

PSR-7, middlewares e o futuro dos frameworks

Quem?

Elton Minetto

Developer since 1997

Teacher since 2004

Co-founder at Coderockr since 2010

Zend Framework Evangelist

Author of <http://bit.ly/zf2napratica> and <http://bit.ly/>

[doctrinenapratica](http://bit.ly/doctrinenapratica)

[@eminetto](http://bit.ly/@eminetto)

PSR-7

"Http Is The Foundation
Of The Web"

Um cliente manda uma request

```
1 $method = $request->getMethod( );  
2 $accept = $request->getHeader( 'Accept' );  
3 $path = $request->getUri( )->getPath( );  
4 $controller = $request->getAttribute( 'controller' );
```

O servidor retorna uma response

```
1 $response->getBody( )->write( 'Hello world!' );  
2 $response = $response  
3   ->withStatus( 200, 'OK' )  
4   ->withHeader( 'Content-Type', 'text/plain' );
```

Middlewares

"Between the request and response"

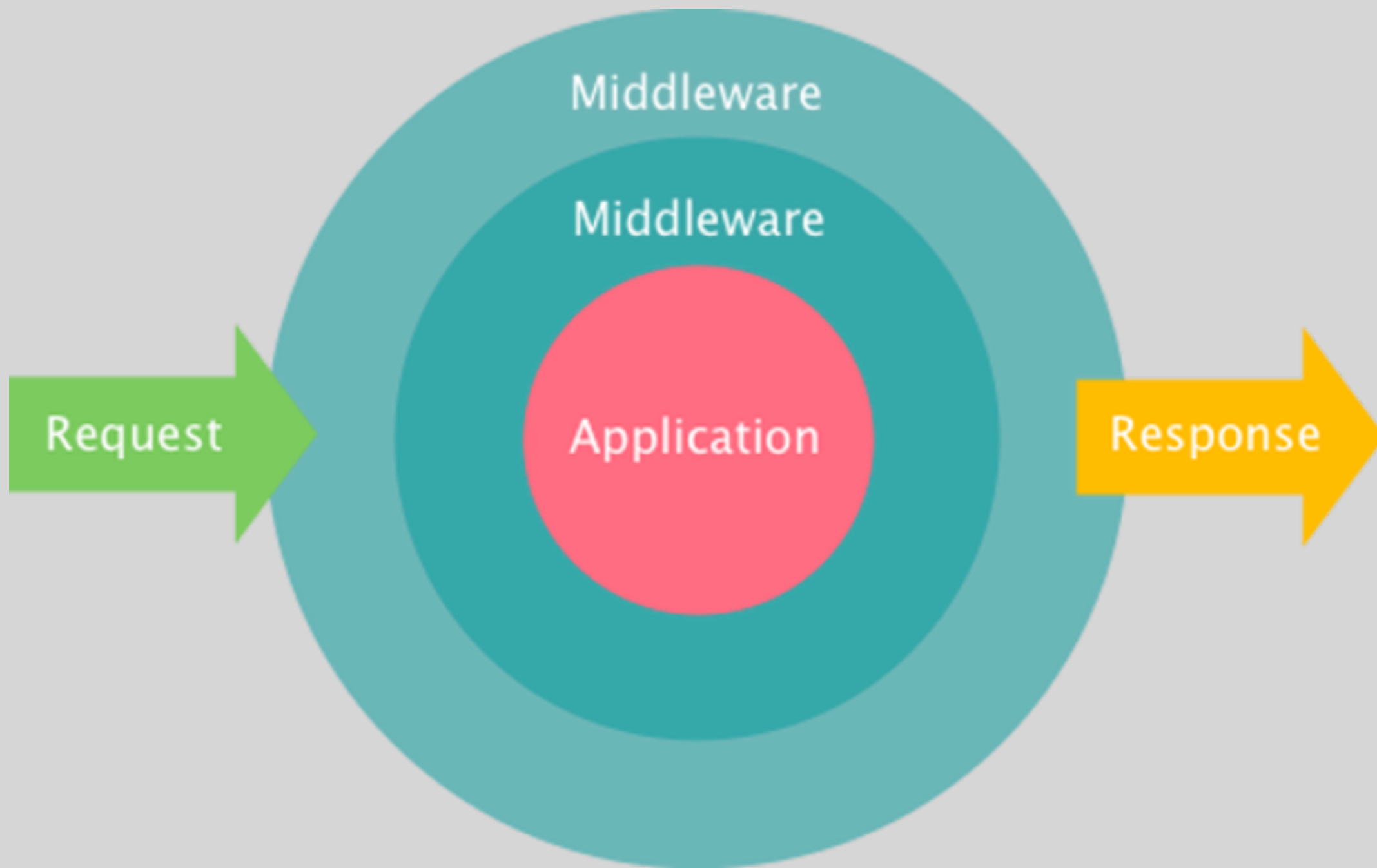
Enrico Zimuel

"HTTP middleware is not for your Domain work.
The middleware is a path in to, and out of, the
core Domain."

Paul M. Jones

"In other words, we'll no longer write Zend Framework or Symfony or Laravel or framework-flavor-of-the-day applications or modules/bundles/packages/what-have-you. We'll write middleware that solves a discrete problem, potentially using other third-party libraries, and then compose them into our applications — whether those are integrated into a framework or otherwise."

