

# API Testing - Colecciones y variables de entorno con Postman

**DigitalHouse** >  
Coding School



**Certified Tech  
Developer**  
The Ultimate Degree

# Índice

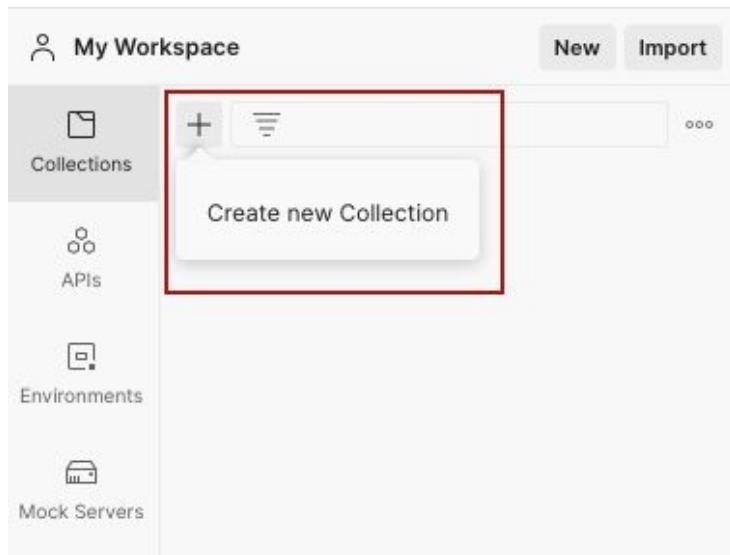
1. [Colecciones](#)
2. [Variables de entorno](#)
3. [\*Runner\* para correr colecciones](#)

# 1 | Colecciones

# Pasos para crear una Colección

1

Primero, se debe hacer clic en el botón "Crear nueva Colección" dentro de la pestaña colecciones.



“

Las colecciones son un grupo de **peticiones guardadas** que pueden organizarse en carpetas. Esto nos permite agrupar y administrar nuestras peticiones de manera más eficiente.



”

# Pasos

2

El siguiente paso es completar los datos de la nueva colección. ¿Cómo?

1. Ingresa el nombre de la colección.
2. Ingresa una descripción para la misma (opcional).
3. Haz clic en **CREAR** para crear la nueva colección.

The screenshot shows a 'CREATE A NEW COLLECTION' dialog box with a dark header and a close button. The main area has a 'Name' field with a placeholder 'Collection Name' and a 'Description' tab selected. The description field contains placeholder text. At the bottom, there are 'Cancel' and 'Create' buttons. Red numbers 1, 2, and 3 are overlaid on the image to indicate the steps: 1 points to the 'Name' field, 2 points to the 'Description' text area, and 3 points to the 'Create' button.

CREATE A NEW COLLECTION

Name

Collection Name

Description Authorization Pre-request Scripts Tests Variables

This description will show in your collection's documentation, along with the descriptions of its folders and requests.

Make things easier for your teammates with a complete request description.

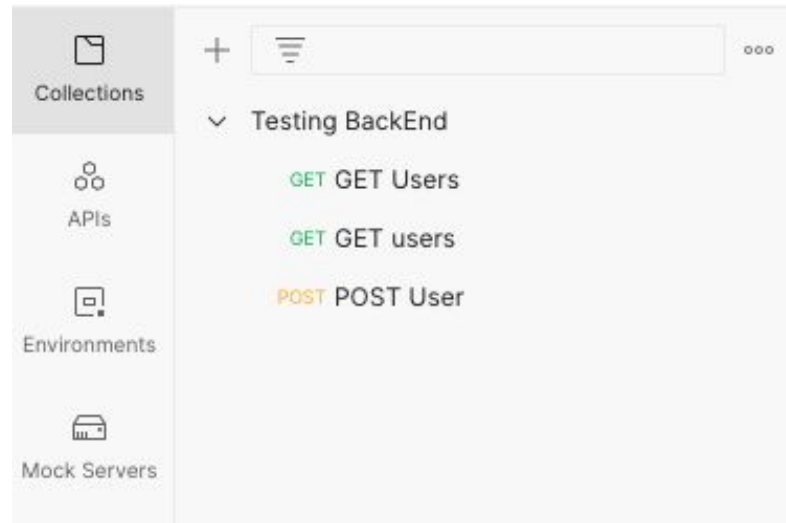
Descriptions support [Markdown](#)

Cancel Create

# Pasos

3

Puedes agregar cualquier número de peticiones a una colección. Solo debes arrastrar la petición a la carpeta de la colección.



