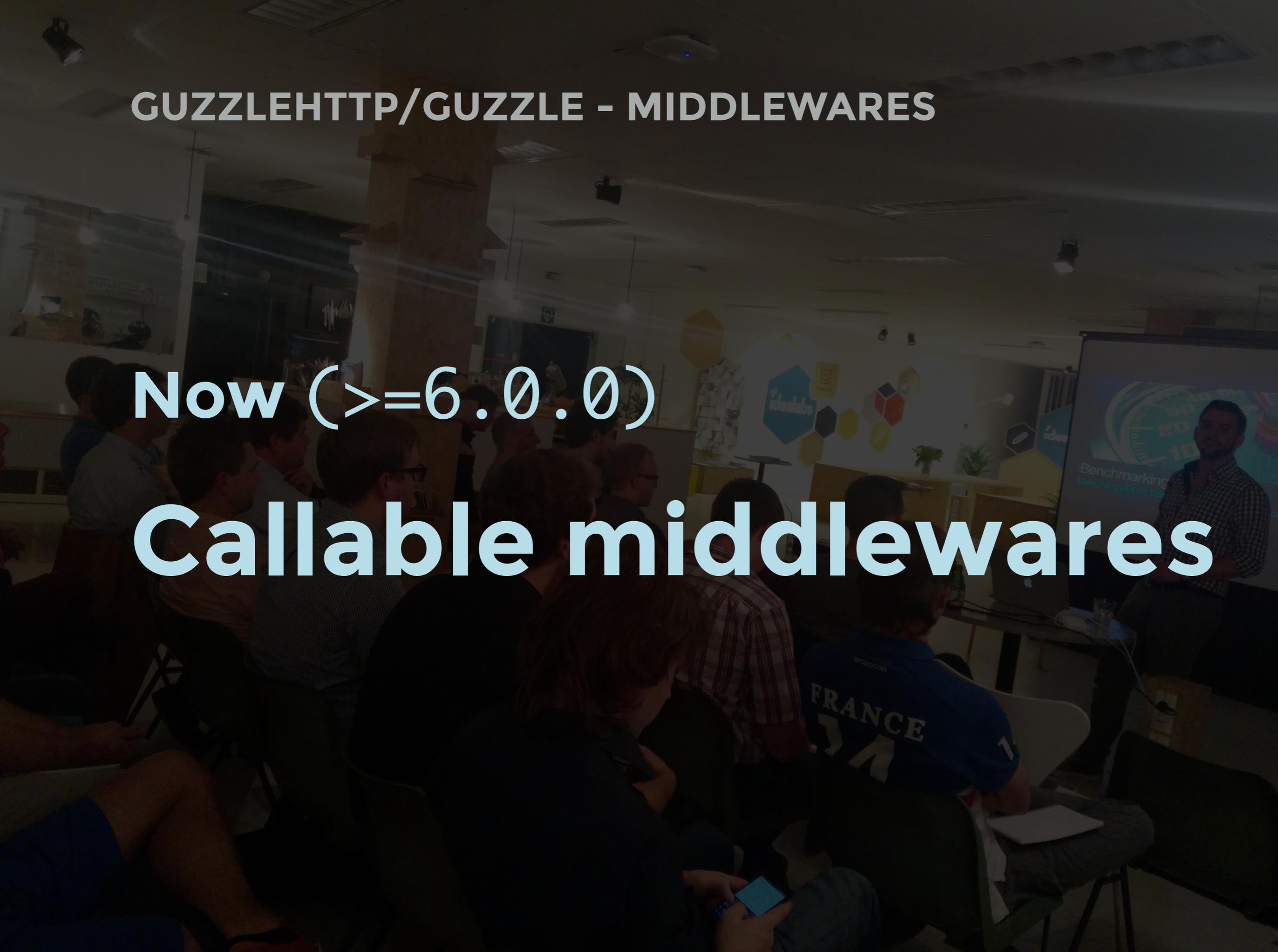
`GuzzleHttp\Event\EmitterInterface`

`GuzzleHttp\Event\SubscriberInterface`

# GUZZLEHTTP/GUZZLE - MIDDLEWARES

Now ( $\geq 6.0.0$ )

