Pydantic

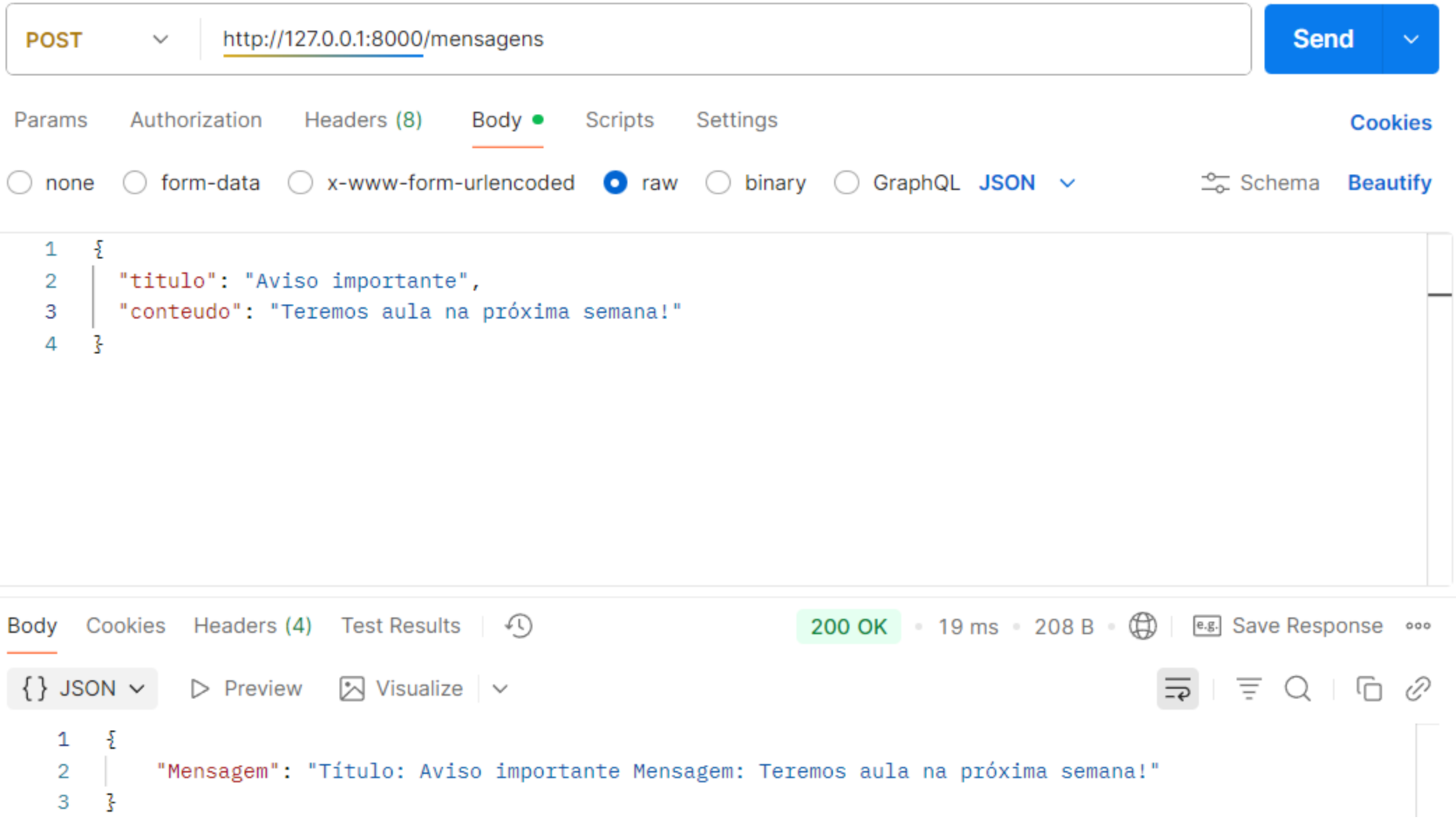
Pydantic

- É uma biblioteca do Python usada para **validar** e **converter dados automaticamente** com base em modelos definidos por **classes**.
- Ela garante que os dados recebidos (por exemplo, de uma requisição HTTP) tenham o formato e o tipo **corretos**.

