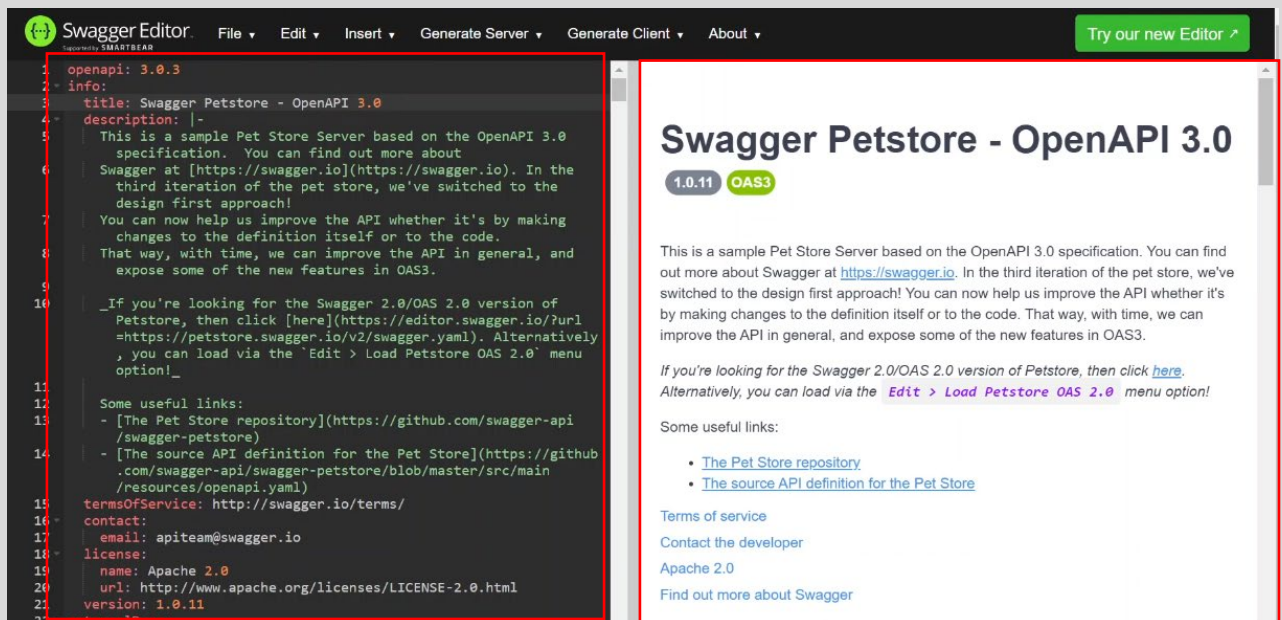


# Documentação de API

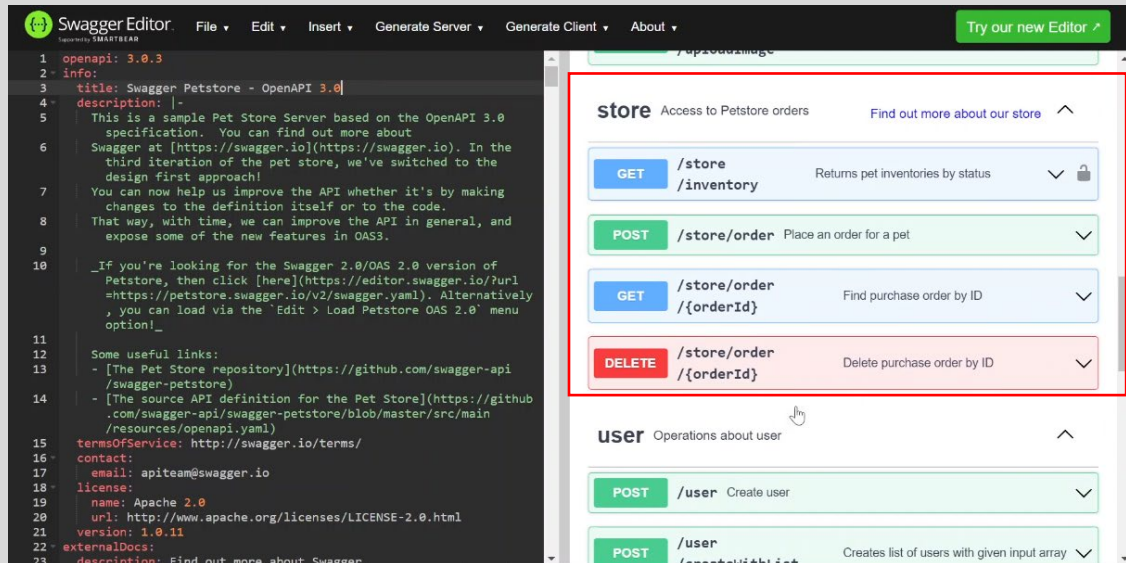
## Swagger Editor

Nesse conteúdo, você vai conhecer o Swagger Editor, uma ferramenta online para documentação de API em json ou YAML.

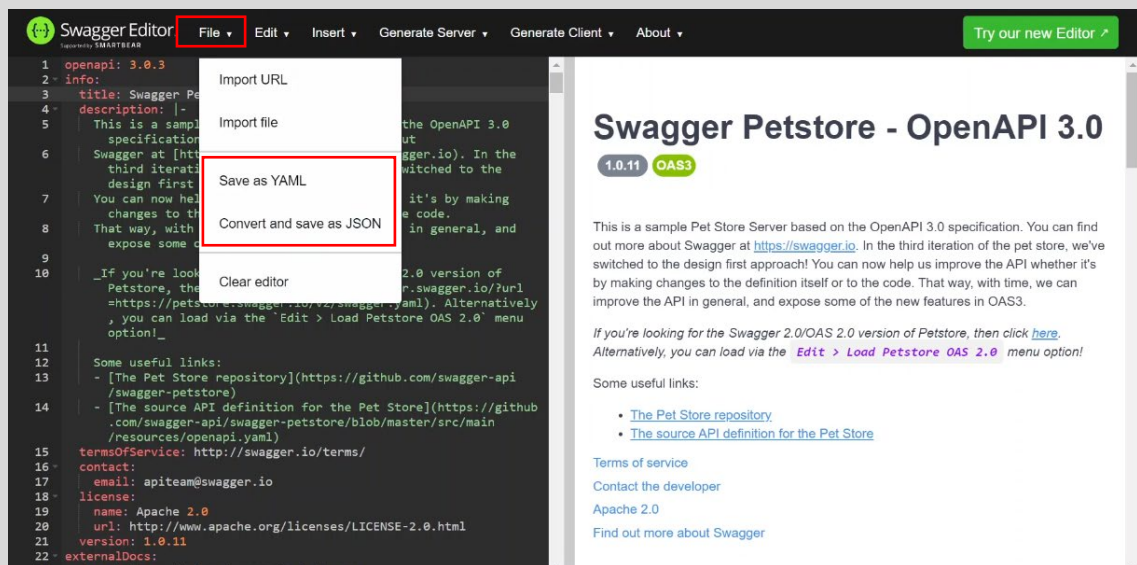
1. Acesse o endereço **editor.swagger.io**. O lado esquerdo é a parte do editor de código, que pode ser json ou YAML. O lado direito é a visualização HTML.



2. O Swagger mostra as ações disponíveis, a finalidade de cada uma, os métodos e urls.



3. Seu arquivo de documentação pode ser baixado em json ou YAML. Para isso, clique em **File** no menu superior e então escolha **Save as YAML** ou **Convert and save as JSON**.



## Saiba mais



Json (Javascript Object Notation) é uma representação de dados baseada em Javascript, muito usada para comunicação de sistemas web.