# 2 | Variables de entorno



# Variables de entorno en Postman

Solemos utilizar la misma solicitud varias veces con datos diferentes. Postman nos permite parametrizar estos datos y guardarlos en forma de archivo o en variables de entorno.

Una **variable de entorno** se guarda en el entorno de trabajo. Estas se pueden crear de manera estática o dinámica.

Podríamos tener diferentes entornos para Dev, QA y Producción, con sus respectivas variables.

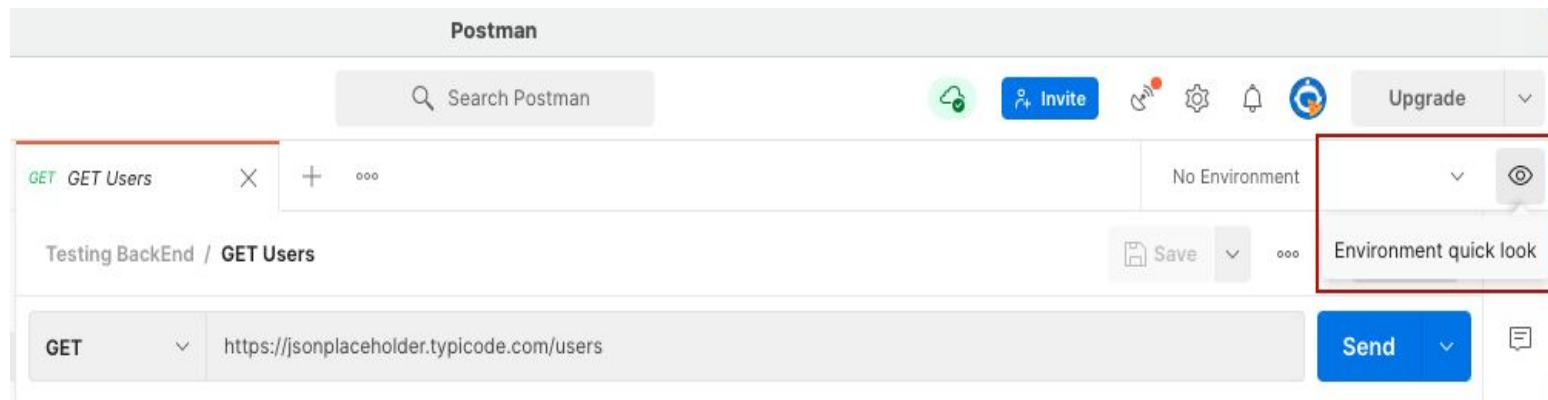


POSTMAN

# Pasos para crear una variable de entorno

1

Primero, se debe hacer clic en el botón del ojo para crear nuestras variables de entorno.



# Pasos

**2** El siguiente paso es completar los datos del entorno y de las variables. ¿Cómo?

- Ingresa el nombre del entorno (DEV, QA o Producción).
- Ingresa las variables de ese entorno. Para ello hay que completar el nombre de la variable y su valor.
- Finalmente haz clic en el botón **Guardar** para efectuar los cambios.

Entorno QA			Fork 0	Save  Share
	VARIABLE	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ	Persist All Reset All
<input checked="" type="checkbox"/>	URL	https://jsonplaceholder.typicode...	https://jsonplaceholder.typicode.com/users	
Add a new variable				

# Pasos

3

Por último, se crean mediante el uso de llaves dobles con el nombre de la variable a utilizar.



