

# Laravel 5.7 desde cero (Aulas 1 a 15)

## Aprendible

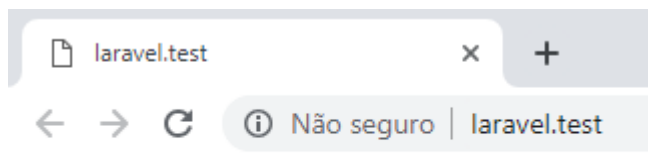
[https://www.youtube.com/watch?v=QRwuPHeagaQ&list=PLpKWS6gp0jd\\_uZiWmjucLY7LAMaD8UJhc](https://www.youtube.com/watch?v=QRwuPHeagaQ&list=PLpKWS6gp0jd_uZiWmjucLY7LAMaD8UJhc)

Resumo do curso feito por Roberto Pinheiro

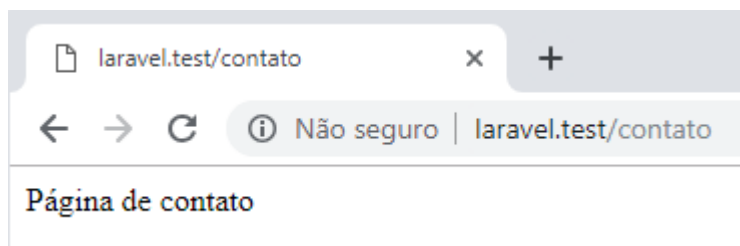
## Aula 04 - Rotas

### routes/web.php

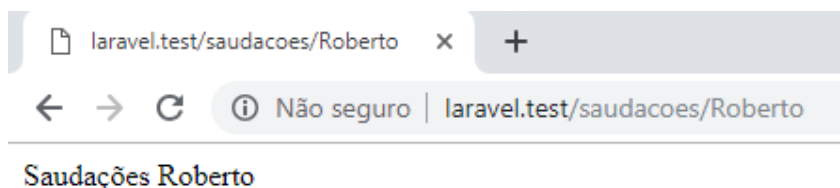
```
Route::get('/', function () {  
    return 'Página inicial';  
});
```



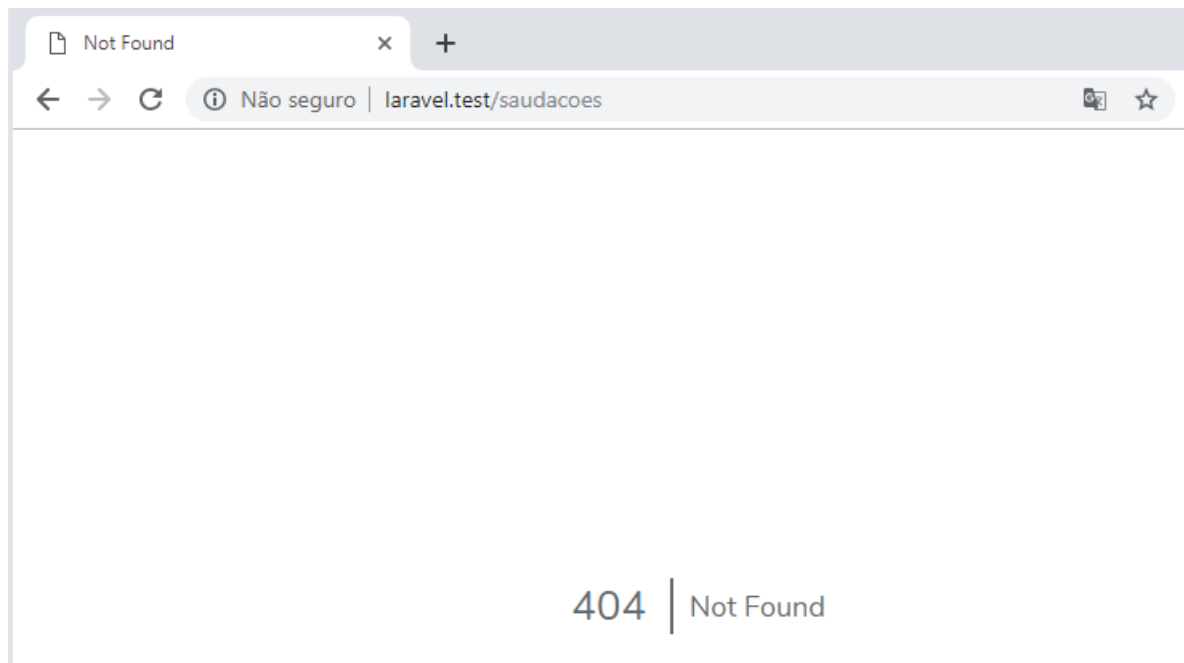
```
Route::get('/contato', function () {  
    return 'Página de contato';  
});
```



```
Route::get('/saudacoes/{nome}', function ($nome) {  
    return 'Saudações ' . $nome;  
});
```



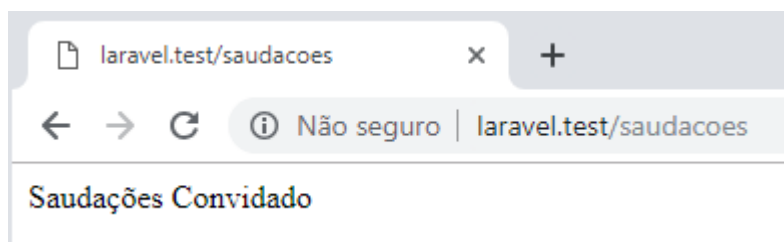
Dessa forma o parâmetro é obrigatório e se não for especificado é retornado uma mensagem de erro:



Para que o parâmetro seja opcional (não obrigatório ) a rota deve ser especificada da seguinte forma:

```
Route::get('/saudacoes/{nome?}', function ($nome='Convidado') {  
    return 'Saudações ' . $nome;  
});
```

Onde é passado um valor padrão para o parâmetro (que também pode ser nulo):

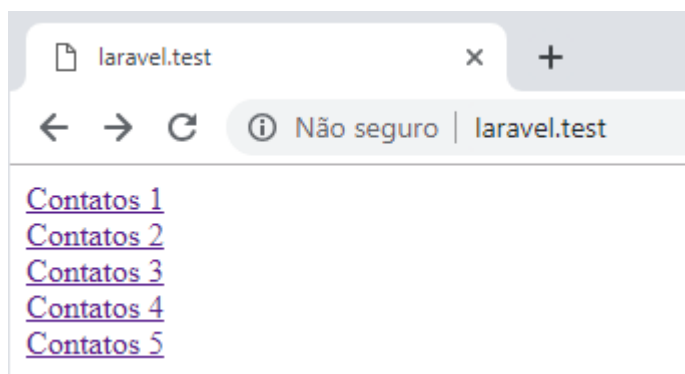


## Aula 05 - Rotas nomeadas

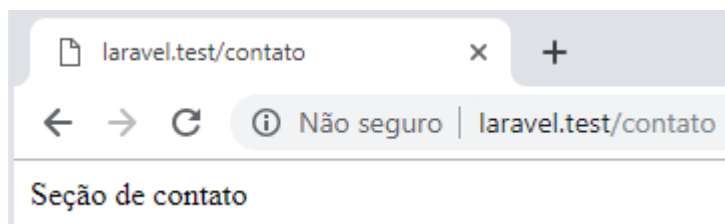
### routes/web.php

```
Route::get('/contato', function () {  
    return 'Seção de contato';  
})->name('contatos');
```

```
Route::get('/', function () {  
    echo "<a href='" . route('contatos') . "'>Contatos 1</a><br />";  
    echo "<a href='" . route('contatos') . "'>Contatos 2</a><br />";  
    echo "<a href='" . route('contatos') . "'>Contatos 3</a><br />";  
    echo "<a href='" . route('contatos') . "'>Contatos 4</a><br />";  
    echo "<a href='" . route('contatos') . "'>Contatos 5</a><br />";  
});
```



Clicando em qualquer um dos links:

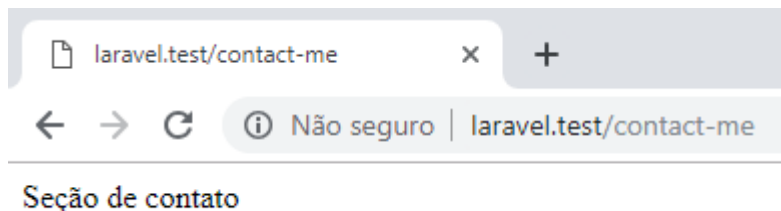


Se for alterada a URL:

```
Route::get('/contact-me', function () {  
    return 'Seção de contato';  
})->name('contatos');
```

```
Route::get('/', function () {  
    echo "<a href='" . route('contatos') . "'>Contatos 1</a><br />";  
    echo "<a href='" . route('contatos') . "'>Contatos 2</a><br />";  
    echo "<a href='" . route('contatos') . "'>Contatos 3</a><br />";  
    echo "<a href='" . route('contatos') . "'>Contatos 4</a><br />";  
    echo "<a href='" . route('contatos') . "'>Contatos 5</a><br />";  
});
```

Clicando em qualquer um dos links:



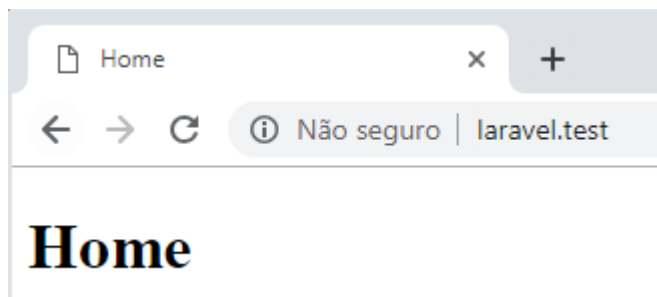
## Aula 6 - Views

### routes/web.php

```
Route::get('/', function () {  
    return view('home');  
})->name('home');
```

### resources/views/home.blade.php

```
<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
    <meta charset="UTF-8">  
    <title>Home</title>  
</head>  
<body>  
    <h1>Home</h1>  
</body>  
</html>
```



## Passando valores para a view

**resources/views/home.blade.php**

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Home</title>
</head>
<body>
  <h1>Home</h1>
  Bem-vindo <?php echo $nome ?>
</body>
</html>
```

### Primeira forma

**routes/web.php**

```
Route::get('/', function () {
    $nome = 'Roberto';
    return view('home')->with('nome', $nome);
})->name('home');
```

### Segunda forma

```
Route::get('/', function () {
    $nome = 'Roberto';
    return view('home')->with(['nome' => $nome]);
})->name('home');
```

### Terceira forma

```
Route::get('/', function () {
    $nome = 'Roberto';
    return view('home', compact('nome'));
})->name('home');
```

### Quarta forma

**routes/web.php**

```
Route::view('/', 'home', ['nome' => 'Roberto'])->name('home');
```



## routes/web.php

```
Route::view('/', 'home')->name('home');
Route::view('/about', 'about')->name('about');
Route::view('/portfolio', 'portfolio')->name('portfolio');
Route::view('/contato', 'contato')->name('contato');
```

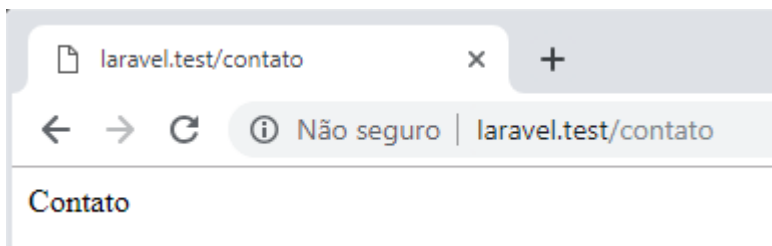
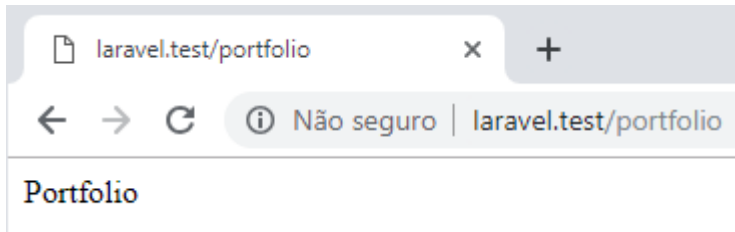
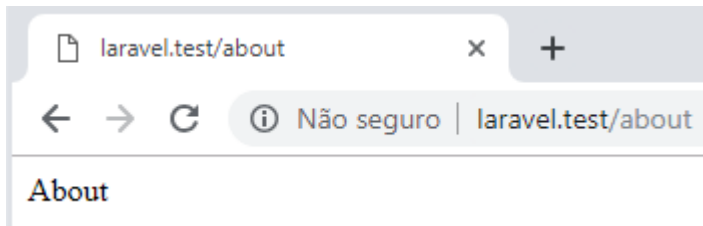
## resources/views/home.blade.php

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="UTF-8">
    <title>Home</title>
  </head>
  <body>
    <nav>
      <ul>
        <li><a href="/">Home</a></li>
        <li><a href="/about">About</a></li>
        <li><a href="/portfolio">Portfolio</a></li>
        <li><a href="/contato">Contato</a></li>
      </ul>
    </nav>

    <h1>Home</h1>
    Bem-vindo <?php echo $nome ?? 'Convidado' ?>
  </body>
</html>
```







Nesse caso, é sempre necessário voltar a página home para ter acesso aos links.

