

Criando aplicações

RESTful com Zend

Framework 2

Quem?

Elton Minetto

Graduado/pós-graduado em Ciência da Computação

Desenvolvedor desde 1997

Professor desde 2004

Autor de três livros sobre PHP e um sobre grid computing

Podcaster no **Biz and Code**

CTO da **Coderockr** e do **Code Squad**

Membro da diretoria da **ABRAPH**

Princípios REST

**Dê a todas as coisas um
identificador**

<http://zf2.dev/api/v1/album.album.json>

<http://zf2.dev/api/v1/album.album.json/1>

<http://example.com/customers/1234>

<http://www.infoq.com/br/articles/rest-introduction>

Vincule as coisas

```
<order self="http://x.com/orders/1234">  
  <amount>23</amount>  
  <product ref="http://x.com/products/4554"></product>  
  <customer ref="http://x.com/customers/1234"></customer>  
</order>
```


Utilize os métodos padrão

GET, PUT, POST, DELETE, HEAD, OPTIONS

Recursos com múltiplas representações

<http://zf2.dev/api/v1/album.album.json>

<http://zf2.dev/api/v1/album.album.xml>

ou usar o header **Accept**

Comunique sem estado

“[...]um servidor **não deveria guardar o estado** da comunicação de qualquer um dos clientes que se comunique com ele **além de uma única requisição**. A razão mais óbvia para isso é **escalabilidade** - o número de clientes que podem interagir com o servidor seria consideravelmente impactado se fosse preciso manter o estado do cliente”

Zend Framework 2

Desenvolvido pela **Zend** junto com a
comunidade PHP

Mais de **2 anos** de desenvolvimento

Mais de **200 profissionais** envolvidos

Apoio **Google, Microsoft**, entre outras

Mão na massa!

Composer

```
mkdir rest_zf2  
cd rest_zf2  
curl -sS https://getcomposer.org/installer | php
```

ZFTool

```
php composer.phar require zendframework/zftool:dev-  
master
```

Criando o projeto

```
php ./vendor/bin/zf.php create project ~/Documents/  
Projects/vagrant/rest_zf2/ToDo
```

```
cd ToDo
```

```
php composer.phar self-update
```

```
php composer.phar install
```

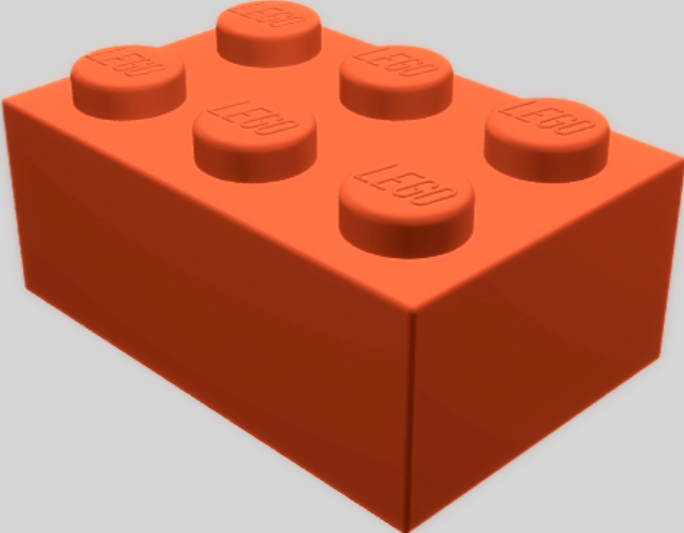
Apache VHost


```
<VirtualHost *:80>
    ServerName todo.dev
    DocumentRoot /vagrant/rest_zf2/ToDo/public
    SetEnv APPLICATION_ENV "development"
    SetEnv PROJECT_ROOT "/vagrant/rest_zf2/ToDo"
    <Directory /vagrant/rest_zf2/ToDo/public>
        DirectoryIndex index.php
        AllowOverride All
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

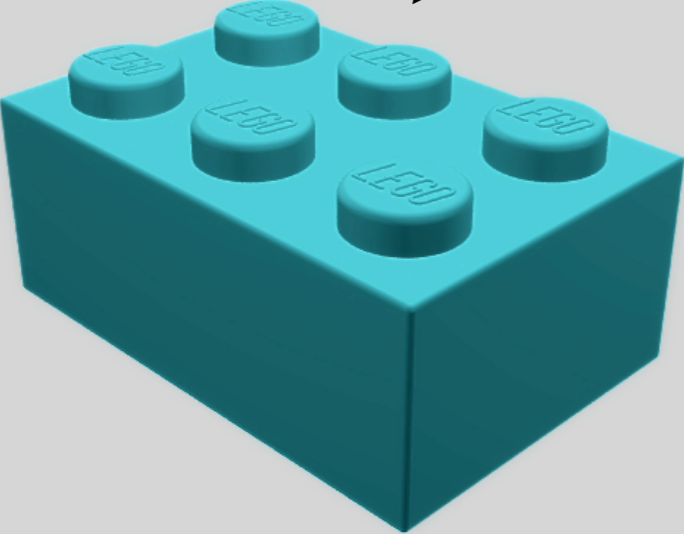
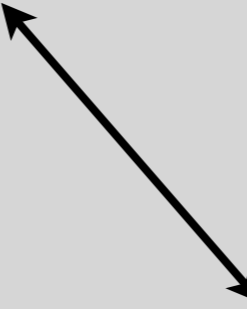
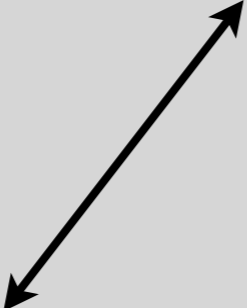
Módulos

"[] um módulo é uma **coleção de código** e outros arquivos que resolve um **problema específico** de uma aplicação ou site."

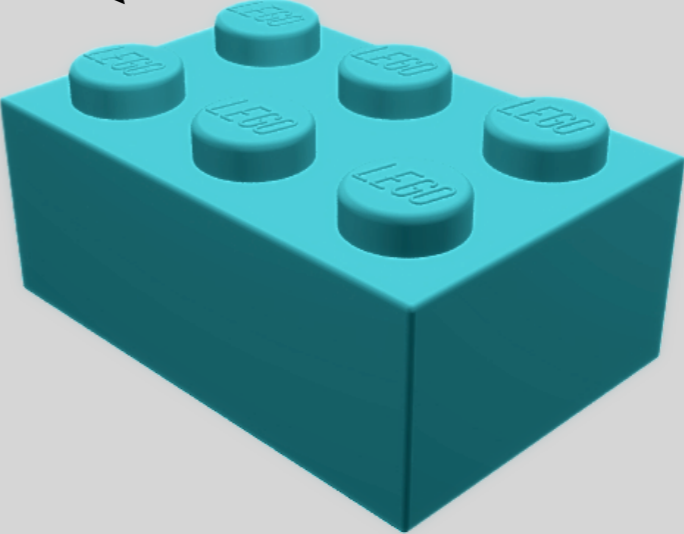
Matthew O'Phinney. Zend Framework project leader



Módulo Api



Módulo Admin



Módulo Application

Criando o módulo Api

