



GeeksHubs
academy _

POSTMAN

POSTMAN



¿Qué es Postman?



Postman surgió originariamente como una extensión para el navegador Google Chrome.

Es una herramienta que nos permite testear nuestras APIs creando una librería de tests no automatizados para poder interactuar con ella manualmente.

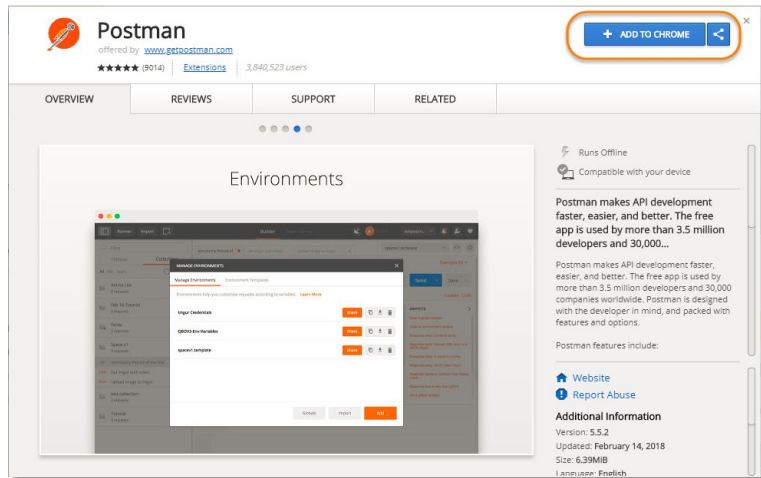
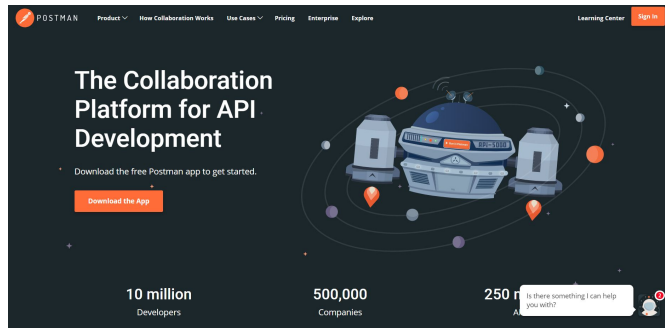
Por lo tanto podemos decir también que Postman es una herramienta muy cómoda y útil para estudiar RESTful APIs hechas por otras personas.



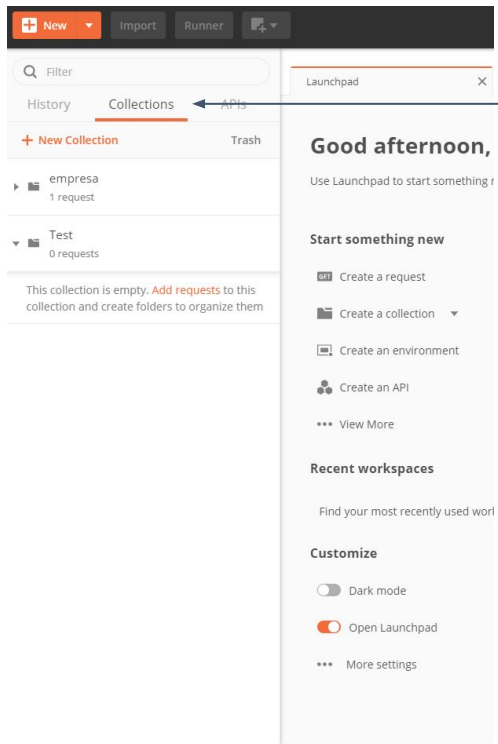
Instalando Postman

Vamos a la página web de [POSTMAN](https://www.getpostman.com) y nos descargamos la aplicación de escritorio para nuestro sistema operativo.

Se trata de una instalación sencilla y bastante rápida.



