

SUMÁRIO

O QUE VEM POR AÍ?	3
HANDS ON	4
SAIBA MAIS.....	5
O QUE VOCÊ VIU NESTA AULA?	10
REFERÊNCIAS.....	11

EMSE

O QUE VEM POR AÍ?

Agora que você ampliou sua visão sobre ferramentas e metodologias no desenvolvimento de APIs, é hora de conhecer o FastAPI, um framework moderno, rápido e intuitivo para criar serviços web. Nesta aula, você verá como o FastAPI facilita a criação de rotas, a validação de dados e a autenticação, além de permitir a integração de tarefas mais complexas, como scraping de dados da web. Tudo isso é feito de maneira clara, reduzindo o tempo de desenvolvimento e aumentando a produtividade.

Imagine criar uma API que, além de armazenar e gerenciar dados, seja capaz de acessar conteúdo externo e extrair informações relevantes. E melhor ainda: com rotas protegidas por autenticação, garantindo que apenas usuários autorizados possam acessar recursos sensíveis. Ao final desta aula, você terá entendido como unir esses elementos em um só projeto, dominando as noções fundamentais do FastAPI.

Prepare-se para as videoaulas e não hesite em revisar este material sempre que precisar!

HANDS ON

Nos vídeos desta seção, você acompanhará a criação de uma aplicação baseada em FastAPI, estudando funcionalidades essenciais e introduzindo recursos mais avançados. As videoaulas mostrarão passo a passo como iniciar um projeto, criar rotas, proteger endpoints com autenticação básica, lidar com dados em memória e realizar scraping de páginas web, tudo integrado de forma simples e organizada.

Este é o momento de “colocar a mão na massa”: abra seu terminal, siga as instruções e experimente cada etapa. Aproveite a possibilidade de pausar, voltar e refazê-las, garantindo que compreenda plenamente os conceitos apresentados.

Aperte o play, acompanhe as demonstrações e, após terminar, volte aqui para consolidar seu aprendizado!

SAIBA MAIS

Nesta seção, vamos estudar mais a fundo os conceitos e ferramentas apresentados. Você verá a lógica por trás do FastAPI, entenderá por que o uso de Pydantic é tão natural neste framework, aprenderá a lidar com autenticação básica e descobrirá como integrar tarefas de scraping em um contexto de APIs.

FastAPI: Por Que Está em Alta?



Figura 1 - Logo do FastAPI
Fonte: Google Imagens (2025)

O FastAPI é um framework para criação de APIs em Python que ganhou popularidade devido à sua velocidade, simplicidade e extensibilidade. Ele utiliza recursos modernos do Python, como tipagem estática, para gerar documentação automática e validação de dados praticamente sem esforço adicional da pessoa desenvolvedora.

Benefícios do FastAPI:

- Performance próxima à do Node.js e Go, graças ao uso do ASGI e do servidor uvicorn.
- Documentação automática interativa, gerada a partir de anotações de tipos e docstrings.
- Suporte nativo a validação de dados com Pydantic, reduzindo bugs e erros.

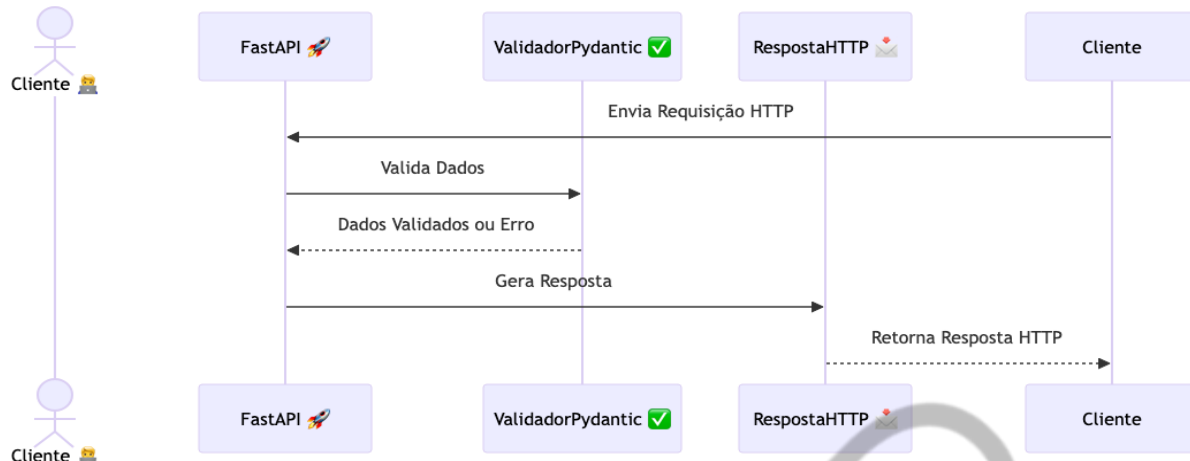


Figura 2 - Fluxo de Requisições no FastAPI
Fonte: Elaborado pelo Autor (2025)

Abra sua aplicação em <http://127.0.0.1:8000/docs> e veja a documentação interativa gerada automaticamente!

Validação e Tipagem com Pydantic

O Pydantic é um dos segredos do FastAPI. Ao definir modelos de dados com Pydantic, você descreve quais campos são esperados, seus tipos e requisitos. Quando a requisição chega, o FastAPI usa esses modelos para validar e converter dados, garantindo segurança e coerência.

