Nexus: Edición de Aplicación — App Overview



La sección **App Overview** es el panel de configuración general para una aplicación dentro de Nexus. Desde esta vista se gestionan las pantallas, la interfaz (encabezados, pies de página, temas), la asignación de usuarios, las claves de API, las variables globales, los tokens de Gateway y los registros de actividad.

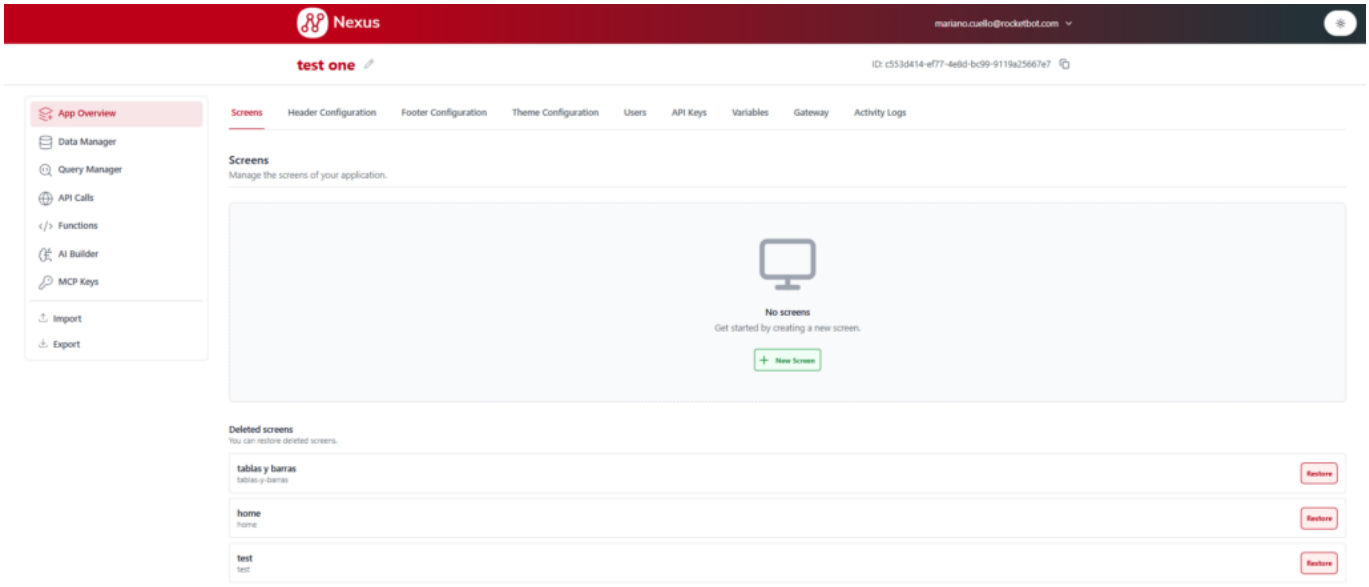
1. Cabecera e Identificador de App

En la parte superior de la interfaz de edición se muestran los datos de identificación de la aplicación:

- **Nombre de la Aplicación:** Se visualiza con un icono de lápiz a su lado para modificar el nombre del proyecto.
- **ID:** Muestra el identificador único de la aplicación junto a un botón para copiarlo al portapapeles.

2. Pantallas (*Screens*)

Esta sección está destinada a la gestión y creación de las vistas de la aplicación.



> 📝 **Nota sobre la edición de Pantallas:** La configuración, diseño y maquetación detallada de las pantallas cuenta con su propia documentación dedicada debido a su complejidad. Puedes acceder a ella a través del siguiente enlace: **[Guía detallada de edición de Screens]**.

Opciones disponibles en el panel:

- **+ New Screen:** Botón para crear una nueva pantalla.
- **Pantallas Eliminadas (*Deleted screens*):** Muestra el listado de pantallas que han sido borradas, permitiendo recuperarlas mediante el botón **Restore**.

