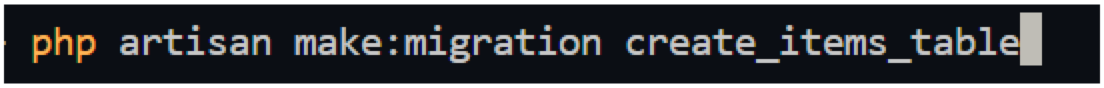
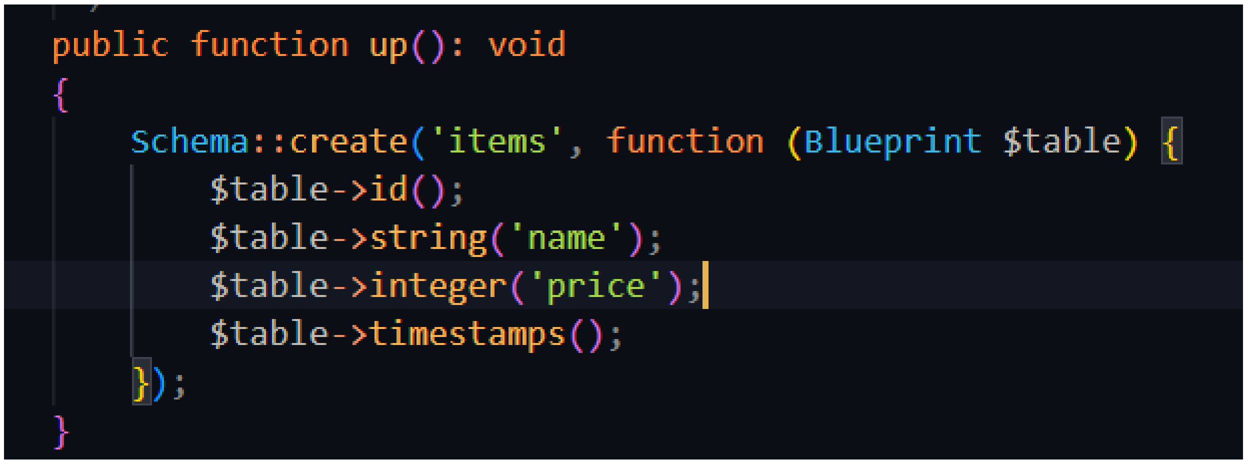
# CRUD Operation

**Step 1:**



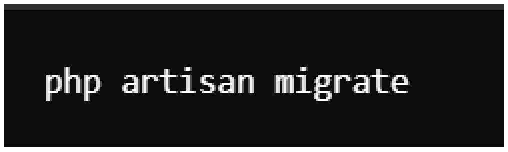
```
php artisan make:migration create_items_table
```

**Step 2: Edit the migration file**



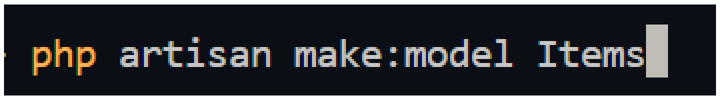
```
public function up(): void
{
    Schema::create('items', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->integer('price');
        $table->timestamps();
    });
}
```

