

VEM SER

Módulo 4 - Testes de API REST
Aula 2 - Postman (Parte 1)



DBC

Aula 2: Iniciando Teste de API com Postman (Parte 1)





Aula 2 - Conteúdo Programático

2.1 Introdução ao Postman

2.2 Criando Requisições

2.3 Ambientes e Variáveis

2.4 Dominando Scripts

2.5 De olho nas dicas!

2.6 Task API-02

2.1 Introdução ao Postman



2.1.1 O que é o Postman?

O Postman é uma plataforma popular utilizada para **testar, desenvolver e documentar APIs**.

Ele simplifica cada etapa do ciclo de vida da API e permite que você crie um **fluxo de trabalho mais eficiente**.



2.1.1 O que é o Postman?



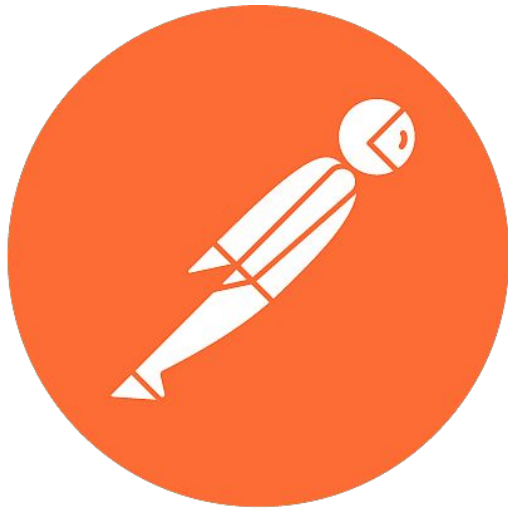
Fornece uma **interface gráfica amigável** que permite enviar solicitações para uma API e visualizar os resultados.

2.1.2 Principais Recursos



- Envio de requisições HTTP (GET, POST, PUT, DELETE, etc.);
- Testes automatizados de APIs;
- Colaboração em equipe e compartilhamento de coleções de API;
- Mock Servers e documentação de API.

2.1.3 Download e Instalação



Você pode instalar o aplicativo de duas maneiras:

- Como um aplicativo nativo (para o sistema operacional);
- Como uma extensão do Chrome.

É recomendado fazer instalação como **aplicativo nativo**:

- Mais recursos e suporte;
- <https://www.postman.com/downloads/>

2.2 Criando Requisições



2.2 Criando Requisições

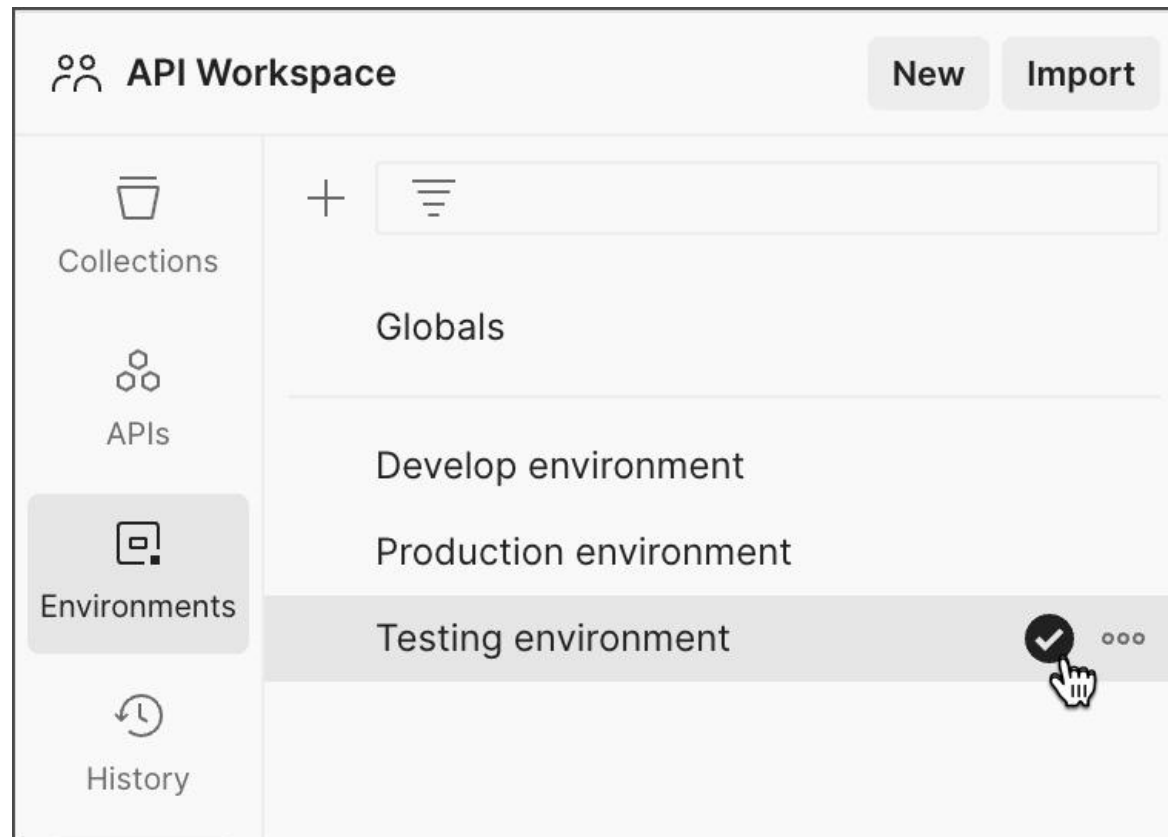
POST, GET, PUT e DELETE



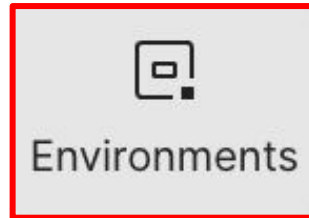
2.3 Ambientes e Variáveis no Postman



2.3 Ambientes e Variáveis no Postman



2.3.1 Ambientes



Pensem nos ambientes como seus próprios universos de trabalho. Cada ambiente é um **conjunto de variáveis** que você pode usar para simular diferentes situações de teste.

Isso é especialmente útil quando você está trabalhando em vários estágios de desenvolvimento, como desenvolvimento, teste e produção, sem precisar reconfigurar suas solicitações todas as vezes.

Para saber mais:

<https://learning.postman.com/docs/sending-requests/managing-environments/>

2.3.2 Variáveis

Podemos definir Variáveis como entidades que permitem **armazenar e manipular** valores no Postman.

	VARIABLE	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ
<input checked="" type="checkbox"/>	accessToken		
<input checked="" type="checkbox"/>	accessTokenExpiry		

2.3.2 Variáveis

Elas são utilizadas para armazenar informações temporárias, como tokens, URLs dinâmicas ou qualquer outro dado relevante para o fluxo de teste.

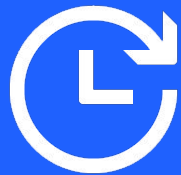
O uso de variáveis no Postman nos permite criar testes mais dinâmicos e flexíveis, pois em vez de repetir os mesmos valores várias vezes, você define uma variável e a usa em diversos lugares.

Para saber mais:

<https://learning.postman.com/docs/sending-requests/variables/>

Agora é a sua vez!

Prática em 45 min



Testando Endpoints de Login e Usuários no Postman

Objetivo:

- Aplicar “casos de teste” em ambiente de API real
- Utilizar Postman para enviar e avaliar respostas
- Aprofundar conhecimentos em testes de integração e validação de dados.

2.4 Dominando Scripts no Postman



2.4 Dominando Scripts no Postman

Scripts no Postman não são nada assustadores. Eles são, na verdade, uma maneira poderosa de adicionar **lógica e comportamentos dinâmicos** às suas requisições.

