# 📦 Guía de instalación: ChatBot para Mattermost con Errbot

## ****✅ 1. Requisitos previos****

Asegúrate de tener:

* Un servidor de Mattermost activo con APIv4 habilitada.
* Cuenta de bot creada en Mattermost (con privilegios de administrador si es necesario).
* Python 3.4 o superior
* Instaladas las siguientes librerías de Python:
  + websockets==3.2
  + mattermostdriver>=4.0 (se recomienda la versión 7.3.1 para evitar errores con WebSockets).

## ✅ ****2. Clonar el backend de Errbot para Mattermost****

***git clone https://github.com/errbotio/err-backend-mattermost***

***cd err-backend-mattermost***

## ****✅ 3. Instalar dependencias****

Instálalas en un entorno virtual:

***python -m venv venv***

***source venv/bin/activate # Linux/macOS***

***venv\Scripts\activate # Windows***

***# Acceder a la carpeta donde este requirements.txt***

***pip install -r requirements.txt***

***pip install websockets==3.2***

***pip install mattermostdriver>=4.0***

## ****✅ 4. Crear cuenta de bot en Mattermost****

* Ve al servidor de Mattermost como administrador.
* Ingresas a las integraciones y en bot account, configure los campose necesarios y obten el token de accesso.

## ✅ ****5. Crear webhook entrante para adjuntos****

* En Mattermost: **Integrations > Incoming Webhooks**
* Crear uno nuevo, anota la **URL del webhook** (la necesitarás para enviar adjuntos).

## ✅ 6****. Iniciar Errbot****

**En la terminal donde esta el archivo de requirements.txt iniciar el bot con el comando**

*****errbot --init*****

## ✅ 7****. Configurar**** config.py ****de Errbot****

***import logging***

***import os***

***BACKEND = 'Mattermost'***

***BOT\_ADMINS = ('@tu\_usuario\_mattermost',)***

***local\_dir\_path = os.path.dirname(\_\_file\_\_)***

***BOT\_DATA\_DIR = os.path.join(local\_dir\_path, "data")***

***BOT\_EXTRA\_PLUGIN\_DIR = os.path.join(local\_dir\_path, "plugins")***

***BOT\_EXTRA\_BACKEND\_DIR = os.path.join(local\_dir\_path, "..")***

***BOT\_LOG\_LEVEL = logging.DEBUG***

***BOT\_IDENTITY = {***

***'team': 'main',***

***'server': 'tu-servidor.mattermost.com',***

***'token': 'tu-token-del-bot',***

***'scheme': 'https',***

***'port': 443,***

***'timeout': 30,***

***'verify': True, # Usa False solo en entornos de prueba con certificados autofirmados***

***# Si usas adjuntos:***

***# 'webhook\_id': 'abc123xyz'***

***}***

***CHATROOM\_PRESENCE = ()***

***BOT\_PREFIX = "!"***

***BOT\_PREFIX\_OPTIONAL\_ON\_CHAT = True***

***BOT\_ALT\_PREFIXES = ("@",)***

***BOT\_ALT\_PREFIX\_SEPARATORS = (":", ",")***

***DIVERT\_TO\_PRIVATE = ()***

***BOT\_ASYNC = False***

***AUTOINSTALL\_DEPS = False***

🔹 ¿Cuándo necesitas el webhook?

Solo necesitas el webhook si vas a usar métodos como send\_card() o send\_attachment() para enviar mensajes enriquecidos.

Si solo usas send() (texto plano), no necesitas configurar el webhook.

🔹 ¿Qué es un "attachment" en Mattermost?

Un **attachment** (llamado también “card”) es una sección visual enriquecida con campos como:

* título
* texto
* enlaces
* botones
* colores

Ejemplo:  
Un mensaje con una tarjeta de GitHub que muestra el estado de un issue, con color y botones.