Matthew O'phinney



Onion Style

```
1 class Auth implements MiddlewareInterface
2 {
3     public function __invoke(Request $request, Response $response, callable
4         $out = null)
5     {
6         if(! $request->hasHeader('authorization')){
7             return $response->withStatus(401);
8         }
9
10        if (!$this->isValid($request)) {
11            return $response->withStatus(403);
12        }
13
14        return $out($request, $response);
15    }
16 }
```

Stacks or Pipelines

```
1 $api = AppFactory::create();
2 $api->get('/', function ($request, $response, $next) {
3     $response->getBody()->write('Hello, beers of world!');
4     return $response;
5 });
6
7 $app = AppFactory::create();
8 $app->pipe(new Auth());
9 $app->pipe($api);
10 $app->pipe(new Format());
11
12 $app->run();
```

Lambda Style

```
1 $middleware = function (Request $request)
2 {
3     // do some work
4     return $response;
5 };
6 $response = $middleware($request);
```

Exemplo

RestBeer

API about 

Multiple response formats (JSON, HTML)

<http://restbeer.com/beer>

<http://restbeer.com/style>

<http://restbeer.com/beer/Guinness>

<http://restbeer.com/beer/Heineken>

<http://restbeer.com/style/Pilsen>

<http://restbeer.com/style/Stout>

Expressive

"Expressive allows you to write PSR-7 middleware applications for the web. It is a simple micro-framework built on top of Stratigility, providing:

