

<https://chatgpt.com/c/6986bd77-6244-8326-b14f-86b49f19ad27>

Vous : bjr

Bjr 🙌

Tu veux qu'on parle de quoi ? (Tu peux m'écrire en français ou en "mode texto".)

Pouvez vous réaliser un système de chat en ligne en utilisant une API django côté backend. Pour le côté frontend vous pouvez utiliser JAVASCRIPT et HTML et CSS. Mais pour le backend, utilisez seulement une API django. Django==5.2.8

Oui. Voici un **mini-système de chat en ligne** avec :

- **Backend : Django 5.2.8 + Django REST Framework**
- **Temps réel : WebSocket via Django Channels** (c'est la manière "propre" de faire du chat en ligne)
- **API Django** pour l'historique / listes de salons / messages
- **Frontend : HTML + CSS + JavaScript** (WebSocket + fetch API)

Remarque : pour un "chat en ligne" (temps réel), les **WebSockets sont indispensables**. L'API REST seule implique du polling (moins bon).

1) Backend (Django API + WebSocket)

A. Installation

```
python -m venv venv
source venv/bin/activate # (Windows: venv\Scripts\activate)
pip install "Django==5.2.8" djangorestframework channels channels_redis
django-admin startproject config .
python manage.py startapp chat
```

B. config/settings.py

Ajoute les apps + Channels :

```
INSTALLED_APPS = [
```

```

# ...
"rest_framework",
"channels",
"chat",
]

ASGI_APPLICATION = "config.asgi.application"

# Pour dev simple (sans Redis), tu peux utiliser l'InMemory layer.
# Pour prod, préfère Redis (channels_redis).
CHANNEL_LAYERS = {
    "default": {
        "BACKEND": "channels.layers.InMemoryChannelLayer",
    }
}

```

Si tu veux Redis :

```

CHANNEL_LAYERS = {
    "default": {
        "BACKEND": "channels_redis.core.RedisChannelLayer",
        "CONFIG": {"hosts": [("127.0.0.1", 6379)]},
    }
}

```

C. Modèles `chat/models.py`

```

from django.db import models

class Room(models.Model):
    name = models.CharField(max_length=100, unique=True)

    def __str__(self):
        return self.name

class Message(models.Model):
    room = models.ForeignKey(Room, on_delete=models.CASCADE, related_name="messages")
    author = models.CharField(max_length=80)
    content = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)

    class Meta:
        ordering = ["created_at"]

```

Migration :

```
python manage.py makemigrations
python manage.py migrate
```

D. Sérialiseurs `chat/serializers.py`

```
from rest_framework import serializers
from .models import Room, Message

class RoomSerializer(serializers.ModelSerializer):
    class Meta:
        model = Room
        fields = ["id", "name"]

class MessageSerializer(serializers.ModelSerializer):
    class Meta:
        model = Message
        fields = ["id", "room", "author", "content", "created_at"]
        read_only_fields = ["id", "created_at"]
```

E. API Views `chat/views.py`

```
from rest_framework import generics
from .models import Room, Message
from .serializers import RoomSerializer, MessageSerializer

class RoomListCreateView(generics.ListCreateAPIView):
    queryset = Room.objects.all()
    serializer_class = RoomSerializer

class MessageListView(generics.ListAPIView):
    serializer_class = MessageSerializer

    def get_queryset(self):
        room_id = self.kwargs["room_id"]
        return Message.objects.filter(room_id=room_id).order_by("created_at")[:200]
```

F. URLs API `chat/urls.py`

```
from django.urls import path
from .views import RoomListCreateView, MessageListView
```

```
urlpatterns = [
    path("rooms/", RoomListCreateView.as_view(), name="rooms"),
    path("rooms/<int:room_id>/messages/", MessageListView.as_view(), name="room-messages"),
]
```

Inclure dans config/urls.py :

```
from django.contrib import admin
from django.urls import path, include
```

```
urlpatterns = [
    path("admin/", admin.site.urls),
    path("api/", include("chat.urls")),
]
```

2) WebSocket (Channels)

A. Consumer chat/consumers.py

```
import json
from channels.generic.websocket import AsyncWebsocketConsumer
from channels.db import database_sync_to_async
from .models import Room, Message

class ChatConsumer(AsyncWebsocketConsumer):
    async def connect(self):
        self.room_name = self.scope["url_route"]["kwargs"]["room_name"]
        self.group_name = f"chat_{self.room_name}"

        await self.channel_layer.group_add(self.group_name, self.channel_name)
        await self.accept()

        # Assure que la room existe
        await self.get_or_create_room(self.room_name)

    async def disconnect(self, close_code):
        await self.channel_layer.group_discard(self.group_name, self.channel_name)

    async def receive(self, text_data):
        data = json.loads(text_data)
        author = data.get("author", "anonymous")
        content = data.get("content", "").strip()
```

```

    if not content:
        return

    msg = await self.create_message(self.room_name, author, content)

    await self.channel_layer.group_send(
        self.group_name,
        {
            "type": "chat_message",
            "id": msg["id"],
            "author": msg["author"],
            "content": msg["content"],
            "created_at": msg["created_at"],
        }
    )

    async def chat_message(self, event):
        await self.send(text_data=json.dumps(event))

    @database_sync_to_async
    def get_or_create_room(self, name):
        return Room.objects.get_or_create(name=name)

    @database_sync_to_async
    def create_message(self, room_name, author, content):
        room = Room.objects.get(name=room_name)
        m = Message.objects.create(room=room, author=author, content=content)
        return {
            "id": m.id,
            "author": m.author,
            "content": m.content,
            "created_at": m.created_at.isoformat(),
        }

```

B. Routing WS `chat/routing.py`

```

from django.urls import re_path
from .consumers import ChatConsumer

websocket_urlpatterns = [
    re_path(r"ws/chat/(?P<room_name>[^/]+)/$", ChatConsumer.as_asgi()),
]

```

C. config/asgi.py

Remplace par une config Channels :

```
import os
from django.core.asgi import get_asgi_application
from channels.routing import ProtocolTypeRouter, URLRouter
from channels.auth import AuthMiddlewareStack
import chat.routing

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "config.settings")

django_asgi_app = get_asgi_application()

application = ProtocolTypeRouter({
    "http": django_asgi_app,
    "websocket": AuthMiddlewareStack(
        URLRouter(chat.routing.websocket_urlpatterns)
    ),
})
```

Lancer :

```
python manage.py runserver
```

3) Frontend (HTML + CSS + JS)

Crée un fichier `index.html` (tu peux le servir via n'importe quel serveur statique, ou même ouvrir en local pour tester si CORS OK).

index.html

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Chat Django</title>
  <style>
    body { font-family: Arial, sans-serif; background:#f5f5f5; margin:0; }
    .wrap { max-width:900px; margin: 30px auto; padding: 16px; }
    .card { background:#fff; border-radius:12px; padding:16px; box-shadow:0 6px 20px rgba(0,0,0,.08); }
```

```

    .row { display:flex; gap:12px; flex-wrap:wrap; }
    input, button { padding:10px; border-radius:10px; border:1px solid #ddd; }
    button { cursor:pointer; background:#222; color:#fff; border:none; }
    #messages { height: 420px; overflow:auto; border:1px solid #eee; border-radius:12px; padding:12px; background:#fafafa; }
    .msg { padding:10px 12px; margin:8px 0; border-radius:12px; background:#fff; border:1px solid #eee; }
    .meta { font-size:12px; color:#666; margin-bottom:6px; }
  </style>
</head>
<body>
  <div class="wrap">
    <div class="card">
      <h2>Chat en ligne</h2>

      <div class="row">
        <input id="author" placeholder="Votre nom" value="Mario" />
        <input id="room" placeholder="Salon (ex: general)" value="general" />
        <button id="join">Rejoindre</button>
        <button id="load">Charger l'historique</button>
      </div>

      <div style="margin-top:14px" id="messages"></div>

      <div class="row" style="margin-top:14px">
        <input id="text" placeholder="Écrire un message..." style="flex:1" />
        <button id="send">Envoyer</button>
      </div>
    </div>
  </div>

<script>
  const messagesEl = document.getElementById("messages");
  const authorEl = document.getElementById("author");
  const roomEl = document.getElementById("room");
  const textEl = document.getElementById("text");

  let socket = null;

  function addMessage({author, content, created_at}) {
    const div = document.createElement("div");
    div.className = "msg";
    div.innerHTML = `
      <div class="meta"><b>${escapeHtml(author)}</b> • ${new Date(created_at).toLocaleString()}</div>
      <div>${escapeHtml(content)}</div>
    `;
    messagesEl.appendChild(div);
    messagesEl.scrollTop = messagesEl.scrollHeight;
  }

```

```

}

function escapeHtml(s) {
  return (s ?? "").replace(/[\&<>"']/g, m => ({
    "&": "&amp;", "<": "&lt;", ">": "&gt;", "\"": "&quot;", "'": "&#039;";
  }[m]));
}

async function loadHistory() {
  const roomName = roomEl.value.trim();
  if (!roomName) return;

  // 1) récupérer la liste des rooms pour obtenir l'id
  const rooms = await fetch("http://127.0.0.1:8000/api/rooms/").then(r => r.json());
  const room = rooms.find(x => x.name === roomName);
  if (!room) {
    alert("Salon inexistant. Cliquez 'Rejoindre' d'abord (WS crée le salon).");
    return;
  }

  // 2) récupérer les messages
  const msgs = await fetch(`http://127.0.0.1:8000/api/rooms/${room.id}/messages/`).then(r => r.json());
  messagesEl.innerHTML = "";
  msgs.forEach(addMessage);
}

function connectWS() {
  const roomName = roomEl.value.trim();
  const author = authorEl.value.trim() || "anonymous";
  if (!roomName) return alert("Choisissez un salon");

  if (socket) socket.close();

  socket = new WebSocket(`ws://127.0.0.1:8000/ws/chat/${encodeURIComponent(roomName)}/`);

  socket.onopen = () => addMessage({author: "system", content: `Connecté au salon "${roomName}"`, created_at: new Date().toISOString()});
  socket.onmessage = (e) => {
    const data = JSON.parse(e.data);
    if (data.type === "chat_message") addMessage(data);
  };
  socket.onclose = () => addMessage({author: "system", content: `Déconnecté`, created_at: new Date().toISOString()});

  document.getElementById("send").onclick = () => {
    const content = textEl.value.trim();
    if (!content || !socket || socket.readyState !== 1) return;
  }
}

```



```
        socket.send(JSON.stringify({ author, content }));
        textEl.value = "";
        textEl.focus();
    };

    textEl.onkeydown = (ev) => {
        if (ev.key === "Enter") document.getElementById("send").click();
    };
}

document.getElementById("join").onclick = connectWS;
document.getElementById("load").onclick = loadHistory;
</script>
</body>
</html>
```

4) Test rapide

1. Lance Django :

```
python manage.py runserver
```

2. Ouvre `index.html` dans 2 onglets, même salon (`general`), deux noms différents.
 3. Clique **Rejoindre** puis envoie des messages.
 4. Clique **Charger l'historique** pour recharger via l'API REST.
-

5) Améliorations faciles (si tu veux)

- Auth (JWT) au lieu du champ `author` libre
 - “typing...” / présence en ligne
 - Pagination / recherche messages
 - CORS (si ton frontend est sur un autre domaine)
 - Redis + déploiement (Daphne/Uvicorn + Nginx)
-

Si tu veux, je peux aussi te donner une **structure de projet complète** (arborescence + tous les fichiers) ou ajouter **authentification JWT** et **CORS** proprement.

Oui je voudrais une structure de projet complète (arborescence + tous les fichiers) au très grand complet. Pouvez vous également mettre le tout dans un fichier winrar ?

Voici une **structure de projet complète** (arborescence + tous les fichiers) prête à l'emploi.

Je ne peux pas générer un **.rar WinRAR** directement ici, mais je t'ai mis le projet dans une archive **.zip** (WinRAR ouvre parfaitement les .zip).



Téléchargement : **django_chat_api_full.zip**

Arborescence du projet

```
django_chat_api/
├── .gitignore
├── README.md
├── manage.py
├── requirements.txt
├── config/
│   ├── __init__.py
│   ├── asgi.py
│   ├── settings.py
│   ├── urls.py
│   └── wsgi.py
├── chat/
│   ├── __init__.py
│   ├── admin.py
│   ├── apps.py
│   ├── consumers.py
│   ├── models.py
│   ├── routing.py
│   ├── serializers.py
│   ├── urls.py
│   ├── views.py
│   └── migrations/
│       └── __init__.py
└── frontend/
    └── index.html
```

Lancer le projet

```
python -m venv venv
# Windows:
venv\Scripts\activate
# Linux/macOS:
# source venv/bin/activate

pip install -r requirements.txt
python manage.py migrate
python manage.py runserver
```

Puis ouvre :

- frontend/index.html (double-clic)
 - Backend : `http://127.0.0.1:8000/`
 - API : `http://127.0.0.1:8000/api/rooms/`
 - WebSocket : `ws://127.0.0.1:8000/ws/chat/<room>/`
-

Si tu veux absolument un **.rar**, dis-moi juste si tu préfères :

- **RAR4** ou **RAR5**,
et je te donne la commande WinRAR (GUI ou ligne de commande) pour convertir le `.zip` en `.rar` en 10 secondes.

Est ce que vous m'avez donné tous les fichier au grand complet dans le fichier zip ?

Oui — le **.zip** contient bien tout le projet “au complet” pour ce chat (backend Django + API + WebSocket + frontend), c’est-à-dire :

- Tous les fichiers Django nécessaires : `manage.py`, `config/*`, `chat/*`
- L’API REST (rooms + messages)
- Le WebSocket (Channels) : `consumers.py` + `routing.py` + `asgi.py`
- Le frontend : `frontend/index.html`
- `requirements.txt`, `README.md`, `.gitignore`

Ce que le **zip** ne peut pas contenir (normal) tant que tu n’as pas exécuté le projet chez toi :

- `db.sqlite3` (la base n'existe qu'après `migrate`)
- Le dossier `venv/` (environnement virtuel)
- Les fichiers de migrations générés (`0001_initial.py`, etc.) : ils apparaissent après `makemigrations`.
→ Mais c'est prévu : Django les génère localement.

Pour obtenir exactement ces fichiers “générés”, fais :

```
python manage.py makemigrations
python manage.py migrate
```

Si tu veux, je peux aussi te fournir une version avec une migration `0001_initial.py` déjà incluse (donc *encore plus “prête”*), mais celle-ci est déjà correcte et standard.

Pouvez vous dire quelle est la version de Django utilisée dans ce projet ?

Dans le projet que je t'ai donné, la version de Django prévue est **Django 5.2.8**.

Tu peux le vérifier dans le fichier `requirements.txt` de l'archive, où il y a :

- `Django==5.2.8`

Ok je vais l'essayer

Parfait. 🍌

Si tu tombes sur un souci, voici les 3 erreurs les plus fréquentes et quoi faire :

1. **WebSocket ne se connecte pas**

- Vérifie que tu lances bien :

```
python manage.py runserver
```

- Et que l'URL WS est bien : `ws://127.0.0.1:8000/ws/chat/general/`

2. **Erreur CORS (si tu ouvres le HTML en local)**

- Dans le projet, `CORS_ALLOW_ALL_ORIGINS = True` est déjà activé, donc l'API REST devrait passer.

- Si ton navigateur bloque quand même, essaye de servir le frontend via un petit serveur :

```
python -m http.server 5500
```

Puis ouvre `http://127.0.0.1:5500/frontend/index.html`

3. Channels/Redis



- En dev, ça marche déjà avec `InMemoryChannelLayer`.
- Si tu passes à Redis plus tard, lance Redis et active la config Redis dans `settings.py`.

Si tu me copies/colles l'erreur exacte (console Django + console navigateur), je te dis précisément quoi corriger.

Page not found (404) Request Method: GET Request URL: `http://127.0.0.1:8000/api/` Using the URLconf defined in `monsite.urls`, Django tried these URL patterns, in this order: `admin/` `api/` `rooms/` `[name='rooms']` `api/` `rooms/<int:room_id>/messages/` `[name='room-messages']` The current path, `api/`, didn't match any of these. You're seeing this error because you have `DEBUG = True` in your Django settings file. Change that to `False`, and Django will display a standard 404 page.

C'est normal : dans ce projet, `/api/` n'est pas une page en soi. On a seulement déclaré des routes **sous** `/api/`.

Tu as exactement ces endpoints :

-  `http://127.0.0.1:8000/api/rooms/`
-  `http://127.0.0.1:8000/api/rooms/<room_id>/messages/`

Donc pour tester tout de suite :

1. Ouvre :

- `http://127.0.0.1:8000/api/rooms/`
Tu devrais voir `[]` au début.

2. Crée un salon (POST) via curl par exemple :

```
curl -X POST http://127.0.0.1:8000/api/rooms/ -H "Content-Type: application/json" -d '{"name":"general"}'
```

Optionnel : afficher quelque chose sur /api/

Si tu veux que `http://127.0.0.1:8000/api/` affiche une page JSON “welcome”, ajoute ça :

1) `chat/views.py` (ajoute en haut) :

```
from rest_framework.response import Response
from rest_framework.views import APIView

class ApiRootView(APIView):
    def get(self, request):
        return Response({
            "rooms": "/api/rooms/",
            "messages": "/api/rooms/<room_id>/messages/"
        })
```

2) `chat/urls.py` :

```
from .views import RoomListCreateView, MessageListView, ApiRootView

urlpatterns = [
    path("", ApiRootView.as_view(), name="api-root"),
    path("rooms/", RoomListCreateView.as_view(), name="rooms"),
    path("rooms/<int:room_id>/messages/", MessageListView.as_view(), name="room-messages"),
]
```

Après ça, `/api/` ne fera plus 404.

Si tu me dis si tu utilises DRF “Browsable API” (interface web) ou uniquement JSON, je te propose la version la plus pratique.

`venv\Lib\site-packages\pip_internal\resolution\resolvelib\candidates.py: config_settings=template.config_settings, venv\Lib\site-packages\pip_internal\utils\appdirs.py: def _macos_user_config_dir(appname: str, roaming: bool = True) -> str: venv\Lib\site-packages\pip_internal\utils\appdirs.py: # Use a Linux-like ~/.config/pip, by default. venv\Lib\site-packages\pip_internal\utils\appdirs.py: linux_like_path = "~/.config/" venv\Lib\site-packages\pip_internal\utils\appdirs.py: def user_config_dir(appname: str, roaming: bool = True) -> str: venv\Lib\site-packages\pip_internal\utils\appdirs.py: return _macos_user_config_dir(appname, roaming) venv\Lib\site-packages\pip_internal\utils\appdirs.py: return _appdirs.user_config_dir(appname, appauthor=False, roaming=roaming) venv\Lib\site-packages\pip_internal\utils\appdirs.py: # for the discussion regarding site_config_dir locations venv\Lib\site-packages\pip_internal\utils\appdirs.py: def site_config_dirs(appname: str) -> List[str]: venv\Lib\site-packages\pip_internal\utils\appdirs.py: dirval = _appdirs.site_config_dir(appname, appauthor=False, multipath=True) venv\Lib\site-packages\pip_internal\utils\compat.py: # dist.location (py27:sysconfig.get_paths()['stdlib'], venv\Lib\site-packages\pip_internal\utils\compat.py: # py26:sysconfig.get_config_vars('LIBDEST')), but fear platform variation may venv\Lib\site-packages\pip_internal\utils\logging.py: """Configures and sets up`

```

all of the logging venv\Lib\site-packages\pip\_internal\utils\logging.py: logging.config.dictConfig( venv\Lib\site-packages\pip\_internal\utils\misc.py:import
sysconfig venv\Lib\site-packages\pip\_internal\utils\misc.py: "ConfiguredBuildBackendHookCaller", venv\Lib\site-packages\pip\_internal\utils\misc.py: If the
`EXTERNALLY-MANAGED` config file is found, the current environment venv\Lib\site-packages\pip\_internal\utils\misc.py: marker =
os.path.join(sysconfig.get_path("stdlib"), "EXTERNALLY-MANAGED") venv\Lib\site-packages\pip\_internal\utils\misc.py: raise
ExternallyManagedEnvironment.from_config(marker) venv\Lib\site-packages\pip\_internal\utils\misc.py:class
ConfiguredBuildBackendHookCaller(BuildBackendHookCaller): venv\Lib\site-packages\pip\_internal\utils\misc.py: config_holder: Any, venv\Lib\site-
packages\pip\_internal\utils\misc.py: self.config_holder = config_holder venv\Lib\site-packages\pip\_internal\utils\misc.py: config_settings: Optional[Dict[str,
Union[str, List[str]]]] = None, venv\Lib\site-packages\pip\_internal\utils\misc.py: cs = self.config_holder.config_settings venv\Lib\site-
packages\pip\_internal\utils\misc.py: wheel_directory, config_settings=cs, metadata_directory=metadata_directory venv\Lib\site-
packages\pip\_internal\utils\misc.py: config_settings: Optional[Dict[str, Union[str, List[str]]]] = None, venv\Lib\site-packages\pip\_internal\utils\misc.py: cs =
self.config_holder.config_settings venv\Lib\site-packages\pip\_internal\utils\misc.py: return super().build_sdists(sdist_directory, config_settings=cs)
venv\Lib\site-packages\pip\_internal\utils\misc.py: config_settings: Optional[Dict[str, Union[str, List[str]]]] = None, venv\Lib\site-
packages\pip\_internal\utils\misc.py: cs = self.config_holder.config_settings venv\Lib\site-packages\pip\_internal\utils\misc.py: wheel_directory,
config_settings=cs, metadata_directory=metadata_directory venv\Lib\site-packages\pip\_internal\utils\misc.py: self, config_settings: Optional[Dict[str,
Union[str, List[str]]]] = None venv\Lib\site-packages\pip\_internal\utils\misc.py: cs = self.config_holder.config_settings venv\Lib\site-
packages\pip\_internal\utils\misc.py: return super().get_requires_for_build_wheel(config_settings=cs) venv\Lib\site-packages\pip\_internal\utils\misc.py: self,
config_settings: Optional[Dict[str, Union[str, List[str]]]] = None venv\Lib\site-packages\pip\_internal\utils\misc.py: cs = self.config_holder.config_settings
venv\Lib\site-packages\pip\_internal\utils\misc.py: return super().get_requires_for_build_sdist(config_settings=cs) venv\Lib\site-
packages\pip\_internal\utils\misc.py: self, config_settings: Optional[Dict[str, Union[str, List[str]]]] = None venv\Lib\site-packages\pip\_internal\utils\misc.py:
cs = self.config_holder.config_settings venv\Lib\site-packages\pip\_internal\utils\misc.py: return super().get_requires_for_build_editable(config_settings=cs)
venv\Lib\site-packages\pip\_internal\utils\misc.py: config_settings: Optional[Dict[str, Union[str, List[str]]]] = None, venv\Lib\site-
packages\pip\_internal\utils\misc.py: cs = self.config_holder.config_settings venv\Lib\site-packages\pip\_internal\utils\misc.py: config_settings=cs,
venv\Lib\site-packages\pip\_internal\utils\misc.py: config_settings: Optional[Dict[str, Union[str, List[str]]]] = None, venv\Lib\site-
packages\pip\_internal\utils\misc.py: cs = self.config_holder.config_settings venv\Lib\site-packages\pip\_internal\utils\misc.py: config_settings=cs,
venv\Lib\site-packages\pip\_internal\utils\setuptools_build.py: no_user_config: bool = False, venv\Lib\site-packages\pip\_internal\utils\setuptools_build.py:
:param no_user_config: If True, disables personal user configuration. venv\Lib\site-packages\pip\_internal\utils\setuptools_build.py: if no_user_config:
venv\Lib\site-packages\pip\_internal\utils\setuptools_build.py: no_user_config: bool, venv\Lib\site-packages\pip\_internal\utils\setuptools_build.py:
no_user_config=no_user_config, venv\Lib\site-packages\pip\_internal\utils\setuptools_build.py: no_user_config: bool, venv\Lib\site-
packages\pip\_internal\utils\setuptools_build.py: args = make_setuptools_shim_args(setup_py_path, no_user_config=no_user_config) venv\Lib\site-
packages\pip\_internal\utils\temp_dir.py: """Get configured auto-delete flag for a given TempDirectory type, venv\Lib\site-
packages\pip\_internal\utils\virtualenv.py: # return True for both checks, but is only able to use the PEP 405 config. venv\Lib\site-
packages\pip\_internal\utils\virtualenv.py: # return True for both checks, but is only able to use the PEP 405 config. venv\Lib\site-
packages\pip\_internal\utils\virtualenv.py: make_command("config", "remote.origin.url", url), venv\Lib\site-packages\pip\_internal\utils\virtualenv.py: url configured.
venv\Lib\site-packages\pip\_internal\utils\virtualenv.py: ["config", "--get-regexp", r"remote\.*\.url"], venv\Lib\site-packages\pip\_internal\utils\virtualenv.py:import
configparser venv\Lib\site-packages\pip\_internal\utils\virtualenv.py: repo_config = os.path.join(dest, self.dirname, "hgrc") venv\Lib\site-
packages\pip\_internal\utils\virtualenv.py: config = configparser.RawConfigParser() venv\Lib\site-packages\pip\_internal\utils\virtualenv.py:
config.read(repo_config) venv\Lib\site-packages\pip\_internal\utils\virtualenv.py: config.set("paths", "default", url.secret) venv\Lib\site-
packages\pip\_internal\utils\virtualenv.py: with open(repo_config, "w") as config_file: venv\Lib\site-packages\pip\_internal\utils\virtualenv.py:

```