## Aula 7 - Blade, o motor de templates

resources/views/layouts/template.blade.php

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>@yield('title', 'Aprendible')</title>
</head>
<body>
  <nav>
    <ul>
      <li><a href="/">Home</a></li>
      <li><a href="/about">About</a></li>
      <li><a href="/portfolio">Portfolio</a></li>
      <li><a href="/contato">Contato</a></li>
    </ul>
  </nav>

  @yield('content')

</body>
</html>
```

**resources/views/home.blade.php**

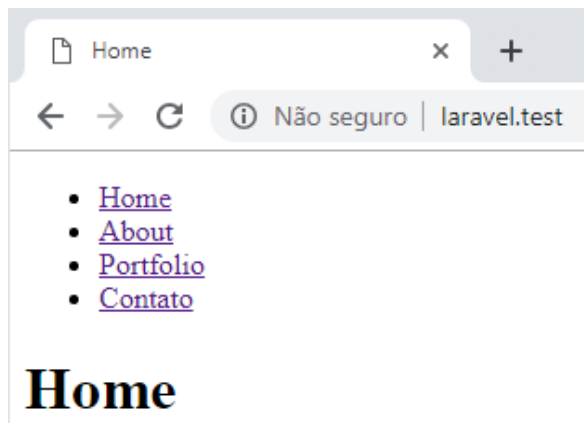
```
@extends('layouts.template')
```

```
@section('title','Home')
```

```
@section('content')
```

```
<h1>Home</h1>
```

```
@endsection
```



**resources/views/about.blade.php**

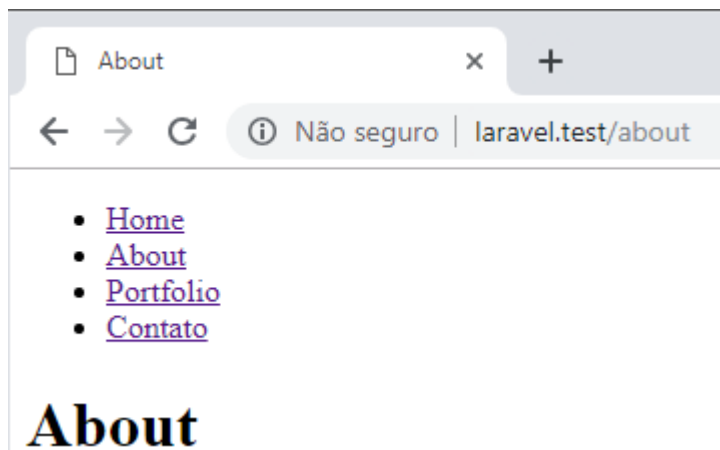
```
@extends('layouts.template')
```

```
@section('title','About')
```

```
@section('content')
```

```
<h1>About</h1>
```

```
@endsection
```



**resources/views/portfolio.blade.php**

```
@extends('layouts.template')
```

```
@section('title','Portfolio')
```

```
@section('content')  
    <h1>Portfolio</h1>  
@endsection
```

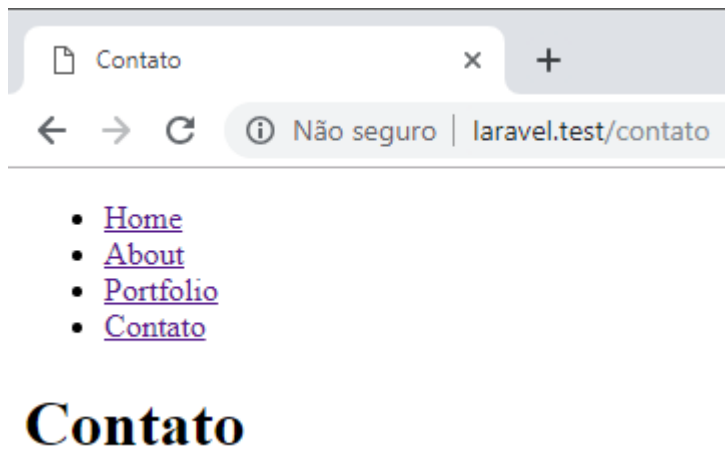


**resources/views/contato.blade.php**

```
@extends('layouts.template')
```

```
@section('title','Contato')
```

```
@section('content')  
    <h1>Contato</h1>  
@endsection
```



## Aula 8 - Estruturas de controle

### routes/web.php

```
$portfolio = [  
    ["title" => "Projeto #1"],  
    ["title" => "Projeto #2"],  
    ["title" => "Projeto #3"],  
    ["title" => "Projeto #4"]  
];  
  
Route::view('/', 'home')->name('home');  
Route::view('/about', 'about')->name('about');  
Route::view('/portfolio', 'portfolio', compact('portfolio'))->name('portfolio');  
Route::view('/contato', 'contato')->name('contato');
```

### resources/views/portfolio.blade.php

```
@extends('layouts.template')  
  
@section('title', 'Portfolio')  
  
@section('content')  
    <h1>Portfolio</h1>  
  
    <ul>  
        <?php  
            foreach($portfolio as $portfolioItem)  
            {  
                echo "<li>" . $portfolioItem['title'] . "</li>";  
            }  
        ?>  
    </ul>  
@endsection
```

Com Blade:

**resources/views/portfolio.blade.php**

```
@extends('layouts.template')
```

```
@section('title','Portfolio')
```

```
@section('content')
```

```
<h1>Portfolio</h1>
```

```
<ul>
```

```
    @foreach($portfolio as $portfolioItem)
```

```
        <li>{{ $portfolioItem['title'] }}</li>
```

```
    @endforeach
```

```
</ul>
```

```
@endsection
```



## routes/web.php

```
$portfolio = [
    // ["title" => "Projeto #1"],
    // ["title" => "Projeto #2"],
    // ["title" => "Projeto #3"],
    // ["title" => "Projeto #4"]
];

Route::view('/', 'home')->name('home');
Route::view('/about', 'about')->name('about');
Route::view('/portfolio', 'portfolio', compact('portfolio'))->name('portfolio');
Route::view('/contato', 'contato')->name('contato');
```

## resources/views/portfolio.blade.php

```
@extends('layouts.template')

@section('title','Portfolio')

@section('content')
    <h1>Portfolio</h1>
    <ul>
        @if ($portfolio)
            @foreach($portfolio as $portfolioItem)
                <li>{{ $portfolioItem['title'] }}</li>
            @endforeach
        @else
            <li>Não há projetos para mostrar!</li>
        @endif
    </ul>
@endsection
```



## routes/web.php

```
// $portfolio = [  
    // ["title" => "Projeto #1"],  
    // ["title" => "Projeto #2"],  
    // ["title" => "Projeto #3"],  
    // ["title" => "Projeto #4"]  
//];
```

Route::view('/', 'home')->name('home');

Route::view('/about', 'about')->name('about');

Route::view('/portfolio', 'portfolio', compact('portfolio'))->name('portfolio');

Route::view('/contato', 'contato')->name('contato');

Whoops! There was an error.

Não seguro | laravel.test/portfolio

**ErrorException (E\_ERROR)**  
**Undefined variable: portfolio (View: C:\laragon\www\laravel\resources\views\portfolio.blade.php)**

Application frames (4)  
All frames (58)

57 ErrorException  
... \storage\framework\views\b8699482f44948145dd1e172a271eb645cc24c4  
:7

56 ErrorException  
... \storage\framework\views\b8699482f44948145dd1e172a271eb645cc24c4  
:7

55 Illuminate\Foundation\Bootstrap\HandleExceptions  
handleError  
... \storage\framework\views\b8699482f44948145dd1e172a271eb645cc24c4  
:7

C:\laragon\www\laravel\storage\framework\views\b8699482f44948145dd1e172a271eb645cc24c4

```
1. <?php /* C:\laragon\www\laravel\resources\views\portfolio.blade.php */ ?>  
2. <?php $__env->startSection('title','Portfolio'); ?>  
3.  
4. <?php $__env->startSection('content'); ?>  
5.     <h1>Portfolio</h1>  
6.     <ul>  
7.         <?php if($portfolio): ?>  
8.             <?php $__currentLoopData = $portfolio; $__env-  
9. >addLoop($__currentLoopData); foreach($__currentLoopData as  
10. $portfolioItem): $__env->incrementLoopIndices(); $loop = $__env-  
11. >getLastLoop(); ?>  
12.                 <li><?php echo e($portfolioItem['title']); ?></li>  
13.             <?php endforeach; $__env->popLoop(); $loop = $__env-  
14. >getLastLoop(); ?>  
15.         <?php else: ?>  
16.             <li>Não há projetos para mostrar!</li>  
17.         <?php endif; ?>  
18.     </ul>  
19. <?php $__env->stopSection(); ?>  
20.  
21. <?php echo $__env->make('layouts.template',  
22. \Illuminate\Support\Arr::except(get_defined_vars(), ['__data', '__path']))-  
23. >render(); ?>
```

Arguments

- "Undefined variable: portfolio (View: C:\laragon\www\laravel\resources\views\portfolio.blade.php)"

No comments for this stack frame.

Environment & details:

GET Data empty



## Diretiva @isset

### routes/web.php

```
$portfolio = [
    // ["title" => "Projeto #1"],
    // ["title" => "Projeto #2"],
    // ["title" => "Projeto #3"],
    // ["title" => "Projeto #4"]
];

Route::view('/', 'home')->name('home');
Route::view('/about', 'about')->name('about');
Route::view('/portfolio', 'portfolio', compact('portfolio'))->name('portfolio');
Route::view('/contato', 'contato')->name('contato');
```

### resources/views/portfolio.blade.php

```
@extends('layouts.template')

@section('title', 'Portfolio')

@section('content')
    <h1>Portfolio</h1>
    <ul>
        @isset($portfolio)
            @foreach($portfolio as $portfolioItem)
                <li>{{ $portfolioItem['title'] }}</li>
            @endforeach
        @else
            <li>Não há projetos para mostrar!</li>
        @endisset
    </ul>
@endsection
```



## Diretiva @forelse

resources/views/portfolio.blade.php

```
@extends('layouts.template')
```

```
@section('title','Portfolio')
```

```
@section('content')
```

```
<h1>Portfolio</h1>
```

```
<ul>
```

```
    @forelse($portfolio as $portfolioItem)
```

```
        <li>{{ $portfolioItem['title'] }}</li>
```

```
    @empty
```

```
        <li>Não há projetos para mostrar!</li>
```

```
    @endforelse
```

```
</ul>
```

```
@endsection
```



## routes/web.php

```
$portfolio = [  
    ["title" => "Projeto #1"],  
    ["title" => "Projeto #2"],  
    // ["title" => "Projeto #3"],  
    // ["title" => "Projeto #4"]  
];
```

```
Route::view('/', 'home')->name('home');
```

```
Route::view('/about', 'about')->name('about');
```

```
Route::view('/portfolio', 'portfolio', compact('portfolio'))->name('portfolio');
```

```
Route::view('/contato', 'contato')->name('contato');
```



## resources/views/portfolio.blade.php

```
@extends('layouts.template')
```

```
@section('title','Portfolio')
```

```
@section('content')
```

```
<h1>Portfolio</h1>
```

```
<ul>
```

```
    @forelse($portfolio as $portfolioItem)
```

```
        <li>{{ $portfolioItem['title'] }} <pre>{{var_dump($loop)}}</pre></li>
```

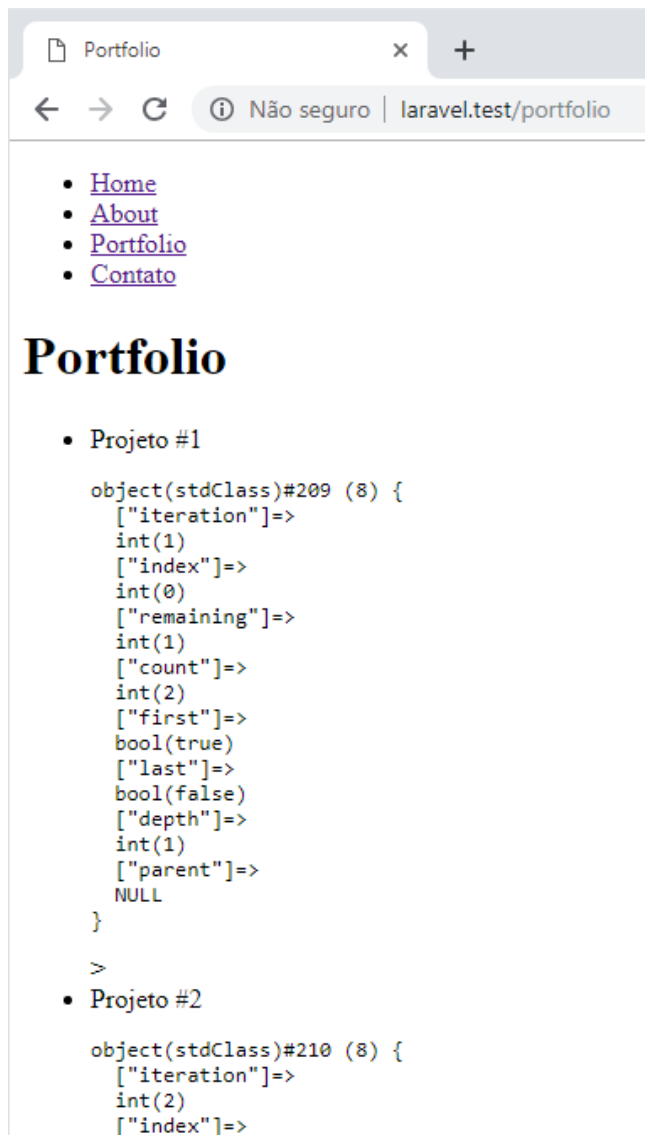
```
    @empty
```