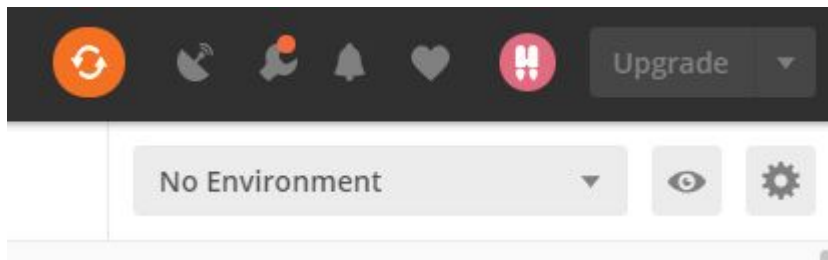
Configurando Postman



En la parte izquierda de nuestra aplicación de postman tenemos las Colecciones, vamos a crear una nueva colección para empezar a trabajar en ella.

Configurando Postman

Para configurar los entornos y sus variables, verás que en la esquina superior derecha de la interfaz hay un desplegable en el que se puede leer “No environments”. Si haces clic en el botón de ajustes que está al lado, aparece una opción llamada “Manage environments” donde se pueden configurar cada uno de ellos:



Configurando Postman

Una de las virtudes más potentes de Postman, desde mi punto de vista, es la posibilidad de definir todas las variables que necesites y clasificarlas por entornos de trabajo.

Después de tener nuestra colección creada, necesitamos configurar un entorno en el que guardar nuestras variables de entorno como por ejemplo la URL de nuestra API.

Dentro de esta configuración de entorno vamos a añadirle el siguiente conjunto clave-valor:

`url -> localhost:8080`

MANAGE ENVIRONMENTS

Environment Name

Pruebas

	VARIABLE	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ	...	Persist All	Reset All
<input checked="" type="checkbox"/>	url	localhost:8080	localhost:8080			
	Add a new variable					

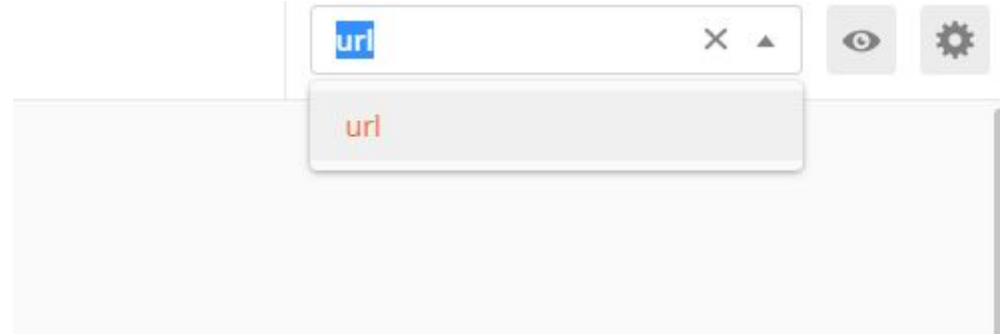
ⓘ Use variables to reuse values in different places. The current value is used while sending a request and is never synced to Postman's servers. The initial value is auto-updated to reflect the current value. [Change](#) this behaviour from Settings. [Learn more about variable values](#)

Cancel

Update

Seleccionando Nuestro Entorno

En la esquina superior derecha vemos un desplegable de selección de entorno donde vamos a seleccionar el que acabamos de crear.