```

config.write(config_file) venv\Lib\site-packages\pip\_internal\vc\mercurial.py: except (OSError, configparser.NoSectionError) as exc: venv\Lib\site-
packages\pip\_internal\vc\mercurial.py: ["showconfig", "paths.default"], venv\Lib\site-packages\pip\_internal\vc\versioncontrol.py: url configured.
venv\Lib\site-packages\pip\_vendor\distlib\compat.py: import ConfigParser as configparser venv\Lib\site-packages\pip\_vendor\distlib\compat.py: import
configparser venv\Lib\site-packages\pip\_vendor\distlib\compat.py:import sysconfig venv\Lib\site-packages\pip\_vendor\distlib\compat.py: from
logging.config import BaseConfigurator, valid_ident venv\Lib\site-packages\pip\_vendor\distlib\compat.py: # Each wrapper should have a configurator
attribute holding the actual venv\Lib\site-packages\pip\_vendor\distlib\compat.py: # configurator to use for conversion. venv\Lib\site-
packages\pip\_vendor\distlib\compat.py: result = self.configurator.convert(value) venv\Lib\site-packages\pip\_vendor\distlib\compat.py: result =
self.configurator.convert(value) venv\Lib\site-packages\pip\_vendor\distlib\compat.py: result = self.configurator.convert(value) venv\Lib\site-
packages\pip\_vendor\distlib\compat.py: result = self.configurator.convert(value) venv\Lib\site-packages\pip\_vendor\distlib\compat.py: result =
self.configurator.convert(value) venv\Lib\site-packages\pip\_vendor\distlib\compat.py: result = self.configurator.convert(value) venv\Lib\site-
packages\pip\_vendor\distlib\compat.py: class BaseConfigurator(object): venv\Lib\site-packages\pip\_vendor\distlib\compat.py: The configurator base class
which defines some useful defaults. venv\Lib\site-packages\pip\_vendor\distlib\compat.py: def __init__(self, config): venv\Lib\site-
packages\pip\_vendor\distlib\compat.py: self.config = ConvertingDict(config) venv\Lib\site-packages\pip\_vendor\distlib\compat.py: self.config.configurator =
self venv\Lib\site-packages\pip\_vendor\distlib\compat.py: d = self.config[m.groups()[0]] venv\Lib\site-packages\pip\_vendor\distlib\compat.py:
value.configurator = self venv\Lib\site-packages\pip\_vendor\distlib\compat.py: value.configurator = self venv\Lib\site-
packages\pip\_vendor\distlib\compat.py: value.configurator = self venv\Lib\site-packages\pip\_vendor\distlib\compat.py: def configure_custom(self, config):
venv\Lib\site-packages\pip\_vendor\distlib\compat.py: """Configure an object with a user-supplied factory.""" venv\Lib\site-
packages\pip\_vendor\distlib\compat.py: c = config.pop('') venv\Lib\site-packages\pip\_vendor\distlib\compat.py: props = config.pop('.', None) venv\Lib\site-
packages\pip\_vendor\distlib\compat.py: kwargs = dict([(k, config[k]) for k in config if valid_ident(k)]) venv\Lib\site-packages\pip\_vendor\distlib\index.py:
self.read_configuration() venv\Lib\site-packages\pip\_vendor\distlib\index.py: Get the distutils command for interacting with PyPI configurations.
venv\Lib\site-packages\pip\_vendor\distlib\index.py: def read_configuration(self): venv\Lib\site-packages\pip\_vendor\distlib\index.py: Read the PyPI access
configuration as supported by distutils. This populates venv\Lib\site-packages\pip\_vendor\distlib\index.py: configuration. venv\Lib\site-
packages\pip\_vendor\distlib\index.py: def save_configuration(self): venv\Lib\site-packages\pip\_vendor\distlib\index.py: Save the PyPI access configuration.
You must have set 'username and venv\Lib\site-packages\pip\_vendor\distlib\scripts.py:from .compat import sysconfig, detect_encoding, ZipFile
venv\Lib\site-packages\pip\_vendor\distlib\scripts.py: elif not sysconfig.is_python_build(): venv\Lib\site-packages\pip\_vendor\distlib\scripts.py:
sysconfig.get_path('scripts'), venv\Lib\site-packages\pip\_vendor\distlib\scripts.py: 'python%s' % sysconfig.get_config_var('EXE')) venv\Lib\site-
packages\pip\_vendor\distlib\scripts.py: sysconfig.get_config_var('BINDIR'), venv\Lib\site-packages\pip\_vendor\distlib\scripts.py: 'python%s' %
(sysconfig.get_config_var('EXE')) venv\Lib\site-packages\pip\_vendor\distlib\scripts.py: sysconfig.get_config_var('BINDIR'), venv\Lib\site-
packages\pip\_vendor\distlib\scripts.py: 'python%s%s' % (sysconfig.get_config_var('VERSION'), venv\Lib\site-packages\pip\_vendor\distlib\scripts.py:
sysconfig.get_config_var('EXE')) venv\Lib\site-packages\pip\_vendor\distlib\util.py: HTTPHandler, BaseConfigurator, valid_ident, venv\Lib\site-
packages\pip\_vendor\distlib\util.py: Container, configparser, URLError, ZipFile, fsdecode, venv\Lib\site-packages\pip\_vendor\distlib\util.py: cp =
configparser.ConfigParser() venv\Lib\site-packages\pip\_vendor\distlib\util.py: except configparser.MissingSectionHeaderError: venv\Lib\site-
packages\pip\_vendor\distlib\util.py: cp = configparser.ConfigParser() venv\Lib\site-packages\pip\_vendor\distlib\util.py:# Configurator functionality
venv\Lib\site-packages\pip\_vendor\distlib\util.py:class Configurator(BaseConfigurator): venv\Lib\site-packages\pip\_vendor\distlib\util.py: value_converters
= dict(BaseConfigurator.value_converters) venv\Lib\site-packages\pip\_vendor\distlib\util.py: def __init__(self, config, base=None): venv\Lib\site-
packages\pip\_vendor\distlib\util.py: super(Configurator, self).__init__(config) venv\Lib\site-packages\pip\_vendor\distlib\util.py: def configure_custom(self,
config): venv\Lib\site-packages\pip\_vendor\distlib\util.py: result = self.configure_custom(o) venv\Lib\site-packages\pip\_vendor\distlib\util.py: c =

```



```

config.pop('()') venv\Lib\site-packages\pip\_vendor\distlib\util.py: props = config.pop('.', None) venv\Lib\site-packages\pip\_vendor\distlib\util.py: args =
config.pop('[]', ()) venv\Lib\site-packages\pip\_vendor\distlib\util.py: items = [(k, convert(config[k])) for k in config if valid_ident(k)] venv\Lib\site-
packages\pip\_vendor\distlib\util.py: result = self.config[key] venv\Lib\site-packages\pip\_vendor\distlib\util.py: self.config[key] = result =
self.configure_custom(result) venv\Lib\site-packages\pip\_vendor\distlib\util.py:# Get the distutils command for interacting with PyPI configurations.
venv\Lib\site-packages\pip\_vendor\distlib\util.py:# from distutils.config import PyPIRCCommand venv\Lib\site-packages\pip\_vendor\distlib\util.py: config =
configparser.RawConfigParser() venv\Lib\site-packages\pip\_vendor\distlib\util.py: config.read(self.filename) venv\Lib\site-
packages\pip\_vendor\distlib\util.py: sections = config.sections() venv\Lib\site-packages\pip\_vendor\distlib\util.py: index_servers = config.get('distutils',
'index-servers') venv\Lib\site-packages\pip\_vendor\distlib\util.py: result['username'] = config.get(server, 'username') venv\Lib\site-
packages\pip\_vendor\distlib\util.py: if config.has_option(server, key): venv\Lib\site-packages\pip\_vendor\distlib\util.py: result[key] = config.get(server, key)
venv\Lib\site-packages\pip\_vendor\distlib\util.py: # section of their config set to the HTTP (rather than venv\Lib\site-packages\pip\_vendor\distlib\util.py: if
config.has_option(server, 'repository'): venv\Lib\site-packages\pip\_vendor\distlib\util.py: repository = config.get(server, 'repository') venv\Lib\site-
packages\pip\_vendor\distlib\util.py: 'username': config.get(server, 'username'), venv\Lib\site-packages\pip\_vendor\distlib\util.py: 'password':
config.get(server, 'password'), venv\Lib\site-packages\pip\_vendor\distlib\util.py: config = configparser.RawConfigParser() venv\Lib\site-
packages\pip\_vendor\distlib\util.py: config.read(fn) venv\Lib\site-packages\pip\_vendor\distlib\util.py: if not config.has_section('pypi'): venv\Lib\site-
packages\pip\_vendor\distlib\util.py: config.add_section('pypi') venv\Lib\site-packages\pip\_vendor\distlib\util.py: config.set('pypi', 'username', username)
venv\Lib\site-packages\pip\_vendor\distlib\util.py: config.set('pypi', 'password', password) venv\Lib\site-packages\pip\_vendor\distlib\util.py: config.write(f)
venv\Lib\site-packages\pip\_vendor\distlib\util.py: Read the PyPI access configuration as supported by distutils. venv\Lib\site-
packages\pip\_vendor\distlib\util.py: from distutils import sysconfig venv\Lib\site-packages\pip\_vendor\distlib\util.py: import sysconfig venv\Lib\site-
packages\pip\_vendor\distlib\util.py: sysconfig.get_config_vars(), osname, release, machine) venv\Lib\site-packages\pip\_vendor\distlib\wheel.py:from
.compat import sysconfig, ZipFile, fsdecode, text_type, filter venv\Lib\site-packages\pip\_vendor\distlib\wheel.py:VER_SUFFIX =
sysconfig.get_config_var('py_version_nodot') venv\Lib\site-packages\pip\_vendor\distlib\wheel.py:ABI = sysconfig.get_config_var('SOABI') venv\Lib\site-
packages\pip\_vendor\distlib\wheel.py: if sysconfig.get_config_var('Py_DEBUG'): venv\Lib\site-packages\pip\_vendor\distlib\wheel.py: wpm =
sysconfig.get_config_var('WITH_PYMALLOC') venv\Lib\site-packages\pip\_vendor\distlib\wheel.py: us = sysconfig.get_config_var('Py_UNICODE_SIZE')
venv\Lib\site-packages\pip\_vendor\packaging\tags.py:import sysconfig venv\Lib\site-packages\pip\_vendor\packaging\tags.py:def _get_config_var(name: str,
warn: bool = False) -> Union[int, str, None]: venv\Lib\site-packages\pip\_vendor\packaging\tags.py: value = sysconfig.get_config_var(name) venv\Lib\site-
packages\pip\_vendor\packaging\tags.py: "Config variable '%s' is unset, Python ABI tag may be incorrect", name venv\Lib\site-
packages\pip\_vendor\packaging\tags.py: with_debug = _get_config_var("Py_DEBUG", warn) venv\Lib\site-packages\pip\_vendor\packaging\tags.py:
with_pymalloc = _get_config_var("WITH_PYMALLOC", warn) venv\Lib\site-packages\pip\_vendor\packaging\tags.py: unicode_size =
_get_config_var("Py_UNICODE_SIZE", warn) venv\Lib\site-packages\pip\_vendor\packaging\tags.py: abi = sysconfig.get_config_var("SOABI")
venv\Lib\site-packages\pip\_vendor\packaging\tags.py: linux = _normalize_string(sysconfig.get_platform()) venv\Lib\site-
packages\pip\_vendor\packaging\tags.py: yield _normalize_string(sysconfig.get_platform()) venv\Lib\site-packages\pip\_vendor\packaging\tags.py: version =
_get_config_var("py_version_nodot", warn=warn) venv\Lib\site-packages\pip\_vendor\packaging\_musllinux.py: import sysconfig venv\Lib\site-
packages\pip\_vendor\packaging\_musllinux.py: plat = sysconfig.get_platform() venv\Lib\site-packages\pip\_vendor\pkg_resources\__init__.py: from
sysconfig import get_platform venv\Lib\site-packages\pip\_vendor\platformdirs\android.py: def user_config_dir(self) -> str: venv\Lib\site-
packages\pip\_vendor\platformdirs\android.py: :return: config directory tied to the user, e.g. \ venv\Lib\site-packages\pip\_vendor\platformdirs\android.py: def
site_config_dir(self) -> str: venv\Lib\site-packages\pip\_vendor\platformdirs\android.py: """:return: config directory shared by the users, same as
user_config_dir"""" venv\Lib\site-packages\pip\_vendor\platformdirs\android.py: return self.user_config_dir venv\Lib\site-

```

```

packages\pip\_vendor\platformdirs\api.py: def user_config_dir(self) -> str: venv\Lib\site-packages\pip\_vendor\platformdirs\api.py: """return: config directory
tied to the user""" venv\Lib\site-packages\pip\_vendor\platformdirs\api.py: def site_config_dir(self) -> str: venv\Lib\site-
packages\pip\_vendor\platformdirs\api.py: """return: config directory shared by the users""" venv\Lib\site-packages\pip\_vendor\platformdirs\api.py: def
user_config_path(self) -> Path: venv\Lib\site-packages\pip\_vendor\platformdirs\api.py: """return: config path tied to the user""" venv\Lib\site-
packages\pip\_vendor\platformdirs\api.py: return Path(self.user_config_dir) venv\Lib\site-packages\pip\_vendor\platformdirs\api.py: def site_config_path(self)
-> Path: venv\Lib\site-packages\pip\_vendor\platformdirs\api.py: """return: config path shared by the users""" venv\Lib\site-
packages\pip\_vendor\platformdirs\api.py: return Path(self.site_config_dir) venv\Lib\site-packages\pip\_vendor\platformdirs\macos.py: def
user_config_dir(self) -> str: venv\Lib\site-packages\pip\_vendor\platformdirs\macos.py: """return: config directory tied to the user, same as
user_data_dir""" venv\Lib\site-packages\pip\_vendor\platformdirs\macos.py: def site_config_dir(self) -> str: venv\Lib\site-
packages\pip\_vendor\platformdirs\macos.py: """return: config directory shared by the users, same as site_data_dir""" venv\Lib\site-
packages\pip\_vendor\platformdirs\unix.py: from configparser import ConfigParser venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: def
user_config_dir(self) -> str: venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: :return: config directory tied to the user, e.g.
`~/ .config/$appname/$version` or venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: `XDG_CONFIG_HOME/$appname/$version` venv\Lib\site-
packages\pip\_vendor\platformdirs\unix.py: path = os.environ.get("XDG_CONFIG_HOME", "") venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py:
path = os.path.expanduser("~/ .config") # noqa: PTH111 venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: def site_config_dir(self) -> str:
venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: :return: config directories shared by users (if multipath
<platformdirs.api.PlatformDirsABC.multipath> venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: # XDG default for $XDG_CONFIG_DIRS
only first, if multipath is False venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: path = os.environ.get("XDG_CONFIG_DIRS", "") venv\Lib\site-
packages\pip\_vendor\platformdirs\unix.py: def site_config_path(self) -> Path: venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: """return: config
path shared by the users. Only return first item, even if `multipath` is set to True""" venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: return
self.first_item_as_path_if_multipath(self.site_config_dir) venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: """Return directory from user-dirs.dirs
config file. See https://freedesktop.org/wiki/Software/xdg-user-dirs/.""" venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: user_dirs_config_path =
Path(Unix().user_config_dir) / "user-dirs.dirs" venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: if user_dirs_config_path.exists(): venv\Lib\site-
packages\pip\_vendor\platformdirs\unix.py: parser = ConfigParser() venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: with
user_dirs_config_path.open() as stream: venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: # Add fake section header, so ConfigParser doesn't complain
venv\Lib\site-packages\pip\_vendor\platformdirs\windows.py: def user_config_dir(self) -> str: venv\Lib\site-packages\pip\_vendor\platformdirs\windows.py:
"""return: config directory tied to the user, same as user_data_dir""" venv\Lib\site-packages\pip\_vendor\platformdirs\windows.py: def site_config_dir(self)
-> str: venv\Lib\site-packages\pip\_vendor\platformdirs\windows.py: """return: config directory shared by the users, same as site_data_dir""" venv\Lib\site-
packages\pip\_vendor\platformdirs\__init__.py: def user_config_dir(venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: :returns: config directory
tied to the user venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: ).user_config_dir venv\Lib\site-
packages\pip\_vendor\platformdirs\__init__.py: def site_config_dir(venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: :returns: config directory
shared by the users venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: ).site_config_dir venv\Lib\site-
packages\pip\_vendor\platformdirs\__init__.py: def user_config_path(venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: :returns: config path tied to
the user venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: ).user_config_path venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: def
site_config_path(venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: :returns: config path shared by the users venv\Lib\site-
packages\pip\_vendor\platformdirs\__init__.py: ).site_config_path venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: "user_config_dir",
venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: "site_config_dir", venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py:

```

"user_config_path", venv\Lib\site-packages\pip_vendor\platformdirs__init__.py: "site_config_path", venv\Lib\site-packages\pip_vendor\platformdirs__main__.py: "user_config_dir", venv\Lib\site-packages\pip_vendor\platformdirs__main__.py: "site_config_dir", venv\Lib\site-packages\pip_vendor\pygments\formatters\html.py: # some configurations seem to mess up the formatting... venv\Lib\site-packages\pip_vendor\pygments\formatters\img.py: if line.startswith(b'Fontconfig warning:'): venv\Lib\site-packages\pip_vendor\pygments\lexers\python.py: 'show_config', 'shuffle', 'sign', 'signbit', 'sin', 'sinc', 'sinh', venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'ApacheConfLexer': (pip._vendor.pygments.lexers.configs, 'ApacheConf', ('apacheconf', 'aconf', 'apache'), ('.htaccess', 'apache.conf', 'apache2.conf'), ('text/x-apacheconf',)), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'AugeasLexer': (pip._vendor.pygments.lexers.configs, 'Augeas', ('augeas',), (*'.aug',), ()), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'Cfengine3Lexer': (pip._vendor.pygments.lexers.configs, 'CFEngine3', ('cfengine3', 'cf3'), (*'.cf',), ()), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'DockerLexer': (pip._vendor.pygments.lexers.configs, 'Docker', ('docker', 'dockerfile'), ('Dockerfile', '*.docker'), ('text/x-dockerfile-config',)), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'IniLexer': (pip._vendor.pygments.lexers.configs, 'INI', ('ini', 'cfg', 'dosini'), (*'.ini', '*.cfg', '*.inf', '.editorconfig', '*.service', '*.socket', '*.device', '*.mount', '*.automount', '*.swap', '*.target', '*.path', '*.timer', '*.slice', '*.scope'), ('text/x-ini', 'text/inf')), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'KconfigLexer': (pip._vendor.pygments.lexers.configs, 'Kconfig', ('kconfig', 'menuconfig', 'linux-config', 'kernel-config'), ('Kconfig*', '*Config.in*', 'external.in*', 'standard-modules.in'), ('text/x-kconfig',)), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'LighttpdConfLexer': (pip._vendor.pygments.lexers.configs, 'Lighttpd configuration file', ('lighttpd', 'lighty'), ('lighttpd.conf',), ('text/x-lighttpd-conf',)), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'NestedTextLexer': (pip._vendor.pygments.lexers.configs, 'NestedText', ('nestedtext', 'nt'), (*'.nt',), ()), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'NginxConfLexer': (pip._vendor.pygments.lexers.configs, 'Nginx configuration file', ('nginx',), ('nginx.conf',), ('text/x-nginx-conf',)), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'PacmanConfLexer': (pip._vendor.pygments.lexers.configs, 'PacmanConf', ('pacmanconf',), ('pacman.conf',), ()), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'PkgConfigLexer': (pip._vendor.pygments.lexers.configs, 'PkgConfig', ('pkgconfig',), (*'.pc',), ()), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'PropertiesLexer': (pip._vendor.pygments.lexers.configs, 'Properties', ('properties', 'jproperties'), (*'.properties',), ('text/x-java-properties',)), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'RegeditLexer': (pip._vendor.pygments.lexers.configs, 'reg', ('registry',), (*'.reg',), ('text/x-windows-registry',)), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'SingularityLexer': (pip._vendor.pygments.lexers.configs, 'Singularity', ('singularity',), (*'.def', 'Singularity'), ()), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'SquidConfLexer': (pip._vendor.pygments.lexers.configs, 'SquidConf', ('squidconf', 'squid.conf', 'squid'), ('squid.conf',), ('text/x-squidconf',)), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'TOMLLexer': (pip._vendor.pygments.lexers.configs, 'TOML', ('toml',), (*'.toml', 'Pipfile', 'poetry.lock'), ()), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'TermcapLexer': (pip._vendor.pygments.lexers.configs, 'Termcap', ('termcap',), ('termcap', 'termcap.src'), ()), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'TerminfoLexer': (pip._vendor.pygments.lexers.configs, 'Terminfo', ('terminfo',), ('terminfo', 'terminfo.src'), ()), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'TerraformLexer': (pip._vendor.pygments.lexers.configs, 'Terraform', ('terraform', 'tf', 'hcl'), (*'.tf', '*.hcl'), ('application/x-tf', 'application/x-terraform')), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'UnixConfigLexer': (pip._vendor.pygments.lexers.configs, 'Unix/Linux config files', ('unixconfig', 'linuxconfig'), ()), venv\Lib\site-packages\pip_vendor\pyparsing\core.py: __config_flags, venv\Lib\site-packages\pip_vendor\pyparsing\core.py: class __compat__(__config_flags): venv\Lib\site-packages\pip_vendor\pyparsing\core.py: A cross-version compatibility configuration for pyparsing features that will be venv\Lib\site-packages\pip_vendor\pyparsing\core.py: released in a future version. By setting values in this configuration to True, venv\Lib\site-packages\pip_vendor\pyparsing\core.py: class __diag__(__config_flags): venv\Lib\site-packages\pip_vendor\pyparsing\core.py: Diagnostic configuration (all default to disabled) venv\Lib\site-packages\pip_vendor\pyparsing\core.py: del __config_flags venv\Lib\site-packages\pip_vendor\pyparsing\testing.py:

Context manager to be used when writing unit tests that modify pyparsing config values: venv\Lib\site-packages\pip_vendor\pyparsing\util.py: class

