**POSTMAN**

GET – Recuperar dados de API

POST – Enviar novos dados API

PATCH/PUT – Atualizar dados existentes

DELETE – Remove dados existentes

Sumário

[DICAS ÚTEIS 3](#_Toc85446135)

[COLEÇÕES 4](#_Toc85446136)

[WORKSPACE 5](#_Toc85446137)

[REQUEST 6](#_Toc85446138)

[ENDPOINT: 6](#_Toc85446139)

[PASSAGEM DE PARAMETRO METODO GET 6](#_Toc85446140)

[Query 6](#_Toc85446141)

[Path 6](#_Toc85446142)

[AUTENTICAÇÃO 7](#_Toc85446143)

[AGRUPAMENTO DE REQUESTS EM COLEÇÕES 7](#_Toc85446144)

[VARIÁVEIS 8](#_Toc85446145)

[Local 8](#_Toc85446146)

[Ambiente 8](#_Toc85446147)

[Coleção 8](#_Toc85446148)

[Global 9](#_Toc85446149)

[USANDO VARIAVEIS NOS PARAMETROS DE QUERY 10](#_Toc85446150)

[GERENCIANDO AMBIENTES 10](#_Toc85446151)

[SCRIPTS NO POSTMAN 11](#_Toc85446152)

[SCRIPTS DE PRÉ-REQUISIÇÃO 11](#_Toc85446153)

[Exemplos de Scripts de pré-requisição pm.info 12](#_Toc85446154)

[Exemplos de Scripts de pré-requisição pm.environment 12](#_Toc85446155)

[CRIANDO E EXECUTANDO TESTES 12](#_Toc85446156)

[CRIANDO ESTRUTURA DE TESTES 13](#_Toc85446157)

[DOCUMENTANDO TESTES 13](#_Toc85446158)

[Como fazer? 13](#_Toc85446159)

[UTILIZANDO O COLLECTION RUNNER 14](#_Toc85446160)

[AGENDANDO EXECUÇÕES 14](#_Toc85446161)

[DESENVOLVENDO FLUXOS DE REQUISIÇÕES 15](#_Toc85446162)

[IMPORTANTO ARQUIVOS DE DADOS 16](#_Toc85446163)

[EXECUTANDO COLEÇÕES VIA LINHA DE COMANDO - NEWMAN 16](#_Toc85446164)

[PROMPT: 16](#_Toc85446165)

[Comandos: 17](#_Toc85446166)

[INTEGRAÇÃO CONTINUA COM POSTMAN API 18](#_Toc85446167)

[Como gerar API Key? 19](#_Toc85446168)

[Minha API 19](#_Toc85446169)

[LINKS ÚTEIS API 19](#_Toc85446170)

[Executar no Newman 19](#_Toc85446171)

[INTEGRANDO COM JENKINS 20](#_Toc85446172)

[Executar no Newman 20](#_Toc85446173)

[Como usar o Jenkins? 20](#_Toc85446174)

[Configurar para rodar os testes automaticamente 20](#_Toc85446175)

[Exportar resultado com .json 20](#_Toc85446176)

[GERANDO RELATORIOS HTML COM NEWMAN 21](#_Toc85446177)

[HTML REPORTER 21](#_Toc85446178)

[COMANDOS 21](#_Toc85446179)

[NEWMAN HTML REPORTER EXTRA 22](#_Toc85446180)

[COMANDOS 22](#_Toc85446181)

[Lista de parâmetros para usar com html extra (clique 2 vezes para abrir) 23](#_Toc85446182)

[COMO FAZER EXECUÇÃO VIA LINHA DE COMANDO E GERAR RELATÓRIO PERSONALIZADO? 23](#_Toc85446183)

[Documentação: 25](#_Toc85446184)