**✅ Instrucciones para usar este archivo**

1. Reemplaza en BOT\_IDENTITY:
   * 'server': por el dominio de tu Mattermost (cloud.mattermost.com, por ejemplo).
   * 'token': por el token que te dio Mattermost al crear el bot.
   * 'team': si tu equipo fue el creado automáticamente sera main, de otra forma coloca el nombre de equipo.
2. Reemplaza en BOT\_ADMINS tu usuario de Mattermost, con @ incluido.
3. Ejecuta el bot con: errbot

# 🛠️ Solución al error: Cannot create a client socket with a PROTOCOL\_TLS\_SERVER context

Este error ocurre por una configuración incorrecta de SSL/WebSocket. Aquí se explica cómo solucionarlo:

## ✅ PASO 1: Asegúrate de usar la versión correcta del mattermostdriver

Debes tener una versión **>4.0**, pero **las más nuevas (como 7.x)** pueden haber cambiado cómo manejan los WebSockets. Se recomienda fijar una versión estable como:

***pip install "mattermostdriver==7.3.1"***

## ✅ PASO 2: Corrige la configuración SSL en el config.py

A veces Errbot o el backend de Mattermost usa una configuración de WebSockets insegura por defecto. Agrega explícitamente esta opción en BOT\_IDENTITY:

***BOT\_IDENTITY = {***

***# Corrige problemas de SSL en WebSocket***

***'verify': True, # o False si estás en localhost con certificados autofirmados***

***}***

🔒 Usa 'verify': False solo para desarrollo local o entornos con certificados autofirmados (self-signed).

## ✅ PASO 3: Corrige archivo err-backend-mattermost

### a. Edita el archivo principal (backend.py o donde esté serve\_once)

Despues de los imports de defecto que tiene colocar las siguientes lineas

***import ssl***

***print("Default SSL context protocol:", ssl.create\_default\_context().protocol)***

***context = ssl.create\_default\_context()***

**Reemplaza dentro de** def serve\_once(self):**:**

***self.driver = Driver(***

***{***

***"scheme": self.\_scheme,***

***"url": self.url,***

***"port": self.\_port,***

***"verify": not self.insecure,***

***"timeout": self.timeout,***

***"login\_id": self.\_login,***

***"password": self.\_password,***

***"token": self.\_personal\_access\_token,***

***"mfa\_token": self.\_mfa\_token,***

***"websocket\_kw\_args": {***

***"ssl": context # 👈 Esto es clave***

***}***

***}***

***)***

### ****b. Edita**** websocket.py

**Busca el archivo con el nombre de websocket.py y la line context = ssl.create\_default\_context**

**donde se cambiara de**

*****context = ssl.create\_default\_context(purpose=ssl.Purpose.CLIENT\_AUTH)*****

**a**

*****context = ssl.create\_default\_context(purpose=ssl.Purpose.SERVER\_AUTH)*****

**Reemplaza el bloque de** kw\_args = {} **por:**

*****kw\_args = {}*****

*****if self.options['websocket\_kw\_args'] is not None:*****

*****kw\_args = self.options['websocket\_kw\_args'].copy() # copia para no modificar original*****

*****# Evitar pasar ssl dos veces*****

*****if 'ssl' in kw\_args:*****

*****# O eliminar 'ssl' si ya tienes context configurado*****

*****if context is not None:*****

*****del kw\_args['ssl']*****

*****websocket = await websockets.connect(*****

*****url,*****

*****ssl=context,*****

*****\*\*kw\_args,*****

*****)*****

# ****✅ Probar el bot****

**Una vez configurado todo correctamente:**

*****errbot*****

**Puedes usar comandos como en el chat de mattermost donde este invitado el bot:**

*****!tryme*****

**Para crear nuevos comandos o plugins, usa como referencia los archivos ***example.py*** y ***example.plug***.**

## ****📌 Notas finales****

**Usa verify: False solo en entornos de desarrollo con certificados autofirmados.**

**Mantén tus tokens y credenciales seguros.**

**Usa plugins de forma modular y versionada para mantenimiento más fácil.**