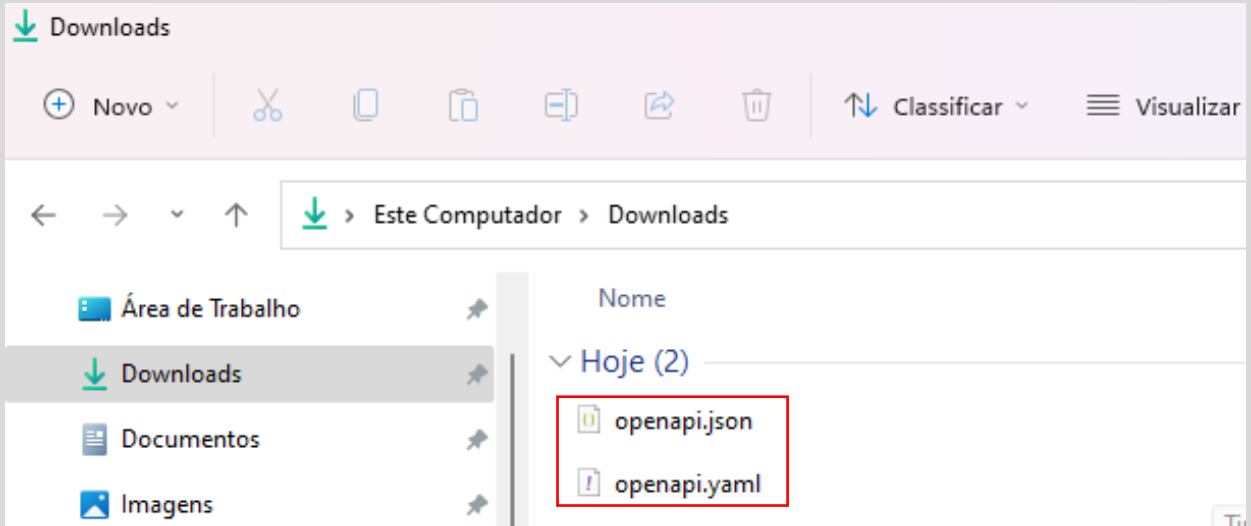
O json é baseado na estrutura chave/valor. Podemos dizer que um json representa um objeto, que tem atributos (chaves) e cada atributo tem um valor.

A sintaxe json é simples: chaves delimitam um bloco, colchetes delimitam um array, dois pontos separam chaves de valores e vírgulas separam os atributos chave-valor. Os dados podem ser string, número, booleano ou nulo.

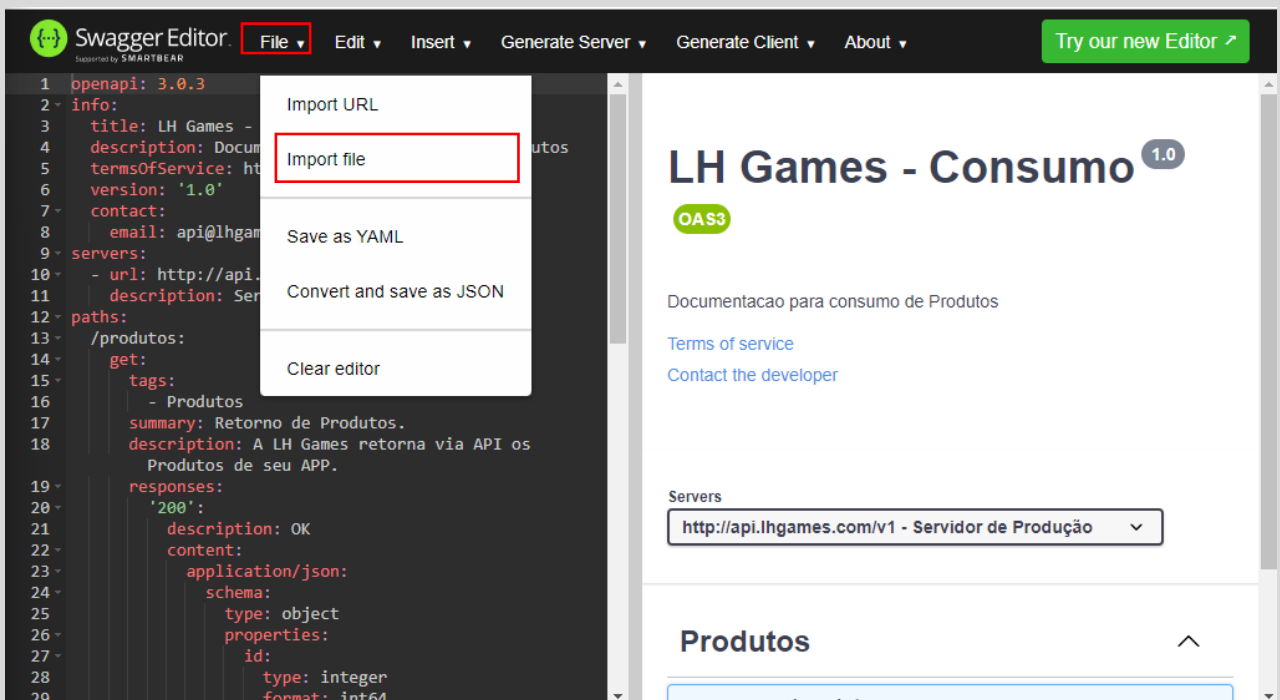
Usando linguagens de programação adequadas, é possível manipular objetos json.