# Callable middlewares



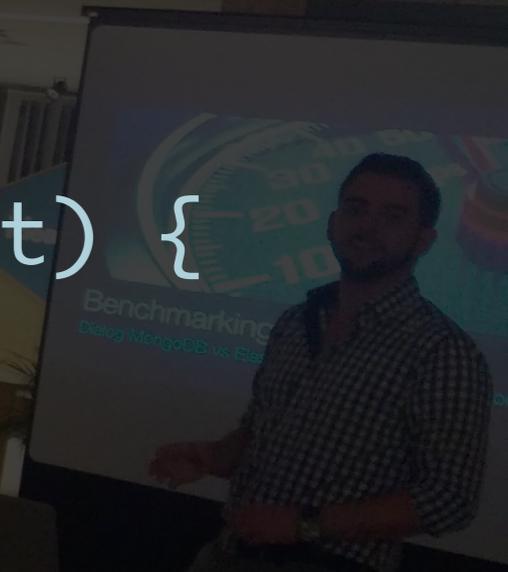
# GUZZLEHTTP/GUZZLE - MIDDLEWARES

```
$middleware = function ($handler) {  
    return function ($request, $options) use ($handler) {  
        // Alter request, if you want  
        return $handler($request, $options)  
        ->then(function (ResponseInterface $response) {  
            // Alter response, if you want  
            return $response;  
        });  
    };  
};  
}
```

# GUZZLEHTTP/GUZZLE - MIDDLEWARES

```
use GuzzleHttp\Middleware;
```

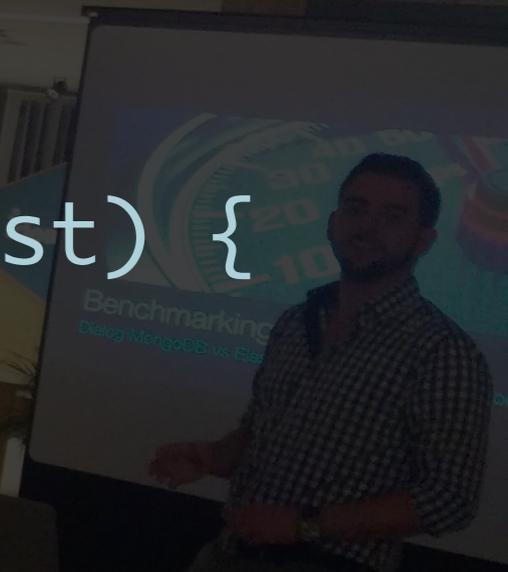
```
$middleware = Middleware::mapRequest(  
    function (RequestInterface $request) {  
        // Modify request  
        return $request;  
    }  
);
```



# GUZZLEHTTP/GUZZLE - MIDDLEWARES

```
use GuzzleHttp\Middleware;
```

```
$middleware = Middleware::mapResponse(  
    function (ResponseInterface $request) {  
        // Modify response  
        return $response;  
    }  
);
```



# GUZZLEHTTP/GUZZLE - MIDDLEWARES

```
$middleware = GuzzleHttp\Middleware::log(  
    $psr3Logger,  
    new GuzzleHttp\MessageFormatter()  
);
```



# GUZZLEHTTP/GUZZLE - MIDDLEWARES

```
$stack = HandlerStack::create();
```

```
$stack->push($middleware);
```

```
$stack->unshift($middleware);
```

```
$stack->remove($middleware);
```

```
$client = new Client(['handler' => $stack]);
```

# GUZZLEHTTP/GUZZLE - MIDDLEWARES

# Demo

<https://github.com/hannesvdvreken/psr7-demo/tree/master/4-middleware>

# GUZZLEHTTP/GUZZLE - RECAP

**Provides PSR-7 interfaced objects  
Middlewares instead of Eventing  
Async & Promises**

# **3. More PSR-7 usages**

# Decoupling HTTP Clients



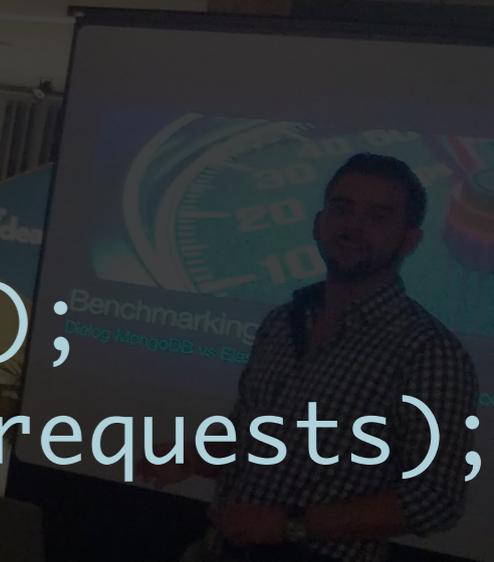
# PHP-HTTP/HTTPPLUG - SOLUTION

php-http/httpplug



# PHP-HTTP/HTTPPLUG - SOLUTION

```
interface HttpAdapter
{
    public function sendRequest($request);
    public function sendRequests(array $requests);
}
```



