

# Creating an API with Expressive

Who?

# Elton Minetto

Developer since 1997

Teacher since 2004

Co-founder at Coderockr since 2010

Zend Framework Evangelist

Author of <http://bit.ly/zf2napratica> and [@eminetto](http://bit.ly/doctrinenapratica)

API

What?

"[] a set of routines, protocols, and tools for building software applications. An API expresses a software component in terms of its operations, inputs, outputs, and underlying types []"

Wikipedia

Why?

Multiple interfaces  
(web, mobile, CLI)



Integration with internal services  
Software/Infrastructure as a Service  
(Amazon, Parse, Pusher, Filepicker, etc)

Example

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

```
{  
    "require": {  
        "zendframework/zend-expressive": "^0.1.0",  
        "aura/router": "^2.3",  
        "zendframework/zend-servicemanager": "^2.6"  
    }  
}
```

## Installing

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



Show 🍺🍺

```
<?php
use Aura\Router\RouterFactory;
use Zend\Expressive\AppFactory;
use Zend\Expressive\Router\Aura as AuraBridge;

require 'vendor/autoload.php';

$auraRouter = (new RouterFactory())->newInstance();
$router = new AuraBridge($auraRouter);
$app = AppFactory::create(null, $router);

$beers = array(
    'brands' => array('Heineken', 'Guinness', 'Skol', 'Colorado'),
    'styles' => array('Pilsen' , 'Stout')
);

$app->get('/', function ($request, $response, $next) {
    $response->getBody()->write('Hello, beers of world!');
    return $response;
});

$app->get('/brand', function ($request, $response, $next) use ($beers) {
    $response->getBody()->write(implode(',', $beers['brands']));
    return $response;
});

$app->get('/style', function ($request, $response, $next) use ($beers) {
    $response->getBody()->write(implode(',', $beers['styles']));
    return $response;
});

$app->run();
```

# Testing

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

Show a 

```
$app->get('/beer{/id}', function ($request, $response,
$next) use ($beers) {
    $id = $request->getAttribute('id');
    if ($id == null) {
        $response->getBody()->write(
            implode(',', $beers['brands'])
        );
        return $response;
    }

    $key = array_search($id, $beers['brands']);
    if ($key === false) {
        $response->getBody()->write('Not found');
        return $response;
    }

    $response->getBody()->write($beers['brands'][$key]);

    return $response;
});
```

JSON!

<https://gist.github.com/eminetto/5477c1da39f36d061fee>



# Authentication

index.php

<https://gist.github.com/eminetto/197e48af07ad75db0fc2>

src\RestBeer\Auth.php

<https://gist.github.com/eminetto/26a2e40f63fb4f1c7546>

Add a 🍺

<https://gist.github.com/eminetto/f9e44c18314b7aaaf66b>

# HTML and JSON!

```
{  
  "require": {  
    "zendframework/zend-expressive": "^0.1.0",  
    "aura/router": "^2.3",  
    "zendframework/zend-servicemanager": "^2.6",  
    "twig/twig": "*"  
  }  
}
```

```
<!-- views/content.twig -->
{% for c in content %}
    {{c}}<br>
{% endfor %}
```

index.php

<https://gist.github.com/eminetto/4509a6cc5defe2b49d5a>

src\RestBeer\Format.php

<https://gist.github.com/eminetto/88114420d9c2b57708f6>



# Links

<http://zend-expressive.readthedocs.org/en/stable/>  
[https://github.com/zendframework/zend-](https://github.com/zendframework/zend-stratigility)  
[stratigility](https://github.com/zendframework/zend-stratigility)

Contact

<http://eltonminetto.net>

[@eminetto](mailto:@eminetto)

[eminetto@coderochr.com](mailto:eminetto@coderochr.com)

<http://asemanaphp.com.br>

<http://coderochr.com>