

Questão 4

Enunciado: Utilizando uma ferramenta de IA generativa, crie um endpoint com FastAPI que valide e processe a entrada de um objeto do tipo `Item`. Siga as especificações abaixo:

1. Estrutura do `Item`:
 1. **nome**: `string` com tamanho máximo de 25 caracteres.
 2. **valor**: `float`
 3. **data**: valor do tipo `date`, que não pode ser superior à data atual.
2. Requisitos
 1. O endpoint deve validar os valores recebidos.
 2. Após a validação, o corpo da requisição (`Item`) deve ser retornado com um novo campo adicional: `uuid`. Este campo deve conter um identificador único gerado dinamicamente.

Prompt: Crie um endpoint com FastAPI para processar um objeto `Item` com as seguintes características:

1. O `Item` possui os campos:
 - **nome**: `string` (máximo de 25 caracteres)
 - **valor**: `float`
 - **data**: `date` (não pode ser superior à data atual)
2. O endpoint deve validar a entrada e retornar o mesmo objeto com um campo adicional: `uuid` (identificador único gerado dinamicamente).
3. Use Pydantic para validação e inclua mensagens de erro apropriadas."

Resposta do ChatGPT:

 ChatGPT

Crie um endpoint com FastAPI para processar um objeto Item com as seguintes características:

O Item possui os campos:

nome: string (máximo de 25 caracteres)

valor: float

data: date (não pode ser superior à data atual)

O endpoint deve validar a entrada e retornar o mesmo objeto com um campo adicional: uuid (identificador único gerado dinamicamente).

Use Pydantic para validação e inclua mensagens de erro apropriadas.*

Para criar um endpoint com FastAPI que atenda aos requisitos, podemos usar o Pydantic para validação dos campos e o `uuid` para gerar um identificador único. Vou te mostrar como fazer isso.

Passos:

1. Instalar as dependências necessárias:

- `fastapi` e `uvicorn` para o servidor.
- `pydantic` para validação de dados.

Bash

 Copiar código

```
pip install fastapi uvicorn pydantic
```



Mensagem para o ChatGPT



ChatGPT



Bash

 Copiar código

```
pip install fastapi uvicorn pydantic
```

1. Criar o arquivo Python com a estrutura do endpoint.

Exemplo de código:

Python

 Copiar código

```
from fastapi import FastAPI, HTTPException
from pydantic import BaseModel, constr, condecimal
from datetime import date
from uuid import uuid4
from typing import Optional

# Criação do app FastAPI
app = FastAPI()

# Definindo o modelo do Item com validações
class Item(BaseModel):
    nome: constr(max_length=25) # Nome com até 25
    caracteres
    valor: condecimal(gt=0) # Valor deve ser maior
    que 0
    data: date # Data no formato 'yyyy-mm-dd'

    # Método para validar a data não ser superior à
    data atual
    @classmethod
    def validate_data(cls, data):
        if data > date.today():
            raise ValueError("A data não pode ser
superior à data atual.")
        return data

    # Aplica a validação para o campo data
    @classmethod
    def __get_validators__(cls):
        yield cls.validate_data

# Endpoint para criar um item
@app.post("/items/", response_model=Item)
```



Mensagem para o ChatGPT



 ChatGPT

```
@app.post("/itens/", response_model=Item)
async def criar_item(item: Item):
    # Gerando o UUID para o item
    item_dict = item.dict()
    item_dict["uuid"] = str(uuid4()) # Adiciona o
    campo 'uuid' ao item

    return item_dict

# Para rodar o app, use o comando abaixo no terminal
# uvicorn nome_do_arquivo:app --reload
```

Explicação do código:

1. Modelo Item:

- A classe `Item` é definida usando o Pydantic. Ela possui três campos: `nome`, `valor` e `data`, cada um com validação específica.
 - ▶ O `nome` deve ter no máximo 25 caracteres (`constr(max_length=25)`).
 - ▶ O `valor` deve ser um número decimal positivo (`condecimal(gt=0)`).
 - ▶ A `data` é validada para garantir que não seja superior à data atual. A validação da data é feita dentro de um método de classe, onde comparamos a data enviada com a data atual.

2. Endpoint `/itens/`:

- O endpoint recebe um objeto do tipo `Item` e, após validar os dados, gera um identificador único (`uuid`), que é retornado junto com os dados originais do item.

3. Validação de data:



Mensagem para o ChatGPT



ChatGPT



3. Validação de data:

- O campo `data` é validado para garantir que não seja superior à data atual. Caso seja, uma exceção `HTTPException` é lançada com a mensagem de erro apropriada.

Testando o endpoint:

1. Para rodar a aplicação, execute o seguinte comando no terminal:

Bash

 Copiar código

```
uvicorn nome_do_arquivo:app --reload
```

1. Acesse a documentação interativa do FastAPI em <http://127.0.0.1:8000/docs>.
2. Você pode testar o endpoint `/itens/` enviando um JSON com os campos `nome`, `valor` e `data`.

Exemplo de entrada:

Json

 Copiar código

```
{  
  "nome": "Produto Exemplo",  
  "valor": 99.99,  
  "data": "2025-03-05"  
}
```

Exemplo de saída:



Mensagem para o ChatGPT



ChatGPT



Bash

 Copiar código

```
uvicorn nome_do_arquivo:app --reload
```

1. Acesse a documentação interativa do FastAPI em <http://127.0.0.1:8000/docs>.
2. Você pode testar o endpoint `/itens/` enviando um JSON com os campos `nome`, `valor` e `data`.

Exemplo de entrada:

Json

 Copiar código

```
{
  "nome": "Produto Exemplo",
  "valor": 99.99,
  "data": "2025-03-05"
}
```

Exemplo de saída:

Json

 Copiar código

```
{
  "nome": "Produto Exemplo",
  "valor": 99.99,
  "data": "2025-03-05",
  "uuid": "dfe9ff6b-022a-4c80-86d4-e8bdbed39680"
}
```

Se você enviar uma data superior à data atual ou um valor inválido, o FastAPI retornará uma mensagem de erro clara e apropriada.



Mensagem para o ChatGPT

