

Overview and Setup

Laravel is a web application framework with expressive, elegant syntax. It aims to take the pain out of development by easing common tasks used in the majority of web projects.

- Install: `composer global require laravel/installer`.
- New Project: `laravel new my-app`.
- Start Server: `php artisan serve`.
- Help: `php artisan list`.

Artisan CLI (The Swiss Army Knife)

- `php artisan make:controller PostController`: Create a new controller.
- `php artisan make:model Post -m`: Create a model with a migration file.
- `php artisan make:middleware CheckAuth`: Create a middleware.
- `php artisan migrate`: Run database migrations.
- `php artisan tinker`: Interactive shell to interact with your application.

Routing (routes/web.php)

- **Basic Route:** `Route::get('/', function () { return view('welcome'); });`
- **Route with Parameters:** `Route::get('/user/{id}', [UserController::class, 'show']);`
- **Named Routes:** `Route::get('/profile', [UserProfileController::class, 'show'])->name('profile');` Usage in code: `route('profile')`.

Controllers

Located in `app/Http/Controllers`, they handle the request logic.

```
namespace App\Http\Controllers;
```

```
use App\Models\Post;
use Illuminate\Http\Request;
```

```
class PostController extends Controller
```

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

    public function show(Post $post) //
Route Model Binding
    {
        return view('posts.show',
        ['post' => $post]);
    }
}
```

Blade Templating Engine

Laravel's powerful and simple templating system.

- **Output Data:** `{{ $variable }}` (Automatically escaped).
- **Comments:** `{!— This is a comment —!}`.
- **Control Structures:**
 - `@if ($cond) ... @elseif ... @else ... @endif`
 - `@foreach ($items as $item) ... @endforeach`
 - `@auth ... @endauth` (Current user is logged in).
- **Layouts:**
 - `@extends('layouts.app')`
 - `@section('content') ... @endsection`
 - `@yield('content')` (In parent layout).

Eloquent ORM (Active Record)

Interact with your database using an objective-oriented syntax.

- **Read:**
 - `Post::all()`: Get all records.
 - `Post::find(1)`: Find by primary key.
 - `Post::where('active', 1)->orderBy('title')->get()`.
- **Create:**
 - `Post::create(['title' => 'New Post', 'content' => '...'])`.
- **Update:**

```
▸ $post = Post::find(1); $post->title = 'Updated'; $post->save();
```

- **Delete:**
 - `$post->delete()`.

Relationships

- One to One: `$this->hasOne(Profile::class)`.
- One to Many: `$this->hasMany(Comment::class)`.
- Many to Many: `$this->belongsToMany(Role::class)`.
- Eager Loading: `Post::with('comments')->get()` (Avoids N+1 query problem).

Middleware and Validation

- **Middleware:** Filter HTTP requests (e.g., `auth`, `admin`).
- **Validation:**

```
$validated = $request->validate([
    'title' => 'required|unique:posts|max:255',
    'body' => 'required',
]);
```

Pro Tips

Service Container

Laravel's powerful tool for managing class dependencies and performing dependency injection.

Collections

The `Illuminate\Support\Collection` class provides a wrapper for working with arrays of data. Use methods like `map`, `filter`, `reduce` for functional processing.

Queues and Jobs

Use `php artisan make:job ProcessPodcast` to handle time-consuming tasks (like sending emails) in the background.