



Manual de Instalación y Uso: Agente Comercial Kavak IA

Introducción

Este manual te guiará a través de los pasos para instalar y ejecutar localmente el Agente Comercial Kavak IA. Este bot es una aplicación Django contenerizada con Docker, diseñada para simular las interacciones de un agente comercial de Kavak utilizando la API de OpenAI.

Requisitos Previos

Antes de comenzar, asegúrate de tener instalado lo siguiente en tu sistema:

1. **Docker:** Necesitarás Docker para construir y ejecutar la imagen del contenedor de la aplicación.
 - Puedes descargarlo e instalarlo desde la página oficial de Docker: [Get Docker](#).
 - Asegúrate de que el servicio de Docker esté en ejecución.
2. **Clave API de OpenAI:** El bot utiliza la API de OpenAI para su lógica de inteligencia artificial. Necesitarás una clave API válida.
 - Puedes obtenerla desde tu cuenta en platform.openai.com/account/api-keys.

Instalación y Configuración

Sigue estos pasos para poner en marcha el bot:

Paso 1: Obtener el Código Fuente

- Si tienes el código en un archivo ZIP, descárgalo y descomprímelo en una carpeta de tu elección.
- Navega hasta el directorio raíz del proyecto (la carpeta que contiene el archivo `Dockerfile`, por ejemplo, `commercial_agent_bot/`).

Paso 2: Configurar la Clave API de OpenAI

El bot carga la clave API de OpenAI desde un archivo de entorno llamado `.env`.

1. En el **directorio raíz de tu proyecto** (donde se encuentra el `Dockerfile`), crea un archivo llamado `.env`.
2. Abre el archivo `.env` con un editor de texto y añade la siguiente línea, reemplazando `TU_OPENAI_API_KEY_AQUI` con tu clave API real.

```
OPENAI_API_KEY=TU_OPENAI_API_KEY_AQUI
```

Paso 3: Construir la Imagen Docker

Abre una terminal o línea de comandos, asegúrate de estar en el directorio raíz del proyecto (el que contiene el `Dockerfile`), y ejecuta el siguiente comando para construir la imagen Docker:

Bash

```
docker build -t kavak_commercial_agent .
```

- `docker build`: El comando para construir imágenes.
- `t kavak_commercial_agent`: Asigna un nombre (tag) a la imagen, en este caso `kavak_commercial_agent`. Puedes elegir otro nombre.
- `.`: Indica que el contexto de construcción (incluyendo el `Dockerfile`) se encuentra en el directorio actual.

Este proceso puede tardar unos minutos la primera vez, ya que Docker descargará las imágenes base y las dependencias necesarias.

Paso 4: Ejecutar el Contenedor Docker

Una vez que la imagen se haya construido correctamente, puedes ejecutar un contenedor a partir de ella con el siguiente comando:

Bash

```
docker run -d -p 8000:8000 --env-file .env --name kavak_bot_container kavak_commercial_agent
```

- `docker run` : El comando para ejecutar contenedores.
- `d` : Ejecuta el contenedor en modo "detached" (en segundo plano).
- `p 8000:8000` : Mapea el puerto 8000 de tu máquina host al puerto 8000 del contenedor (donde Unicorn está escuchando dentro del contenedor).
- `-env-file .env` : Carga las variables de entorno definidas en tu archivo `.env` dentro del contenedor.
- `-name kavak_bot_container` : Asigna un nombre al contenedor en ejecución para facilitar su gestión (opcional, pero útil).
- `kavak_commercial_agent` : El nombre de la imagen que construiste en el paso anterior.

Si todo ha ido bien, el bot debería estar ejecutándose dentro del contenedor Docker y accesible en el puerto 8000 de tu máquina local.

Puedes verificar si el contenedor está en ejecución con:

Bash

```
docker ps
```

Deberías ver `kavak_bot_container` en la lista.

Uso del Bot

Para interactuar con el bot una vez que el contenedor está en ejecución, puedes enviar solicitudes HTTP POST al endpoint de la API. El endpoint principal, según la estructura de tu proyecto (`interaction_api/urls.py`), es `/api/v1/interaction/` .

Enviar un Mensaje (Ejemplo con `curl`)

Abre una nueva terminal o línea de comandos y utiliza `curl` para enviar un mensaje al bot:

Bash

```
curl -X POST -H "Content-Type: application/json" \
-d '{"customer_message": "Hola, estoy buscando información sobre autos seminuevos de menos de 300,000 pesos."' \
http://localhost:8000/api/v1/interaction/
```

Explicación de la Petición `curl` :

- `curl` : Herramienta de línea de comandos para transferir datos con URLs.
- `X POST` : Especifica que la solicitud es de tipo POST.
- `H "Content-Type: application/json"` : Indica que el cuerpo de la solicitud está en formato JSON.
- `d '{"customer_message": "Hola, estoy buscando información sobre autos seminuevos."}'` : Es el cuerpo de la solicitud en formato JSON. El bot espera un campo `customer_message` con el mensaje del usuario.
- `http://localhost:8000/api/v1/interaction/` : Es la URL del endpoint del bot.

Respuesta Esperada

El bot procesará el mensaje y deberías recibir una respuesta en formato JSON, similar a esto (el contenido exacto de `bot_reply` variará según la lógica del bot y la respuesta de OpenAI):

JSON

```
{
  "tracking_id": "una-cadena-uuid-aleatoria",
  "bot_reply": "¡Hola! Claro, en Kavak tenemos una amplia variedad de autos seminuevos de menos de 300,000 pesos, acá hay unos que podrían gustarte..."
}
```

Detener el Contenedor

Cuando hayas terminado de usar el bot, puedes detener y eliminar el contenedor para liberar recursos:

1. Detener el contenedor: Bash

```
docker stop kavak_bot_container
```

2. Eliminar el contenedor (opcional, si ya no lo necesitas): Bash

```
docker rm kavak_bot_container
```

Uso del agente vía whatsapp

Usando Twilio Sandbox se puede interactuar con el bot.

Enviando el mensaje **stick-slowly** vía whatsapp al número +14155238886 habilitará el uso del Bot para que se pueda probar directo desde whatsapp.



Tomar en cuenta que por el ambiente operativo del bot puede tardar más en responder ya que el servidor es puesto en suspensión mientras no se usa.