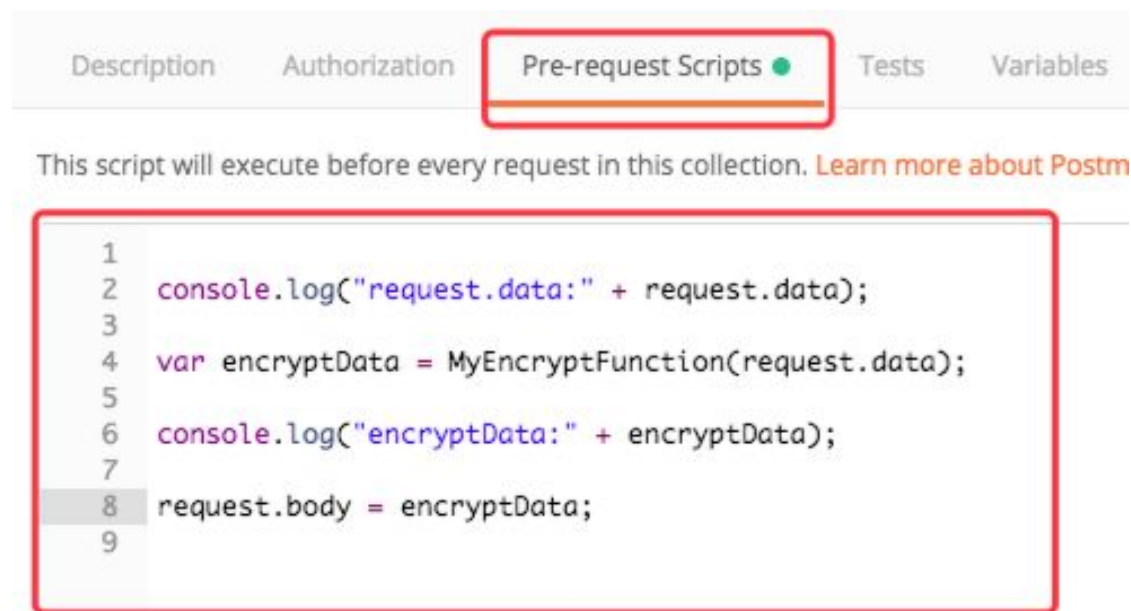
Tipos de Scripts:

- **Pre-request:** Executado antes da requisição.
- **Test:** Executado depois que uma resposta é recebida.

2.4.1 Preparando o terreno com Scripts de Pré-requisição

Scripts de pré-requisição preparam o ambiente, ajustam variáveis e podem até alterar os detalhes da requisição.

Por exemplo: Digamos que você quer adicionar um timestamp atual a cada requisição. Um script de pré-requisição faz isso facilmente.



The screenshot shows the Postman interface for editing a Pre-request Script. The 'Pre-request Scripts' tab is selected and highlighted with a red box. Below the tabs, a text line states: 'This script will execute before every request in this collection. [Learn more about Postman](#)'. A code editor area, also outlined with a red box, contains the following JavaScript code:

```

1
2 console.log("request.data:" + request.data);
3
4 var encryptData = MyEncryptFunction(request.data);
5
6 console.log("encryptData:" + encryptData);
7
8 request.body = encryptData;
9

```


2.4.2 Assegurando a Qualidade com Scripts de Teste



The screenshot shows the 'Tests' tab in Postman. The interface includes tabs for Params, Authorization, Headers (8), Body, Pre-request Script, Tests (selected), and Settings. The test scripts are as follows:

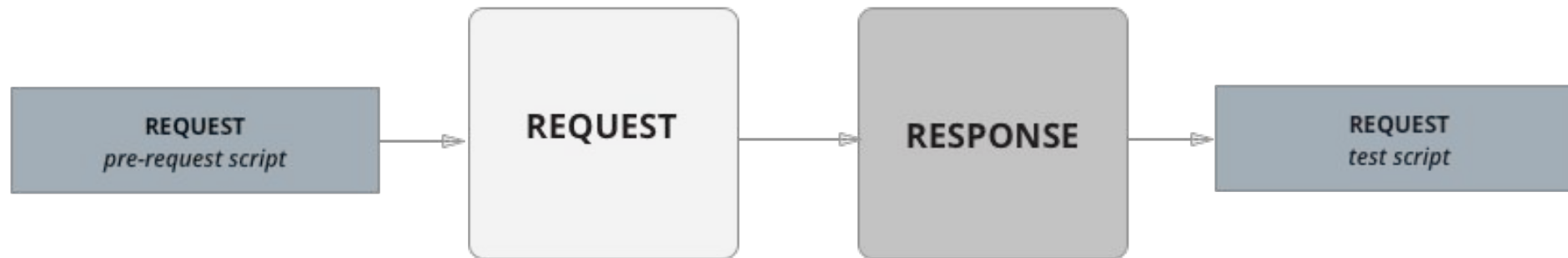
```
1 pm.test("response is ok", function () {  
2   pm.response.to.have.status(200);  
3 });  
4  
5 pm.test("response body has json with request queries", function () {  
6   pm.response.to.have.jsonBody('args.foo1', 'bar1')  
7   .and.have.jsonBody('args.foo2', 'bar2');  
8 });
```

Scripts de teste são executados após a resposta da requisição e podem validar se tudo ocorreu como esperado.