```
        <li>Não há projetos para mostrar!</li>
```

```
    @endforelse
```

```
</ul>
```

```
@endsection
```



## routes/web.php

```
$portfolio = [
    ["title" => "Projeto #1"],
    ["title" => "Projeto #2"],
    ["title" => "Projeto #3"],
    ["title" => "Projeto #4"]
];

Route::view('/', 'home')->name('home');
Route::view('/about', 'about')->name('about');
Route::view('/portfolio', 'portfolio', compact('portfolio'))->name('portfolio');
Route::view('/contato', 'contato')->name('contato');
```

## resources/views/portfolio.blade.php

```
@extends('layouts.template')

@section('title', 'Portfolio')

@section('content')
    <h1>Portfolio</h1>
    <ul>
        @forelse($portfolio as $portfolioItem)
            <li>{{ $portfolioItem['title'] }} <small>{{ $loop->last ? 'Último loop' : '' }}</small></li>
        @empty
            <li>Não há projetos para mostrar!</li>
        @endforelse
    </ul>
@endsection
```



## resources/views/portfolio.blade.php

```
@extends('layouts.template')
```

```
@section('title','Portfolio')
```

```
@section('content')
```

```
<h1>Portfolio</h1>
```

```
<ul>
```

```
    @forelse($portfolio as $portfolioItem)
```

```
        <li>{{ $portfolioItem['title'] }} <small>{{ $loop->first ? 'Primeiro loop' : ''}}</small></li>
```

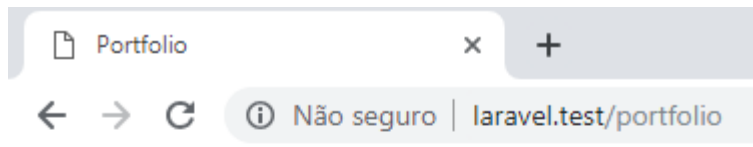
```
    @empty
```

```
        <li>Não há projetos para mostrar!</li>
```

```
    @endforelse
```

```
</ul>
```

```
@endsection
```



- [Home](#)
- [About](#)
- [Portfolio](#)
- [Contato](#)

## Portfolio

- Projeto #1 Primeiro loop
- Projeto #2
- Projeto #3
- Projeto #4

## Aula 9 - Controllers

### Listando as rotas existentes

php artisan route:list

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ php artisan route:list
```

| Domain | Method   | URI       | Name      | Action                            | Middleware   |
|--------|----------|-----------|-----------|-----------------------------------|--------------|
|        | GET HEAD | /         | home      | Illuminate\Routing\ViewController | web          |
|        | GET HEAD | about     | about     | Illuminate\Routing\ViewController | web          |
|        | GET HEAD | api/user  |           | Closure                           | api,auth:api |
|        | GET HEAD | contato   | contato   | Illuminate\Routing\ViewController | web          |
|        | GET HEAD | portfolio | portfolio | Illuminate\Routing\ViewController | web          |

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ |
```

### Comentando a rota em api.php

routes/api.php

```
// Route::middleware('auth:api')->get('/user', function (Request $request) {
//     return $request->user();
// });
```

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ php artisan route:list
```

| Domain | Method   | URI       | Name      | Action                            | Middleware |
|--------|----------|-----------|-----------|-----------------------------------|------------|
|        | GET HEAD | /         | home      | Illuminate\Routing\ViewController | web        |
|        | GET HEAD | about     | about     | Illuminate\Routing\ViewController | web        |
|        | GET HEAD | contato   | contato   | Illuminate\Routing\ViewController | web        |
|        | GET HEAD | portfolio | portfolio | Illuminate\Routing\ViewController | web        |

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ |
```

## Controller - Opções

php artisan make:controller PortfolioController -h

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ php artisan make:controller PortfolioController -h
Description:
  Create a new controller class

Usage:
  make:controller [options] [--] <name>

Arguments:
  name                The name of the class

Options:
  -m, --model[=MODEL]  Generate a resource controller for the given model.
  -r, --resource        Generate a resource controller class.
  -i, --invokable       Generate a single method, invokable controller class.
  -p, --parent[=PARENT] Generate a nested resource controller class.
  --api                Exclude the create and edit methods from the controller.
  -h, --help            Display this help message
  -q, --quiet           Do not output any message
  -V, --version         Display this application version
  --ansi               Force ANSI output
  --no-ansi            Disable ANSI output
  -n, --no-interaction Do not ask any interactive question
  --env[=ENV]          The environment the command should run under
  -v|vv|vvv, --verbose Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ |
```

## Criando um controller sem opções

php artisan make:controller PortfolioController

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ php artisan make:controller PortfolioController
Controller created successfully.

Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ |
```

app/Http/Controllers/PortfolioController.php

```
<?php
```

```
namespace App\Http\Controllers;
use Illuminate\Http\Request;
```

```
class PortfolioController extends Controller
{
    //
}
```



## app/Http/Controllers/Controller.php

```
<?php
```

```
namespace App\Http\Controllers;
```

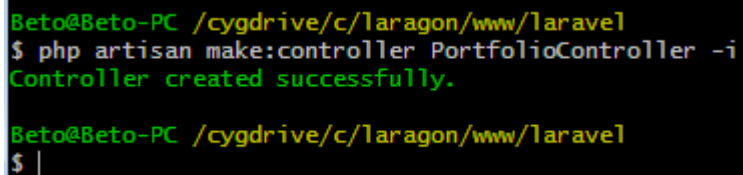
```
use Illuminate\Foundation\Bus\DispatchesJobs;  
use Illuminate\Routing\Controller as BaseController;  
use Illuminate\Foundation\Validation\ValidatesRequests;  
use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
```

```
class Controller extends BaseController  
{  
    use AuthorizesRequests, DispatchesJobs, ValidatesRequests;  
}
```

## Criando um controller com método invoke

Elimine o controller PortfolioController e o recrie com a opção i:

```
php artisan make:controller PortfolioController -i
```



```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel  
$ php artisan make:controller PortfolioController -i  
Controller created successfully.  
  
Beto@Beto-PC /cygdrive/c/laragon/www/laravel  
$ |
```

## routes/web.php

```
Route::view('/', 'home')->name('home');  
Route::view('/about', 'about')->name('about');  
Route::get('/portfolio', 'PortfolioController')->name('portfolio');  
Route::view('/contato', 'contato')->name('contato');
```

## app/Http/Controllers/PortfolioController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class PortfolioController extends Controller
{
    public function __invoke(Request $request)
    {
        $portfolio = [
            ["title" => "Projeto #1"],
            ["title" => "Projeto #2"],
            ["title" => "Projeto #3"],
            ["title" => "Projeto #4"]
        ];

        return view('portfolio', compact('portfolio'));
    }
}
```



# Listando as rotas

php artisan route:list

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ php artisan route:list
```

| Domain | Method   | URI       | Name      | Action                                   | Middleware |
|--------|----------|-----------|-----------|--|------------|
|        | GET HEAD | /         | home      | Illuminate\Routing\ViewController        | web        |
|        | GET HEAD | about     | about     | Illuminate\Routing\ViewController        | web        |
|        | GET HEAD | contato   | contato   | Illuminate\Routing\ViewController        | web        |
|        | GET HEAD | portfolio | portfolio | App\Http\Controllers\PortfolioController | web        |

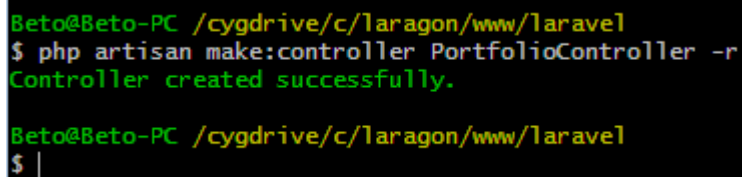
```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ |
```

## Aula 10 - Controllers resources & API

### Controllers do tipo resource

- Elimine o controller PortfolioController.
- Comente a rota criada anteriormente para ele.
- Recrie o controller PortfolioController com o comando:

`php artisan make:controller PortfolioController -r`



```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ php artisan make:controller PortfolioController -r
Controller created successfully.

Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ |
```

Neste controlador são criados sete métodos:

1. index
2. create
3. store
4. show
5. edit
6. update
7. destroy

- index: lista os projetos;
- create: usado para criar um novo projeto para portfolio;
- store: usado para guardar recursos (projetos) que foram enviados previamente pelo método create na base de dados;
- show: exibe um recurso (projeto) específico encontrado por seu identificador;
- edit: exibe um formulário para editar um recurso (projeto) já existente (um projeto previamente criado).
- update: guarda as alterações feitas no formulário de edição (alterações nos projetos);
- destroy: elimina o recurso (um projeto) pelo seu identificador.

## app/Http/Controllers/PortfolioController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class PortfolioController extends Controller
{
    public function index()
    {
        $portfolio = [
            ["title" => "Projeto #1"],
            ["title" => "Projeto #2"],
            ["title" => "Projeto #3"],
            ["title" => "Projeto #4"]
        ];
        return view('portfolio', compact('portfolio'));
    }
    public function create()
    {
        //
    }
    public function store(Request $request)
    {
        //
    }
    public function show($id)
    {
        //
    }

    public function edit($id)
    {
        //
    }
    public function update(Request $request, $id)
    {
        //
    }
    public function destroy($id)
    {
        //
    }
}
```

routes/web.php

Route::resource('projetos','PortfolioController');

Listando as rotas

php artisan route:list

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ php artisan route:list
```

| Domain | Method    | URI                     | Name             | Action   | Middleware |
|--------|-----------|-------------------------|------------------|--|------------|
|        | GET HEAD  | projetos                | projetos.index   | App\Http\Controllers\PortfolioController@index   | web        |
|        | POST      | projetos                | projetos.store   | App\Http\Controllers\PortfolioController@store   | web        |
|        | GET HEAD  | projetos/create         | projetos.create  | App\Http\Controllers\PortfolioController@create  | web        |
|        | GET HEAD  | projetos/{projeto}      | projetos.show    | App\Http\Controllers\PortfolioController@show    | web        |
|        | PUT PATCH | projetos/{projeto}      | projetos.update  | App\Http\Controllers\PortfolioController@update  | web        |
|        | DELETE    | projetos/{projeto}      | projetos.destroy | App\Http\Controllers\PortfolioController@destroy | web        |
|        | GET HEAD  | projetos/{projeto}/edit | projetos.edit    | App\Http\Controllers\PortfolioController@edit    | web        |

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ |
```

## Roteando apenas alguns métodos

routes/web.php

```
Route::resource('projetos','PortfolioController')->only(['index', 'show']);
```

php artisan route:list

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ php artisan route:list
```

| Domain | Method   | URI                | Name           | Action   | Middleware |
|--------|----------|--------------------|----------------|--|------------|
|        | GET HEAD | projetos           | projetos.index | App\Http\Controllers\PortfolioController@index | web        |
|        | GET HEAD | projetos/{projeto} | projetos.show  | App\Http\Controllers\PortfolioController@show  | web        |

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ |
```

```
Route::resource('projetos','PortfolioController')->except(['index', 'show']);
```

Listando as rotas

php artisan route:list

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ php artisan route:list
```

| Domain | Method    | URI                     | Name             | Action   | Middleware |
|--------|-----------|-------------------------|------------------|--|------------|
|        | POST      | projetos                | projetos.store   | App\Http\Controllers\PortfolioController@store   | web        |
|        | GET HEAD  | projetos/create         | projetos.create  | App\Http\Controllers\PortfolioController@create  | web        |
|        | PUT PATCH | projetos/{projeto}      | projetos.update  | App\Http\Controllers\PortfolioController@update  | web        |
|        | DELETE    | projetos/{projeto}      | projetos.destroy | App\Http\Controllers\PortfolioController@destroy | web        |
|        | GET HEAD  | projetos/{projeto}/edit | projetos.edit    | App\Http\Controllers\PortfolioController@edit    | web        |

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ |
```

## Criando controller do tipo API

`php artisan make:controller PortfolioController --api`

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ php artisan make:controller PortfolioController --api
Controller created successfully.

Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ |
```

São criados cinco métodos:

1. index
2. store
3. show
4. update
5. destroy

Obs.: Os métodos create e edit não são criados porque não são utilizados em API.



### **app/Http/Controllers/PortfolioController.php**

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class PortfolioController extends Controller
{
    public function index()
    {
        //
    }
    public function store(Request $request)
    {
        //
    }
    public function show($id)
    {
        //
    }

    public function update(Request $request, $id)
    {
        //
    }
    public function destroy($id)
    {
        //
    }
}
```

### **routes/web.php**

```
Route::apiResource('projetos','PortfolioController');
```

## Listando as rotas

php artisan route:list

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ php artisan route:list
```

| Domain | Method    | URI                | Name             | Action   | Middleware |
|--------|-----------|--------------------|------------------|--|------------|
|        | GET HEAD  | projetos           | projetos.index   | App\Http\Controllers\PortfolioController@index   | web        |
|        | POST      | projetos           | projetos.store   | App\Http\Controllers\PortfolioController@store   | web        |
|        | GET HEAD  | projetos/{projeto} | projetos.show    | App\Http\Controllers\PortfolioController@show    | web        |
|        | PUT PATCH | projetos/{projeto} | projetos.update  | App\Http\Controllers\PortfolioController@update  | web        |
|        | DELETE    | projetos/{projeto} | projetos.destroy | App\Http\Controllers\PortfolioController@destroy | web        |

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ |
```

## Alterando o nome de métodos padrões

app/Providers/AppServiceProvider.php

```
<?php
```

```
namespace App\Providers;
```

```
use Illuminate\Support\Facades\Route;
use Illuminate\Support\ServiceProvider;
```

```
class AppServiceProvider extends ServiceProvider
{
    public function register()
    {
        //
    }
    public function boot()
    {
        Route::resourceVerbs([
            'create' => 'criar',
            'edit' => 'editar'
        ]);
    }
}
```

routes/web.php

Route::resource('projetos','PortfolioController');

Listando as rotas

php artisan route:list

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ php artisan route:list
```

| Domain | Method    | URI                       | Name             | Action   | Middleware |
|--------|-----------|---------------------------|------------------|--|------------|
|        | GET HEAD  | projetos                  | projetos.index   | App\Http\Controllers\PortfolioController@index   | web        |
|        | POST      | projetos                  | projetos.store   | App\Http\Controllers\PortfolioController@store   | web        |
|        | GET HEAD  | projetos/criar            | projetos.create  | App\Http\Controllers\PortfolioController@create  | web        |
|        | GET HEAD  | projetos/{projeto}        | projetos.show    | App\Http\Controllers\PortfolioController@show    | web        |
|        | PUT PATCH | projetos/{projeto}        | projetos.update  | App\Http\Controllers\PortfolioController@update  | web        |
|        | DELETE    | projetos/{projeto}        | projetos.destroy | App\Http\Controllers\PortfolioController@destroy | web        |
|        | GET HEAD  | projetos/{projeto}/editar | projetos.edit    | App\Http\Controllers\PortfolioController@edit    | web        |

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ |
```

## app/Http/Controllers/PortfolioController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class PortfolioController extends Controller
{
    public function index()
    {
        $portfolio = [
            ["title" => "Projeto #1"],
            ["title" => "Projeto #2"],
            ["title" => "Projeto #3"],
            ["title" => "Projeto #4"]
        ];
        return view('portfolio', compact('portfolio'));
    }
    public function store(Request $request)
    {
        //
    }
    public function show($id)
    {
        //
    }
    public function update(Request $request, $id)
    {
        //
    }
    public function destroy($id)
    {
        //
    }
    public function criar(){

    }
    public function editar(){

    }
}
```

## routes/web.php

```
Route::view('/', 'home')->name('home');
Route::view('/about', 'about')->name('about');
Route::get('/portfolio', 'PortfolioController@index')->name('portfolio');
Route::view('/contato', 'contato')->name('contato');
```

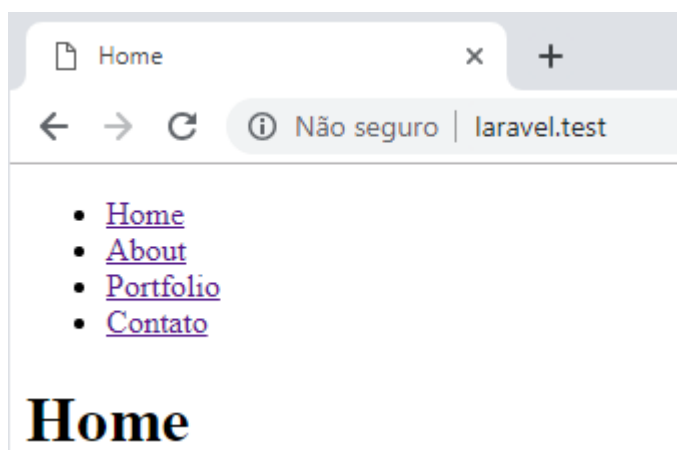
## app/Http/Controllers/PortfolioController.php

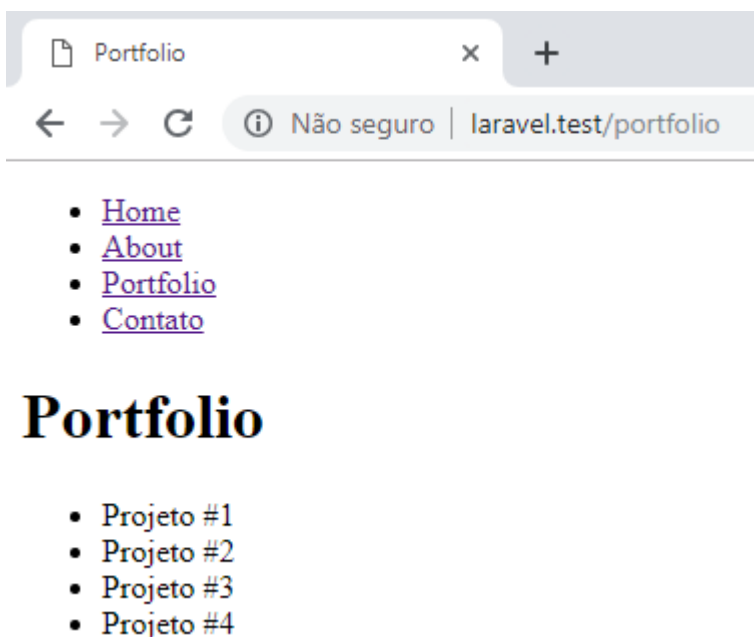
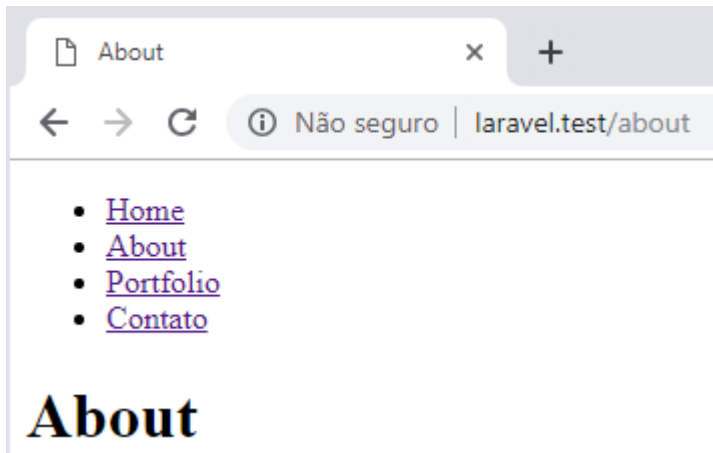
```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
class PortfolioController extends Controller
{
    public function index()
    {
        $portfolio = [
            ["title" => "Projeto #1"],
            ["title" => "Projeto #2"],
            ["title" => "Projeto #3"],
            ["title" => "Projeto #4"]
        ];
        return view('portfolio', compact('portfolio'));
    }
}
```





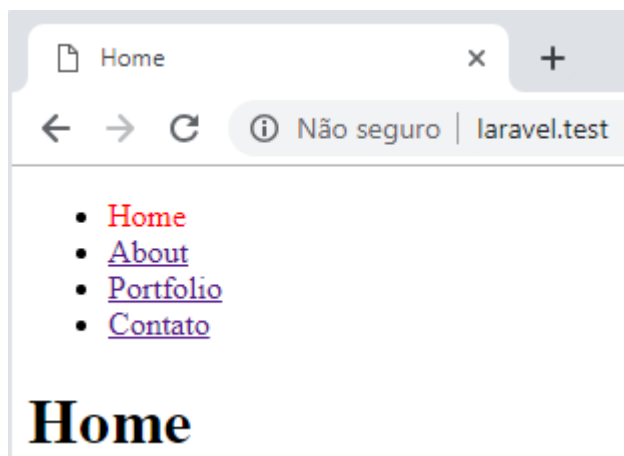
## Aula 11 - Ativação de links de navegação

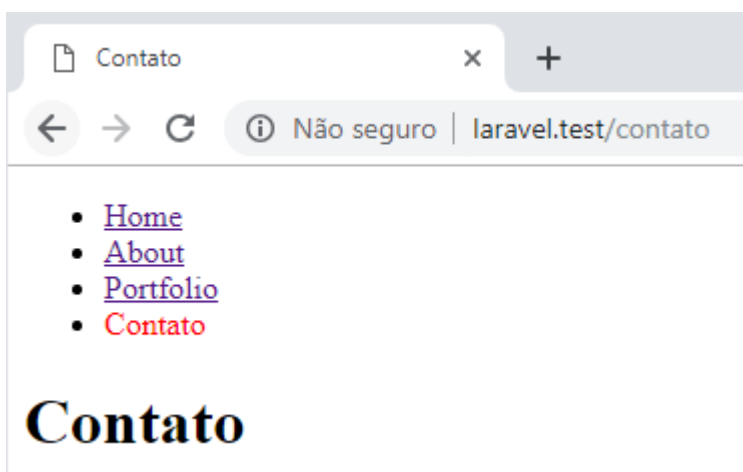
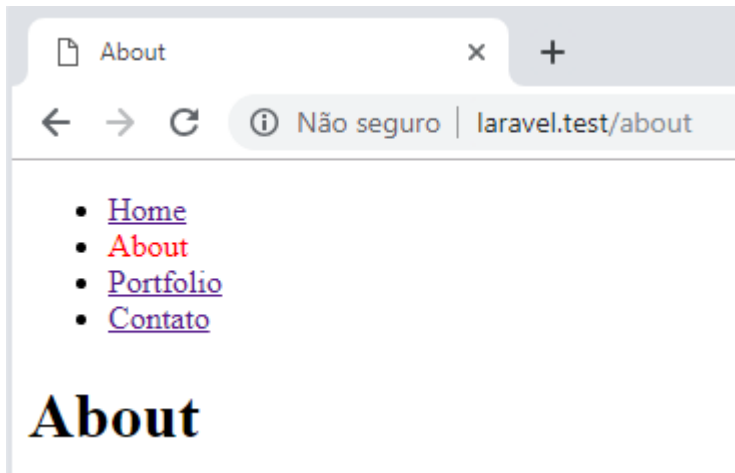
resources/views/layouts/template.blade.php

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>@yield('title', 'Aprendível')</title>
  <style>
    .active a{
      color: red;
      text-decoration: none;
    }
  </style>
</head>
<body>
  <nav>
    <ul>
      <li class={{request()->routeIs('home') ? 'active' : ''}}>
        <a href="/">Home</a></li>
      <li class={{request()->routeIs('about') ? 'active' : ''}}>
        <a href="/about">About</a></li>
      <li class={{request()->routeIs('portfolio') ? 'active' : ''}}>
        <a href="/portfolio">Portfolio</a></li>
      <li class={{request()->routeIs('contato') ? 'active' : ''}}>
        <a href="/contato">Contato</a></li>
    </ul>
  </nav>

  @yield('content')

</body>
</html>
```







## Criando uma função reutilizável

### app/helpers.php

```
<?php

function setActive($routeName)
{
    return request()->routeIs($routeName) ? 'active' : '';
}
```

### resources/views/layouts/template.blade.php

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>@yield('title', 'Aprendible')</title>
    <style>
        .active a{
            color: red;
            text-decoration: none;
        }
    </style>
</head>
<body>
    <nav>
        <ul>
            <li class={{ setActive('home') }}><a href="/">Home</a></li>
            <li class={{ setActive('about') }}><a href="/about">About</a></li>
            <li class={{ setActive('portfolio') }}><a href="/portfolio">Portfolio</a></li>
            <li class={{ setActive('contato') }}><a href="/contato">Contato</a></li>
        </ul>
    </nav>

