



# Como Utilizar o Postman: Um Guia Completo de Recursos e Funcionalidades

O Postman é uma ferramenta robusta que permite aos desenvolvedores testar APIs REST, enviar requisições HTTP, automatizar testes e muito mais.

No contexto de desenvolvimento de APIs e integrações, entender todos os recursos oferecidos pelo Postman pode otimizar o tempo de desenvolvimento e aumentar a confiabilidade das APIs.

Este guia aborda o uso completo do Postman, com uma abordagem prática e exemplos avançados.

## Instalação e Configuração Inicial

- **Download e Instalação:**
  - Disponível para Windows, macOS e Linux. Faça o download a partir do site oficial do Postman.
  - Instale e configure o Postman com base no seu sistema operacional.
- **Criação de Conta e Sincronização na Nuvem:**
  - Criar uma conta permite sincronizar as coleções e ambientes, além de armazenar automaticamente as configurações e preferências.
- **Interface do Usuário:**
  - **Workspace:** Um ambiente colaborativo onde você organiza coleções e recursos de API.
  - **Coleções:** Para organizar APIs em pastas e subpastas.
  - **Sidebar e Console de Depuração:** Acesso rápido aos recursos e verificação de logs de requisição.
  - **Tabs:** A interface baseada em abas permite trabalhar com múltiplas requisições simultaneamente.

## Criando e Executando Requisições HTTP

- **Tipos de Requisições HTTP:**
  - **GET:** Recupera dados de uma API. Exemplo: GET /api/usuarios.
  - **POST:** Envia dados para serem processados. Exemplo: POST /api/usuarios.
  - **PUT e PATCH:** Atualizações parciais (PATCH) ou completas (PUT) de recursos.
  - **DELETE:** Remove um recurso específico.



- **Configuração de Headers e Body:**
  - **Headers:** Configuração de cabeçalhos para definir Content-Type, Authorization, etc.
  - **Body:** Corpo da requisição, com opções de raw, form-data, x-www-form-urlencoded, entre outros.
- **Params e Path Variables:**
  - **Params:** Adiciona parâmetros de consulta (?key=value), úteis para filtros.
  - **Path Variables:** Define variáveis na URL, por exemplo, GET /api/usuarios/{id}.

## Utilizando Coleções e Folders

- **Organizando Requisições em Coleções:**
  - Coleções permitem agrupar requisições e fornecer organização hierárquica, facilitando o acesso e o compartilhamento.
  - Para APIs com múltiplos endpoints, cada endpoint pode ser salvo e categorizado em uma coleção.
- **Execução em Lote e Controle de Fluxo:**
  - **Folders:** Criação de pastas e subpastas dentro das coleções para uma organização ainda mais detalhada.
  - **Execução Sequencial:** Possibilidade de configurar a ordem de execução das requisições.
  - **Exemplo Prático:** Uma coleção para uma API CRUD completa com requests organizadas por operações.

## Variáveis e Ambientes

- **Tipos de Variáveis:**
  - **Variáveis Locais e Globais:** Definidas para requisições individuais (local) ou em todo o Postman (global).
  - **Variáveis de Ambiente:** Úteis para configurar parâmetros que mudam de acordo com o ambiente (desenvolvimento, produção, etc.).
- **Criação e Configuração de Ambientes:**
  - Exemplos de ambientes: dev, staging, production.
  - Configuração de variáveis como {{baseUrl}} para URLs e chaves de API diferentes.
- **Exemplo Prático:**
  - Configuração de variáveis e utilização de {{}} para substituir parâmetros diretamente nas requisições.

## Testes e Scripts

- **Scripts de Pré-Requisição e Testes:**
  - **Pré-Requisição:** Executa scripts em JavaScript antes da requisição. Útil para configurar variáveis dinamicamente ou para realizar cálculos necessários antes do envio da requisição.
  - **Testes:** Scripts de validação para testar o comportamento das respostas, como o status da resposta, conteúdo JSON, e headers.
- **Automatizando Validações:**
  - Exemplo: Validação automática para verificar se o código de resposta é 200, se o JSON contém determinados campos, etc.
  - **Exemplo de Script:**



#### javascript

```
pm.test("Status code is 200", function () {  
  pm.response.to.have.status(200);  
});  
  
pm.test("Response has JSON data", function () {  
  pm.response.to.have.header("Content-Type", "application/json");  
});
```

## Monitoramento e Automação com Collection Runner

- **Collection Runner:**
  - Ferramenta para executar coleções de requisições sequencialmente com opções de entrada de dados em massa.
  - Configuração de parâmetros para criar testes mais avançados e obter relatórios de resultados.
- **Testes Automatizados e Monitoramento:**
  - **Configuração de Monitores:** Define intervalos específicos para testar as APIs automaticamente.
  - **Alertas e Logs:** Configuração de notificações por e-mail para falhas nos testes, permitindo intervenções rápidas.

## Integrações com CI/CD

- **Postman e Integração Contínua/Entrega Contínua (CI/CD):**
  - Postman oferece suporte à integração com CI/CD, essencial para automatizar os testes de API durante o desenvolvimento e as atualizações de software.
- **Exportação e Execução com Newman:**
  - **Newman:** Uma CLI para executar coleções do Postman fora do ambiente Postman, ideal para automação em CI/CD.
  - **Exemplo de Execução:**

#### bash

```
newman run minha-colecao.postman_collection.json
```

- **Relatórios Personalizados:** Utilização de plugins para geração de relatórios em formato HTML, JSON, entre outros.

## Conclusão

Esse guia detalhado do Postman proporciona uma visão ampla de como utilizar a ferramenta ao máximo. Além de facilitar testes manuais, os recursos de automação e integração com CI/CD tornam o Postman essencial no pipeline de desenvolvimento.