

API REST con Spring

¿Qué es Postman?

Postman es una herramienta fundamental para desarrolladores de software, diseñada para facilitar la prueba y el desarrollo de **APIs (Interfaz de Programación de Aplicaciones)**. Permite enviar solicitudes HTTP a una API, inspeccionar las respuestas y realizar pruebas de **funcionalidad, seguridad y rendimiento** de manera eficiente.

Gracias a su interfaz gráfica intuitiva, Postman simplifica la creación, organización y ejecución de pruebas de API. Además, ofrece funciones avanzadas como:

- **Automatización de pruebas** para validar respuestas y flujos de trabajo.
- **Generación de documentación de APIs** para facilitar su uso y comprensión.
- **Colaboración en equipo**, permitiendo compartir colecciones de solicitudes y entornos.

Postman permite realizar pruebas tanto **manuales como automatizadas**, convirtiéndose en una herramienta versátil para el desarrollo y la depuración de APIs.

¿Cómo utilizar Postman?

1. Aplicación de escritorio

- Disponible para Windows, macOS y Linux.
- Ofrece acceso completo a todas las funcionalidades, incluyendo la capacidad de trabajar sin conexión.
- **Recomendado para pruebas en servidores locales (localhost)**, ya que algunas funciones no están disponibles en la versión web.



Puedes descargar la aplicación desde el [sitio oficial de Postman](#).

2. Versión web

- Permite ejecutar pruebas directamente desde el navegador sin necesidad de instalación.
- Requiere iniciar sesión con una cuenta de Postman para sincronizar colecciones y entornos.

 Puedes acceder a la versión web en [Postman Web](#).

Para ambas opciones, es recomendable crear una cuenta para sincronizar tus solicitudes y colecciones en la nube, facilitando el acceso desde cualquier dispositivo.

ELEMENTOS PRINCIPALES DE LA HERRAMIENTA

1. Espacio de trabajo en Postman

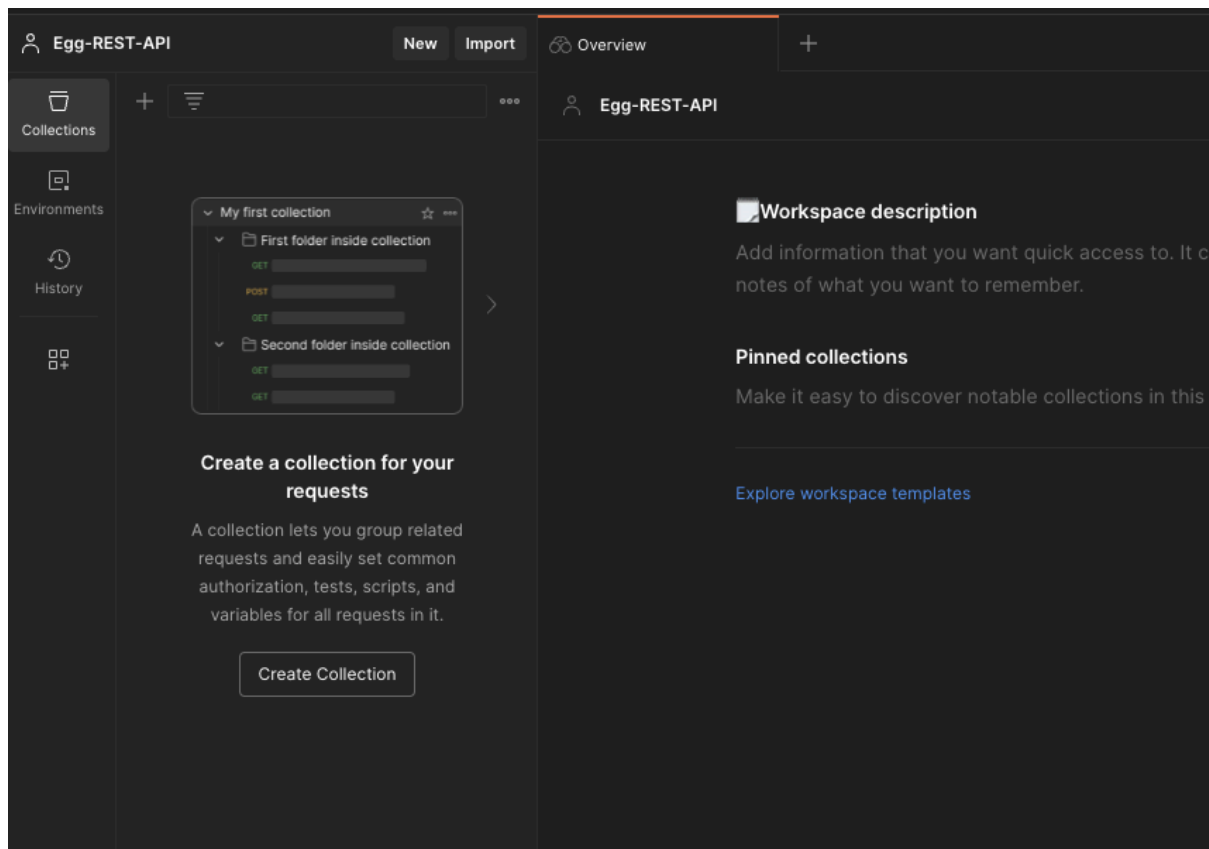
Una de las principales ventajas de Postman es su **espacio de trabajo**, ya que permite organizar y gestionar los diferentes elementos de un proyecto, como:

- **Colecciones** de solicitudes.
- **APIs** y documentación.
- **Variables de entorno** para diferentes configuraciones.
- **Monitores** para pruebas automatizadas.
- **Colaboración** con otros miembros del equipo.

Creación de un espacio de trabajo

Para crear un nuevo espacio de trabajo en Postman, sigue estos pasos:

1. Abre el menú **Workspaces** y selecciona **Create Workspace**.
2. Asigna un **nombre** al espacio de trabajo.
3. Define quiénes tendrán acceso: puede ser privado (solo tú) o compartido con tu equipo.



Navegación en el espacio de trabajo

Una vez creado el espacio de trabajo, encontrarás en la parte izquierda una **barra de acciones** que te permitirá acceder y gestionar los distintos elementos.

Si presionas el icono de **más opciones** , podrás ver las acciones disponibles para cada elemento dentro del espacio de trabajo.

2. Las colecciones en Postman

Postman permite realizar peticiones a un servicio de forma sencilla. Sin embargo, lo más recomendable en el desarrollo diario es **organizar las peticiones en colecciones**.

Una colección en Postman te permite:

- **Agrupar** diferentes solicitudes relacionadas con un servicio.
- **Organizar** mejor las pruebas y escenarios de validación.
- **Estructurar** el proyecto cuando trabajas con múltiples servicios.
- **Facilitar la colaboración**, permitiendo que otros miembros del equipo accedan y utilicen las mismas solicitudes.
- **Compartir** las colecciones mediante un enlace o exportarlas para su uso en otros entornos.

¿Por qué usar colecciones?

