

ببساطة — محاضرة سريعة Minimal APIs

!شرح عملي بـ 10 صفحات فقط

مقدمة سريعة

API؟ يعني إيه

رد → أنت → أمر (Server) → المطبخ → (API) طلب → الويتر → (Client) أنت

API = backend يطلب بيانات من ال frontend طريقة آمنة بتسمح للـ

Minimal؟ يعني إيه

(معقد) Resources و Requests و Controllers بدل ما تكتب

```
// ✗ معقد: 5 ملفات
Route::apiResource('posts', PostController::class);
```

(بسيط) route تكتب مباشرة في ال

```
// ✔ بسيط: ملف واحد
Route::get('/posts', fn() => Post::all());
Route::post('/posts', fn(Request $r) => Post::create($r->validated()));
```

الأساسية Concepts ال

HTTP Methods:

Method	معنى	مثال
GET	اجلب بيانات	GET /posts
POST	انشئ جديد	POST /posts + data
PATCH	عدّل جزء	PATCH /posts/1
DELETE	احذف	DELETE /posts/1

Status Codes:

الكود	المعنى
-------	--------

الكود	المعنى
200	نجاح + بيانات
201	اتنشأت حاجة جديدة
204	data نجاح بدون
400	الطلب غلط
401	ما متسجل
403	ما عندك صلاحية
404	مش موجود
422	فشل validation ال
500	حاجة غلط في الكود

مثال عملي: Todo API

ال Routes ([routes/api.php](#)):

```
<?php

use App\Models\Todo;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;

// 🔍 جلب كل ال Todos
Route::get('/todos', fn() => Todo::all());

// 🔍 واحد Todo جلب
Route::get('/todos/{id}', function ($id) {
    $todo = Todo::find($id) ?? abort(404);
    return $todo;
});

// ✨ جديد Todo انشئ
Route::post('/todos', function (Request $request) {
    $request->validate([
        'title' => 'required|string|max:255',
        'is_completed' => 'nullable|boolean'
    ]);

    return response()->json(
        Todo::create($request->validated()),
        201
    );
});

// ✎ عدل Todo
```

```
Route::patch('/todos/{id}', function (Request $request, $id) {
    $todo = Todo::find($id) ?? abort(404);
    $todo->update($request->validate([
        'title' => 'nullable|string|max:255',
        'is_completed' => 'nullable|boolean'
    ]));
    return $todo;
});

// 🗑️ احذف Todo
Route::delete('/todos/{id}', function ($id) {
    Todo::find($id)?->delete() ?? abort(404);
    return response()->json(null, 204);
});
```

JI Model:

```
<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Todo extends Model
{
    protected $fillable = ['title', 'is_completed'];
    protected $casts = ['is_completed' => 'boolean'];
}
```

🔧 خطوات سريعة

```
# 1. انشئ مشروع
composer create-project laravel/laravel api-demo && cd api-demo

# 2. انشئ Model و Migration
php artisan make:model Todo -m

# 3. أضف الـ fields في Migration:
# $table->id();
# $table->string('title');
# $table->boolean('is_completed')->default(false);
# $table->timestamps();

# 4. شغل Migration
php artisan migrate

# 5. اكتب الـ Routes (اللي فوق)

# 6. شغل السيرفر
```

```
php artisan serve

# اختبار 7:
curl http://localhost:8000/api/todos
curl -X POST http://localhost:8000/api/todos \
  -H "Content-Type: application/json" \
  -d '{"title":"Learn APIs"}'
```

✖ معالجة الأخطاء

404 — Resource موجود: منس:

```
Route::get('/todos/{id}', function ($id) {
    return Todo::findOrFail($id); // ترجع 404 تلقائياً
});
```

422 — Validation فشل:

```
// Laravel validation 422 ترجع وتعمل الـ
$request->validate([
    'title' => 'required|string|max:255',
]);
```

Error Response:

```
{
  "message": "The given data was invalid.",
  "errors": {
    "title": ["The title field is required."]
  }
}
```



