

Atividade Prática 5.5 - Construção do Controller, Rotas e Views para a Entidade Book

Introdução e Objetivo

A entidade **Book** possui relacionamentos 1 para N com as entidades **Publisher**, **Author** e **Category**. Para fins didáticos, implementaremos duas versões da operação de criação:

1. **Create com Input de ID:** Os relacionamentos serão definidos manualmente, digitando os IDs das entidades relacionadas (**Publisher**, **Author**, **Category**).
2. **Create com Input Select:** Os relacionamentos serão definidos de forma mais intuitiva, utilizando elementos **select** no formulário para selecionar as entidades relacionadas.

As operações terão **rotas e actions distintas**, garantindo que ambas as abordagens possam ser implementadas e testadas separadamente.

Passo 1: Configurar Rotas e Actions

Rotas

Adicione as rotas no arquivo **routes/web.php** para diferenciar as versões de criação:

```
use App\Http\Controllers\BookController;

// Rotas para criação de livros
Route::get('/books/create-id-number', [BookController::class,
'createWithId'])->name('books.create.id');
Route::post('/books/create-id-number', [BookController::class,
'storeWithId'])->name('books.store.id');

Route::get('/books/create-select', [BookController::class,
'createWithSelect'])->name('books.create.select');
Route::post('/books/create-select', [BookController::class,
'storeWithSelect'])->name('books.store.select');

// Rotas RESTful para index, show, edit, update, delete (tem que ficar depois das
rotas /books/create-id-number e /books/create-select)
Route::resource('books', BookController::class)->except(['create', 'store']);
```

Actions no Controller

Crie ou ajuste o **BookController** para incluir as actions necessárias para as duas versões de criação:

```
namespace App\Http\Controllers;

use App\Models\Book;
use App\Models\Publisher;
use App\Models\Author;
use App\Models\Category;
use Illuminate\Http\Request;

class BookController extends Controller
{
    // Formulário com input de ID
    public function createWithId()
    {
        return view('books.create-id');
    }

    // Salvar livro com input de ID
    public function storeWithId(Request $request)
    {
        $request->validate([
            'title' => 'required|string|max:255',
            'publisher_id' => 'required|exists:publishers,id',
            'author_id' => 'required|exists:authors,id',
            'category_id' => 'required|exists:categories,id',
        ]);

        Book::create($request->all());

        return redirect()->route('books.index')->with('success', 'Livro criado com sucesso.');
```

```
    }

    // Formulário com input select
    public function createWithSelect()
    {
        $publishers = Publisher::all();
        $authors = Author::all();
        $categories = Category::all();

        return view('books.create-select', compact('publishers', 'authors', 'categories'));
    }

    // Salvar livro com input select
```

```
public function storeWithSelect(Request $request)
{
    $request->validate([
        'title' => 'required|string|max:255',
        'publisher_id' => 'required|exists:publishers,id',
        'author_id' => 'required|exists:authors,id',
        'category_id' => 'required|exists:categories,id',
    ]);

    Book::create($request->all());

    return redirect()->route('books.index')->with('success', 'Livro criado com
sucesso.');
```

Passo 2: Criar as Views

View: `create-id.blade.php`

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

@section('content')
<div class="container">
    <h1 class="my-4">Adicionar Livro (Com ID)</h1>

    <form action="{{ route('books.store.id') }}" method="POST">
        @csrf
        <div class="mb-3">
            <label for="title" class="form-label">Título</label>
            <input type="text" class="form-control @error('title') is-invalid
@enderror" id="title" name="title" required>
            @error('title')
                <div class="invalid-feedback">
                    {{ $message }}
                </div>
            @enderror
        </div>

        <div class="mb-3">
            <label for="publisher_id" class="form-label">ID da Editora</label>
            <input type="number" class="form-control @error('publisher_id')
is-invalid @enderror" id="publisher_id" name="publisher_id" required>
            @error('publisher_id')
                <div class="invalid-feedback">
                    {{ $message }}
                </div>
            @enderror
        </div>