3. Configuración del Encabezado (Header Configuration)

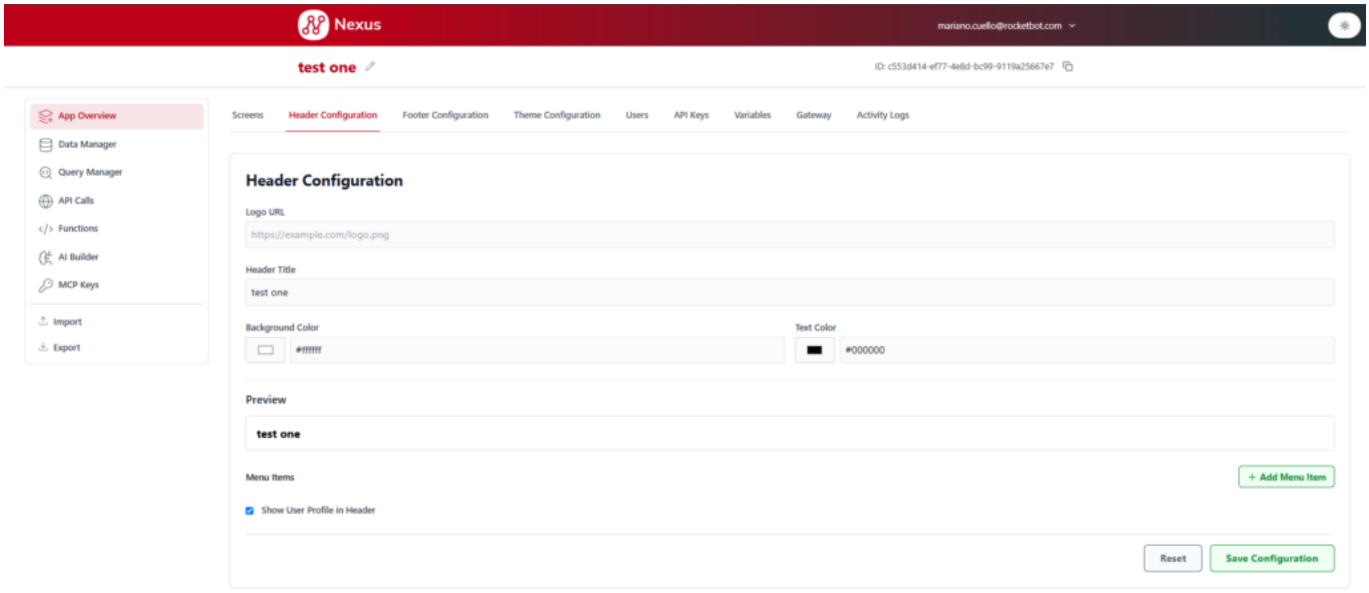
Permite definir los elementos visuales de la barra superior en la aplicación:

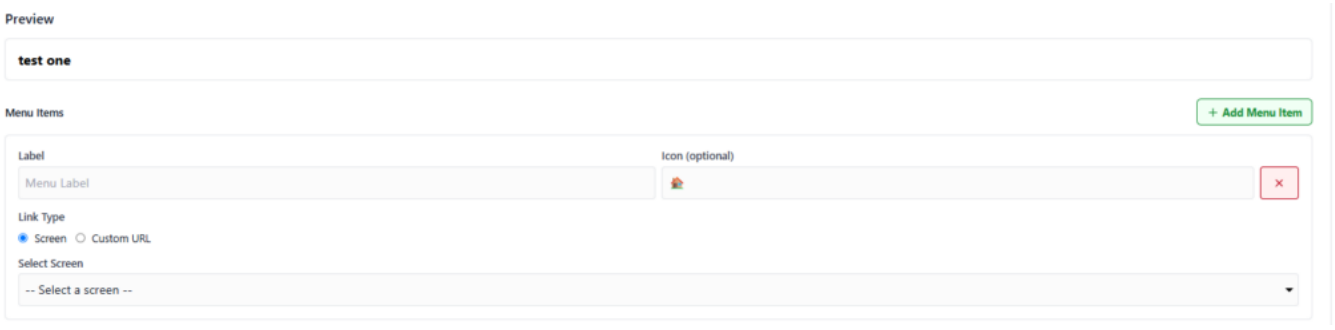
- **Logo URL:** Campo para ingresar el enlace de la imagen del logotipo.
- **Header Title:** Campo para definir el título visible en el encabezado.
- **Background Color / Text Color:** Controles para elegir el color del fondo y del texto mediante código hexadecimal o selector visual.
- **Preview:** Vista previa interactiva de la barra según los colores y el título ingresados.
- **+ Add Menu Item:** Botón para añadir elementos o secciones de menú al encabezado. Al hacer clic en él, se despliega un formulario para configurar el elemento con las siguientes opciones:
  - **Label:** Campo de texto para ingresar el nombre visible del item del menú.
  - **Icon (optional):** Permite asignar un ícono o emoji para acompañar al texto del enlace.