```
__config_flags: venv\Lib\site-packages\pip\_vendor\pyparsing\util.py: _type_desc = "configuration" venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: def get_requires_for_build_wheel(self, config_settings=None): venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: 'config_settings': config_settings venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: self, metadata_directory, config_settings=None, venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: 'config_settings': config_settings, venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: self, wheel_directory, config_settings=None, venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: 'config_settings': config_settings, venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: def get_requires_for_build_editable(self, config_settings=None): venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: 'config_settings': config_settings venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: self, metadata_directory, config_settings=None, venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: 'config_settings': config_settings, venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: self, wheel_directory, config_settings=None, venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: 'config_settings': config_settings, venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: def get_requires_for_build_sdist(self, config_settings=None): venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: 'config_settings': config_settings venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: def build_sdist(self, sdist_directory, config_settings=None): venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: 'config_settings': config_settings, venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: def get_requires_for_build_wheel(config_settings): venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: return hook(config_settings) venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: def get_requires_for_build_editable(config_settings): venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: return hook(config_settings) venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: metadata_directory, config_settings, _allow_fallback): venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: return hook(metadata_directory, config_settings) venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: whl_basename = backend.build_wheel(metadata_directory, config_settings) venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: config_settings) venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: metadata_directory, config_settings, _allow_fallback): venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: whl_basename = build_wheel(metadata_directory, config_settings) venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: config_settings) venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: return hook(metadata_directory, config_settings) venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: whl_basename, metadata_directory, config_settings): venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: def build_wheel(wheel_directory, config_settings, metadata_directory=None): venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: return _build_backend().build_wheel(wheel_directory, config_settings, metadata_directory=None): venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: return hook(wheel_directory, config_settings, metadata_directory) venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: def get_requires_for_build_sdist(config_settings): venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: return hook(config_settings) venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: def build_sdist(sdist_directory, config_settings): venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: return backend.build_sdist(sdist_directory, config_settings) venv\Lib\site-packages\pip\_vendor\requests\adapters.py: "config", venv\Lib\site-packages\pip\_vendor\requests\adapters.py: self.config = {} venv\Lib\site-packages\pip\_vendor\requests\adapters.py: self.config = {} venv\Lib\site-packages\pip\_vendor\requests\adapters.py: :param proxy_kwargs: Extra keyword
```

arguments used to configure the Proxy Manager. venv\Lib\site-packages\pip_vendor\requests\sessions.py: """This method re-evaluates the proxy configuration by considering the venv\Lib\site-packages\pip_vendor\requests\sessions.py: NO_PROXY, we strip the proxy configuration. Otherwise, we set missing venv\Lib\site-packages\pip_vendor\requests\sessions.py: Provides cookie persistence, connection-pooling, and configuration. venv\Lib\site-packages\pip_vendor\requests\sessions.py: #: Trust environment settings for proxy configuration, default venv\Lib\site-packages\pip_vendor\requests\sessions.py: # Look for requests environment configuration venv\Lib\site-packages\pip_vendor\requests\utils.py: """This method takes proxy information from a request and configuration venv\Lib\site-packages\pip_vendor\requests\utils.py: such a NO_PROXY to strip proxy configurations. venv\Lib\site-packages\pip_vendor\requests\utils.py: :param trust_env: Boolean declaring whether to trust environment configs venv\Lib\site-packages\pip_vendor\rich\console.py: Some terminals may require configuration changes before you can set the title. venv\Lib\site-packages\pip_vendor\rich\logging.py: Be careful when enabling console markup in log messages if you have configured logging for libraries not venv\Lib\site-packages\pip_vendor\rich\logging.py: logging.basicConfig(venv\Lib\site-packages\pip_vendor\rich\logging.py: "Loading configuration file /adasd/asdasd/qeqwe/qwrqwrqwr/sdgsdgsdg/werwerwer/dfgerert/ertertert/ertetert/werwerwer" venv\Lib\site-packages\pip_vendor\rich\theme.py:import configparser venv\Lib\site-packages\pip_vendor\rich\theme.py: def config(self) -> str: venv\Lib\site-packages\pip_vendor\rich\theme.py: """Get contents of a config file for this theme.""" venv\Lib\site-packages\pip_vendor\rich\theme.py: config = "[styles]\n" + "\n".join(venv\Lib\site-packages\pip_vendor\rich\theme.py: return config venv\Lib\site-packages\pip_vendor\rich\theme.py: cls, config_file: IO[str], source: Optional[str] = None, inherit: bool = True venv\Lib\site-packages\pip_vendor\rich\theme.py: config_file (IO[str]): An open conf file. venv\Lib\site-packages\pip_vendor\rich\theme.py: config = configparser.ConfigParser() venv\Lib\site-packages\pip_vendor\rich\theme.py: config.read_file(config_file, source=source) venv\Lib\site-packages\pip_vendor\rich\theme.py: styles = {name: Style.parse(value) for name, value in config.items("styles")} venv\Lib\site-packages\pip_vendor\rich\theme.py: path (str): Path to a config file readable by Python configparser module. venv\Lib\site-packages\pip_vendor\rich\theme.py: encoding (str, optional): Encoding of the config file. Defaults to None. venv\Lib\site-packages\pip_vendor\rich\theme.py: with open(path, "rt", encoding=encoding) as config_file: venv\Lib\site-packages\pip_vendor\rich\theme.py: return cls.from_file(config_file, source=path, inherit=inherit) venv\Lib\site-packages\pip_vendor\rich\theme.py: print(theme.config) venv\Lib\site-packages\pip_vendor\rich__init__.py: __all__ = ["get_console", "reconfigure", "print", "inspect", "print_json"] venv\Lib\site-packages\pip_vendor\rich__init__.py:def reconfigure(*args: Any, **kwargs: Any) -> None: venv\Lib\site-packages\pip_vendor\rich__init__.py: """Reconfigures the global console by replacing it with another. venv\Lib\site-packages\pip_vendor\six.py: MovedModule("configparser", "ConfigParser"), venv\Lib\site-packages\pip_vendor\truststore_api.py: from ._windows import _configure_context, _verify_peercerts_impl venv\Lib\site-packages\pip_vendor\truststore_api.py: from ._macos import _configure_context, _verify_peercerts_impl venv\Lib\site-packages\pip_vendor\truststore_api.py: from ._openssl import _configure_context, _verify_peercerts_impl venv\Lib\site-packages\pip_vendor\truststore_api.py: with _configure_context(self._ctx): venv\Lib\site-packages\pip_vendor\truststore_api.py: with _configure_context(self._ctx): venv\Lib\site-packages\pip_vendor\truststore_macos.py:def _configure_context(ctx: ssl.SSLContext) -> typing.Iterator[None]: venv\Lib\site-packages\pip_vendor\truststore_macos.py: # explicitly okay with from SSLContext configuration venv\Lib\site-packages\pip_vendor\truststore_openssl.py:def _configure_context(ctx: ssl.SSLContext) -> typing.Iterator[None]: venv\Lib\site-packages\pip_vendor\truststore_openssl.py: # or the path configured when OpenSSL was compiled, venv\Lib\site-packages\pip_vendor\truststore_openssl.py: # or the path configured when OpenSSL was compiled, venv\Lib\site-packages\pip_vendor\truststore_windows.py:class CERT_CHAIN_ENGINE_CONFIG(Structure): venv\Lib\site-packages\pip_vendor\truststore_windows.py:PCERT_CHAIN_ENGINE_CONFIG = POINTER(CERT_CHAIN_ENGINE_CONFIG) venv\Lib\site-packages\pip_vendor\truststore_windows.py: PCERT_CHAIN_ENGINE_CONFIG, venv\Lib\site-packages\pip_vendor\truststore_windows.py: cert_chain_engine_config = CERT_CHAIN_ENGINE_CONFIG() venv\Lib\site-packages\pip_vendor\truststore_windows.py: cert_chain_engine_config.cbSize = sizeof(cert_chain_engine_config) venv\Lib\site-packages\pip_vendor\truststore_windows.py:

cert_chain_engine_config.hExclusiveRoot = hRootCertStore venv\Lib\site-packages\pip_vendor\truststore_windows.py: pConfig = pointer(cert_chain_engine_config) venv\Lib\site-packages\pip_vendor\truststore_windows.py: pConfig, venv\Lib\site-packages\pip_vendor\truststore_windows.py: def _configure_context(ctx: ssl.SSLContext) -> typing.Iterator[None]: venv\Lib\site-packages\pip_vendor\urllib3\connection.py: Additional keyword parameters are used to configure attributes of the connection. venv\Lib\site-packages\pip_vendor\urllib3\connection.py: self.proxy_config = kw.pop("proxy_config", None) venv\Lib\site-packages\pip_vendor\urllib3\connection.py: proxy_config = self.proxy_config venv\Lib\site-packages\pip_vendor\urllib3\connection.py: ssl_context = proxy_config.ssl_context venv\Lib\site-packages\pip_vendor\urllib3\connectionpool.py: Retry configuration to use by default with requests in this pool. venv\Lib\site-packages\pip_vendor\urllib3\connectionpool.py: _proxy_config=None, venv\Lib\site-packages\pip_vendor\urllib3\connectionpool.py: self.proxy_config = _proxy_config venv\Lib\site-packages\pip_vendor\urllib3\connectionpool.py: self.conn_kw["proxy_config"] = self.proxy_config venv\Lib\site-packages\pip_vendor\urllib3\connectionpool.py: Configure the number of retries to allow before raising a venv\Lib\site-packages\pip_vendor\urllib3\connectionpool.py: self.proxy, self.proxy_config, destination_scheme venv\Lib\site-packages\pip_vendor\urllib3\contrib\appengine.py::class:PoolManager without any configuration or special environment variables. venv\Lib\site-packages\pip_vendor\urllib3\contrib\securetransport.py: # We save off the previously-configured timeout and then set it to venv\Lib\site-packages\pip_vendor\urllib3\exceptions.py: """Warned when certain TLS/SSL configuration is not available on a platform.""" venv\Lib\site-packages\pip_vendor\urllib3\packages\six.py: MovedModule("configparser", "ConfigParser"), venv\Lib\site-packages\pip_vendor\urllib3\poolmanager.py: "key__proxy_config", # class venv\Lib\site-packages\pip_vendor\urllib3\poolmanager.py: _proxy_config_fields = ("ssl_context", "use_forwarding_for_https") venv\Lib\site-packages\pip_vendor\urllib3\poolmanager.py: ProxyConfig = collections.namedtuple("ProxyConfig", _proxy_config_fields) venv\Lib\site-packages\pip_vendor\urllib3\poolmanager.py: #: Each PoolManager makes a copy of this dictionary so they can be configured venv\Lib\site-packages\pip_vendor\urllib3\poolmanager.py: proxy_config = None venv\Lib\site-packages\pip_vendor\urllib3\poolmanager.py: self.proxy, self.proxy_config, parsed_url.scheme venv\Lib\site-packages\pip_vendor\urllib3\poolmanager.py: if six.PY2 and not self.proxy_config.use_forwarding_for_https: venv\Lib\site-packages\pip_vendor\urllib3\poolmanager.py: self.proxy_config = ProxyConfig(proxy_ssl_context, use_forwarding_for_https) venv\Lib\site-packages\pip_vendor\urllib3\poolmanager.py: connection_pool_kw["_proxy_config"] = self.proxy_config venv\Lib\site-packages\pip_vendor\urllib3\poolmanager.py: if not connection_requires_http_tunnel(self.proxy, self.proxy_config, u.scheme): venv\Lib\site-packages\pip_vendor\urllib3\util\proxy.py: proxy_url=None, proxy_config=None, destination_scheme=None venv\Lib\site-packages\pip_vendor\urllib3\util\proxy.py: :param ProxyConfig proxy_config: venv\Lib\site-packages\pip_vendor\urllib3\util\proxy.py: Proxy configuration from poolmanager.py venv\Lib\site-packages\pip_vendor\urllib3\util\proxy.py: and proxy_config venv\Lib\site-packages\pip_vendor\urllib3\util\proxy.py: and proxy_config.use_forwarding_for_https venv\Lib\site-packages\pip_vendor\urllib3\util\retry.py: """Retry configuration. venv\Lib\site-packages\pip_vendor\urllib3\util\ssl.py: "urllib3 from configuring SSL appropriately and may cause " venv\Lib\site-packages\pip_vendor\urllib3\util\timeout.py: """Timeout configuration. venv\Lib\site-packages\pip_vendor\urllib3\util\timeout.py: Timeout object to ensure each one has its own start/stop configured. venv\Lib\site-packages\pip_vendor\urllib3\util\wait.py: # to use it, either due to strange configuration or broken monkeypatching venv\Lib\site-packages\redis\asyncio\client.py: configuration, an instance will either use a ConnectionPool, or venv\Lib\site-packages\redis\asyncio\client.py: Return a Redis client object configured from the given URL venv\Lib\site-packages\redis\asyncio\client.py: 2. Define a `Retry` object with configured 'supported_errors' and set venv\Lib\site-packages\redis\asyncio\client.py: When 'connection_pool' is provided - the retry configuration of the venv\Lib\site-packages\redis\asyncio\cluster.py: <https://redis.io/docs/manual/scaling/#redis-cluster-configuration-parameters> venv\Lib\site-packages\redis\asyncio\cluster.py: | @deprecated - Please configure the 'retry' object instead venv\Lib\site-packages\redis\asyncio\cluster.py: Return a Redis client object configured from the given URL. venv\Lib\site-packages\redis\asyncio\cluster.py: reason="Please configure the 'load_balancing_strategy' instead", venv\Lib\site-packages\redis\asyncio\cluster.py: reason="Please configure the 'retry' object instead", venv\Lib\site-packages\redis\asyncio\cluster.py: # Call our

on_connect function to configure READONLY mode venv\Lib\site-packages\redis\asyncio\cluster.py: # Sending READONLY command to server to configure connection as venv\Lib\site-packages\redis\asyncio\cluster.py: # rc.cluster_save_config(rc.get primaries()) venv\Lib\site-packages\redis\asyncio\cluster.py: # We are configuring the connection pool not to retry venv\Lib\site-packages\redis\asyncio\connection.py: # the command name, e.g., 'CONFIG GET'. The Redis server expects these venv\Lib\site-packages\redis\asyncio\connection.py: Return a connection pool configured from the given URL. venv\Lib\site-packages\redis\asyncio\multidb\client.py: from redis.asyncio.multidb.config import DEFAULT_GRACE_PERIOD, MultiDbConfig venv\Lib\site-packages\redis\asyncio\multidb\client.py: def __init__(self, config: MultiDbConfig): venv\Lib\site-packages\redis\asyncio\multidb\client.py: self._databases = config.databases() venv\Lib\site-packages\redis\asyncio\multidb\client.py: config.default_health_checks() venv\Lib\site-packages\redis\asyncio\multidb\client.py: if not config.health_checks venv\Lib\site-packages\redis\asyncio\multidb\client.py: else config.health_checks venv\Lib\site-packages\redis\asyncio\multidb\client.py: self._health_check_interval = config.health_check_interval venv\Lib\site-packages\redis\asyncio\multidb\client.py: self._health_check_policy: HealthCheckPolicy = config.health_check_policy.value(venv\Lib\site-packages\redis\asyncio\multidb\client.py: config.health_check_probes, config.health_check_delay venv\Lib\site-packages\redis\asyncio\multidb\client.py: config.default_failure_detectors() venv\Lib\site-packages\redis\asyncio\multidb\client.py: if not config.failure_detectors venv\Lib\site-packages\redis\asyncio\multidb\client.py: else config.failure_detectors venv\Lib\site-packages\redis\asyncio\multidb\client.py: config.default_failover_strategy() venv\Lib\site-packages\redis\asyncio\multidb\client.py: if config.failover_strategy is None venv\Lib\site-packages\redis\asyncio\multidb\client.py: else config.failover_strategy venv\Lib\site-packages\redis\asyncio\multidb\client.py: self._auto_fallback_interval = config.auto_fallback_interval venv\Lib\site-packages\redis\asyncio\multidb\client.py: self._event_dispatcher = config.event_dispatcher venv\Lib\site-packages\redis\asyncio\multidb\client.py: self._command_retry = config.command_retry venv\Lib\site-packages\redis\asyncio\multidb\client.py: failover_attempts=config.failover_attempts, venv\Lib\site-packages\redis\asyncio\multidb\client.py: failover_delay=config.failover_delay, venv\Lib\site-packages\redis\asyncio\multidb\client.py: self._config = config venv\Lib\site-packages\redis\asyncio\multidb\command_executor.py: from redis.multidb.config import DEFAULT_AUTO_FALLBACK_INTERVAL venv\Lib\site-packages\redis\asyncio\multidb\config.py: class DatabaseConfig: venv\Lib\site-packages\redis\asyncio\multidb\config.py: Dataclass representing the configuration for a database connection. venv\Lib\site-packages\redis\asyncio\multidb\config.py: This class is used to store configuration settings for a database connection, venv\Lib\site-packages\redis\asyncio\multidb\config.py: attributes and allows for the creation of customized configurations for various venv\Lib\site-packages\redis\asyncio\multidb\config.py: from_pool (Optional[ConnectionPool]): A pre-configured connection pool to use. venv\Lib\site-packages\redis\asyncio\multidb\config.py: class MultiDbConfig: venv\Lib\site-packages\redis\asyncio\multidb\config.py: Configuration class for managing multiple database connections in a resilient and fail-safe manner. venv\Lib\site-packages\redis\asyncio\multidb\config.py: databases_config: A list of database configurations. venv\Lib\site-packages\redis\asyncio\multidb\config.py: Retrieves a collection of database clients managed by weighted configurations. venv\Lib\site-packages\redis\asyncio\multidb\config.py: Initializes database clients based on the provided configuration and removes venv\Lib\site-packages\redis\asyncio\multidb\config.py: with defined retry and backoff configurations. venv\Lib\site-packages\redis\asyncio\multidb\config.py: databases_config: List[DatabaseConfig] venv\Lib\site-packages\redis\asyncio\multidb\config.py: for database_config in self.databases_config: venv\Lib\site-packages\redis\asyncio\multidb\config.py: database_config.client_kwargs.update(venv\Lib\site-packages\redis\asyncio\multidb\config.py: if database_config.from_url: venv\Lib\site-packages\redis\asyncio\multidb\config.py: database_config.from_url, **database_config.client_kwargs venv\Lib\site-packages\redis\asyncio\multidb\config.py: elif database_config.from_pool: venv\Lib\site-packages\redis\asyncio\multidb\config.py: database_config.from_pool.set_retry(venv\Lib\site-packages\redis\asyncio\multidb\config.py: connection_pool=database_config.from_pool venv\Lib\site-packages\redis\asyncio\multidb\config.py: client = self.client_class(**database_config.client_kwargs) venv\Lib\site-packages\redis\asyncio\multidb\config.py: database_config.default_circuit_breaker() venv\Lib\site-packages\redis\asyncio\multidb\config.py: if database_config.circuit is None venv\Lib\site-packages\redis\asyncio\multidb\config.py: else database_config.circuit venv\Lib\site-packages\redis\asyncio\multidb\config.py:

weight=database_config.weight, venv\Lib\site-packages\redis\asyncio\multidb\config.py: health_check_url=database_config.health_check_url, venv\Lib\site-packages\redis\asyncio\multidb\config.py: database_config.weight, venv\Lib\site-packages\redis\asyncio\multidb\healthcheck.py: "Database health check url is not set. Please check DatabaseConfig for the current database." venv\Lib\site-packages\redis\auth\idp.py:class IdentityProviderConfigInterface(ABC): venv\Lib\site-packages\redis\auth\idp.py: Configuration class that provides a configured identity provider. venv\Lib\site-packages\redis\auth\token_manager.py:class TokenManagerConfig: venv\Lib\site-packages\redis\auth\token_manager.py: self, identity_provider: IdentityProviderInterface, config: TokenManagerConfig venv\Lib\site-packages\redis\auth\token_manager.py: self._config = config venv\Lib\site-packages\redis\auth\token_manager.py: if self._retries < self._config.get_retry_policy().get_max_attempts(): venv\Lib\site-packages\redis\auth\token_manager.py: sleep(self._config.get_retry_policy().get_delay_in_ms() / 1000) venv\Lib\site-packages\redis\auth\token_manager.py: if self._retries < self._config.get_retry_policy().get_max_attempts(): venv\Lib\site-packages\redis\auth\token_manager.py: self._config.get_retry_policy().get_delay_in_ms() / 1000 venv\Lib\site-packages\redis\auth\token_manager.py: - self._config.get_lower_refresh_bound_millis() venv\Lib\site-packages\redis\auth\token_manager.py: token_ttl * self._config.get_expiration_refresh_ratio() venv\Lib\site-packages\redis\cache.py:class CacheConfigurationInterface(ABC): venv\Lib\site-packages\redis\cache.py: def config(self) -> CacheConfigurationInterface: venv\Lib\site-packages\redis\cache.py: cache_config: CacheConfigurationInterface, venv\Lib\site-packages\redis\cache.py: self._cache_config = cache_config venv\Lib\site-packages\redis\cache.py: self._eviction_policy = self._cache_config.get_eviction_policy().value() venv\Lib\site-packages\redis\cache.py: def config(self) -> CacheConfigurationInterface: venv\Lib\site-packages\redis\cache.py: return self._cache_config venv\Lib\site-packages\redis\cache.py: if self._cache_config.is_exceeds_max_size(len(self._cache)): venv\Lib\site-packages\redis\cache.py: return self._cache_config.is_allowed_to_cache(key.command) venv\Lib\site-packages\redis\cache.py:class CacheConfig(CacheConfigurationInterface): venv\Lib\site-packages\redis\cache.py: def __init__(self, cache_config: Optional[CacheConfig] = None): venv\Lib\site-packages\redis\cache.py: self._config = cache_config venv\Lib\site-packages\redis\cache.py: if self._config is None: venv\Lib\site-packages\redis\cache.py: self._config = CacheConfig() venv\Lib\site-packages\redis\cache.py: cache_class = self._config.get_cache_class() venv\Lib\site-packages\redis\cache.py: return cache_class(cache_config=self._config) venv\Lib\site-packages\redis\client.py:from redis.cache import CacheConfig, CacheInterface venv\Lib\site-packages\redis\client.py: MaintNotificationsConfig, venv\Lib\site-packages\redis\client.py: configuration, an instance will either use a ConnectionPool, or venv\Lib\site-packages\redis\client.py: Return a Redis client object configured from the given URL venv\Lib\site-packages\redis\client.py: cache_config: Optional[CacheConfig] = None, venv\Lib\site-packages\redis\client.py: maint_notifications_config: Optional[MaintNotificationsConfig] = None, venv\Lib\site-packages\redis\client.py: 2. Define a `Retry` object with configured 'supported_errors' and set venv\Lib\site-packages\redis\client.py: When 'connection_pool' is provided - the retry configuration of the venv\Lib\site-packages\redis\client.py: maint_notifications_config: venv\Lib\site-packages\redis\client.py: configuration the pool to support maintenance notifications - see venv\Lib\site-packages\redis\client.py: redis.maint_notifications.MaintNotificationsConfig for details. venv\Lib\site-packages\redis\client.py: if (cache_config or cache) and protocol in [3, "3"]: venv\Lib\site-packages\redis\client.py: "cache_config": cache_config, venv\Lib\site-packages\redis\client.py: maint_notifications_config and maint_notifications_config.enabled venv\Lib\site-packages\redis\client.py: if maint_notifications_config: venv\Lib\site-packages\redis\client.py: "maint_notifications_config": maint_notifications_config, venv\Lib\site-packages\redis\client.py: if (cache_config or cache) and self.connection_pool.get_protocol() not in [venv\Lib\site-packages\redis\cluster.py:from redis.cache import CacheConfig, CacheFactory, CacheFactoryInterface, CacheInterface venv\Lib\site-packages\redis\cluster.py:from redis.maint_notifications import MaintNotificationsConfig venv\Lib\site-packages\redis\cluster.py: "cache_config", venv\Lib\site-packages\redis\cluster.py: "CONFIG SET", venv\Lib\site-packages\redis\cluster.py: "CONFIG REWRITE", venv\Lib\site-packages\redis\cluster.py: "CONFIG RESETSTAT", venv\Lib\site-packages\redis\cluster.py: "CLUSTER SET-CONFIG-EPOCH", venv\Lib\site-packages\redis\cluster.py: "CONFIG GET", venv\Lib\site-packages\redis\cluster.py: "FT.CONFIG", venv\Lib\site-packages\redis\cluster.py: "CONFIG SET", venv\Lib\site-packages\redis\cluster.py: "CONFIG REWRITE", venv\Lib\site-packages\redis\cluster.py: "CONFIG RESETSTAT",

venv\Lib\site-packages\redis\cluster.py: Return a Redis client object configured from the given URL venv\Lib\site-packages\redis\cluster.py: reason="Please configure the 'load_balancing_strategy' instead", venv\Lib\site-packages\redis\cluster.py: reason="Please configure the 'retry' object instead", venv\Lib\site-packages\redis\cluster.py: cache_config: Optional[CacheConfig] = None, venv\Lib\site-packages\redis\cluster.py: <https://redis.io/topics/cluster-tutorial#redis-cluster-configuration-parameters> venv\Lib\site-packages\redis\cluster.py: @deprecated - Please configure the 'retry' object instead venv\Lib\site-packages\redis\cluster.py: # and there we provide retry configuration without retries allowed. venv\Lib\site-packages\redis\cluster.py: if (cache_config or cache) and protocol not in [3, "3"]: venv\Lib\site-packages\redis\cluster.py: cache_config=cache_config, venv\Lib\site-packages\redis\cluster.py: # Sending READONLY command to server to configure connection as venv\Lib\site-packages\redis\cluster.py: # rc.cluster_save_config(rc.get_primaries()) venv\Lib\site-packages\redis\cluster.py: config option "self.retry" which defaults to 3 unless manually venv\Lib\site-packages\redis\cluster.py: configured. venv\Lib\site-packages\redis\cluster.py: cache_config: Optional[CacheConfig] = None, venv\Lib\site-packages\redis\cluster.py: self._cache_config = cache_config venv\Lib\site-packages\redis\cluster.py: # We are configuring the connection pool not to retry venv\Lib\site-packages\redis\cluster.py: node_retry_config = Retry(venv\Lib\site-packages\redis\cluster.py: {"maint_notifications_config": MaintNotificationsConfig(enabled=False)} venv\Lib\site-packages\redis\cluster.py: kwargs.update({"retry": node_retry_config}) venv\Lib\site-packages\redis\cluster.py: retry=node_retry_config, venv\Lib\site-packages\redis\cluster.py: if self._cache is None and self._cache_config is not None: venv\Lib\site-packages\redis\cluster.py: self._cache = CacheFactory(self._cache_config).get_cache() venv\Lib\site-packages\redis\cluster.py: reason="Please configure the 'retry' object instead", venv\Lib\site-packages\redis\cluster.py: "CONFIG GET", venv\Lib\site-packages\redis\cluster.py: "CONFIG RESETSTAT", venv\Lib\site-packages\redis\cluster.py: "CONFIG REWRITE", venv\Lib\site-packages\redis\cluster.py: "CONFIG SET", venv\Lib\site-packages\redis\cluster.py: "CONFIG", venv\Lib\site-packages\redis\cluster.py: the retries in config option "self.retry" venv\Lib\site-packages\redis\cluster.py: which defaults to 3 unless manually configured. venv\Lib\site-packages\redis\cluster.py: # rc.cluster_save_config(rc.get_primaries()) venv\Lib\site-packages\redis\commands\cluster.py: Get Cluster config for the node. venv\Lib\site-packages\redis\commands\cluster.py: Reconfigure a node as a slave of the specified master node venv\Lib\site-packages\redis\commands\cluster.py: def cluster_save_config(venv\Lib\site-packages\redis\commands\cluster.py: For more information see <https://redis.io/commands/cluster-saveconfig> venv\Lib\site-packages\redis\commands\cluster.py: return self.execute_command("CLUSTER SAVECONFIG", target_nodes=target_nodes) venv\Lib\site-packages\redis\commands\cluster.py: def cluster_set_config_epoch(venv\Lib\site-packages\redis\commands\cluster.py: Set the configuration epoch in a new node venv\Lib\site-packages\redis\commands\cluster.py: For more information see <https://redis.io/commands/cluster-set-config-epoch> venv\Lib\site-packages\redis\commands\cluster.py: "CLUSTER SET-CONFIG-EPOCH", epoch, target_nodes=target_nodes venv\Lib\site-packages\redis\commands\core.py: Load ACL rules from the configured `aclfile`. venv\Lib\site-packages\redis\commands\core.py: Note that the server must be configured with the `aclfile` venv\Lib\site-packages\redis\commands\core.py: Save ACL rules to the configured `aclfile`. venv\Lib\site-packages\redis\commands\core.py: Note that the server must be configured with the `aclfile` venv\Lib\site-packages\redis\commands\core.py: def config_get(venv\Lib\site-packages\redis\commands\core.py: Return a dictionary of configuration based on the `pattern` venv\Lib\site-packages\redis\commands\core.py: For more information, see <https://redis.io/commands/config-get> venv\Lib\site-packages\redis\commands\core.py: return self.execute_command("CONFIG GET", pattern, *args, **kwargs) venv\Lib\site-packages\redis\commands\core.py: def config_set(venv\Lib\site-packages\redis\commands\core.py: """Set config item `name` with value `value` venv\Lib\site-packages\redis\commands\core.py: For more information, see <https://redis.io/commands/config-set> venv\Lib\site-packages\redis\commands\core.py: return self.execute_command("CONFIG SET", name, value, *args, **kwargs) venv\Lib\site-packages\redis\commands\core.py: def config_resetstat(self, **kwargs) -> ResponseT: venv\Lib\site-packages\redis\commands\core.py: For more information, see <https://redis.io/commands/config-resetstat> venv\Lib\site-packages\redis\commands\core.py: return self.execute_command("CONFIG RESETSTAT", **kwargs) venv\Lib\site-packages\redis\commands\core.py: def config_rewrite(self, **kwargs) -> ResponseT: venv\Lib\site-packages\redis\commands\core.py: Rewrite config file with the minimal change to reflect running config. venv\Lib\site-packages\redis\commands\core.py: For more information, see <https://redis.io/commands/config-rewrite> venv\Lib\site-packages\redis\commands\core.py: return

```

self.execute_command("CONFIG REWRITE", **kwargs) venv\Lib\site-packages\redis\commands\core.py: """Shutdown the Redis server. If Redis has
persistence configured, venv\Lib\site-packages\redis\commands\core.py: `save will force a DB saving operation even if no save points are
configured. venv\Lib\site-packages\redis\commands\core.py: are configured. venv\Lib\site-packages\redis\commands\core.py: """Shutdown the Redis
server. If Redis has persistence configured, venv\Lib\site-packages\redis\commands\core.py: configured. If the "nosave" option is set, no data flush will be
venv\Lib\site-packages\redis\commands\core.py: Loads a module from a dynamic library at runtime with configuration directives. venv\Lib\site-
packages\redis\commands\core.py: pieces.append("CONFIG") venv\Lib\site-packages\redis\commands\search\commands.py: HybridPostProcessingConfig,
venv\Lib\site-packages\redis\commands\search\commands.py: CONFIG_CMD = "FT.CONFIG" venv\Lib\site-packages\redis\commands\search\commands.py:
def _parse_config_get(self, res, **kwargs): venv\Lib\site-packages\redis\commands\search\commands.py: post_processing:
Optional[HybridPostProcessingConfig] = None, venv\Lib\site-packages\redis\commands\search\commands.py: - **post_processing**:
HybridPostProcessingConfig object venv\Lib\site-packages\redis\commands\search\commands.py: Contains the post processing configuration venv\Lib\site-
packages\redis\commands\search\commands.py: - **cursor**: HybridCursorQuery object - contains the cursor configuration venv\Lib\site-
packages\redis\commands\search\commands.py: reason="deprecated since Redis 8.0, call config_set from core module instead", venv\Lib\site-
packages\redis\commands\search\commands.py: def config_set(self, option: str, value: str) -> bool: venv\Lib\site-
packages\redis\commands\search\commands.py: """Set runtime configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: -
**option**: the name of the configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: - **value**: a value for the configuration
option. venv\Lib\site-packages\redis\commands\search\commands.py: For more information see FT.CONFIG SET
<https://redis.io/commands/ft.config-set>_. venv\Lib\site-packages\redis\commands\search\commands.py: cmd = [CONFIG_CMD, "SET", option,
value] venv\Lib\site-packages\redis\commands\search\commands.py: reason="deprecated since Redis 8.0, call config_get from core module instead",
venv\Lib\site-packages\redis\commands\search\commands.py: def config_get(self, option: str) -> str: venv\Lib\site-
packages\redis\commands\search\commands.py: """Get runtime configuration option value. venv\Lib\site-packages\redis\commands\search\commands.py: -
**option**: the name of the configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: For more information see FT.CONFIG GET
<https://redis.io/commands/ft.config-get>_. venv\Lib\site-packages\redis\commands\search\commands.py: cmd = [CONFIG_CMD, "GET", option]
venv\Lib\site-packages\redis\commands\search\commands.py: return self._parse_results(CONFIG_CMD, res) venv\Lib\site-
packages\redis\commands\search\commands.py: post_processing: Optional[HybridPostProcessingConfig] = None, venv\Lib\site-
packages\redis\commands\search\commands.py: - **post_processing**: HybridPostProcessingConfig object venv\Lib\site-
packages\redis\commands\search\commands.py: Contains the post processing configuration venv\Lib\site-packages\redis\commands\search\commands.py: -
**cursor**: HybridCursorQuery object - contains the cursor configuration venv\Lib\site-packages\redis\commands\search\commands.py: reason="deprecated
since Redis 8.0, call config_set from core module instead", venv\Lib\site-packages\redis\commands\search\commands.py: async def config_set(self, option: str,
value: str) -> bool: venv\Lib\site-packages\redis\commands\search\commands.py: """Set runtime configuration option. venv\Lib\site-
packages\redis\commands\search\commands.py: - **option**: the name of the configuration option. venv\Lib\site-
packages\redis\commands\search\commands.py: - **value**: a value for the configuration option. venv\Lib\site-
packages\redis\commands\search\commands.py: For more information see FT.CONFIG SET <https://redis.io/commands/ft.config-set>_.
venv\Lib\site-packages\redis\commands\search\commands.py: cmd = [CONFIG_CMD, "SET", option, value] venv\Lib\site-
packages\redis\commands\search\commands.py: reason="deprecated since Redis 8.0, call config_get from core module instead", venv\Lib\site-
packages\redis\commands\search\commands.py: async def config_get(self, option: str) -> str: venv\Lib\site-packages\redis\commands\search\commands.py:
"""Get runtime configuration option value. venv\Lib\site-packages\redis\commands\search\commands.py: - **option**: the name of the configuration option.
venv\Lib\site-packages\redis\commands\search\commands.py: For more information see FT.CONFIG GET <https://redis.io/commands/ft.config-get>_.

```

venv\Lib\site-packages\redis\commands\search\commands.py: cmd = [CONFIG_CMD, "GET", option] venv\Lib\site-packages\redis\commands\search\commands.py: return self._parse_results(CONFIG_CMD, res) venv\Lib\site-packages\redis\commands\search\hybrid_query.py: class HybridPostProcessingConfig: venv\Lib\site-packages\redis\commands\search\hybrid_query.py: Create a new hybrid post processing configuration object. venv\Lib\site-packages\redis\commands\search__init__.py: CONFIG_CMD, venv\Lib\site-packages\redis\commands\search__init__.py: CONFIG_CMD: self._parse_config_get, venv\Lib\site-packages\redis\commands\sentinel.py: configuration will be published so that the other Sentinels will venv\Lib\site-packages\redis\commands\sentinel.py: update their configurations). venv\Lib\site-packages\redis\commands\sentinel.py: Check if the current Sentinel configuration is able to reach the venv\Lib\site-packages\redis\commands\sentinel.py: def sentinel_flushconfig(self): venv\Lib\site-packages\redis\commands\sentinel.py: Force Sentinel to rewrite its configuration on disk, including the venv\Lib\site-packages\redis\commands\sentinel.py: Normally Sentinel rewrites the configuration every time something venv\Lib\site-packages\redis\commands\sentinel.py: However sometimes it is possible that the configuration file is lost venv\Lib\site-packages\redis\commands\sentinel.py: configuration managers. In those cases a way to to force Sentinel to venv\Lib\site-packages\redis\commands\sentinel.py: rewrite the configuration file is handy. venv\Lib\site-packages\redis\commands\sentinel.py: This command works even if the previous configuration file is venv\Lib\site-packages\redis\commands\sentinel.py: return self.execute_command("SENTINEL FLUSHCONFIG") venv\Lib\site-packages\redis\commands\timeseries\info.py: https://redis.io/docs/latest/develop/data-types/timeseries/configuration/#duplicate_policy venv\Lib\site-packages\redis\connection.py: MaintNotificationsConfig, venv\Lib\site-packages\redis\connection.py: # the command name, e.g., 'CONFIG GET'. The Redis server expects these venv\Lib\site-packages\redis\connection.py: maint_notifications_config: Optional[MaintNotificationsConfig], venv\Lib\site-packages\redis\connection.py: maint_notifications_config (MaintNotificationsConfig): The configuration for maintenance notifications. venv\Lib\site-packages\redis\connection.py: self.maint_notifications_config = maint_notifications_config venv\Lib\site-packages\redis\connection.py: self._configure_maintenance_notifications(venv\Lib\site-packages\redis\connection.py: def _configure_maintenance_notifications(venv\Lib\site-packages\redis\connection.py: not self.maint_notifications_config venv\Lib\site-packages\redis\connection.py: or not self.maint_notifications_config.enabled venv\Lib\site-packages\redis\connection.py: "To configure maintenance notifications, a parser must be provided!" venv\Lib\site-packages\redis\connection.py: MaintNotificationsConnectionHandler(self, self.maint_notifications_config) venv\Lib\site-packages\redis\connection.py: self, maint_notifications_pool_handler.config venv\Lib\site-packages\redis\connection.py: self._maint_notifications_connection_handler.config = (venv\Lib\site-packages\redis\connection.py: maint_notifications_pool_handler.config venv\Lib\site-packages\redis\connection.py: # When the maint_notifications_config enabled mode is "auto", venv\Lib\site-packages\redis\connection.py: and self.maint_notifications_config venv\Lib\site-packages\redis\connection.py: and self.maint_notifications_config.enabled venv\Lib\site-packages\redis\connection.py: maint_notifications_config=self.maint_notifications_config, venv\Lib\site-packages\redis\connection.py: self, maint_notifications_config: MaintNotificationsConfig, check_health=True venv\Lib\site-packages\redis\connection.py: endpoint_type = maint_notifications_config.get_endpoint_type(host, self) venv\Lib\site-packages\redis\connection.py: and maint_notifications_config.enabled == "auto" venv\Lib\site-packages\redis\connection.py: maint_notifications_config: Optional[MaintNotificationsConfig] = None, venv\Lib\site-packages\redis\connection.py: maint_notifications_config, venv\Lib\site-packages\redis\connection.py: # if enabled in the configuration venv\Lib\site-packages\redis\connection.py: self._conn.maint_notifications_config, venv\Lib\site-packages\redis\connection.py: maint_notifications_config: Optional[MaintNotificationsConfig] = None, venv\Lib\site-packages\redis\connection.py: if maint_notifications_config is None and is_protocol_supported: venv\Lib\site-packages\redis\connection.py: maint_notifications_config = MaintNotificationsConfig() venv\Lib\site-packages\redis\connection.py: if maint_notifications_config and maint_notifications_config.enabled: venv\Lib\site-packages\redis\connection.py: self, maint_notifications_config venv\Lib\site-packages\redis\connection.py: The maintenance notifications config is stored in the pool handler. venv\Lib\site-packages\redis\connection.py: maint_notifications_config = (venv\Lib\site-packages\redis\connection.py: self._maint_notifications_pool_handler.config venv\Lib\site-packages\redis\connection.py: return maint_notifications_config and maint_notifications_config.enabled venv\Lib\site-packages\redis\connection.py: def