# DICAS ÚTEIS

* Em Body ->> raw eu posso colar o JSON/XML/HTML, depois só clicar em “Text”, escolher o tipo de linguagem e depois clicar em “Beautify” para indentar;
* Em Authorization eu adiciono o token da API caso haja;
* Para publicar uma coleção selecione a coleção ->> 3 pontinhos ->> View Documentation ->> Publish
* **Sempre a variavel menos abrangente vai sobrescrever a variavel mais abrangente**, ou seja, uma variavel local sempre sobrescreve a variavel global, por exemplo.
* Pode-se setar uma variavel na parte de autorização com o token, por exemplo, indo na variavel global e declarando ela
* Pode-se em uma requisição POST no “Body” substituir os valores por variáveis globais **SEMPRE ENTRE ASPAS E CHAVES DUPLAS**

Texto

Descrição gerada automaticamente

* Pode-se declarar uma variavel de ambiente com o link da URL base utilizada e substituir o link no método. URL Base: <https://gorest.co.in>



* Para escrever algo no console utilize: **console.log(“Texto no console”)**
* pm.environment.unset("id\_post") 🡪 limpa variável específica

pm.globals.clear() 🡪 apaga variaveis

* Para pular algum(s) requests use no Pre-request ou Tests em algum request: **postman.setNextRequest(‘Nome ou ID do request que será o próximo após o atual').** Para pegar o ID digite no Pre-request ou Tests: console.log(pm.info.requestId)
* Para deixar request em loop:

**postman.setNextRequest(‘O nome ou ID do request atual’)**

* Executar requests até certo ponto:

**postman.setNextRequest(null) 🡨** colocar esse comando em qual request for o último a ser rodado

* Para puxar um dado de um arquivo faça no Pre-request Script:

**Pm.IteractionData.get(‘nome\_da\_variavel\_declarada\_no\_Body’)**

# COLEÇÕES

* Agrupa requisições em pastas;
* Reuso de requisições;
* Podem ser importadas e exportadas.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

# WORKSPACE

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

# REQUEST

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

ENDPOINT: URL base + recurso a ser utilizado



# PASSAGEM DE PARAMETRO METODO GET

Query**:** URL base + recurso a ser utilizado + ? + chave + valor pretendido (caso queria pesquisa mais de um valor coloque & + chave + valor).

* **Ex:** [**https://gorest.co.in/public/v1/users?name=varma**](https://gorest.co.in/public/v1/users?name=varma)🡨Aqui ele irá trazer todos os users que tem como nome “Varma”

Path**:** URL base + recurso a ser utilizado + id.

* **Ex:** [**https://gorest.co.in/public/v1/users/43**](https://gorest.co.in/public/v1/users/43)🡨Retornará os dados do user com id 43

Pode-se colocar “:” após a URL + recurso a ser utilizado e passar o valor dentro de “Value”.

* **Ex:** **public/v1/users/:id**

# AUTENTICAÇÃO

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

# AGRUPAMENTO DE REQUESTS EM COLEÇÕES

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente com confiança média

# VARIÁVEIS

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Local**:** {{nome\_variavel}} e em “Pre-request Script” selecione “Set a global variable” e mude “Globals” para “variables”

Interface gráfica do usuário, Aplicativo, Site

Descrição gerada automaticamente

Ambiente**:** Para criar uma variavel de ambiente basta clicar no olho no canto superior direito, clicar em Add, adicionar a variavel com seu valor ao lado

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Coleção**:** Clique nos 3 pontinhos da coleção ->> Variables e adiciona a variavel juntamente com seu valor

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Global**:** Para criar uma variavel global basta clicar no olho no canto superior direito, em “Globals” clicar em Add

Tela de celular com aplicativo aberto

Descrição gerada automaticamente

**\*\*Pode-se também passar uma variavel da seguinte forma:**

Tela de celular com aplicativo aberto

Descrição gerada automaticamente

## USANDO VARIAVEIS NOS PARAMETROS DE QUERY

Tela de celular

Descrição gerada automaticamente

**\***Dessa forma pode-se passar os valores das variáveis criando variáveis globais ou no Pre-request Script

# GERENCIANDO AMBIENTES

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

# SCRIPTS NO POSTMAN

Texto

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

# SCRIPTS DE PRÉ-REQUISIÇÃO

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Exemplos de Scripts de pré-requisição pm.info**:**

* **pm.info.eventName**
* **pm.info.iteraction**
* **pm.info.iteractionCount**
* **pm.info.requestName**
* **pm.info.requestId**