A YAML é outra forma de estruturar dados, de alta compatibilidade com o XML. De sintaxe intuitiva, a YAML usa a indentação para estruturar os dados e identifica seus elementos com sinais simples, como colchetes e traços

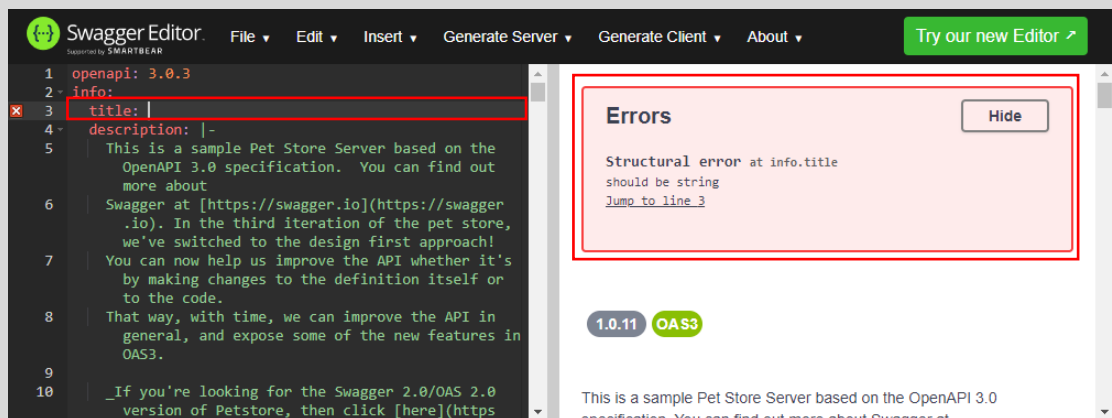
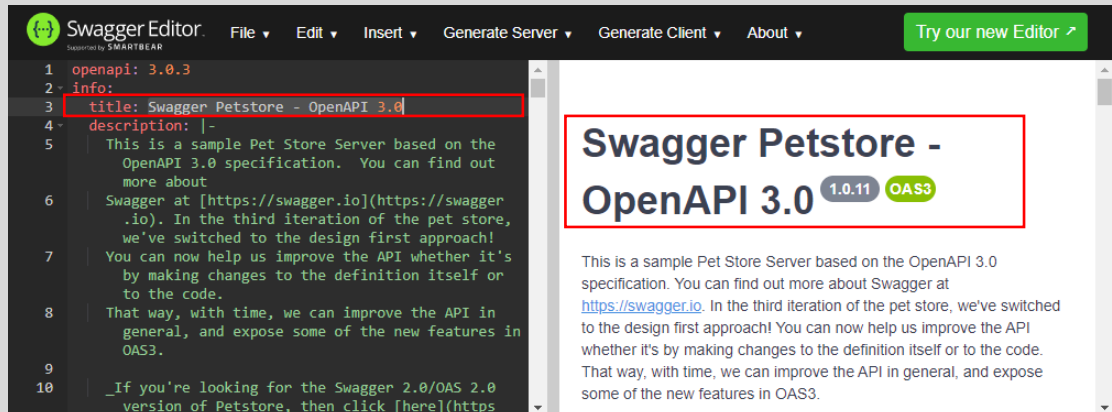
4. O arquivo baixado em sua máquina virá já nomeado como **openapi** e pode ser alterado em qualquer editor de código ou texto, como o VS Code ou bloco de notas.