    - **Link Type:** Selección mediante botón de opción para definir el tipo de destino:
        - **Screen:** Redirige a una pantalla interna de la aplicación. Habilita el menú desplegable **Select Screen** (-- Select a screen --) para seleccionar la pantalla deseada.
        - **Custom URL:** Permite ingresar un enlace o dirección web externa.

    - **Botón de eliminación (X rojo):** Permite remover el item de menú creado.

- **Show User Profile in Header:** Casilla de verificación para activar o desactivar la visualización del perfil de usuario en la barra superior.
- **Botones de acción:**
    - **Reset:** Para restaurar los valores anteriores.
    - **Save Configuration:** Para guardar los cambios realizados.

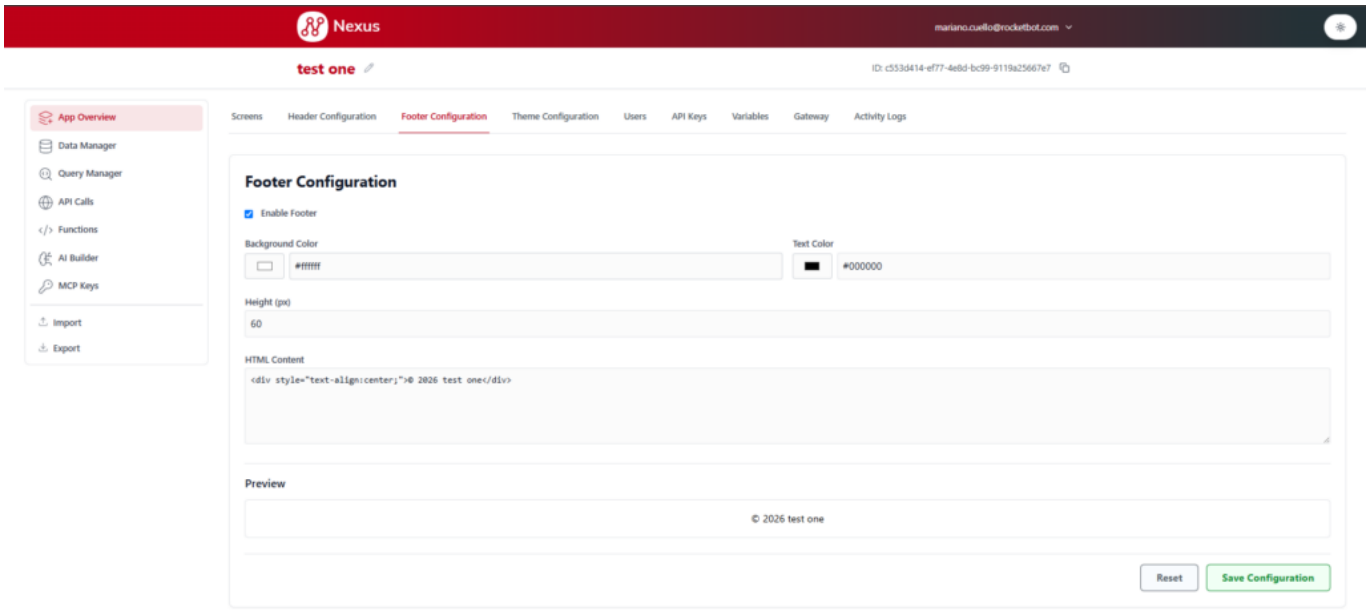




4. Configuración del Pie de Página (*Footer Configuration*)

Administra la barra inferior de la aplicación:

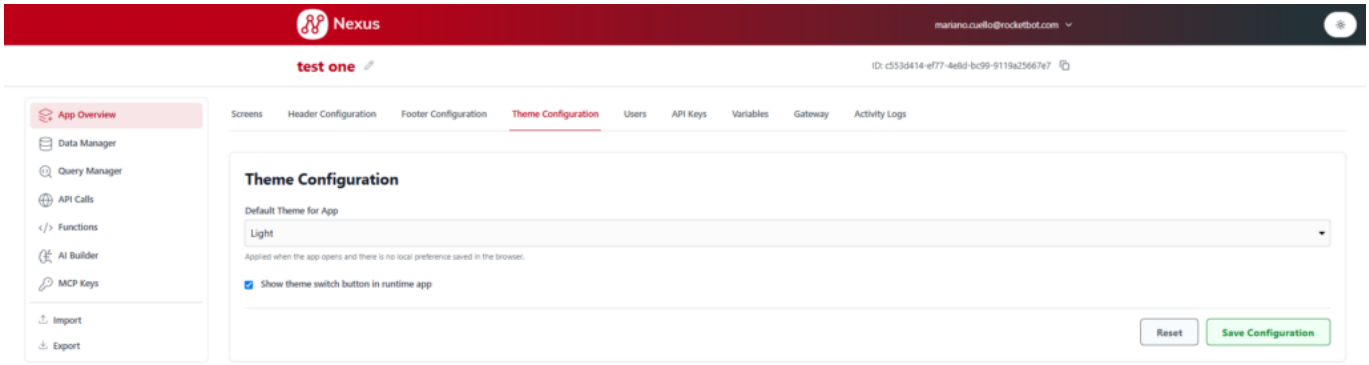
- **Enable Footer:** Casilla de verificación para habilitar o deshabilitar la presencia del pie de página.
- **Background Color / Text Color:** Ajuste de color del fondo y texto del footer.
- **Height (px):** Campo numérico para definir la altura en píxeles del pie de página.
- **HTML Content:** Área de texto que permite ingresar código HTML personalizado (como etiquetas , estilos o texto).
- **Preview:** Muestra en pantalla el resultado final del código HTML y colores seleccionados.
- **Botones de acción: Reset** y **Save Configuration**.



5. Configuración de Tema (*Theme Configuration*)

Establece las preferencias de apariencia visual de la app:

- **Default Theme for App:** Selector desplegable para definir el tema predeterminado (Light) que se aplicará al abrir la app cuando no haya preferencias guardadas en el navegador del usuario.
- **Show theme switch button in runtime app:** Casilla de verificación que, al estar activa, muestra el botón para que los usuarios alternen el tema visual durante la ejecución de la app.
- **Botones de acción: Reset** y **Save Configuration**.



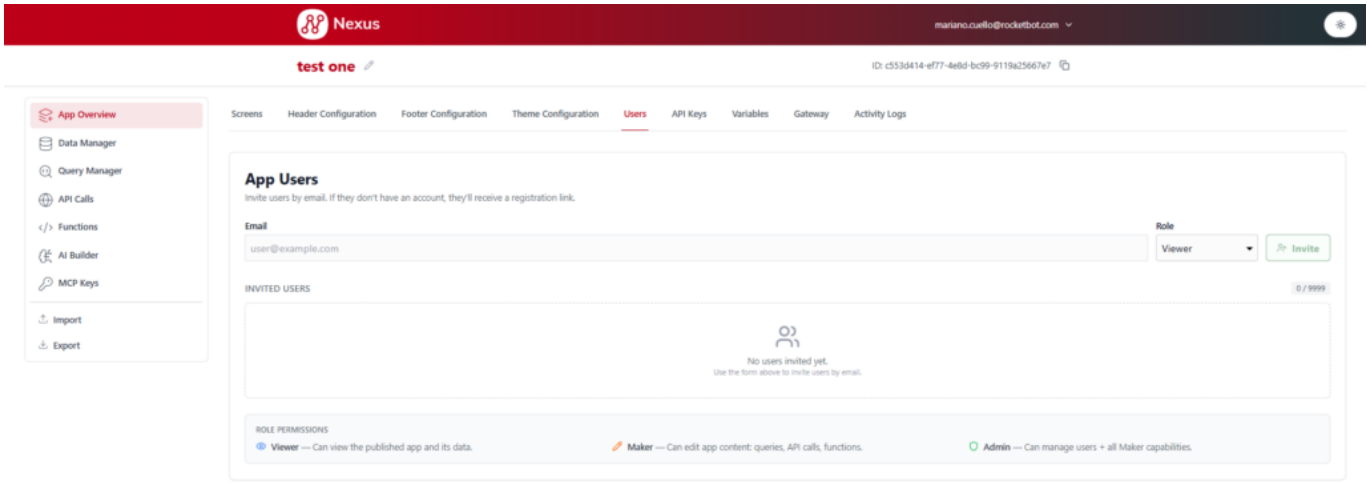
6. Usuarios de la Aplicación (*Users*)