    @yield('content')
</body>
</html>
```

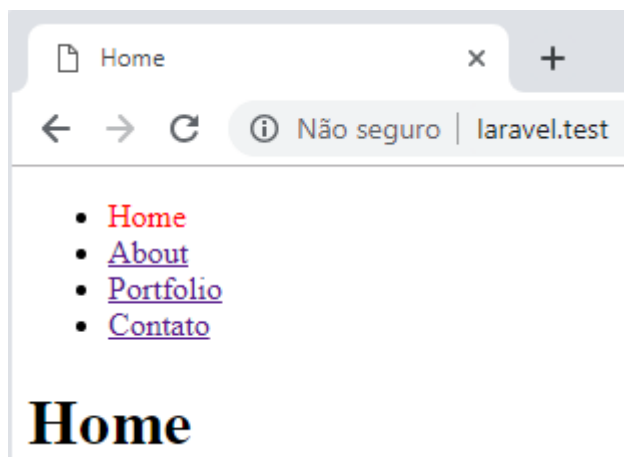
## composer.json

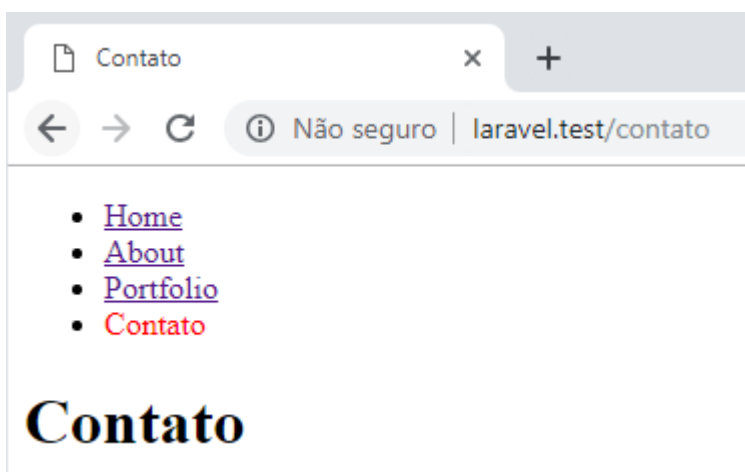
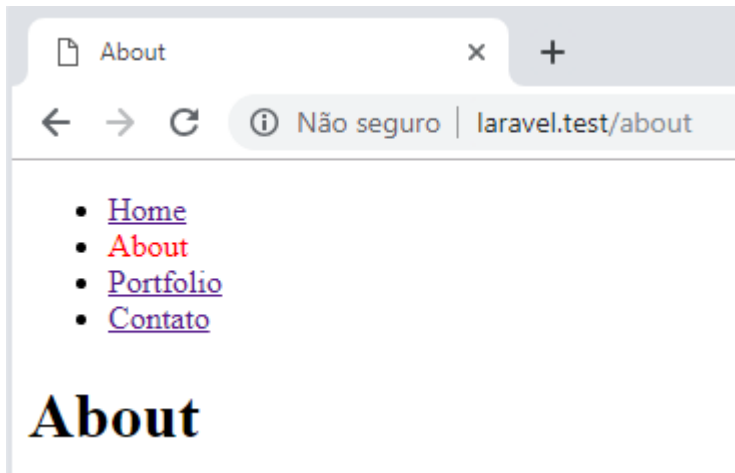
```
"autoload": {
    "psr-4": {
        "App\\": "app/"
    },
    "classmap": [
        "database/seeds",
        "database/factories"
    ],
    "files": [
        "app/helpers.php"
    ]
}
```

## Atualizando o arquivo de dependências

composer dump-autoload

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ composer dump-autoload
Generating optimized autoload files> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: beyondcode/laravel-dump-server
Discovered Package: fideloper/proxy
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
Generated optimized autoload files containing 3520 classes
```





## resources/views/partials/nav.blade.php

```
<nav>
  <ul>
    <li class={{ setActive('home') }}><a href="/">Home</a></li>
    <li class={{ setActive('about') }}><a href="/about">About</a></li>
    <li class={{ setActive('portfolio') }}><a href="/portfolio">Portfolio</a></li>
    <li class={{ setActive('contato') }}><a href="/contato">Contato</a></li>
  </ul>
</nav>
```

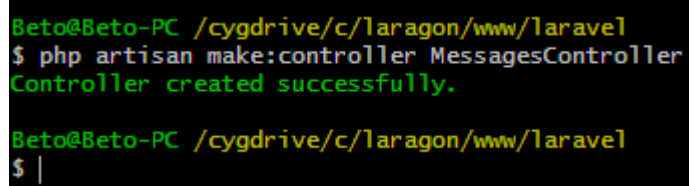
## resources/views/layouts/template.blade.php

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>@yield('title', 'Aprendible')</title>
  <style>
    .active a{
      color: red;
      text-decoration: none;
    }
  </style>
</head>
<body>
  @include('partials.nav')
  @yield('content')
</body>
</html>
```

## Aula 12 - Como enviar formulários

### Criando o controller

```
php artisan make:controller MessagesController
```

A terminal window with a black background and green text. The prompt is 'Beto@Beto-PC /cygdrive/c/laragon/www/laravel'. The command entered is '\$ php artisan make:controller MessagesController'. The output is 'Controller created successfully.' followed by a new prompt line '\$ |'.

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ php artisan make:controller MessagesController
Controller created successfully.
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ |
```

routes/web.php

```
Route::post('/contato','MessagesController@store');
```

## Formulário de contato

resources/views/contato.blade.php

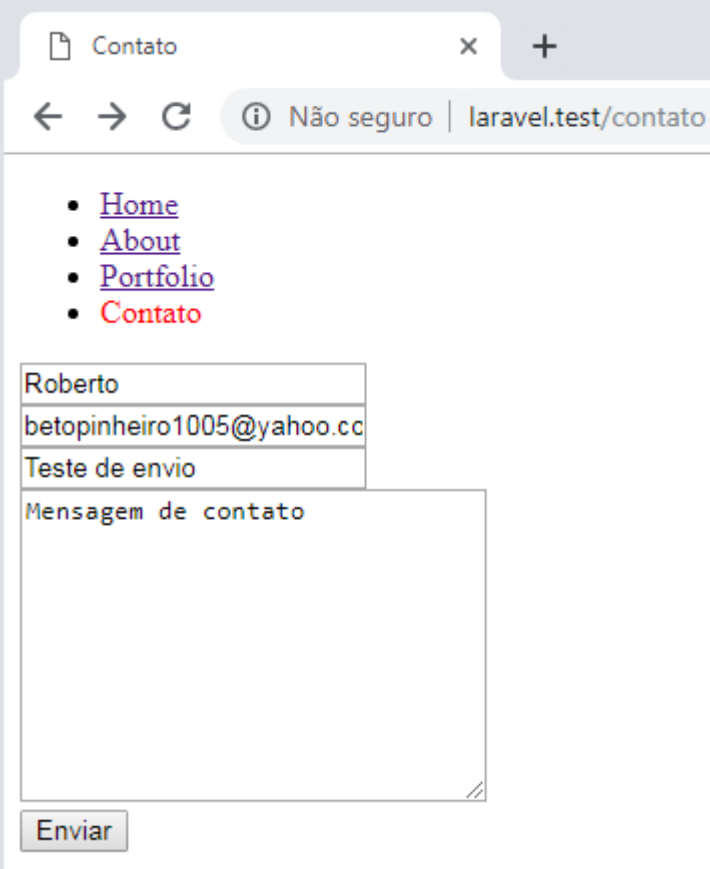
```
@extends('layouts.template')
```

```
@section('title', 'Contato')
```

```
@section('content')
```

```
<form method="POST" action="{{ route('contato') }}">
    @csrf
    <input type="text" name="nome" placeholder="Nome" value="Roberto"><br>
    <input type="email" name="email" placeholder="Email" value="betopinheiro1005@yahoo.com.br"><br>
    <input type="text" name="subject" placeholder="Assunto" value="Teste de envio"><br>
    <textarea name="message" id="message" cols="30" rows="10" placeholder="Mensagem"
value="Mensagem a ser enviada"></textarea><br>
    <button type="submit">Enviar</button>
</form>
```

```
@endsection
```



The screenshot shows a web browser window with the title 'Contato'. The address bar displays 'laravel.test/contato' with a 'Não seguro' (Not secure) warning. The page content includes a navigation menu with links for 'Home', 'About', 'Portfolio', and 'Contato' (which is highlighted in red). Below the menu is a contact form with the following fields and values:

|                           |
|---------------------------|
| Roberto                   |
| betopinheiro1005@yahoo.cc |
| Teste de envio            |
| Mensagem de contato       |

At the bottom of the form is an 'Enviar' (Send) button.

## Retornando todos os dados do formulário

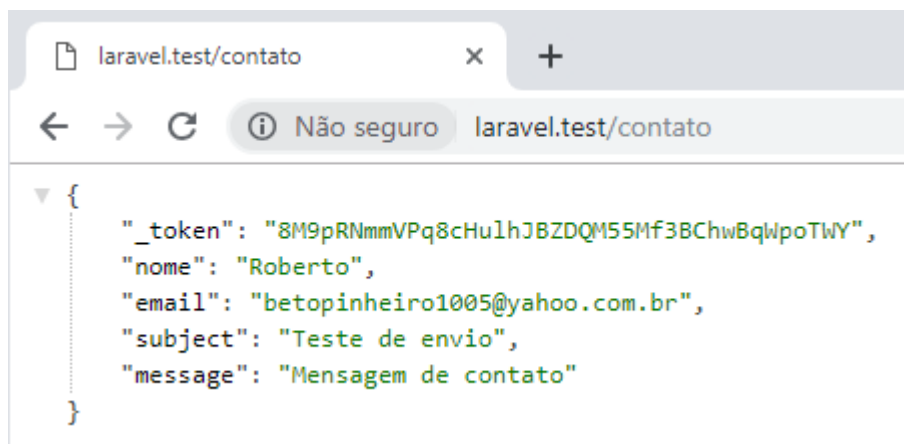
**app/http/Controllers/MessagesController.php**

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
class MessagesController extends Controller
{
    public function store(Request $request){
        // return 'Processar o formulário';
        return $request;
    }
}
```



## Retornando dados parciais do formulário

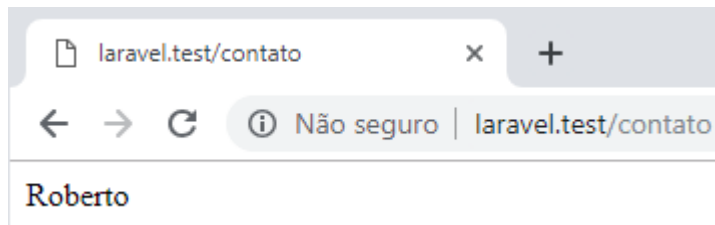
**app/http/Controllers/MessagesController.php**

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class MessagesController extends Controller
{
    public function store(Request $request){
        return $request->get('nome');  }
}
```

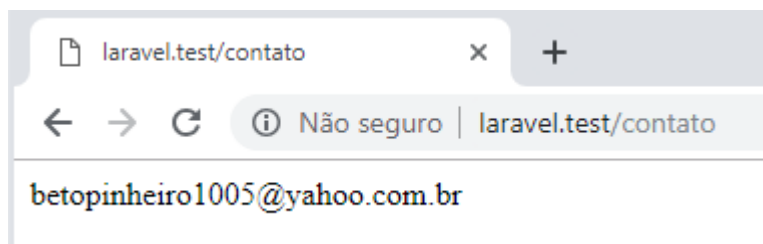


**app/http/Controllers/MessagesController.php**

```
<?php

namespace App\Http\Controllers;

class MessagesController extends Controller
{
    public function store(){
        return request('email');
    }
}
```





## Aula 13 - Como validar formulários

### Exibindo a lista de todos os erros de validação

**app/http/Controllers/MessagesController.php**

```
<?php

namespace App\Http\Controllers;

class MessagesController extends Controller
{
    public function store(){

        request()->validate([
            'nome' => 'required',
            'email' => 'required'
        ]);

        return 'Dados validados!';
    }
}
```

**resources/views/contato.blade.php**

```
@extends('layouts.template')

@section('title', 'Contato')

@section('content')

@if($errors->any())
    <ul>
        @foreach($errors->all() as $error)
            <li>{{ $error }}</li>
        @endforeach
    </ul>
@endif
```

```

<form method="POST" action="{{ route('contato') }}">
    @csrf
    <input type="text" name="nome" placeholder="Nome" value=""><br>
    <input type="email" name="email" placeholder="Email" value=""><br>
    <input type="text" name="subject" placeholder="Assunto" value="Teste de envio"><br>
    <textarea name="message" id="message" cols="30" rows="10" placeholder="Mensagem">Mensagem de
contato</textarea><br>
    <button type="submit">Enviar</button>
</form>

@endsection

```

Contato

← → ↻ ⓘ Não seguro | laravel.test/contato

- [Home](#)
- [About](#)
- [Portfolio](#)
- **Contato**

- The nome field is required.
- The email field is required.

Nome

Email

Teste de envio

Mensagem de contato

Enviar

## Exibindo os erros abaixo de cada campo

### app/http/Controllers/MessagesController.php

```
<?php

namespace App\Http\Controllers;

class MessagesController extends Controller
{
    public function store(){

        request()->validate([
            'nome' => 'required',
            'email' => 'required|email',
            'subject' => 'required',
            'content' => 'required|min:3'
        ]);

        return 'Dados validados!';
    }
}
```

### resources/views/contato.blade.php

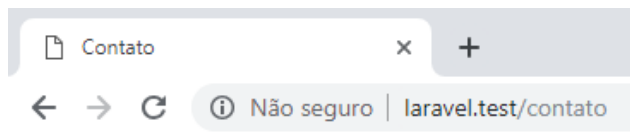
```
@extends('layouts.template')

@section('title', 'Contato')

@section('content')

<form method="POST" action="{{ route('contato') }}">
    @csrf
    <input type="text" name="nome" placeholder="Nome" value="{{ old('nome') }}"><br>
    {!! $errors->first('nome', '<small>:message</small><br>') !!}
    <input type="text" name="email" placeholder="Email" value="{{ old('email') }}"><br>
    {!! $errors->first('email', '<small>:message</small><br>') !!}
    <input type="text" name="subject" placeholder="Assunto" value="{{ old('subject') }}"><br>
    {!! $errors->first('subject', '<small>:message</small><br>') !!}
    <textarea name="content" cols="30" rows="10" placeholder="Mensagem">{{ old('content') }}</textarea><br>
    {!! $errors->first('content', '<small>:message</small><br>') !!}
    <button type="submit">Enviar</button>
</form>

@endsection
```



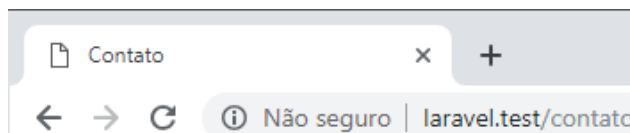
- [Home](#)
- [About](#)
- [Portfolio](#)
- [Contato](#)

The nome field is required.

