Rotas e View

Para que o sistema de ordenação funcione corretamente em um módulo, é necessário definir duas **rotas principais** responsáveis por exibir e processar a ordenação.
Essas rotas devem ser criadas dentro do **grupo de rotas do módulo** e vinculadas ao seu respectivo **controller**.

Rotas

Os módulos que implementarem a ordenação precisam conter as seguintes rotas:

Descrição das rotas:

Método	Caminho	Controller / Método	Descrição
GET	**/ordenar**	**order()**	Responsável por carregar a view principal da ordenação e inicializar o componente com base nas configurações definidas no controller.
POST	**/ordenar**	**processOrder()**	Recebe a nova ordem enviada pelo frontend e processa a atualização dos registros no banco de dados.

Ambas as rotas utilizam o middleware "*panel.check.module.permission:update*", garantindo que apenas usuários com permissão de **edição** possam acessar e modificar a ordem dos registros.

Exemplo:

```
Route::get('/ordernar', [AuthorController::class, 'order'])
->name('order')->middleware('panel.check.module.permission:update');


Route::post('/ordernar', [AuthorController::class, 'processOrder'])
->name('processOrder')->middleware('panel.check.module.permission:update');
```

View

Por padrão, o sistema de ordenação utiliza uma **view global** compartilhada entre todos os módulos, localizada em:

"panel.layouts.modules.order"

Essa view é responsável por renderizar a interface do sistema de ordenação, exibindo os registros conforme as configurações definidas no método **orderConfig()** do controller.
Ela já contém toda a estrutura necessária para:

- Exibir os níveis de ordenação e breadcrumbs;
- Listar os registros conforme o nível atual;
- Enviar a nova ordem para o backend através da rota **processOrder**;
- Atualizar dinamicamente a interface conforme o usuário interage com os níveis.

Personalização da View

Embora exista uma view global, o sistema permite **substituí-la por uma view personalizada**, caso o módulo necessite de um layout ou comportamento específico.
Para isso, basta informar o **caminho da nova view** dentro da configuração **$generalConfig** do método **orderConfig()**.

Exemplo:

```
$generalConfig = [ 'selfGroupedRecords' => false, 'view' => 'panel.modules.produtos.orderCustom', ];
```

Nesse caso, o sistema utilizará a view "*panel.modules.produtos.orderCustom*" em vez da padrão "*panel.layouts.modules.order*".

> **Observação:**
> A view personalizada deve seguir a mesma estrutura básica da view padrão — ou seja, deve conter o componente responsável por interagir com o sistema de ordenação e enviar as alterações para o backend.
> Dessa forma, mantém-se a compatibilidade com a lógica existente, mesmo quando o layout é adaptado.

Revisão #5
Criado 13 outubro 2025 12:50:47
Atualizado 22 maio 2026 21:15:39