# PHP-HTTP/HTTPPLUG - SOLUTION

## Virtual package

```
"provide": {  
    "php-http/client-implementation": "^0.1.0",  
}
```



# PHP-HTTP/HTTPPLUG - ADAPTERS

**guzzle6-adapter**

**buzz-adapter**

**guzzle5-adapter**

**react-adapter**

**guzzle4-adapter**

**socket-client**

**guzzle3-adapter**

**fopen-adapter**

**zend2-adapter**

**file-get-contents-adapter**

**zend1-adapter**

**curl-client**

**cake-adapter**

**mock-client**

# PHP-HTTP/HTTPPLUG - COMPOSER.JSON

```
"require": {  
  "guzzlehttp/guzzle": "^5.0",  
  "your-awesome/sdk": "^1.0",  
  "php-http/guzzle5-adapter": "^0.1.0"  
}
```



# PHP-HTTP/HTTPPLUG - COMPOSER.JSON

```
"require": {  
-   "guzzlehttp/guzzle": "^5.0",  
+   "guzzlehttp/guzzle": "^6.0",  
   "your-awesome/sdk": "^1.0",  
-   "php-http/guzzle5-adapter": "^0.1.0"  
+   "php-http/guzzle6-adapter": "^0.1.0"  
}
```



START USING PSR-7 IN YOUR APP

What about **existing** Symfony  
Request/Response objects?



# START USING PSR-7 IN YOUR APP

[symfony/psr-http-message-bridge](https://symfony.com/doc/current/http_message.html)



