

Documentación del proceso de instalación y configuración del módulo Supabase en Odoo 17

Índice

1. Introducción
2. Preparación del entorno
3. Estructura del módulo y ubicación
4. Configuración del archivo `odoo.conf`
5. Errores encontrados y soluciones aplicadas
6. Activación y verificación del módulo
7. Conclusiones y recomendaciones

1. Introducción

En este documento se recopila todo el proceso realizado para instalar, depurar y activar un módulo personalizado de Odoo 17, llamado **supabase_conector**, desarrollado a partir de las tablas proporcionadas por el usuario.

El objetivo era crear un módulo funcional que conectara Odoo con Supabase y se integrara correctamente dentro de la instancia de Odoo.

2. Preparación del entorno

Se comenzó trabajando en un entorno Linux con Odoo instalado bajo el directorio:

```
/opt/odoo/odoo/
```

Además, se comprobó la existencia de un entorno virtual (venv) donde se ejecutaba Odoo:

```
/opt/odoo/odoo/venv/bin/python3 /opt/odoo/odoo/odoo-bin -c  
/etc/odoo/odoo.conf
```

Se verificó que el servicio estuviera activo con:

```
sudo systemctl status odoo
```

Resultado:

Active: active (running)

3. Estructura del módulo y ubicación

El módulo supabase_conector se subió manualmente al servidor, dentro del directorio:

```
/odoo/custom/addons/supabase_conector
```

Al listar los contenidos se confirmó su presencia:

```
ls /odoo/custom/addons/
```

4. Configuración del archivo odoo.conf

Inicialmente, el archivo /etc/odoo/odoo.conf tenía dos líneas que definían rutas de addons:

```
addons_path = /odoo/odoo-server/addons,/odoo/custom/addons  
addons_path = /opt/odoo/odoo/addons
```

Esto generó un conflicto, ya que Odoo solo toma la **última definición**.

Por tanto, se modificó para dejar una sola línea correcta con todas las rutas necesarias:

```
addons_path = /opt/odoo/odoo/addons,/odoo/custom/addons
```

Luego se reinició el servicio para aplicar los cambios:

```
sudo systemctl restart odoo
```

5. Errores encontrados y soluciones aplicadas

Error 1:

“FileNotFoundError: No such file or directory: '/odoo/odoo-server/addons'”

Causa:

La ruta /odoo/odoo-server/addons no existía realmente en el sistema.

Solución:

Actualizar el addons_path en el archivo de configuración (ver sección anterior).

Error 2:

Permiso denegado al listar /opt/odoo/

Comando:

```
ls /opt/odoo/
```

Error:

```
Permission denied
```

Solución:

Ejecutar con permisos de superusuario:

```
sudo ls -l /opt/odoo/
```

Resultado correcto:

```
drwxrwxr-x 10 odoo odoo 4096 Oct  7 14:17 odoo
```

Error 3:

El módulo no aparecía en la lista de aplicaciones.

Causa:

Odoo no estaba escaneando la carpeta /odoo/custom/addons correctamente.

Solución:

Una vez corregido el addons_path y reiniciado el servicio, se actualizó la lista de módulos desde la interfaz web:

- Activar **Modo desarrollador**
- Ir a **Aplicaciones → Actualizar lista de aplicaciones**
- Buscar **supabase_conector**

El módulo apareció correctamente y se instaló sin errores.

Error 4:

Mensajes GET /websocket?version=17.0-1 HTTP/1.0" 400

Causa:

Esto es un mensaje informativo del sistema de notificaciones de Odoo (bus/websocket), no un error del módulo.

No afecta la funcionalidad.

6. Activación y verificación del módulo

Una vez visible en el listado de aplicaciones, se procedió a instalarlo desde la interfaz de Odoo.

En los logs se pudo observar que el módulo fue cargado correctamente:

```
INFO ventas odoo.addons.supabase_conector: module loaded
```

Desde el panel de **Aplicaciones** se confirmó que el módulo estaba **instalado y activo**.

7. Conclusiones y recomendaciones

- El problema principal radicaba en la configuración duplicada del `addons_path`.
- Una vez corregido y con los permisos adecuados, Odoo reconoció y activó el módulo correctamente.
- Es recomendable mantener una estructura de rutas clara y coherente, y evitar rutas antiguas o no existentes.
- Para futuros módulos, basta con:
 - Copiar el módulo a `/odoo/custom/addons/`
 - Añadir esa ruta a `addons_path` (si no está ya)
 - Reiniciar Odoo y actualizar la lista de aplicaciones.

Comando para generar el PDF localmente

Si quieres guardar este documento en PDF, guarda el texto como `documentacion.md` y ejecuta:

```
pip install py pandoc
py pandoc.convert_text(open('documentacion.md').read(), 'pdf',
format='md', outputfile='documentacion.pdf',
```