        <div class="mb-3">
            <label for="author_id" class="form-label">ID do Autor</label>
            <input type="number" class="form-control @error('author_id')
is-invalid @enderror" id="author_id" name="author_id" required>
            @error('author_id')
                <div class="invalid-feedback">
                    {{ $message }}
                </div>
            @enderror
        </div>

        <div class="mb-3">
            <label for="category_id" class="form-label">ID da Categoria</label>
            <input type="number" class="form-control @error('category_id')
is-invalid @enderror" id="category_id" name="category_id" required>
            @error('category_id')
                <div class="invalid-feedback">
```

```

                {{ $message }}
            </div>
        @enderror
    </div>

    <button type="submit" class="btn btn-success">Salvar</button>
</form>
</div>
@endsection

```

View: `create-select.blade.php`

```

@extends('layouts.app')

@section('content')
<div class="container">
    <h1 class="my-4">Adicionar Livro (Com Select)</h1>

    <form action="{{ route('books.store.select') }}" method="POST">
        @csrf
        <div class="mb-3">
            <label for="title" class="form-label">Título</label>
            <input type="text" class="form-control @error('title') is-invalid
@enderror" id="title" name="title" required>
            @error('title')
                <div class="invalid-feedback">
                    {{ $message }}
                </div>
            @enderror
        </div>

        <div class="mb-3">
            <label for="publisher_id" class="form-label">Editora</label>
            <select class="form-select @error('publisher_id') is-invalid
@enderror" id="publisher_id" name="publisher_id" required>
                <option value="" selected>Selecione uma editora</option>
                @foreach($publishers as $publisher)
                    <option value="{{ $publisher->id }}">{{ $publisher->name
}}</option>
                @endforeach
            </select>
            @error('publisher_id')
                <div class="invalid-feedback">
                    {{ $message }}
                </div>
            @enderror
        </div>
    </form>
</div>

```

```

        </div>
    @enderror
</div>

<div class="mb-3">
    <label for="author_id" class="form-label">Autor</label>
    <select class="form-select @error('author_id') is-invalid @enderror"
id="author_id" name="author_id" required>
        <option value="" selected>Selezione um autor</option>
        @foreach($authors as $author)
            <option value="{{ $author->id }}">{{ $author->name }}</option>
        @endforeach
    </select>
    @error('author_id')
        <div class="invalid-feedback">
            {{ $message }}
        </div>
    @enderror
</div>

<div class="mb-3">
    <label for="category_id" class="form-label">Categoria</label>
    <select class="form-select @error('category_id') is-invalid @enderror"
id="category_id" name="category_id" required>
        <option value="" selected>Selezione uma categoria</option>
        @foreach($categories as $category)
            <option value="{{ $category->id }}">{{ $category->name
    }}</option>
        @endforeach
    </select>
    @error('category_id')
        <div class="invalid-feedback">
            {{ $message }}
        </div>
    @enderror
</div>

    <button type="submit" class="btn btn-success">Salvar</button>
</form>
</div>
@endsection

```

Passo 3: Implementar a Operação de Edit para a Entidade Book

A operação de **Edit** será feita apenas com a versão utilizando `select`. O nome padrão da rota e da action será mantido.

Controller: Método Edit

Adicione o método `edit` no `BookController` para carregar as informações necessárias para a edição de um livro:

```
public function edit(Book $book)
{
    $publishers = Publisher::all();
    $authors = Author::all();
    $categories = Category::all();

    return view('books.edit', compact('book', 'publishers', 'authors',
    'categories'));
}
```

Controller: Método Update

Adicione o método `update` no `BookController` para salvar as alterações feitas no livro:

```
public function update(Request $request, Book $book)
{
    $request->validate([
        'title' => 'required|string|max:255',
        'publisher_id' => 'required|exists:publishers,id',
        'author_id' => 'required|exists:authors,id',
        'category_id' => 'required|exists:categories,id',
    ]);