**Step 3: Run the migration to create the table**



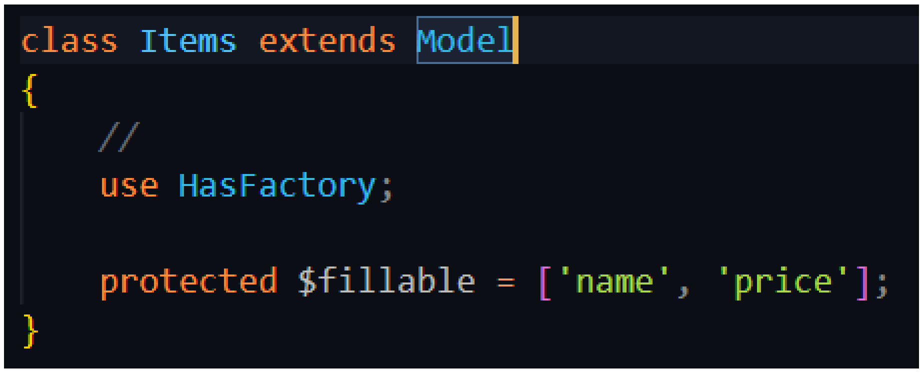
```
php artisan migrate
```

**Step 4: Create a Model**



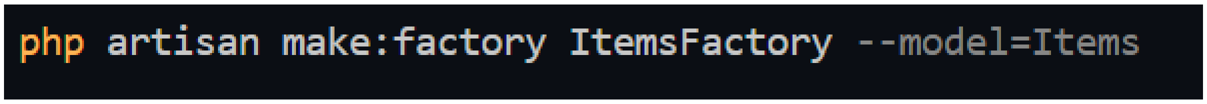
```
php artisan make:model Items
```

**Step 5: Open app/Models/Items.php and define the fillable fields:**

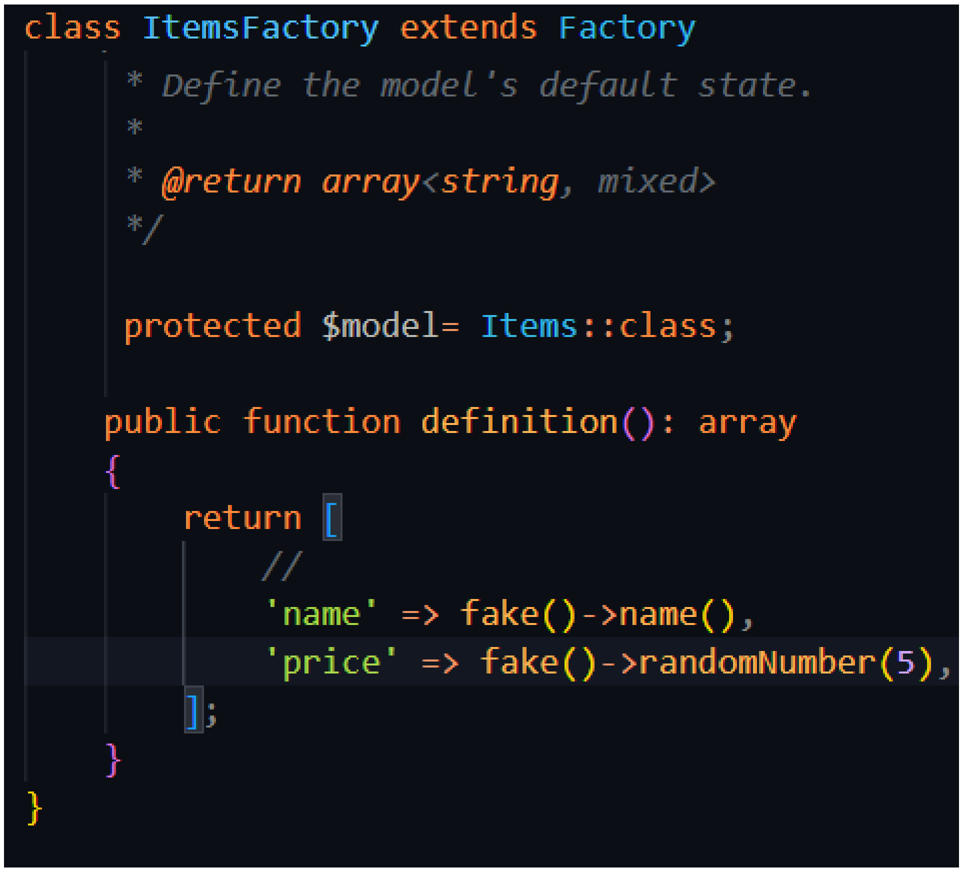


```php
class Items extends Model
{
    //
    use HasFactory;

    protected $fillable = ['name', 'price'];
}
```

**Step 6: Create a Factory Associated with the Model**

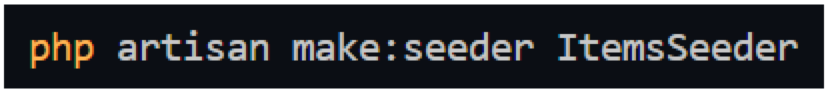


```
php artisan make:factory ItemsFactory --model=Items
```