```

update_maint_notifications_config( venv\Lib\site-packages\redis\connection.py: self, maint_notifications_config: MaintNotificationsConfig venv\Lib\site-
packages\redis\connection.py: Updates the maintenance notifications configuration. venv\Lib\site-packages\redis\connection.py: and not
maint_notifications_config.enabled venv\Lib\site-packages\redis\connection.py: self, maint_notifications_config venv\Lib\site-packages\redis\connection.py:
self. _maint_notifications_pool_handler.config = maint_notifications_config venv\Lib\site-packages\redis\connection.py:
self. _update_maint_notifications_configs_for_connections( venv\Lib\site-packages\redis\connection.py: "maint_notifications_config":
maint_notifications_pool_handler.config, venv\Lib\site-packages\redis\connection.py: # configured the original values yet venv\Lib\site-
packages\redis\connection.py: def _update_maint_notifications_configs_for_connections( venv\Lib\site-packages\redis\connection.py: """Update the
maintenance notifications config for all connections in the pool.""" venv\Lib\site-packages\redis\connection.py: conn.maint_notifications_config = (
venv\Lib\site-packages\redis\connection.py: maint_notifications_pool_handler.config venv\Lib\site-packages\redis\connection.py:
conn.maint_notifications_config = ( venv\Lib\site-packages\redis\connection.py: maint_notifications_pool_handler.config venv\Lib\site-
packages\redis\connection.py: "connected_address", "configured_address", "notification_hash" venv\Lib\site-packages\redis\connection.py: elif
matching_pattern == "configured_address": venv\Lib\site-packages\redis\connection.py: "connected_address", "configured_address", "notification_hash"
venv\Lib\site-packages\redis\connection.py: If `maint_notifications_config` is provided, the connection pool will support venv\Lib\site-
packages\redis\connection.py: If the `maint_notifications_config` is not provided but the protocol is 3, venv\Lib\site-
packages\redis\connection.py: Return a connection pool configured from the given URL. venv\Lib\site-packages\redis\connection.py:
maint_notifications_config: Optional[MaintNotificationsConfig] = None, venv\Lib\site-packages\redis\connection.py: if
connection_kwargs.get("cache_config") or connection_kwargs.get("cache"): venv\Lib\site-packages\redis\connection.py:
self. _connection_kwargs.get("cache_config") venv\Lib\site-packages\redis\connection.py: connection_kwargs.pop("cache_config", None) venv\Lib\site-
packages\redis\connection.py: maint_notifications_config=maint_notifications_config, venv\Lib\site-packages\redis\http\http_client.py: retry: Retry
configuration for failed requests. venv\Lib\site-packages\redis\http\http_client.py: # Load client certificate for mTLS if configured venv\Lib\site-
packages\redis\maint_notifications.py:class MaintNotificationsConfig: venv\Lib\site-packages\redis\maint_notifications.py: Configuration class for
maintenance notifications handling behaviour. Notifications are received through venv\Lib\site-packages\redis\maint_notifications.py: Initialize a new
MaintNotificationsConfig. venv\Lib\site-packages\redis\maint_notifications.py: If None, the endpoint type will be automatically determined based on the host
and TLS configuration. venv\Lib\site-packages\redis\maint_notifications.py: config: MaintNotificationsConfig, venv\Lib\site-
packages\redis\maint_notifications.py: self.config = config venv\Lib\site-packages\redis\maint_notifications.py: copy =
MaintNotificationsPoolHandler(self.pool, self.config) venv\Lib\site-packages\redis\maint_notifications.py: not self.config.proactive_reconnect venv\Lib\site-
packages\redis\maint_notifications.py: and not self.config.is_relaxed_timeouts_enabled() venv\Lib\site-packages\redis\maint_notifications.py:
self.config.proactive_reconnect venv\Lib\site-packages\redis\maint_notifications.py: or self.config.is_relaxed_timeouts_enabled() venv\Lib\site-
packages\redis\maint_notifications.py: relaxed_timeout=self.config.relaxed_timeout, venv\Lib\site-packages\redis\maint_notifications.py: if
self.config.proactive_reconnect: venv\Lib\site-packages\redis\maint_notifications.py: # Update config for new connections: venv\Lib\site-
packages\redis\maint_notifications.py: if self.config.is_relaxed_timeouts_enabled(): venv\Lib\site-packages\redis\maint_notifications.py: "socket_timeout":
self.config.relaxed_timeout, venv\Lib\site-packages\redis\maint_notifications.py: "socket_connect_timeout": self.config.relaxed_timeout, venv\Lib\site-
packages\redis\maint_notifications.py: reset_relaxed_timeout = self.config.is_relaxed_timeouts_enabled() venv\Lib\site-packages\redis\maint_notifications.py:
reset_host_address = self.config.proactive_reconnect venv\Lib\site-packages\redis\maint_notifications.py: config: MaintNotificationsConfig, venv\Lib\site-
packages\redis\maint_notifications.py: self.config = config venv\Lib\site-packages\redis\maint_notifications.py: or not
self.config.is_relaxed_timeouts_enabled() venv\Lib\site-packages\redis\maint_notifications.py: tmp_relaxed_timeout=self.config.relaxed_timeout
venv\Lib\site-packages\redis\maint_notifications.py: self.connection.update_current_socket_timeout(self.config.relaxed_timeout) venv\Lib\site-

```

packages\redis\maint_notifications.py: or not self.config.is_relaxed_timeouts_enabled() venv\Lib\site-packages\redis\multidb\client.py: from
 redis.multidb.config import DEFAULT_GRACE_PERIOD, MultiDbConfig venv\Lib\site-packages\redis\multidb\client.py: def __init__(self, config:
 MultiDbConfig): venv\Lib\site-packages\redis\multidb\client.py: self._databases = config.databases() venv\Lib\site-packages\redis\multidb\client.py:
 config.default_health_checks() venv\Lib\site-packages\redis\multidb\client.py: if not config.health_checks venv\Lib\site-packages\redis\multidb\client.py: else
 config.health_checks venv\Lib\site-packages\redis\multidb\client.py: self._health_check_interval = config.health_check_interval venv\Lib\site-
 packages\redis\multidb\client.py: self._health_check_policy: HealthCheckPolicy = config.health_check_policy.value(venv\Lib\site-
 packages\redis\multidb\client.py: config.health_check_probes, config.health_check_probes_delay venv\Lib\site-packages\redis\multidb\client.py:
 config.default_failure_detectors() venv\Lib\site-packages\redis\multidb\client.py: if not config.failure_detectors venv\Lib\site-packages\redis\multidb\client.py:
 else config.failure_detectors venv\Lib\site-packages\redis\multidb\client.py: config.default_failover_strategy() venv\Lib\site-packages\redis\multidb\client.py:
 if config.failover_strategy is None venv\Lib\site-packages\redis\multidb\client.py: else config.failover_strategy venv\Lib\site-packages\redis\multidb\client.py:
 self._auto_fallback_interval = config.auto_fallback_interval venv\Lib\site-packages\redis\multidb\client.py: self._event_dispatcher = config.event_dispatcher
 venv\Lib\site-packages\redis\multidb\client.py: self._command_retry = config.command_retry venv\Lib\site-packages\redis\multidb\client.py:
 failover_attempts=config.failover_attempts, venv\Lib\site-packages\redis\multidb\client.py: failover_delay=config.failover_delay, venv\Lib\site-
 packages\redis\multidb\client.py: self._config = config venv\Lib\site-packages\redis\multidb\command_executor.py: from redis.multidb.config import
 DEFAULT_AUTO_FALLBACK_INTERVAL venv\Lib\site-packages\redis\multidb\config.py: class DatabaseConfig: venv\Lib\site-
 packages\redis\multidb\config.py: Dataclass representing the configuration for a database connection. venv\Lib\site-packages\redis\multidb\config.py: This
 class is used to store configuration settings for a database connection, venv\Lib\site-packages\redis\multidb\config.py: attributes and allows for the creation of
 customized configurations for various venv\Lib\site-packages\redis\multidb\config.py: from_pool (Optional[ConnectionPool]): A pre-configured connection
 pool to use. venv\Lib\site-packages\redis\multidb\config.py: class MultiDbConfig: venv\Lib\site-packages\redis\multidb\config.py: Configuration class for
 managing multiple database connections in a resilient and fail-safe manner. venv\Lib\site-packages\redis\multidb\config.py: databases_config: A list of
 database configurations. venv\Lib\site-packages\redis\multidb\config.py: Retrieves a collection of database clients managed by weighted configurations.
 venv\Lib\site-packages\redis\multidb\config.py: Initializes database clients based on the provided configuration and removes venv\Lib\site-
 packages\redis\multidb\config.py: with defined retry and backoff configurations. venv\Lib\site-packages\redis\multidb\config.py: databases_config:
 List[DatabaseConfig] venv\Lib\site-packages\redis\multidb\config.py: for database_config in self.databases_config: venv\Lib\site-
 packages\redis\multidb\config.py: database_config.client_kwargs.update(venv\Lib\site-packages\redis\multidb\config.py: if database_config.from_url:
 venv\Lib\site-packages\redis\multidb\config.py: database_config.from_url, **database_config.client_kwargs venv\Lib\site-packages\redis\multidb\config.py:
 elif database_config.from_pool: venv\Lib\site-packages\redis\multidb\config.py: database_config.from_pool.set_retry(venv\Lib\site-
 packages\redis\multidb\config.py: connection_pool=database_config.from_pool venv\Lib\site-packages\redis\multidb\config.py: client =
 self.client_class(**database_config.client_kwargs) venv\Lib\site-packages\redis\multidb\config.py: database_config.default_circuit_breaker() venv\Lib\site-
 packages\redis\multidb\config.py: if database_config.circuit is None venv\Lib\site-packages\redis\multidb\config.py: else database_config.circuit venv\Lib\site-
 packages\redis\multidb\config.py: weight=database_config.weight, venv\Lib\site-packages\redis\multidb\config.py:
 health_check_url=database_config.health_check_url, venv\Lib\site-packages\redis\multidb\config.py: database_config.weight, venv\Lib\site-
 packages\redis\multidb\healthcheck.py: "Database health check url is not set. Please check DatabaseConfig for the current database." venv\Lib\site-
 packages\redis_parsers\base.py: # user send an AUTH cmd to a server without authorization configured venv\Lib\site-packages\redis_parsers\base.py:
 "configured for the default user. Are you sure " venv\Lib\site-packages\redis_parsers\base.py: "your configuration is correct?": AuthenticationError,
 venv\Lib\site-packages\redis_parsers\helpers.py: "config-epoch": int, venv\Lib\site-packages\redis_parsers\helpers.py: def parse_config_get(response,
 **options): venv\Lib\site-packages\redis_parsers\helpers.py: "CLUSTER SAVECONFIG": bool_ok, venv\Lib\site-packages\redis_parsers\helpers.py:

```

"CLUSTER SET-CONFIG-EPOCH": bool_ok, venv\Lib\site-packages\redis\_parsers\helpers.py: "CONFIG RESETSTAT": bool_ok, venv\Lib\site-
packages\redis\_parsers\helpers.py: "CONFIG SET": bool_ok, venv\Lib\site-packages\redis\_parsers\helpers.py: "SENTINEL FLUSHCONFIG": bool_ok,
venv\Lib\site-packages\redis\_parsers\helpers.py: "CONFIG GET": parse_config_get, venv\Lib\site-packages\redis\_parsers\helpers.py: "CONFIG GET":
lambda r: { venv\Lib\site-packages\rest_framework\apps.py:from django.apps import AppConfig venv\Lib\site-packages\rest_framework\apps.py:class
RestFrameworkConfig(AppConfig): venv\Lib\site-packages\rest_framework\authtoken\apps.py:from django.apps import AppConfig venv\Lib\site-
packages\rest_framework\authtoken\apps.py:class AuthTokenConfig(AppConfig): venv\Lib\site-packages\rest_framework\checks.py:def
pagination_system_check(app_configs, **kwargs): venv\Lib\site-packages\rest_framework\compat.py: extension_configs = { venv\Lib\site-
packages\rest_framework\compat.py: extensions=extensions, extension_configs=extension_configs venv\Lib\site-packages\rest_framework\fields.py: Quantize
the decimal value to the configured precision. venv\Lib\site-packages\rest_framework\filters.py:from django.core.exceptions import FieldDoesNotExist,
ImproperlyConfigured venv\Lib\site-packages\rest_framework\filters.py: raise ImproperlyConfigured(msg % self.__class__.__name__) venv\Lib\site-
packages\rest_framework\generics.py: method if you want to apply the configured filtering backend to the venv\Lib\site-
packages\rest_framework\management\commands\generateschema.py: help = "Generates configured API schema for project." venv\Lib\site-
packages\rest_framework\pagination.py: page object, or None if pagination is not configured for this view. venv\Lib\site-
packages\rest_framework\relations.py:from django.core.exceptions import ImproperlyConfigured, ObjectDoesNotExist venv\Lib\site-
packages\rest_framework\relations.py: attributes are not configured to correctly match the URL conf. venv\Lib\site-packages\rest_framework\relations.py: #
Return the hyperlink, or error if incorrectly configured. venv\Lib\site-packages\rest_framework\relations.py: 'model in your API, or incorrectly configured the '
venv\Lib\site-packages\rest_framework\relations.py: raise ImproperlyConfigured(msg % self.view_name) venv\Lib\site-
packages\rest_framework\renderers.py:from django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\renderers.py: raise
ImproperlyConfigured( venv\Lib\site-packages\rest_framework\routers.py:from django.core.exceptions import ImproperlyConfigured venv\Lib\site-
packages\rest_framework\routers.py: raise ImproperlyConfigured(msg) venv\Lib\site-packages\rest_framework\routers.py: raise ImproperlyConfigured(msg %
', '.join(not_allowed)) venv\Lib\site-packages\rest_framework\serializers.py:from django.core.exceptions import FieldDoesNotExist, ImproperlyConfigured
venv\Lib\site-packages\rest_framework\serializers.py: raise ImproperlyConfigured( venv\Lib\site-packages\rest_framework\settings.py: # View configuration
venv\Lib\site-packages\rest_framework\test.py:from django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\test.py:
raise ImproperlyConfigured('requests must be installed in order to use RequestsClient.') venv\Lib\site-packages\rest_framework\test.py: raise
ImproperlyConfigured('coreapi must be installed in order to use CoreAPIClient.') venv\Lib\site-packages\rest_framework\throttling.py:from
django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\throttling.py: raise ImproperlyConfigured(msg) venv\Lib\site-
packages\rest_framework\throttling.py: raise ImproperlyConfigured(msg) venv\Lib\site-packages\rest_framework\views.py: # Filter out None values which
may happen in case of config / rate venv\Lib\site-packages\sqlparse\lexer.py: """"The Lexer supports configurable syntax. venv\Lib\site-
packages\sqlparse\lexer.py: """"Clear all syntax configurations. venv\Lib\site-packages\sqlparse\lexer.py: Useful if you want to load a reduced set of syntax
configurations. venv\Lib\site-packages\tutorial\asgi.py:ASGI config for tutorial project. venv\Lib\site-packages\tutorial\quickstart\apps.py:class
QuickstartConfig(AppConfig): venv\Lib\site-packages\tutorial\wsgi.py:WSGI config for tutorial project. (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> python manage.py makemigrations Migrations for 'chat': chat\migrations\0001_initial.py + Create model Room + Create
model Message (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py migrate Operations to perform: Apply all
migrations: admin, auth, chat, contenttypes, sessions Running migrations: Applying contenttypes.0001_initial... OK Applying auth.0001_initial... OK Applying
admin.0001_initial... OK Applying admin.0002_logentry_remove_auto_add... OK Applying admin.0003_logentry_add_action_flag_choices... OK Applying
contenttypes.0002_remove_content_type_name... OK Applying auth.0002_alter_permission_name_max_length... OK Applying
auth.0003_alter_user_email_max_length... OK Applying auth.0004_alter_user_username_opts... OK Applying auth.0005_alter_user_last_login_null... OK

```





[illegible]

information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/> Not Found: / [07/Feb/2026 17:03:24] "GET / HTTP/1.1" 404 2305 Not Found: / [07/Feb/2026 17:03:25] "GET / HTTP/1.1" 404 2305 Not Found: / [07/Feb/2026 17:03:27] "GET / HTTP/1.1" 404 2305 [07/Feb/2026 17:04:21] "GET /admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:04:22] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 4252 [07/Feb/2026 17:04:22] "GET /static/admin/css/nav_sidebar.css HTTP/1.1" 200 2810 [07/Feb/2026 17:04:22] "GET /static/admin/css/base.css HTTP/1.1" 200 22120 [07/Feb/2026 17:04:22] "GET /static/admin/css/login.css HTTP/1.1" 200 951 [07/Feb/2026 17:04:22] "GET /static/admin/css/dark_mode.css HTTP/1.1" 200 2808 [07/Feb/2026 17:04:22] "GET /static/admin/css/responsive.css HTTP/1.1" 200 16565 [07/Feb/2026 17:04:22] "GET /static/admin/js/theme.js HTTP/1.1" 200 1653 [07/Feb/2026 17:04:22] "GET /static/admin/js/nav_sidebar.js HTTP/1.1" 200 3063 Not Found: /favicon.ico [07/Feb/2026 17:04:22] "GET /favicon.ico HTTP/1.1" 404 2356 Not Found: /api/ [07/Feb/2026 17:04:37] "GET /api/ HTTP/1.1" 404 2739 Not Found: / [07/Feb/2026 17:04:48] "GET / HTTP/1.1" 404 2305 [07/Feb/2026 17:05:01] "GET /admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:05:01] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 4252 Not Found: /api/ [07/Feb/2026 17:05:12] "GET /api/ HTTP/1.1" 404 2739 Not Found: / [07/Feb/2026 17:24:49] "GET /api/rooms/ HTTP/1.1" 200 2 Not Found: /favicon.ico [07/Feb/2026 17:24:49] "GET /favicon.ico HTTP/1.1" 404 2356 (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> curl -X POST http://127.0.0.1:8000/api/rooms/ -H "Content-Type: application/json" -d '{"name":"general"}' >> ^C (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> curl -X POST http://127.0.0.1:8000/api/rooms/ -H "Content-Type: application/json" -d '{"name":"general"}' Invoke-WebRequest : Impossible de lier le paramètre «Headers». Impossible de convertir la valeur «Content-Type: application/json» du type «System.String» en type «System.Collections.IDictionary». Au caractère Ligne:1 : 50 + ... 127.0.0.1:8000/api/rooms/ -H "Content-Type: application/json" -d '{" ... + ~~~~~ + CategoryInfo : InvalidArgument : (:) [Invoke-WebRequest], ParameterBindingException + FullyQualifiedErrorId : CannotConvertArgumentNoMessage,Microsoft.PowerShell.Commands.InvokeWebRequestCommand (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario>

Tu es **sur la bonne voie** : tes migrations sont OK, Channels est installé, et `/api/rooms/` répond bien (200 avec []). 👍
Les “problèmes” que tu vois sont surtout des détails normaux + une commande `curl` qui n’est pas la bonne sous PowerShell.

1) Pourquoi / et `/api/` font 404 ?

Parce qu’on n’a **pas défini** de route pour `/` ni pour `/api/` (seulement `/api/rooms/` et `/api/rooms/<id>/messages/`).
Donc :

-  `/api/rooms/` → OK
-  `/admin/` → OK
-  `/` → 404 normal
-  `/api/` → 404 normal (sauf si tu ajoutes une vue “ApiRoot”)

Si tu veux, je te redonne le patch pour que `/api/` affiche un JSON “welcome”, mais ce n’est pas obligatoire.

2) L'erreur `import chat.routing` sans settings

Ça aussi c'est normal : si tu fais un import Django qui touche aux modèles, **il faut initialiser Django** (`DJANGO_SETTINGS_MODULE + django.setup()`).

Tu as fait :

```
python -c "import os; os.environ.setdefault('DJANGO_SETTINGS_MODULE','monsite.settings'); import django; django.setup(); import chat.routing; print('routing ok')"
```



Et ça marche. Donc parfait.

3) Ton vrai souci : la commande `curl` sous PowerShell

Sous Windows **PowerShell**, `curl` est souvent un alias de `Invoke-WebRequest`, donc les options `-H` et `-d` ne marchent pas comme sur Linux/Mac.

Option A — la bonne commande PowerShell (recommandée)

```
Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/" `
  -Method Post `
  -ContentType "application/json" `
  -Body '{"name":"general"}'
```

Option B — utiliser `curl.exe` (pas l'alias)

```
curl.exe -X POST "http://127.0.0.1:8000/api/rooms/" ^
  -H "Content-Type: application/json" ^
  -d "{\"name\":\"general\"}"
```

Option C — via l'admin Django

Tu peux aussi créer une Room “general” depuis :

/admin/ → Room → Add

4) Test WebSocket (très simple)

1. Lance le serveur :

```
py manage.py runserver
```

2. Ouvre `frontend/index.html`
3. Clique **Rejoindre**
4. Envoie un message

Si ça ne marche pas, ouvre la console du navigateur (F12 → Console) et regarde si tu as une erreur du style :

- “WebSocket connection failed”
 - 403 / 404 sur `/ws/chat/...`
-

5) Petit conseil important (dans ton cas)

Je vois que ton projet s’appelle **monsite** chez toi (dans la sortie : `using settings 'monsite.settings'`).
Ça marche tant que :

- `ASGI_APPLICATION = "monsite.asgi.application"`
- `config/asgi.py` correspond au bon nom de projet

👉 Si tu as copié/collé des fichiers “`config/...`” mais ton projet s’appelle “monsite”, il faut que les chemins soient cohérents.

Si tu veux, colle-moi juste :

- le nom exact de ton dossier projet Django (ex: `monsite/` ou `config/`)
- le contenu de `settings.py` (les 5 lignes autour de `ASGI_APPLICATION + INSTALLED_APPS`)
et je te dis exactement quoi ajuster en 10 secondes.

Chat en ligne (Django + Channels) REST: `http://127.0.0.1:8000/api/` • WS: `ws://127.0.0.1:8000/ws/chat/<room>/` system • 07/02/2026 17:31:01 Déconnecté
system • 07/02/2026 17:31:04 Déconnecté system • 07/02/2026 17:31:32 Déconnecté system • 07/02/2026 17:31:58 Déconnecté system • 07/02/2026 17:32:00

Déconnecté system • 07/02/2026 17:32:01 Déconnecté system • 07/02/2026 17:32:01 Déconnecté system • 07/02/2026 17:32:02 Déconnecté system • 07/02/2026 17:32:04

Ça indique que le navigateur **arrive à ouvrir puis ferme immédiatement** la connexion WebSocket (ou qu'elle est refusée/404 et donc "close" direct). Les 2 causes les plus fréquentes dans ton cas :

1. **Le serveur tourne en WSGI au lieu d'ASGI** (WS impossible)
2. **Ton `asgi.py/routing.py` n'est pas celui du projet actif** (tu as `monsite.settings` dans tes logs)

On va aller au plus efficace.

1) Vérifie que tu utilises bien ASGI (obligatoire pour WebSocket)

Dans la console où tu lances le serveur, lance **exactement** :

```
py manage.py runserver
```

Puis, **dans une autre console**, teste que ton ASGI est bien chargé :

```
py -c "import os; os.environ.setdefault('DJANGO_SETTINGS_MODULE','monsite.settings'); import monsite.asgi as a; print('ASGI OK', a.application)"
```

✅ Si ça affiche `ASGI OK ...` sans erreur, on est bien sur le bon fichier `asgi.py`.

2) Le point critique : ton projet s'appelle `monsite` (pas `config`)

Dans l'archive que je t'ai donnée, le projet s'appelait `config`.

Mais chez toi, Django dit : `using settings 'monsite.settings'`.

➡ Donc il faut que tes chemins soient cohérents :

A. Dans `monsite/settings.py`

Tu dois avoir :

```
ASGI_APPLICATION = "monsite.asgi.application"
```

B. Dans `monsite/asgi.py`

Tu dois avoir (version Channels) :

```
import os
from django.core.asgi import get_asgi_application
from channels.routing import ProtocolTypeRouter, URLRouter
from channels.auth import AuthMiddlewareStack
import chat.routing

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "monsite.settings")

django_asgi_app = get_asgi_application()

application = ProtocolTypeRouter({
    "http": django_asgi_app,
    "websocket": AuthMiddlewareStack(
        URLRouter(chat.routing.websocket_urlpatterns)
    ),
})
```

C. Dans `chat/routing.py`

Tu dois avoir :