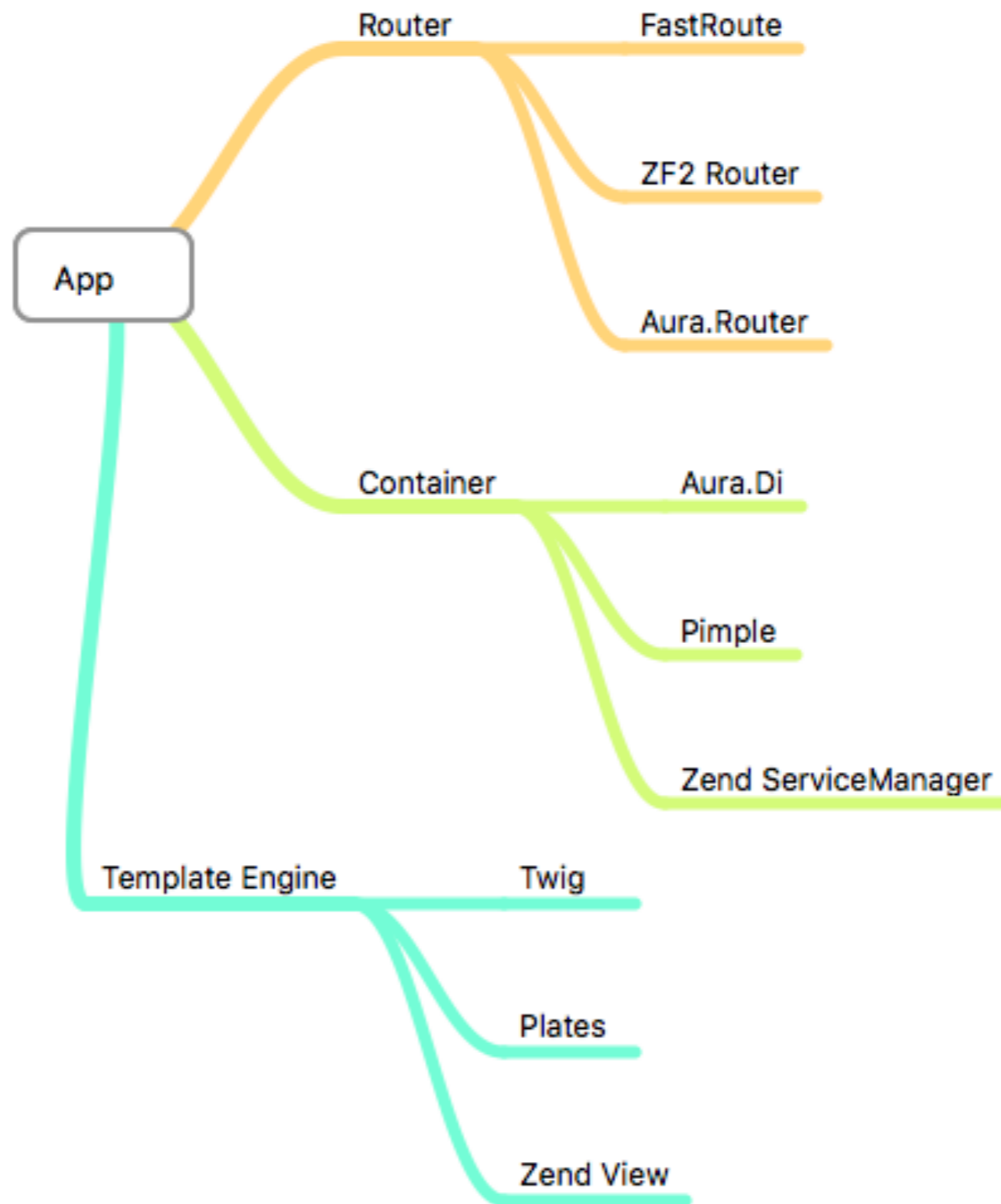
Dynamic routing

Dependency injection via container-interop

Templating

Error Handling"

<http://devzone.zend.com/6615/announcing-expressive/>



Installing


composer.json

```
1 {  
2     "require": {  
3         "zendframework/zend-expressive": "^0.5",  
4         "zendframework/zend-expressive-fastroute": "^0.1",  
5         "zendframework/zend-expressive-twigrenderer": "^0.1",  
6         "zendframework/zend-servicemanager": "^2.6"  
7     }  
8 }
```


Installing


```
curl -s http://getcomposer.org/installer | php  
php composer.phar install
```


