

POSTMAN CHRONICLES

O CÓDIGO SAGRADO DAS
REQUISIÇÕES



Jean Felipe



Introdução



Postman

Por que o Postman é a
Espada do Dev Backend

O Postman é uma ferramenta essencial para qualquer desenvolvedor backend. Ele permite testar, documentar e automatizar requisições HTTP de forma simples e visual. Com ele, você pode enviar requisições para suas APIs, validar respostas e até simular fluxos complexos, tudo sem precisar escrever uma linha de código de front-end.



Exemplo prático:

Você criou uma API que retorna informações de usuários:



```
GET https://api.meusistema.com/users
```

Em vez de montar uma interface para testar, você simplesmente cola a URL no Postman, clica em Send e vê o resultado em segundos.

01

 Criando sua
Primeira Requisição



Requisições

HTTP – GET, POST, PUT e DELETE.

O coração do Postman são as requisições HTTP – GET, POST, PUT e DELETE.

Cada uma representa uma ação no servidor: buscar, criar, atualizar e remover dados.



Exemplo prático:

Imagine que você está testando um endpoint de cadastro:

```
POST https://api.meusistema.com/users
```

No corpo da requisição (Body), escolha o tipo JSON e insira:

```
{  
  "nome": "Jean Felipe",  
  "email": "jean@email.com"  
}
```

Ao clicar em Send, você envia o mesmo tipo de requisição que o front-end faria. Isso ajuda a validar se o endpoint está funcionando corretamente antes mesmo da interface existir.

02



Variáveis o Poder da Reutilização



Variáveis

URL base, token ou IDs.

Com variáveis, você evita repetir informações em cada requisição, como URL base, token ou IDs.



Exemplo prático:

Crie uma variável chamada `{{base_url}}` e defina o valor:



```
https://api.meusistema.com
```

Agora suas requisições podem ser assim:



```
GET {{base_url}}/users
```

Se precisar mudar o ambiente (produção, teste, local), basta alterar o valor da variável — e todas as requisições se atualizam automaticamente.

03

 Autenticação
Testando APIs com
Segurança



Autenticação

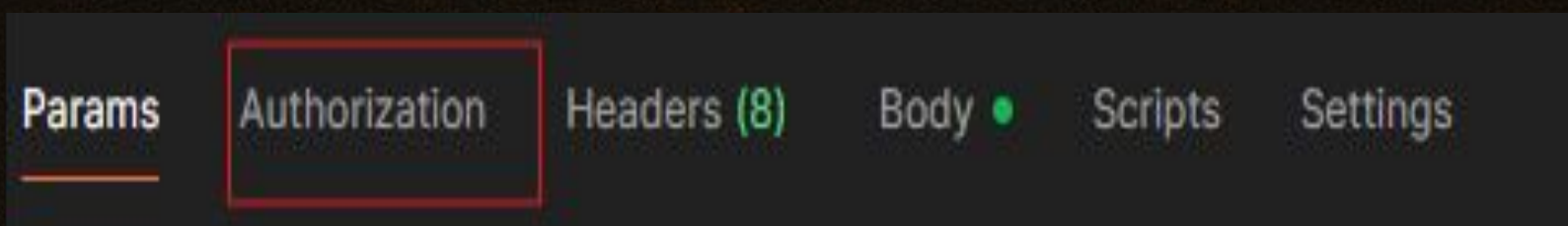
Bearer Token

A maioria das APIs precisa de autenticação (por exemplo, via Bearer Token).

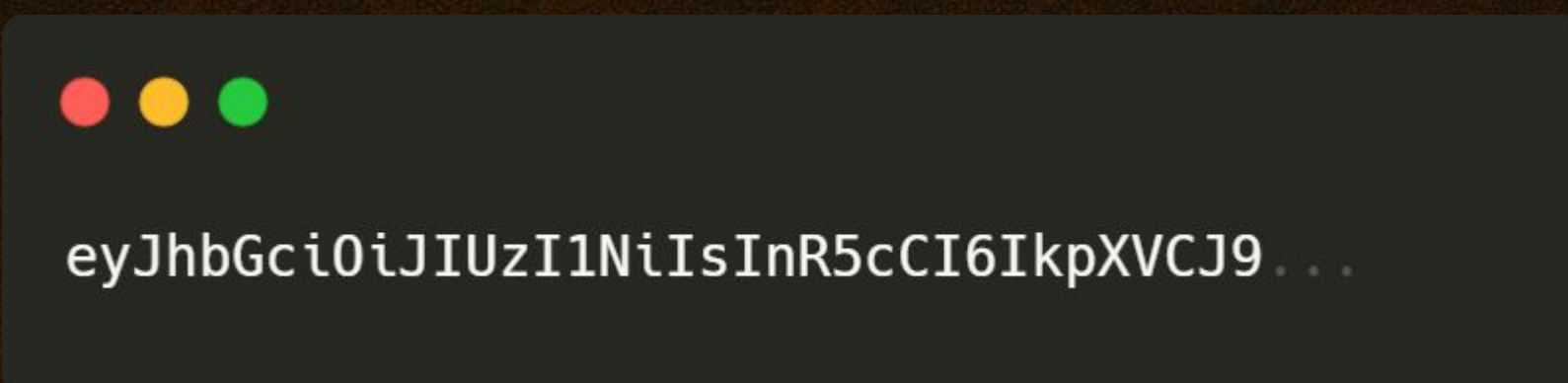
No Postman, você pode configurar isso facilmente na aba Authorization.