Sección **App Users** destinada a invitar y asignar roles a otros usuarios mediante correo electrónico.

Invitar usuarios:

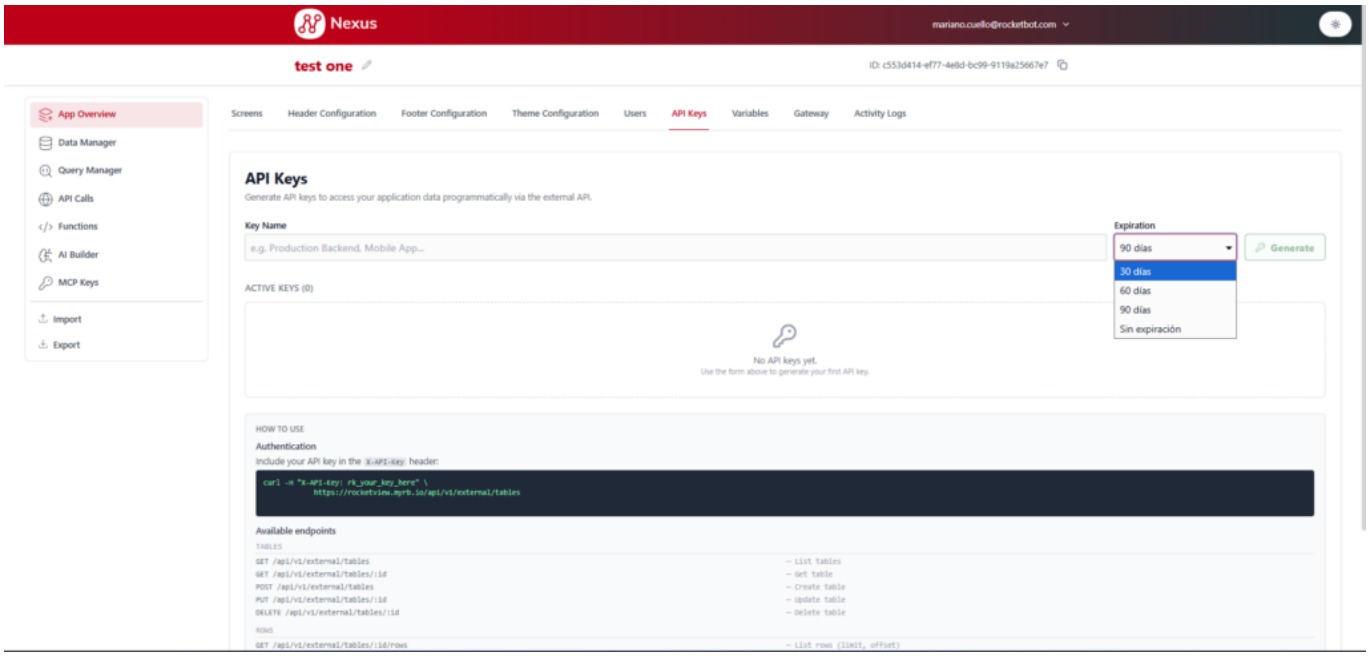
- **Email:** Campo de texto (user@example.com) para colocar la dirección del invitado.
- **Role:** Desplegable con tres niveles de acceso:
  - **Viewer:** Puede ver la aplicación publicada y sus datos.
  - **Maker:** Puede editar el contenido de la app (queries, llamadas a API, funciones).
  - **Admin:** Puede gestionar usuarios además de poseer todas las capacidades de un *Maker*.