The email field is required.

The subject field is required.

The content field is required.



- [Home](#)
- [About](#)
- [Portfolio](#)
- [Contato](#)

The email must be a valid email address.

The subject field is required.

The content must be at least 3 characters.

# Aula 14 - Como traduzir suas aplicações

## Traduzindo mensagens de validação

config/app.php

```
'locale' => 'pt-BR',

'fallback_locale' => 'en',
```

resources/lang/pt-BR/validation.php

```
<?php
return [
    /*
    |-----
    | Validation Language Lines
    |-----
    |
    | The following language lines contain the default error messages used by
    | the validator class. Some of these rules have multiple versions such
    | as the size rules. Feel free to tweak each of these messages.
    |
    */
    'accepted'      => 'O campo :attribute deve ser aceito.',
    'active_url'     => 'O campo :attribute deve conter uma URL válida.',
    'after'          => 'O campo :attribute deve conter uma data posterior a :date.',
    'after_or_equal' => 'O campo :attribute deve conter uma data superior ou igual a :date.',
    'alpha'          => 'O campo :attribute deve conter apenas letras.',
    'alpha_dash'     => 'O campo :attribute deve conter apenas letras, números e traços.',
    'alpha_num'      => 'O campo :attribute deve conter apenas letras e números.',
    'array'          => 'O campo :attribute deve conter um array.',
    'before'         => 'O campo :attribute deve conter uma data anterior a :date.',
    'before_or_equal' => 'O campo :attribute deve conter uma data inferior ou igual a :date.',
    'between'        => [
        'numeric' => 'O campo :attribute deve conter um número entre :min e :max.',
        'file'    => 'O campo :attribute deve conter um arquivo de :min a :max kilobytes.',
        'string'  => 'O campo :attribute deve conter entre :min a :max caracteres.',
        'array'   => 'O campo :attribute deve conter de :min a :max itens.',
    ],
    'boolean'        => 'O campo :attribute deve conter o valor verdadeiro ou falso.',
    'confirmed'      => 'A confirmação para o campo :attribute não coincide.',
    'date'           => 'O campo :attribute não contém uma data válida.',
    'date_equals'    => 'O campo :attribute deve ser uma data igual a :date.',
    'date_format'    => 'A data informada para o campo :attribute não respeita o formato :format.',
    'different'      => 'Os campos :attribute e :other devem conter valores diferentes.',
    'digits'         => 'O campo :attribute deve conter :digits dígitos.',
    'digits_between' => 'O campo :attribute deve conter entre :min a :max dígitos.',
    'dimensions'     => 'O valor informado para o campo :attribute não é uma dimensão de imagem válida.',
```

```

'distinct'      => 'O campo :attribute contém um valor duplicado.',
'email'         => 'O campo :attribute não contém um endereço de email válido.',
'exists'        => 'O valor selecionado para o campo :attribute é inválido.',
'file'          => 'O campo :attribute deve conter um arquivo.',
'filled'        => 'O campo :attribute é obrigatório.',
'gt'            => [
    'numeric' => 'O campo :attribute deve ser maior que :value.',
    'file'    => 'O arquivo :attribute deve ser maior que :value kilobytes.',
    'string'  => 'O campo :attribute deve ser maior que :value caracteres.',
    'array'   => 'O campo :attribute deve ter mais que :value itens.',
],
'gte'          => [
    'numeric' => 'O campo :attribute deve ser maior ou igual a :value.',
    'file'    => 'O arquivo :attribute deve ser maior ou igual a :value kilobytes.',
    'string'  => 'O campo :attribute deve ser maior ou igual a :value caracteres.',
    'array'   => 'O campo :attribute deve ter :value itens ou mais.',
],
'image'         => 'O campo :attribute deve conter uma imagem.',
'in'            => 'O campo :attribute não contém um valor válido.',
'in_array'      => 'O campo :attribute não existe em :other.',
'integer'       => 'O campo :attribute deve conter um número inteiro.',
'ip'           => 'O campo :attribute deve conter um IP válido.',
'ipv4'         => 'O campo :attribute deve conter um IPv4 válido.',
'ipv6'         => 'O campo :attribute deve conter um IPv6 válido.',
'json'         => 'O campo :attribute deve conter uma string JSON válida.',
'lt'           => [
    'numeric' => 'O campo :attribute deve ser menor que :value.',
    'file'    => 'O arquivo :attribute ser menor que :value kilobytes.',
    'string'  => 'O campo :attribute deve ser menor que :value caracteres.',
    'array'   => 'O campo :attribute deve ter menos que :value itens.',
],
'lte'          => [
    'numeric' => 'O campo :attribute deve ser menor ou igual a :value.',
    'file'    => 'O arquivo :attribute ser menor ou igual a :value kilobytes.',
    'string'  => 'O campo :attribute deve ser menor ou igual a :value caracteres.',
    'array'   => 'O campo :attribute não deve ter mais que :value itens.',
],
'max'          => [
    'numeric' => 'O campo :attribute não pode conter um valor superior a :max.',
    'file'    => 'O campo :attribute não pode conter um arquivo com mais de :max kilobytes.',
    'string'  => 'O campo :attribute não pode conter mais de :max caracteres.',
    'array'   => 'O campo :attribute deve conter no máximo :max itens.',
],
'mimes'        => 'O campo :attribute deve conter um arquivo do tipo: :values.',
'mimetypes'    => 'O campo :attribute deve conter um arquivo do tipo: :values.',
'min'          => [
    'numeric' => 'O campo :attribute deve conter um número superior ou igual a :min.',
    'file'    => 'O campo :attribute deve conter um arquivo com no mínimo :min kilobytes.',
    'string'  => 'O campo :attribute deve conter no mínimo :min caracteres.',
    'array'   => 'O campo :attribute deve conter no mínimo :min itens.',
],
'not_in'       => 'O campo :attribute contém um valor inválido.',
'not_regex'    => 'O formato do valor :attribute é inválido.',
'numeric'      => 'O campo :attribute deve conter um valor numérico.',
'present'      => 'O campo :attribute deve estar presente.',
'regex'        => 'O formato do valor informado no campo :attribute é inválido.',
'required'     => 'O campo :attribute é obrigatório.',

```

```

'required_if'      => 'O campo :attribute é obrigatório quando o valor do campo :other é igual a :value.',
'required_unless'  => 'O campo :attribute é obrigatório a menos que :other esteja presente em :values.',
'required_with'    => 'O campo :attribute é obrigatório quando :values está presente.',
'required_with_all' => 'O campo :attribute é obrigatório quando um dos :values está presente.',
'required_without' => 'O campo :attribute é obrigatório quando :values não está presente.',
'required_without_all' => 'O campo :attribute é obrigatório quando nenhum dos :values está presente.',
'same'             => 'Os campos :attribute e :other devem conter valores iguais.',
'size'             => [
    'numeric' => 'O campo :attribute deve conter o número :size.',
    'file'    => 'O campo :attribute deve conter um arquivo com o tamanho de :size kilobytes.',
    'string'  => 'O campo :attribute deve conter :size caracteres.',
    'array'   => 'O campo :attribute deve conter :size itens.',
],
'starts_with'     => 'O campo :attribute deve começar com um dos seguintes valores: :values',
'string'          => 'O campo :attribute deve ser uma string.',
'timezone'        => 'O campo :attribute deve conter um fuso horário válido.',
'unique'          => 'O valor informado para o campo :attribute já está em uso.',
'uploaded'        => 'Falha no Upload do arquivo :attribute.',
'url'             => 'O formato da URL informada para o campo :attribute é inválido.',
'uuid'           => 'O campo :attribute deve ser um UUID válido.',
/*
|-----
| Custom Validation Language Lines
|-----
|
| Here you may specify custom validation messages for attributes using the
| convention "attribute.rule" to name the lines. This makes it quick to
| specify a specific custom language line for a given attribute rule.
|
*/
'custom' => [
    'attribute-name' => [
        'rule-name' => 'custom-message',
    ],
],
],

```

```

/*
|-----
| Custom Validation Attributes
|-----
|
| The following language lines are used to swap attribute place-holders
| with something more reader friendly such as E-Mail Address instead
| of "email". This simply helps us make messages a little cleaner.
|
*/
'attributes' => [
    'address'      => 'endereço',
    'age'          => 'idade',
    'body'         => 'conteúdo',
    'city'         => 'cidade',
    'country'      => 'país',
    'date'         => 'data',
    'day'          => 'dia',
    'description'  => 'descrição',
    'excerpt'      => 'resumo',
    'first_name'   => 'primeiro nome',
    'gender'       => 'gênero',
    'hour'         => 'hora',
    'last_name'    => 'sobrenome',
    'message'      => 'mensagem',
    'minute'       => 'minuto',
    'mobile'       => 'celular',
    'month'        => 'mês',
    'name'         => 'nome',
    'password_confirmation' => 'confirmação da senha',
    'password'     => 'senha',
    'phone'        => 'telefone',
    'second'       => 'segundo',
    'sex'          => 'sexo',
    'state'        => 'estado',
    'subject'      => 'assunto',
    'text'         => 'texto',
    'time'         => 'hora',
    'title'        => 'título',
    'username'     => 'usuário',
    'year'         => 'ano',
    'email'        => 'e-mail',
    'remember'     => 'lembrar-se de mim',
],
];

```



## resources/lang/pt-BR/auth.php

```
<?php
return [
    /*
    |-----
    | Authentication Language Lines
    |-----
    |
    | The following language lines are used during authentication for various
    | messages that we need to display to the user. You are free to modify
    | these language lines according to your application's requirements.
    |
    */
    'failed' => 'Credenciais informadas não correspondem com nossos registros.',
    'throttle' => 'Você realizou muitas tentativas de login. Por favor, tente novamente em :seconds segundos.',
];
```

## resources/lang/pt-BR/pagination.php

```
<?php
return [
    /*
    |-----
    | Pagination Language Lines
    |-----
    |
    | The following language lines are used by the paginator library to build
    | the simple pagination links. You are free to change them to anything
    | you want to customize your views to better match your application.
    |
    */
    'previous' => '&laquo; Anterior',
    'next' => 'Próxima &raquo;',
];
```

## resources/lang/pt-BR/passwords.php

```
<?php
return [
    /*
    |-----
    | Password Reminder Language Lines
    |-----
    |
    | The following language lines are the default lines which match reasons
    | that are given by the password broker for a password update attempt
    | has failed, such as for an invalid token or invalid new password.
    |
    */
    'password' => 'A senha deve conter pelo menos seis caracteres e ser igual à confirmação.',
    'reset'    => 'Sua senha foi redefinida!',
    'sent'     => 'Enviamos um link para redefinir a sua senha por e-mail.',
    'token'    => 'Esse código de redefinição de senha é inválido.',
    'user'     => 'Não conseguimos encontrar nenhum usuário com o endereço de e-mail informado.',
];
```

Contato

← → ↻ ⓘ Não seguro | laravel.test/contato

- [Home](#)
- [About](#)
- [Portfolio](#)
- **Contato**

Nome

O campo nome é obrigatório.

Email

O campo e-mail é obrigatório.

Assunto

O campo assunto é obrigatório.

Mensagem

O campo content é obrigatório.

Enviar

Contato

← → ↻ ⓘ Não seguro | laravel.test/contato

- [Home](#)
- [About](#)
- [Portfolio](#)
- **Contato**

Roberto

betopinheiro1005

O campo e-mail não contém um endereço de email válido.

Assunto

O campo assunto é obrigatório.

ab

O campo content deve conter no mínimo 3 caracteres.