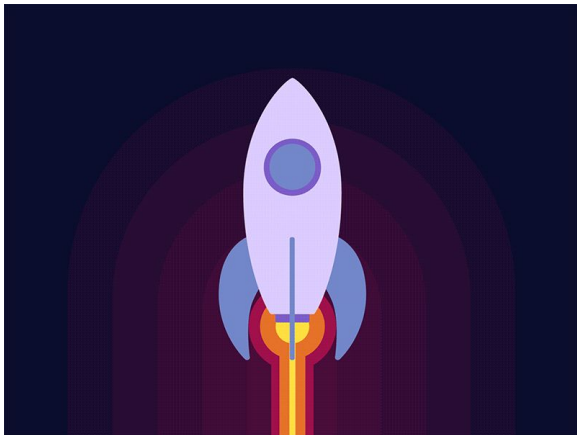
Investigación Tecnológica



Investigación tecnológica: Postman

Estudia acerca de los conceptos “colecciones de peticiones”, “token de autenticación” y finalmente “cabeceras asociadas a la petición”.

Ánimo Geek! No hemos hecho más que empezar!



GET

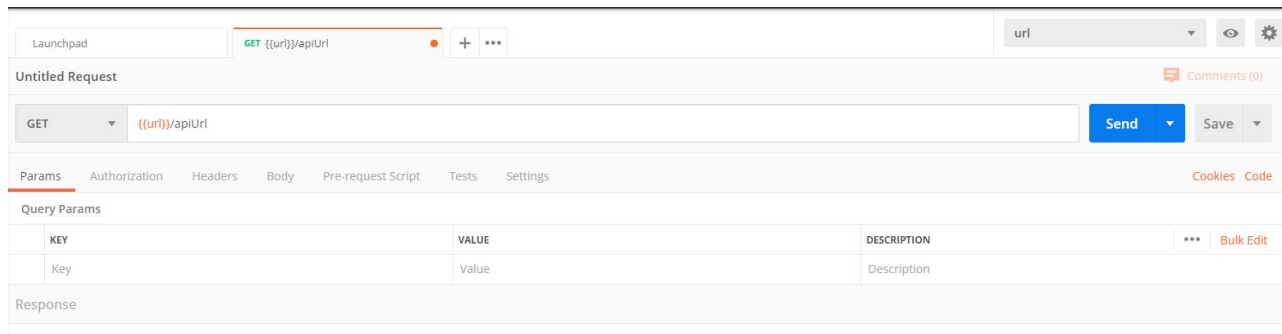


Creando una petición GET

En la parte central de Postman tenemos pestañas para peticiones, vamos a seleccionar el método GET y escribir la url de la siguiente manera:

```
{{url}}/apiUrl
```

La variable de la url de va a recoger del entorno en el que estemos trabajando, de esta manera tenemos acceso a las variables de entorno.

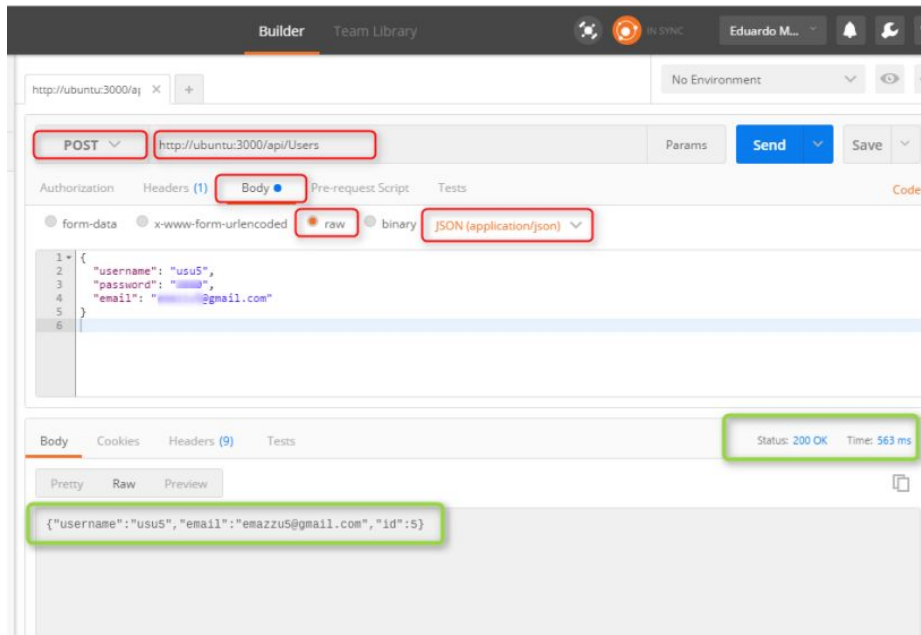


Explorando la respuesta.

