

API Testing - Coleções e variáveis de ambiente com Postman

Índice

- 01** Coleções
- 02** Variáveis de ambiente
- 03** *Runner* para executar coleções



01

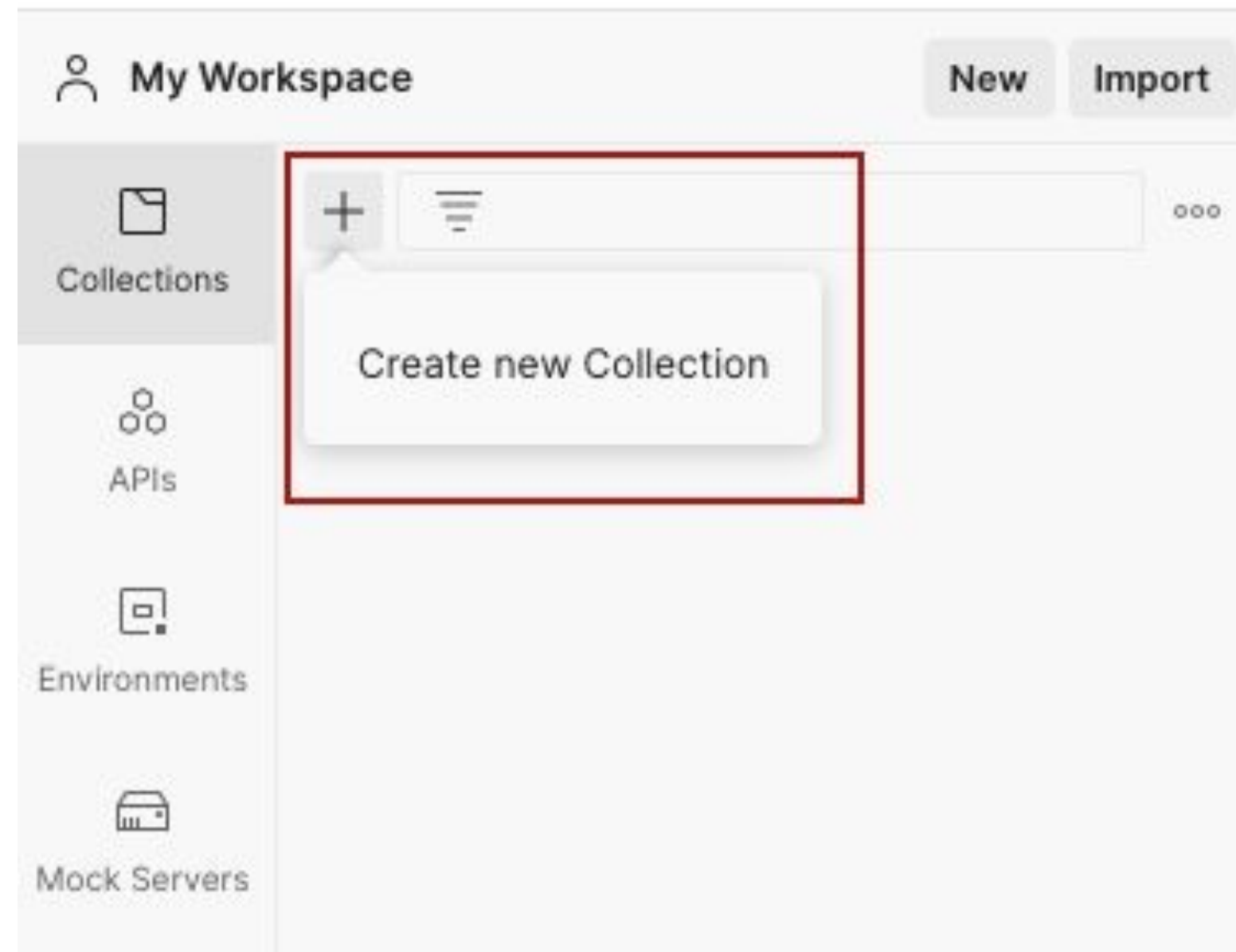
Coleções

As coleções são um grupo de **solicitações salvas** que podem ser organizadas em pastas. Isso nos permite agrupar e gerenciar nossas solicitações com mais eficiência.