E você pode importá-lo no Swagger Editor, clicando em **File** no menu principal e então em **Import file**.



5. Uma das vantagens do Swagger Editor é conferir as alterações em tempo real. As imagens a seguir mostram uma alteração no título da documentação.



O Swagger mostra um alerta de erro na visualização quando encontra falhas no código, como mostrado na imagem anterior.

## Desenvolvimento da documentação

Você tem várias maneiras para desenvolver ou alterar sua documentação. Se você quiser visualizar as alterações em tempo real, edite o YAML direto no editor do site. Se preferir trabalhar localmente (em sua máquina), baixe o arquivo openapi (em json ou YAML) e ajuste sua documentação em um editor de código ou texto.

### Dica!

Editores de código, como o VS Code, facilitam a visualização e o uso da sintaxe correta.



Nesse exemplo, vamos desenvolver um arquivo em json no VS Code. Para visualizar as alterações, você deve salvar o arquivo ajustado e importá-lo para o Swagger Editor.

## Elementos base da estrutura

Resumidamente, os elementos básicos são:

- 1) Especificação – versão de seu OpenAPI
- 2) Info – informações gerais
- 3) Servers – lista de servidores
- 4) Paths – as ações disponibilizadas

## 1) Especificação

```
"openapi": "3.0.3",
```

No exemplo, usamos OpenAPI especificação 3. No código, do lado esquerdo, a especificação está na primeira linha. Na visualização, do lado direito, a especificação aparece após o número da versão do documento, na sigla OAS3 (OpenAPI Specification 3).

The screenshot shows a development environment with two panels. The left panel displays the OpenAPI specification in a JSON file named 'openapi.json'. The first line of the file is highlighted with a red box: `"openapi": "3.0.3",`. The rest of the file defines an API for 'LH Games - Consumo' with a version of '1.0'. The right panel shows the Swagger UI for this API. The title 'LH Games - Consumo' is followed by a '1.0' version number and an 'OAS3' badge. Below the title, there are links for 'Documentacao para consumo de Produtos', 'Terms of service', and 'Contact the developer'. A 'Servers' dropdown menu is set to 'http://api.lhgames.com/v1 - Servidor de Produção'. Under the 'Produtos' section, two endpoints are listed: a GET endpoint for '/produtos' (Retorno de Produtos) and a POST endpoint for '/produtos' (Adicionar novo Produto).

### Importante

As especificações anteriores usavam estrutura e elementos diferentes. A ausência de alguns elementos pode ocasionar alertas de erro na visualização ou até impedir a interpretação do seu arquivo.





## 2) Info

```
"info": {
  "title": "LH Games - Consumo",
  "description": "Documentacao para consumo de Produtos",
  "termsOfService": "http://lhgames.com/v1/api-docs",
  "version": "1.0",
  "contact": {
    "email": "api@lhgames.com"
  }
},
```

São as informações básicas, como título, descrição, termos de serviço, versão da documentação e contato. No código, são as linhas do primeiro bloco após a especificação. Na visualização, do lado direito, as informações aparecem no topo da página.

