

Lets talk about . . .



and the idea behind PSR-7

## Hi, i'm Julius Beckmann

and i work at



# Request/Response in PHP

```
<?php
// The whole request
// var_dump(
// $_GET, $_POST, $_SERVER,
// $_FILES, $_COOKIES
//);
// Creating a response
header("HTTP/1.1 404 Not Found");
echo "Not found";
exit;
```

#### PHP SAPI - Server Application Programming Interface

Gives us:

- Superglobals
  - ▶ \$\_GET
    - ▶ \$\_POST
    - ► \$\_SERVER

Primitives

- header()
- echo()
  - exit()
    - ...

#### PHP-SAPI Problems

- ► Global state
- ► Hard to inspect / debug
- ► Hard to reproduce / test

# Lets do it better!

### **HTTP**

A protocol is a interface.

#### The HttpKernelInterface

```
<?php
namespace Symfony\Component\HttpKernel;
interface HttpKernelInterface
    const MASTER_REQUEST = 1;
    const SUB_REQUEST = 2;
    /**
     * Handles a Request to convert it to a Response.
     * Oreturn Response
     */
    public function handle(
        Request $request,
        $type = self::MASTER_REQUEST,
        $catch = true
    );
```

#### Example HttpKernelInterface usage

```
<?php
class AppKernel implements HttpKernelInterface
    public function handle(Request $request) {
        return new Response('Hello World');
$kernel = new AppKernel();
$response = $kernel->handle(Request::createFromGlobals());
$response->send();
```

#### Why use HttpKernelInterface?

Request, AppKernel and Response are values.

- ► Inspectable
- ► Reuseable
- Composeable

#### PHP SAPI is **global state**.

Nothing of the above

#### Its nothing new . . .

- ▶ 1997: Java Servlet
- ▶ 2003: Python WSGI (Web Server Gateway Interface)
- ▶ 2007: Ruby Rack
- 2009: Perl PSGI/Plack
- ▶ 2011: Symfony HttpKernelInterface
- $\rightarrow$  Idea is more than 17 Years around, lets use it!

# Leverage

HttpKernelInterface

using StackPHP





Composing HttpKernelInterface middlewares since 2013!







#### Brought to you by



@igorwhiletrue



@beausimensen



@hochchristoph

StackPHP is ...

StackPHP is **NOT** a Framework
StackPHP is **NOT** a Library
StackPHP is a Toolset and Conventions

#### StackPHP Conventions

- 1. Implement the HttpKernelInterface
- 2. Take the decorated app as the first constructor argument
- 3. **Decorate** the handle call, **delegate** to the decorated app

**More** can be found here: *stackphp.com/specs*.

### AppKernel

# ightarrow Response

Request  $\rightarrow$ 

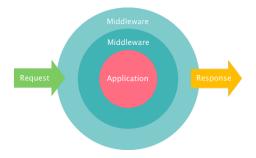
## Request ightarrow

ightarrow Response

Middleware(AppKernel)

#### **Middlewares**

Adding functionality by decorating Application.



#### Example Middleware

```
<?php
class ExampleMiddleware implements HttpKernelInterface
    private $app;
    public function __construct(HttpKernelInterface $app)
        $this->app = $app;
    public function handle(Request $request)
       // Before request
        $response = $this->app->handle($request, $type, $catch);
        // After response
       return $response;
```

How would you ....

restrict Access by IP?

#### With IpRestrict by Alsar

```
<?php
$kernel = new AppKernel();
$kernel = new Alsar\Stack\IpRestrict(
    $kernel.
    array('127.0.0.1', '192.168.0.1')
);
$response = $kernel->handle(Request::createFromGlobals());
$response->send();
```

https://github.com/alsar/stack-ip-restrict

#### Source of IpRestrict

```
<?php
class IpRestrict implements HttpKernelInterface
    private $app;
    private $allowedIps;
    public function __construct(HttpKernelInterface $app, array $allowedIps)
        $this->app = $app;
        $this->allowedIps = $allowedIps;
    public function handle(Request $request)
        $ip = $request->getClientIp();
        if (!in_array($ip, $this->allowedIps)) {
            return new Response(sprintf('IP %s is not allowed.', $ip), 403);
        return $this->app->handle($request, $type, $catch);
```

https://github.com/alsar/stack-ip-restrict

#### List of middlewares

#### stackphp.com/middlewares/

- HttpCache (by Symfony)
- CookieGuard (by Laravel)
- Geolp (by geocoder)
- ► IpRestrict (by alsar)
- Backstage (by atst)
- ▶ OAuth (by igorw)
- Basic Authentication (by dflydev)