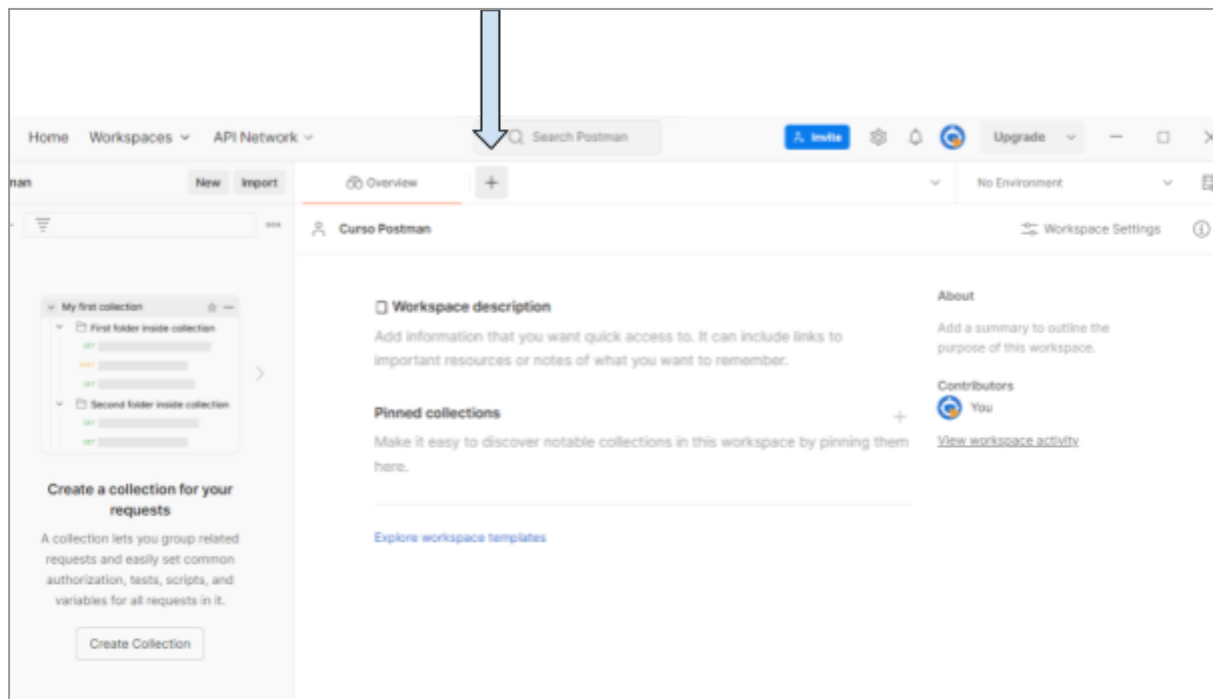
Si tu proyecto involucra varios servicios desplegados, lo ideal es crear una **colección específica para cada uno**. Esto no solo ayuda a mantener un flujo de trabajo ordenado y bien documentado, sino que también permite a los diferentes miembros del equipo trabajar en paralelo sin interferencias.

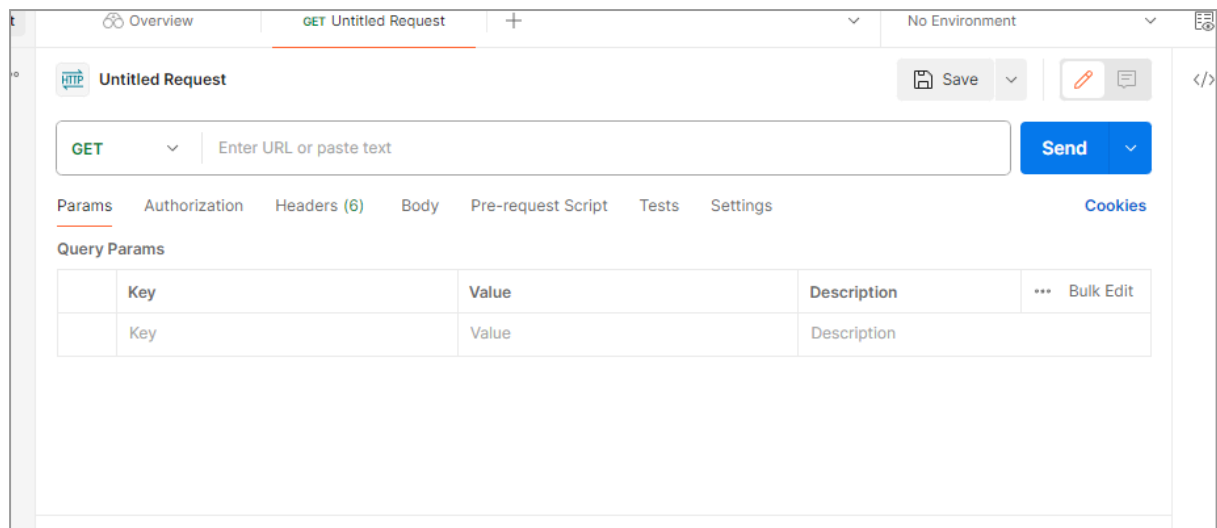
IMPORTANTE - En Postman, las colecciones **no solo agrupan peticiones**, también pueden representar **flujos de trabajo** o **conjuntos de pruebas automatizadas**.

