apibot-Bitrix

1. Introducción del proyecto

Integración del motor conversacional Ánima Bot con Bitrix24, utilizando una arquitectura backend puro en Laravel. Permite recibir mensajes desde Bitrix (Telegram, WhatsApp, etc.) y procesarlos mediante árboles conversacionales definidos en Ánima Bot, en el archivo .env example se encuentra un ejemplo de como se debe configurar el archivo .env para producción, importante llamar las variables como se muestra en el archivo de ejemplo, la url del portal que se usa para diferentes configuraciones es la url donde se creo la cuenta de bitrix, mas adelante se muestra el ejemplo.

2. Requisitos previos

- Laravel 10 o superior
- PHP 8.2+
- Composer
- MySQL/MariaDB
- ngrok para exponer localmente (ojo: únicamente si el proyecto se ejecuta de manera local).
- Cuenta de administrador en Bitrix24.

3. Instalación de dependencias

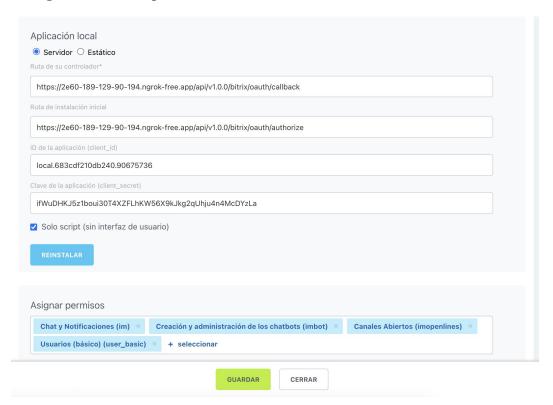
- Clonar el repositorio
- Ejecutar 'composer install'
- Crear y configurar el archivo `.env`
- Ejecutar 'php artisan migrate'

4. Registro de la App en Bitrix

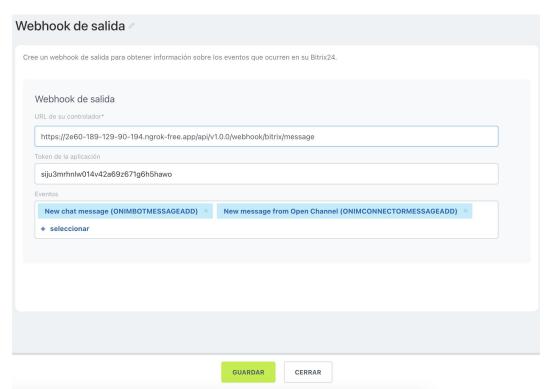
A continuación se muestra la configuración usada para registrar la Aplicación Local y el Webhook de salida, las imágenes que se muestran después de estas instrucciones del paso 4 son tomadas de la configuración con las que se estuvo realizando las pruebas en el local y cuenta de pruebas de bitrix.

Configuración de Aplicación Local Bitrix, en todos los casos donde se ve la url configurada de ngrok se debe poner la url del proyecto nuevo, muy importante configurar los permisos adecuados en la aplicación local y configurar los eventos indicados en el Webhook de salida, para que la aplicación funcione correctamente.

Configuración de la aplicación local



Configuración del Webhook de salida



5. Insertar credenciales desde Laravel ejecutando endpoint.

POST /api/v1.0.0/bitrix/instance/setup Campos: portal, client_id, client_secret, auth_token

```
Bearer Token: <auth_token>
Body

{
    "portal": "b24-4bkru2.bitrix24.mx",
    "client_id": "local.68535e07c46c85.62081250",
    "client_secret": "pMEptEHsONHkLXOW4liAPWqw7j02yKhHAHjWUrPzdkW23W2Blo",
    "auth_token": "g3q77o79cslj9tczp4z6gryfmied462e"
}
```

6. Insertar hash del chat para la instacia desde Laravel ejecutando endpoint.

```
POST /api/v1.0.0/bitrix/update-hash
Campos: portal, hash
Bearer Token: <auth_token>
Body
{
    "portal": "b24-4bkru2.bitrix24.mx",
    "hash": "eyJpdil6Im....."
}
```

7. Endpoint de instalación OAuth (se ejecutan cuando se instala o repíntala la aplicación local de Bitrix)

- Ruta de instalación inicial: /api/v1.0.0/bitrix/oauth/authorize
- Ruta del controlador: /api/v1.0.0/bitrix/oauth/callback

8. Flujo conversacional

Bitrix recibe mensaje \rightarrow Webhook hacia Laravel \rightarrow Laravel detecta portal y sesión \rightarrow Llama al árbol de Ánima \rightarrow Procesa respuesta \rightarrow Responde a Bitrix (Telegram, WhatsApp, etc.)

9. Seguridad del Webhook

El middleware 'verify.bitrix' protege el webhook validando:

- Bearer Token (auth_token) enviado en la cabecera.
- application_token en el payload, validado contra la BD.