```
cd ../  
php ./vendor/bin/zf.php create module Api ~/Documents/Projects/vagrant/rest_zf2/TODO/
```

Banco de dados

global.php

<https://gist.github.com/eminetto/5709289>

```
sqlite3 data/todo.db
```

```
create table item (id INTEGER PRIMARY KEY  
AUTOINCREMENT, description text, done int);
```


Entidades e TableGateway

src/Application/Model/Item.php

<https://gist.github.com/eminetto/5709294>

src/Application/Model/ItemTable.php

<https://gist.github.com/eminetto/5709297>

Application/Module.php

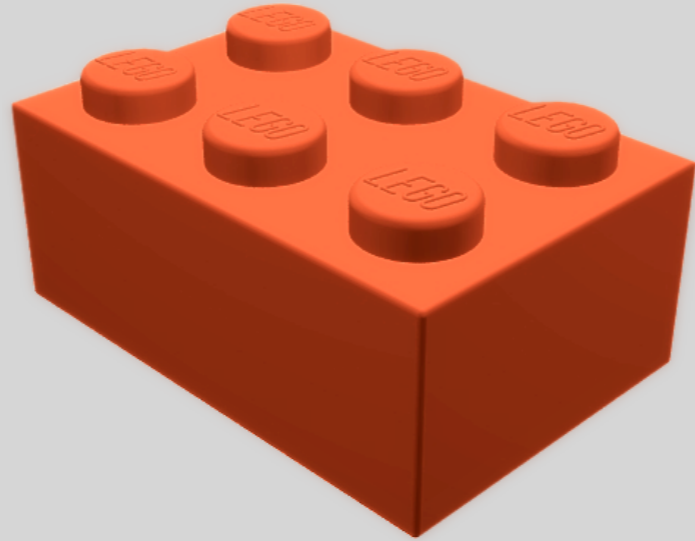
<https://gist.github.com/eminetto/5710850>

Rota

Api/config/module.config.php

<https://gist.github.com/eminetto/5709324>

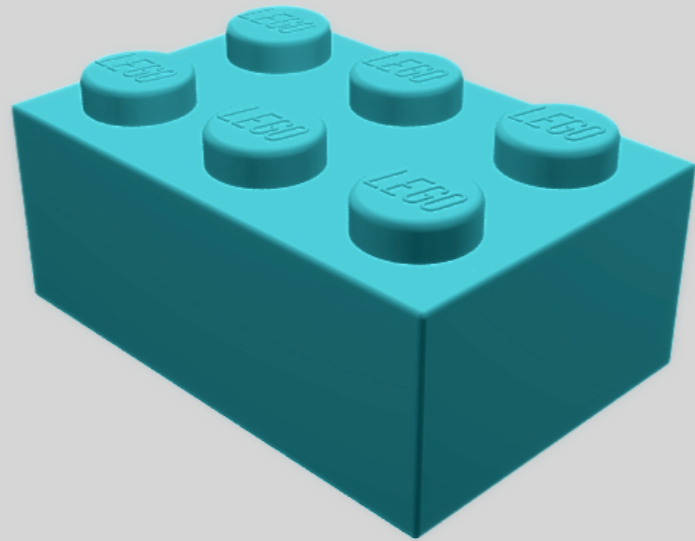
RestController



Módulo Api



Api\Controller\RestController



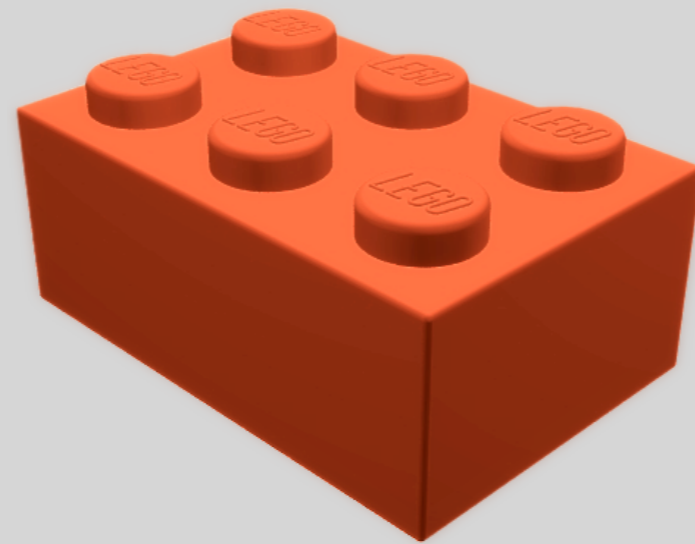
Módulo Application

Api/src/Controller/RestController

<https://gist.github.com/eminetto/5710862>

Eventos

Um **EventManager** é um objeto que agrega **listeners** para um ou mais **eventos** e que gera (**trigger**) outros eventos. Um **listener** é uma função/método (**callback**) que **reage** a um evento



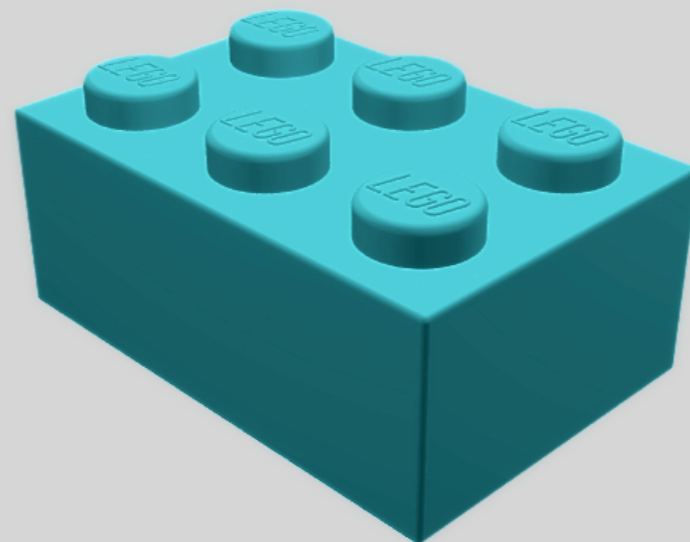
Módulo Api