3. Creación de nueva petición

Para realizar una petición en Postman, puedes seguir estos pasos:

- **Crear una nueva petición:** Haz clic en el ícono de **"Nueva petición"**.
- **Definir los parámetros clave:**
 - **Verbo HTTP** (GET, POST, PUT, PATCH, DELETE, etc.).
 - **URL del servicio** al que deseas enviar la solicitud.
 - **Encabezados** necesarios (como **Content-Type** o **Authorization**).
 - **Cuerpo de la petición** (si aplica).
 - **Guardar la configuración** antes de ejecutar la petición.
 - **Enviar la solicitud** presionando el botón **"Enviar"**.





4. Nuevo Entorno

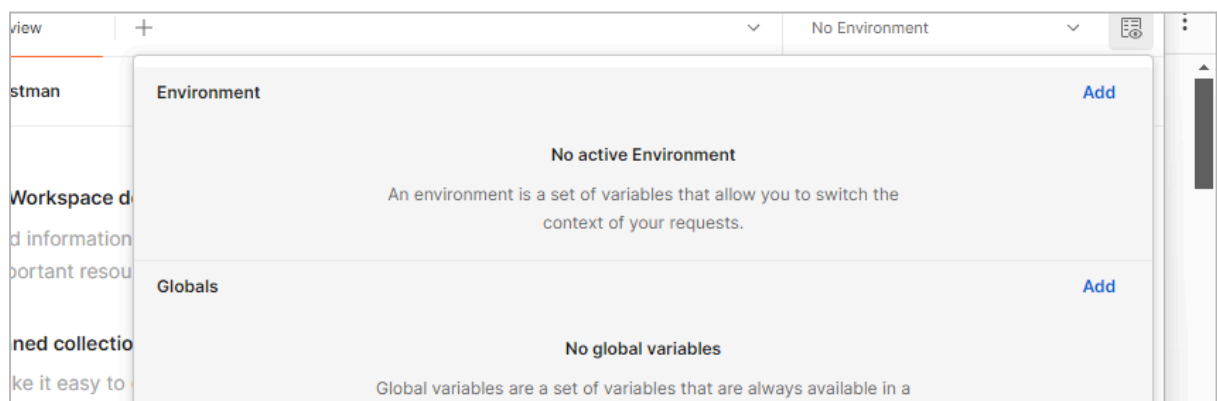
En Postman, un **entorno** es una configuración personalizada que permite manejar **variables reutilizables** dentro de tus peticiones, como:

- **URLs de servidor** (desarrollo, pruebas, producción).
- **Tokens de autenticación** y claves de API.
- **Cualquier otro valor dinámico** utilizado en múltiples solicitudes.

Trabajar con entornos te permite:

- Simular distintos escenarios sin modificar manualmente cada petición.
- Evitar la reescritura de valores al cambiar de ambiente (ejemplo: desarrollo, producción, calidad).
- Mejorar la organización y eficiencia al realizar pruebas en múltiples entornos.

Para crear un entorno, ve al menú **"Environments"**, haz clic en **"New Environment"** y define las variables que utilizarás en tus peticiones.



5. Configuración de variables

Las **variables** en Postman permiten almacenar información dinámica, como:

- **URLs de servidores** (desarrollo, pruebas, producción).
- **Tokens de autenticación y claves de API**.
- **Datos de prueba** (como identificadores de usuario, parámetros de consulta, etc.).

El uso de variables hace que las solicitudes sean **más dinámicas y adaptables** a diferentes entornos o escenarios sin necesidad de modificar cada petición manualmente.

Tipos de Variables en Postman

- ♦ **Variables de entorno:** Se definen dentro de un entorno específico (ejemplo: desarrollo, producción, pruebas).
- ♦ **Variables globales:** Son accesibles desde cualquier entorno y colección en Postman.
- ♦ **Variables locales:** Se usan dentro de una sola solicitud y desaparecen cuando esta finaliza.
- ♦ **Variables de colección:** Se asocian a una colección específica y pueden ser utilizadas en todas sus solicitudes.
- ♦ **Variables de usuario:** Se configuran a nivel de usuario y permiten almacenar datos privados o sensibles.

Cómo Crear una Variable en Postman

- Abre **Postman** y ve a la pestaña "**Environments**".
- **Edita un entorno** existente o crea uno nuevo.
- Agrega un **nombre, tipo y valor** a la variable.
- Guarda los cambios y usa la variable en tus peticiones con la sintaxis `{{nombreVariable}}`.