# START USING PSR-7 IN YOUR APP

Symfony request/response object



PSR-7 request/response object

# START USING PSR-7 IN YOUR APP

Symfony request/response object



Zend request/response object

`\Psr\Http\Message\RequestInterface`

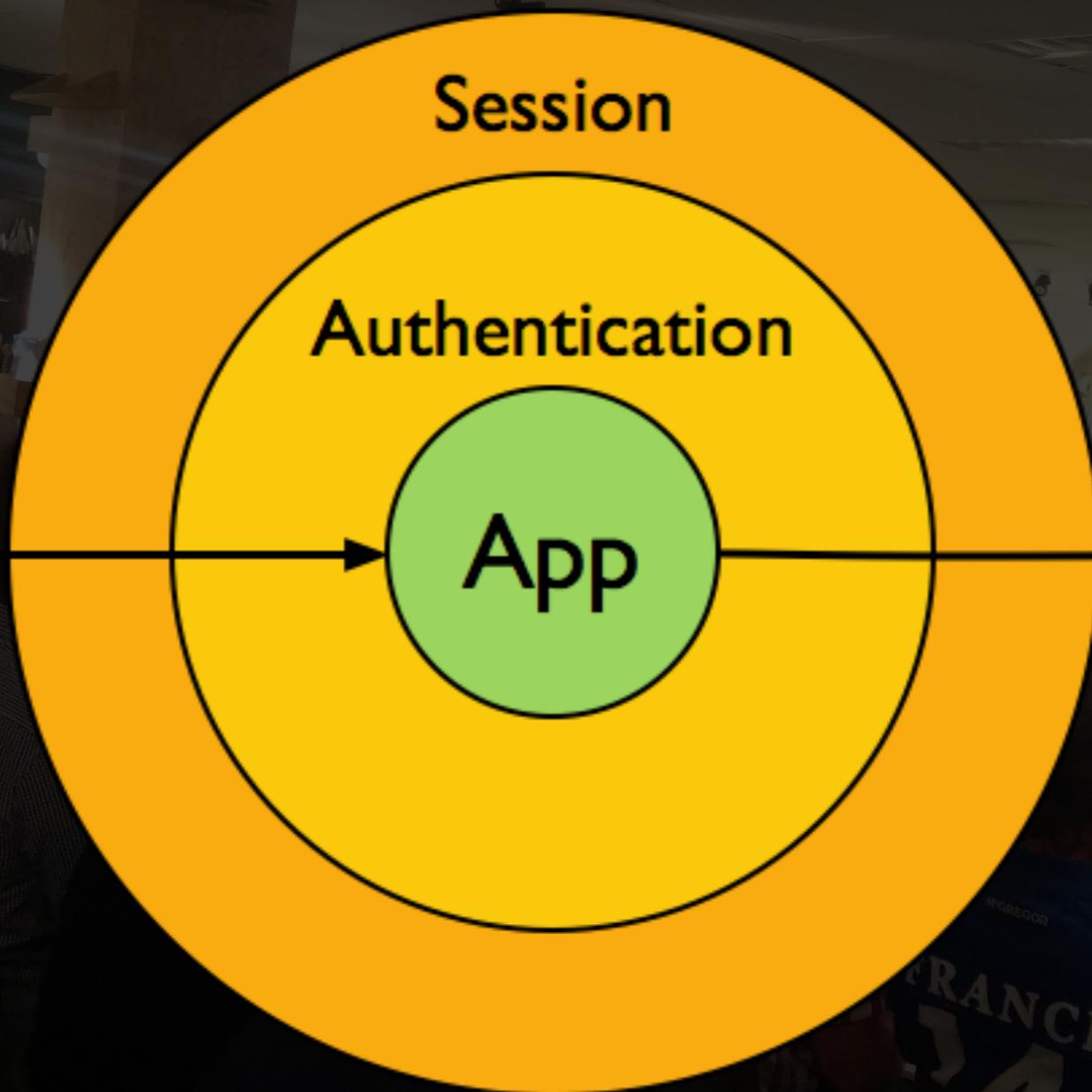
type hinted

# Middlewares

wrapping HTTP applications



**Request**



**Response**

# MIDDLEWARE DEFINITION

```
function (  
  RequestInterface $request,  
  ResponseInterface $response,  
  callable $next  
) {...};
```



## MIDDLEWARE DEFINITION

```
function ($request, $response, $next) {  
    // Alter request, if you want  
    $response = $next($request, $response);  
    // Alter response, if you want  
    return $response;  
}
```



