Instituto Federal do Norte de Minas Gerais - Campus Januária

ASGI e FastAPI

Sistemas Distribuídos

Geovana de Sousa
Milena Mota

WSGI

01 **Atua como intermediário entre a aplicação Python e o Servidor.**

02 **Funções Síncronas.**

03 **Sua limitação pode causar problemas de escalabilidade.**

ASGI

01 **Introduzido em reposta ao gargalo de WSGI**

02 **Função Assíncrona Única e chamável.**

Funcionamento do ASGI

01 *scope:* dicionário (dict) que contém detalhes sobre a conexão específica.

02 *send:* função assíncrona que permite à aplicação enviar mensagens de eventos para o cliente.

03 *receive:* função assíncrona que permite à aplicação receber mensagens de eventos do cliente.

04

```
async def application(scope, receive, send):
    event = await receive()
    ...
    await send({"type": "websocket.send", ...})
```

API Síncrona

01 Executam requisições em um fluxo simples.

02 *É bloqueante.*

03 *O uso ideal desse tipo de API é em cenários que precisem de retorno imediato, como por exemplo, representações de banco de dados*

Fluxo de trabalho Síncrono

- *Request.*
- *Processing.*
- *Response.*

API Síncrona

Vantagens

Fácil Implementação

Geralmente custam menos para desenvolver e manter.

Maior Precisão

Sua sincronização resulta em uma maior precisão para muitas aplicações, já que o cliente é imediatamente informado sobre a sua requisição

API Síncrona

Limitações

Bloqueio do Cliente

O cliente está bloqueado de fazer qualquer outra requisição.

Escalabilidade Limitada

O modelo síncrono é ineficiente no gerenciamento de recursos do servidor, especialmente em tarefas longas.

API Assíncrona

01 Gestão de Longa Duração.

02 *O Conceito de Não-Bloqueio*

03 *Processamento no Próprio Tempo do Servidor*

Fluxo de trabalho Assíncrono

- *Request.*
- *Acknowledgement.*
- *Processing.*
- *Response.*

API Assíncrona

Vantagens

Eficiência

A ausência de espera nos fluxos assíncronos resulta em serviços mais eficientes

Escalabilidade

O servidor não prende threads esperando I/O.

API Assíncrona

Limitações

Custo elevado

o código necessário para implementá-los muitas vezes introduz custos que não escalam bem com ecossistemas mais complexos.

A resposta não é imediata

Sistemas assíncronos também não fornecem respostas em tempo real, o que significa que estão em desvantagem para sistemas que exigem interatividade imediata.

Gunicorn x Uvicorn

- WSGI
- Flask, Django
- Multiple worker processes
- Eficiente

- ASGI
- AsyncIO
- Single-process
- Limitado

Lib FastAPI

- Framework web para a criação de APIs com Python
- Recursos assíncronos do Python 3.7+
- Otimiza o desenvolvimento com tipagem estática e validação automática de dados, facilitando a implementação rápida e escalável de APIs.

Lib FastAPI

Vantagens

Alto desempenho

Devido ao seu suporte nativo para programação assíncrona (async/await) e ao uso do servidor ASGI Uvicorn

Documentação Automática

Gera automaticamente documentação interativa da API, compatível com os padrões OpenAPI e JSON Schema

Validação de Dados

Utiliza o Pydantic para validação e serialização automática de dados.

Lib FastAPI

Vantagens

Tipagem Estática

Suporta totalmente as type hints do Python, o que permite um melhor suporte de IDEs (IntelliSense/autocompletar) e facilita a detecção de erros antes da execução do código.

Injeção de Dependências

Simplifica a organização do código e o gerenciamento de conexões com bancos de dados, autenticação, entre outros.

Lib FastAPI

Limitações

Ausência de Componentes Nativos

Diferente do Django, o FastAPI não vem com um ORM (Mapeamento Objeto-Relacional), painel administrativo, sistema de autenticação de usuários completo integrado.

