

Tham Khảo Nhanh Laravel

Artisan, routing, Eloquent, Blade, middleware, xác thực

Artisan CLI

Lệnh Phổ Biến

```
php artisan list # list all commands
php artisan make:controller Post # generate controller
php artisan make:model Post -m # model + migration
php artisan make:middleware Auth # generate middleware
php artisan migrate # run migrations
php artisan db:seed # seed database
```

Lệnh Hữu Dụng

```
php artisan route:list # Hiển thị tất cả routes đã đăng ký
php artisan tinker # REPL tương tác
php artisan cache:clear # Xóa application cache
php artisan config:cache # Cache file cấu hình
php artisan queue:work # Xử lý queue jobs
php artisan storage:link # Tạo symlink public/storage
```

Routing

Route Cơ Bản

```
Route::get('/posts', [PostController::class, 'index']);
Route::post('/posts', [PostController::class, 'store']);
Route::put('/posts/{id}', [PostController::class, 'update']);
Route::delete('/posts/{id}', [PostController::class, 'destroy']);
```

Tính Năng Route

```
Route::resource('posts', PostController::class) # Đăng ký đầy đủ 7 routes CRUD
->name('posts.index') # Đặt tên cho route
->middleware('auth') # Áp dụng middleware
Route::prefix('admin')->group(...) # Nhóm các routes có prefix
```

Tham Số Route

```
Route::get('/posts/{id}', fn($id) => Post::find($id));
Route::get('/posts/{post}', fn(Post $post) => $post); // model binding
Route::get('/u/{name?}', fn($name = 'guest') => $name);
```

Controllers

Resource Controller

```
class PostController extends Controller {
    public function index() {
        return Post::all();
    }
    public function store(Request $request) {
        return Post::create($request->validated());
    }
    public function show(Post $post) {
        return $post;
    }
}
```

Validation

```
$validated = $request->validate([
    'title' => 'required|string|max:255',
    'email' => 'required|email|unique:users',
    'price' => 'required|numeric|min:0',
    'image' => 'nullable|image|max:2048',
]);
```

Eloquent ORM

Truy Vấn Cơ Bản

```
Post::all();
Post::find(1);
Post::where('active', true)->get();
Post::where('views', '>', 100)->orderBy('title')->get();
Post::create(['title' => 'Hello', 'body' => '...']);
Post::findOrFail(1)->update(['title' => 'New']);
Post::destroy(1);
```

Quan Hệ

```
// In Post model
public function user() {
    return $this->belongsTo(User::class);
}
public function comments() {
    return $this->hasMany(Comment::class);
}
// Eager loading
Post::with('user', 'comments')->get();
```

Phương Thức Truy Vấn

```
->first() # Lấy bản ghi đầu tiên phù hợp
->findOrFail() # Lấy hoặc báo lỗi 404
->count() # Đếm số bản ghi
->pluck('name') # Lấy một cột thành collection
->paginate(15) # Phân trang với số lượng mỗi trang
->chunk(100, fn) # Xử lý theo lô
```

Migrations

Cấu Trúc Migration

```
Schema::create('posts', function (Blueprint $table) {
    $table->id();
    $table->string('title');
    $table->text('body')->nullable();
    $table->foreignId('user_id')->constrained();
    $table->boolean('published')->default(false);
    $table->timestamps();
});
```

Kiểu Cột

```
string / text / longText # Dữ liệu chuỗi văn bản
integer / bigInteger # Số nguyên
decimal(8, 2) # Số thập phân với độ chính xác
boolean # True/false
json # Cột JSON
timestamp / timestamps() # Timestamp / created_at + updated_at
foreignId / constrained() # Khóa ngoại với ràng buộc
```

Blade Templates

Cú Pháp Cơ Bản

```
{{ $variable }} # {{-- escaped output --}}
{!! $html !!} # {{-- raw HTML --}}
{{-- this is a comment --}}
@if($condition) ... @endif
@foreach($items as $item) ... @endforeach
@for($i=0; $i<10; $i++) ... @endfor
```

Kế Thừa Layout

```
{{-- layouts/app.blade.php --}}
<body>@yield('content')</body>
```

```
{{-- posts/index.blade.php --}}
@extends('layouts.app')
@section('content')
    <h1>Posts</h1>
@endsection
```

Components & Directives

```
@include('partials.nav') # Nhúng view con
@component('alert') # Render component
@auth / @guest # Kiểm tra trạng thái xác thực
@can('edit', $post) # Kiểm tra quyền authorization
@csrf # Thêm CSRF token field
@method('PUT') # Giải lập HTTP method
```

Middleware

Tạo Middleware

```
class CheckAge {
    public function handle(Request $request, Closure $next) {
        if ($request->age < 18) {
            return redirect('home');
        }
        return $next($request);
    }
}
```

Đăng Ký & Sử Dụng

```
// In bootstrap/app.php
$middleware->alias(['age' => CheckAge::class]);

// On route
Route::get('/adult', fn() => ...)->middleware('age');
Route::middleware(['auth', 'age']->group(fn() => ...);
```

Authentication

Xác Thực Cơ Bản

```
Auth::attempt(['email' => $e, 'password' => $p]);
Auth::login($user);
Auth::logout();
Auth::check(); // is logged in?
Auth::user(); // current user
```

Bảo Vệ Routes

```
Route::middleware('auth')->group(function () {
    Route::get('/dashboard', fn() => view('dashboard'));
});
```

Laravel Starter Kits

```
Laravel Breeze # Xác thực tối giản dùng Blade hoặc Inertia
Laravel Jetstream # Xác thực đầy đủ tính năng + team
Sanctum # Xác thực API với token đơn giản
Passport # Triển khai OAuth2 đầy đủ
```

Common Patterns

Service Container

```
// Bind interface to implementation
$this->app->bind(PaymentGateway::class, StripeGateway::class);
// Resolve
$gateway = app(PaymentGateway::class);
```

Tham Khảo Nhanh Laravel

Queue Jobs

```
// Dispatch
ProcessPodcast::dispatch($podcast)->onQueue('media');
// In Job handle():
public function handle() { /* process */ }
// Config: .env QUEUE_CONNECTION=redis
```

Events & Listeners

```
event(new OrderPlaced($order));

class SendOrderConfirmation {
    public function handle(OrderPlaced $event) {
        Mail::to($event->order->user)->send(...);
    }
}
```