Code Conventions

Dit document beschrijft de coding conventions binnen ons team.

---

1. Algemene Richtlijnen

- Gebruik **camelCase** voor variabelen en methoden.
- Gebruik **PascalCase** voor klassen en modelnamen.
- Gebruik **snake_case** voor database-kolomnamen en configuratiebestanden.
- Gebruik **kebab-case** voor route-namen.
- Gebruik tabs voor indentation

---

2. Bestanden en Mappen Structuur

2.1 Controllers

- Geplaatst in app/Http/Controllers/
- Gebruik **PascalCase** voor controllers (bijv. ProductController.php).
- Methoden in controllers volgen **camelCase** (bijv. getProductList()).
- Gebruik **resource controllers** waar mogelijk.

Voorbeeld:

```
class ProductController extends Controller
{
    public function index() {
        return view('products.index');
    }

    public function show($id) {
        return view('products.show', ['id' => $id]);
    }
}
```

2.2 Models

- Geplaatst in app/Models/
- Modelnamen zijn **enkelvoud** en gebruiken **PascalCase** (bijv. Product.php).
- Gebruik $fillable voor mass assignment bescherming.

Voorbeeld:

```
namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Product extends Model
{
    protected $fillable = ['name', 'price'];
}
```

2.3 Routes

- Geplaatst in routes/web.php of routes/api.php.
- Gebruik **kebab-case** voor route-namen (bijv. products-list).

Voorbeeld:

```
Route::get('/products-list', [ProductController::class, 'index'])->name('products.list');
```

---

3. Naamgevingsconventies

- **Klassen**: PascalCase (bijv. ProductController).
- **Methoden**: camelCase (bijv. getAllProducts()).
- **Variabelen**: camelCase (bijv. $productList).
- **Route namen**: kebab-case (bijv. products-list).
- **Config keys**: snake_case (bijv. database_connections).
- **Database tabellen**: meervoud, snake_case (bijv. products).
- **Database kolommen**: snake_case (bijv. product_name).

---

4. Blade Templates (Views)

- Geplaatst in resources/views/.
- Gebruik **snake_case** voor bestandsnamen (bijv. product_list.blade.php).
- Gebruik Blade directives zoals @foreach, @if en @include.

Voorbeeld:

```
<x-app-layout>
    <h1>{{ $title }}</h1>
    @foreach ($products as $product)
        <p>{{ $product->name }} - €{{ $product->price }}</p>
    @endforeach
</x-app-layout>
```

---

5. Eloquent Best Practices

- Gebruik **mass assignment** met $fillable.
- Gebruik **scopes** voor herbruikbare query-logica.

Voorbeeld:

```
class Product extends Model
{
   protected $fillable = ['name', 'price'];

   public function scopeExpensive($query)
   {
      return $query->where('price', '>', 100);
   }
}
```

---

6. Middleware en Requests

- Middleware bevindt zich in app/Http/Middleware/.
- Request validaties worden uitgevoerd in app/Http/Requests/.

Voorbeeld:

```
class StoreProductRequest extends FormRequest
{
   public function rules()
   {
      return [
         'name' => 'required|string|max:255',
         'price' => 'required|numeric|min:0',
      ];
   }
}
```

---

7. API Best Practices

- API-routes in routes/api.php.
- Gebruik **RESTful principes** (GET, POST, PUT, DELETE).
- Gebruik **resource controllers** voor API's.

Voorbeeld:

```
Route::apiResource('products', ProductController::class);
```