FOLDERS


 **PalestraMiddlewares**


 **src**


 **RestBeer**


 Auth.php


 Format.php


 **vendor**

 **views**

 content.twig

 {} composer.json

 * composer.lock

 index.php

index.php

<https://gist.github.com/eminetto/558ab7c0a10ff48d6652>

src\RestBeer\Auth.php

<https://gist.github.com/eminetto/d2eea1f559a02a10549a>

src\RestBeer\Format.php

<https://gist.github.com/eminetto/47b66d47715468da73fe>

views\content.twig

<https://gist.github.com/eminetto/89e3cbc4631aabf3cb47>

Testing

```
php -S localhost:8080
```

Outros middlewares

<https://packagist.org/search/?q=middleware>

Estrutura mais
"profissional"

```
composer create-project -s rc zendframework/zend-expressive-  
skeleton expressive
```

Installing zendframework/zend-expressive-skeleton (0.4.0)

- Installing zendframework/zend-expressive-skeleton (0.4.0)

Downloading: 100%

Created project in /Users/samsonasik/www/zend-expressive-skeleton

> App\Composer\OptionalPackages::install

Setup data and cache dir

Setting up optional packages

Which router you want to use?

[1] Aura.Router

[2] FastRoute

[3] Zend Router

Make your selection or type a composer package name and version (Aura.Router): 3

- Adding package zendframework/zend-mvc (^2.5)

- Adding package zendframework/zend-psr7bridge (^0.1.0)

- Copying /config/autoload/routes.global.php

Which container you want to use for dependency injection?

[1] Aura.Di

[2] Pimple-interop

[3] Zend ServiceManager

Make your selection or type a composer package name and version (Zend ServiceManager): 3

- Adding package zendframework/zend-servicemanager (^2.5)

- Adding package ocradius/proxy-manager (^1.0)

- Copying /config/container.php

Which template engine you want to use?

[1] Plates

[2] Twig

[3] Zend View installs Zend ServiceManager

[n] None of the above

Make your selection or type a composer package name and version (n): 3

- Adding package zendframework/zend-view (^2.5)

- Adding package zendframework/zend-filter (^2.5)

- Adding package zendframework/zend-i18n (^2.5)

- Adding package zendframework/zend-servicemanager (^2.5)

- Copying /config/autoload/templates.global.php

- Copying /templates/error/404.phtml

- Copying /templates/error/error.phtml

- Copying /templates/layout/default.phtml

- Copying /templates/app/home-page.phtml

Which error handler do you want to use during development?

[1] Whoops

[n] None of the above

Make your selection or type a composer package name and version (Whoops): 1

- Adding package filp/whoops (^1.1)

FOLDERS

expressive

config

autoload

config.php

container.php

data

public

src

Action

HomePageAction.php

HomePageFactory.php

PingAction.php

templates

app

error

layout

* .gitkeep

test

vendor

* .gitignore

Y .travis.yml

CHANGELOG.md

{ } composer.json

* composer.lock

CONTRIBUTING.md

LICENSE.md

phpcs.xml

* phpunit.xml.dist

README.md

O futuro?

"We see ZF3 as a movement: **an end to framework silos**, by providing quality, commodity code that can be used everywhere and anywhere. An **end** to saying "I'm a ZF developer," or "I'm a Laravel developer", and a return to, **"I'm a PHP developer."**"

Matthew O'phinney

Links

<https://github.com/zendframework/zend-diactoros/blob/master/doc/book/custom-responses.md>

<https://github.com/zendframework/zend-stratigility>

<https://mwop.net/blog/2015-01-08-on-http-middlewares-and-psr-7.html>

<https://beau.io/talks/2015/05/16/hello-psr-7-phpday-2015/>

<http://paul-m-jones.com/archives/6125>

<https://mwop.net/blog/2015-01-26-psr-7-by-example.html>

<http://srcmvn.com/blog/2015/02/18/psr-7-and-the-future-of-php/>

<http://www.sitepoint.com/stackphp-explained/>

<http://framework.zend.com/blog/zend-framework-3-update-and-roadmap.html>

Contato

<http://eltonminetto.net>

@eminetto

eminetto@coderochr.com

<http://asemanaphp.com.br>

<http://coderochr.com>