Usos Comuns: Você pode verificar códigos de status, validar respostas JSON, e até mesmo medir tempos de resposta. Tudo automaticamente.

2.4.3 Scripts - Ordem de execução

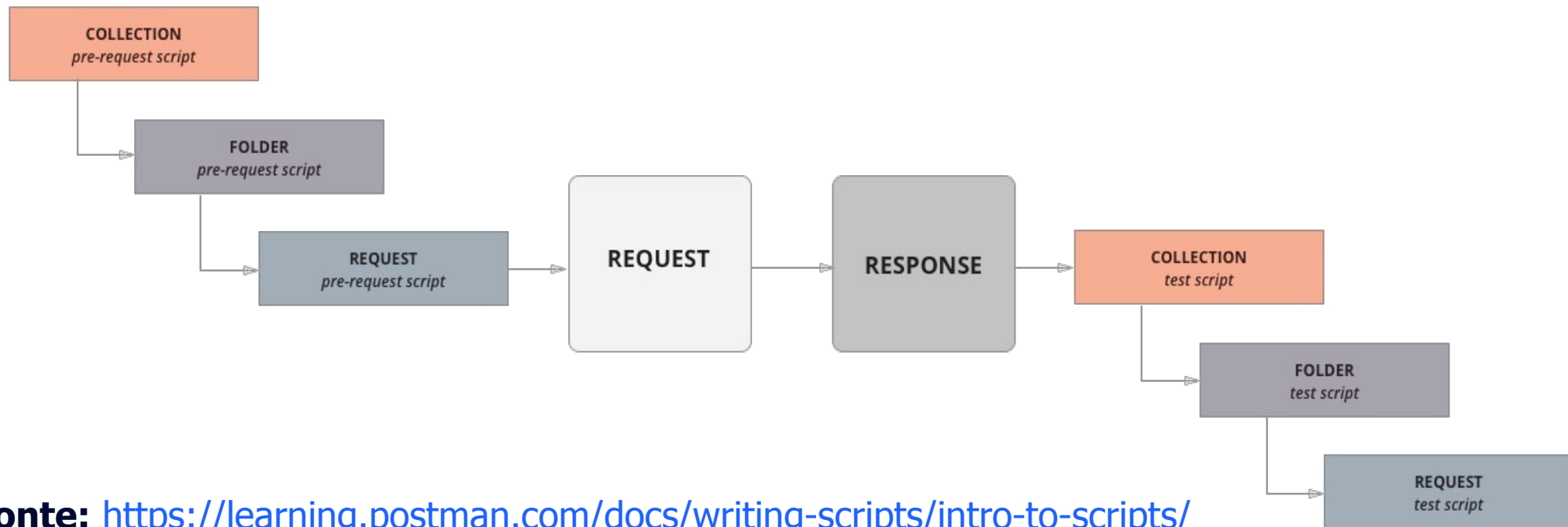
Para uma única solicitação:



Fonte: <https://learning.postman.com/docs/writing-scripts/intro-to-scripts/>

2.4.3 Scripts - Ordem de execução

Para cada solicitação em uma coleção:



Fonte: <https://learning.postman.com/docs/writing-scripts/intro-to-scripts/>

2.4.4 Hands On



2.5 De Olho nas Dicas



2.5 De Olho nas Dicas!

Dica 1: Gerando massa de dados

https://www.youtube.com/watch?v=QLuOwR_B6fs&list=PLH8edgEd8YTkXH4dMY66A7Xg64-_5BPZS&index=17

Dica 2: Criando validações para os testes

https://www.youtube.com/watch?v=p-82VJSPNaM&list=PLH8edgEd8YTkXH4dMY66A7Xg64-_5BPZS&index=18





Let's *Tech Up Together*