Una vez hagamos una petición GET a una URL válida que proporcione una respuesta.

`{{url}}/users/findbyid`

Podemos leerla justo debajo en la misma pestaña, donde nos va a devolver la información en forma de texto plano, tenemos que seleccionar el tipo de respuesta y la forma en la que nos va a mostrar los datos para poder tener un texto claro y legible.



Guardando la petición

Para poder reutilizar la petición, tenemos que guardarla desde el botón SAVE en la esquina superior derecha seleccionando la colección en la que queremos que se guarde.

SAVE REQUEST

Requests in Postman are saved in collections (a group of requests).
[Learn more about creating collections](#)

Request name

Request description (Optional)

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

Descriptions support [Markdown](#)

Select a collection or folder to save to:

◀ Test

+ Create Folder

Cancel

Save to Test

KATA



KATA I: Postman - Petición GET

El primer paso consistirá en tener un servidor activo y tener conocimiento de su url.

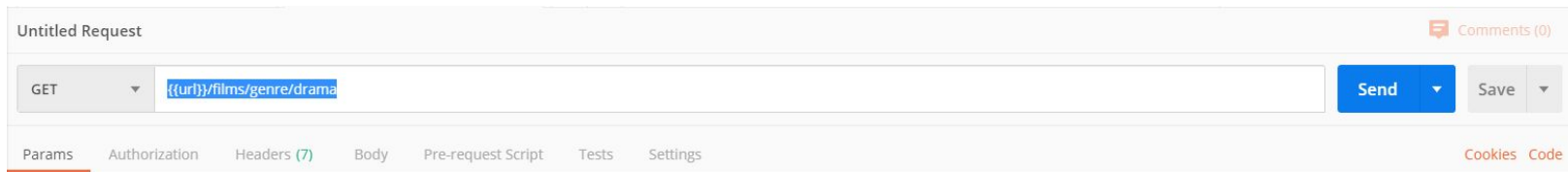
En mi caso voy a realizar una petición GET sobre una base de datos que tengo alojada de forma personal a la que se puede acceder desde un servidor local, haciendo uso de una API.



Paso 2: Envío de la petición GET

Introduzco la siguiente URL en la barra de peticiones de POSTMAN.

`{{url}}/films/genre/drama`



Reto



Reto: Realiza tu primera petición GET

En el caso de que dispongas de una base de datos y un servidor de acceso en entorno local, además **por supuesto** de la API correspondiente, realiza tu primera petición GET a través de Postman.