Api\PreProcessor\PreProcessor
(segurança)

Api\Controller\RestController

Api\PostProcessor\PostProcessor
(serialização)



Módulo Application

Criando os eventos

Alterar o api/Module.php

<https://gist.github.com/eminetto/5710879>

PreProcessor:

<https://gist.github.com/eminetto/5710894>

Application/config/entities.config.php (Segurança)

<https://gist.github.com/eminetto/5710859>

PostProcessor:

<https://gist.github.com/eminetto/5710898>

Serializando

Múltiplos formatos

```
php composer.phar require jms/serializer:*
```

```
src/Api/PostProcessor/AbstractPostProcessor.php
```

```
https://gist.github.com/eminetto/5710903
```

```
src/Api/PostProcessor/Json.php
```

```
https://gist.github.com/eminetto/5710908
```

```
src/Api/PostProcessor/Xml.php
```

```
https://gist.github.com/eminetto/5710915
```

Referências

<http://www.infoq.com/br/articles/rest-introduction>

<http://www.slideshare.net/Alganet/rest-key>

<http://framework.zend.com/manual/2.2/en/user-guide/overview.html>

<http://jmsyst.com/libs/serializer>

<https://github.com/Coderockr/ApiModule>

@eminetto

<http://www.eltonminetto.net>

<http://coderochr.com>

<http://code-squad.com>

eminetto@coderochr.com