

# Saturn Studio – Webhooks



Los **Webhooks** son herramientas de comunicación en tiempo real que permiten a aplicaciones externas (como Facebook, GitHub o servicios de pago) enviar notificaciones automáticas a **Saturn Studio**. En lugar de que tu robot esté preguntando constantemente si hay novedades, el sistema externo envía la información de manera inmediata a una dirección URL única, lo que dispara la ejecución del bot al instante.

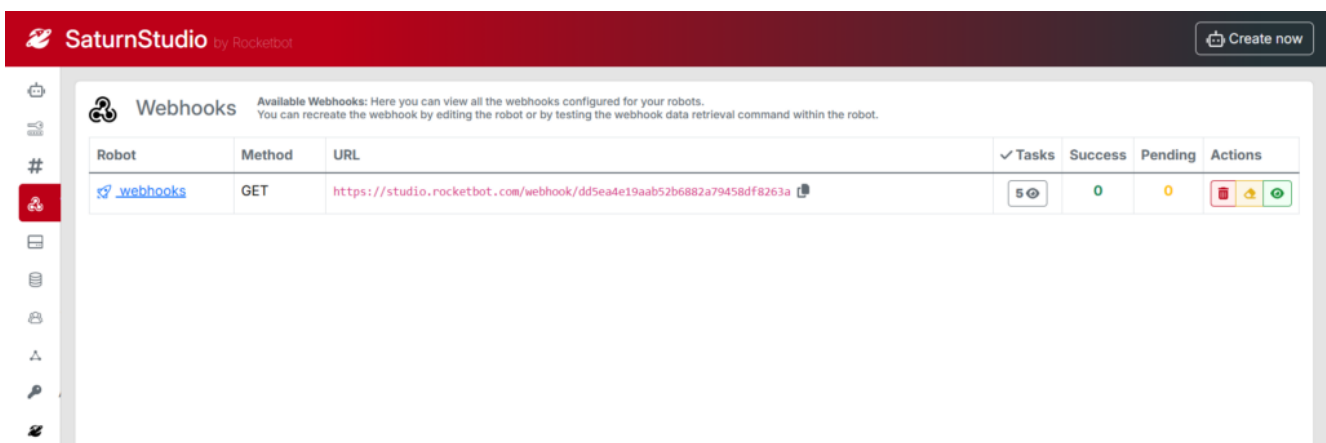
## Interfaz de Gestión (Panel Lateral)

Una vez que has configurado y guardado un robot que contenga un comando de Webhook, la información de control aparecerá automáticamente en la sección **Webhooks** del menú lateral izquierdo.

| Columna       | Descripción   |
|---------------|---|
| <b>Robot</b>  | El nombre del bot que tiene el receptor activo.           |
| <b>Method</b> | El tipo de petición (POST, GET, etc.) que espera recibir. |

| Columna           | Descripción  |
|-------------------|--|
| URL               | La dirección única que debes copiar y pegar en la aplicación externa.                    |
| Tasks             | El contador de cuántas veces se ha ejecutado el flujo por este medio.                    |
| Success / Pending | Muestra cuántas solicitudes terminaron bien y cuántas están en proceso.                  |
| Actions           | Aparecerán las opciones para eliminar el webhook, borrar tareas o ver la cola de tareas. |

## Visualizar cola de tareas / View tasks queue



### Botones de Acción (Actions)

- **Papelera (Rojo):** Elimina permanentemente el Webhook y su URL del servidor.
- **Escoba/Borrador (Amarillo):** Limpia el historial de tareas y pone los contadores a cero.
- **Ojo (Verde):** Permite ver los detalles técnicos de la configuración actual del Webhook.

Al inspeccionar las ejecuciones de un Webhook, se despliega este modal que lista cada petición recibida de forma individual.

- **ID:** Identificador único alfanumérico de la petición recibida.
- **Header / Body / Query / Data:** Indican la cantidad de parámetros o el peso de la información recibida en cada sección de la petición HTTP.
- **Status:** Muestra si la tarea específica fue completada o si presentó algún error.
- **Updated:** Tiempo transcurrido desde que se recibió la petición.
- **Botón de Ver (Ojo Azul):** Abre el desglose detallado de los datos crudos (raw data) de esa tarea específica.

