

ENSINO MÉDIO INTEGRADO - INFORMÁTICA

Disciplina de Desenvolvimento Web

Aula 07: Framework Laravel - Validação de Formulários

Gil Eduardo de Andrade

Conceitos Preliminares

(<https://laravel.com/docs/8.x>)

Validação - Laravel

O Framework Laravel fornece diferentes abordagens para validação dos dados de entrada da aplicação. Normalmente, é utilizado o método “*validate*”, disponibilizado em todas as requisições HTTP.

O Laravel inclui, também, uma ampla variedade de regras de validação que podem ser aplicadas aos dados de entrada da aplicação, como, por exemplo, verificar se uma dada entrada é única na tabela do banco de dados.



Codificação – Laravel / Validação

(<https://laravel.com/docs/8.x/validation>)

Observação: nesta aula serão utilizadas: *Controller*, *Model* e *Views* disponibilizadas na aula anterior, para o item “Endereços”

Validação de Formulário - Método “*validate()*”

(Arquivo: “/app/Http/Controllers/EnderecoController.php”)

```
...  
public function store(Request $request) {
```

```
$request->validate([
    'rua' => 'required',
    'numero' => 'required',
    'cep' => 'required'
]);

Endereco::create([
    'rua' => mb_strtoupper($request->rua, 'UTF-8'),
    'numero' => $request->numero,
    'cep' => $request->cep,
]);

return redirect()->route('enderecos.index');
}
...
```

Arquivo de Controle - Endereço / Método “store()”

[required] - indica que o campo é de preenchimento obrigatório
[validate()] - método da Classe Request que recebe um array de validações

(Arquivo-fonte: 07-Validate/app/Http/Controller/EnderecoController.php)

(Arquivo: “/resources/views/enderecos/create.blade.php”)

```
...

@if(isset($errors))

    @foreach($errors->all() as $erro)

        <div class="alert alert-danger">

            {{ $erro }}

        </div>

    @endforeach
```

@endif

...

Arquivo de Visualização - Endereço / Create

[\$errors->all()] - objeto com todos os erros de validação indicados pelo validate()

(Arquivo-fonte: 07-Validate/resources/views/enderecos/create.blade.php)

Novo Endereço

Logradouro

Número

CEP



Voltar

Confirmar



The rua field is required.

The numero field is required.

The cep field is required.

(Execução da codificação anterior)

Conjunto de Validações

(Arquivo: “/app/Http/Controllers/EnderecoController.php”)

```
...  
  
public function store(Request $request) {  
  
    $request->validate([  
  
        'rua' => 'required|max:100|min:10',  
  
        'numero' => 'required|max:7|min:1',  
  
        'cep' => 'required|max:12|min:9|unique:enderecos'  
  
    ]);  
  
    Endereco::create([  
  
        'rua' => mb_strtoupper($request->rua, 'UTF-8'),  
  
        'numero' => $request->numero,  
  
        'cep' => $request->cep,  
  
    ]);  
  
    return redirect()->route('enderecos.index');  
  
}  
  
...
```

Arquivo de Controle - Endereço / Método “store()”

[max | min] - indica, respectivamente, o tamanho máximo e mínimo do campo
[unique:enderecos] - indica que é um campo de registro único na tabela

(Arquivo-fonte: 07-Validate/app/Http/Controller/EnderecoController.php)

Novo Endereço

Logradouro
Rua João

Número
10987650

CEP
83203

← Voltar

Confirmar ✓

The rua must be at least 10 characters.

The numero must not be greater than 7 characters.

The cep must be at least 9 characters.

(Execução da codificação anterior)

Customizando as Mensagens de Erro

(Arquivo: "/app/Http/Controllers/EnderecoController.php")

```
...  
  
public function store(Request $request) {  
  
    $msgs = [  
  
        "rua.required" => "O preenchimento do campo [logradouro] é obrigatório!",  
  
        "rua.max" => "O campo [logradouro] possui tamanho máximo de 100 caracteres!",  
  
        "rua.min" => "O campo [logradouro] possui tamanho mínimo de 10 caracteres!",  
  
        "numero.required" => "O preenchimento do campo [número] é obrigatório!",  
  
    ]  
  
}
```


Novo Endereço

Logradouro
Rua João

Número
10987650

CEP
83203



Voltar

Confirmar



O campo [logradouro] possui tamanho mínimo de 10 caracteres!