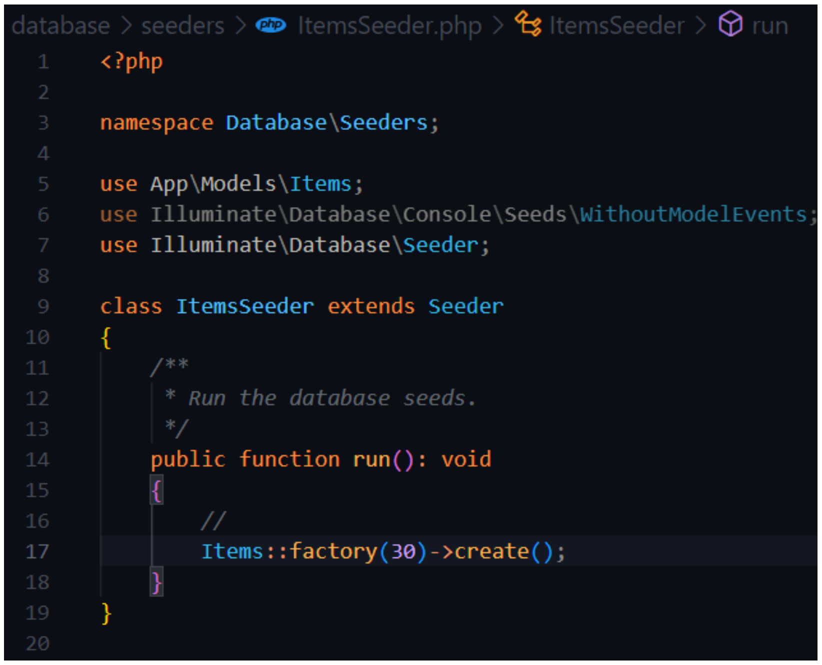


```php
class ItemsFactory extends Factory
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */

     protected $model= Items::class;

    public function definition(): array
    {
        return [
            //
            'name' => fake()->name(),
            'price' => fake()->randomNumber(5),
        ];
    }
}
```

**Step 7: Create a Seeder**

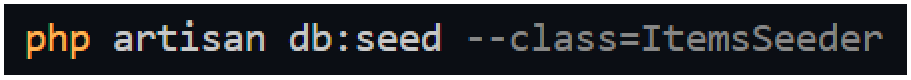


```
php artisan make:seeder ItemsSeeder
```



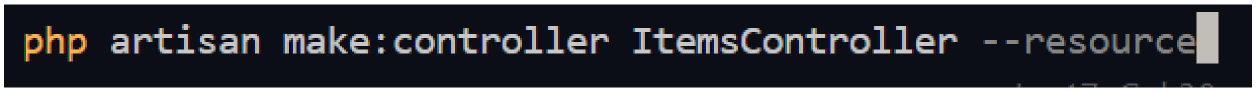
```php
<?php

namespace Database\Seeders;

use App\Models\Items;
use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

class ItemsSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        //
        Items::factory(30)->create();
    }
}

```

**Step 8: Run Seeder**



```
php artisan db:seed --class=ItemsSeeder
```

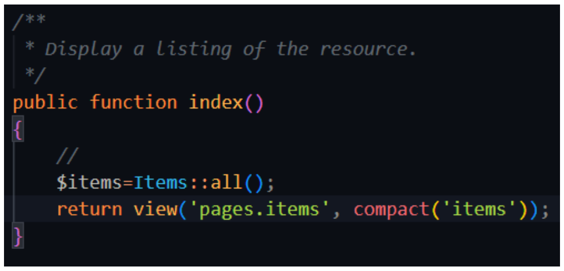
**Step 9: Create the Controller**



```
php artisan make:controller ItemsController --resource
```

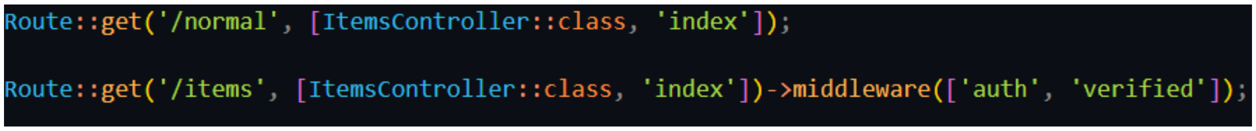
**Step 10: Update the controller**

**Show all items**



```php
/**
 * Display a listing of the resource.
 */
public function index()
{
    //
    $items=Items::all();
    return view('pages.items', compact('items'));
}
```