Exemplo de Modelo Pydantic:

```

from pydantic import BaseModel

class Item(BaseModel):
    name: str
    description: str | None = None
    price: float | None = None
    quantity: int | None = None
  
```

Código-fonte 1 – Exemplo de Modelo Pydantic
Fonte: Elaborado pelo autor (2025)

Ao receber um POST com dados incorretos (por exemplo, price como texto não-numérico), o FastAPI retorna automaticamente uma resposta 422 Unprocessable Entity, indicando que os dados não passaram na validação.

Experimente enviar dados inválidos a um endpoint e veja como o FastAPI responde!

Autenticação Básica e Outros Métodos

Embora o JWT seja muito usado, a autenticação básica (HTTP Basic) também pode ser suficiente em cenários simples ou internos. Neste caso, o cliente envia o usuário e senha codificados em Base64. O servidor valida as credenciais e, se corretas, concede acesso aos recursos.

Outros métodos de autenticação (como OAuth2, OpenID Connect) podem ser **fácilmente** integrados ao **FastAPI**. É tudo uma questão de escolher o esquema certo para seu contexto.

MÉTODO	VANTAGEM	USO COMUM
HTTP Basic	Simples, rápido de implementar.	Ambientes internos.
Bearer Token/JWT	Sessões stateless, escalável.	Serviços públicos e apps.
OAuth2	Integração com terceiros.	Login com Google, Github.

Tabela 1 - Métodos de Autenticação
Fonte: Elaborado pelo Autor (2025)

Avalie se seu projeto precisa de algo mais complexo que Basic Auth. Em caso positivo, teste OAuth2 no seu protótipo!

CRUD: Armazenamento Simples

O exemplo da aula mostra rotas CRUD (Create, Read, Update, Delete) que armazenam itens em memória. Mesmo sem um banco de dados externo, é fácil entender a dinâmica: o GET retorna itens, o POST cria, o PUT atualiza e o DELETE remove. Em produção, você usaria um banco real, mas a lógica básica permanece:

- **Criar (POST):** recebe um modelo de dados, valida e insere.
- **Listar (GET):** retorna todos ou um subconjunto de dados.
- **Atualizar (PUT):** altera dados existentes com base em um identificador.
- **Remover (DELETE):** deleta o recurso especificado.

Teste criar, atualizar e deletar itens e observe a resposta do servidor!

Integrando Selenium para Scraping



Figura 3 - Logo do Selenium
Fonte: Google Imagens (2025)

O **Selenium** é uma ferramenta que permite interagir com navegadores de forma automatizada. Integrar Selenium ao FastAPI cria endpoints que “navegam” em páginas externas e extraem títulos, cabeçalhos, parágrafos e links.

Por que usar Selenium?

- Acesso a páginas dinâmicas, renderizadas por JavaScript, impossíveis de serem extraídas com requests + BeautifulSoup sozinhos.
- Simulação de um usuário real clicando e navegando.

No exemplo, a configuração headless (sem interface gráfica) torna o processo mais leve e adequado a ambientes de produção. É uma forma criativa de enriquecer sua API, permitindo que clientes obtenham dados de outros sites sem precisar conhecer a lógica interna do scraping.

Experimente modificar as funções de scraping para extrair informações diferentes, como imagens ou meta tags!

Boas Práticas e Performance

O FastAPI é rápido e escalável. Para maximizar isso:

- Use async/await: tornar as rotas assíncronas evita bloqueios, aumentando o desempenho.
- Mantenha dependências separadas e organizadas.

- Em produção, combine FastAPI com servidores como uvicorn ou gunicorn e configure workers, balanceadores de carga e cache.

Experimente rodar múltiplas instâncias do uvicorn para testar escalabilidade!

Referências e Leituras Sugeridas

- [Documentação do FastAPI](#).
- [Pydantic Documentation](#).
- [Selenium Docs](#).
- Livros recomendados: FastAPI Cookbook (Dennis Gurock) e High Performance Python (Micha Gorelick, Ian Ozsvald).

Consulte a documentação oficial e tente implementar uma funcionalidade nova após ler o material recomendado!

O QUE VOCÊ VIU NESTA AULA?

Nesta aula, você aprendeu os fundamentos do FastAPI, sua velocidade e facilidade de uso; como utilizar Pydantic para validação automática de dados; implementar autenticação básica e estruturar rotas com segurança; criar endpoints CRUD, armazenar e manipular dados; e integrar Selenium para scraping, agregando funcionalidades avançadas à sua API.

Relembre os conceitos, assista às videoaulas novamente, coloque-os em prática e avance com confiança para os próximos passos!

REFERÊNCIAS

GORELICK, M.; OZSVALD, I. **High Performance Python**: Practical Performant Programming for Humans. [s.l.]: O'Reilly Media, 2020.

JOHNSON, D.; LEE, E. Scalable Machine Learning Deployments with APIs. **Journal of Applied AI**, v. 12, n. 3, 2019, p. 45-58.

SMITH, A.; JOHNSON, B.; WILLIAMS, C. **APIs in Machine Learning**: Bridging the Gap Between Models and Applications. [s.l.]: Tech Publishers, 2020.

PALAVRAS-CHAVE

Palavras-chave: APIs. Machine Learning. Escalabilidade.

EMSE



POSTECH