Passos para criar uma coleção

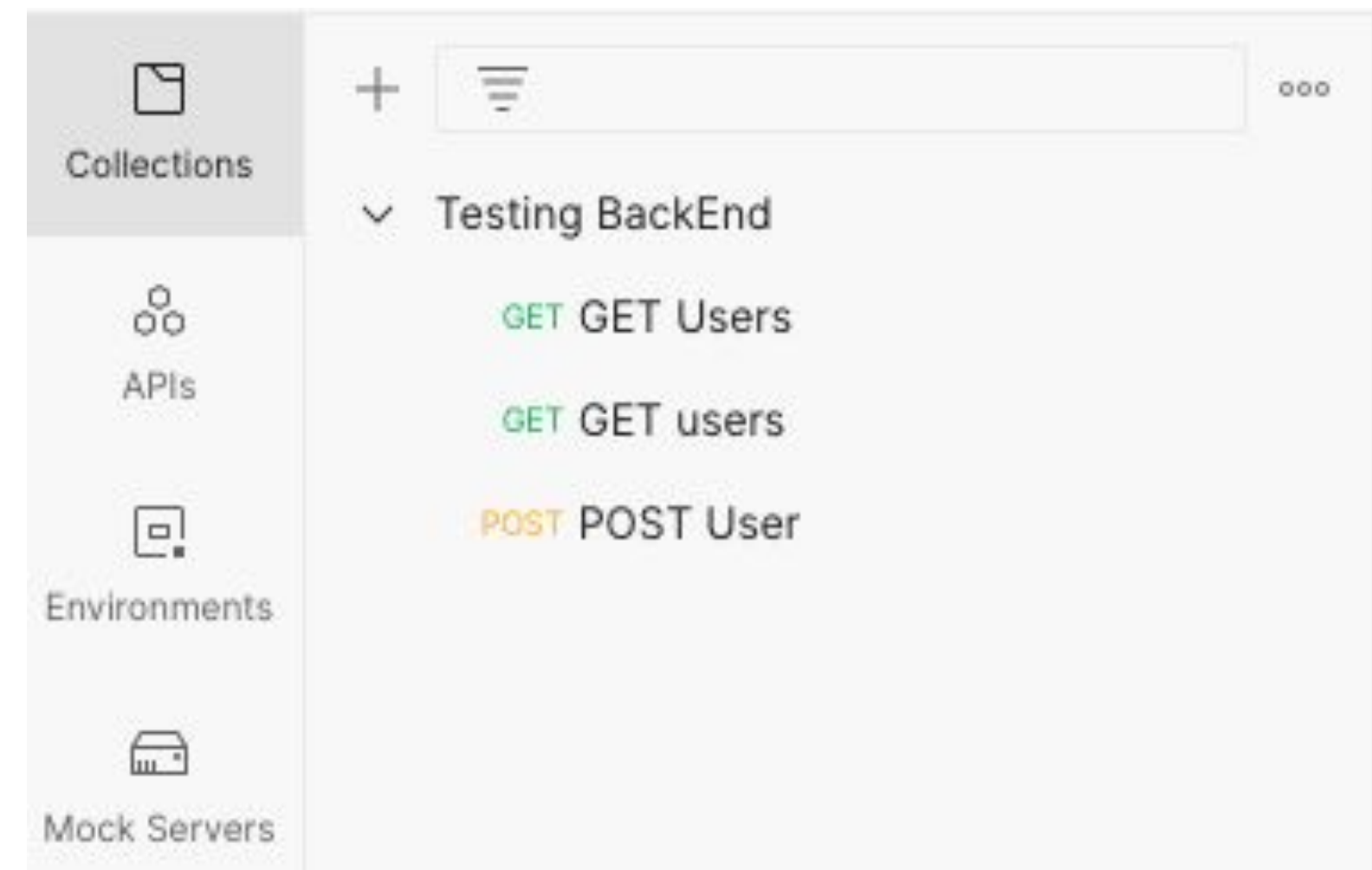
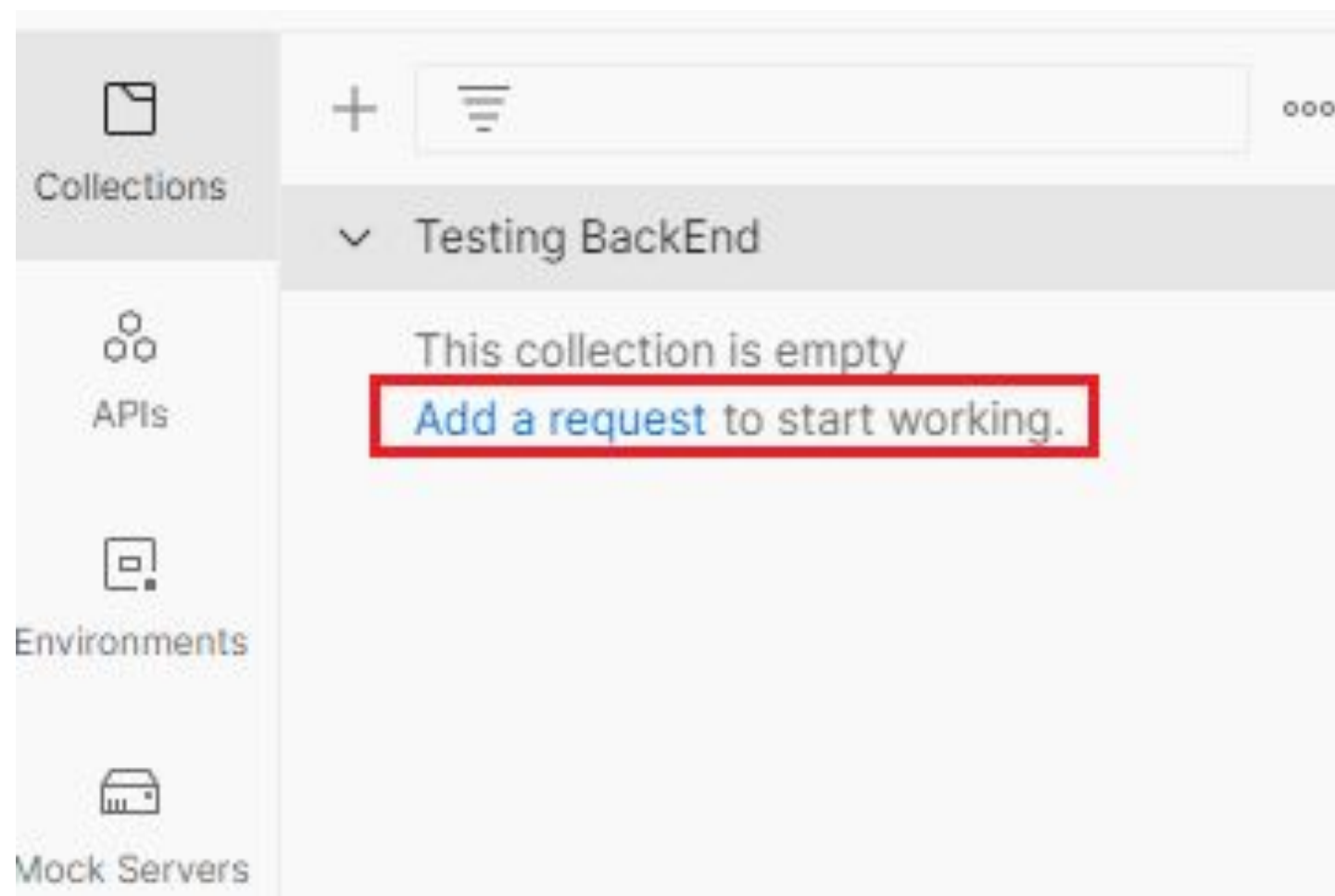
1 Primeiro, deve-se clicar na guia **Collections** e dentro desta guia clicar no botão “+”.