```
from django.urls import re_path
from .consumers import ChatConsumer

websocket_urlpatterns = [
    re_path(r"ws/chat/(?P<room_name>[^/]+)/$", ChatConsumer.as_asgi()),
]
```

3) Vérifie que Django reçoit bien la requête WS

Quand tu cliques **Rejoindre**, regarde la console Django.

Tu devrais voir une ligne du style (varie selon config) :

- `WebSocket CONNECT /ws/chat/general/ ...`

Si tu ne vois **rien**, c'est que ton serveur ne gère pas WS (souvent WSGI / mauvais asgi.py).



4) Test direct sans le navigateur (pour être sûr)

Installe `websockets` :

```
pip install websockets
```

Puis teste :

```
py -c "import asyncio, websockets;
async def t():
    async with websockets.connect('ws://127.0.0.1:8000/ws/chat/general/') as ws:
        await ws.send('{\"author\": \"test\", \"content\": \"hello\"}')
        print(await ws.recv())
    asyncio.run(t())"
```

-  Si tu reçois un JSON message → WS OK
 -  Si connexion refusée / 404 → problème routing/asgi
-

5) Le symptôme “Déconnecté” en boucle dans ton UI

Ça arrive si tu cliques plusieurs fois “Rejoindre” ou si la connexion ferme immédiatement.
Une fois le WS OK, tu verras d’abord “**Connecté au salon ...**” puis tu pourras envoyer.

Le plus probable, vu tes logs

👉 Corrige juste `ASGI_APPLICATION` + `monsite/asgi.py` comme ci-dessus et ça repart.

- `monsite/asgi.py`
- la ligne `ASGI_APPLICATION` de `monsite/settings.py`

[illegible]

Ton erreur :

```
AppRegistryNotReady: Apps aren't loaded yet.
```

arrive parce que `monsite/asgi.py` importe `chat.routing` trop tôt, avant que Django ait fini de charger les apps.

✅ **Fix simple (recommandé) : importer `chat.routing` après `get asgi application()`**

Ouvre `monsite/asgi.py` et remplace-le par ceci :

```
import os
from django.core.asgi import get_asgi_application
```

```
from channels.routing import ProtocolTypeRouter, URLRouter
from channels.auth import AuthMiddlewareStack

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "monsite.settings")

django_asgi_app = get_asgi_application()

# IMPORTANT: importer routing APRÈS avoir initialisé Django
import chat.routing

application = ProtocolTypeRouter({
    "http": django_asgi_app,
    "websocket": AuthMiddlewareStack(
        URLRouter(chat.routing.websocket_urlpatterns)
    ),
})
```

Ensuite, relance et teste

1. Stop le serveur (CTRL+C), puis relance :

```
py manage.py runserver
```

2. Refais ton test ASGI :

```
py -c "import os; os.environ.setdefault('DJANGO_SETTINGS_MODULE','monsite.settings'); import monsite.asgi as a; print('ASGI OK')"
```



Ça doit afficher ASGI OK sans erreur.

3. Ouvre ton frontend/index.html, clique **Rejoindre**.
Cette fois tu devrais voir :

- Connecté au salon "general"
et plus de déconnexions instantanées.

Pourquoi ça marche ?

`get_asgi_application()` initialise Django (apps registry).

Si tu importes `chat.routing` avant, ça charge tes modèles trop tôt → Apps pas prêtes → crash.

Si après ce fix tu vois encore “Déconnecté”, dis-moi ce que montre :

- la **console Django** au moment du clic “Rejoindre”
- la **console navigateur** (F12 → Console + Network → WS)

`index.html` comment le lancer dans django ?

Tu as 2 façons simples.

Option 1 (la plus simple) : ouvrir le fichier HTML

Tu **n’as pas besoin** de Django pour “servir” ce `index.html`.

- Lance Django : `py manage.py runserver`
- Puis ouvre `frontend/index.html` (double-clic) dans ton navigateur.

✅ Ça marche si ton `settings.py` a bien `CORS_ALLOW_ALL_ORIGINS = True` (c’est déjà le cas dans le projet).

Option 2 (propre) : servir `index.html` via Django (templates + static)

1) Déplace le fichier

Crée ces dossiers :

- `chat/templates/chat/index.html`
- `chat/static/chat/` (si tu as des CSS/JS séparés)

Mets ton `index.html` dans :

`chat/templates/chat/index.html`

2) Ajoute une vue Django

Dans `chat/views.py`, ajoute :

```
from django.shortcuts import render

def chat_page(request):
    return render(request, "chat/index.html")
```

3) Ajoute une URL

Dans `chat/urls.py`, ajoute une route :

```
from django.urls import path
from .views import RoomListCreateView, MessageListView, chat_page

urlpatterns = [
    path("", chat_page, name="chat-page"),

    path("rooms/", RoomListCreateView.as_view(), name="rooms"),
    path("rooms/<int:room_id>/messages/", MessageListView.as_view(), name="room-messages"),
]
```

⚠ Attention : ici `path("")` est **dans `/api/`** si tu gardes `path("api/", include("chat.urls"))`.

Donc tu auras la page à :

- `http://127.0.0.1:8000/api/` (si tu inclus `chat.urls` sous `/api/`)

Option 2bis (encore mieux) : page à `/` et API à `/api/`

C'est la structure la plus claire.

`monsite/urls.py`

```
from django.contrib import admin
from django.urls import path, include
```

```
from chat.views import chat_page

urlpatterns = [
    path("admin/", admin.site.urls),
    path("", chat_page),          # page web
    path("api/", include("chat.urls")) # API
]
```

chat/urls.py (API seulement)

```
from django.urls import path
from .views import RoomListCreateView, MessageListView

urlpatterns = [
    path("rooms/", RoomListCreateView.as_view(), name="rooms"),
    path("rooms/<int:room_id>/messages/", MessageListView.as_view(), name="room-messages"),
]
```




Résultat :

- **Page chat** : `http://127.0.0.1:8000/`
- **API** : `http://127.0.0.1:8000/api/rooms/`

Si tu me dis ce que tu préfères :

- **page à /** (recommandé)
ou
- **page dans /api/**
je te donne exactement les fichiers finaux “copier/coller” sans ambiguïté.

Dans `http://127.0.0.1:8000/api/rooms/` , dans "Données brutes" , je vois []

Parfait  [] sur `http://127.0.0.1:8000/api/rooms/` veut dire :

- l'API REST fonctionne
- il n'y a juste **aucun salon créé** pour l'instant

Créer un salon “general” (Windows PowerShell)

Comme tu es sur PowerShell, fais :

```
Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/" `
  -Method Post `
  -ContentType "application/json" `
  -Body '{"name":"general"}'
```

Ensuite, recharge `http://127.0.0.1:8000/api/rooms/` → tu devrais voir quelque chose comme :

```
[{"id":1,"name":"general"}]
```

Alternative rapide : via l’admin Django

1. Va sur `http://127.0.0.1:8000/admin/`
2. Crée une **Room** avec `name = general`

Et après ?

- Ouvre ton `index.html`
 - Mets `room = general`
 - Clique **Rejoindre**
 - Envoie un message
- Puis clique **Charger l’historique** : tu verras les messages via l’API.

Si tu me dis ce que tu vois côté console navigateur quand tu cliques “Rejoindre” (F12 → Console), je te confirme que ton WebSocket est bien OK.

Chat en ligne (Django + Channels) REST: `http://127.0.0.1:8000/api/` • WS: `ws://127.0.0.1:8000/ws/chat/<room>/` system • 07/02/2026 17:45:18 Déconnecté
system • 07/02/2026 17:45:22 Déconnecté

OK — l’API marche, mais **le WebSocket se ferme tout de suite**. Là, on doit vérifier **2 choses** : (A) est-ce que Django *voit* la connexion WS ? (B) est-ce que l’URL WS est bien routée par Channels/ASGI ?

1) Regarde la console où `runserver` tourne

Clique **Rejoindre** puis regarde immédiatement la console Django :

- Si tu vois une ligne du type :
 - `WebSocket CONNECT /ws/chat/general/ puis DISCONNECT`
→ le WS arrive bien au serveur, on corrigera la cause de fermeture.
- Si tu vois :
 - `Not Found: /ws/chat/general/ (404)`
→ ton routing WS n'est pas pris en compte (ASGI/routing).
- Si tu ne vois **rien du tout** à chaque clic
→ tu n'es pas sur le bon serveur ASGI (ou mauvaise config).

👉 Dis-moi quelle ligne apparaît (copie/colle 2–3 lignes autour).

2) Assure-toi que tu lances bien le bon ASGI (monsite)

Dans `monsite/settings.py`, tu dois avoir :

```
INSTALLED_APPS = [  
    # ...  
    "channels",  
    "chat",  
]
```

```
ASGI_APPLICATION = "monsite.asgi.application"
```

Et dans `monsite/asgi.py`, tu dois avoir **l'import routing après** `get_asgi_application()` :

```
os.environ.setdefault("DJANGO_SETTINGS_MODULE", "monsite.settings")  
django_asgi_app = get_asgi_application()  
import chat.routing
```

(ce point est crucial, tu avais justement l'erreur "Apps aren't loaded yet" avant)

3) Vérifie que l'URL WS correspond exactement au routing

Dans `chat/routing.py` :

```
websocket_urlpatterns = [
    re_path(r"ws/chat/(?P<room_name>[^/]+)/$", ChatConsumer.as_asgi()),
]
```

Donc ton navigateur doit appeler :

`ws://127.0.0.1:8000/ws/chat/general/`

✅ avec le / final.

4) Test WS “brut” (sans navigateur) pour isoler le problème

Dans PowerShell :

```
pip install websockets
py -c "import os, asyncio, websockets;
async def t():
    async with websockets.connect('ws://127.0.0.1:8000/ws/chat/general/') as ws:
        await ws.send({'author': 'test', 'content': 'hello'})
        print(await ws.recv())
asyncio.run(t())"
```

- ✅ Si tu reçois un JSON → le backend WS est OK, et le souci est côté page/URL.
 - ❌ Si ça échoue (404/403/connection closed) → config ASGI/routing.
-

5) Cause fréquente chez toi : tu as `monsite` mais le projet initial était `config`

Donc parfois, tu as bien copié `chat/...` mais ton `monsite/asgi.py` n’est pas en “Channels mode” ou pointe vers le mauvais `DJANGO_SETTINGS_MODULE`.

Donne-moi juste ceci (copie/colle)

1. Les lignes qui apparaissent dans la **console Django** quand tu cliques “Rejoindre”
2. Le contenu complet de `monsite/asgi.py`

Et je te dis exactement la correction (2-3 lignes) pour que ça reste connecté.

ou se trouve la console Django

La “console Django”, c’est tout simplement **la fenêtre du terminal** où tu as lancé le serveur avec :

```
py manage.py runserver
```

Sur Windows (PowerShell)

1. Ouvre PowerShell dans ton dossier projet (là où il y a `manage.py`)
2. Lance :

```
py manage.py runserver
```

3. **Ne ferme pas cette fenêtre** : elle reste ouverte et affiche des lignes comme :

- Starting development server at `http://127.0.0.1:8000/`
- puis des logs quand tu visites une page :
 - `"GET /api/rooms/ HTTP/1.1" 200 ...`
 - ou des erreurs `Not Found: ...`

👉 Ensuite, quand tu cliques **Rejoindre** dans ton navigateur, **reviens dans cette fenêtre PowerShell** : tu verras ce que Django reçoit pour la WebSocket.

Astuce

Si tu as plusieurs fenêtres PowerShell ouvertes : celle qui est la “console Django” est celle où tu vois défiler des logs **en direct**.

Si tu veux, envoie-moi ce que tu vois dans cette fenêtre juste après avoir cliqué “Rejoindre” (2–5 lignes), et je te dis exactement pourquoi ça se déconnecte.

Rien ne s'affiche dans la console django

Réflexion durant 16s

Si **rien ne s’affiche** dans la console où `runserver` tourne quand tu cliques **Rejoindre**, ça veut dire que la tentative WebSocket **n’arrive même pas** jusqu’à Django (donc fermeture côté navigateur/réseau avant handshake).

Voilà comment diagnostiquer en 2 minutes.

1) Vérifie que Django log bien quelque chose (test simple)



Dans ton navigateur, ouvre juste :

- `http://127.0.0.1:8000/api/rooms/`

→ Est-ce que **ça affiche une ligne** dans la console PowerShell du serveur ?

Exemple :

```
"GET /api/rooms/ HTTP/1.1" 200 ...
```



-  Oui → ta console est la bonne, Django log OK
-  Non → tu ne regardes pas la bonne fenêtre terminal (ou serveur pas lancé)

2) Test “est-ce que le navigateur atteint l’URL WS ?”

Toujours dans le navigateur, tape cette URL (en HTTP, pas WS) :

- `http://127.0.0.1:8000/ws/chat/general/`

Même si ce n’est pas une vraie requête WebSocket, Django devrait au minimum logger un `GET /ws/chat/general/` (souvent 404/400).

-  Si ça log dans la console → le chemin existe côté serveur, c’est juste le handshake WS qui pose souci
-  Si ça ne log pas → ton navigateur n’atteint pas le serveur sur ce chemin (souvent problème d’ASGI/routing/port)

3) Regarde l’erreur WebSocket dans le navigateur (très important)

Ouvre les DevTools :

- Chrome/Edge : **F12** → onglet **Console**

- puis clique **Rejoindre**

Tu verras presque toujours une erreur du style :

- `WebSocket connection to 'ws://127.0.0.1:8000/ws/chat/general/' failed: ...`

Copie/colle **cette ligne** ici, c'est elle qui dit exactement pourquoi ça ferme.

4) Vérifie que le serveur écoute bien sur le port 8000

Dans PowerShell :

```
netstat -ano | findstr :8000
```

Tu dois voir une ligne en `LISTENING`.

5) Correction fréquente : URL WS “en dur” dans index.html

Si tu sers ta page autrement (ex: via Django ou autre port), mieux vaut générer l'URL automatiquement :

Dans `index.html`, remplace la création du socket par :

```
const wsScheme = (window.location.protocol === "https:") ? "wss" : "ws";
const wsHost = window.location.host || "127.0.0.1:8000"; // si ouvert en file://
socket = new WebSocket(`${wsScheme}://${wsHost}/ws/chat/${encodeURIComponent(roomName)}`);
```

Ça évite les erreurs quand tu n'es pas exactement sur `127.0.0.1:8000`.

Dis-moi juste 2 choses et je te donne la correction exacte

1. Quand tu ouvres `http://127.0.0.1:8000/api/rooms/`, est-ce que ça log dans la console serveur ? (oui/non)

2. La ligne d'erreur WebSocket dans **F12** → **Console** quand tu cliques "Rejoindre".

Avec ça, je te dis précisément si c'est un souci **ASGI**, **routing**, **port**, **URL**, ou **navigateur/firewall**.

Voici ce qui s'affiche quand je saisis `http://127.0.0.1:8000/ws/chat/general/` : Page not found (404) Request Method: GET Request URL: `http://127.0.0.1:8000/ws/chat/general/` Using the URLconf defined in `monsite.urls`, Django tried these URL patterns, in this order: `admin/` `api/` The current path, `ws/chat/general/`, didn't match any of these. You're seeing this error because you have `DEBUG = True` in your Django settings file. Change that to `False`, and Django will display a standard 404 page.

Nickel, ça explique tout  : **ton serveur ne route pas du tout `/ws/` . . .**

Le message dit que Django (URLconf `monsite.urls`) ne connaît que :

- `admin/`
- `api/`

Donc **Channels/ASGI ne prend pas la main sur les WebSockets** → ton navigateur se "déconnecte" immédiatement.

Il faut corriger **2 fichiers** pour activer Channels correctement.

1) `monsite/asgi.py` (obligatoire)

Assure-toi que ton `monsite/asgi.py` contient exactement une config Channels comme ceci :

```
import os
from django.core.asgi import get_asgi_application
from channels.routing import ProtocolTypeRouter, URLRouter
from channels.auth import AuthMiddlewareStack

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "monsite.settings")

django_asgi_app = get_asgi_application()

import chat.routing  # après init Django

application = ProtocolTypeRouter({
    "http": django_asgi_app,
```

```
        "websocket": AuthMiddlewareStack(
            URLRouter(chat.routing.websocket_urlpatterns)
        ),
    })
```

2) monsite/settings.py

Vérifie ces lignes :

a) INSTALLED_APPS doit contenir channels et chat

```
INSTALLED_APPS = [
    # ...
    "channels",
    "chat",
]
```

b) ASGI_APPLICATION doit pointer sur le bon fichier

```
ASGI_APPLICATION = "monsite.asgi.application"
```

3) Redémarre le serveur (important)

Stoppe le serveur (CTRL+C) puis relance :

```
py manage.py runserver
```

4) Test rapide (à faire après redémarrage)

Retape :

```
http://127.0.0.1:8000/ws/chat/general/
```

Tu n'auras pas une “page” WebSocket (c'est normal), mais **tu ne dois plus voir** une 404 qui liste seulement admin/ et api/. Et surtout, quand tu cliqueras **Rejoindre**, la console Django devrait afficher des logs liés au WebSocket.

Pourquoi tu avais cette 404 ?

Parce que les **WebSockets ne passent pas par `monsite.urls`** (HTTP), ils passent par **`asgi.py` + `chat.routing`**.

Si ton serveur tourne sans Channels (ou avec un `asgi.py` “classique”), Django ne connaît que `urls.py` → donc `/ws/...` = 404.

Si tu veux la page chat à / aussi

Actuellement tu n’as pas de route `/` (d’où tes 404 sur `/`). Je peux te faire une route propre :

- `/` = page chat (template)
- `/api/...` = REST
- `/ws/...` = WebSocket

Mais d’abord, fais les corrections ASGI ci-dessus.

Si tu me colles le contenu actuel de `monsite/asgi.py`, je te dis exactement ce qui manque (souvent 5 lignes).

```
asgi.py
Python
settings.py
Python
urls.py
Python
wsgi.py
Python
Réflexion durant 11s
```


Je viens de regarder tes fichiers : ta config **Channels/ASGI est correcte**.

asgi

settings

Ce que tu as fait ici :

taper `http://127.0.0.1:8000/ws/chat/general/` dans la barre d'adresse

...produit **forcément** une 404, et c'est **normal** , parce que :

- la barre d'adresse envoie une requête **HTTP GET classique**
- or ton `urls.py` ne déclare que `admin/` et `api/`

`urls`

- les WebSockets **ne passent pas** par `monsite.urls` : ils passent par `asgi.py` et ne sont routés que si la requête est un vrai **handshake WebSocket** (avec header `Upgrade: websocket`)

Donc : **404 en HTTP sur `/ws/...` = normal**. Ça ne prouve pas que WebSocket ne marche pas.

Comment tester “pour de vrai” le WebSocket

Test 1 (le plus simple) : via ton `index.html`

1. Lance Django :

```
py manage.py runserver
```

2. Ouvre `frontend/index.html`
3. Mets `general` et clique **Rejoindre**





Ensuite, ouvre **F12 → Network → WS** (WebSocket) :

- tu dois voir une connexion vers `/ws/chat/general/`
- clique dessus pour voir les Frames / erreurs

Test 2 (PowerShell) : test WS en ligne de commande

```
pip install websockets
py -c "import asyncio, websockets
async def t():
    async with websockets.connect('ws://127.0.0.1:8000/ws/chat/general/') as ws:
        await ws.send('{\"author\": \"test\", \"content\": \"hello\"}')
        print(await ws.recv())
asyncio.run(t())"
```

-  Si ça affiche une réponse JSON → backend WS OK
-  Si ça échoue → copie/colle l'erreur exacte

Petite amélioration utile : voir la page chat à /

Actuellement / renvoie 404 (normal) car tu n'as pas de route pour la page. Ton `urls.py` ne contient que `admin/` et `api/`.

`urls`

Si tu veux que `http://127.0.0.1:8000/` affiche la page, dis-moi et je te donne le patch “page / + API /api/ + WS /ws/”.

Ce dont j'ai besoin pour régler ton “Déconnecté” (si ça continue)



Copie/colle la ligne d'erreur dans **F12** → **Console** au moment où ça déconnecte (souvent c'est 404/403/connection closed), et je te dis exactement quoi corriger.

Sources

```
PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> & "D:/PYTHON DJANGO HOPES/Chat_API_DJANGO_mario/venv/Scripts/Activate.ps1"
(venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> py -c "import os;
os.environ.setdefault('DJANGO_SETTINGS_MODULE','monsite.settings'); import monsite.asgi as a; print('ASGI OK', a.application)" Traceback (most recent
call last): File "<string>", line 1, in <module> File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\monsite\asgi.py", line 5, in <module> import
chat.routing File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\routing.py", line 2, in <module> from .consumers import ChatConsumer
File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\consumers.py", line 4, in <module> from .models import Room, Message File
```

```
"D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\models.py", line 3, in <module> class Room(models.Model): File "D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\db\models\base.py", line 131, in __new__ app_config =
apps.get_containing_app_config(module) ~~~~~ File "D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\apps\registry.py", line 260, in get_containing_app_config self.check_apps_ready() File
"D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\apps\registry.py", line 138, in check_apps_ready raise
AppRegistryNotReady("Apps aren't loaded yet.") django.core.exceptions.AppRegistryNotReady: Apps aren't loaded yet. (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> py manage.py runserver Watching for file changes with StatReloader Performing system checks... February 07, 2026 -
17:35:11 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK.
WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on
production servers see: https://docs.djangoproject.com/en/5.2/howto/deployment/ (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario>
python manage.py createsuperuser Nom d'utilisateur (leave blank to use 'spasse02men'): Mario Adresse électronique: halley mariomen@gmail.com Password:
Password (again): Le mot de passe est trop semblable au champ « nom d'utilisateur ». Ce mot de passe est trop court. Il doit contenir au minimum 8 caractères.
Ce mot de passe est trop courant. Bypass password validation and create user anyway? [y/N]: y Superuser created successfully. (venv) PS D:\PYTHON
DJANGO HOPES\Chat_API_DJANGO_mario> py manage.py runserver Watching for file changes with StatReloader Performing system checks... February
07, 2026 - 17:47:00 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-
BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more
information on production servers see: https://docs.djangoproject.com/en/5.2/howto/deployment/ (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> py manage.py runserver Watching for file changes with StatReloader Performing system checks... February 07, 2026 -
18:07:49 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK.
WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on
production servers see: https://docs.djangoproject.com/en/5.2/howto/deployment/ (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> pip
install websockets Collecting websockets Downloading websockets-16.0-cp312-cp312-win_amd64.whl.metadata (7.0 kB) Downloading websockets-16.0-
cp312-cp312-win_amd64.whl (178 kB) ~~~~~ 178.7/178.7 kB
348.0 kB/s eta 0:00:00 Installing collected packages: websockets Successfully installed websockets-16.0 [notice] A new release of pip is available: 24.0 ->
26.0.1 [notice] To update, run: python.exe -m pip install --upgrade pip (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> py -c "import
asyncio, websockets >> async def t(): >> async with websockets.connect('ws://127.0.0.1:8000/ws/chat/general/') as ws: >> await
ws.send({'author':'test','content':'hello'}) >> print(await ws.recv()) >> asyncio.run(t())" recv())\x0aasyncio.run(t())";1fa60f14-0586-49a2-9638-
f81a6a3c3c0e<string>:4: SyntaxWarning: invalid escape sequence '\' (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario>
```

Je vois **2 problèmes distincts** dans ce que tu as collé :

1.  **Ton monsite/asgi.py est encore “mauvais”** (il importe `chat.routing` trop tôt) → `AppRegistryNotReady: Apps aren't loaded yet`.
2.  Ton test WebSocket en PowerShell a été **mal interprété** (commande multi-lignes / échappements), donc il **n'a pas réellement testé** la connexion.

On corrige ça proprement.

1) Corriger définitivement `monsite/asgi.py`

Ouvre `D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\monsite\asgi.py`

➡ Et mets **exactement** ceci (important : import `chat.routing` **APRÈS** `get_asgi_application()`):

```
import os
from django.core.asgi import get_asgi_application
from channels.routing import ProtocolTypeRouter, URLRouter
from channels.auth import AuthMiddlewareStack

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "monsite.settings")

django_asgi_app = get_asgi_application()

# IMPORTANT: import après initialisation Django
import chat.routing

application = ProtocolTypeRouter({
    "http": django_asgi_app,
    "websocket": AuthMiddlewareStack(
        URLRouter(chat.routing.websocket_urlpatterns)
    ),
})
```

✅ Après ça, ce test doit marcher :

```
py -c "import os; os.environ.setdefault('DJANGO_SETTINGS_MODULE','monsite.settings'); import monsite.asgi as a; print('ASGI OK')"
```

2) Tester le WebSocket correctement (PowerShell)

⚠ Important : le serveur doit tourner dans une autre fenêtre.

Fenêtre 1 (serveur)

```
py manage.py runserver
```

Fenêtre 2 (test WS)

Ton `py -c` multi-ligne a été cassé par PowerShell. Fais plutôt un **one-liner** :

```
py -c "import asyncio, websockets; async def t():
    async with websockets.connect('ws://127.0.0.1:8000/ws/chat/general/') as ws:
        await ws.send('{\"author\": \"test\", \"content\": \"hello\"}');
        print(await ws.recv());
    asyncio.run(t())"
```

Si PowerShell râle encore avec les retours à la ligne, la méthode la plus sûre est de créer un fichier.

3) Méthode 100% fiable : fichier `test_ws.py`

Dans ton dossier projet (là où il y a `manage.py`), crée `test_ws.py` :

```
import asyncio
import websockets

async def main():
    async with websockets.connect("ws://127.0.0.1:8000/ws/chat/general/") as ws:
        await ws.send('{\"author\": \"test\", \"content\": \"hello\"}')
        print(await ws.recv())

asyncio.run(main())
```

Puis lance :

```
py test_ws.py
```



Si tout est bon, tu verras une réponse JSON imprimée.

4) Pourquoi tu es “Déconnecté” dans la page

Parce que tant que `asgi.py` n’est pas chargé correctement par Django/Channels, la route `/ws/...` n’existe pas en WebSocket, donc le navigateur ferme.

Si tu veux, colle ici le contenu actuel de ton `monsite/asgi.py` après modification, et je te confirme en 5 secondes que c'est bon.