# MIDDLEWARE DEFINITION

**Session**

**Logging**

**Authentication**

**Robots.txt file**

**HTTP Cache**

**Firewall**

**Cors**

**Throttling**

**...**

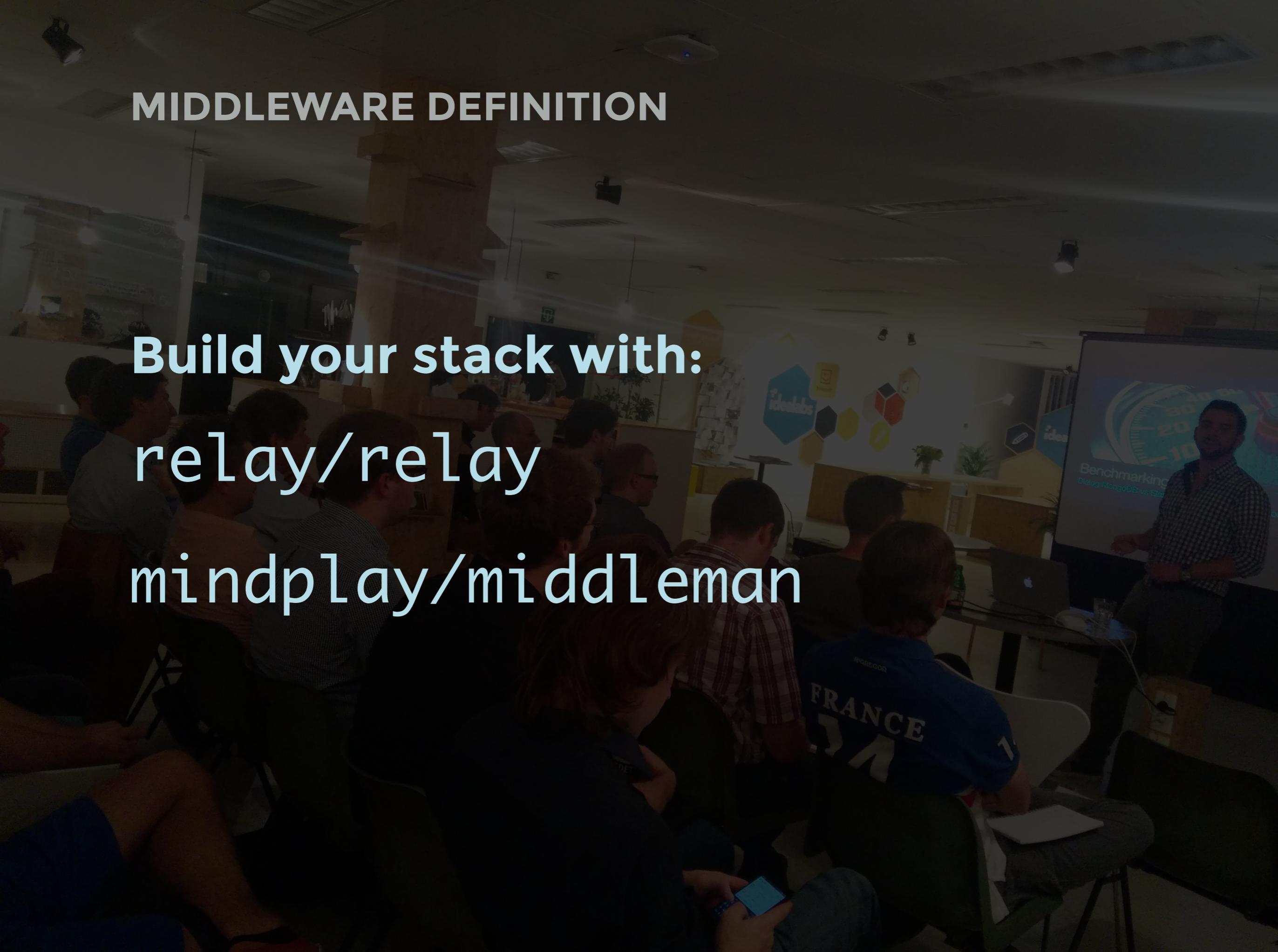


# MIDDLEWARE DEFINITION

**Build your stack with:**

**relay/relay**

**mindplay/middleman**



MIGRATION FROM STACKPHP