Ecossistema e comunidade

Ecossistema menor que o de Django e Flask, com menos extensões “oficiais”. Muitas integrações existem via bibliotecas de terceiros ou código próprio, o que exige mais decisões de arquitetura e manutenção.

Lib FastAPI

Limitações

Acoplamento Forte com Pydantic e Starlette

O FastAPI espera classes do Pydantic para funcionar corretamente e gerar a documentação automática (Swagger). Por isso, usar outras bibliotecas de serialização pode se tornar uma tarefa difícil

Complexidade do código assíncrono

É importante ter o cuidado no uso de bibliotecas síncronas, para evitar bloqueio e perda de performance.

Lib FastAPI

Instalação

- O FastAPI requer o Python 3.7+.
- A maneira recomendada de instalar o FastAPI é com as dependências adicionais.

```
$ pip install "fastapi[standard]"
```

```
$ pip install fastapi uvicorn
```

Lib FastAPI

```python
from fastapi import FastAPI
from pydantic import BaseModel

app = FastAPI()

books = [
    {"id": 1, "title": "Python Basics", "author": "Real P.", "pages": 635},
    {"id": 2, "title": "Breaking the Rules", "author": "Stephen G.", "pages": 99},
]

class Book(BaseModel):
    title: str
    author: str
    pages: int

@app.get("/")
def home():
    return {"message": "Welcome to the Book API!"}

@app.get("/books")
def get_books(limit: int | None = None):
    if limit:
        return {"books": books[:limit]}
    return {"books": books}

```

```
$ uvicorn <nome_arquivo>:<nome_app> --reload
```

Estrutura básica

Importações das bibliotecas

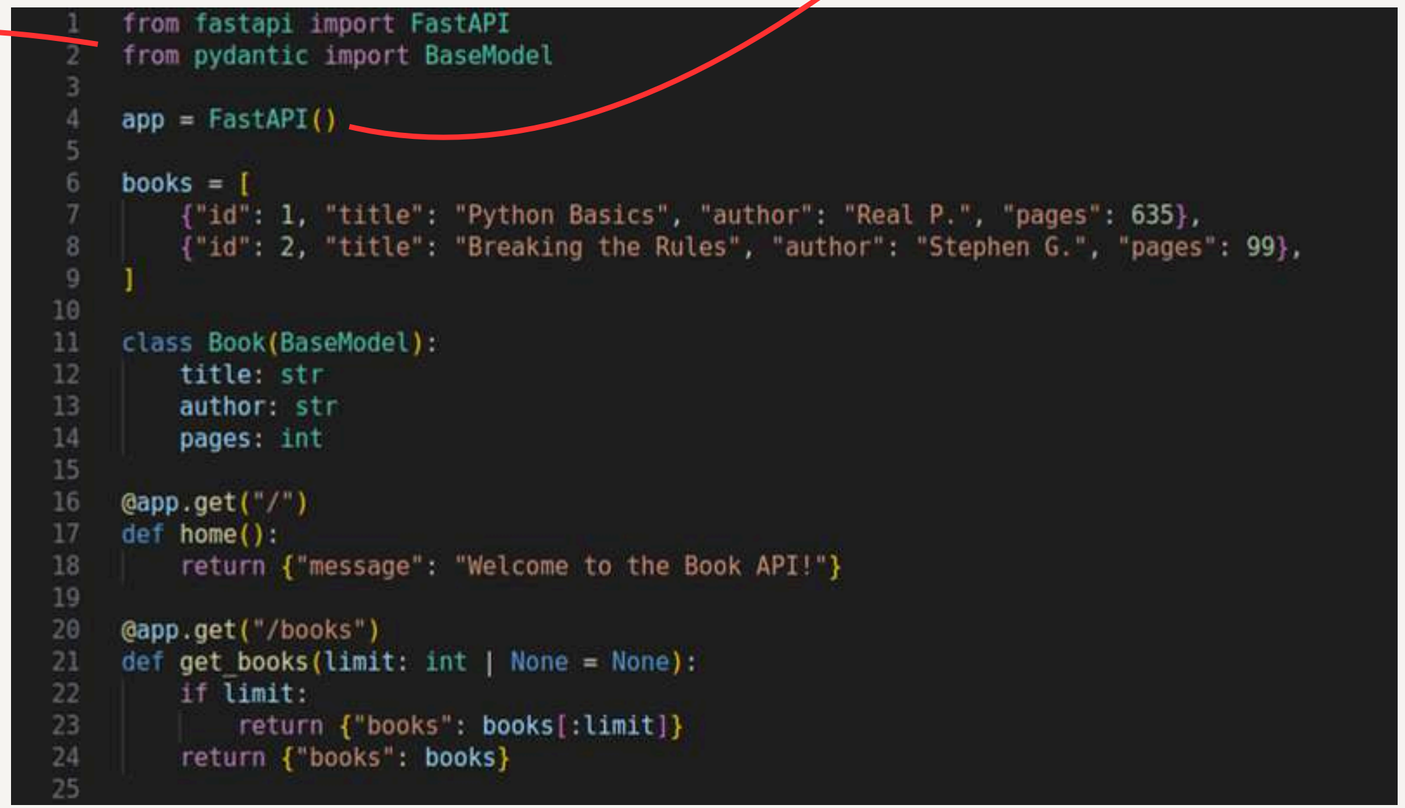
```python
from fastapi import FastAPI
from pydantic import BaseModel

app = FastAPI()

books = [
    {"id": 1, "title": "Python Basics", "author": "Real P.", "pages": 635},
    {"id": 2, "title": "Breaking the Rules", "author": "Stephen G.", "pages": 99},
]

class Book(BaseModel):
    title: str
    author: str
    pages: int

@app.get("/")
def home():
    return {"message": "Welcome to the Book API!"}

@app.get("/books")
def get_books(limit: int | None = None):
    if limit:
        return {"books": books[:limit]}
    return {"books": books}