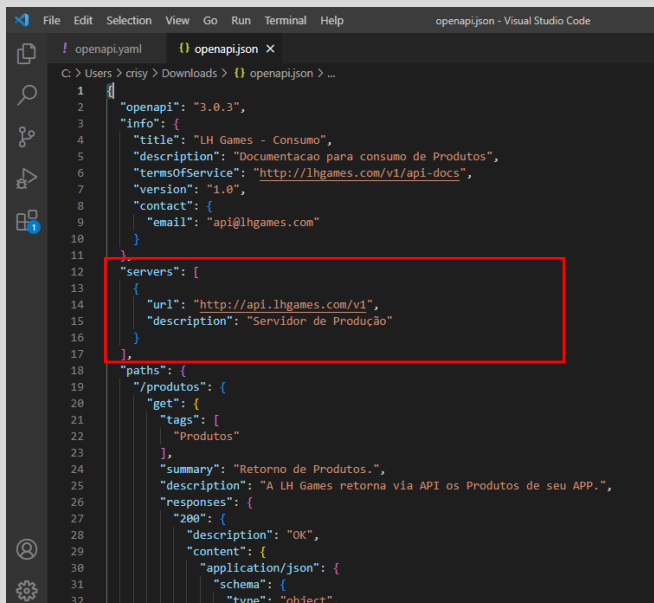
The image shows a side-by-side comparison of an OpenAPI specification file and its rendered UI. On the left, the VS Code editor displays the 'openapi.yaml' file. A red box highlights the 'info' section, which contains the title 'LH Games - Consumo', description 'Documentacao para consumo de Produtos', terms of service 'http://lhgames.com/v1/api-docs', version '1.0', and contact email 'api@lhgames.com'. On the right, the OpenAPI Editor visualizes this information. The title 'LH Games - Consumo' is prominently displayed with a '1.0' version badge and an 'OAS3' specification version indicator. Below the title, the description and links for 'Terms of service' and 'Contact the developer' are shown. The 'Servers' section lists the production server 'http://api.lhgames.com/v1 - Servidor de Produção'. The 'Produtos' section lists two endpoints: a GET endpoint for '/produtos' (Retorno de Produtos) and a POST endpoint for '/produtos' (Adicionar novo Produto).



### 3) Servers

```
"servers": [  
  {  
    "url": "http://api.lhgames.com/v1",  
    "description": "Servidor de Produção"  
  }  
],
```

Você pode fazer uma lista de servidores, separando o ambiente de testes da visualização do cliente, por exemplo. No código, do lado esquerdo, a lista de servidores está no bloco seguinte ao bloco da info. Na visualização, do lado direito, os servidores estão entre o topo e as paths, dentro de um menu drop-down.



```
1  "openapi": "3.0.3",  
2  
3  "info": {  
4    "title": "LH Games - Consumo",  
5    "description": "Documentação para consumo de Produtos",  
6    "termsOfService": "http://lhgames.com/v1/api-docs",  
7    "version": "1.0",  
8    "contact": {  
9      "email": "api@lhgames.com"  
10   }  
11 },  
12 "servers": [  
13   {  
14     "url": "http://api.lhgames.com/v1",  
15     "description": "Servidor de Produção"  
16   }  
17 ],  
18 "paths": {  
19   "/produtos": {  
20     "get": {  
21       "tags": [  
22         "Produtos"  
23       ],  
24       "summary": "Retorno de Produtos.",  
25       "description": "A LH Games retorna via API os Produtos de seu APP.",  
26       "responses": {  
27         "200": {  
28           "description": "OK",  
29           "content": {  
30             "application/json": {  
31               "schema": {  
32                 "type": "object",
```



## 4) paths

Os paths são as urls das ações da sua API que poderão ser consumidas. Confira o código das paths dentro do código completo, no final desse conteúdo.

No exemplo, dentro da rota /produtos e com a mesma tag Produtos, temos um método get e um método post.

# LH Games - Consumo

1.0 OAS3

Documentacao para consumo de Produtos

[Terms of service](#)

[Contact the developer](#)

Servers

http://api.lhgames.com/v1 - Servidor de Produção

## Produtos

GET

/produtos

Retorno de Produtos.

✓


POST

/produtos

Adicionar novo Produto

✓

## GET


**GET** **/produtos** Retorno de Produtos. 

ALH Games retorna via API os Produtos de seu APP.

