

Desarrollo web en entorno servidor



Desarrollo de API
Problem details for HTTP API – RFC 7807 y RFC 9457

IES Clara del Rey – Madrid

Información sobre errores en API HTTP

En una API, sobre todo si ofrece servicios a otros sistemas o aplicaciones, el manejo y la información sobre los errores producidos es clave.

Sin un formato estandarizado, cada API puede responder con estructuras diferentes, dificultando su consumo a los distintos clientes

Los códigos HTTP ayudan, pero no siempre explican la causa exacta del error, se limitan a dar una información general.

Actualmente se utiliza un estándar denominado "Problem details for HTTP API", definido en la RFC 9457, que es una mejora de la RFC 7808.

Devolver los errores en este formato facilita su procesamiento en cliente, la integración en sistemas de log, su análisis automatizado, etc.

RFC 9457 y RFC 7807

Definen un formato estándar para devolver información sobre errores en una API HTTP.

Utilizan un esquema basado en JSON o XML con detalles estructurados, con una forma específica.

La RFC 9457 es una actualización (julio de 2023) de la RFC 7807.

Todavía es habitual encontrar artículos que se refieren a la 7807 sin tener en cuenta su actualización.

RFC 9457 mantiene la base de la 7807, pero introduce algunas mejoras.

Estructura de un mensaje de error

Tienen cinco campos principales, que permiten que el error pueda ser interpretado tanto por un humano como por un sistema automatizado.

Obligatorios:

- title: breve descripción del problema en texto plano.
- status: código HTTP correspondiente al error.

Opcionales:

- type: identificador del tipo de error. Cada tipo de error tiene un id único. Puede ser (se recomienda) URL, que NO tiene por que existir.
- detail: explicación detallada del problema en la solicitud.
- instance: Identificador de la solicitud que falló, suele ser la URL.

Ejemplos

Ejemplo: problema "saldo insuficiente" al intentar retirar dinero.

```
{  
  "type": "https://example.com/probs/out-of-credit",  
  "title": "Saldo insuficiente",  
  "status": 403,  
  "detail": "Tu saldo es 30, pero necesitas 50.",  
  "instance": "/cuenta/12345/retirar/50"  
}
```

Sería válido también con menos campos:

```
{  
  "title": "Saldo insuficiente",  
  "status": 403,  
  "detail": "Tu saldo es 30, pero necesitas 50.",  
  "instance": "/cuenta/12345/retirar/50"  
}
```

Ampliando los detalles del error

La RFC no impone restricciones sobre campos adicionales, siempre que se mantenga la estructura básica. Estos campos pueden proporcionar información extra que sea útil para el consumidor de la API, pero no deben interferir con los campos estándar. Ejemplo ampliado:

```
{  
    "type": "https://example.com/probs/out-of-credit",  
    "title": "Saldo insuficiente",  
    "status": 403,  
    "detail": "Tu saldo es 30, pero necesitas 50.",  
    "instance": "/cuenta/12345/retirar/50",  
    "suggestion": "Por favor recarga tu saldo para continuar.",  
    "errorCode": "INSUFFICIENT_FUNDS",  
    "retryAfter": "2025-02-17T12:00:00Z"  
}
```

Problem details en Spring Boot

Spring tiene una clase "ProblemDetail" diseñada específicamente para devolver este tipo de información cuando una llamada a un método de controlador falla.

Tiene los atributos necesarios para cumplir la especificación.

Lo habitual es utilizarla en métodos anotados con @ExceptionHandler:

```
@ExceptionHandler(CustomNotFoundException.class)  
public ResponseEntity<ProblemDetail> handleException(CustomNotFoundException ex) {  
    ProblemDetail problem = ProblemDetail.forStatus(HttpStatus.NOT_FOUND);  
    problem.setTitle(ex.getMessage());  
    problem.setDetail(ex.getDetails()); // Específico de CustomNotFoundException  
    return ResponseEntity.status(HttpStatus.NOT_FOUND).body(problem);  
}
```

Ampliando ProblemDetail en Spring Boot

Para añadir información adicional a una respuesta de errores, se puede usar el método "set":

```
// Propiedades estándares:  
ProblemDetail problem = ProblemDetail.forStatus(HttpStatus.CONFLICT);  
problem.setTitle("Error de saldo");  
problem.setDetail("No tienes suficiente saldo");  
problem.setType(URI.create("https://example.com/probs/out-of-credit"));  
  
// Propiedades adicionales  
problem.set("errorCode", "INSUFFICIENT_FUNDS"); // Campo código interno de error  
problem.set("suggestion", "Recarga tu saldo para continuar.");  
problem.set("retryAfter", "2025-02-17T12:00:00Z"); // Indicar un tiempo para  
                                                    reintentar la operación  
return ResponseEntity.status(HttpStatus.CONFLICT).body(problem);
```