| ID                            | Header | Body | Query | Data | Status | Updated        |  |
|-------------------------------|--------|------|-------|------|--------|----------------|--|
| 84adcf1bf0cdaa039d6ffad5d3... | 20     | 0    | 0     | 0    |        | 35 minutes ago |  |
| c1c6868bdc700625066287f17...  | 18     | 0    | 0     | 0    |        | 35 minutes ago |  |
| c443c9ced3220c900232394...    | 20     | 0    | 0     | 0    |        | 35 minutes ago |  |
| e85689e06c54820539cd3192...   | 20     | 0    | 0     | 0    |        | 35 minutes ago |  |
| 3d223cdb2045e98ad65a7569...   | 20     | 0    | 0     | 0    |        | 35 minutes ago |  |

Page Size: 10 | First | Prev | 1 | Next | Last

## Detalle de Datos de Tarea (View tasks data)

Este panel es fundamental para la depuración (debugging), ya que muestra exactamente qué información envió la aplicación externa a Saturn Studio.

- **Headers:** Contiene los metadatos de la conexión (IP de origen, tipo de navegador, tokens de autenticación, etc.).
- **Body:** El contenido principal del mensaje, generalmente en formato JSON.

- **Query:** Parámetros enviados directamente en la URL (ej: ?id=123).
- **Params:** Parámetros de ruta procesados.
- **Response:** La respuesta que Saturn Studio devolvió al servidor emisor tras recibir el mensaje.

## View tasks data



### Headers

```
{
  "host": "studio.rocketbot.com",
  "x-real-ip": "200.105.110.230",
  "x-forwarded-for": "200.105.110.230",
  "x-forwarded-proto": "https",
  "connection": "close",
  "cache-control": "max-age=0",
  "sec-ch-ua": "\"Not:A-Brand\";v=\"99\", \"Google Chrome\";v=\"145\", \"Chromium\";v=\"145\"",
  "sec-ch-ua-mobile": "?0",
  "sec-ch-ua-platform": "\"Windows\"",
  "upgrade-insecure-requests": "1",
  "user-agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/145.0.0.0 Safari/537.36",
  "accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7",
  "sec-fetch-site": "none",
  "sec-fetch-mode": "navigate",
  "sec-fetch-user": "?1",
  "sec-fetch-dest": "document",
  "accept-encoding": "gzip, deflate, br, zstd",
  "accept-language": "en-US,en;q=0.9,es;q=0.8",
  "cookie": "_gid=GA1.2.519211266.1772549334; _legacy_auth0.Bq16mHIGo7LF2uXbQfLbgJGX282tEe5w.is.authenticated=true; auth0.Bq16mHIGo7LF2uXbQfLbgJGX282tEe5w.is.authenticated=true; _ga=GA1.1.916201240.1771594458; _ga_JF6Z76QC5Q=GS2.1.s1772799177$o27$g1$t1772799575$j27$l0$h0; _ga_RMYQMPF73F=GS2.1.s1772799542$o9$g1$t1772799575$j27$l0$h0",
  "if-none-match": "W/\"15-BVEW84Yxx4bH1R0h00C0oTmUjFw\""
}
```

### Body

```
{}
```

### Query

```
{}
```

### Params

```
{}
```

### Response

## Descripción de los Comandos

Para usar esta tecnología, encontrarás los siguientes módulos dentro de la categoría **Webhooks** del editor.



## 1. Recibir Webhook (Receive Webhook)

- **Webhook URL:** Es la dirección estática que genera el sistema para tu bot.
- **HTTP Method:** El protocolo de comunicación. Puedes elegir entre **GET, POST, PUT o DELETE**.
- **Response mode:** Define cómo se le avisa al emisor que el dato llegó. Puede ser **Immediately** (al instante), **After robot execution** (al terminar el bot) o de forma manual con el comando de respuesta.
- **Assign result to Variable:** El nombre de la variable donde se guardará toda la información recibida (el cuerpo del mensaje).

## 2. Responder Webhook (Response Webhook)

Se utiliza cuando necesitas enviar una respuesta personalizada o específica después de procesar los datos.

