OpenAPI

Especifique o contrato HTTP da agenda

Pré-requisitos

Conheça método, caminho, status HTTP e JSON. O exemplo é um documento OpenAPI 3.1.1 escrito em YAML. Use Python 3 com acesso ao registro de pacotes para instalar o validador em uma pasta de laboratório.

Escolha o que o contrato precisa dizer [1]

OpenAPI descreve operações de APIs HTTP de maneira legível por pessoas e ferramentas. A seção paths reúne caminhos e métodos. Cada operação pode declarar parâmetros, corpo de entrada e respostas. Em responses, o status identifica o desfecho. Content identifica o tipo de mídia e o schema declara a forma do dado. Um example demonstra uma instância concreta.

Distinga as duas versões

O campo openapi seleciona a versão da especificação usada para interpretar o documento. O campo info.version identifica a versão do documento da API. Já uma versão no caminho, como /v1/reservas, seria uma decisão de roteamento do serviço, independente dos dois campos.

Reaproveite apenas contratos equivalentes

Use components.schemas e $ref para referenciar a mesma estrutura em operações que compartilham a representação. Um consumidor pode depender de estados definidos no enum. Acrescentar um estado exige examinar clientes com tratamento exaustivo. Retirar um campo obrigatório de uma resposta também pode quebrar integrações existentes.

Separe validação do documento da execução

Um documento válido ainda precisa ser comparado com o comportamento real. Teste status, tipo de mídia e conteúdo das respostas com a API em funcionamento. O contrato de estudo cobre duas respostas e não representa implantação ou mecanismo de acesso a reservas.

OpenAPI

Exercício resolvido • Consulte uma reserva

Enunciado e dados de partida

Para GET /reservas/{id}, documente um código com R seguido de dígitos. R45 existe e está confirmada. Para uma reserva inexistente, a API didática retorna 404 com código RESERVA_INEXISTENTE.

1. Salve este início em agenda.yaml

```
openapi: 3.1.1
info:
  title: Agenda de aulas
  version: 1.0.0
paths:
  /reservas/{id}:
    get:
      operationId: consultarReserva
      parameters:
        - name: id
          in: path
          required: true
          schema:
            type: string
            pattern: '^R[0-9]+$'
      responses:
        '200':
          description: Reserva encontrada
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Reserva'
              example: {id: R45, estado: confirmada}
        '404':
          description: Reserva inexistente
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Erro'
              example: {codigo: RESERVA_INEXISTENTE}
```

Leia a operação

O parâmetro id corresponde a {id} e é obrigatório. As chaves 200 e 404 estão entre aspas para garantir que sejam strings no YAML. O schema da resposta é referenciado por um fragmento local.

OpenAPI

Conclua e valide o contrato

2. Acrescente ao final, sem indentar components

```
components:
  schemas:
    Reserva:
      type: object
      required: [id, estado]
      properties:
        id:
          type: string
          pattern: '^R[0-9]+$'
        estado:
          type: string
          enum: [pendente, confirmada, cancelada]
    Erro:
      type: object
      required: [codigo]
      properties:
        codigo:
          type: string
```

3. Execute no PowerShell [2]

```
python -m venv .venv
.venv/Scripts/python.exe -m pip install openapi-spec-validator
.venv/Scripts/python.exe -m openapi_spec_validator agenda.yaml
```

Confira a resposta

A validação deve terminar com sucesso. Para R45, o corpo inclui id e estado da lista permitida. Para 404, o corpo requer codigo. Um estado como ‘paga’ falha no schema Reserva. A validação estrutural do documento não substitui a verificação de instâncias de resposta.

Caso de erro • Retire a obrigatoriedade do caminho

Troque required: true do parâmetro id por required: false e rode o último comando novamente. O validador deve falhar com código diferente de zero. Restaure true e confirme o sucesso. Em seguida, confira se $ref aponta para o componente do mesmo desfecho e se os exemplos atendem aos campos obrigatórios.

Referências primárias

[1] OpenAPI Initiative • Especificação OpenAPI 3.1.1

[2] OpenAPI Spec Validator • Interface de linha de comando

OpenAPI

Documente os resultados que mudam a interface

Modele também as falhas esperadas

Documente o código e o esquema de cada resultado que o cliente precisa interpretar. No conflito de horário, inclua um código de domínio estável. O aplicativo pode usar esse valor para decidir se deve atualizar a disponibilidade.

Exercício adicional

O contrato de criação só descreve 201, mas a interface precisa tratar horário ocupado. O que adicionar?

Resolução

Adicione uma resposta 409 com application/json, esquema de erro e um exemplo com codigo HORARIO_OCUPADO. Teste a resposta real contra o documento e faça o cliente atualizar a lista sem perder os demais dados.

Ao assistir ao vídeo

Identifique onde o contrato descreve o corpo do pedido e as respostas por código HTTP.

Vídeo complementar

Introdução ao Swagger Editor