

## Manual de utilização do BACKPACK e LARAVEL UI

O **Laravel UI** fornece autenticação básica com Blade, enquanto o **Backpack** cuida do painel administrativo. Como o Backpack **não inclui autenticação por padrão**, combinar os dois é uma ótima ideia.

---

### Como integrar Backpack com Laravel UI

Aqui está um guia passo a passo para usar **Laravel UI** e **Backpack** juntos:

#### Instale o Laravel UI

Se ainda não tiver instalado, execute:

```
composer require laravel/ui
```

Agora, gere a autenticação com Bootstrap:

```
php artisan ui bootstrap --auth  
npm install && npm run dev
```

Isso criará as rotas de login, registro e a estrutura de autenticação básica.

---

#### Instale o Backpack

Agora, instale o Backpack Admin Panel:

```
composer require backpack/crud  
php artisan backpack:install
```

Isso criará o painel administrativo acessível em /admin.

---

### Configurar a autenticação do Backpack

O Backpack, por padrão, não usa o mesmo sistema de autenticação do Laravel UI. Para unificá-los, faça o seguinte:

**Diga ao Backpack para usar o sistema de autenticação do Laravel UI.** No arquivo config/backpack/base.php, altere:

```
'guard' => 'web', // Mudar de 'backpack' para 'web'
```

Isso fará com que o Backpack use o mesmo sistema de login.

**Adicione a Middleware de Admin**  
Edite app/Http/Kernel.php e dentro de \$routeMiddleware, adicione:

```
'admin' => \App\Http\Middleware\AdminMiddleware::class,
```

Agora, crie esse middleware:

```
php artisan make:middleware AdminMiddleware
```

E edite app/Http/Middleware/AdminMiddleware.php:

```
namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class AdminMiddleware
{
    public function handle(Request $request, Closure $next)
    {
        if (Auth::check() && Auth::user()->is_admin) { // Ajuste conforme seu banco
            return $next($request);
        }
        return redirect('/');
    }
}
```

Isso impede que usuários normais acessem o painel admin.

### Proteger as rotas do Backpack

No arquivo routes/backpack/custom.php, adicione o middleware:

```
Route::group([
    'prefix' => config('backpack.base.route_prefix', 'admin'),
    'middleware' => ['web', 'admin'], // Protege com middleware
    'namespace' => 'App\Http\Controllers\Admin',
], function () {
    // Rotas do Backpack
    Route::crud('users', 'UserCrudController');
});
```

Agora, apenas administradores poderão acessar o painel.

---

### Adicione um campo "is\_admin" na tabela de usuários

Se quiser diferenciar **usuários comuns** de **administradores**, adicione um campo is\_admin na tabela de usuários:

Crie a migration:

```
php artisan make:migration add_is_admin_to_users --table=users
```

No arquivo gerado em database/migrations/, adicione:

```
public function up()
{
    Schema::table('users', function (Blueprint $table) {
        $table->boolean('is_admin')->default(false);
    });
}
```

Execute:

php artisan migrate

Agora, edite app/Models/User.php:

```
public function isAdmin()
{
    return $this->is_admin;
}
```

E pronto! Agora, usuários com is\_admin = 1 poderão acessar o painel do Backpack, e usuários comuns ficarão restritos.

---

## Resumo

Laravel UI cuida da autenticação (login, registro, etc.).

Backpack gerencia o painel administrativo.

Configuramos o Backpack para usar o mesmo sistema de login do Laravel UI.

Criamos uma proteção para que apenas administradores acessem o painel.

Agora seu sistema terá **autenticação Laravel UI + painel Backpack funcionando juntos!**

Precisa de mais alguma coisa?