

Laravel Cheat Sheet

Aanmaken nieuw project

1. Navigeer naar de map van je webserver (www of htdocs) in de command line
2. Voer in: `laravel new <projectnaam>`
3. Navigeer naar de aangemaakte folder (`cd <projectnaam>`)
4. Voer in: `php artisan serve`
5. Start laragon of xampp

Belangrijke bestanden/mappen

- `/resources/views/` (alle webpagina's en onderdelen (html) die op de website gebruikt worden)
- `/routes/web.php`
- `/app/http/controllers/` (alle aangemaakte controllers vind je hier)
- `/app/http/models` (alle aangemaakte models vind je hier)
- `.env` file (hier wordt je database credentials en andere persoonlijke configuratie items opgeslagen.

Belangrijke artisan commando's

- **`php artisan serve`** (start een webserver zodat je de site kan bezoeken via localhost:8000)
- **`php artisan make:migration createProductsTable`** (maakt een migration voor een table genaamd 'products')
- **`php artisan make:controller ProductsController`** (maakt een controller aan)
 - **`php artisan make:controller ProductsController --resource`** (maakt een controller aan en geeft je alle lege methodes die nodig zijn voor een volledig crud systeem)
- **`php artisan make:model Product`** (maakt een model aan voor een Product. Deze model synchroniseert dankzij de naamgeving met de tabelnaam 'products')
- **`php artisan migrate`** (voert de migrations uit die nog niet zijn uitgevoerd)
 - **`php artisan migrate:fresh`** (reset alle migrations en voert alles helemaal opnieuw uit)
 - **`php artisan migrate:rollback`** (reset de laatste batch van migrations die uitgevoerd zijn)
 - **`php artisan migrate:status`** (laat een log zien van alle migrations en welke er al gerund zijn, en in welke batch deze gerund zijn)
- **`php artisan route:list`** (geeft je een overzicht van alle routes die momenteel in je applicatie beschikbaar zijn.

Code voorbeelden:

Hier zijn een aantal voorbeelden van routes die je tegen kunt komen in je laravel applicatie:

```
// voorbeeld van een route om een lijst van producten te tonen
Route::get("products", [ProductController::class, "index"])
    ->name('products.index');

// voorbeeld van een route om een enkel product te tonen
Route::get("products/{id}", [ProductController::class, "show"])
    ->name('products.show');

// voorbeeld van een route om naar een edit form te gaan
Route::get('products/{id}/edit', [ProductController::class, 'edit'])
    ->name('products.edit');

// voorbeeld van een route om naar een create form te gaan
Route::get('products/create', [ProductController::class, 'create'])
    ->name('products.create');

// voorbeeld van een route om een product te verwijderen
Route::delete('products/{id}', [ProductController::class, 'destroy'])
    ->name('products.destroy');

// voorbeeld van een route om een product op te slaan
Route::post("products", [ProductController::class, "store"])
    ->name('products.store');

// voorbeeld van een route om een product te wijzigen
Route::put("products/{id}", [ProductController::class, "update"])
    ->name('products.update');
```

Een shortcut om al deze routes in één keer te registreren:

```
Route::resource('products', 'ProductController');
```

Een voorbeeld van een create form die kan horen bij bovenstaande route (products.store)

```
<form action="{ route('products.store') }}" method="post">
    @csrf
    <div class="form-group">
        <label for="">name</label>
        <input type="text" name="name" id="">
    </div>

    <div class="form-group">
        <label for="">price</label>
        <input type="text" name="price" id="">
    </div>

    <input type="submit" value="Product opslaan">
</form>
```

Een voorbeeld van een delete form die kan horen bij bovenstaande route (products.destroy)

```
<form action="{{ route('products.destroy', $product->id) }}" method="post">
  @method('delete')
  @csrf
  <input type="submit" value="Product verwijderen">
</form>
```

Een voorbeeld van een overzicht om alle producten te laten zien:

```
<div class="products">
  @foreach($products as $product)
    <div class="product">
      <h3>{{ $product->name }}</h3>
      <p>{{ $product->price }}</p>
    </div>
  @endforeach
</div>
```

Een voorbeeld van een link naar een specifiek product:

```
<a href="{{ route('products.show', $product->id) }}"> {{ $product->name }} </a>
```

Een voorbeeld van een index functie in de productscontroller

```
public function index() {
    $products = Product::all();
    return view('products.index', [
        'products' => $products
    ]);
}
```

Een voorbeeld van een store functie in de productscontroller

```
public function store(Request $request, $product) {  
    $this->validate($request, [  
        'name' => 'required',  
        'price' => 'required'  
    ]);  
  
    $product = new Product();  
    $product->name = $request->name;  
    $product->price = $request->price;  
    $product->save();  
  
    return back()->with('success', 'Product created successfully!');  
}
```

Zaken waar je extra op kunt letten of wat vaak vergeten wordt:

- Ieder form moet een @csrf functie bevatten in het form zelf
- Forms die naar een put of een delete route leiden moeten de @method('PUT') of @method('DELETE') functie in het form krijgen
- Bij het gebruik van packages zoals breeze komt ook vaak javascript packages bij kijken. Als je wilt dat alles netjes werkt, vergeet dan geen 'watcher' aan te zetten, door middel van npm run dev. Deze bundelt na iedere change alle javascript en css die gebruikt wordt door verschillende packages en bundelt ze naar één efficiënte js en css file.
- De volgorde van migrations kan uitmaken als je foreignKeys gebruikt, handigst is om eerst de tabellen te migreren die géén foreign keys bevatten, en daarna pas de tabellen die wel een foreign key bevat, of als laatste een migration te doen waarbij je voor alle tabellen pas de foreign keys gaat migreren
-