Vamos modificar a rota mensagens (Post)

```python
from fastapi import FastAPI
from fastapi.params import Body
from pydantic import BaseModel

app = FastAPI()

class Mensagem(BaseModel):
 titulo: str
 conteudo: str

@app.get("/")
async def root():
    return {"message": "Hello World"}

@app.post("/mensagens")
def criar_mensagem(nova_mensagem: Mensagem):
 print(nova_mensagem)
 return {"Mensagem": f"Título: {nova_mensagem.titulo} Mensagem: {nova_mensagem.conteudo}"}
```

Postman

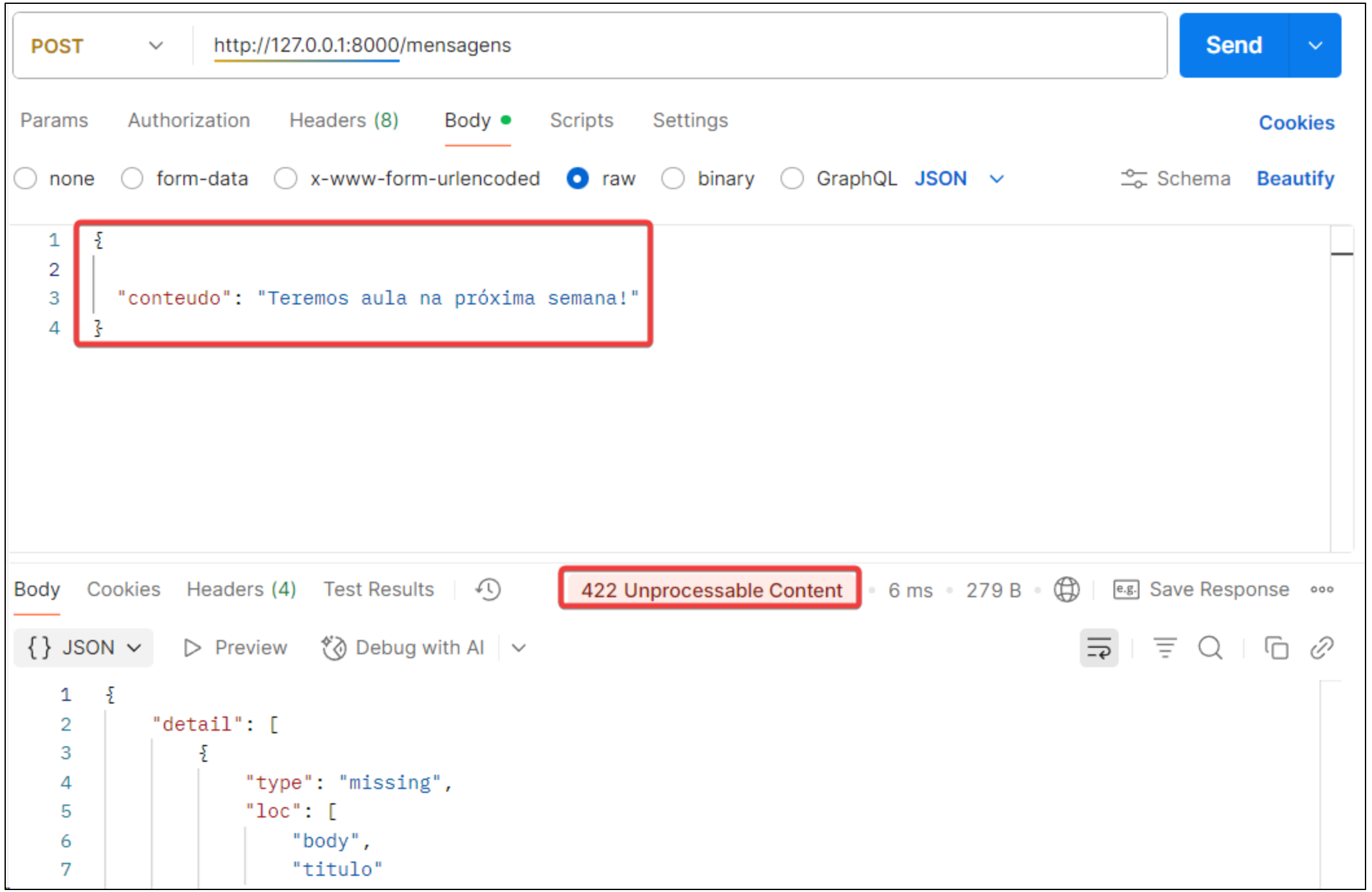

POST
http://127.0.0.1:8000/mensagens
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Schema
Beautify
1 {
