

◆ 1. API REST en Symfony + Frontend Independiente (🚀 Mejor opción para proyectos modernos)

- ✂ Backend: Symfony actúa como un servidor de API.
- ✂ Frontend: Se desarrolla en React, Vue, Angular, o incluso HTML + JavaScript.
- ✂ Comunicación: Se usa JSON con API REST (o GraphQL).

◆ Cómo hacerlo en Symfony (Backend API)

1. Instalar API Platform (opcional pero recomendado)

```
bash

composer require api
```

📄 Copiar ✎ Editar

Esto convierte automáticamente tus entidades en endpoints de API.

2. Crear un controlador REST manualmente Si prefieres un control total, puedes crear una API sin API Platform:

```
php

// src/Controller/ApiController.php
namespace App\Controller;

use App\Entity\Recurso;
use Doctrine\ORM\EntityManagerInterface;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\JsonResponse;
use Symfony\Component\Routing\Annotation\Route;

#[Route('/api')]
class ApiController extends AbstractController
{
    #[Route('/recursos', methods: ['GET'])]
    public function listarRecursos(EntityManagerInterface $entityManager): JsonResponse
    {
        $recursos = $entityManager->getRepository(Recurso::class)->findAll();
        return $this->json($recursos);
    }
}
```

📄 Copiar ✎ Editar

Esto devuelve los datos como JSON.

3. Activar CORS para permitir peticiones desde el frontend

✂ Edita `config/packages/cors.yaml`:

```
yaml

nelmio_cors:
    defaults:
        allow_origin: ['*']
        allow_methods: ['GET', 'POST', 'PUT', 'DELETE', 'OPTIONS']
        allow_headers: ['Content-Type', 'Authorization']
```

📄 Copiar ✎ Editar

4. Ejemplo de consumo en el frontend (JavaScript + Fetch)

```
js

fetch('http://localhost:8000/api/recursos')
    .then(response => response.json())
    .then(data => console.log(data));
```

📄 Copiar ✎ Editar