O campo [número] possui tamanho máximo de 14 caracteres!

O campo [CEP] possui tamanho mínimo de 13 caracteres!

(Execução da codificação anterior)

Tornando as Mensagens de Erro Genéricas

(Arquivo: "/app/Http/Controllers/EnderecoController.php")

```
...  
  
public function store(Request $request) {  
  
    $regras = [  
  
        'rua' => 'required|max:100|min:10',  
  
        'numero' => 'required|max:7|min:1',  
  
        'cep' => 'required|max:12|min:9|unique:enderecos'  
  
    ];  
  
    $msgs = [  

```

```
"required" => "O preenchimento do campo [:attribute] é obrigatório!",  
"max" => "O campo [:attribute] possui tamanho máximo de [:max] caracteres!",  
"min" => "O campo [:attribute] possui tamanho mínimo de [:min] caracteres!",  
"unique" => "Já existe um endereço cadastrado com esse [:attribute]!"  
  
];  
  
$request->validate($regras, $msgs);  
}  
...
```

Arquivo de Controle - Endereço / Método "store()"

[:attribute] - possui o nome do atributo/campo que contém o erro de validação
[:max | :min] - contém o valor especificado para a regra de validação "max" e "min"

(Arquivo-fonte: 07-Validate/app/Http/Controller/EnderecoController.php)

Novo Endereço

Logradouro
Rua João

Número
10987650

CEP
83203

← Voltar

Confirmar ✓

O campo [rua] possui tamanho mínimo de [10] caracteres!

O campo [numero] possui tamanho máximo de [7] caracteres!

O campo [cep] possui tamanho mínimo de [9] caracteres!

(Execução da codificação anterior)

Customizando as Apresentação dos Erros na View

(Arquivo: "/resources/views/enderecos/create.blade.php")

```
...  
  
<div class="row">  
  <div class="col" >  
    <div class="form-floating mb-3">  
      <input  
        type="text"  
        class="form-control {{ $errors->has('rua') ? 'is-invalid' : '' }}"
```

```
        name="rua"

        placeholder="rua"

        value="{{old('rua')}}"

    />

    @if($errors->has('rua'))

        <div class='invalid-feedback'>

            {{ $errors->first('rua') }}

        </div>

    @endif

    <label for="rua">Logradouro</label>

</div>

</div>

</div>

<div class="row">

    <div class="col" >

        <div class="form-floating mb-3">

            <input

                type="number"

                class="form-control {{ $errors->has('numero') ? 'is-invalid' : ''

                }}"

                name="numero"

                placeholder="Número"

                value="{{old('numero')}}"

            />

            @if($errors->has('numero'))

                <div class='invalid-feedback'>

                    {{ $errors->first('numero') }}

                </div>

            @endif

        </div>

    </div>

</div>

</div>
```

```
@endif

<label for="numero">Número</label>

</div>

</div>

</div>

<div class="row">

  <div class="col" >

    <div class="form-floating mb-3">

      <input

        type="number"

        class="form-control {{ $errors->has('cep') ? 'is-invalid' : '' }}"

        name="cep"

        placeholder="Número"

        value="{{old('cep')}}"

      />

      @if($errors->has('cep'))

        <div class='invalid-feedback'>

          {{ $errors->first('cep') }}

        </div>

      @endif

      <label for="cep">CEP</label>

    </div>

  </div>

</div>

...


```

Arquivo de Visualização - Endereço / Create

[? 'is-invalid' : ''] - operador ternário de condição

```
[? 'is-invalid' : ''] - se verdadeiro (?) aplica a class 'is-invalid' do bootstrap  
[? 'is-invalid' : ''] - se falso (:) não aplica a class  
[$errors->first('cep')] - obtém o primeiro erro identificado para o campo "cep"
```

(Arquivo-fonte: 07-Validate/resources/views/enderecos/create.blade.php)

Novo Endereço

Logradouro
Rua João



O campo [rua] possui tamanho mínimo de [10] caracteres!

Número
10987650





O campo [numero] possui tamanho máximo de [7] caracteres!

CEP
83203



O campo [cep] possui tamanho mínimo de [9] caracteres!

 Voltar

Confirmar 

(Execução da codificação anterior)