What about

**StackPHP** (Symfony HTTP Kernel)?

# MIGRATION FROM STACKPHP

[h4cc/stack-psr7-bridge](https://github.com/h4cc/stack-psr7-bridge)

**uses:**

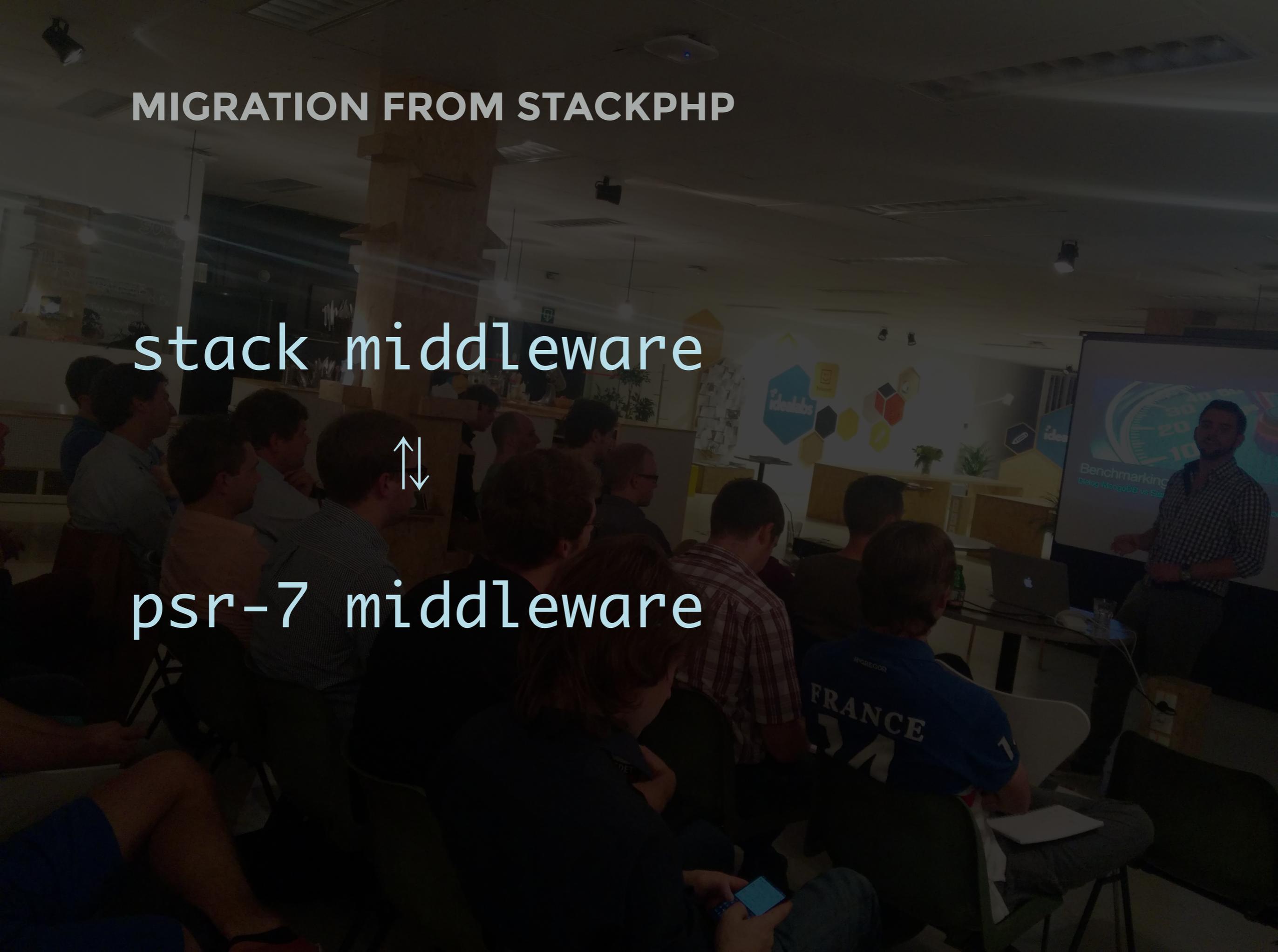
[symfony/psr-http-message-bridge](https://github.com/symfony/psr-http-message-bridge)