POST

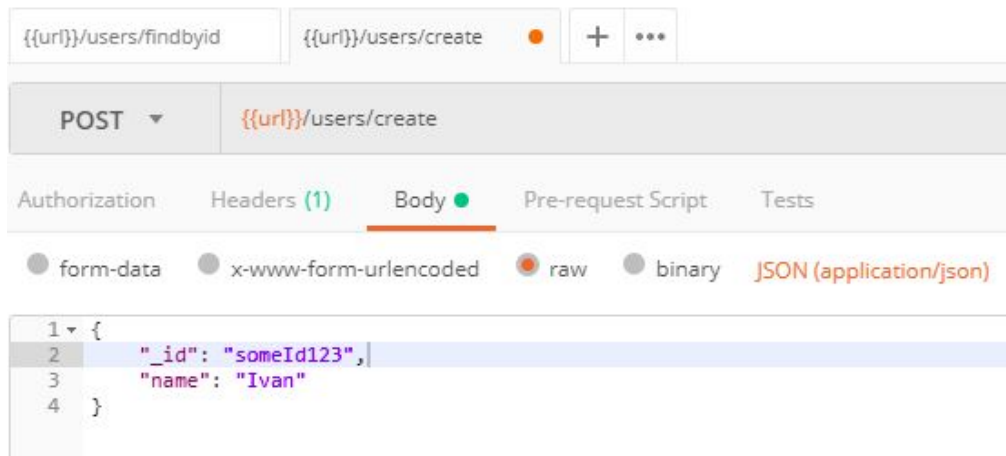


Enviando datos mediante POST

Para ello tenemos que crear una petición de tipo POST e indicar en la pestaña "Body" la información que queremos enviar.

Como estamos trabajando con formato JSON, seleccionaremos formato "raw" y dentro de él cambiaremos el formato de Text a JSON(application/json).

Ahora ya podemos enviar nuestros datos escribiendo un objeto en formato JSON.



KATA II



KATA II: Postman - Petición POST

A continuación voy a realizar una petición POST sobre una base de datos que tengo alojada de forma personal a la que se puede acceder desde un servidor local, haciendo uso de una API.



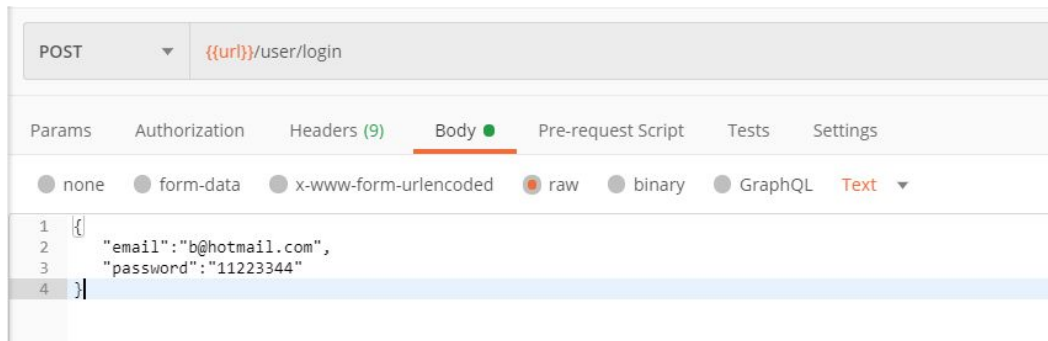
Paso 2: Envío de la petición POST

Introduzco la siguiente URL en la barra de peticiones de POSTMAN.

```
{{url}}/user/login
```

En el apartado correspondiente al body, el siguiente código:

```
{  
  "email":"b@hotmail.com",  
  "password":"11223344"  
}
```



Reto



Reto: Realiza tu primera petición POST

Al igual que en el anterior reto, si dispones de una base de datos y un servidor de acceso en entorno local, además **por supuesto** de la API correspondiente, realiza tu primera petición POST a través de Postman.