**Parameters** [Try it out](#)

No parameters

**Responses**


Code	Description	Links
200	OK	No links
<div>Media type</div> <div>application/json </div> <div>Controls Accept header.</div> <div>Example Value   Schema</div> <pre>{   "id": 1,   "nome": "God of War",   "foto": ".\\capa1.jpg",   "descricao": "aventura",   "preco": 150 }</pre>		
404	API não encontrada	No links
405	Consumo sem permissão	No links

O get tem como summary (título) o Retorno de Produtos e como description uma breve descrição do funcionamento da API. Não foram configurados parâmetros para esse método.

O método get é usado para obter dados, portanto a API vai retornar informações sobre um produto. Essas informações, como nome, foto, descrição e preço, serão organizadas em um modelo de dados chamado schema. Os valores dos dados estruturados dentro do schema vêm do arquivo dbgames.db, que é o banco de dados fornecido para consulta.

Em caso de erro, o sistema retorna apenas uma mensagem de erro. Se o erro for o 404, a API retorna a mensagem “API não encontrada”. Se o erro for 405, a API retorna a mensagem “Consumo sem permissão”.


## POST

**POST** `/produtos` Adicionar novo Produto 

A LH Games recebe via API post os Produtos de seu APP.

**Parameters** Try it out

No parameters

Request body application/json 

**Example Value** | **Schema**

```
{
  "id": 1,
  "nome": "God of War"
}
```

**Responses**

Code	Description	Links
200	OK	No links

O post tem como summary (título) Adicionar novo Produto e como description uma breve descrição do funcionamento da API. Não foram configurados parâmetros para esse método.

O método post é usado para enviar dados, portanto a API vai enviar informações sobre um produto. Essas informações, como nome e id serão organizadas também em schema. Os valores enviados que poderão ser inseridos no banco de dados pela programação backend.

Confira a seguir o código completo do arquivo openapi.json.

```
{
  "openapi": "3.0.3",
  "info": {
    "title": "LH Games - Consumo",
    "description": "Documentacao para consumo de Produtos",
    "termsOfService": "http://lhgames.com/v1/api-docs",
    "version": "1.0",
    "contact": {
      "email": "api@lhgames.com"
    }
  },
  "servers": [
    {
      "url": "http://api.lhgames.com/v1",
      "description": "Servidor de Produção"
    }
  ],
}
```

```
"paths": {
  "/produtos": {
    "get": {
      "tags": [
        "Produtos"
      ],
      "summary": "Retorno de Produtos.",
      "description": "A LH Games retorna via API os Produtos de seu APP.",
      "responses": {
        "200": {
          "description": "OK",
          "content": {
            "application/json": {
              "schema": {
                "type": "object",
                "properties": {
                  "id": {
                    "type": "integer",
                    "format": "int64",
                    "example": 1
                  },
                  "nome": {
                    "type": "string",
                    "example": "God of War"
                  },
                  "foto": {
                    "type": "string",
                    "example": ".\\capa1.jpg"
                  },
                  "descricao": {
                    "type": "string",
                    "example": "aventura"
                  },
                  "preco": {
                    "type": "number",
                    "format": "float",
                    "example": 150
                  }
                }
              }
            }
          }
        },
        "404": {
          "description": "API não encontrada"
        },
        "405": {
          "description": "Consumo sem permissão"
        }
      }
    }
  }
},
```



```
"post": {
  "tags": [
    "Produtos"
  ],
  "summary": "Adicionar novo Produto",
  "description": "A LH Games recebe via API post os Produtos de seu APP.",
  "requestBody": {
    "content": {
      "application/json": {
        "schema": {
          "type": "object",
          "properties": {
            "id": {
              "type": "integer",
              "format": "int64",
              "example": 1
            },
            "nome": {
              "type": "string",
              "example": "God of War"
            }
          }
        }
      }
    }
  },
  "responses": {
    "200": {
      "description": "OK"
    }
  }
}
```