Enviar

## resources/lang/pt-BR/pt-BR.json

```
{
  "A fresh verification link has been sent to your email address.": "Um novo link de verificação foi enviado para seu endereço de e-mail.",
  "All rights reserved.": "Todos os direitos reservados.",
  "Before proceeding, please check your email for a verification link.": "Antes de prosseguir, verifique seu e-mail em busca de um link de verificação.",
  "click here to request another": "clique aqui para solicitar outro",
  "Confirm Password": "Confirmar senha",
  "E-Mail Address": "Endereço de e-mail",
  "Error": "Erro",
  "Forbidden": "Proibido",
  "Forgot Your Password?": "Esqueceu a senha?",
  "Go Home": "Ir para o início",
  "Hello!": "Olá!",
  "hi": "oi",
  "If you did not create an account, no further action is required.": "Se você não criou uma conta, ignore este e-mail.",
  "If you did not receive the email": "Se você não recebeu o e-mail",
  "If you did not request a password reset, no further action is required.": "Se você não solicitou essa redefinição de senha, ignore este e-mail.",
  "If you're having trouble clicking the \":actionText\" button, copy and paste the URL below\nto your web browser: [:actionURL](actionURL)": "Se você estiver com problemas para clicar no botão \":actionText\", copie e cole o URL abaixo\nem seu navegador da web: [:actionURL](actionURL)",
  "Login": "Entrar",
  "Logout": "Sair",
}
```

```

    "Name": "Nome",
    "Oh no": "Ah não",
    "Page Expired": "Página expirou",
    "Page Not Found": "Página não encontrada",
    "Password": "Senha",
    "Please click the button below to verify your email address.": "Por favor, clique no botão abaixo para verificar seu endereço de e-mail.",
    "Regards": "Atenciosamente",
    "Register": "Registrar",
    "Remember Me": "Lembre-se de mim",
    "Reset Password": "Redefinir senha",
    "Reset Password Notification": "Notificação de redefinição de senha",
    "Send Password Reset Link": "Link de redefinição de senha enviado",
    "Service Unavailable": "Serviço indisponível",
    "Sorry, the page you are looking for could not be found.": "Desculpe, a página que você está procurando não foi encontrada.",
    "Sorry, you are forbidden from accessing this page.": "Desculpe, você está proibido de acessar esta página.",
    "Sorry, you are making too many requests to our servers.": "Desculpe, você está fazendo muitas solicitações para nossos servidores.",
    "Sorry, you are not authorized to access this page.": "Desculpe, você não está autorizado a acessar esta página.",
    "Sorry, your session has expired. Please refresh and try again.": "Desculpe, sua sessão expirou. Por favor, atualize e tente novamente.",
    "Sorry, we are doing some maintenance. Please check back soon.": "Desculpe, estamos fazendo algumas manutenções. Por favor, volte em breve.",
    "This password reset link will expire in :count minutes.": "O Link de redefinição de senha irá expirar em :count minutos.",
    "Toggle navigation": "Alternar navegação",
    "Too Many Requests": "Muitas solicitações",
    "Unauthorized": "Não autorizado",
    "Verify Email Address": "Verifique o endereço de e-mail",
    "Verify Your Email Address": "Verifique seu endereço de e-mail",
    "You are receiving this email because we received a password reset request for your account.": "Você recebeu esse e-mail porque foi solicitado uma redefinição de senha na sua conta.",
    "Whoops!": "Ops!",
    "Whoops, something went wrong on our servers.": "Ops, algo deu errado em nossos servidores."
}

```

Obs: Esses arquivos podem ser baixados em:

<https://github.com/caouecs/Laravel-lang/tree/master/src>  
<https://github.com/caouecs/Laravel-lang/tree/master/json>

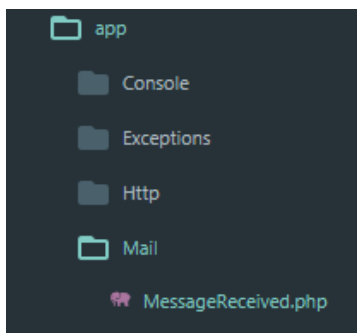
# Aula 15 - Como enviar emails

## Criando um mailable

php artisan make:mail MessageReceived

```
Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ php artisan make:mail MessageReceived
Mail created successfully.

Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ |
```



### app/Mail/MessageReceived.php

```
<?php
```

```
namespace App\Mail;
```

```
use Illuminate\Bus\Queueable;
```

```
use Illuminate\Mail\Mailable;
```

```
use Illuminate\Queue\SerializesModels;
```

```
use Illuminate\Contracts\Queue\ShouldQueue;
```

```
class MessageReceived extends Mailable
```

```
{
```

```
    use Queueable, SerializesModels;
```

```
    public function __construct()
```

```
    {
```

```
        //
```

```
    }
```

```
    public function build()
```

```
    {
```

```
        return $this->view('emails.message-received');
```

```
    }
```

```
}
```

**resources/views/emails/message-received.blade.php**

Conteúdo do email

**app/Http/Controller/MessagesController.php**

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Mail\MessageReceived;  
use Illuminate\Support\Facades\Mail;
```

```
class MessagesController extends Controller
```

```
{
```

```
    public function store(){
```

```
        request()->validate([
```

```
            'nome' => 'required',
```

```
            'email' => 'required|email',
```

```
            'subject' => 'required',
```

```
            'content' => 'required|min:3'
```

```
        ]);
```

```
        /* Enviar email */
```

```
        Mail::to('betopinheiro1005@yahoo.com.br')->send(new MessageReceived);
```

```
        // return 'Dados validados!';
```

```
        return 'Mensagem enviada!';
```

```
    }
```

```
}
```

Contato

← → ↻ ⓘ Não seguro | laravel.test/contato

- [Home](#)
- [About](#)
- [Portfolio](#)
- **Contato**

|                           |
|---------------------------|
| Roberto Pinheiro          |
| betopinheiro1005@yahoo.cc |
| Orçamento                 |

Necessito de um orçamento para um projeto.

Enviar

laravel.test/contato

← → ↻ ⓘ Não seguro | laravel.test/contato

Mensagem enviada!

### storage/logs/laravel-2019-03-12.log

```
[2019-03-12 02:13:04] local.DEBUG: Message-ID:
<66e3a116829bf8018f3afb00935e5f31@laravel.test>
Date: Tue, 12 Mar 2019 02:13:04 +0000
Subject: Message Received
From: Example <hello@example.com>
To: betopinheiro1005@yahoo.com.br
MIME-Version: 1.0
Content-Type: text/html; charset=utf-8
Content-Transfer-Encoding: quoted-printable
```

Conteúdo do email

## Enviando mensagem em HTML

**resources/views/emails/message-received.blade.php**

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Message Received</title>
</head>
<body>
  <p>Conteúdo do email</p>
</body>
</html>
```

**storage/logs/laravel-2019-03-12.log**

```
[2019-03-12 02:20:50] local.DEBUG: Message-ID:
<1e70e7a3a71338bd0ef94241a4f02ad7@laravel.test>
Date: Tue, 12 Mar 2019 02:20:50 +0000
Subject: Message Received
From: Example <hello@example.com>
To: betopinheiro1005@yahoo.com.br
MIME-Version: 1.0
Content-Type: text/html; charset=utf-8
Content-Transfer-Encoding: quoted-printable
```

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Message Received</title>
</head>
<body>
  <p>Conteúdo do email</p>
</body>
</html>
```



## Alterando o endereço de email do remetente

### config/mail.php

```
'from' => [  
    'address' => env('MAIL_FROM_ADDRESS', 'contato@oriontecinfo.com.br'),  
    'name' => env('MAIL_FROM_NAME', 'Roberto Pinheiro'),  
],
```

ou:

### .env

```
MAIL_DRIVER=log  
MAIL_HOST=smtp.mailtrap.io  
MAIL_PORT=2525  
MAIL_USERNAME=null  
MAIL_PASSWORD=null  
MAIL_ENCRYPTION=null  
MAIL_FROM_ADDRESS='contato@oriontecinfo.com.br'  
MAIL_FROM_NAME='Roberto Pinheiro'
```

### storage/logs/laravel-2019-03-12.log

```
[2019-03-12 02:27:58] local.DEBUG: Message-ID:  
<a2007c36b68bf029af07eb27f27712a7@laravel.test>  
Date: Tue, 12 Mar 2019 02:27:58 +0000  
Subject: Message Received  
From: Roberto Pinheiro <contato@oriontecinfo.com.br>  
To: betopinheiro1005@yahoo.com.br  
MIME-Version: 1.0  
Content-Type: text/html; charset=utf-8  
Content-Transfer-Encoding: quoted-printable
```

```
<!DOCTYPE html>  
<html lang="pt-br">  
<head>  
    <meta charset="UTF-8">  
    <title>Message Received</title>  
</head>  
<body>  
    <p>Conteúdo do email</p>  
</body>  
</html>
```

## Alterando o assunto

**app/Mail/MessageReceived.php**

```
<?php

namespace App\Mail;

use Illuminate\Bus\Queueable;
use Illuminate\Mail\Mailable;
use Illuminate\Queue\SerializesModels;
use Illuminate\Contracts\Queue\ShouldQueue;

class MessageReceived extends Mailable
{
    use Queueable, SerializesModels;

    public $subject = 'Mensagem recebida';

    public function __construct()
    {
        //
    }

    public function build()
    {
        return $this->view('emails.message-received');
    }
}
```

**storage/logs/laravel-2019-03-12.log**

[2019-03-12 02:42:15] local.DEBUG: Message-ID:  
<dc78987d537e79811075cf52d10389b1@laravel.test>  
Date: Tue, 12 Mar 2019 02:42:15 +0000  
Subject: **Mensagem recebida**  
From: Roberto Pinheiro <contato@oriontecinfo.com.br>  
To: betopinheiro1005@yahoo.com.br  
MIME-Version: 1.0  
Content-Type: text/html; charset=utf-8  
Content-Transfer-Encoding: quoted-printable

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Message Received</title>
</head>
<body>
  <p>Conteúdo do email</p>
</body>
</html>
```

## Obtendo as informações do formulário de contato

### app/Mail/MessageReceived.php

```
<?php

namespace App\Mail;

use Illuminate\Bus\Queueable;
use Illuminate\Mail\Mailable;
use Illuminate\Queue\SerializesModels;
use Illuminate\Contracts\Queue\ShouldQueue;

class MessageReceived extends Mailable
{
    use Queueable, SerializesModels;

    public $subject = 'Mensagem recebida';
    public $msg;

    public function __construct($msg)
    {
        $this->msg = $msg;
    }

    public function build()
    {
        return $this->view('emails.message-received');
    }
}
```

## resources/views/emails/message-received.blade.php

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Message Received</title>
</head>
<body>
    <p>Conteúdo do email</p>
    {{ var_dump($msg) }}
</body>
</html>
```

## storage/logs/laravel-2019-03-12.log

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Message Received</title>
</head>
<body>
    <p>Conteúdo do email</p>
    array(4) {
        ["nome"]=>
        string(16) "Roberto Pinheiro"
        ["email"]=>
        string(29) "betopinheiro1005@yahoo.com.br"
        ["subject"]=>
        string(10) "Orçamento"
        ["content"]=>
        string(43) "Necessito de um orçamento para um projeto."
    }
</body>
</html>
```

## resources/views/emails/message-received.blade.php

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Message Received</title>
</head>
<body>
    <p>Recebestes uma mensagem de {{ $msg['nome'] }} - {{ $msg['email'] }}</p>
    <p><strong>Assunto: </strong> {{ $msg['subject'] }}</p>
    <p><strong>Conteúdo: </strong> {{ $msg['content'] }}</p>
</body>
</html>
```

## storage/logs/laravel-2019-03-12.log

```
[2019-03-12 03:26:09] local.DEBUG: Message-ID: <4af02db853e2ce3c3b4654ca9968c305@laravel.test>
Date: Tue, 12 Mar 2019 03:26:09 +0000
Subject: Mensagem recebida
From: Roberto Pinheiro <contato@oriontecinfo.com.br>
To: betopinheiro1005@yahoo.com.br
MIME-Version: 1.0
Content-Type: text/html; charset=utf-8
Content-Transfer-Encoding: quoted-printable
```

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Message Received</title>
</head>
<body>
    <p>Recebestes uma mensagem de Roberto Pinheiro - betopinheiro1005@yahoo.com.br</p>
    <p><strong>Assunto: </strong> Orçamento</p>
    <p><strong>Conteúdo: </strong> Necessito de um orçamento para um projeto.</p>
</body>
</html>
```

## Utilizando queue (processos em segundo plano)

**app/Http/Controller/MessagesController.php**

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Mail\MessageReceived;
```

```
use Illuminate\Support\Facades\Mail;
```

```
class MessagesController extends Controller
```

```
{
```

```
    public function store(){
```

```
        // return request('email');
```

```
        $message = request()->validate([
```

```
            'nome' => 'required',
```

```
            'email' => 'required|email',
```

```
            'subject' => 'required',
```

```
            'content' => 'required|min:3'
```

```
        ]);
```

```
        /* Enviar email */
```

```
        Mail::to('betopinheiro1005@yahoo.com.br')->queue(new MessageReceived($message));
```

```
        // return 'Dados validados!';
```

```
        return 'Mensagem enviada!';
```

```
    }
```

```
}
```

## Visualizando o email no navegador

```
<?php

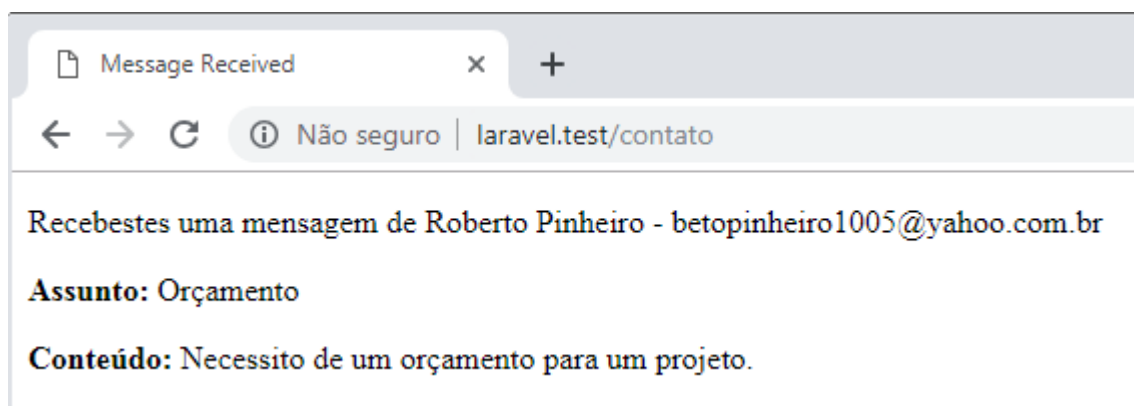
namespace App\Http\Controllers;

use App\Mail\MessageReceived;
use Illuminate\Support\Facades\Mail;

class MessagesController extends Controller
{
    public function store(){
        // return request('email');