Exemplos de Scripts de pré-requisição pm.environment**:**

* **pm.environment.name**
* **pm.environment.has**
* **pm.environment.get**
* **pm.environment.set**
* **pm.environment.unset**
* **pm.environment.clear()**

# CRIANDO E EXECUTANDO TESTES

Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

# CRIANDO ESTRUTURA DE TESTES

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

# DOCUMENTANDO TESTES

Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

## Como fazer?

1. Rode os testes;
2. Altere o nome do request e a descrição adicionando que tipo de teste é e os resultados esperados para ele (canto superior direito)

**Ex:** Teste para criação de usuários no recurso de Usuários

Resultados esperados:

- Response code = 201

- Tempo de espera < 2000ms

1. Salve como exemplo no canto inferior direito
2. Salve tudo
3. Gere uma documentação 3 pontinhos (coleção) ->> View Documentation ->> Publish

# UTILIZANDO O COLLECTION RUNNER

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente com confiança média

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

# AGENDANDO EXECUÇÕES

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

# DESENVOLVENDO FLUXOS DE REQUISIÇÕES

Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

* Para pular algum(s) requests use no Pre-request ou Tests em algum request: **postman.setNextRequest(‘Nome ou ID do request que será o próximo após o atual').** Para pegar o ID digite no Pre-request ou Tests: console.log(pm.info.requestId)
* Para deixar request em loop:

**postman.setNextRequest(‘O nome ou ID do request atual’)**

* Executar requests até certo ponto:

**postman.setNextRequest(null) 🡨** colocar esse comando em qual request for o último a ser rodado

# IMPORTANTO ARQUIVOS DE DADOS

Texto

Descrição gerada automaticamente com confiança média

Texto

Descrição gerada automaticamente

* Para puxar um dado de um arquivo faça no Pre-request Script:

**pm.iterationData.get(‘nome\_da\_variavel\_declarada\_no\_Body’)**

# EXECUTANDO COLEÇÕES VIA LINHA DE COMANDO - NEWMAN

Texto

Descrição gerada automaticamente

## PROMPT:

Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

## Comandos:

* Para abrir a pasta do arquivo( se estiver salvo em C: ) – **cd c:/Nome\_pasta\_arquivo**
* **newman run c:\Postman\Nome\_arquivo.json** 🡪 O ARQUIVO NÃO PODE TER ESPAÇOS. Abre o arquivo .json exportado no Postman. Para exportá-lo selecione a coleção e clique em seus 3 pontinhos, após isso clique em “Export” , salve o arquivo sem espaços e com .json no final
* **newman run c:/Postman/Nome\_arquivo.json -e c:/Postman/Nome\_ambiente.json** 🡪 O ARQUIVO NÃO PODE TER ESPAÇOS. Abre o arquivo juntamente com seu ambiente. Para exportar o ambiente vá até o olho no canto superior direito e clique em “Edit” na tela que abrir clique nos 3 pontos e em “Export”, salve o arquivo sem espaços e com .json no final
* **newman run -n 3 c:/Postman/GORESTAPI-AULA17-25.postman\_collection.json -e c:/Postman/Testes\_environment.json** 🡪 O ARQUIVO NÃO PODE TER ESPAÇOS. Abrirá o arquivo com o ambiente, porém rodará a iteração 3 vezes
* **newman run** -h 🡪 traz um menu de ajuda
* **newman run Nome\_arquivo.json -e Nome\_ambiente.json --folder Nome\_pasta\_coleção** (respeitando acentos, letras maiúsculas e minúsculas) 🡪 Executa apenas a pasta selecionada da coleção
* **newman run Nome\_arquivo.json -e Nome\_ambiente.json --folder Nome\_pasta\_coleção --folder Nome\_pasta\_coleção\_2** 🡪 Executa duas pastas
* **newman run --delay-request 10000 Nome\_arquivo.json -e Nome\_ambiente.json --folder Nome\_pasta\_coleção** 🡪 Executa cada request com um delay de 10 segundos (10000)
* **newman run Nome\_arquivo.json -e Nome\_ambiente.json --bail** 🡪 Para quando houver algum erro de assert
* **newman run Nome\_arquivo.json -e Nome\_ambiente.json --verbose** 🡪 Adiciona mais informações às requisições
* **newman run Nome\_arquivo.json -e Nome\_ambiente.json --silent** 🡪 Executa as requisições sem avisar nada na tela
* **newman run -d Nome\_arquivo\_dados\_json.json Nome\_arquivo.json -e Nome\_ambiente.json --folder Nome\_pasta\_coleção** 🡪 Carregará o arquivo .json de dados (salvo em aulas passadas)
* **newman run -d Nome\_arquivo\_dados\_csv.csv Nome\_arquivo.json -e Nome\_ambiente.json --folder Nome\_pasta\_coleção** 🡪 Carregará o arquivo .csv (Excel, mas criado em notepad++) de dados

