



SETUP & ARTISAN

```
composer create-project laravel/laravel
myapp
php artisan serve
php artisan make:controller
UserController --resource
php artisan make:model User -m (with
migration)
php artisan migrate
php artisan tinker (REPL)
```

ROUTING

```
Route::get('/users',
[UserController::class, 'index']);
Route::post('/users',
[UserController::class, 'store']);
Route::apiResource('users',
UserController::class);
Route::get('/profile', fn() =>
view('profile'))->middleware('auth');
Route::prefix('api')->group(fn() => {
Route::get('/users',...); });
```

CONTROLLERS

```
class UserController extends Controller
{
public function index() {
return User::paginate(15);
}
public function store(StoreUserRequest
$req) {
$user = User::create($req->validated());
return response()->json($user, 201);
}
}
```

ELOQUENT ORM

```
User::all();
User::find(1);
User::where('active',
true)->orderBy('name')->get();
User::firstOrCreate(['email' => $email],
['name' => $name]);
$user->save(); // $user->delete();
User::create(['name' => 'Alice']); //
mass assignment
User::chunk(100, fn($users) => ...); //
batches
```

MIGRATIONS

```
Schema::create('users', function
(Blueprint $table) {
$table->id();
$table->string('name');
$table->string('email')->unique();
$table->timestamps();
});
php artisan migrate / migrate:rollback
/ migrate:fresh
```

BLADE TEMPLATES

```
{{ $user->name }} {!! $html !!}
@if($user->isAdmin()) Admin @endif
@foreach($users as $user) {{ $user->name
}} @endforeach
@extends('layouts.app')
@section('content') ... @endsection
@component('alert', ['type' => 'error'])
```

MIDDLEWARE

```
php artisan make:middleware
EnsureUserIsAdmin
public function handle(Request $req,
Closure $next) {
if (!$req->user()->isAdmin())
abort(403);
return $next($req);
}
// Register in app/Http/Kernel.php or
bootstrap/app.php
Route::middleware(['auth',
'admin'])->group(...);
```

VALIDATION

```
$validated = $request->validate([
'name' => 'required|string|max:255',
'email' =>
'required|email|unique:users',
'age' => 'nullable|integer|min:0',
]);
// Form Request:
class StoreUserRequest extends
FormRequest {
public function rules(): array { ... }
}
```

ELOQUENT RELATIONSHIPS

```
// User.php:
public function orders(): HasMany {
return $this->hasMany(Order::class);
}
public function role(): BelongsTo {
return $this->belongsTo(Role::class);
}
// Eager loading:
User::with('orders', 'role')->get();
```

QUERY BUILDER

```
DB::table('users')
->select('name', 'email')
->where('active', true)
->orderBy('name')
->paginate(15);
DB::table('users')->insert(['name'=>'Alice']);
DB::transaction(function() { ... });
```

COMMON ARTISAN COMMANDS

```
php artisan
make:model/controller/middleware/job/event/listener
php artisan route:list / cache:clear
/ config:cache
php artisan queue:work / queue:listen
php artisan storage:link /
key:generate
php artisan schedule:work (runs
scheduler locally)
```

COMMON GOTCHAS

Mass assignment: \$fillable or
\$guarded on model
N+1 queries: always use with() for
relationships
APP_KEY required generate with
artisan key:generate
Storage link needed for public disk
file access
Eloquent soft deletes need
SoftDeletes trait + deleted_at
column