Passos para criar uma coleção

2

Você pode adicionar quantas solicitações forem necessárias a uma coleção, independente da quantidade. Basta clicar em **“Add a request”** na pasta da coleção.



02

Variáveis de ambiente

Variáveis de ambiente no Postman

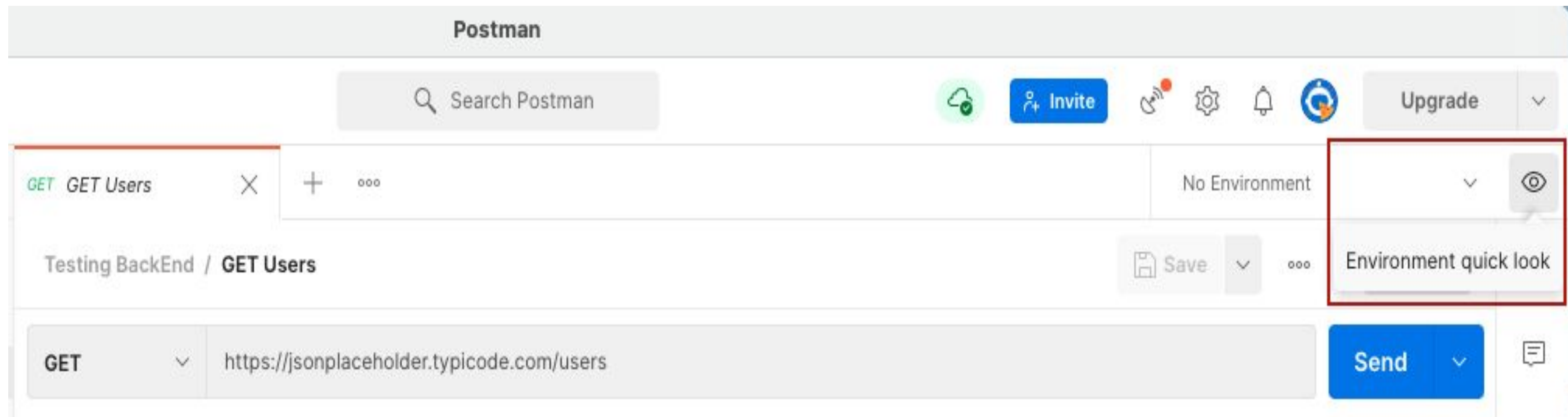
Temos a tendência de utilizar a mesma solicitação várias vezes com dados diferentes. O Postman nos permite parametrizar estes dados e salvá-los como um arquivo ou em variáveis de ambiente.

Uma **variável de ambiente** é salva no ambiente de trabalho. Elas podem ser criadas de maneira estática ou dinâmica. Poderíamos ter ambientes diferentes para Dev, QA e Produção, com suas respectivas variáveis.



Passos para criar uma variável de ambiente

1 Para criar uma variável de ambiente, devemos clicar no ícone de visualização.



Passos para criar uma variável de ambiente

- 2** O próximo passo é preencher os dados do ambiente e das variáveis. Como?
- Insira um nome para o ambiente. Por exemplo: DEV, QA ou Produção.
 - Insira as variáveis para este ambiente. Para fazer isto, você deve completar o nome da variável e seu valor. Por exemplo:
<https://jsonplaceholder.typicode.com/users>
 - Por fim, clique em **Save** para efetuar as alterações.