```

Estrutura básica

Importações das bibliotecas

Instância do app

É nela que o Uvicorn vai procurar para rodar o app



```python
from fastapi import FastAPI
from pydantic import BaseModel

app = FastAPI()

books = [
    {"id": 1, "title": "Python Basics", "author": "Real P.", "pages": 635},
    {"id": 2, "title": "Breaking the Rules", "author": "Stephen G.", "pages": 99},
]

class Book(BaseModel):
    title: str
    author: str
    pages: int

@app.get("/")
def home():
    return {"message": "Welcome to the Book API!"}

@app.get("/books")
def get_books(limit: int | None = None):
    if limit:
        return {"books": books[:limit]}
    return {"books": books}

```

Estrutura básica

Instância do app

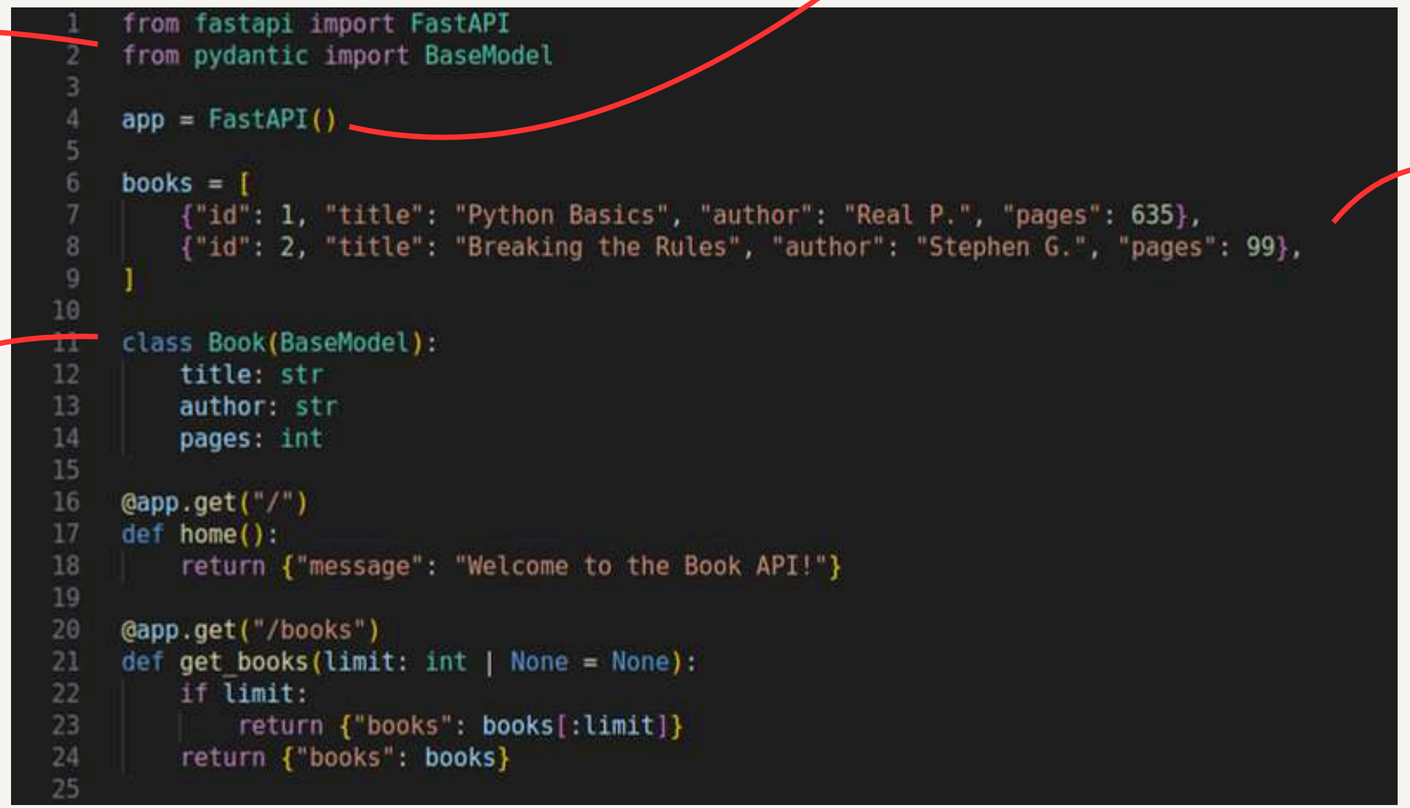
É nela que o Uvicorn vai procurar para rodar o app