2 "titulo": "Aviso importante",
3 "conteudo": "Teremos aula na próxima semana!"
4 }
Body
Cookies
Headers (4)
Test Results
200 OK
19 ms
208 B
Save Response
{} JSON
Preview
Visualize
1 {
2 "Mensagem": "Título: Aviso importante Mensagem: Teremos aula na próxima semana!"
3 }

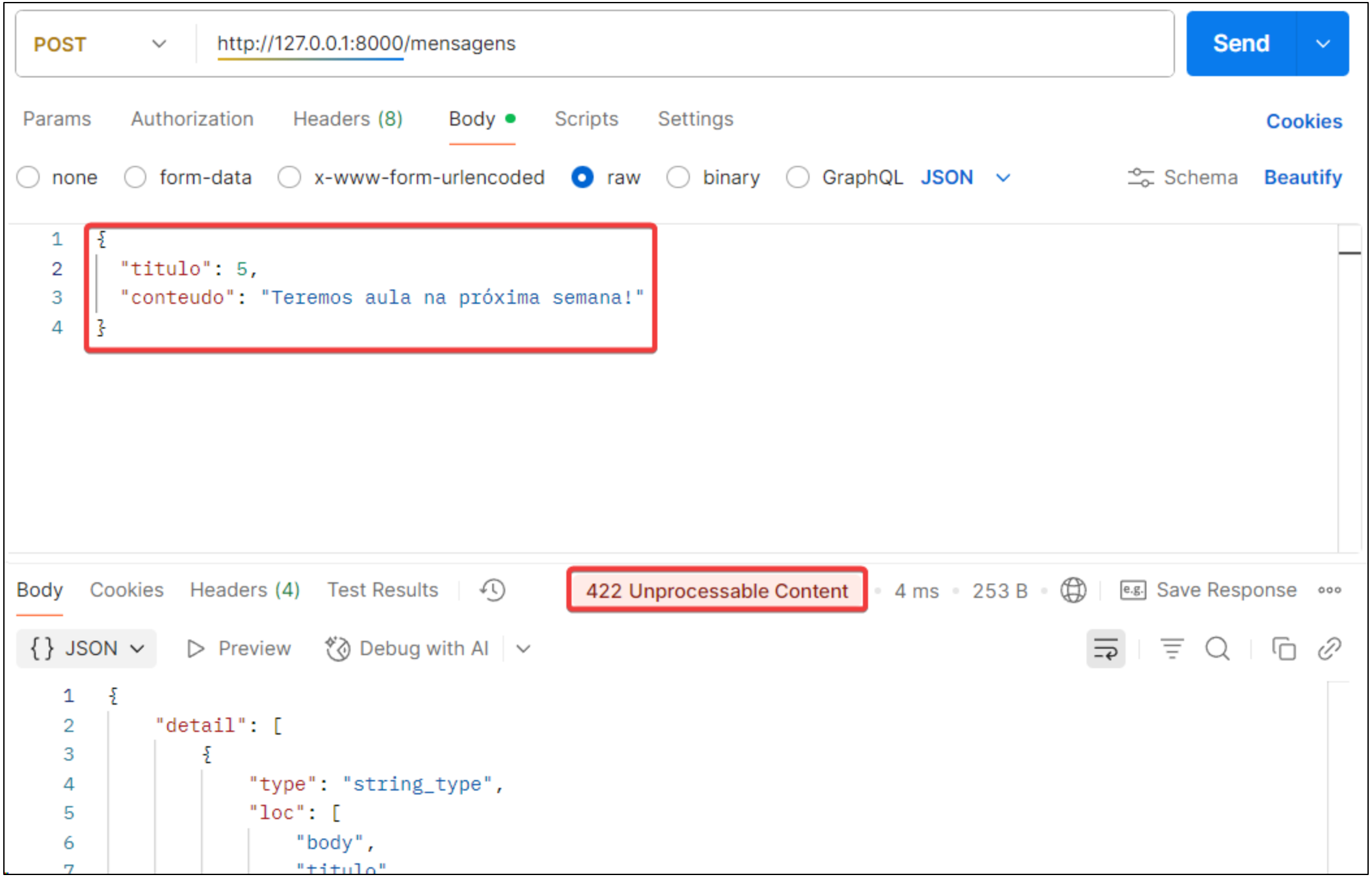
Vamos modificar a rota mensagens (Post)

```python
from pydantic import BaseModel

class Mensagem(BaseModel):
 titulo: str
 conteudo: str

@app.post("/mensagens")
def criar_mensagem(nova_mensagem: Mensagem):
 print(nova_mensagem)
 return {"Mensagem": f"Título: {nova_mensagem.titulo} Mensagem: {nova_mensagem.conteudo}"}
```