Exemplo prático:



Selecione o tipo Bearer Token e insira:



Agora todas as requisições dessa coleção já vão usar esse token, sem precisar adicioná-lo manualmente.

04

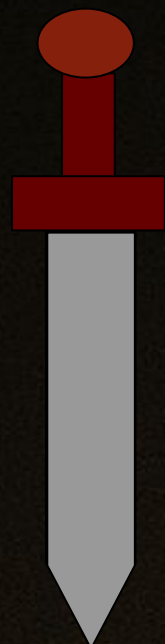
 Coleções
Organizando suas
Aventuras de API



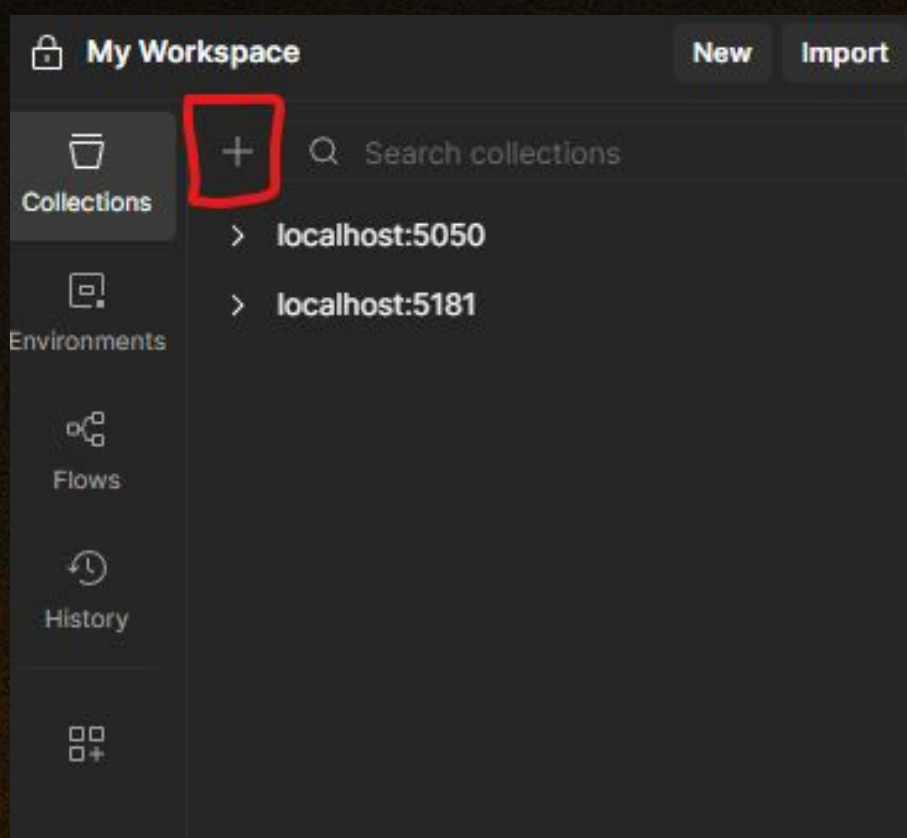
Coleções

As coleções são como capítulos do seu projeto.

Elas permitem agrupar requisições de um mesmo sistema, salvar configurações e compartilhar com o time.



Exemplo prático:



Crie uma coleção chamada API de Usuários e adicione dentro dela:

- GET /users
-
- POST /users
-
- PUT /users/:id
-
- DELETE /users/:id

Assim, toda a documentação do backend fica em um só lugar – pronta para ser exportada e enviada para o time.

05

⚡ Tests Automatizando
Validações



Tests

Scripts automáticos

0 Postman permite adicionar scripts automáticos em JavaScript para validar respostas.



Exemplo prático:

Na aba Tests, adicione:



```
pm.test("Status é 200", function () {  
  pm.response.to.have.status(200);  
});
```

Agora, sempre que rodar a requisição, o Postman dirá se o teste passou ou falhou. Isso é ótimo para monitorar endpoints em pipelines de CI/CD.

06

 Environments
Mudando de Cenário
sem Dor



Environments

Ambientes

Com ambientes, você alterna facilmente entre desenvolvimento, staging e produção.



Exemplo prático:

Crie três ambientes:

- Dev ➔ `base_url = http://localhost:3000`
- Staging ➔ `base_url = https://staging.api.meusistema.com`
- Prod ➔ `base_url = https://api.meusistema.com`
- Use `{{base_url}}` em todas as requisições. Assim, basta mudar o ambiente no topo do Postman e testar outro servidor instantaneamente.

07



Collections Runner

Testes em Massa



Collections Runners

Quer rodar todas as requisições de uma vez?

Use o Collection Runner. Ele executa todas as chamadas em sequência, ideal para validar fluxos completos.



Exemplo prático:

Execute:

- POST /login
- GET /users
- PUT /users/1
- DELETE /users/1

Perfeito para simular um ciclo completo de uso da API.

03

 Conclusão o
Grimório do Dev
Backend



Conclusão

aliado estratégico

O Postman é muito mais do que uma ferramenta de testes – é um aliado estratégico no desenvolvimento backend. Ele economiza tempo, melhora a colaboração entre times e transforma testes manuais em aventuras previsíveis e rápidas.