بسر سريع Advanced

1 Authentication (Sanctum):

```
// token محمي بـ:
Route::middleware('auth:sanctum')->post('/todos', function (Request $r) {
    return auth()->user()->todos()->create($r->validated());
});

// Login:
```

```
Route::post('/login', function (Request $r) {
    $user = User::where('email', $r->email)->first();

    if (!Hash::check($r->password, $user->password))
        return response()->json(['error' => 'Invalid'], 401);

    return ['token' => $user->createToken('api')->plainTextToken];
});
```

2 Pagination:

```
Route::get('/todos', fn(Request $r) =>
    Todo::paginate($r->get('per_page', 15))
);

// GET /todos?per_page=10
```

3 Filtering & Sorting:

```
Route::get('/todos', function (Request $r) {
    $query = Todo::query();

    if ($r->has('completed'))
        $query->where('is_completed', $r->boolean('completed'));

    if ($r->has('sort'))
        $query->orderBy($r->sort, $r->get('order', 'desc'));

    return $query->paginate();
});

// GET /todos?completed=1&sort=created_at&order=asc
```

4 API Versioning:

```
Route::prefix('v1')->group(fn() =>
    Route::apiResource('todos', TodoControllerV1::class)
);

Route::prefix('v2')->group(fn() =>
    Route::apiResource('todos', TodoControllerV2::class)
);
```

```
// ✅ 1. استخدم Resources للـ clean responses:
Route::get('/todos', fn() => TodoResource::collection(Todo::all()));

// ✅ 2. Validate دائماً:
$request->validate(['title' => 'required|string|max:255']);

// ✅ 3. صم HTTP Methods استخدم الـ:
// GET = اِجلب | POST = انشئ | PATCH = عدّل | DELETE = احذف

// ✅ 4. الصيغ status code الرد بـ الـ:
response()->json($data, 201); // 201 للـ POST
response()->json(null, 204); // 204 للـ DELETE

// ✅ 5. (بتعملها تلقائياً Laravel) منع SQL Injection:
Todo::create($request->validated()); // ✓ آمن
Todo::create($request->all()); // ✗ خطر!
```

Cheat Sheet

الـ 5 الأساسية Routes الـ:

```
Route::get('/posts', ...); // Read all
Route::get('/posts/{id}', ...); // Read one
Route::post('/posts', ...); // Create
Route::patch('/posts/{id}', ...); // Update
Route::delete('/posts/{id}', ...); // Delete

// أو بـ سطر واحد:
Route::apiResource('posts', PostController::class);
```

Middleware:

```
// Authentication:
Route::middleware('auth:sanctum')->group(fn() => ...);

// Rate limiting (60 الدقيقة في طلب):
Route::middleware('throttle:60,1')->group(fn() => ...);
```

Quick Commands:

```
php artisan serve # تشغيل السيرفر
php artisan make:model Todo -m # Model + Migration
php artisan migrate # تشغيل Migrations
php artisan route:list # عرض الـ Routes
```

Mini-Challenge

Products: ل API ابني

```
POST /api/products → انشئ منتج
GET /api/products → pagination جلب كل المنتجات (مع)
GET /api/products/{id} → جلب منتج واحد
PATCH /api/products/{id} → عدل منتج
DELETE /api/products/{id} → احذف منتج
```

Validation: name (required), price (numeric), category (in:a,b,c)

سريع FAQ

Q: Minimal أم MVC?

A: Minimal السريع والبسيط، MVC معقد.

Q: Resources ليه?

A: response عشان تتحكم في الحقول اللي بتظهر وتنسق ال.

Q: API إزاي أحمي ال?

A: Validate استخدم، دائماً، Auth، Rate Limiting.

Q: Controllers احتاج?

A: البسيطة: لا، لل معقدة: أيوا logic لل.

الخلاصة

- ✓ **Minimal routes** — ابدأ بسيط
- ✓ **Validate** — دائماً تحقق من البيانات
- ✓ **Status codes** — استخدم ال صحيحة
- ✓ **Test** — curl أو Postman
- ✓ **Expand** — لما احتجت features أضف

Happy API Building! 