The screenshot shows the environment configuration interface. At the top, there's a header with 'Test', 'GET Untitled Request', 'Ambiente QA', and 'No Environment'. Below this, there's a table with columns: 'VARIABLE', 'INITIAL VALUE', 'CURRENT VALUE', and actions. A variable named 'URL' is added with the value 'https://jsonplaceholder.typicode.com/users'. The 'Save' button is highlighted in red.

VARIABLE	INITIAL VALUE	CURRENT VALUE	Actions
<input checked="" type="checkbox"/> URL	https://jsonplaceholder.typicod...	https://jsonplaceholder.typicode.com/users	<input type="checkbox"/> Persist All <input type="checkbox"/> Reset All

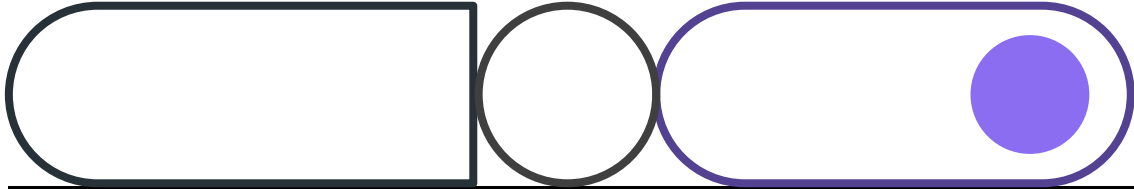
Passos para criar uma variável de ambiente

3 Por último, elas são criadas mediante o uso de chaves duplas com o nome da variável a ser utilizada, como na imagem a seguir.