# MIGRATION FROM STACKPHP

stack middleware



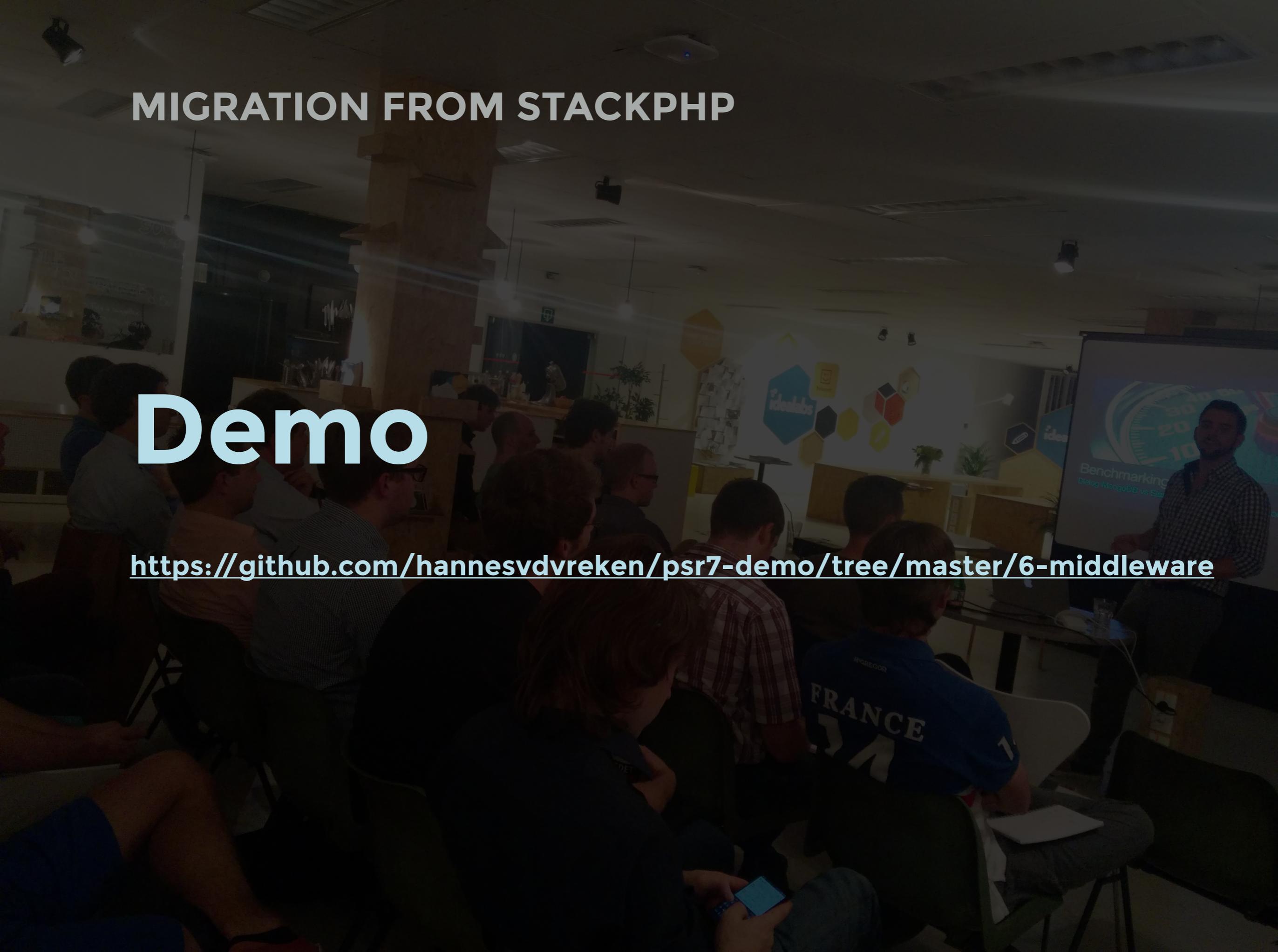
psr-7 middleware



# MIGRATION FROM STACKPHP

# Demo

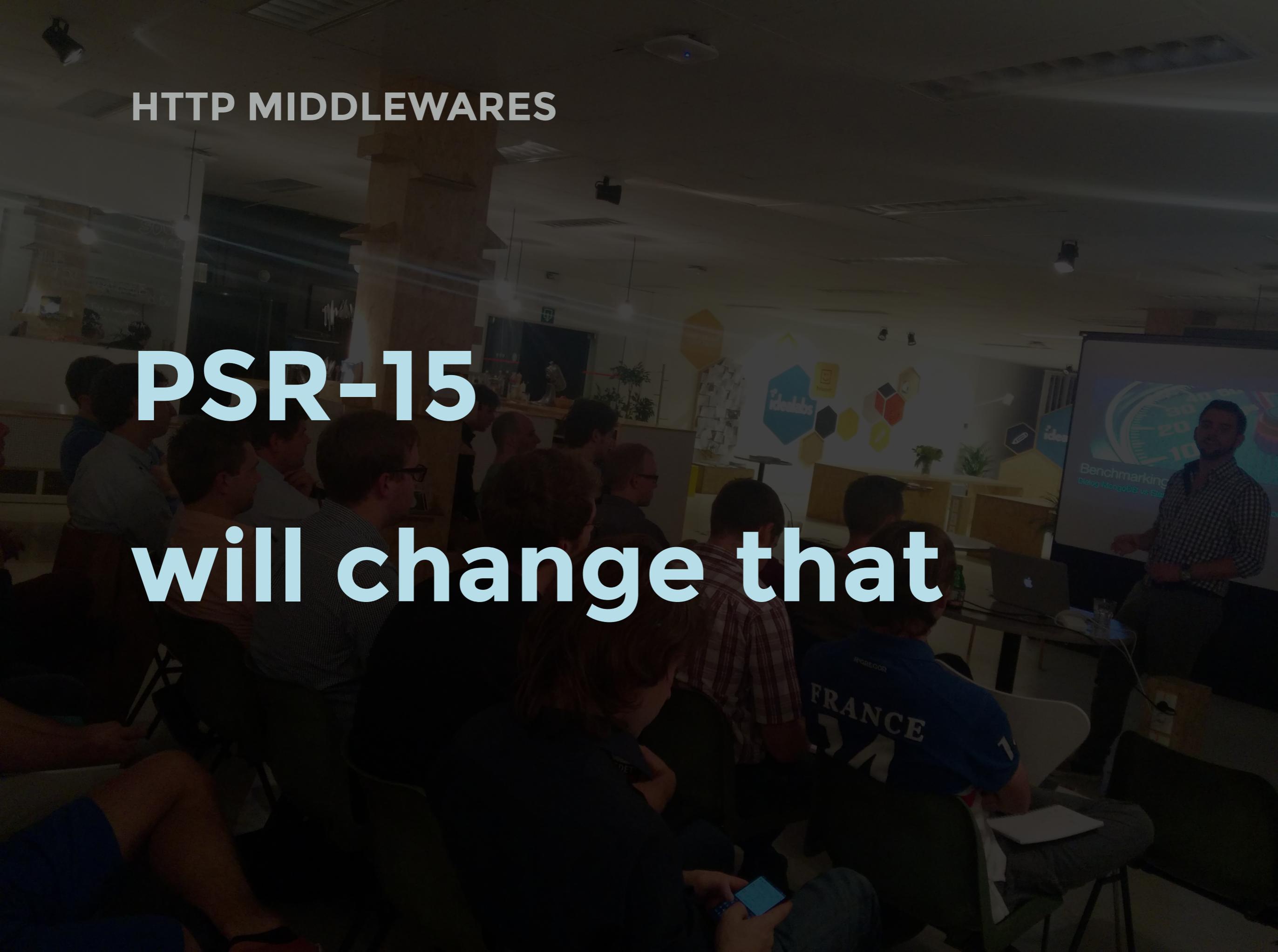
<https://github.com/hannesvdvreken/psr7-demo/tree/master/6-middleware>