- **+ Invite:** Botón para enviar la invitación al usuario.



7. Claves de API (*API Keys*)

Sección para generar llaves de acceso programático a los datos de la aplicación a través de la API externa.



Generación de llaves:

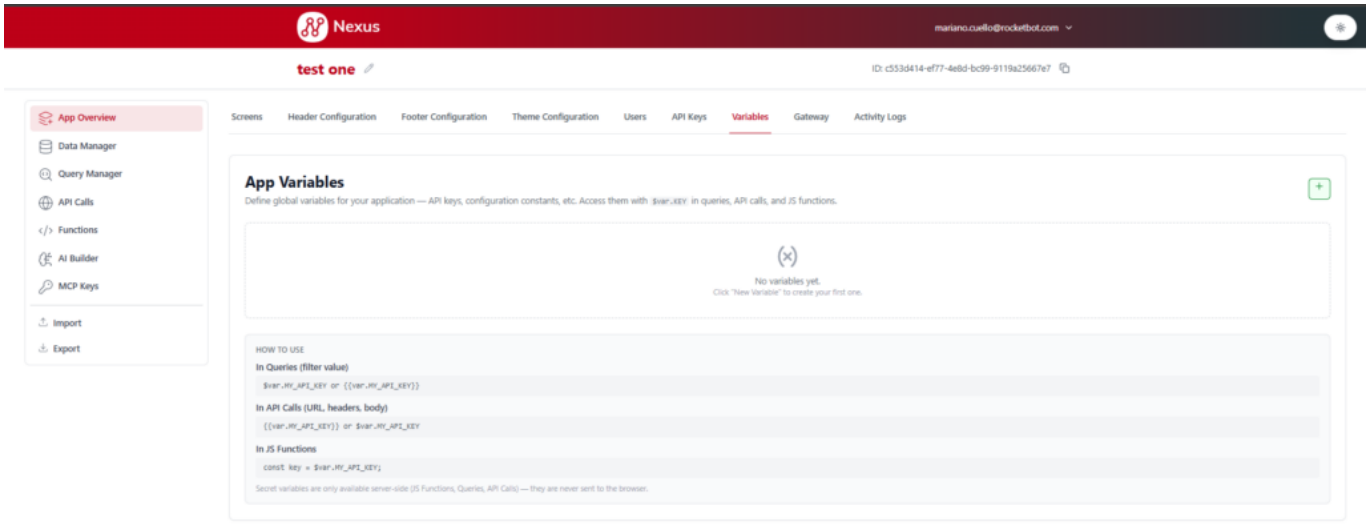
- **Key Name:** Campo de texto para nombrar la clave (ej. *Production Backend*, *Mobile App…*).
- **Expiration:** Selector desplegable con opciones de caducidad: **30 días**, **60 días**, **90 días** o **Sin expiración**.
- **Generate:** Botón para crear la API Key.

Guía de uso en interfaz (*How To Use*):