```
venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: def user_config_dir(self) -> str: venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: :return:
config directory tied to the user, e.g. '~/.config/$appname/$version or venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py:
`$XDG_CONFIG_HOME/$appname/$version venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: path = os.environ.get("XDG_CONFIG_HOME", "")
venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: path = os.path.expanduser("~/config") # noqa: PTH111 venv\Lib\site-
packages\pip\_vendor\platformdirs\unix.py: def site_config_dir(self) -> str: venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: :return: config directories
shared by users (if multipath <platformdirs.api.PlatformDirsABC.multipath> venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: # XDG
default for $XDG_CONFIG_DIRS only first, if multipath is False venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: path =
os.environ.get("XDG_CONFIG_DIRS", "") venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: def site_config_path(self) -> Path: venv\Lib\site-
packages\pip\_vendor\platformdirs\unix.py: """:return: config path shared by the users. Only return first item, even if `multipath` is set to True"""
venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: return self._first_item_as_path_if_multipath(self.site_config_dir) venv\Lib\site-
packages\pip\_vendor\platformdirs\unix.py: """:Return directory from user-dirs.dirs config file. See https://freedesktop.org/wiki/Software/xdg-user-dirs/."""
venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: user_dirs_config_path = Path(Unix().user_config_dir) / "user-dirs.dirs" venv\Lib\site-
packages\pip\_vendor\platformdirs\unix.py: if user_dirs_config_path.exists(): venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: parser =
ConfigParser() venv\Lib\site-packages\pip\_vendor\platformdirs\unix.py: with user_dirs_config_path.open() as stream: venv\Lib\site-
packages\pip\_vendor\platformdirs\unix.py: # Add fake section header, so ConfigParser doesn't complain venv\Lib\site-
packages\pip\_vendor\platformdirs\windows.py: def user_config_dir(self) -> str: venv\Lib\site-packages\pip\_vendor\platformdirs\windows.py: """:return:
config directory tied to the user, same as user_data_dir""" venv\Lib\site-packages\pip\_vendor\platformdirs\windows.py: def site_config_dir(self) -> str:
venv\Lib\site-packages\pip\_vendor\platformdirs\windows.py: """:return: config directory shared by the users, same as site_data_dir""" venv\Lib\site-
packages\pip\_vendor\platformdirs\__init__.py: def user_config_dir(venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: :returns: config directory
tied to the user venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: ).user_config_dir venv\Lib\site-
packages\pip\_vendor\platformdirs\__init__.py: def site_config_dir(venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: :returns: config directory
shared by the users venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: ).site_config_dir venv\Lib\site-
packages\pip\_vendor\platformdirs\__init__.py: def user_config_path(venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: :returns: config path tied to
the user venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: ).user_config_path venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: def
site_config_path(venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: :returns: config path shared by the users venv\Lib\site-
packages\pip\_vendor\platformdirs\__init__.py: ).site_config_path venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: "user_config_dir",
venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: "site_config_dir", venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py:
"user_config_path", venv\Lib\site-packages\pip\_vendor\platformdirs\__init__.py: "site_config_path", venv\Lib\site-
packages\pip\_vendor\platformdirs\__main__.py: "user_config_dir", venv\Lib\site-packages\pip\_vendor\platformdirs\__main__.py: "site_config_dir",
venv\Lib\site-packages\pip\_vendor\pygments\formatters\html.py: # some configurations seem to mess up the formatting... venv\Lib\site-
packages\pip\_vendor\pygments\formatters\img.py: if line.startswith(b'Fontconfig warning:'): venv\Lib\site-packages\pip\_vendor\pygments\lexers\python.py:
'show_config', 'shuffle', 'sign', 'signbit', 'sin', 'sinc', 'sinh', venv\Lib\site-packages\pip\_vendor\pygments\lexers\_mapping.py: 'ApacheConfLexer':
('pip\_vendor.pygments.lexers.configs', 'ApacheConf', ('apacheconf', 'aconf', 'apache'), ('htaccess', 'apache.conf', 'apache2.conf'), ('text/x-apacheconf',)),
```

venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'AugeasLexer': ('pip._vendor.pygments.lexers.configs', 'Augeas', ('augeas',), (*aug',), ()),
venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'Cfengine3Lexer': ('pip._vendor.pygments.lexers.configs', 'CFEngine3', ('cfengine3', 'cf3'),
(*.cf',), ()), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'DockerLexer': ('pip._vendor.pygments.lexers.configs', 'Docker', ('docker',
'dockerfile'), ('Dockerfile', '*.docker'), ('text/x-dockerfile-config',)), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'IniLexer':
('pip._vendor.pygments.lexers.configs', 'INI', ('ini', 'cfg', 'dosini'), (*.ini', '*.cfg', '*.inf', '.editorconfig', '*.service', '*.socket', '*.device', '*.mount', '*.automount',
.swap', '.target', '*.path', '*.timer', '*.slice', '*.scope'), ('text/x-ini', 'text/inf')), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py:
'KconfigLexer': ('pip._vendor.pygments.lexers.configs', 'Kconfig', ('kconfig', 'menuconfig', 'linux-config', 'kernel-config'), ('Kconfig*', '*Config.in*',
'external.in*', 'standard-modules.in'), ('text/x-kconfig',)), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'LighttpdConfLexer':
('pip._vendor.pygments.lexers.configs', 'Lighttpd configuration file', ('lighttpd', 'lighty'), ('lighttpd.conf',), ('text/x-lighttpd-conf',)), venv\Lib\site-
packages\pip_vendor\pygments\lexers_mapping.py: 'NestedTextLexer': ('pip._vendor.pygments.lexers.configs', 'NestedText', ('nestedtext', 'nt'), (*.nt',), ()),
venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'NginxConfLexer': ('pip._vendor.pygments.lexers.configs', 'Nginx configuration file',
('nginx',), ('nginx.conf',), ('text/x-nginx-conf',)), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'PacmanConfLexer':
('pip._vendor.pygments.lexers.configs', 'PacmanConf', ('pacmanconf',), ('pacman.conf',), ()), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py:
'PkgConfigLexer': ('pip._vendor.pygments.lexers.configs', 'PkgConfig', ('pkgconfig',), (*.pc',), ()), venv\Lib\site-
packages\pip_vendor\pygments\lexers_mapping.py: 'PropertiesLexer': ('pip._vendor.pygments.lexers.configs', 'Properties', ('properties', 'jproperties'),
(*.properties',), ('text/x-java-properties',)), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'RegeditLexer':
('pip._vendor.pygments.lexers.configs', 'reg', ('registry',), (*.reg',), ('text/x-windows-registry',)), venv\Lib\site-
packages\pip_vendor\pygments\lexers_mapping.py: 'SingularityLexer': ('pip._vendor.pygments.lexers.configs', 'Singularity', ('singularity',), (*.def',
'Singularity'), ()), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'SquidConfLexer': ('pip._vendor.pygments.lexers.configs', 'SquidConf',
('squidconf', 'squid.conf', 'squid'), ('squid.conf',), ('text/x-squidconf',)), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'TOMLLexer':
('pip._vendor.pygments.lexers.configs', 'TOML', ('toml',), (*.toml', 'Pipfile', 'poetry.lock'), ()), venv\Lib\site-
packages\pip_vendor\pygments\lexers_mapping.py: 'TermcapLexer': ('pip._vendor.pygments.lexers.configs', 'Termcap', ('termcap',), ('termcap', 'termcap.src'),
()), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'TerminfoLexer': ('pip._vendor.pygments.lexers.configs', 'Terminfo', ('terminfo',),
('terminfo', 'terminfo.src'), ()), venv\Lib\site-packages\pip_vendor\pygments\lexers_mapping.py: 'TerraformLexer': ('pip._vendor.pygments.lexers.configs',
'Terraform', ('terraform', 'tf', 'hcl'), (*.tf', '*.hcl'), ('application/x-tf', 'application/x-terraform')), venv\Lib\site-
packages\pip_vendor\pygments\lexers_mapping.py: 'UnixConfigLexer': ('pip._vendor.pygments.lexers.configs', 'Unix/Linux config files', ('unixconfig',
'linuxconfig'), ()), venv\Lib\site-packages\pip_vendor\pyparsing\core.py: __config_flags, venv\Lib\site-packages\pip_vendor\pyparsing\core.py: class
__compat__(__config_flags): venv\Lib\site-packages\pip_vendor\pyparsing\core.py: A cross-version compatibility configuration for pyparsing features that
will be venv\Lib\site-packages\pip_vendor\pyparsing\core.py: released in a future version. By setting values in this configuration to True, venv\Lib\site-
packages\pip_vendor\pyparsing\core.py: class __diag__(__config_flags): venv\Lib\site-packages\pip_vendor\pyparsing\core.py: Diagnostic configuration (all
default to disabled) venv\Lib\site-packages\pip_vendor\pyparsing\core.py: del __config_flags venv\Lib\site-packages\pip_vendor\pyparsing\testing.py:
Context manager to be used when writing unit tests that modify pyparsing config values: venv\Lib\site-packages\pip_vendor\pyparsing\util.py: class
__config_flags: venv\Lib\site-packages\pip_vendor\pyparsing\util.py: _type_desc = "configuration" venv\Lib\site-
packages\pip_vendor\pyproject_hooks_impl.py: def get_requires_for_build_wheel(self, config_settings=None): venv\Lib\site-
packages\pip_vendor\pyproject_hooks_impl.py: 'config_settings': config_settings venv\Lib\site-packages\pip_vendor\pyproject_hooks_impl.py: self,
metadata_directory, config_settings=None, venv\Lib\site-packages\pip_vendor\pyproject_hooks_impl.py: 'config_settings': config_settings, venv\Lib\site-
packages\pip_vendor\pyproject_hooks_impl.py: self, wheel_directory, config_settings=None, venv\Lib\site-packages\pip_vendor\pyproject_hooks_impl.py:

```

'config_settings': config_settings, venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: def get_requires_for_build_editable(self,
config_settings=None): venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: 'config_settings': config_settings venv\Lib\site-
packages\pip\_vendor\pyproject_hooks\_impl.py: self, metadata_directory, config_settings=None, venv\Lib\site-
packages\pip\_vendor\pyproject_hooks\_impl.py: 'config_settings': config_settings, venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: self,
wheel_directory, config_settings=None, venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: 'config_settings': config_settings, venv\Lib\site-
packages\pip\_vendor\pyproject_hooks\_impl.py: def get_requires_for_build_sdist(self, config_settings=None): venv\Lib\site-
packages\pip\_vendor\pyproject_hooks\_impl.py: 'config_settings': config_settings venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: def
build_sdist(self, sdist_directory, config_settings=None): venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_impl.py: 'config_settings': config_settings,
venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: def get_requires_for_build_wheel(config_settings): venv\Lib\site-
packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: return hook(config_settings) venv\Lib\site-
packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: def get_requires_for_build_editable(config_settings): venv\Lib\site-
packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: return hook(config_settings) venv\Lib\site-
packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: metadata_directory, config_settings, _allow_fallback): venv\Lib\site-
packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: return hook(metadata_directory, config_settings) venv\Lib\site-
packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: whl_basename = backend.build_wheel(metadata_directory, config_settings) venv\Lib\site-
packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: config_settings) venv\Lib\site-
packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: metadata_directory, config_settings, _allow_fallback): venv\Lib\site-
packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: whl_basename = build_hook(metadata_directory, config_settings) venv\Lib\site-
packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: config_settings) venv\Lib\site-
packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: return hook(metadata_directory, config_settings) venv\Lib\site-
packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: whl_basename, metadata_directory, config_settings): venv\Lib\site-
packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: def build_wheel(wheel_directory, config_settings, metadata_directory=None):
venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: return _build_backend().build_wheel(wheel_directory, config_settings,
venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: def build_editable(wheel_directory, config_settings,
metadata_directory=None): venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: return hook(wheel_directory, config_settings,
metadata_directory) venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: def get_requires_for_build_sdist(config_settings):
venv\Lib\site-packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: return hook(config_settings) venv\Lib\site-
packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: def build_sdist(sdist_directory, config_settings): venv\Lib\site-
packages\pip\_vendor\pyproject_hooks\_in_process\_in_process.py: return backend.build_sdist(sdist_directory, config_settings) venv\Lib\site-
packages\pip\_vendor\requests\adapters.py: "config", venv\Lib\site-packages\pip\_vendor\requests\adapters.py: self.config = {} venv\Lib\site-
packages\pip\_vendor\requests\adapters.py: self.config = {} venv\Lib\site-packages\pip\_vendor\requests\adapters.py: :param proxy_kwargs: Extra keyword
arguments used to configure the Proxy Manager. venv\Lib\site-packages\pip\_vendor\requests\sessions.py: """This method re-evaluates the proxy configuration
by considering the venv\Lib\site-packages\pip\_vendor\requests\sessions.py: NO_PROXY, we strip the proxy configuration. Otherwise, we set missing
venv\Lib\site-packages\pip\_vendor\requests\sessions.py: Provides cookie persistence, connection-pooling, and configuration. venv\Lib\site-
packages\pip\_vendor\requests\sessions.py: #: Trust environment settings for proxy configuration, default venv\Lib\site-
packages\pip\_vendor\requests\sessions.py: # Look for requests environment configuration venv\Lib\site-packages\pip\_vendor\requests\utils.py: """This
method takes proxy information from a request and configuration venv\Lib\site-packages\pip\_vendor\requests\utils.py: such a NO_PROXY to strip proxy

```

configurations. venv\Lib\site-packages\pip_vendor\requests\utils.py: :param trust_env: Boolean declaring whether to trust environment configs venv\Lib\site-packages\pip_vendor\rich\console.py: Some terminals may require configuration changes before you can set the title. venv\Lib\site-packages\pip_vendor\rich\logging.py: Be careful when enabling console markup in log messages if you have configured logging for libraries not venv\Lib\site-packages\pip_vendor\rich\logging.py: logging.basicConfig(venv\Lib\site-packages\pip_vendor\rich\logging.py: "Loading configuration file /adasd/asdasd/qeqwe/qwrqwrqwr/sdgsdgsdg/werwerwer/dfgerert/ertertert/ertetert/werwerwer" venv\Lib\site-packages\pip_vendor\rich\theme.py:import configparser venv\Lib\site-packages\pip_vendor\rich\theme.py: def config(self) -> str: venv\Lib\site-packages\pip_vendor\rich\theme.py: """Get contents of a config file for this theme.""" venv\Lib\site-packages\pip_vendor\rich\theme.py: config = "[styles]\n" + "\n".join(venv\Lib\site-packages\pip_vendor\rich\theme.py: return config venv\Lib\site-packages\pip_vendor\rich\theme.py: cls, config_file: IO[str], source: Optional[str] = None, inherit: bool = True venv\Lib\site-packages\pip_vendor\rich\theme.py: config_file (IO[str]): An open conf file. venv\Lib\site-packages\pip_vendor\rich\theme.py: config = configparser.ConfigParser() venv\Lib\site-packages\pip_vendor\rich\theme.py: config.read_file(config_file, source=source) venv\Lib\site-packages\pip_vendor\rich\theme.py: styles = {name: Style.parse(value) for name, value in config.items("styles")} venv\Lib\site-packages\pip_vendor\rich\theme.py: path (str): Path to a config file readable by Python configparser module. venv\Lib\site-packages\pip_vendor\rich\theme.py: encoding (str, optional): Encoding of the config file. Defaults to None. venv\Lib\site-packages\pip_vendor\rich\theme.py: with open(path, "rt", encoding=encoding) as config_file: venv\Lib\site-packages\pip_vendor\rich\theme.py: return cls.from_file(config_file, source=path, inherit=inherit) venv\Lib\site-packages\pip_vendor\rich\theme.py: print(theme.config) venv\Lib\site-packages\pip_vendor\rich__init__.py: __all__ = ["get_console", "reconfigure", "print", "inspect", "print_json"] venv\Lib\site-packages\pip_vendor\rich__init__.py: def reconfigure(*args: Any, **kwargs: Any) -> None: venv\Lib\site-packages\pip_vendor\rich__init__.py: """Reconfigures the global console by replacing it with another. venv\Lib\site-packages\pip_vendor\six.py: MovedModule("configparser", "ConfigParser"), venv\Lib\site-packages\pip_vendor\truststore_api.py: from ._windows import _configure_context, _verify_peercerts_impl venv\Lib\site-packages\pip_vendor\truststore_api.py: from ._macos import _configure_context, _verify_peercerts_impl venv\Lib\site-packages\pip_vendor\truststore_api.py: from ._openssl import _configure_context, _verify_peercerts_impl venv\Lib\site-packages\pip_vendor\truststore_api.py: with _configure_context(self._ctx): venv\Lib\site-packages\pip_vendor\truststore_api.py: with _configure_context(self._ctx): venv\Lib\site-packages\pip_vendor\truststore_macos.py: def _configure_context(ctx: ssl.SSLContext) -> typing.Iterator[None]: venv\Lib\site-packages\pip_vendor\truststore_macos.py: # explicitly okay with from SSLContext configuration venv\Lib\site-packages\pip_vendor\truststore_openssl.py: def _configure_context(ctx: ssl.SSLContext) -> typing.Iterator[None]: venv\Lib\site-packages\pip_vendor\truststore_openssl.py: # or the path configured when OpenSSL was compiled, venv\Lib\site-packages\pip_vendor\truststore_openssl.py: # or the path configured when OpenSSL was compiled, venv\Lib\site-packages\pip_vendor\truststore_windows.py: class CERT_CHAIN_ENGINE_CONFIG(Structure): venv\Lib\site-packages\pip_vendor\truststore_windows.py: PCERT_CHAIN_ENGINE_CONFIG = POINTER(CERT_CHAIN_ENGINE_CONFIG) venv\Lib\site-packages\pip_vendor\truststore_windows.py: PCERT_CHAIN_ENGINE_CONFIG, venv\Lib\site-packages\pip_vendor\truststore_windows.py: cert_chain_engine_config = CERT_CHAIN_ENGINE_CONFIG() venv\Lib\site-packages\pip_vendor\truststore_windows.py: cert_chain_engine_config.cbSize = sizeof(cert_chain_engine_config) venv\Lib\site-packages\pip_vendor\truststore_windows.py: cert_chain_engine_config.hExclusiveRoot = hRootCertStore venv\Lib\site-packages\pip_vendor\truststore_windows.py: pConfig = pointer(cert_chain_engine_config) venv\Lib\site-packages\pip_vendor\truststore_windows.py: pConfig, venv\Lib\site-packages\pip_vendor\truststore_windows.py: def _configure_context(ctx: ssl.SSLContext) -> typing.Iterator[None]: venv\Lib\site-packages\pip_vendor\urllib3\connection.py: Additional keyword parameters are used to configure attributes of the connection. venv\Lib\site-packages\pip_vendor\urllib3\connection.py: self.proxy_config = kw.pop("proxy_config", None) venv\Lib\site-packages\pip_vendor\urllib3\connection.py: proxy_config = self.proxy_config venv\Lib\site-packages\pip_vendor\urllib3\connection.py: ssl_context = proxy_config.ssl_context venv\Lib\site-

packages\pip_vendor\urllib3\connectionpool.py: Retry configuration to use by default with requests in this pool. venv\Lib\site-packages\pip_vendor\urllib3\connectionpool.py: _proxy_config=None, venv\Lib\site-packages\pip_vendor\urllib3\connectionpool.py: self.proxy_config = _proxy_config venv\Lib\site-packages\pip_vendor\urllib3\connectionpool.py: self.conn_kw["proxy_config"] = self.proxy_config venv\Lib\site-packages\pip_vendor\urllib3\connectionpool.py: Configure the number of retries to allow before raising a venv\Lib\site-packages\pip_vendor\urllib3\connectionpool.py: self.proxy, self.proxy_config, destination_scheme venv\Lib\site-packages\pip_vendor\urllib3\contrib\appengine.py::class:PoolManager without any configuration or special environment variables. venv\Lib\site-packages\pip_vendor\urllib3\contrib\securetransport.py: # We save off the previously-configured timeout and then set it to venv\Lib\site-packages\pip_vendor\urllib3\exceptions.py: """Warned when certain TLS/SSL configuration is not available on a platform.""" venv\Lib\site-packages\pip_vendor\urllib3\packages\six.py: MovedModule("configparser", "ConfigParser"), venv\Lib\site-packages\pip_vendor\urllib3\poolmanager.py: "key__proxy_config", # class venv\Lib\site-packages\pip_vendor\urllib3\poolmanager.py: _proxy_config_fields = ("ssl_context", "use_forwarding_for_https") venv\Lib\site-packages\pip_vendor\urllib3\poolmanager.py: ProxyConfig = collections.namedtuple("ProxyConfig", _proxy_config_fields) venv\Lib\site-packages\pip_vendor\urllib3\poolmanager.py:#: Each PoolManager makes a copy of this dictionary so they can be configured venv\Lib\site-packages\pip_vendor\urllib3\poolmanager.py: proxy_config = None venv\Lib\site-packages\pip_vendor\urllib3\poolmanager.py: self.proxy, self.proxy_config, parsed_url.scheme venv\Lib\site-packages\pip_vendor\urllib3\poolmanager.py: if six.PY2 and not self.proxy_config.use_forwarding_for_https: venv\Lib\site-packages\pip_vendor\urllib3\poolmanager.py: self.proxy_config = ProxyConfig(proxy_ssl_context, use_forwarding_for_https) venv\Lib\site-packages\pip_vendor\urllib3\poolmanager.py: connection_pool_kw["_proxy_config"] = self.proxy_config venv\Lib\site-packages\pip_vendor\urllib3\poolmanager.py: if not connection_requires_http_tunnel(self.proxy, self.proxy_config, u.scheme): venv\Lib\site-packages\pip_vendor\urllib3\util\proxy.py: proxy_url=None, proxy_config=None, destination_scheme=None venv\Lib\site-packages\pip_vendor\urllib3\util\proxy.py: :param ProxyConfig proxy_config: venv\Lib\site-packages\pip_vendor\urllib3\util\proxy.py: Proxy configuration from poolmanager.py venv\Lib\site-packages\pip_vendor\urllib3\util\proxy.py: and proxy_config venv\Lib\site-packages\pip_vendor\urllib3\util\proxy.py: and proxy_config.use_forwarding_for_https venv\Lib\site-packages\pip_vendor\urllib3\util\retry.py: """Retry configuration. venv\Lib\site-packages\pip_vendor\urllib3\util\ssl_.py: "urllib3 from configuring SSL appropriately and may cause " venv\Lib\site-packages\pip_vendor\urllib3\util\timeout.py: """Timeout configuration. venv\Lib\site-packages\pip_vendor\urllib3\util\timeout.py: Timeout object to ensure each one has its own start/stop configured. venv\Lib\site-packages\pip_vendor\urllib3\util\wait.py: # to use it, either due to strange configuration or broken monkeypatching venv\Lib\site-packages\redis\asyncio\client.py: configuration, an instance will either use a ConnectionPool, or venv\Lib\site-packages\redis\asyncio\client.py: Return a Redis client object configured from the given URL venv\Lib\site-packages\redis\asyncio\client.py: 2. Define a Retry object with configured 'supported_errors' and set venv\Lib\site-packages\redis\asyncio\client.py: When 'connection_pool' is provided - the retry configuration of the venv\Lib\site-packages\redis\asyncio\cluster.py: <https://redis.io/docs/manual/scaling/#redis-cluster-configuration-parameters> venv\Lib\site-packages\redis\asyncio\cluster.py: | @deprecated - Please configure the 'retry' object instead venv\Lib\site-packages\redis\asyncio\cluster.py: Return a Redis client object configured from the given URL. venv\Lib\site-packages\redis\asyncio\cluster.py: reason="Please configure the 'load_balancing_strategy' instead", venv\Lib\site-packages\redis\asyncio\cluster.py: reason="Please configure the 'retry' object instead", venv\Lib\site-packages\redis\asyncio\cluster.py: # Call our on_connect function to configure READONLY mode venv\Lib\site-packages\redis\asyncio\cluster.py: # Sending READONLY command to server to configure connection as venv\Lib\site-packages\redis\asyncio\cluster.py: # rc.cluster_save_config(rc.get primaries()) venv\Lib\site-packages\redis\asyncio\cluster.py: # We are configuring the connection pool not to retry venv\Lib\site-packages\redis\asyncio\connection.py: # the command name, e.g., 'CONFIG GET'. The Redis server expects these venv\Lib\site-packages\redis\asyncio\connection.py: Return a connection pool configured from the given URL. venv\Lib\site-packages\redis\asyncio\multidb\client.py: from redis.asyncio.multidb.config import DEFAULT_GRACE_PERIOD, MultiDbConfig venv\Lib\site-packages\redis\asyncio\multidb\client.py: def __init__(self, config: MultiDbConfig): venv\Lib\site-packages\redis\asyncio\multidb\client.py: self._databases =

config.databases() venv\Lib\site-packages\redis\asyncio\multidb\client.py: config.default_health_checks() venv\Lib\site-packages\redis\asyncio\multidb\client.py: if not config.health_checks venv\Lib\site-packages\redis\asyncio\multidb\client.py: else config.health_checks venv\Lib\site-packages\redis\asyncio\multidb\client.py: self.health_check_interval = config.health_check_interval venv\Lib\site-packages\redis\asyncio\multidb\client.py: self.health_check_policy: HealthCheckPolicy = config.health_check_policy.value(venv\Lib\site-packages\redis\asyncio\multidb\client.py: config.health_check_probes, config.health_check_delay venv\Lib\site-packages\redis\asyncio\multidb\client.py: config.default_failure_detectors() venv\Lib\site-packages\redis\asyncio\multidb\client.py: if not config.failure_detectors venv\Lib\site-packages\redis\asyncio\multidb\client.py: else config.failure_detectors venv\Lib\site-packages\redis\asyncio\multidb\client.py: config.default_failover_strategy() venv\Lib\site-packages\redis\asyncio\multidb\client.py: if config.failover_strategy is None venv\Lib\site-packages\redis\asyncio\multidb\client.py: else config.failover_strategy venv\Lib\site-packages\redis\asyncio\multidb\client.py: self._auto_fallback_interval = config.auto_fallback_interval venv\Lib\site-packages\redis\asyncio\multidb\client.py: self._event_dispatcher = config.event_dispatcher venv\Lib\site-packages\redis\asyncio\multidb\client.py: self._command_retry = config.command_retry venv\Lib\site-packages\redis\asyncio\multidb\client.py: failover_attempts=config.failover_attempts, venv\Lib\site-packages\redis\asyncio\multidb\client.py: failover_delay=config.failover_delay, venv\Lib\site-packages\redis\asyncio\multidb\client.py: self._config = config venv\Lib\site-packages\redis\asyncio\multidb\command_executor.py: from redis.multidb.config import DEFAULT_AUTO_FALLBACK_INTERVAL venv\Lib\site-packages\redis\asyncio\multidb\config.py: class DatabaseConfig: venv\Lib\site-packages\redis\asyncio\multidb\config.py: Dataclass representing the configuration for a database connection. venv\Lib\site-packages\redis\asyncio\multidb\config.py: This class is used to store configuration settings for a database connection, venv\Lib\site-packages\redis\asyncio\multidb\config.py: attributes and allows for the creation of customized configurations for various venv\Lib\site-packages\redis\asyncio\multidb\config.py: from_pool (Optional[ConnectionPool]): A pre-configured connection pool to use. venv\Lib\site-packages\redis\asyncio\multidb\config.py: class MultiDbConfig: venv\Lib\site-packages\redis\asyncio\multidb\config.py: Configuration class for managing multiple database connections in a resilient and fail-safe manner. venv\Lib\site-packages\redis\asyncio\multidb\config.py: databases_config: A list of database configurations. venv\Lib\site-packages\redis\asyncio\multidb\config.py: Retrieves a collection of database clients managed by weighted configurations. venv\Lib\site-packages\redis\asyncio\multidb\config.py: Initializes database clients based on the provided configuration and removes venv\Lib\site-packages\redis\asyncio\multidb\config.py: with defined retry and backoff configurations. venv\Lib\site-packages\redis\asyncio\multidb\config.py: databases_config: List[DatabaseConfig] venv\Lib\site-packages\redis\asyncio\multidb\config.py: for database_config in self.databases_config: venv\Lib\site-packages\redis\asyncio\multidb\config.py: database_config.client_kwargs.update(venv\Lib\site-packages\redis\asyncio\multidb\config.py: if database_config.from_url: venv\Lib\site-packages\redis\asyncio\multidb\config.py: database_config.from_url, **database_config.client_kwargs venv\Lib\site-packages\redis\asyncio\multidb\config.py: elif database_config.from_pool: venv\Lib\site-packages\redis\asyncio\multidb\config.py: database_config.from_pool.set_retry(venv\Lib\site-packages\redis\asyncio\multidb\config.py: connection_pool=database_config.from_pool venv\Lib\site-packages\redis\asyncio\multidb\config.py: client = self.client_class(**database_config.client_kwargs) venv\Lib\site-packages\redis\asyncio\multidb\config.py: database_config.default_circuit_breaker() venv\Lib\site-packages\redis\asyncio\multidb\config.py: if database_config.circuit is None venv\Lib\site-packages\redis\asyncio\multidb\config.py: else database_config.circuit venv\Lib\site-packages\redis\asyncio\multidb\config.py: weight=database_config.weight, venv\Lib\site-packages\redis\asyncio\multidb\config.py: health_check_url=database_config.health_check_url, venv\Lib\site-packages\redis\asyncio\multidb\config.py: database_config.weight, venv\Lib\site-packages\redis\asyncio\multidb\healthcheck.py: "Database health check url is not set. Please check DatabaseConfig for the current database." venv\Lib\site-packages\redis\auth\idp.py: class IdentityProviderConfigInterface(ABC): venv\Lib\site-packages\redis\auth\idp.py: Configuration class that provides a configured identity provider. venv\Lib\site-packages\redis\auth\token_manager.py: class TokenManagerConfig: venv\Lib\site-packages\redis\auth\token_manager.py: self, identity_provider: IdentityProviderInterface, config: TokenManagerConfig venv\Lib\site-packages\redis\auth\token_manager.py: self._config = config venv\Lib\site-