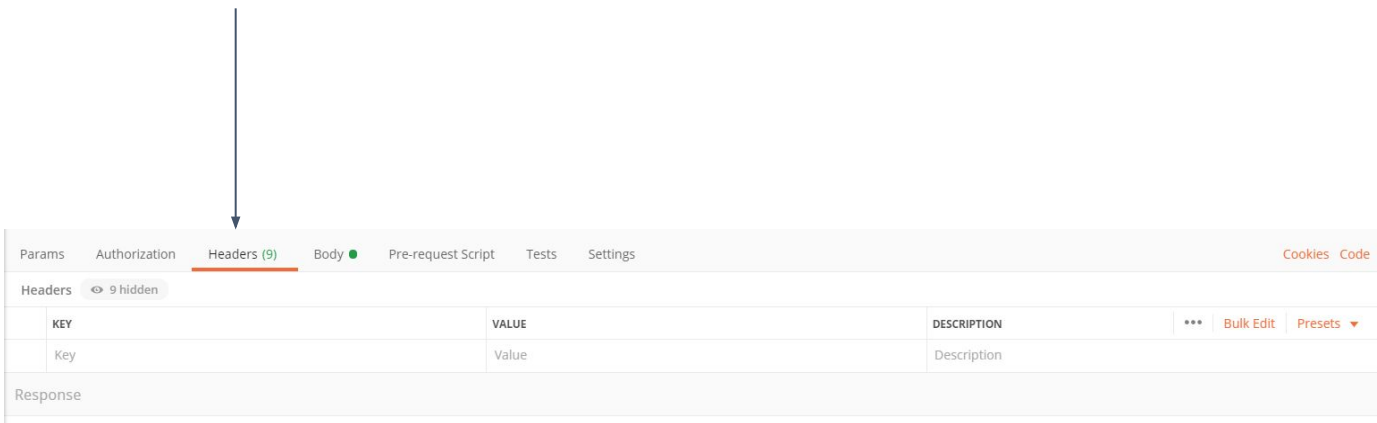
Headers, Variables y Scripts



Mandando HEADERS

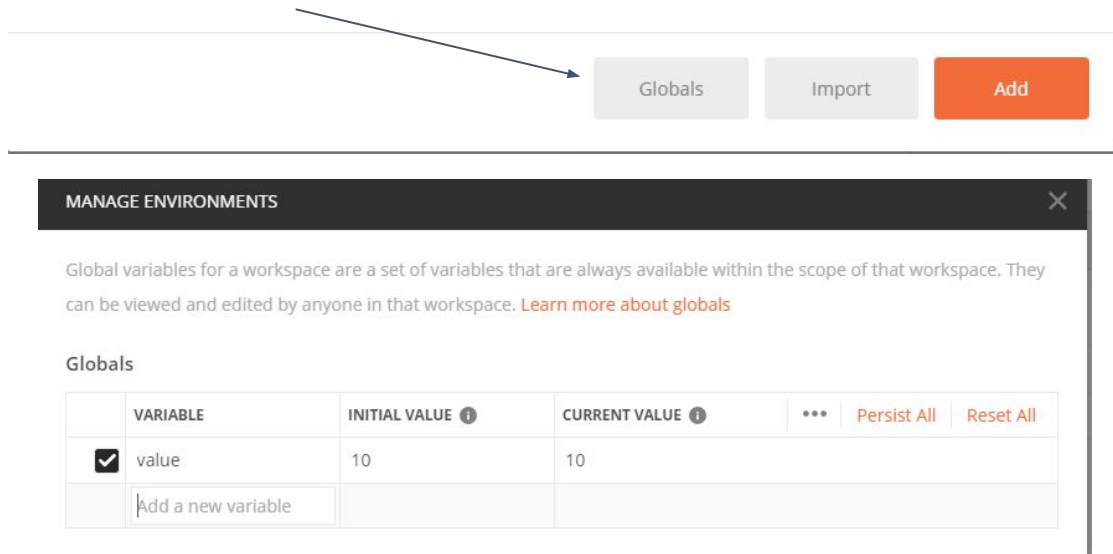
En la pestaña de la izquierda de Body podemos elegir mediante un formulario de clave-valor los headers que se van a enviar en nuestra petición.

De esta manera podemos enviar un token de autenticación fijo añadido manualmente.



Creación de variables globales

De la misma manera que hemos creado la variable URL, podemos crear variables de entorno dinámicas para trabajar con nuestra API sin tener que preocuparnos de cambiar manualmente todas las cabeceras o URLs.