# INTEGRAÇÃO CONTINUA COM POSTMAN API

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente com confiança média

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente com confiança média

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

## Como gerar API Key?

1. No Postman selecione a coleção desejada
2. No canto superior direito selecione para abrir a documentação
3. Clique na última opção “Info”
4. Clique em “View all options”
5. Selecione “Postman API”
6. Em “Overview” clique no link “integrations dashboard”
7. Gere uma Key, adicionando um nome qualquer a ela e copie

### Minha API

PMAK-61682d90101805004d1e8500-f0a79f5abf22419ac1602d1bb694188b00

**uid coleção Aula 17 - 25**: 17874192-37e37a1e-4ed9-4da4-ada3-2831e33445d5

## LINKS ÚTEIS API

* **https://api.getpostman.com/collections?apikey=PMAK-61682d90101805004d1e8500-f0a79f5abf22419ac1602d1bb694188b00** 🡪 Com ele consegue-se ver tudo que há no meu Workspace (coleções e tudo mais)
* **https://api.getpostman.com/collections/17874192-37e37a1e-4ed9-4da4-ada3-2831e33445d5?apikey=PMAK-61682d90101805004d1e8500-f0a79f5abf22419ac1602d1bb694188b00** 🡪 Retorna coleção completa
* **https://api.getpostman.com/environments?apikey=PMAK-61682d90101805004d1e8500-f0a79f5abf22419ac1602d1bb694188b00** 🡪 Retorna os ambientes presentes no Workspace

## Executar no Newman

* **newman run** **https://api.getpostman.com/collections/17874192-37e37a1e-4ed9-4da4-ada3-2831e33445d5?apikey=PMAK-61682d90101805004d1e8500-f0a79f5abf22419ac1602d1bb694188b00 --environment https://api.getpostman.com/environments/17874192-02f39b37-a151-466f-8822-9ee64a1c6a8d?apikey=PMAK-61682d90101805004d1e8500-f0a79f5abf22419ac1602d1bb694188b00**

**uid ambiente de Testes:** 17874192-02f39b37-a151-466f-8822-9ee64a1c6a8d

# INTEGRANDO COM JENKINS

Texto

Descrição gerada automaticamente

## Executar no Newman

* **newman run** **https://api.getpostman.com/collections/17874192-37e37a1e-4ed9-4da4-ada3-2831e33445d5?apikey=PMAK-61682d90101805004d1e8500-f0a79f5abf22419ac1602d1bb694188b00 --environment https://api.getpostman.com/environments/17874192-02f39b37-a151-466f-8822-9ee64a1c6a8d?apikey=PMAK-61682d90101805004d1e8500-f0a79f5abf22419ac1602d1bb694188b00**

## Como usar o Jenkins?

1. Exportar tudo no Postman para a mesma pasta
2. No Jenkins criar um job e selecionar a opção "Provide Node..."
3. Digitar o comando abrindo a pasta (Ex: cd C:\C:\Users\Vini-PC\Desktop\APIHunter) e na outra linha incluir o newman run -d arquivo\_dados.csv (para puxar os dados) nome\_arquivo.json -e (puxar ambiente) nome\_ambiente.json --folder nome\_pasta
4. Caso tenha mais de uma pasta no projeto adicionar um novo comando Windows e fazer a mesma coisa