⚡ **Ejemplo:** Si defines una variable llamada `baseUr1`, puedes usarla en una solicitud así: `{{baseUr1}}/api/usuarios`

Esto te permitirá cambiar el valor de `baseUr1` sin modificar cada solicitud manualmente.

New Environment			Edit
Variable	Initial value	Current value	
base_url	https://jsonplaceholder.typicode.com	https://jsonplaceholder.typicode.com	
Globals			Add
No global variables			
Global variables are a set of variables that are always available in a workspace.			

6. Revisión de respuestas y estados

Cuando realizamos una solicitud en Postman, podemos analizar la **respuesta** que nos devuelve el servidor, lo que nos permite validar si nuestra API está funcionando correctamente.

Elementos Clave en la Respuesta

- **Cuerpo de la respuesta (Response Body):** Contiene la información devuelta por el servidor. Puede estar en formato **JSON, XML, HTML o texto plano**.
- **Código de estado HTTP:** Indica el resultado de la solicitud. Algunos códigos comunes son:
 - **200 OK** → La solicitud fue exitosa.
 - **201 Created** → Se creó un nuevo recurso correctamente.
 - **400 Bad Request** → La solicitud tiene errores en los datos enviados.
 - **401 Unauthorized** → Falta autenticación o las credenciales son incorrectas.
 - **404 Not Found** → El recurso solicitado no existe.
 - **500 Internal Server Error** → Error en el servidor.
- **Tiempo de respuesta:** Muestra cuánto tiempo tardó el servidor en procesar la solicitud.
- **Encabezados de la respuesta (Response Headers):** Contienen metadatos importantes sobre la respuesta, como el tipo de contenido o la política de caché.

¿Dónde Ver esta Información en Postman?

Una vez enviada la solicitud, Postman mostrará los resultados en la parte inferior de la pantalla, organizados en pestañas como **"Body"**, **"Headers"**, **"Cookies"** y **"Test Results"**.

Si necesitas depurar errores, revisa el código de estado y el cuerpo de la respuesta para entender qué puede estar fallando en la solicitud.



7. Buenas prácticas al utilizar la herramienta

Para optimizar el uso de **Postman** en la prueba de APIs, es recomendable seguir estas **buenas prácticas**:

- **Definir el Punto de Entrada:** Asegúrate de conocer la **URL base** y el **endpoint** que vas a probar. Comprueba si necesitas autenticación o parámetros adicionales.
- **Comprender el Funcionamiento de la API:** Revisa la **documentación** de la API para entender cómo interactuar con ella. Conoce qué **métodos HTTP** soporta: **GET**, **POST**, **PUT**, **PATCH**, **DELETE**, etc.
- **Definir el Propósito de la Prueba:** Antes de ejecutar una solicitud, pregunta:
 - ¿Qué acción realizará esta llamada?
 - ¿Qué respuesta espero obtener?
 - ¿Requiere datos en el cuerpo de la petición?

- **Configurar los Parámetros Correctamente:** Postman muestra respuestas en JSON, XML, HTML o texto plano. Familiarízate con la estructura esperada y asegúrate de que los datos sean correctos.
- **Identificar Precondiciones y Postcondiciones:** Antes de ejecutar una prueba, revisa si la API necesita ciertos datos previos. Después de la ejecución, verifica que los cambios en la base de datos o sistema sean los esperados.
- **Ejecutar y Verificar las Pruebas:** Realiza pruebas unitarias con diferentes escenarios, incluyendo **casos positivos y negativos**. Observa los **códigos de estado HTTP** y el contenido de la respuesta.
- **Reportar Errores de Forma Clara:** Si encuentras fallos en la API, documenta:
 - **Qué solicitud hiciste** (método, endpoint, parámetros).
 - **Qué respuesta obtuviste** (código de estado, mensaje de error).
 - **Qué esperabas recibir.**

Siguiendo estas prácticas, mejorarás la eficiencia y precisión en las pruebas de API con Postman.

Postman es una herramienta con múltiples funcionalidades y opciones avanzadas. Te recomendamos explorar su [documentación oficial](#) para profundizar en los temas abordados y descubrir más sobre su uso.