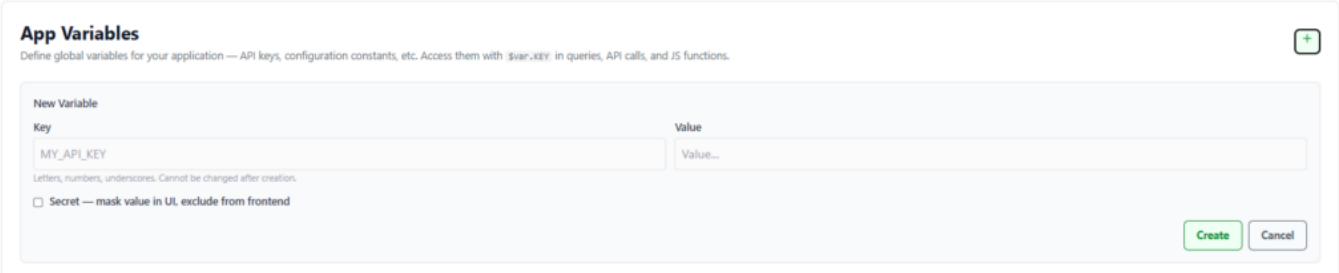
- **Autenticación:** Requiere incluir la clave en el encabezado X-API-Key de la petición HTTP.
- Endpoints disponibles (*Available endpoints*):
  - **Tablas (*TABLES*):**
    - GET /api/v1/external/tables – Listar tablas (*List tables*).
    - GET /api/v1/external/tables/:id – Obtener tabla (*Get table*).
    - POST /api/v1/external/tables – Crear tabla (*Create table*).
    - PUT /api/v1/external/tables/:id – Actualizar tabla (*Update table*).
    - DELETE /api/v1/external/tables/:id – Eliminar tabla (*Delete table*).
  - **Filas (*ROWS*):**
    - GET /api/v1/external/tables/:id/rows – Listar filas con opciones de límite y desplazamiento (*List rows (limit, offset)*).
    - POST /api/v1/external/tables/:id/rows – Insertar fila (*Insert row*).
    - PUT /api/v1/external/rows/:id – Actualizar fila (*Update row*).
    - DELETE /api/v1/external/rows/:id – Eliminar fila (*Delete row*).
    - DELETE /api/v1/external/tables/:id/rows – Borrar todas las filas (*Erase all rows*).
    - PUT /api/v1/external/tables/:id/rows/:rowId/cells/:col – Actualizar celda específica (*Update cell*).
  - **Consultas (*QUERIES*):**
    - GET /api/v1/external/queries – Listar consultas (*List queries*).
    - POST /api/v1/external/queries/:id/execute – Ejecutar consulta (*Execute query*).

8. Variables de la Aplicación (*Variables*)

Permite definir variables globales (*App Variables*) como llaves de API o constantes de configuración para usarlas en queries, llamadas API y funciones JS.



Creación de variables (+ / New Variable):



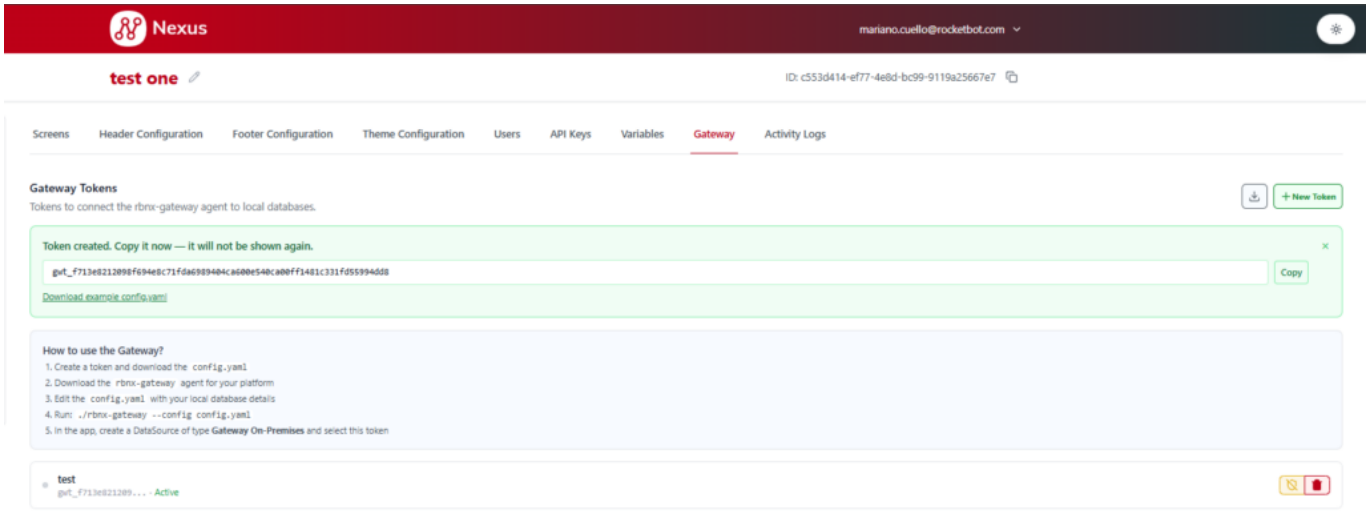
- **Key:** Nombre de la variable (acepta letras, números y guiones bajos; no se puede editar una vez creada).
- **Value:** Valor asignado a la variable.
- **Secret:** Casilla opcional para enmascarar el valor en la interfaz gráfica y excluirlo del frontend (disponible únicamente del lado del servidor).
- **Botones: Create** y **Cancel**.