    $book->update($request->all());

    return redirect()->route('books.index')->with('success', 'Livro atualizado com
    sucesso.');
```

View: `edit.blade.php`

Crie o arquivo `resources/views/books/edit.blade.php` para renderizar o formulário de edição com inputs `select`:

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

@section('content')
<div class="container">
    <h1 class="my-4">Editar Livro</h1>

    <form action="{{ route('books.update', $book) }}" method="POST">
        @csrf
        @method('PUT')
        <div class="mb-3">
            <label for="title" class="form-label">Título</label>
            <input type="text" class="form-control @error('title') is-invalid
@enderror" id="title" name="title" value="{{ old('title', $book->title) }}"
required>
            @error('title')
                <div class="invalid-feedback">
                    {{ $message }}
                </div>
            @enderror
        </div>

        <div class="mb-3">
            <label for="publisher_id" class="form-label">Editora</label>
            <select class="form-select @error('publisher_id') is-invalid
@enderror" id="publisher_id" name="publisher_id" required>
                <option value="" disabled>Selecione uma editora</option>
                @foreach($publishers as $publisher)
                    <option value="{{ $publisher->id }}" {{ $publisher->id ==
$book->publisher_id ? 'selected' : '' }}>
                        {{ $publisher->name }}
                    </option>
                @endforeach
            </select>
            @error('publisher_id')
                <div class="invalid-feedback">
                    {{ $message }}
                </div>
            @enderror
        </div>

        <div class="mb-3">
            <label for="author_id" class="form-label">Autor</label>
            <select class="form-select @error('author_id') is-invalid @enderror"
id="author_id" name="author_id" required>
                <option value="" disabled>Selecione um autor</option>
```



```

        @foreach($authors as $author)
            <option value="{{ $author->id }}" {{ $author->id ==
$book->author_id ? 'selected' : '' }}>
                {{ $author->name }}
            </option>
        @endforeach
    </select>
    @error('author_id')
        <div class="invalid-feedback">
            {{ $message }}
        </div>
    @enderror
</div>

<div class="mb-3">
    <label for="category_id" class="form-label">Categoria</label>
    <select class="form-select @error('category_id') is-invalid @enderror"
id="category_id" name="category_id" required>
        <option value="" disabled>Selecione uma categoria</option>
        @foreach($categories as $category)
            <option value="{{ $category->id }}" {{ $category->id ==
$book->category_id ? 'selected' : '' }}>
                {{ $category->name }}
            </option>
        @endforeach
    </select>
    @error('category_id')
        <div class="invalid-feedback">
            {{ $message }}
        </div>
    @enderror
</div>

    <button type="submit" class="btn btn-success">Atualizar</button>
    <a href="{{ route('books.index') }}" class="btn
btn-secondary">Cancelar</a>
</form>
</div>
@endsection

```

Passo 4: Implementar a Operação Show para a Entidade Book

Neste passo, implementaremos a operação de visualização (**show**) para a entidade **Book**, exibindo os detalhes do livro, incluindo os nomes do autor, editora e categoria, com links para a ação **show** das entidades relacionadas.

Controller: Método Show

Adicione o método **show** no **BookController** para carregar os dados do livro e exibir na view:

```
public function show(Book $book)
{
    // Carregando autor, editora e categoria do livro com eager loading
    $book->load(['author', 'publisher', 'category']);

    return view('books.show', compact('book'));
}
```

Explicação do Eager Loading

O método **load** carrega os relacionamentos especificados (**author**, **publisher**, **category**) junto com a entidade principal (**book**) em uma única query, reduzindo o número de acessos ao banco de dados.