```

packages\redis\auth\token_manager.py: if self._retries < self._config.get_retry_policy().get_max_attempts(): venv\Lib\site-
packages\redis\auth\token_manager.py: sleep(self._config.get_retry_policy().get_delay_in_ms() / 1000) venv\Lib\site-packages\redis\auth\token_manager.py:
if self._retries < self._config.get_retry_policy().get_max_attempts(): venv\Lib\site-packages\redis\auth\token_manager.py:
self._config.get_retry_policy().get_delay_in_ms() / 1000 venv\Lib\site-packages\redis\auth\token_manager.py: - self._config.get_lower_refresh_bound_millis()
venv\Lib\site-packages\redis\auth\token_manager.py: token_ttl * self._config.get_expiration_refresh_ratio() venv\Lib\site-packages\redis\cache.py:class
CacheConfigurationInterface(ABC): venv\Lib\site-packages\redis\cache.py: def config(self) -> CacheConfigurationInterface: venv\Lib\site-
packages\redis\cache.py: cache_config: CacheConfigurationInterface, venv\Lib\site-packages\redis\cache.py: self._cache_config = cache_config venv\Lib\site-
packages\redis\cache.py: self._eviction_policy = self._cache_config.get_eviction_policy().value() venv\Lib\site-packages\redis\cache.py: def config(self) ->
CacheConfigurationInterface: venv\Lib\site-packages\redis\cache.py: return self._cache_config venv\Lib\site-packages\redis\cache.py: if
self._cache_config.is_exceeds_max_size(len(self._cache)): venv\Lib\site-packages\redis\cache.py: return
self._cache_config.is_allowed_to_cache(key.command) venv\Lib\site-packages\redis\cache.py:class CacheConfig(CacheConfigurationInterface):
venv\Lib\site-packages\redis\cache.py: def __init__(self, cache_config: Optional[CacheConfig] = None): venv\Lib\site-packages\redis\cache.py: self._config =
cache_config venv\Lib\site-packages\redis\cache.py: if self._config is None: venv\Lib\site-packages\redis\cache.py: self._config = CacheConfig()
venv\Lib\site-packages\redis\cache.py: cache_class = self._config.get_cache_class() venv\Lib\site-packages\redis\cache.py: return
cache_class(cache_config=self._config) venv\Lib\site-packages\redis\client.py:from redis.cache import CacheConfig, CacheInterface venv\Lib\site-
packages\redis\client.py: MaintNotificationsConfig, venv\Lib\site-packages\redis\client.py: configuration, an instance will either use a ConnectionPool, or
venv\Lib\site-packages\redis\client.py: Return a Redis client object configured from the given URL venv\Lib\site-packages\redis\client.py: cache_config:
Optional[CacheConfig] = None, venv\Lib\site-packages\redis\client.py: maint_notifications_config: Optional[MaintNotificationsConfig] = None,
venv\Lib\site-packages\redis\client.py: 2. Define a Retry object with configured 'supported_errors' and set venv\Lib\site-packages\redis\client.py: When
'connection_pool' is provided - the retry configuration of the venv\Lib\site-packages\redis\client.py: maint_notifications_config: venv\Lib\site-
packages\redis\client.py: configuration the pool to support maintenance notifications - see venv\Lib\site-packages\redis\client.py:
redis.maint_notifications.MaintNotificationsConfig for details. venv\Lib\site-packages\redis\client.py: if (cache_config or cache) and protocol in [3,
"3"]: venv\Lib\site-packages\redis\client.py: "cache_config": cache_config, venv\Lib\site-packages\redis\client.py: maint_notifications_config and
maint_notifications_config.enabled venv\Lib\site-packages\redis\client.py: if maint_notifications_config: venv\Lib\site-packages\redis\client.py:
"maint_notifications_config": maint_notifications_config, venv\Lib\site-packages\redis\client.py: if (cache_config or cache) and
self.connection_pool.get_protocol() not in [ venv\Lib\site-packages\redis\cluster.py:from redis.cache import CacheConfig, CacheFactory,
CacheFactoryInterface, CacheInterface venv\Lib\site-packages\redis\cluster.py:from redis.maint_notifications import MaintNotificationsConfig venv\Lib\site-
packages\redis\cluster.py: "cache_config", venv\Lib\site-packages\redis\cluster.py: "CONFIG SET", venv\Lib\site-packages\redis\cluster.py: "CONFIG
REWRITE", venv\Lib\site-packages\redis\cluster.py: "CONFIG RESETSTAT", venv\Lib\site-packages\redis\cluster.py: "CLUSTER SET-CONFIG-EPOCH",
venv\Lib\site-packages\redis\cluster.py: "CONFIG GET", venv\Lib\site-packages\redis\cluster.py: "FT.CONFIG", venv\Lib\site-packages\redis\cluster.py:
"CONFIG SET", venv\Lib\site-packages\redis\cluster.py: "CONFIG REWRITE", venv\Lib\site-packages\redis\cluster.py: "CONFIG RESETSTAT",
venv\Lib\site-packages\redis\cluster.py: Return a Redis client object configured from the given URL venv\Lib\site-packages\redis\cluster.py: reason="Please
configure the 'load_balancing_strategy' instead", venv\Lib\site-packages\redis\cluster.py: reason="Please configure the 'retry' object instead", venv\Lib\site-
packages\redis\cluster.py: cache_config: Optional[CacheConfig] = None, venv\Lib\site-packages\redis\cluster.py: https://redis.io/topics/cluster-tutorial#redis-
cluster-configuration-parameters venv\Lib\site-packages\redis\cluster.py: @deprecated - Please configure the 'retry' object instead venv\Lib\site-
packages\redis\cluster.py: # and there we provide retry configuration without retries allowed. venv\Lib\site-packages\redis\cluster.py: if (cache_config or
cache) and protocol not in [3, "3"]: venv\Lib\site-packages\redis\cluster.py: cache_config=cache_config, venv\Lib\site-packages\redis\cluster.py: # Sending

```

READONLY command to server to configure connection as `venv\Lib\site-packages\redis\cluster.py: # rc.cluster_save_config(rc.get primaries())`
`venv\Lib\site-packages\redis\cluster.py: config option "self.retry" which defaults to 3 unless manually venv\Lib\site-packages\redis\cluster.py: configured.`
`venv\Lib\site-packages\redis\cluster.py: cache_config: Optional[CacheConfig] = None, venv\Lib\site-packages\redis\cluster.py: self._cache_config =`
`cache_config venv\Lib\site-packages\redis\cluster.py: # We are configuring the connection pool not to retry venv\Lib\site-packages\redis\cluster.py:`
`node_retry_config = Retry(venv\Lib\site-packages\redis\cluster.py: {"maint_notifications_config": MaintNotificationsConfig(enabled=False)} venv\Lib\site-`
`packages\redis\cluster.py: kwargs.update({"retry": node_retry_config}) venv\Lib\site-packages\redis\cluster.py: retry=node_retry_config, venv\Lib\site-`
`packages\redis\cluster.py: if self._cache is None and self._cache_config is not None: venv\Lib\site-packages\redis\cluster.py: self._cache =`
`CacheFactory(self._cache_config).get_cache() venv\Lib\site-packages\redis\cluster.py: reason="Please configure the 'retry' object instead", venv\Lib\site-`
`packages\redis\cluster.py: "CONFIG GET", venv\Lib\site-packages\redis\cluster.py: "CONFIG RESETSTAT", venv\Lib\site-packages\redis\cluster.py:`
`"CONFIG REWRITE", venv\Lib\site-packages\redis\cluster.py: "CONFIG SET", venv\Lib\site-packages\redis\cluster.py: "CONFIG", venv\Lib\site-`
`packages\redis\cluster.py: the retries in config option "self.retry" venv\Lib\site-packages\redis\cluster.py: which defaults to 3 unless manually configured.`
`venv\Lib\site-packages\redis\cluster.py: # rc.cluster_save_config(rc.get primaries()) venv\Lib\site-packages\redis\commands\cluster.py: Get Cluster config for`
`the node. venv\Lib\site-packages\redis\commands\cluster.py: Reconfigure a node as a slave of the specified master node venv\Lib\site-`
`packages\redis\commands\cluster.py: def cluster_save_config(venv\Lib\site-packages\redis\commands\cluster.py: For more information see`
`https://redis.io/commands/cluster-saveconfig venv\Lib\site-packages\redis\commands\cluster.py: return self.execute_command("CLUSTER SAVECONFIG",`
`target_nodes=target_nodes) venv\Lib\site-packages\redis\commands\cluster.py: def cluster_set_config_epoch(venv\Lib\site-`
`packages\redis\commands\cluster.py: Set the configuration epoch in a new node venv\Lib\site-packages\redis\commands\cluster.py: For more information see`
`https://redis.io/commands/cluster-set-config-epoch venv\Lib\site-packages\redis\commands\cluster.py: "CLUSTER SET-CONFIG-EPOCH", epoch,`
`target_nodes=target_nodes venv\Lib\site-packages\redis\commands\core.py: Load ACL rules from the configured `aclfile`. venv\Lib\site-`
`packages\redis\commands\core.py: Note that the server must be configured with the `aclfile` venv\Lib\site-packages\redis\commands\core.py: Save ACL`
`rules to the configured `aclfile`. venv\Lib\site-packages\redis\commands\core.py: Note that the server must be configured with the `aclfile` venv\Lib\site-`
`packages\redis\commands\core.py: def config_get(venv\Lib\site-packages\redis\commands\core.py: Return a dictionary of configuration based on the`
``pattern` venv\Lib\site-packages\redis\commands\core.py: For more information, see https://redis.io/commands/config-get venv\Lib\site-`
`packages\redis\commands\core.py: return self.execute_command("CONFIG GET", pattern, *args, **kwargs) venv\Lib\site-packages\redis\commands\core.py:`
`def config_set(venv\Lib\site-packages\redis\commands\core.py: """Set config item `name` with value venv\Lib\site-packages\redis\commands\core.py: For`
`more information, see https://redis.io/commands/config-set venv\Lib\site-packages\redis\commands\core.py: return self.execute_command("CONFIG SET",`
`name, value, *args, **kwargs) venv\Lib\site-packages\redis\commands\core.py: def config_resetstat(self, **kwargs) -> ResponseT: venv\Lib\site-`
`packages\redis\commands\core.py: For more information, see https://redis.io/commands/config-resetstat venv\Lib\site-packages\redis\commands\core.py:`
`return self.execute_command("CONFIG RESETSTAT", **kwargs) venv\Lib\site-packages\redis\commands\core.py: def config_rewrite(self, **kwargs) ->`
`ResponseT: venv\Lib\site-packages\redis\commands\core.py: Rewrite config file with the minimal change to reflect running config. venv\Lib\site-`
`packages\redis\commands\core.py: For more information, see https://redis.io/commands/config-rewrite venv\Lib\site-packages\redis\commands\core.py: return`
`self.execute_command("CONFIG REWRITE", **kwargs) venv\Lib\site-packages\redis\commands\core.py: """Shutdown the Redis server. If Redis has`
`persistence configured, venv\Lib\site-packages\redis\commands\core.py: `save` will force a DB saving operation even if no save points are`
`configured. venv\Lib\site-packages\redis\commands\core.py: are configured. venv\Lib\site-packages\redis\commands\core.py: """Shutdown the Redis`
`server. If Redis has persistence configured, venv\Lib\site-packages\redis\commands\core.py: configured. If the "nosave" option is set, no data flush will be`
`venv\Lib\site-packages\redis\commands\core.py: Loads a module from a dynamic library at runtime with configuration directives. venv\Lib\site-`
`packages\redis\commands\core.py: pieces.append("CONFIG") venv\Lib\site-packages\redis\commands\search\commands.py: HybridPostProcessingConfig,`


```

venv\Lib\site-packages\redis\commands\search\commands.py:CONFIG_CMD = "FT.CONFIG" venv\Lib\site-packages\redis\commands\search\commands.py:
def _parse_config_get(self, res, **kwargs): venv\Lib\site-packages\redis\commands\search\commands.py: post_processing:
Optional[HybridPostProcessingConfig] = None, venv\Lib\site-packages\redis\commands\search\commands.py: - **post_processing**:
HybridPostProcessingConfig object venv\Lib\site-packages\redis\commands\search\commands.py: Contains the post processing configuration venv\Lib\site-
packages\redis\commands\search\commands.py: - **cursor**: HybridCursorQuery object - contains the cursor configuration venv\Lib\site-
packages\redis\commands\search\commands.py: reason="deprecated since Redis 8.0, call config_set from core module instead", venv\Lib\site-
packages\redis\commands\search\commands.py: def config_set(self, option: str, value: str) -> bool: venv\Lib\site-
packages\redis\commands\search\commands.py: """Set runtime configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: -
**option**: the name of the configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: - **value**: a value for the configuration
option. venv\Lib\site-packages\redis\commands\search\commands.py: For more information see FT.CONFIG SET
<https://redis.io/commands/ft.config-set>_. venv\Lib\site-packages\redis\commands\search\commands.py: cmd = [CONFIG_CMD, "SET", option,
value] venv\Lib\site-packages\redis\commands\search\commands.py: reason="deprecated since Redis 8.0, call config_get from core module instead",
venv\Lib\site-packages\redis\commands\search\commands.py: def config_get(self, option: str) -> str: venv\Lib\site-
packages\redis\commands\search\commands.py: """Get runtime configuration option value. venv\Lib\site-packages\redis\commands\search\commands.py: -
**option**: the name of the configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: For more information see FT.CONFIG GET
<https://redis.io/commands/ft.config-get>_. venv\Lib\site-packages\redis\commands\search\commands.py: cmd = [CONFIG_CMD, "GET", option]
venv\Lib\site-packages\redis\commands\search\commands.py: return self._parse_results(CONFIG_CMD, res) venv\Lib\site-
packages\redis\commands\search\commands.py: post_processing: Optional[HybridPostProcessingConfig] = None, venv\Lib\site-
packages\redis\commands\search\commands.py: - **post_processing**: HybridPostProcessingConfig object venv\Lib\site-
packages\redis\commands\search\commands.py: Contains the post processing configuration venv\Lib\site-packages\redis\commands\search\commands.py: -
**cursor**: HybridCursorQuery object - contains the cursor configuration venv\Lib\site-packages\redis\commands\search\commands.py: reason="deprecated
since Redis 8.0, call config_set from core module instead", venv\Lib\site-packages\redis\commands\search\commands.py: async def config_set(self, option: str,
value: str) -> bool: venv\Lib\site-packages\redis\commands\search\commands.py: """Set runtime configuration option. venv\Lib\site-
packages\redis\commands\search\commands.py: - **option**: the name of the configuration option. venv\Lib\site-
packages\redis\commands\search\commands.py: - **value**: a value for the configuration option. venv\Lib\site-
packages\redis\commands\search\commands.py: For more information see FT.CONFIG SET <https://redis.io/commands/ft.config-set>_.
venv\Lib\site-packages\redis\commands\search\commands.py: cmd = [CONFIG_CMD, "SET", option, value] venv\Lib\site-
packages\redis\commands\search\commands.py: reason="deprecated since Redis 8.0, call config_get from core module instead", venv\Lib\site-
packages\redis\commands\search\commands.py: async def config_get(self, option: str) -> str: venv\Lib\site-packages\redis\commands\search\commands.py:
"""Get runtime configuration option value. venv\Lib\site-packages\redis\commands\search\commands.py: - **option**: the name of the configuration option.
venv\Lib\site-packages\redis\commands\search\commands.py: For more information see FT.CONFIG GET <https://redis.io/commands/ft.config-get>_.
venv\Lib\site-packages\redis\commands\search\commands.py: cmd = [CONFIG_CMD, "GET", option] venv\Lib\site-
packages\redis\commands\search\commands.py: return self._parse_results(CONFIG_CMD, res) venv\Lib\site-
packages\redis\commands\search\hybrid_query.py:class HybridPostProcessingConfig: venv\Lib\site-packages\redis\commands\search\hybrid_query.py: Create
a new hybrid post processing configuration object. venv\Lib\site-packages\redis\commands\search\__init__.py: CONFIG_CMD, venv\Lib\site-
packages\redis\commands\search\__init__.py: CONFIG_CMD: self._parse_config_get, venv\Lib\site-packages\redis\commands\sentinel.py: configuration will
be published so that the other Sentinels will venv\Lib\site-packages\redis\commands\sentinel.py: update their configurations). venv\Lib\site-

```

`packages\redis\commands\sentinel.py`: Check if the current Sentinel configuration is able to reach the `venv\Lib\site-packages\redis\commands\sentinel.py`: `def sentinel_flushconfig(self)`: `venv\Lib\site-packages\redis\commands\sentinel.py`: Force Sentinel to rewrite its configuration on disk, including the `venv\Lib\site-packages\redis\commands\sentinel.py`: Normally Sentinel rewrites the configuration every time something `venv\Lib\site-packages\redis\commands\sentinel.py`: However sometimes it is possible that the configuration file is lost `venv\Lib\site-packages\redis\commands\sentinel.py`: configuration managers. In those cases a way to to force Sentinel to `venv\Lib\site-packages\redis\commands\sentinel.py`: rewrite the configuration file is handy. `venv\Lib\site-packages\redis\commands\sentinel.py`: This command works even if the previous configuration file is `venv\Lib\site-packages\redis\commands\sentinel.py`: `return self.execute_command("SENTINEL FLUSHCONFIG")` `venv\Lib\site-packages\redis\commands\timeseries\info.py`: https://redis.io/docs/latest/develop/data-types/timeseries/configuration/#duplicate_policy `venv\Lib\site-packages\redis\connection.py`: `MaintNotificationsConfig`, `venv\Lib\site-packages\redis\connection.py`: # the command name, e.g., 'CONFIG GET'. The Redis server expects these `venv\Lib\site-packages\redis\connection.py`: `maint_notifications_config`: `Optional[MaintNotificationsConfig]`, `venv\Lib\site-packages\redis\connection.py`: `maint_notifications_config` (`MaintNotificationsConfig`): The configuration for maintenance notifications. `venv\Lib\site-packages\redis\connection.py`: `self.maint_notifications_config = maint_notifications_config` `venv\Lib\site-packages\redis\connection.py`: `self._configure_maintenance_notifications()` `venv\Lib\site-packages\redis\connection.py`: `def _configure_maintenance_notifications()` `venv\Lib\site-packages\redis\connection.py`: not `self.maint_notifications_config` `venv\Lib\site-packages\redis\connection.py`: or not `self.maint_notifications_config.enabled` `venv\Lib\site-packages\redis\connection.py`: "To configure maintenance notifications, a parser must be provided!" `venv\Lib\site-packages\redis\connection.py`: `MaintNotificationsConnectionHandler(self, self.maint_notifications_config)` `venv\Lib\site-packages\redis\connection.py`: `self`, `maint_notifications_pool_handler.config` `venv\Lib\site-packages\redis\connection.py`: `self._maint_notifications_connection_handler.config = (` `venv\Lib\site-packages\redis\connection.py`: `maint_notifications_pool_handler.config` `venv\Lib\site-packages\redis\connection.py`: # When the `maint_notifications_config` enabled mode is "auto", `venv\Lib\site-packages\redis\connection.py`: and `self.maint_notifications_config` `venv\Lib\site-packages\redis\connection.py`: and `self.maint_notifications_config.enabled` `venv\Lib\site-packages\redis\connection.py`: `maint_notifications_config=self.maint_notifications_config`, `venv\Lib\site-packages\redis\connection.py`: `self`, `maint_notifications_config`: `MaintNotificationsConfig`, `check_health=True` `venv\Lib\site-packages\redis\connection.py`: `endpoint_type = maint_notifications_config.get_endpoint_type(host, self)` `venv\Lib\site-packages\redis\connection.py`: and `maint_notifications_config.enabled == "auto"` `venv\Lib\site-packages\redis\connection.py`: `maint_notifications_config`: `Optional[MaintNotificationsConfig] = None`, `venv\Lib\site-packages\redis\connection.py`: `maint_notifications_config`, `venv\Lib\site-packages\redis\connection.py`: # if enabled in the configuration `venv\Lib\site-packages\redis\connection.py`: `self._conn.maint_notifications_config`, `venv\Lib\site-packages\redis\connection.py`: `maint_notifications_config`: `Optional[MaintNotificationsConfig] = None`, `venv\Lib\site-packages\redis\connection.py`: if `maint_notifications_config` is `None` and `is_protocol_supported`: `venv\Lib\site-packages\redis\connection.py`: `maint_notifications_config = MaintNotificationsConfig()` `venv\Lib\site-packages\redis\connection.py`: if `maint_notifications_config` and `maint_notifications_config.enabled`: `venv\Lib\site-packages\redis\connection.py`: `self`, `maint_notifications_config` `venv\Lib\site-packages\redis\connection.py`: The maintenance notifications config is stored in the pool handler. `venv\Lib\site-packages\redis\connection.py`: `maint_notifications_config = (` `venv\Lib\site-packages\redis\connection.py`: `self._maint_notifications_pool_handler.config` `venv\Lib\site-packages\redis\connection.py`: `return maint_notifications_config` and `maint_notifications_config.enabled` `venv\Lib\site-packages\redis\connection.py`: `def update_maint_notifications_config()` `venv\Lib\site-packages\redis\connection.py`: `self`, `maint_notifications_config`: `MaintNotificationsConfig` `venv\Lib\site-packages\redis\connection.py`: Updates the maintenance notifications configuration. `venv\Lib\site-packages\redis\connection.py`: and not `maint_notifications_config.enabled` `venv\Lib\site-packages\redis\connection.py`: `self`, `maint_notifications_config` `venv\Lib\site-packages\redis\connection.py`: `self._maint_notifications_pool_handler.config = maint_notifications_config` `venv\Lib\site-packages\redis\connection.py`: `self._update_maint_notifications_configs_for_connections()` `venv\Lib\site-packages\redis\connection.py`: "maint_notifications_config": `maint_notifications_pool_handler.config`, `venv\Lib\site-packages\redis\connection.py`: # configured the original values yet `venv\Lib\site-`

```

packages\redis\connection.py: def _update_maint_notifications_configs_for_connections( venv\Lib\site-packages\redis\connection.py: """Update the
maintenance notifications config for all connections in the pool.""" venv\Lib\site-packages\redis\connection.py: conn.maint_notifications_config = (
venv\Lib\site-packages\redis\connection.py: maint_notifications_pool_handler.config venv\Lib\site-packages\redis\connection.py:
conn.maint_notifications_config = ( venv\Lib\site-packages\redis\connection.py: maint_notifications_pool_handler.config venv\Lib\site-
packages\redis\connection.py: "connected_address", "configured_address", "notification_hash" venv\Lib\site-packages\redis\connection.py: elif
matching_pattern == "configured_address": venv\Lib\site-packages\redis\connection.py: "connected_address", "configured_address", "notification_hash"
venv\Lib\site-packages\redis\connection.py: If `maint_notifications_config` is provided, the connection pool will support venv\Lib\site-
packages\redis\connection.py: If the `maint_notifications_config` is not provided but the protocol is 3, venv\Lib\site-
packages\redis\connection.py: Return a connection pool configured from the given URL. venv\Lib\site-packages\redis\connection.py:
maint_notifications_config: Optional[MaintNotificationsConfig] = None, venv\Lib\site-packages\redis\connection.py: if
connection_kwargs.get("cache_config") or connection_kwargs.get("cache"): venv\Lib\site-packages\redis\connection.py:
self.connection_kwargs.get("cache_config") venv\Lib\site-packages\redis\connection.py: connection_kwargs.pop("cache_config", None) venv\Lib\site-
packages\redis\connection.py: maint_notifications_config=maint_notifications_config, venv\Lib\site-packages\redis\http\http_client.py: retry: Retry
configuration for failed requests. venv\Lib\site-packages\redis\http\http_client.py: # Load client certificate for mTLS if configured venv\Lib\site-
packages\redis\maint_notifications.py:class MaintNotificationsConfig: venv\Lib\site-packages\redis\maint_notifications.py: Configuration class for
maintenance notifications handling behaviour. Notifications are received through venv\Lib\site-packages\redis\maint_notifications.py: Initialize a new
MaintNotificationsConfig. venv\Lib\site-packages\redis\maint_notifications.py: If None, the endpoint type will be automatically determined based on the host
and TLS configuration. venv\Lib\site-packages\redis\maint_notifications.py: config: MaintNotificationsConfig, venv\Lib\site-
packages\redis\maint_notifications.py: self.config = config venv\Lib\site-packages\redis\maint_notifications.py: copy =
MaintNotificationsPoolHandler(self.pool, self.config) venv\Lib\site-packages\redis\maint_notifications.py: not self.config.proactive_reconnect venv\Lib\site-
packages\redis\maint_notifications.py: and not self.config.is_relaxed_timeouts_enabled() venv\Lib\site-packages\redis\maint_notifications.py:
self.config.proactive_reconnect venv\Lib\site-packages\redis\maint_notifications.py: or self.config.is_relaxed_timeouts_enabled() venv\Lib\site-
packages\redis\maint_notifications.py: relaxed_timeout=self.config.relaxed_timeout, venv\Lib\site-packages\redis\maint_notifications.py: if
self.config.proactive_reconnect: venv\Lib\site-packages\redis\maint_notifications.py: # Update config for new connections: venv\Lib\site-
packages\redis\maint_notifications.py: if self.config.is_relaxed_timeouts_enabled(): venv\Lib\site-packages\redis\maint_notifications.py: "socket_timeout":
self.config.relaxed_timeout, venv\Lib\site-packages\redis\maint_notifications.py: "socket_connect_timeout": self.config.relaxed_timeout, venv\Lib\site-
packages\redis\maint_notifications.py: reset_relaxed_timeout = self.config.is_relaxed_timeouts_enabled() venv\Lib\site-packages\redis\maint_notifications.py:
reset_host_address = self.config.proactive_reconnect venv\Lib\site-packages\redis\maint_notifications.py: config: MaintNotificationsConfig, venv\Lib\site-
packages\redis\maint_notifications.py: self.config = config venv\Lib\site-packages\redis\maint_notifications.py: or not
self.config.is_relaxed_timeouts_enabled() venv\Lib\site-packages\redis\maint_notifications.py: tmp_relaxed_timeout=self.config.relaxed_timeout
venv\Lib\site-packages\redis\maint_notifications.py: self.connection.update_current_socket_timeout(self.config.relaxed_timeout) venv\Lib\site-
packages\redis\maint_notifications.py: or not self.config.is_relaxed_timeouts_enabled() venv\Lib\site-packages\redis\multidb\client.py: from
redis.multidb.config import DEFAULT_GRACE_PERIOD, MultiDbConfig venv\Lib\site-packages\redis\multidb\client.py: def __init__(self, config:
MultiDbConfig): venv\Lib\site-packages\redis\multidb\client.py: self.databases = config.databases() venv\Lib\site-packages\redis\multidb\client.py:
config.default_health_checks() venv\Lib\site-packages\redis\multidb\client.py: if not config.health_checks venv\Lib\site-packages\redis\multidb\client.py: else
config.health_checks venv\Lib\site-packages\redis\multidb\client.py: self.health_check_interval = config.health_check_interval venv\Lib\site-
packages\redis\multidb\client.py: self.health_check_policy: HealthCheckPolicy = config.health_check_policy.value( venv\Lib\site-

```

packages\redis\multidb\client.py: config.health_check_probes, config.health_check_probes_delay venv\Lib\site-packages\redis\multidb\client.py:
 config.default_failure_detectors() venv\Lib\site-packages\redis\multidb\client.py: if not config.failure_detectors venv\Lib\site-packages\redis\multidb\client.py:
 else config.failure_detectors venv\Lib\site-packages\redis\multidb\client.py: config.default_failover_strategy() venv\Lib\site-packages\redis\multidb\client.py:
 if config.failover_strategy is None venv\Lib\site-packages\redis\multidb\client.py: else config.failover_strategy venv\Lib\site-packages\redis\multidb\client.py:
 self._auto_fallback_interval = config.auto_fallback_interval venv\Lib\site-packages\redis\multidb\client.py: self._event_dispatcher = config.event_dispatcher
 venv\Lib\site-packages\redis\multidb\client.py: self._command_retry = config.command_retry venv\Lib\site-packages\redis\multidb\client.py:
 failover_attempts=config.failover_attempts, venv\Lib\site-packages\redis\multidb\client.py: failover_delay=config.failover_delay, venv\Lib\site-
 packages\redis\multidb\client.py: self._config = config venv\Lib\site-packages\redis\multidb\command_executor.py:from redis.multidb.config import
 DEFAULT_AUTO_FALLBACK_INTERVAL venv\Lib\site-packages\redis\multidb\config.py:class DatabaseConfig: venv\Lib\site-
 packages\redis\multidb\config.py: Dataclass representing the configuration for a database connection. venv\Lib\site-packages\redis\multidb\config.py: This
 class is used to store configuration settings for a database connection, venv\Lib\site-packages\redis\multidb\config.py: attributes and allows for the creation of
 customized configurations for various venv\Lib\site-packages\redis\multidb\config.py: from_pool (Optional[ConnectionPool]): A pre-configured connection
 pool to use. venv\Lib\site-packages\redis\multidb\config.py:class MultiDbConfig: venv\Lib\site-packages\redis\multidb\config.py: Configuration class for
 managing multiple database connections in a resilient and fail-safe manner. venv\Lib\site-packages\redis\multidb\config.py: databases_config: A list of
 database configurations. venv\Lib\site-packages\redis\multidb\config.py: Retrieves a collection of database clients managed by weighted configurations.
 venv\Lib\site-packages\redis\multidb\config.py: Initializes database clients based on the provided configuration and removes venv\Lib\site-
 packages\redis\multidb\config.py: with defined retry and backoff configurations. venv\Lib\site-packages\redis\multidb\config.py: databases_config:
 List[DatabaseConfig] venv\Lib\site-packages\redis\multidb\config.py: for database_config in self.databases_config: venv\Lib\site-
 packages\redis\multidb\config.py: database_config.client_kwargs.update(venv\Lib\site-packages\redis\multidb\config.py: if database_config.from_url:
 venv\Lib\site-packages\redis\multidb\config.py: database_config.from_url, **database_config.client_kwargs venv\Lib\site-packages\redis\multidb\config.py:
 elif database_config.from_pool: venv\Lib\site-packages\redis\multidb\config.py: database_config.from_pool.set_retry(venv\Lib\site-
 packages\redis\multidb\config.py: connection_pool=database_config.from_pool venv\Lib\site-packages\redis\multidb\config.py: client =
 self.client_class(**database_config.client_kwargs) venv\Lib\site-packages\redis\multidb\config.py: database_config.default_circuit_breaker() venv\Lib\site-
 packages\redis\multidb\config.py: if database_config.circuit is None venv\Lib\site-packages\redis\multidb\config.py: else database_config.circuit venv\Lib\site-
 packages\redis\multidb\config.py: weight=database_config.weight, venv\Lib\site-packages\redis\multidb\config.py:
 health_check_url=database_config.health_check_url, venv\Lib\site-packages\redis\multidb\config.py: database_config.weight, venv\Lib\site-
 packages\redis\multidb\healthcheck.py: "Database health check url is not set. Please check DatabaseConfig for the current database." venv\Lib\site-
 packages\redis_parsers\base.py:# user send an AUTH cmd to a server without authorization configured venv\Lib\site-packages\redis_parsers\base.py:
 "configured for the default user. Are you sure " venv\Lib\site-packages\redis_parsers\base.py: "your configuration is correct?": AuthenticationError,
 venv\Lib\site-packages\redis_parsers\helpers.py: "config-epoch": int, venv\Lib\site-packages\redis_parsers\helpers.py:def parse_config_get(response,
 **options): venv\Lib\site-packages\redis_parsers\helpers.py: "CLUSTER SAVECONFIG": bool_ok, venv\Lib\site-packages\redis_parsers\helpers.py:
 "CLUSTER SET-CONFIG-EPOCH": bool_ok, venv\Lib\site-packages\redis_parsers\helpers.py: "CONFIG RESETSTAT": bool_ok, venv\Lib\site-
 packages\redis_parsers\helpers.py: "CONFIG SET": bool_ok, venv\Lib\site-packages\redis_parsers\helpers.py: "SENTINEL FLUSHCONFIG": bool_ok,
 venv\Lib\site-packages\redis_parsers\helpers.py: "CONFIG GET": parse_config_get, venv\Lib\site-packages\redis_parsers\helpers.py: "CONFIG GET":
 lambda r: { venv\Lib\site-packages\rest_framework\apps.py:from django.apps import AppConfig venv\Lib\site-packages\rest_framework\apps.py:class
 RestFrameworkConfig(AppConfig): venv\Lib\site-packages\rest_framework\authtoken\apps.py:from django.apps import AppConfig venv\Lib\site-
 packages\rest_framework\authtoken\apps.py:class AuthTokenConfig(AppConfig): venv\Lib\site-packages\rest_framework\checks.py:def