Importações das bibliotecas

Simulação de Banco de dados

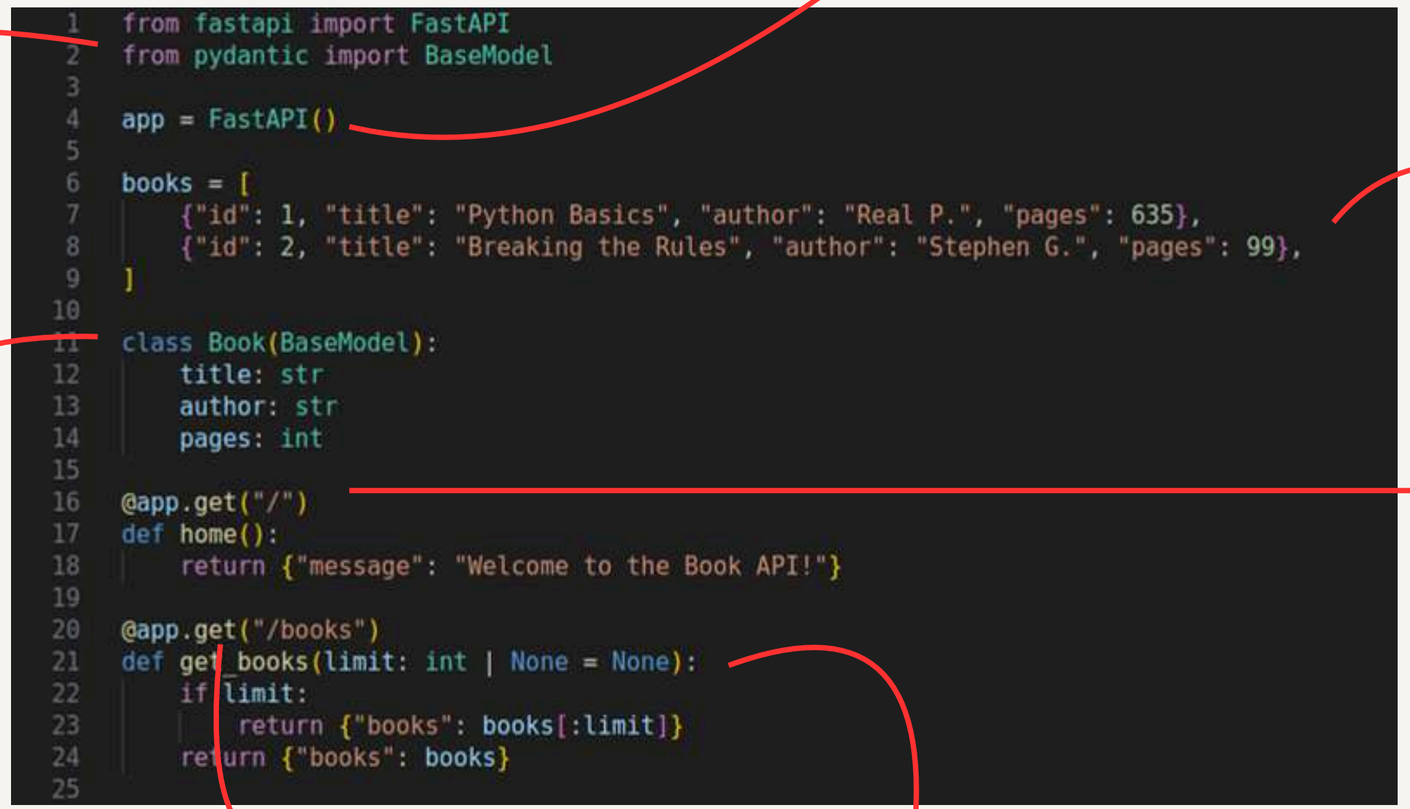
Modelo de dados

Define o que é um dado válido separadamente



```python
from fastapi import FastAPI
from pydantic import BaseModel

app = FastAPI()

books = [
    {"id": 1, "title": "Python Basics", "author": "Real P.", "pages": 635},
    {"id": 2, "title": "Breaking the Rules", "author": "Stephen G.", "pages": 99},
]

class Book(BaseModel):
    title: str
    author: str
    pages: int

@app.get("/")
def home():
    return {"message": "Welcome to the Book API!"}

@app.get("/books")
def get_books(limit: int | None = None):
    if limit:
        return {"books": books[:limit]}
    return {"books": books}

```

Estrutura básica



É nela que o Uvicorn vai procurar para rodar o app

Instância do app

Importações das bibliotecas

Simulação de Banco de dados

Modelo de dados

Rotas (Path Operations)

Define o que é um dado válido separadamente

Método HTTP

Type Hinting

Corrotinas assíncronas (asyncio)

- O FastAPI é construído sobre o Starlette, um framework ASGI que suporta concorrência por meio do loop de eventos asyncio do Python.
- Relembrando...
  - Evite I/O bloqueante
  - Use drivers de banco de dados assíncronos
  - Limite tarefas de longa duração
  - Teste rotas assíncronas

SWAGGER

- Uma das características mais marcantes do FastAPI é sua documentação automática da API.
- Com o servidor ainda em execução, abra o navegador e acesse http://127.0.0.1:8000/docs ou http://127.0.0.1:8000/redoc.
- O Swagger UI exibe todos os seus endpoints, seus métodos HTTP e os formatos esperados de requisição e resposta.
- O ReDoc apresenta as mesmas informações, mas em um formato diferente.

Instituto Federal do Norte de Minas Gerais - Campus Januária

Obrigada

1 Dez 2025

Sistemas Distribuídos

Geovana de Sousa
Milena Mota