        $message = request()->validate([
            'nome' => 'required',
            'email' => 'required|email',
            'subject' => 'required',
            'content' => 'required|min:3'
        ]);

        /* Enviar email */
        return new MessageReceived($message);
    }
}
```

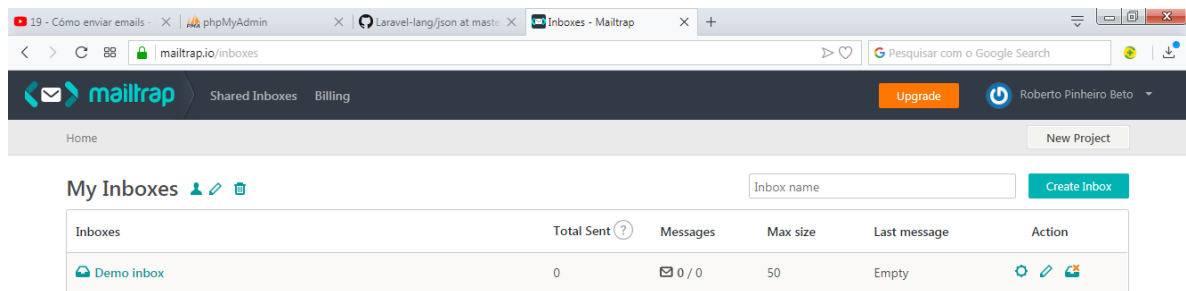
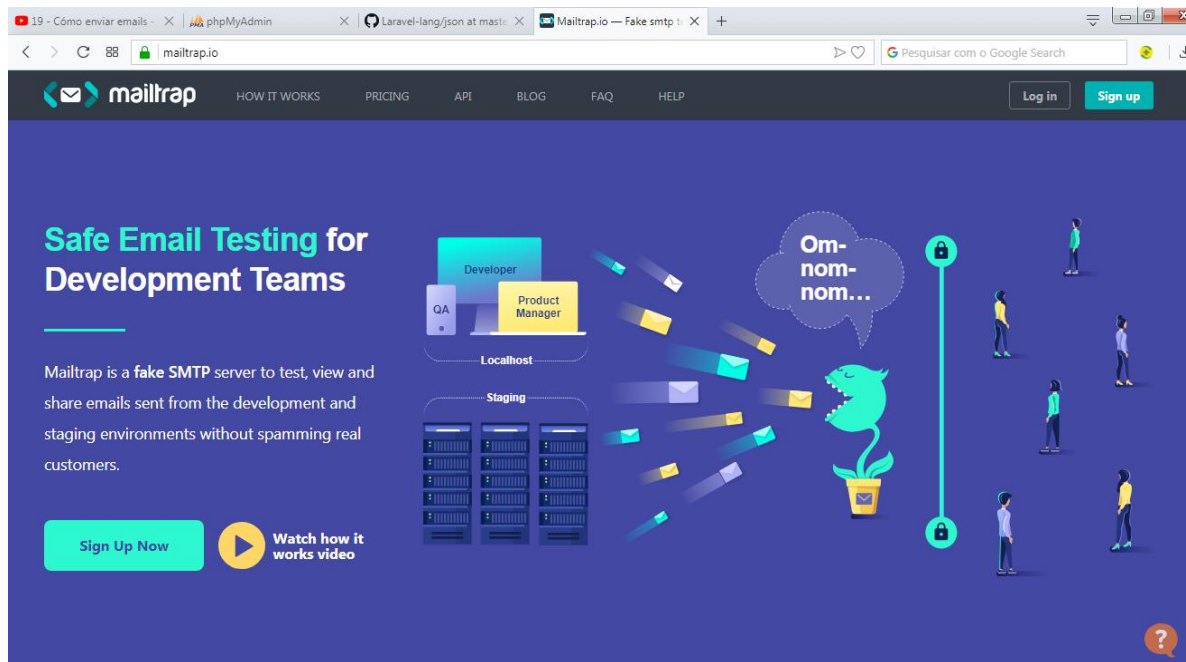




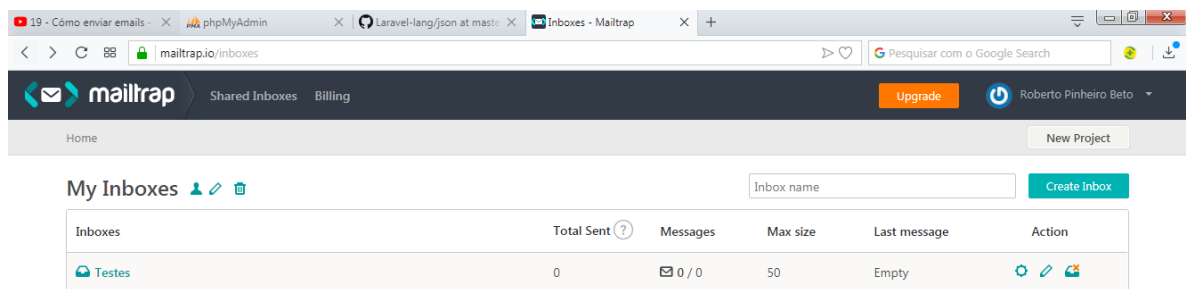
# Usando mailtrap.io

Acesse: <https://mailtrap.io>

E abra uma conta gratuita.



Renomeie "Demo inbox" para "Testes"



Clique no link:

## Integrations

Laravel

Laravel provides a clean, simple API over the popular SwiftMailer library.

With the default Laravel setup you can configure your mailing configuration by setting these values in the .env file in the root directory of your project.

```
MAIL_DRIVER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=f7f9b1c636f954
MAIL_PASSWORD=c6263786f720f6
MAIL_FROM_ADDRESS=from@example.com
MAIL_FROM_NAME=Example
```

Copie os dados de MAIL\_USERNAME e de MAIL\_PASSWORD para o seu arquivo .env e altere MAIL\_DRIVER de log para smtp

### .env

```
MAIL_DRIVER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=f7f9b1c636f954
MAIL_PASSWORD=c6263786f720f6
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS='contato@oriontecinfo.com.br'
MAIL_FROM_NAME='Roberto Pinheiro'
```

Envie novamente o formulário de contato e verifique se ele chegou em sua caixa postal:

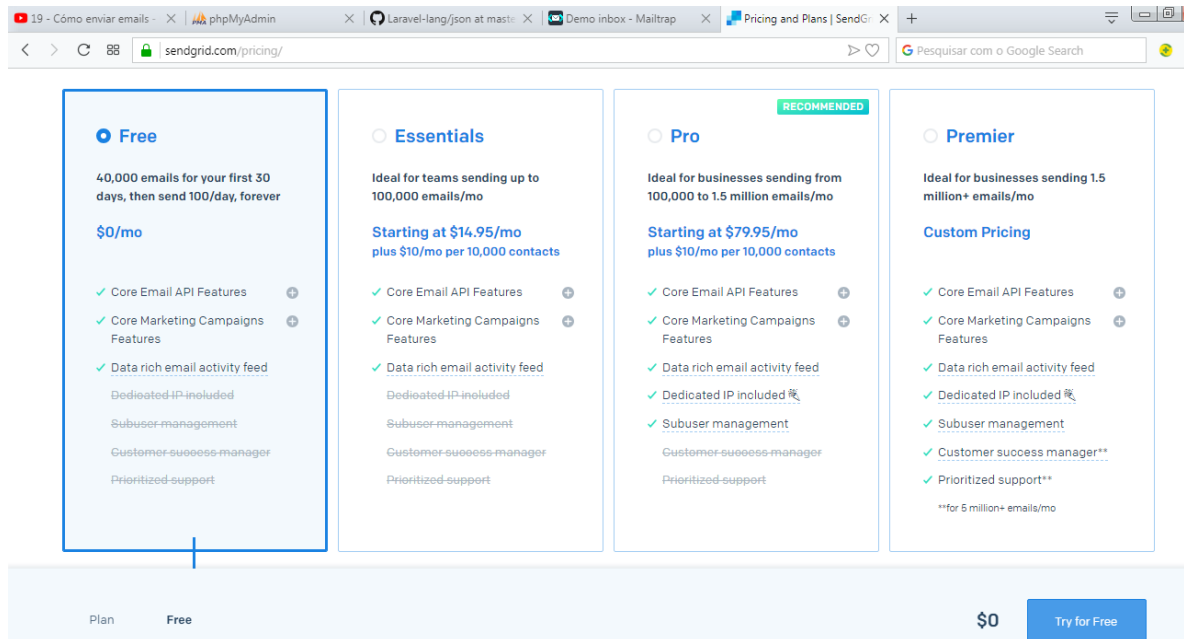
The screenshot shows a web browser with multiple tabs. The active tab is 'Demo inbox - Mailtrap'. The address bar shows 'mailtrap.io/inboxes/563109/messages/1179489058'. The page header includes the Mailtrap logo, 'Shared Inboxes', 'Billing', an 'Upgrade' button, and the user 'Roberto Pinheiro Beto'. The main content area displays a received email with the following details:

- Mensagem recebida**
- To:** <betopinheiro1005@yahoo.com.br> (a few seconds ago)
- From:** Roberto Pinheiro <contato@oriontecinfo.com.br>
- To:** <betopinheiro1005@yahoo.com.br>
- More info**
- HTML** (selected), HTML Source, Text, Raw, Analysis, Check HTML
- Recebestes uma mensagem de Roberto Pinheiro - betopinheiro1005@yahoo.com.br**
- Assunto:** Orçamento
- Conteúdo:** Necessito de um orçamento para um projeto.
- Date:** 2019-03-12 04:02 (a few seconds ago)
- Size:** 688 Bytes

# Sendgrid

Acesse:

<https://sendgrid.com/pricing/>



Selecione o plano FREE que permite enviar até 40 mil emails num período de 1 mês.

## Instalando sendgrid

Por padrão o Laravel não vem nativamente o sendgrid. Portanto é necessário instalá-lo.

Acesse:

<https://github.com/s-ichikawa/laravel-sendgrid-driver>

Instale-o com o composer:

`composer require s-ichikawa/laravel-sendgrid-driver`

```

Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ composer require s-ichikawa/laravel-sendgrid-driver
Using version ^2.0 for s-ichikawa/laravel-sendgrid-driver
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 6 installs, 0 updates, 0 removals
  - Installing ralouphie/getallheaders (2.0.5): Loading from cache
  - Installing psr/http-message (1.0.1): Loading from cache
  - Installing guzzlehttp/psr7 (1.5.2): Loading from cache
  - Installing guzzlehttp/promises (v1.3.1): Loading from cache
  - Installing guzzlehttp/guzzle (6.3.3): Loading from cache
  - Installing s-ichikawa/laravel-sendgrid-driver (2.0.5): Downloading (100%)
s-ichikawa/laravel-sendgrid-driver suggests installing s-ichikawa/sendgrid-api-builder (support to build json for sendgrid api)
Writing lock file
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoLoadDump
> @php artisan package:discover --ansi
Discovered Package: beyondcode/laravel-dump-server
Discovered Package: fideloper/proxy
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Discovered Package: s-ichikawa/laravel-sendgrid-driver
Package manifest generated successfully.

Beto@Beto-PC /cygdrive/c/laragon/www/laravel
$ |

```

## Configurando o arquivo .env

### .env

```

#MAIL_DRIVER=sendgrid
#SENDGRID_API_KEY='YOUR_SENDGRID_API_KEY'

```

```

MAIL_DRIVER=smtp
MAIL_HOST=smtp.mailtrap.io

```

```

MAIL_PORT=2525
MAIL_USERNAME=f7f9b1c636f954
MAIL_PASSWORD=c6263786f720f6
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS='contato@oriontecinfo.com.br'
MAIL_FROM_NAME='Roberto Pinheiro'

```

Na conta do sendgrid, clique em **Settings -> API Keys**. Clique no botão **Create API Key** e insira o número gerado no campo **SENDGRID\_API\_KEY**.

Obs: Em modo local vamos utilizar **mailtrap** e em modo de produção vamos usar **sendgrip**

## config/services.php

Insira as linhas a seguir:

```
<?php
```

```
return [  
  
    'mailgun' => [  
        'domain' => env('MAILGUN_DOMAIN'),  
        'secret' => env('MAILGUN_SECRET'),  
        'endpoint' => env('MAILGUN_ENDPOINT', 'api.mailgun.net'),  
    ],  
  
    'postmark' => [  
        'token' => env('POSTMARK_TOKEN'),  
    ],  
  
    'sendgrid' => [  
        'api_key' => env('SENDGRID_API_KEY'),  
    ],  
  
    'ses' => [  
        'key' => env('AWS_ACCESS_KEY_ID'),  
        'secret' => env('AWS_SECRET_ACCESS_KEY'),  
        'region' => env('AWS_DEFAULT_REGION', 'us-east-1'),  
    ],  
  
    'sparkpost' => [  
        'secret' => env('SPARKPOST_SECRET'),  
    ],  
  
    'stripe' => [  
        'model' => App\User::class,  
        'key' => env('STRIPE_KEY'),  
        'secret' => env('STRIPE_SECRET'),  
        'webhook' => [  
            'secret' => env('STRIPE_WEBHOOK_SECRET'),  
            'tolerance' => env('STRIPE_WEBHOOK_TOLERANCE', 300),  
        ],  
    ],  
];
```