Texto, Aplicativo

Descrição gerada automaticamente

## Configurar para rodar os testes automaticamente

1. Em “Trigger de builds” marque a opção “Construir periodicamente”
2. Em “Agenda” adicione o tempo ex: “H/2” 🡪 A cada 2 minutos

## Exportar resultado com .json

1. No Postman Rode a(s) coleções
2. Clique em “Export Results”

**newman run** **https://api.getpostman.com/collections/17874192-37e37a1e-4ed9-4da4-ada3-2831e33445d5?apikey=PMAK-61682d90101805004d1e8500-f0a79f5abf22419ac1602d1bb694188b00 --environment https://api.getpostman.com/environments/17874192-02f39b37-a151-466f-8822-9ee64a1c6a8d?apikey=PMAK-61682d90101805004d1e8500-f0a79f5abf22419ac1602d1bb694188b00**

# GERANDO RELATORIOS HTML COM NEWMAN

## HTML REPORTER

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

### COMANDOS

* **npm install -g newman-reporter-html** 🡪 instalar o html reporter
* **newman run Nome\_arquivo.json -e Nome\_ambiente.json -r html** 🡪 Gera um html reporter que será salvo em uma pasta chamada “newman” na mesma pasta em que está os arquivos json utilizados
* **newman run Nome\_arquivo.json -e Nome\_ambiente.json --reporters cli,html --reporter-html-template Nome\_pasta\_template/Nome\_template.hbs --reporter-html-export Nome\_pasta\_export/Nome\_export\_salvo.html**

**EX:** newman run GORESTAPI-AULA17-25.postman\_collection.json -e Testes\_environment.json --reporters cli,html --reporter-html-template templates/customHtml.hbs --reporter-html-export export\_reports/export\_aula2.html

## NEWMAN HTML REPORTER EXTRA

Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

### COMANDOS

* **npm install -g newman-reporter-htmlextra** 🡪 Instala o html extra
* **newman run Nome\_arquivo.json -e Nome\_ambiente.json -r htmlextra** 🡪 Gera um html reporter mais bonito e personalizável

### Lista de parâmetros para usar com html extra (clique 2 vezes para abrir)



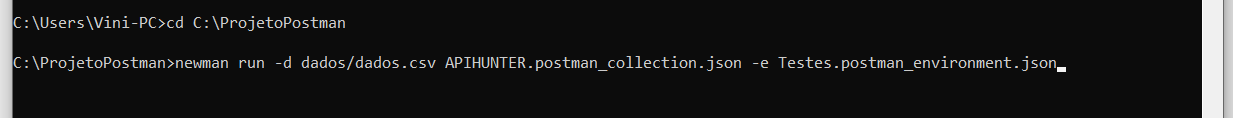
* **newman run Nome\_arquivo.json -e Nome\_ambiente.json -r htmlextra --reporter-htmlextra-export** **Nome\_pasta\_export/Nome\_export\_a\_ser\_salvo.html**

**Ex:** newman run GORESTAPI-AULA17-25.postman\_collection.json -e Testes\_environment.json -r htmlextra --reporter-htmlextra-export export\_reports/export\_extra.html

* **newman run Nome\_arquivo.json -e Nome\_ambiente.json -r htmlextra --reporter-htmlextra-title "Título no relatório aqui"** 🡪 Muda o título no relatório
* **newman run Nome\_arquivo.json -e Nome\_ambiente.json -r htmlextra --reporter-htmlextra-title "Título no relatório aqui" --reporter-htmlextra-titleSize 6** 🡪 Mudará o tamanho do título para 6
* **newman run Nome\_arquivo.json -e Nome\_ambiente.json -r htmlextra --reporter-htmlextra-title "Título no relatório aqui" --reporter-htmlextra-titleSize 6 --reporter-htmlextra-showEnvironmentData 🡪** Mostrará as variáveis de ambiente utilizadas

# COMO FAZER EXECUÇÃO VIA LINHA DE COMANDO E GERAR RELATÓRIO PERSONALIZADO?

1. Crie uma pasta em C:
2. Dentro dela crie pastas de “dados”, “reports” e “template”
3. Adicione os arquivos .json de ambientes e coleção
4. Na pasta de dados adicione os dados usados no projeto
5. Na pasta “Template” adicione os templates disponíveis na pasta Postman da C:
6. No CMD navegue para a pasta criada no passo 1 (cd diretorio\_aqui)
7. No CMD: **newman run -d dados/arquivo\_dados.csv nome\_arquivo.json -e nome\_ambiente.json**