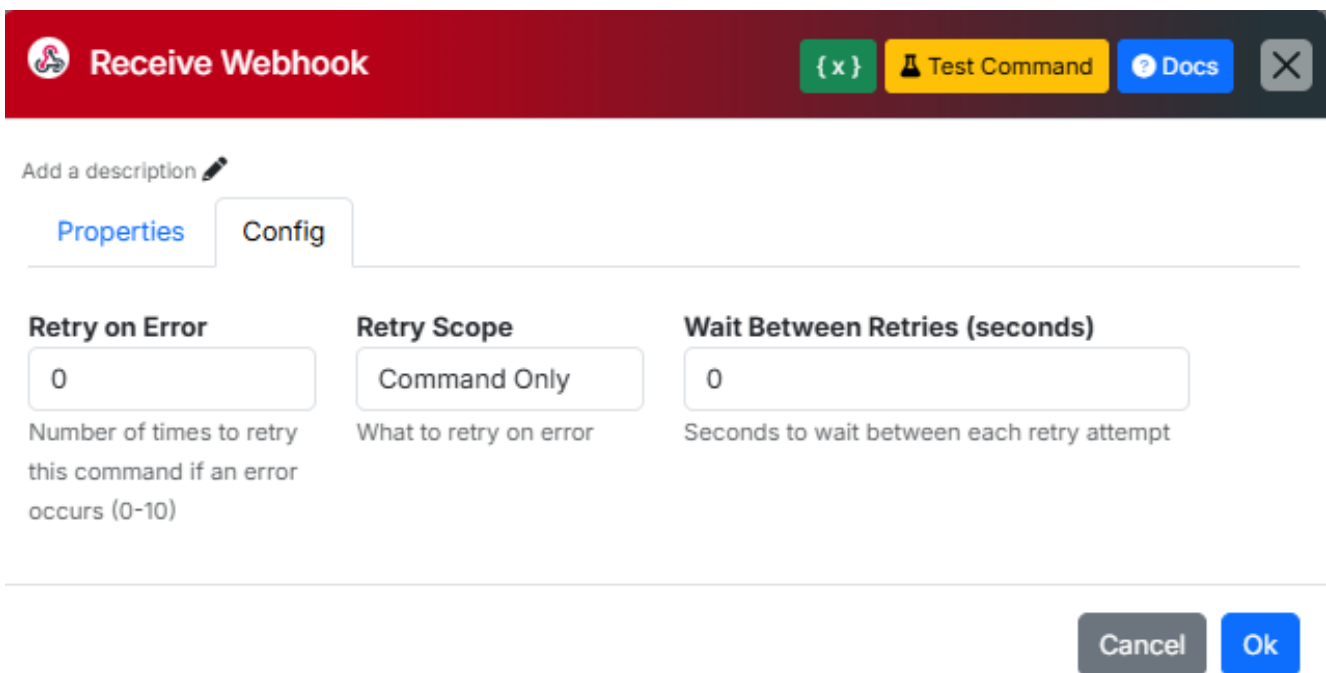
- **Task ID:** El identificador de la tarea a la que estás respondiendo.
- **Response:** El texto o JSON que quieres devolver a la aplicación original.

- **Headers:** Encabezados técnicos opcionales para la respuesta.

## Configuración Avanzada (Pestaña Config)

Dentro de los módulos, puedes ajustar el comportamiento ante fallos:

- **Retry on Error:** Número de intentos (de 0 a 10) si ocurre un fallo al recibir.
- **Wait Between Retries:** Segundos de espera entre cada intento de recuperación.



The screenshot shows the configuration interface for the 'Receive Webhook' module. At the top, there is a red header bar with the module name and icons for '{x}', 'Test Command', 'Docs', and a close button. Below the header, there is a section for 'Add a description' with a pencil icon. Two tabs are visible: 'Properties' and 'Config', with 'Config' being the active tab. The configuration area is divided into three columns:

- Retry on Error:** A text input field containing '0'. Below it, the text reads: 'Number of times to retry this command if an error occurs (0-10)'.
- Retry Scope:** A dropdown menu showing 'Command Only'. Below it, the text reads: 'What to retry on error'.
- Wait Between Retries (seconds):** A text input field containing '0'. Below it, the text reads: 'Seconds to wait between each retry attempt'.

At the bottom right of the configuration area, there are two buttons: 'Cancel' and 'Ok'.

## Cómo funciona todo en conjunto

1. **En el Editor:** Colocas el módulo **Receive Webhook** al principio de tu robot. Al configurarlo, copias la **URL**.
2. **En la App Externa:** Pegas esa URL en la configuración de la aplicación que quieres conectar (por ejemplo, en las

opciones de desarrollador de WhatsApp o Facebook).

3. **El Evento:** Cuando sucede algo afuera, esa app envía un mensaje a la URL.
4. **La Ejecución:** Saturn Studio detecta el mensaje, “despierta” al robot y guarda los datos en la **Variable** que elegiste.
5. **El Monitoreo:** Puedes ir al menú principal, entrar a **Webhooks** y ver cuántas veces ha funcionado y si hubo errores.