The screenshot shows a web interface for managing environments. At the top, there are three buttons: 'Globals' (highlighted with a blue arrow), 'Import', and 'Add'. Below this is a modal window titled 'MANAGE ENVIRONMENTS' with a close button (X). Inside the modal, there is a description of global variables and a link to 'Learn more about globals'. Below the description, the section 'Globals' contains a table with columns for 'VARIABLE', 'INITIAL VALUE', and 'CURRENT VALUE'. The table has one row with a checked checkbox, the variable name 'value', and the value '10'. At the bottom of the table is a text input field labeled 'Add a new variable'.

	VARIABLE	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ	...	Persist All	Reset All
<input checked="" type="checkbox"/>	value	10	10			
	<input type="text" value="Add a new variable"/>					

Cómo crear Scripts

Para crear un script tenemos que ir a la petición donde queremos que se ejecute, y dentro de la pestaña Tests escribiremos el código JavaScript.

A la hora de escribir un test usaremos el siguiente formato:

```
pm.test('nombre_del_test', function() {  
    // closure function que contendrá todos los asserts a ejecutar  
});
```



Cómo crear Scripts

Es hora de ver un ejemplo:

```
pm.test("Payload must be valid and have a body", function () {  
    pm.response.to.be.ok; // equivalente a decir que valide que  
    sea 200  
  
    pm.response.to.be.withBody; // body tiene que existir  
  
    pm.response.to.be.json; // además que sea un JSON válido  
});
```

Cuando la petición se sirve, se ejecuta el test que hemos escrito y el resultado lo podemos ver en la pestaña de **Test Results**.



Investigación Tecnológica



Investigación tecnológica: Headers, Variables & Scripts

Investiga si existe la posibilidad y como podrías hacerlo, para atrapar valores tanto del body como de header de una respuesta.

Rock on!



Atrapando valores & x-auth



Atrapando valores del body

Para conseguir acceder a nuestro body, tenemos que hacerlo haciendo un `JSON.parse` (del body en este caso) para convertirlo a un objeto JavaScript, y después asignamos el valor concreto que queramos a nuestro entorno.

```
let body = JSON.parse(responseBody);  
postman.setEnvironmentVariable('user-id', body.user._id);
```

en este caso la respuesta habría sido parecida a esta:

```
{  
  "user": {  
    "_id": "3f21s9r0",  
    "active": true  
  },  
  "timestamp": 121093432  
}
```



Atrapando valores de los headers

Para acceder a los headers de una respuesta tenemos `postman.getResponseHeader`, que recibe como argumento la clave de valor a recuperar.

```
let token = postman.getResponseHeader('x-auth');  
postman.setEnvironmentVariable('x-auth', token);
```



Usando la nueva variable x-auth

Ahora podemos utilizar estas variables para mandar un token de autenticación dinámico, que recogerá el valor del response en nuestra petición de login.

► POST /users/me/phone

POST ▼	{{url}}/api/users/me/phone	
Authorization	Headers (2)	Body ●
	Pre-request Script	Tests ●
	Key	Value
<input checked="" type="checkbox"/>	Content-Type	application/json
<input checked="" type="checkbox"/>	x-auth	{{x-auth}}
	New key	Value

Más usos para las variables

Además de un token de autenticación, podemos añadir a nuestro entorno, por ejemplo, el ID de la última tarea creada desde postman en otra petición para que se convierta en la tarea predefinida a la que hacer peticiones.

▶ GET /todos/:id

GET ▾	{{url}}/todos/{{todoid}}			
Authorization	Headers (2)	Body	Pre-request Script	Tests
	Key	Value	Description	
<input checked="" type="checkbox"/>	Content-Type	application/json		
<input checked="" type="checkbox"/>	x-auth	{{x-auth}}		
	New key	Value	Description	

Reto-Investigación



Investigación tecnológica: atrapando valores & x-auth

¿Serías capaz de mandar un token de autenticación dinámico? ¿Cuál sería la forma correcta de enviarlo?

¿Serías capaz de crear una variable y almacenarla para a su vez convertirla en una tarea predefinida a la que hacer peticiones?

Ánimo Geek! You code...you can!