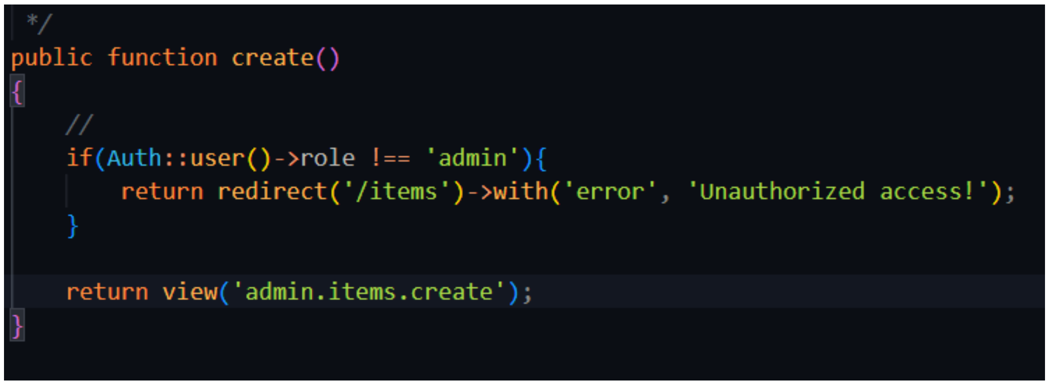
**Routes**



```php
Route::get('/normal', [ItemsController::class, 'index']);

Route::get('/items', [ItemsController::class, 'index'])->middleware(['auth', 'verified']);
```

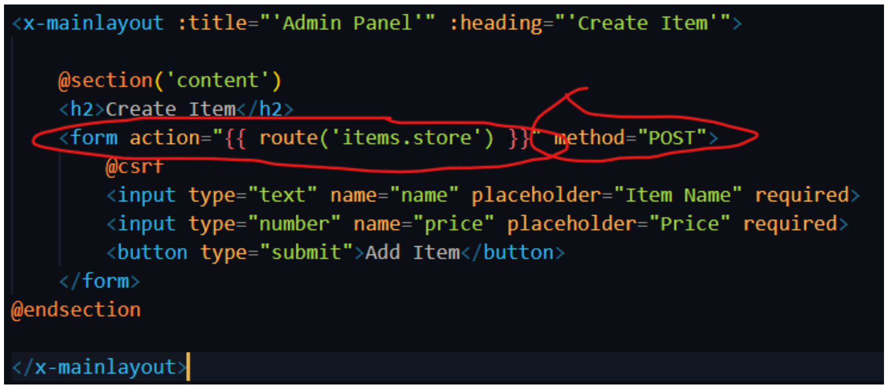
**to access with '/items' user must be authenticated and verified**

**Creating new item**



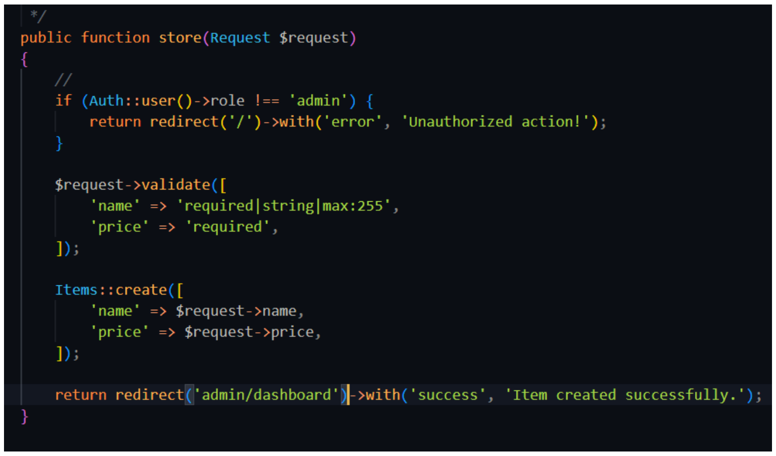
```php
 */
public function create()
{
    //
    if(Auth::user()->role !== 'admin'){
        return redirect('/items')->with('error', 'Unauthorized access!');
    }

    return view('admin.items.create');
}
```