- Aqui é criada uma classe modelo chamada **Mensagem**, que herda de **BaseModel**.
- Ela define a estrutura esperada dos dados que serão enviados para a API:
  - **titulo**: deve ser uma **string**
  - **conteudo**: deve ser uma **string**
- O Pydantic verifica automaticamente se os dados recebidos têm esses campos e tipos corretos.

O FastAPI automaticamente converte o **JSON** recebido no corpo da requisição em uma instância da classe **Mensagem** armazenado na variável **nova_mensagem**.


POST
http://127.0.0.1:8000/mensagens
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Schema
Beautify
1 {
2
3 "conteudo": "Teremos aula na próxima semana!"
4 }
Body
Cookies
Headers (4)
Test Results
422 Unprocessable Content
6 ms
279 B
Save Response
JSON
Preview
Debug with AI
1 {
2 "detail": [
3 {
4 "type": "missing",
5 "loc": [
6 "body",
7 "titulo"


POST
http://127.0.0.1:8000/mensagens
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Schema
Beautify
1 {
2 "titulo": 5,
3 "conteudo": "Teremos aula na próxima semana!"
4 }
Body
Cookies
Headers (4)
Test Results
422 Unprocessable Content
4 ms
253 B
Save Response
JSON
Preview
Debug with AI
1 {
2 "detail": [
3 {
4 "type": "string_type",
5 "loc": [
6 "body",
7 "titulo"

Vamos organizar o código!

- Criar um arquivo **schemas.py**

```
from pydantic import BaseModel

class Mensagem(BaseModel):
 titulo: str
 conteudo: str
```

Vamos organizar o código!

- **main.py**

```
from fastapi import FastAPI
from fastapi.params import Body
from schemas import Mensagem

app = FastAPI()

@app.get("/")
async def root():
    return {"message": "Hello World"}

@app.post("/mensagens")
def criar_mensagem(nova_mensagem: Mensagem):
 print(nova_mensagem)
 return {"Mensagem": f"Título: {nova_mensagem.titulo} Mensagem: {nova_mensagem.conteudo}"}
```

Criando um Valor Default

- **schemas.py**

```
from pydantic import BaseModel

class Mensagem(BaseModel):
 titulo: str
 conteudo: str
 publicada: bool = True
```

Criando um Valor Default

- **main.py**

```
from fastapi import FastAPI
from fastapi.params import Body
from schemas import Mensagem

app = FastAPI()

@app.get("/")
async def root():
    return {"message": "Hello World"}

@app.post("/mensagens")
def criar_mensagem(nova_mensagem: Mensagem):
 print(nova_mensagem)
 return {"Mensagem": f"Título: {nova_mensagem.titulo} Conteúdo: {nova_mensagem.conteudo}
Publicada: {nova_mensagem.publicada}"}
```

POST http://127.0.0.1:8000/mensagens Send

Params Authorization Headers (8) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Schema Beautify

```
{
  "titulo": "Aviso importante",
  "conteudo": "Teremos aula na próxima semana!"
}
```

Body Cookies Headers (4) Test Results 200 OK 11 ms 226 B Save Response

JSON Preview Visualize

```
{
    "Mensagem": "Título: Aviso importante Conteúdo: Teremos aula na próxima semana! Publicada: True"
}
```