- Hawk (by dflydev)
- ► CORS (by asm89)
- Robots (by dongilbert)
- Negotiation (by wildurand)
- Honeypot (by CHH)
- Turbolinks (by Helthe)
- Logger (by h4cc)
- **...**

#### Middleware usage

- Authentification
- Authorization
- ► Error/Exception Handling
- Assets
- Debug Toolbar
- Signed Cookies
- ► Force SSL

### Toolbox



#### Toolbox: Builder

Stacking Kernels can be confusing, the Builder can help here.

```
<?php

$stack = (new Stack\Builder())
    ->push('Stack\Session')
    ->push('SymfonyHttpCache', new Store('../cache'));

$app = $stack->resolve($app);
```

Result: Session(SymfonyHttpCache(\$app, '../cache'))

#### Toolbox: Run

Stack\run() is just a shortcut:

```
<?php
namespace Stack;
use Symfony\Component\HttpKernel\HttpKernelInterface;
use Symfony\Component\HttpKernel\TerminableInterface;
use Symfony\Component\HttpFoundation\Request;
function run(HttpKernelInterface $app, Request $request = null)
    $request = $request ?: Request::createFromGlobals();
    $response = $app->handle($request);
    $response->send();
    if ($app instanceof TerminableInterface) {
        $app->terminate($request, $response);
```

#### Toolbox: Session

Injecting a Session to the Request

```
<?php
$app = new Silex\Application();
$app->get('/', function (Request $request) {
    $session = $request->getSession();
    // use session ...
});
$stack = (new Stack\Builder())->push('Stack\Session');
Stack\run(\$stack->resolve(\$app));
```

#### Toolbox: UrlMap

Mapping URL-Prefixes to Apps.

```
<?php

$map = array(
    '/api' => new ApiKernel(),
    '/blog' => new BlogKernel(),
    '/' => new WebsiteKernel(),
);

$app = (new Stack\Builder())->push('Stack\UrlMap', $map)

Stack\run($stack->resolve($app));
```

#### Toolbox: LazyHttpKernel

Lazy proxy for HttpKernelInterfaces.

```
<?php
$map = array(
    '/api' => Stack\lazy(function () {
        return new ApiKernel();
    '/blog' => Stack\lazy(function () {
        return new BlogKernel();
    '/' => Stack\lazy(function () {
        return new WebsiteKernel();
   }),
```

# Are there

Stackable Apps?

#### List of Apps using HttpKernelInterface

#### symfony.com/components/HttpKernel

- Symfony
- Silex
- Laravel
- ► Proem Framework
- Pagekit
- phpBB

- Drupal
- Bolt
- ► Thelia
- Shopware
- Sculpin
- **•** ..

And PSR-7?

# PSR-7 MessageInterface DRAFT

#### Base for Request and Response

```
<?php
namespace Psr\Http\Message;
interface MessageInterface
    public function getProtocolVersion();
    public function getBody():
    public function setBody(StreamInterface $body = null);
    public function getHeaders();
    public function hasHeader($header);
    public function getHeader($header);
    public function getHeaderAsArray($header);
    public function setHeader($header, $value);
    public function setHeaders(array $headers);
    public function addHeader($header, $value);
    public function addHeaders(array $headers);
    public function removeHeader($header);
```

## PSR-7 RequestInterface DRAFT

```
<!php

namespace Psr\Http\Message;
interface RequestInterface extends MessageInterface
{
    public function getMethod();
    public function setMethod($method);

    public function getUrl();
    public function setUrl($url);
}</pre>
```

# PSR-7 ResponseInterface DRAFT

```
<?php

namespace Psr\Http\Message;
interface ResponseInterface extends MessageInterface
{
   public function getStatusCode();
   public function setStatusCode($code);

   public function getReasonPhrase();
   public function setReasonPhrase($phrase);
}</pre>
```

### Summary

- ▶ Using **Http** as a **interface** with HttpKernelInterface.
- Add functionality through Middlewares.
- Build framework-agnostic features based on HTTP.

When writing what?

# Library? Bundle/Module/ServiceProvider? Middleware?

# Request/Response logger

Packagist: silpion/stack-logger

# **REST Fileserver**

Packagist: h4cc/stack-flysystem

# ReactPHP Adapter

Packagist: yosymfony/httpserver

# Modularized API

#### About me

#### Julius Beckmann

(twitter|github).com/h4cc

- ► PHP
  - Symfony
  - Silex
- BEAM
  - Erlang
  - ► Elixir

- DevOps
  - Server
  - Deployment
- Pandoc
  - Markdown
  - PATEX

Questions?

# KThxBye! :)