View: **show.blade.php**

Crie o arquivo **resources/views/books/show.blade.php** para renderizar os detalhes do livro:

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

@section('content')
<div class="container">
    <h1 class="my-4">Detalhes do Livro</h1>

    <div class="card">
        <div class="card-header">
            <strong>Título:</strong> {{ $book->title }}
        </div>
        <div class="card-body">
            <p><strong>Autor:</strong>
                <a href="{{ route('authors.show', $book->author->id) }}">
                    {{ $book->author->name }}
                </a>
            </p>
        </div>
    </div>
</div>
```

```
        </a>
    </p>
    <p><strong>Editora:</strong>
        <a href="{{ route('publishers.show', $book->publisher->id) }}">
            {{ $book->publisher->name }}
        </a>
    </p>
    <p><strong>Categoria:</strong>
        <a href="{{ route('categories.show', $book->category->id) }}">
            {{ $book->category->name }}
        </a>
    </p>
</div>
</div>

<a href="{{ route('books.index') }}" class="btn btn-secondary mt-3">
    <i class="bi bi-arrow-left"></i> Voltar
</a>
</div>
@endsection
```

Passo 5: Listagem de Livros

Neste passo, criaremos a listagem de livros (`index`) que exibirá o **ID**, **título**, **nome do autor** e botões de ações para **visualizar**, **editar** e **deletar**. O controlador será ajustado para realizar **eager loading** do relacionamento com a entidade `Author`.

Controller: Método Index com Eager Loading

Adicione ou modifique o método `index` no `BookController` para carregar os dados do relacionamento `author` com **eager loading**:

```
public function index()
{
    // Carregar os livros com autores usando eager loading e paginação
    $books = Book::with('author')->paginate(20);

    return view('books.index', compact('books'));
}
```

View: `index.blade.php`

Crie ou ajuste o arquivo `resources/views/books/index.blade.php` para exibir os livros em formato de tabela com os botões de ação:

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

@section('content')
<div class="container">
    <h1 class="my-4">Lista de Livros</h1>

    @if(session('success'))
        <div class="alert alert-success">
            {{ session('success') }}
        </div>
    @endif

    <a href="{{ route('books.create.id') }}" class="btn btn-success mb-3">
        <i class="bi bi-plus"></i> Adicionar Livro (Com ID)
    </a>
    <a href="{{ route('books.create.select') }}" class="btn btn-primary mb-3">
        <i class="bi bi-plus"></i> Adicionar Livro (Com Select)
    </a>
</div>
```

```

<table class="table table-striped">
  <thead>
    <tr>
      <th>ID</th>
      <th>Título</th>
      <th>Autor</th>
      <th>Ações</th>
    </tr>
  </thead>
  <tbody>
    @forelse($books as $book)
      <tr>
        <td>{{ $book->id }}</td>
        <td>{{ $book->title }}</td>
        <td>{{ $book->author->name }}</td>
        <td>
          <!-- Botão de Visualizar -->
          <a href="{{ route('books.show', $book->id) }}" class="btn
btn-info btn-sm">
            <i class="bi bi-eye"></i> Visualizar
          </a>

          <!-- Botão de Editar -->
          <a href="{{ route('books.edit', $book->id) }}" class="btn
btn-primary btn-sm">
            <i class="bi bi-pencil"></i> Editar
          </a>

          <!-- Botão de Deletar -->
          <form action="{{ route('books.destroy', $book->id) }}"
method="POST" style="display: inline;">
            @csrf
            @method('DELETE')
            <button class="btn btn-danger btn-sm" onclick="return
confirm('Deseja excluir este livro?')">
              <i class="bi bi-trash"></i> Deletar
            </button>
          </form>
        </td>
      </tr>
    @empty
      <tr>
        <td colspan="4">Nenhum livro encontrado.</td>
      </tr>
    @endforelse
  </tbody>
</table>

<!-- Controles de Paginação -->
<div class="d-flex justify-content-center">

```

```
        {{ $books->links() }}  
    </div>  
</div>  
@endsection
```

Configurar o Template de Paginação do Bootstrap

Adicione a seguinte linha no método `boot` do arquivo `App\Providers\AppServiceProvider.php` para que o Laravel utilize o template de paginação do Bootstrap. A função `Paginator::useBootstrap()` instrui o Laravel a gerar a paginação com as classes de estilo do Bootstrap, o que corrige o problema de layout.

```
use Illuminate\Pagination\Paginator;  
  
public function boot()  
{  
    Paginator::useBootstrap();  
}
```

Depois de salvar o arquivo, limpe o cache da configuração:

```
php artisan config:clear
```