# 3 | *Runner* para correr colecciones



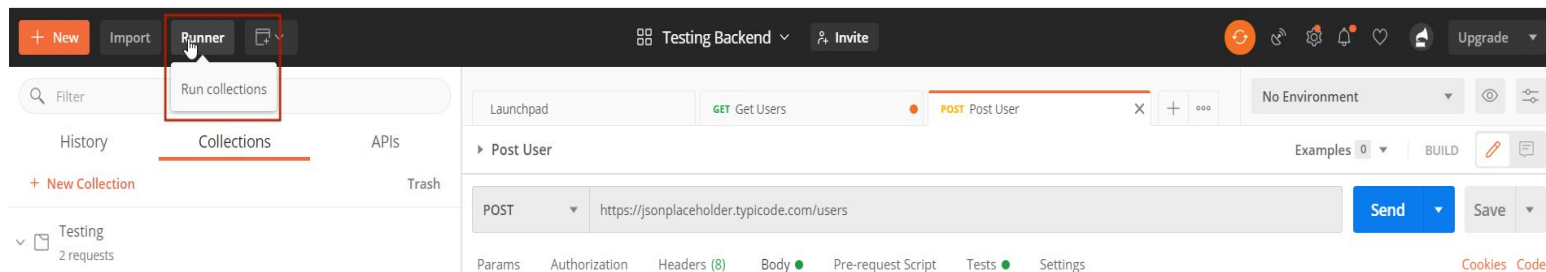
El ***runner*** nos permite ejecutar un conjunto de test de diferentes colecciones al mismo tiempo, otorgando un informe de resultados.



# Pasos para ejecutar el *runner*

1

Primero, se debe hacer clic en el botón *Runner*.

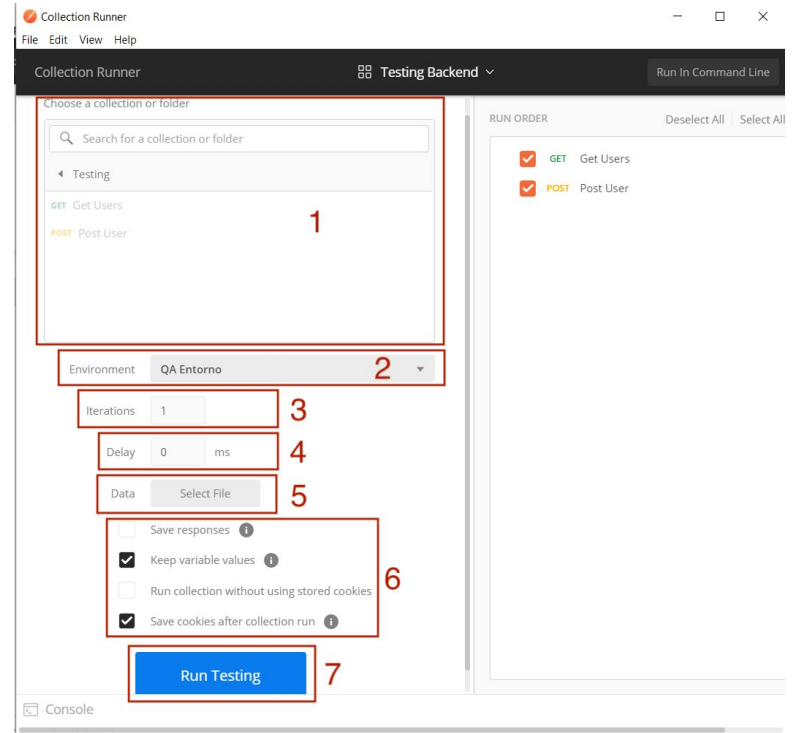


# Pasos

2

El siguiente paso es completar los datos de la ejecución ¿Cómo?

1. Seleccionamos las colecciones y peticiones en las que deseamos ejecutar sus test.
2. Seleccionamos el entorno en cual deseamos correr nuestros test. De esta manera se utilizaran las variables relacionadas con ese entorno.
3. Indicamos la cantidad de veces que vamos a correr los test.