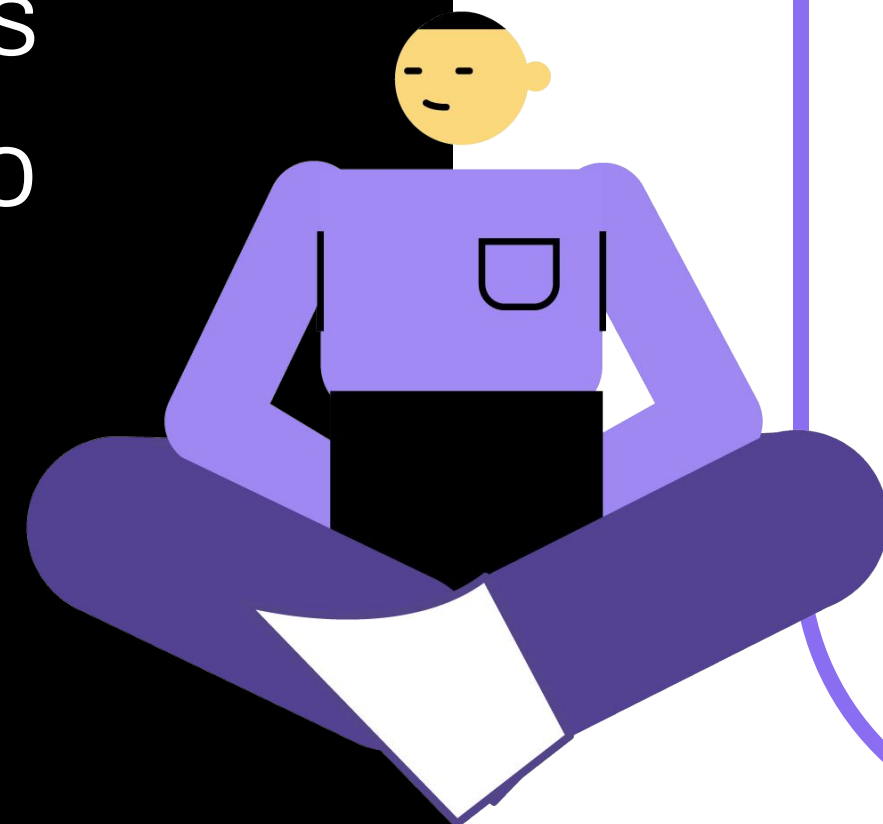


03

Runner para executar coleções

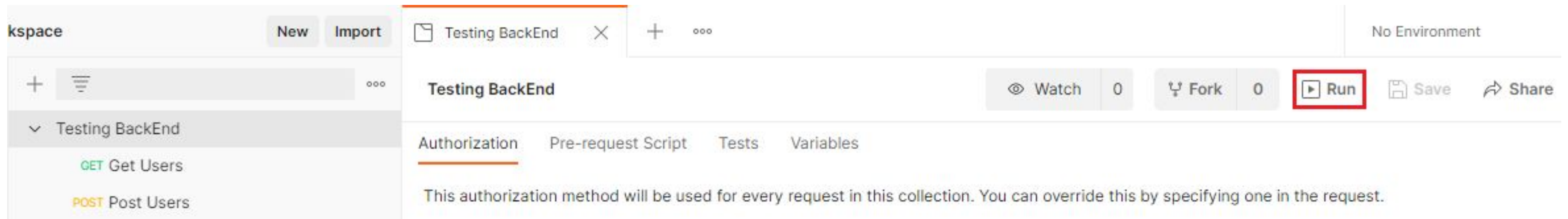


O ***runner*** nos permite executar um conjunto de testes de coleções diferentes ao mesmo tempo, fornecendo um relatório de resultados.