**Blade View**



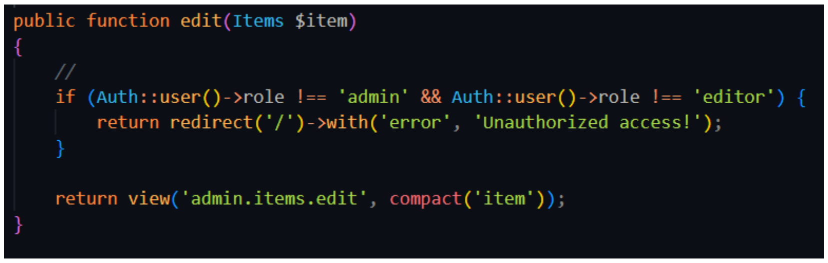
```html
<x-mainlayout :title="'Admin Panel'" :heading="'Create Item'">

    @section('content')
    <h2>Create Item</h2>
    <form action="{{ route('items.store') }}" method="POST">
        @csrf
        <input type="text" name="name" placeholder="Item Name" required>
        <input type="number" name="price" placeholder="Price" required>
        <button type="submit">Add Item</button>
    </form>
@endsection

</x-mainlayout>
```

## Store newly created item



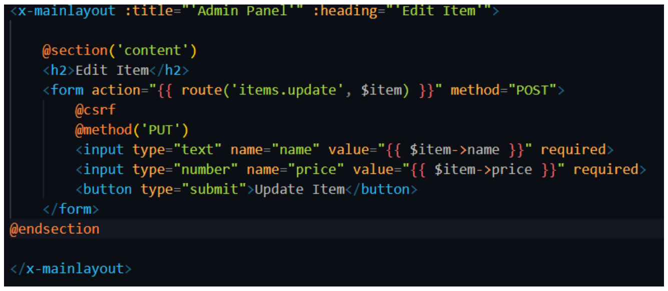
```php
 */
public function store(Request $request)
{
    //
    if (Auth::user()->role !== 'admin') {
        return redirect('/')->with('error', 'Unauthorized action!');
    }

    $request->validate([
        'name' => 'required|string|max:255',
        'price' => 'required',
    ]);

    Items::create([
        'name' => $request->name,
        'price' => $request->price,
    ]);

    return redirect('admin/dashboard')->with('success', 'Item created successfully.');
}
```

## Edit existing item



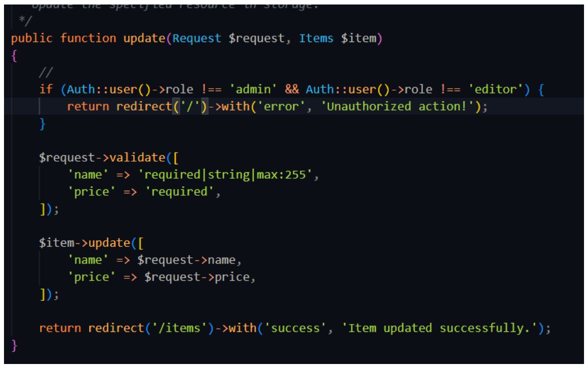
```php
public function edit(Items $item)
{
    //
    if (Auth::user()->role !== 'admin' && Auth::user()->role !== 'editor') {
        return redirect('/')->with('error', 'Unauthorized access!');
    }

    return view('admin.items.edit', compact('item'));
}
```

