

Minimal APIs — محاورة سريعة

شرح عملی بـ 10 صفحات فقط!

مقدمة سريعة

API يعني إيه؟

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

API يطلب بيانات من الـ frontend طريقة آمنة بتسمح للـ backend.

Minimal يعني إيه؟

بدل ما تكتب Controllers وRequests وResources (معقد):

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

(بسيل) route تكتب مباشرة في الـ:

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

الـ Concepts الأساسية

HTTP Methods:

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

Status Codes:

الكود	المعنى
-------	--------

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

💻 مثال عملي: Todo API

الـ Routes ([routes/api.php](#)):

```
<?php

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

// 🔎 اجلب كل الـ Todos
Route::get('/todos', fn() => Todo::all());

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

// ✨ انشئ Todo جديد
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);
});

```

ال 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. Migration، اضف الى fields:
# $table->id();
# $table->string('title');
# $table->boolean('is_completed')->default(false);
# $table->timestamps();

# 4. شغل Migration
php artisan migrate

# 5. اكتب الى Routes (التي فوق)
# 6. شغل الماينر

```

```
php artisan serve

# اختبر :
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: مش

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

422 — Validation: فشل

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

Error Response:

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

🚀 Advanced سريعاً

1 Authentication (Sanctum):

```
// محمي بـ token:
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)
);
```

⭐ Best Practices

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

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

// ✓ 3. استخدم الـ HTTP Methods صح:
// GET = اجلب | POST = انشئ | PATCH = عدل | DELETE = حذف

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

// ✓ 5. (تعملها تلقائيًا) منع SQL Injection (Laravel):
Todo::create($request->validated()); // ✓ آمن
Todo::create($request->all()); // خطير!
```

Cheat Sheet

الـ 5 الأساسية لـ Routes:

```
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 طلب في الدقيقة):
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

ابني API لـ **Products**:

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

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

❓ FAQ سريع

Q: Minimal أم MVC؟

A: Minimal ، لـ بسيط و السريع MVC ، لـ معقد.

Q: ليه Resources؟

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

Q: إزاي أحمي الـ API؟

A: Validate دائمـاً، Auth، Rate Limiting.

Q: احتاج Controllers؟

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

الخلاصة

- ابدأ بسيط — Minimal routes
- Validate — دائمـاً تحقق من البيانات
- Status codes — استخدم الـ صيغة
- Test — curl أو Postman
- Expand — أضف features لما احتجت

Happy API Building! 