Passos para executar o *runner*

1 Primeiro, deve-se clicar no botão **Run**.

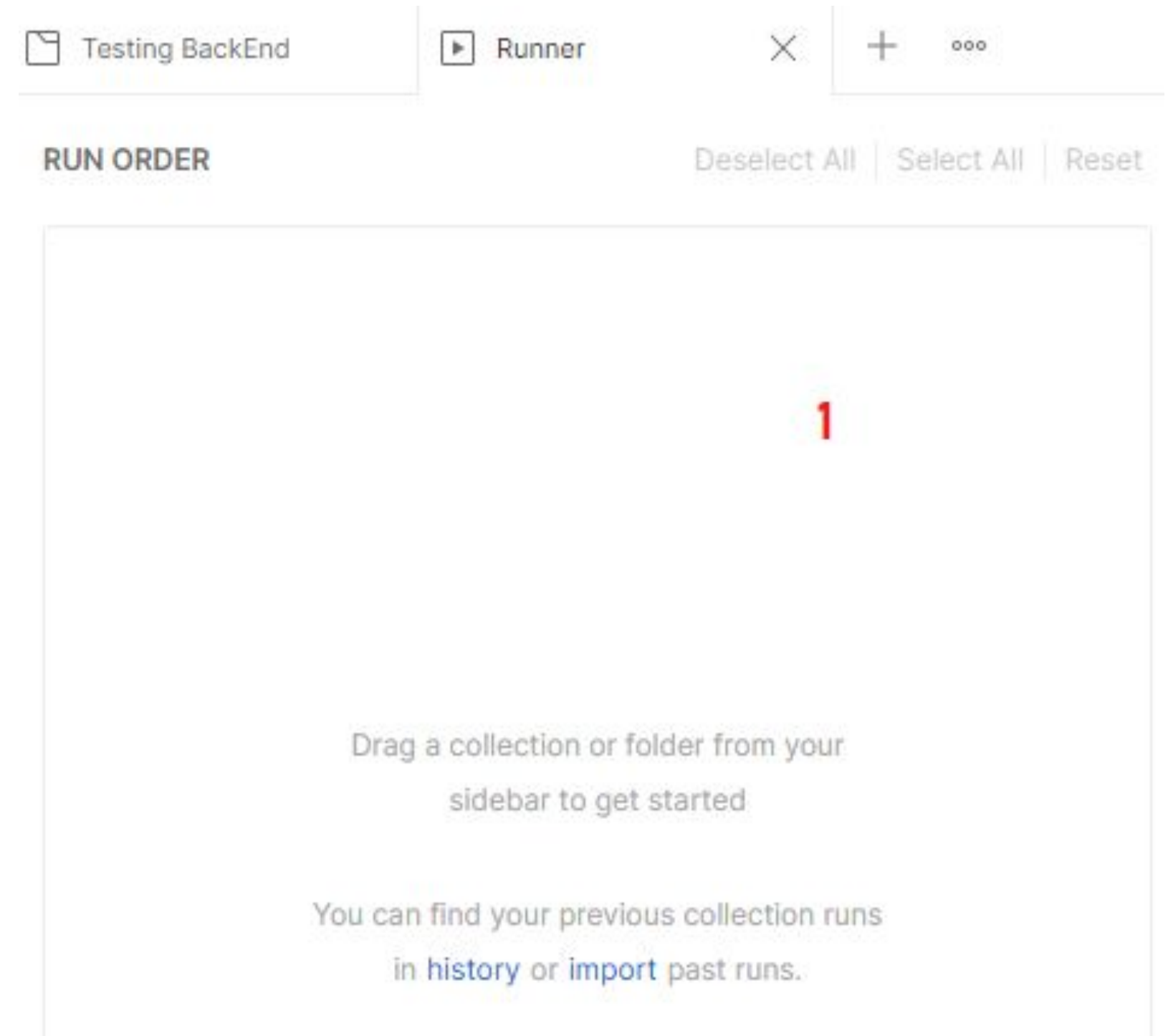


Passos para executar o *runner*

2

O próximo passo é completar os detalhes da execução. Como?

1. Seleccionamos a coleção e as solicitações nas quais desejamos executar seus testes.



Passos para executar o *runner*

2

2. Seleccionamos o ambiente no qual desejamos executar nossos testes. Desta forma, serão utilizadas as variáveis relacionadas a este ambiente.

3. Indicamos o número de vezes que vamos executar os testes.

4. Você pode configurar o tempo de atraso entre os testes.

5. Você pode selecionar um arquivo para salvar seus testes e resultados.

6. Os cookies podem ser salvos para uso em outros testes.

7. Clique no botão **Start Run**.

The screenshot shows a configuration interface for a testing runner. It includes a dropdown menu for selecting an environment, input fields for iterations and delay, a file selection button, and a section for checkbox options. A blue button at the bottom is labeled 'Run Testing BackEnd'.

2 Ambiente QA

Reset

3 Iterations 1

4 Delay 0 ms

5 Data Select File

6

- ☐ Save responses ⓘ
- ☒ Keep variable values ⓘ
- ☐ Run collection without using stored cookies
- ☒ Save cookies after collection run ⓘ

7 Run Testing BackEnd

Passos para executar o *runner*

3 Finalmente, podemos ver o resultado de nossas execuções.

The screenshot displays a testing runner interface. At the top, there are two circular status indicators: a green circle with the number '2' and the word 'PASSED' below it, and a red circle with the number '1' and the word 'FAILED' below it. To the right of these indicators, the text 'Testing' is displayed in a large font, with 'QA Entorno' and 'just now' in smaller text below it. Further right, there are four buttons: 'Run Summary' (orange), 'Export Results' (light gray), 'Retry' (blue), and 'New' (light gray). Below this header, the main content area is titled 'Iteration 1' on the left. It lists two test cases. The first test case is a 'GET' request to 'Get Users' with a placeholder 'URL'. It shows a status of '200 OK', a response time of '251 ms', and a response size of '6.807 KB'. The second test case is a 'POST' request to 'Post User' with a placeholder 'URL'. It shows a status of '201 Created', a response time of '317 ms', and a response size of '1.748 KB'. Below the second test case, there is a red error message: 'Status code is 201 | AssertionError: expected response to have status code 200 but got 201'. On the far right of the interface, there is a vertical orange bar with the number '1' at the top.

Muito obrigado!