## Blade view



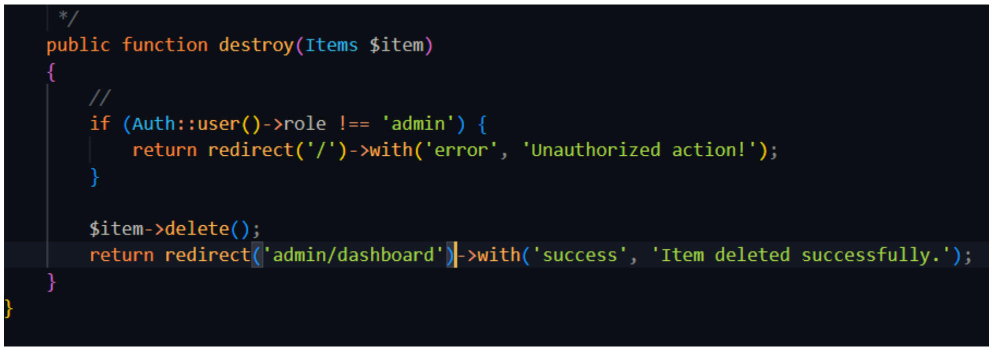
```blade
<x-mainlayout :title="'Admin Panel'" :heading="'Edit Item'">

    @section('content')
    <h2>Edit Item</h2>
    <form action="{{ route('items.update', $item) }}" method="POST">
        @csrf
        @method('PUT')
        <input type="text" name="name" value="{{ $item->name }}" required>
        <input type="number" name="price" value="{{ $item->price }}" required>
        <button type="submit">Update Item</button>
    </form>
@endsection

</x-mainlayout>
```

## Update



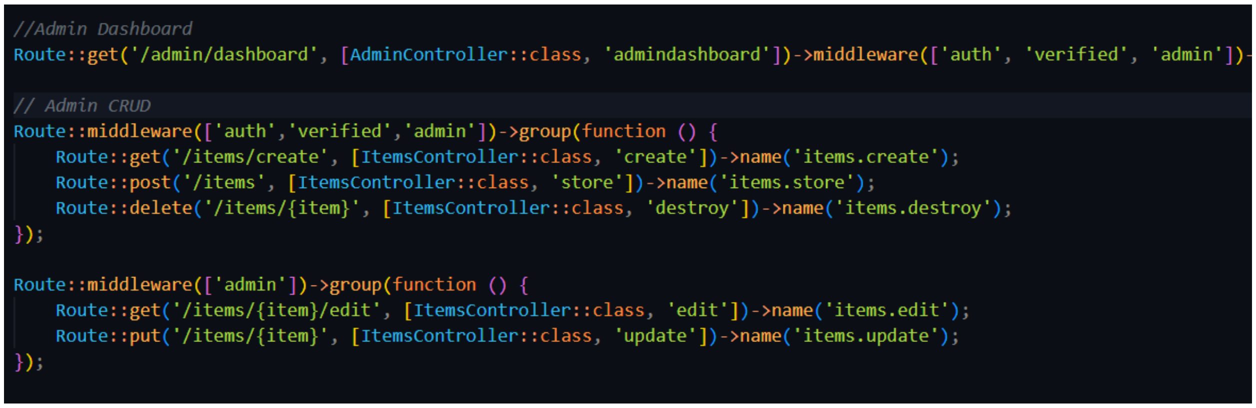
```php
 */
public function update(Request $request, Items $item)
{
    //
    if (Auth::user()->role !== 'admin' && Auth::user()->role !== 'editor') {
        return redirect('/')->with('error', 'Unauthorized action!');
    }

    $request->validate([
        'name' => 'required|string|max:255',
        'price' => 'required',
    ]);

    $item->update([
        'name' => $request->name,
        'price' => $request->price,
    ]);

    return redirect('/items')->with('success', 'Item updated successfully.');
}
```

## Delete



```
     */
    public function destroy(Items $item)
    {
        //
        if (Auth::user()->role !== 'admin') {
            return redirect('/')->with('error', 'Unauthorized action!');
        }

        $item->delete();
        return redirect('admin/dashboard')->with('success', 'Item deleted successfully.');
    }
}
```

## Routes



```
//Admin Dashboard
Route::get('/admin/dashboard', [AdminController::class, 'admindashboard'])->middleware(['auth', 'verified', 'admin'])

// Admin CRUD
Route::middleware(['auth','verified','admin'])->group(function () {
    Route::get('/items/create', [ItemsController::class, 'create'])->name('items.create');
    Route::post('/items', [ItemsController::class, 'store'])->name('items.store');
    Route::delete('/items/{item}', [ItemsController::class, 'destroy'])->name('items.destroy');
});

Route::middleware(['admin'])->group(function () {
    Route::get('/items/{item}/edit', [ItemsController::class, 'edit'])->name('items.edit');
    Route::put('/items/{item}', [ItemsController::class, 'update'])->name('items.update');
});
```