# Pasos

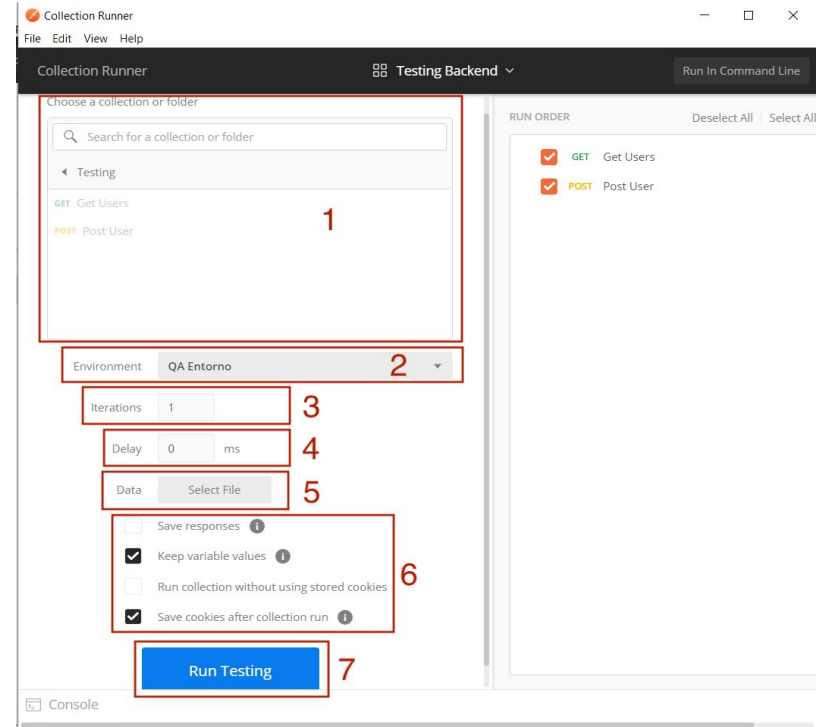
2

4. Puedes configurar el tiempo de demora entre prueba.

5. Puedes seleccionar un archivo para guardar sus pruebas y resultados.

6. Se pueden guardar las *cookies* para utilizarlas en otros test.

7. Haz click en el botón **Empezar Ejecución**.



# Pasos

3

Por último, podemos ver el resultado de nuestras ejecuciones.

The screenshot displays the 'Collection Runner' application window. The title bar includes 'Collection Runner' and standard window controls. The menu bar contains 'File', 'Edit', 'View', and 'Help'. The main interface has a dark header with 'Collection Runner', 'Run Results', and a 'Testing Backend' dropdown. On the right of the header are buttons for 'Run In Command Line' and 'Docs'. Below the header, a summary section shows '2 PASSED' in a green circle and '1 FAILED' in a red circle, followed by the text 'Testing QA Entorno just now'. To the right of this are buttons for 'Run Summary', 'Export Results', 'Retry', and 'New'. The main area shows 'Iteration 1' with a list of test steps. The first step is a 'GET Get Users' request that passed with a 200 OK status. The second step is a 'POST Post User' request that failed with a 201 Created status. The failure message is 'Status code is 201 | AssertionError: expected response to have status code 200 but got 201'. A sidebar on the right contains an 'Upgrade' dropdown, a 'Save' button, and links for 'Cookies' and 'Code'.

Collection Runner

File Edit View Help

Collection Runner Run Results Testing Backend Run In Command Line Docs

2 PASSED 1 FAILED Testing QA Entorno just now Run Summary Export Results Retry New

Iteration 1

Method	Endpoint	Response	Status	Time	Size
GET	Get Users {{URL}}	200 OK	251 ms	6.807 KB	
Status code is 200					
Check if user with id 1 is Leanne Graham					
POST	Post User {{URL}}	201 Created	317 ms	1.748 KB	
Status code is 201   AssertionError: expected response to have status code 200 but got 201					

Upgrade Save Cookies Code

DigitalHouse>  
Coding School