

UD 2

2.2 JSON



¿Qué veremos hoy?

- Introducción a JSON
- JSON y Laravel 11.x



JSON (JavaScript Object Notation)

- Formato de texto ligero orientado al intercambio de datos.
- Es independiente del lenguaje, aunque deriva de la sintaxis de JavaScript.
- JSON representa los datos como pares clave-valor. Los valores pueden ser números, cadenas, arrays, objetos (otros pares clave-valor) o valores booleanos y nulos.
- Gracias a su simplicidad y legibilidad, JSON se ha convertido en el formato estándar para APIs y configuraciones en aplicaciones modernas.

```
{  
  "id": 12345,           // Número  
  "name": "Carlos Martínez", // Cadena  
  "isActive": true,      // Booleano  
  "membership": null,    // Nulo  
  "preferences": {       // Objeto  
    "language": "es",    // Cadena  
    "notifications": {   // Objeto  
      "email": true,     // Booleano  
      "sms": false      // Booleano  
    }  
  },  
  "addresses": [         // Array  
    { "type": "home" },  // Objeto  
    { "type": "work" }   // Objeto  
  ]  
}
```




JSON y Laravel 11.x

Laravel utiliza JSON en áreas clave. A continuación, enumeramos las principales:

1. Respuestas JSON en APIs.
2. Validación de JSON en requests.
3. Resources: respuestas JSON personalizadas.
4. Interacción con la BD (UD3).
5. Casts en Modelos Eloquent (UD3).
6. Migraciones con campos JSON (UD3).
7. Colas y sesión (Redis) (UD4).



Respuestas JSON en APIs.

```
return response()->json([  
    'clave' => 'valor',  
    'status' => 'success',  
    'message' => 'Operación exitosa'  
]);
```



Validación de JSON en requests.

```
$data = $request->validate([  
    'name' => 'required|string',  
    'mail' => 'required',  
    'password' => 'required|min:8'  
]);
```



Resources: respuestas JSON personalizadas

1. Crear el resource
2. Definir su estructura
3. Usar el resource



Resources: respuestas JSON personalizadas

1. Crear el resource

```
php artisan make:resource UserResource
```



Resources: respuestas JSON
personalizadas

2. Definir su estructura

```
namespace App\Http\Resources;

use Illuminate\Http\Request;
use Illuminate\Http\Resources\Json\JsonResource;

class UserResource extends JsonResource
{
    public function toArray($request)
    {
        return [
            'id' => $this->id,
            'nombre' => $this->name,
            'correo' => $this->email,
            'fecha_registro' => $this->created_at->format('Y-m-d H:i:s'),
            'roles' => RoleResource::collection($this->roles), // si el us
        ];
    }
}
```



Resources: respuestas JSON
personalizadas

3. Usar el resource


```
use App\Http\Resources\UserResource;  
use App\Models\User;  
  
public function show($id)  
{  
    $user = User::findOrFail($id);  
    return new UserResource($user);  
}
```

¿Dudas?