Uso dentro de la plataforma (*How To Use*):

- **En Queries (filtro de valor):** $var.MY_API_KEY o {{var.MY_API_KEY}}.
- **En API Calls (URL, headers, body):** {{var.MY_API_KEY}} o $var.MY_API_KEY.
- **En JS Functions:** const key = $var.MY_API_KEY;.

9. Pasarela (*Gateway*)

Sección **Gateway Tokens** encargada de administrar los tokens para conectar el agente rbnx-gateway con bases de datos locales.



Creación y gestión de Tokens:

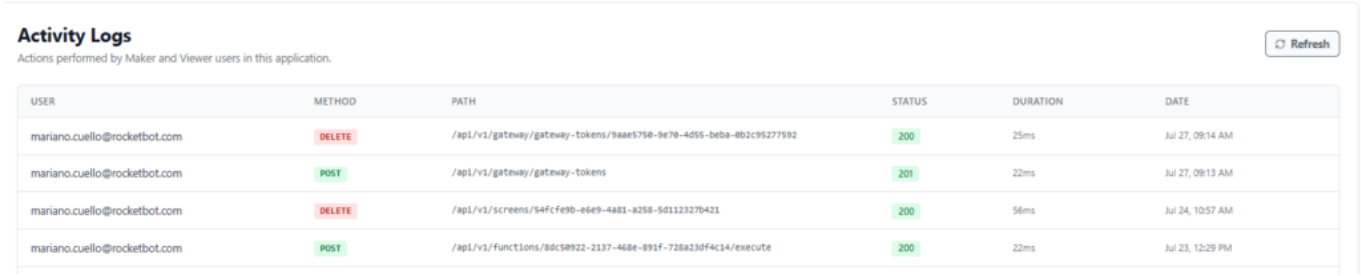
- **+ New Token:** Abre la opción para asignar un **Name** (ej. *Local MySQL Server*) y presionar **Create**.
- **Token generado:** Genera una cadena que debe copiarse en el momento (Copy), ya que no se vuelve a mostrar.
- **Archivos auxiliares:** Incluye enlace de descarga para config.yaml y botón con icono de descarga para el agente ejecutable.
- **Acciones en lista de tokens:** Cada token creado muestra su estado (Active) y cuenta con opciones para deshabilitarlo o eliminarlo (icono de papelera).

Pasos indicados en la interfaz (*How to use the Gateway?*):

1. Crear un token y descargar el archivo config.yaml.
2. Descargar el agente rbnx-gateway para la plataforma correspondiente.
3. Editar config.yaml con los detalles de la base de datos local.
4. Ejecutar el comando ./rbnx-gateway --config config.yaml.
5. En la app, crear un *DataSource* de tipo **Gateway On-Premises** y seleccionar dicho token.

10. Registros de Actividad (*Activity Logs*)

Muestra la lista de acciones realizadas por usuarios con rol *Maker* y *Viewer* en la aplicación.



- **Listado de datos en tabla:**
  - **USER:** Dirección de correo del usuario.
  - **METHOD:** Método de la petición (DELETE, POST, etc.).
  - **PATH:** Ruta del endpoint consultado (ej. /api/v1/gateway/gateway-tokens/...).
  - **STATUS:** Código de respuesta de la petición (ej. 200, 201).
  - **DURATION:** Tiempo de respuesta medido en milisegundos (ej. 25ms).
  - **DATE:** Fecha y hora exacta de la acción.

- **Botón Refresh:** Ubicado en la parte superior derecha para actualizar el historial de eventos.