1. Gere API KEY (página 19) **MINHA API KEY: PMAK-61682d90101805004d1e8500-f0a79f5abf22419ac1602d1bb694188b00**
2. [**https://api.getpostman.com/collections?apikey=APIKEYAQUI**](https://api.getpostman.com/collections?apikey=APIKEYAQUI)🡪 para retornar as coleções
3. **https://api.getpostman.com/collections/uIDColeçãoAqui?apikey=APIKEYAQUI** 🡪 Retornar coleção completa

**uID API HUNTER Coleção:** 17874192-d5858e43-468a-4867-8db6-089fb6a6423b

**uID Ambiente desenvolvimento:** 17874192-ade6c138-b509-4bab-97d7-04652a82fd95

**uID Ambiente Testes:** 17874192-6d46b9ba-3779-4737-aefb-62d888b586c5

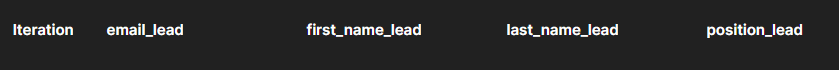
1. **https://api.getpostman.com/environments?apikey= APIKEYAQUI** 🡪 Retorna os ambientes presentes no Workspace
2. **newman run -d dados/dados\_arquivo.csv https://api.getpostman.com/collections/ uIDColeçãoAqui?apikey= APIKEYAQUI --environment https://api.getpostman.com/environments/ uIDAmbienteAqui?apikey= APIKEYAQUI** --folder Leads

* **newman run -d dados/dados.csv https://api.getpostman.com/collections/**17874192-d5858e43-468a-4867-8db6-089fb6a6423b**?apikey=PMAK-61682d90101805004d1e8500-f0a79f5abf22419ac1602d1bb694188b00 --environment https://api.getpostman.com/environments/**17874192-ade6c138-b509-4bab-97d7-04652a82fd95**?apikey=PMAK-61682d90101805004d1e8500-f0a79f5abf22419ac1602d1bb694188b00 --**folder Leads

1. **newman run -d dados/dados\_arquivo.csv https://api.getpostman.com/collections/ uIDColeçãoAqui?apikey= APIKEYAQUI --environment https://api.getpostman.com/environments/ uIDAmbienteAqui?apikey= APIKEYAQUI** --folder Leads -r **htmlextra --reporter-htmlextra-export** **Nome\_pasta\_export/Nome\_export\_a\_ser\_salvo.html --reporter-htmlextra-title "Título aqui"** 🡪 Gerará um Relatório html extra na pasta desejado com o nome e o título desejado

* **newman run -d dados/dados.csv https://api.getpostman.com/collections/17874192-d5858e43-468a-4867-8db6-089fb6a6423b?apikey=PMAK-61682d90101805004d1e8500-f0a79f5abf22419ac1602d1bb694188b00 --environment https://api.getpostman.com/environments/17874192-ade6c138-b509-4bab-97d7-04652a82fd95?apikey=PMAK-61682d90101805004d1e8500-f0a79f5abf22419ac1602d1bb694188b00 --folder Leads -r htmlextra --reporter-htmlextra-export reports/Relatorio\_DesenvolvimentoLeads.html --reporter-htmlextra-title " Relatório Equipe Desenvolvimento Pasta Leads"**

**newman run -d dados/dados.csv https://api.getpostman.com/collections/**17874192-d5858e43-468a-4867-8db6-089fb6a6423b**?apikey=PMAK-61682d90101805004d1e8500-f0a79f5abf22419ac1602d1bb694188b00 --environment https://api.getpostman.com/environments/**17874192-ade6c138-b509-4bab-97d7-04652a82fd95**?apikey=PMAK-61682d90101805004d1e8500-f0a79f5abf22419ac1602d1bb694188b00 --**folder Leads



# Documentação:

https://documenter.getpostman.com/view/17874192/UV5XgGU4