HTTP MIDDLEWARES

PSR-15

will change that



HTTP MIDDLEWARES

PSR-15

is not accepted yet



## HTTP MIDDLEWARES - PSR-15

MiddlewareInterface

> ClientMiddlewareInterface

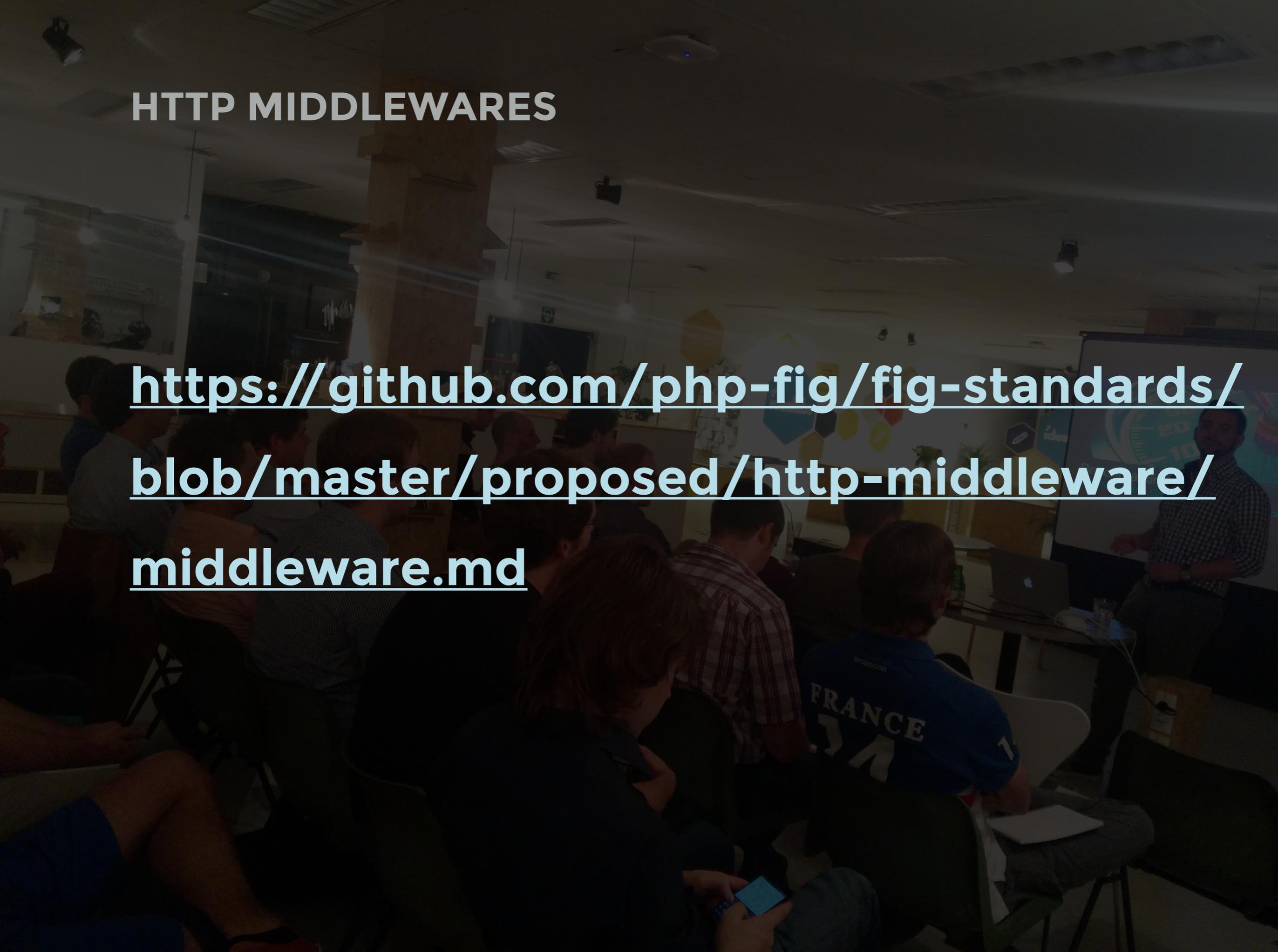
> ServerMiddlewareInterface

FrameInterface

StackInterface

# HTTP MIDDLEWARES

<https://github.com/php-fig/fig-standards/blob/master/proposed/http-middleware/middleware.md>



**RECAP**

**HTTP client decoupling  
Symfony objects to PSR-7 objects  
Application middlewares**



# RECAP

1. `psr/http-message`
2. `guzzlehttp/guzzle`
3. More PSR-7 usages



# Thank you!

<https://joind.in/talk/1786a>



@dpcon #dpc16

@hannesvdvreden

# Time for questions.

 @dpcon #dpc16

@hannesvdvreken

## REFERENCES

- <http://mwl.be>
- <https://github.com/guzzle/guzzle>
- <https://github.com/php-http>
- <http://docs.guzzlephp.org/en/latest/>
- <https://github.com/php-fig/http-message>
- <https://github.com/php-fig/fig-standards/blob/master/accepted/PSR-7-http-message.md>