```

pagination_system_check(app_configs, **kwargs): venv\Lib\site-packages\rest_framework\compat.py: extension_configs = { venv\Lib\site-
packages\rest_framework\compat.py: extensions=extensions, extension_configs=extension_configs venv\Lib\site-packages\rest_framework\fields.py: Quantize
the decimal value to the configured precision. venv\Lib\site-packages\rest_framework\filters.py: from django.core.exceptions import FieldDoesNotExist,
ImproperlyConfigured venv\Lib\site-packages\rest_framework\filters.py: raise ImproperlyConfigured(msg % self.__class__.__name__) venv\Lib\site-
packages\rest_framework\generics.py: method if you want to apply the configured filtering backend to the venv\Lib\site-
packages\rest_framework\management\commands\generateschema.py: help = "Generates configured API schema for project." venv\Lib\site-
packages\rest_framework\pagination.py: page object, or None if pagination is not configured for this view. venv\Lib\site-
packages\rest_framework\relations.py: from django.core.exceptions import ImproperlyConfigured, ObjectDoesNotExist venv\Lib\site-
packages\rest_framework\relations.py: attributes are not configured to correctly match the URL conf. venv\Lib\site-packages\rest_framework\relations.py: #
Return the hyperlink, or error if incorrectly configured. venv\Lib\site-packages\rest_framework\relations.py: 'model in your API, or incorrectly configured the '
venv\Lib\site-packages\rest_framework\relations.py: raise ImproperlyConfigured(msg % self.view_name) venv\Lib\site-
packages\rest_framework\renderers.py: from django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\renderers.py: raise
ImproperlyConfigured( venv\Lib\site-packages\rest_framework\routers.py: from django.core.exceptions import ImproperlyConfigured venv\Lib\site-
packages\rest_framework\routers.py: raise ImproperlyConfigured(msg) venv\Lib\site-packages\rest_framework\routers.py: raise ImproperlyConfigured(msg %
', '.join(not_allowed)) venv\Lib\site-packages\rest_framework\serializers.py: from django.core.exceptions import FieldDoesNotExist, ImproperlyConfigured
venv\Lib\site-packages\rest_framework\serializers.py: raise ImproperlyConfigured( venv\Lib\site-packages\rest_framework\settings.py: # View configuration
venv\Lib\site-packages\rest_framework\test.py: from django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\test.py:
raise ImproperlyConfigured('requests must be installed in order to use RequestsClient.') venv\Lib\site-packages\rest_framework\test.py: raise
ImproperlyConfigured('coreapi must be installed in order to use CoreAPIClient.') venv\Lib\site-packages\rest_framework\throttling.py: from
django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\throttling.py: raise ImproperlyConfigured(msg) venv\Lib\site-
packages\rest_framework\throttling.py: raise ImproperlyConfigured(msg) venv\Lib\site-packages\rest_framework\views.py: # Filter out None values which
may happen in case of config / rate venv\Lib\site-packages\sqlparse\lexer.py: """The Lexer supports configurable syntax. venv\Lib\site-
packages\sqlparse\lexer.py: """Clear all syntax configurations. venv\Lib\site-packages\sqlparse\lexer.py: Useful if you want to load a reduced set of syntax
configurations. venv\Lib\site-packages\tutorial\asgi.py: ASGI config for tutorial project. venv\Lib\site-packages\tutorial\quickstart\apps.py: class
QuickstartConfig(AppConfig): venv\Lib\site-packages\tutorial\wsgi.py: WSGI config for tutorial project. (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> python manage.py makemigrations Migrations for 'chat': chat\migrations\0001_initial.py + Create model Room + Create
model Message (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py migrate Operations to perform: Apply all
migrations: admin, auth, chat, contenttypes, sessions Running migrations: Applying contenttypes.0001_initial... OK Applying auth.0001_initial... OK Applying
admin.0001_initial... OK Applying admin.0002_logentry_remove_auto_add... OK Applying admin.0003_logentry_add_action_flag_choices... OK Applying
contenttypes.0002_remove_content_type_name... OK Applying auth.0002_alter_permission_name_max_length... OK Applying
auth.0003_alter_user_email_max_length... OK Applying auth.0004_alter_user_username_opts... OK Applying auth.0005_alter_user_last_login_null... OK
Applying auth.0006_require_contenttypes_0002... OK Applying auth.0008_alter_user_username_max_length... OK Applying
auth.0010_alter_group_name_max_length... OK Applying auth.0011_update_proxy_permissions... OK Applying
auth.0012_alter_user_first_name_max_length... OK Applying chat.0001_initial... OK Applying sessions.0001_initial... OK (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> python -c "from django.core.asgi import get_asgi_application; print('django ok')" django ok (venv) PS D:\PYTHON
DJANGO HOPES\Chat_API_DJANGO_mario> python -c "import chat.routing; print('routing ok')" Traceback (most recent call last): File "<string>", line 1, in
<module> File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\routing.py", line 2, in <module> from .consumers import ChatConsumer

```

```

File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\consumers.py", line 4, in <module> from models import Room, Message
File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\models.py", line 3, in <module> class Room(models.Model):
File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\db\models\base.py", line 131, in __new__
app_config = apps.get_containing_app_config(module)
File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\apps\registry.py", line 260, in get_containing_app_config
self.check_apps_ready()
File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\apps\registry.py", line 137, in check_apps_ready
File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\conf\__init__.py", line 81, in __getattr__
self._setup(name)
File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\conf\__init__.py", line 61, in _setup
raise ImproperlyConfigured(
django.core.exceptions.ImproperlyConfigured: Requested setting INSTALLED_APPS, but settings are not configured. You must either define the environment variable DJANGO_SETTINGS_MODULE or call settings.configure() before accessing settings.
(venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python -c "import os; os.environ.setdefault('DJANGO_SETTINGS_MODULE','monsite.settings'); import django; django.setup(); import chat.routing; print('routing ok')"
routing ok
(venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py makemigrations
No changes detected
(venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py migrate
Operations to perform: Apply all migrations: admin, auth, chat, contenttypes, sessions
Running migrations: No migrations to apply.
(venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> py manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
February 07, 2026 - 16:54:06
Django version 5.2.8, using settings 'monsite.settings'
Starting development server at http://127.0.0.1:8000/
WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: https://docs.djangoproject.com/en/5.2/howto/deployment/
Not Found: / [07/Feb/2026 16:54:39] "GET / HTTP/1.1" 404 2305
(venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python -m pip show channels >> ^C
(venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python -m pip show channels
Name: channels
Version: 4.1.0
Author: Django Software Foundation
License: BSD
Location: D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages
Requires: asgiref, Django
Required-by: channels-redis
(venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py makemigrations
(venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py migrate
Operations to perform: Apply all migrations: admin, auth, chat, contenttypes, sessions
Running migrations:
(venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py makemigrations
(venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py migrate
Operations to perform: Apply all migrations: admin, auth, chat, contenttypes, sessions
Running migrations:
(venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py makemigrations
No changes detected
(venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py migrate
Operations to perform: Apply all migrations: admin, auth, chat, contenttypes, sessions
Running migrations: No migrations to apply.
(venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> py manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
February 07, 2026 - 17:03:21
Django version 5.2.8, using settings 'monsite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: https://docs.djangoproject.com/en/5.2/howto/deployment/
Not Found: / [07/Feb/2026 17:03:24] "GET / HTTP/1.1" 404 2305
Not Found: / [07/Feb/2026 17:03:25] "GET / HTTP/1.1" 404 2305
Not Found: / [07/Feb/2026 17:03:27] "GET / HTTP/1.1" 404 2305
[07/Feb/2026 17:04:21] "GET /admin/ HTTP/1.1" 302 0
[07/Feb/2026 17:04:22] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 4252
[07/Feb/2026 17:04:22] "GET /static/admin/css/nav_sidebar.css HTTP/1.1" 200 2810
[07/Feb/2026 17:04:22] "GET /static/admin/css/base.css HTTP/1.1" 200 22120
[07/Feb/2026 17:04:22] "GET /static/admin/css/login.css HTTP/1.1" 200 951
[07/Feb/2026 17:04:22] "GET /static/admin/css/dark_mode.css HTTP/1.1" 200 2808
[07/Feb/2026 17:04:22] "GET /static/admin/css/responsive.css HTTP/1.1" 200 16565
[07/Feb/2026 17:04:22] "GET /static/admin/js/theme.js HTTP/1.1" 200 1653

```

[07/Feb/2026 17:04:22] "GET /static/admin/js/nav_sidebar.js HTTP/1.1" 200 3063 Not Found: /favicon.ico [07/Feb/2026 17:04:22] "GET /favicon.ico HTTP/1.1" 404 2356 Not Found: /api/ [07/Feb/2026 17:04:37] "GET /api/ HTTP/1.1" 404 2739 Not Found: / [07/Feb/2026 17:04:48] "GET / HTTP/1.1" 404 2305 [07/Feb/2026 17:05:01] "GET /admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:05:01] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 4252 Not Found: /api/ [07/Feb/2026 17:05:12] "GET /api/ HTTP/1.1" 404 2739 Not Found: / [07/Feb/2026 17:24:49] "GET /api/rooms/ HTTP/1.1" 200 2 Not Found: /favicon.ico [07/Feb/2026 17:24:49] "GET /favicon.ico HTTP/1.1" 404 2356 (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> curl -X POST http://127.0.0.1:8000/api/rooms/ -H "Content-Type: application/json" -d '{"name":"general"}' >> ^C (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> curl -X POST http://127.0.0.1:8000/api/rooms/ -H "Content-Type: application/json" -d '{"name":"general"}' Invoke-WebRequest : Impossible de lier le paramètre «Headers». Impossible de convertir la valeur «Content-Type: application/json» du type «System.String» en type «System.Collections.IDictionary». Au caractère Ligne:1 : 50 + ~~~~~ + CategoryInfo : InvalidArgument : (:) [Invoke-WebRequest], ParameterBindingException + FullyQualifiedErrorId : CannotConvertArgumentNoMessage,Microsoft.PowerShell.Commands.InvokeWebRequestCommand (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/" >> -Method Post >> -ContentType "application/json" Invoke-RestMethod : Impossible de se connecter au serveur distant Au caractère Ligne:1 : 1 + Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/" + ~~~~~ + CategoryInfo : NotSpecified: (:) [Invoke-RestMethod], WebException + FullyQualifiedErrorId : System.Net.WebException,Microsoft.PowerShell.Commands.InvokeRestMethodCommand (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> py manage.py runserver Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 17:30:20 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: https://docs.djangoproject.com/en/5.2/howto/deployment/ Not Found: / [07/Feb/2026 17:30:24] "GET / HTTP/1.1" 404 2305 Not Found: / [07/Feb/2026 17:30:25] "GET / HTTP/1.1" 404 2305 Not Found: /ws/chat/general/ [07/Feb/2026 17:31:01] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:31:04] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:31:24] "GET /api/rooms/ HTTP/1.1" 200 2 Not Found: /ws/chat/general/ [07/Feb/2026 17:31:32] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:31:58] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:32:00] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:32:02] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:32:04] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:21] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:22] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:22] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:32] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:33] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:35] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /api/ \u0222 WS: ws://127.0.0.1:8000/ws/chat/<room>/ [07/Feb/2026 17:35:44] "GET /api/%20%E2%80%A2%20WS:%20ws://127.0.0.1:8000/ws/chat/%3Croom%3E/ HTTP/1.1" 404 2899 Not Found: /api/ [07/Feb/2026 17:35:51] "GET /api/ HTTP/1.1" 404 2739 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:43] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:44] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:44] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:45] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:46] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:47] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:38:30] "GET /api/rooms/ HTTP/1.1" 200 2 Not Found: /favicon.ico [07/Feb/2026 17:38:30] "GET /favicon.ico HTTP/1.1" 404 2356 Not Found: / [07/Feb/2026 17:38:52] "GET / HTTP/1.1" 404 2305 [07/Feb/2026 17:41:00] "GET /admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:41:00] "GET

```

/admin/login/?next=/admin/ HTTP/1.1" 200 4252 [07/Feb/2026 17:41:01] "GET /static/admin/css/dark_mode.css HTTP/1.1" 200 2808 [07/Feb/2026 17:41:01]
"GET /static/admin/css/base.css HTTP/1.1" 200 22120 [07/Feb/2026 17:41:01] "GET /static/admin/css/nav_sidebar.css HTTP/1.1" 200 2810 [07/Feb/2026
17:41:01] "GET /static/admin/js/theme.js HTTP/1.1" 200 1653 [07/Feb/2026 17:41:01] "GET /static/admin/js/nav_sidebar.js HTTP/1.1" 200 3063
[07/Feb/2026 17:41:01] "GET /static/admin/css/login.css HTTP/1.1" 200 951 [07/Feb/2026 17:41:01] "GET /static/admin/css/responsive.css HTTP/1.1" 200
16565 [07/Feb/2026 17:41:35] "POST /admin/login/?next=/admin/ HTTP/1.1" 200 4498 [07/Feb/2026 17:44:04] "POST /admin/login/?next=/admin/
HTTP/1.1" 302 0 [07/Feb/2026 17:44:04] "GET /admin/ HTTP/1.1" 200 7885 [07/Feb/2026 17:44:05] "GET /static/admin/css/dashboard.css HTTP/1.1" 200
441 [07/Feb/2026 17:44:05] "GET /static/admin/img/icon-changelink.svg HTTP/1.1" 200 380 [07/Feb/2026 17:44:05] "GET /static/admin/img/icon-
addlink.svg HTTP/1.1" 200 331 [07/Feb/2026 17:44:27] "GET /admin/chat/room/add/ HTTP/1.1" 200 9637 [07/Feb/2026 17:44:27] "GET
/static/admin/js/jquery.init.js HTTP/1.1" 200 347 [07/Feb/2026 17:44:27] "GET /static/admin/css/forms.css HTTP/1.1" 200 8525 [07/Feb/2026 17:44:27] "GET
/static/admin/js/core.js HTTP/1.1" 200 6208 [07/Feb/2026 17:44:27] "GET /static/admin/js/urlify.js HTTP/1.1" 200 7887 [07/Feb/2026 17:44:27] "GET
/static/admin/js/prepopulate.js HTTP/1.1" 200 1531 [07/Feb/2026 17:44:27] "GET /admin/jsi18n/ HTTP/1.1" 200 8634 [07/Feb/2026 17:44:27] "GET
/static/admin/js/change_form.js HTTP/1.1" 200 606 [07/Feb/2026 17:44:27] "GET /static/admin/js/admin/RelatedObjectLookups.js HTTP/1.1" 200 9777
[07/Feb/2026 17:44:27] "GET /static/admin/js/actions.js HTTP/1.1" 200 8076 [07/Feb/2026 17:44:27] "GET /static/admin/js/prepopulate_init.js HTTP/1.1" 200
586 [07/Feb/2026 17:44:27] "GET /static/admin/css/widgets.css HTTP/1.1" 200 11973 [07/Feb/2026 17:44:28] "GET
/static/admin/js/vendor/xregexp/xregexp.js HTTP/1.1" 200 325171 [07/Feb/2026 17:44:28] "GET /static/admin/js/vendor/jquery/jquery.js HTTP/1.1" 200
285314 [07/Feb/2026 17:44:44] "POST /admin/chat/room/add/ HTTP/1.1" 302 0 [07/Feb/2026 17:44:44] "GET /admin/chat/room/ HTTP/1.1" 200 10874
[07/Feb/2026 17:44:45] "GET /admin/jsi18n/ HTTP/1.1" 200 8634 [07/Feb/2026 17:44:45] "GET /static/admin/css/changelists.css HTTP/1.1" 200 6878
[07/Feb/2026 17:44:45] "GET /static/admin/js/filters.js HTTP/1.1" 200 978 [07/Feb/2026 17:44:45] "GET /static/admin/img/search.svg HTTP/1.1" 200 458
[07/Feb/2026 17:44:45] "GET /static/admin/img/tooltag-add.svg HTTP/1.1" 200 331 [07/Feb/2026 17:44:45] "GET /static/admin/img/icon-yes.svg HTTP/1.1"
200 436 Not Found: / [07/Feb/2026 17:44:58] "GET / HTTP/1.1" 404 2305 Not Found: / [07/Feb/2026 17:44:59] "GET / HTTP/1.1" 404 2305 Not Found:
/ws/chat/general/ [07/Feb/2026 17:45:18] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:45:22] "GET
/ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:06] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found:
/ws/chat/general/ [07/Feb/2026 17:47:08] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:09] "GET
/ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:38] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found:
/ws/chat/general/ [07/Feb/2026 17:47:39] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:41] "GET
/ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:52] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026
17:47:53] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 17:47:53] "GET /api/rooms/1/messages/ HTTP/1.1" 200 2 [07/Feb/2026 17:47:54] "GET
/api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 17:47:54] "GET /api/rooms/1/messages/ HTTP/1.1" 200 2 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:57]
"GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:50:18] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /favicon.ico [07/Feb/2026 17:50:18]
"GET /favicon.ico HTTP/1.1" 404 2356 [07/Feb/2026 17:50:21] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /favicon.ico [07/Feb/2026 17:50:21] "GET
/favicon.ico HTTP/1.1" 404 2356 Not Found: /ws/chat/general/ [07/Feb/2026 17:53:33] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found:
/ws/chat/general/ [07/Feb/2026 17:53:34] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:53:34] "GET
/ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:55:20] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /favicon.ico [07/Feb/2026 17:55:21] "GET
/favicon.ico HTTP/1.1" 404 2356 [07/Feb/2026 17:56:52] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-
known/appspecific/com.chrome.devtools.json [07/Feb/2026 17:57:19] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2467 Not
Found: /ws/chat/general/ [07/Feb/2026 17:58:15] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /.well-known/appspecific/com.chrome.devtools.json
[07/Feb/2026 17:59:50] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2467 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:09]

```


"GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:11] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:12] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:18] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:25] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:25] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:26] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 18:00:27] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 18:00:27] "GET /api/rooms/1/messages/ HTTP/1.1" 200 2 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:28] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:29] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:07:52] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:07:53] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:08:00] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:08:00] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:08:01] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:08:28] "GET /ws/chat/general/ HTTP/1.1" 404 2371
Réflexion durant 37s

Tu es **très proche** : ton backend REST marche (tu as bien `/api/rooms/` et tu as créé un Room via l'admin), mais le point qui te bloque c'est le **WebSocket**.

1) Pourquoi `http://127.0.0.1:8000/ws/chat/general/` renvoie 404 ?

C'est **normal** si tu tapes ça dans la barre d'adresse du navigateur.

Un WebSocket **ne s'ouvre pas** en visitant l'URL comme une page HTTP classique.

Le navigateur doit envoyer une requête spéciale avec l'entête **Upgrade: websocket** (handshake).

Quand tu tapes l'URL à la main, c'est un **GET HTTP normal**, donc Django regarde `urls.py` (admin/, api/) et renvoie 404.

✅ La bonne URL à utiliser côté client est `ws://127.0.0.1:8000/ws/chat/general/` (pas `http://...`) et **via du code JS** (ou un client ws).

2) Ton `asgi.py` / `settings.py` sont OK (pour Channels)

D'après tes fichiers :

- `ASGI_APPLICATION = "monsite.asgi.application"` ✅
- `"channels"` dans `INSTALLED_APPS` ✅
- ton `asgi.py` fait bien `get_asgi_application()` puis `import chat.routing` ✅
Donc la partie "routing websocket" est bien câblée.

3) Comment “lancer index.html” dans Django (la page qui ouvre le WS)

Le plus simple : servir une page d’accueil / qui contient le JS WebSocket.

Option A — via un template Django (recommandé)

1. Crée : chat/templates/chat/index.html

```
<!doctype html>
<html>
<head><meta charset="utf-8"><title>Chat</title></head>
<body>
  <h1>Room: general</h1>
  <input id="msg" placeholder="message..." />
  <button id="send">Envoyer</button>
  <pre id="log"></pre>

  <script>
    const log = (t) => document.getElementById("log").textContent += t + "\n";
    const ws = new WebSocket("ws://127.0.0.1:8000/ws/chat/general/");

    ws.onopen = () => log("✅ WS connecté");
    ws.onmessage = (e) => log("📄 " + e.data);
    ws.onclose = () => log("❌ WS fermé");
    ws.onerror = (e) => log("⚠️ WS erreur");

    document.getElementById("send").onclick = () => {
      const content = document.getElementById("msg").value;
      ws.send(JSON.stringify({author:"Mario", content}));
    };
  </script>
</body>
</html>
```

2. Dans chat/views.py

```
from django.shortcuts import render

def index(request):
```

```
return render(request, "chat/index.html")
```

3. Dans `monsite/urls.py` ajoute la route / :

```
from django.contrib import admin
from django.urls import path, include
from chat import views

urlpatterns = [
    path("", views.index),
    path("admin/", admin.site.urls),
    path("api/", include("chat.urls")),
]
```

Ensuite :

- `py manage.py runserver`
- ouvre <http://127.0.0.1:8000/>
→ Là tu verras “WS connecté” si tout est bon.

4) Tester le WebSocket correctement (sans navigateur)

Test Python propre (évite les soucis de guillemets PowerShell)

Crée un fichier `test_ws.py` :

```
import asyncio, json, websockets

async def main():
    async with websockets.connect("ws://127.0.0.1:8000/ws/chat/general/") as ws:
        await ws.send(json.dumps({"author": "test", "content": "hello"}))
        print("sent")
        msg = await ws.recv()
        print("recv:", msg)

asyncio.run(main())
```

Puis lance :

py test_ws.py

5) Ton `curl` PowerShell a échoué : normal aussi 😊

Dans PowerShell, `curl` = alias de `Invoke-WebRequest`, donc `-H` ne marche pas pareil.




Utilise soit **`curl.exe`** :

```
curl.exe -X POST "http://127.0.0.1:8000/api/rooms/" -H "Content-Type: application/json" -d '{"name":"general"}"
```



Soit `Invoke-RestMethod` **correct** :

```
Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/" -Method Post `
-headers @{ "Content-Type" = "application/json" } `
-Body '{"name":"general"}'
```



En résumé


-  `ws://...` **ne se visite pas** en HTTP → 404 normal
-  il faut une **page HTML/JS** ou un **client ws**
-  ton ASGI/Channels est déjà bien câblé → il reste surtout la partie **client**

Si tu veux, envoie aussi ton `chat/routing.py` et `chat/consumers.py` (copier/coller) et je te dis exactement si ton pattern `/ws/chat/<room>/` **matche bien** et si ton consumer renvoie quelque chose.

Room: general message... Envoyer  WS erreur  WS fermé

Réflexion durant 1m 26s

Ton WS “ **erreur** /  **fermé**” vient presque sûrement du fait que ton serveur lancé avec `python manage.py runserver` est **le runserver “HTTP classique”** (ou celui de `staticfiles`) et **ne gère pas le WebSocket**, donc la tentative de connexion à `ws://127.0.0.1:8000/ws/chat/general/` reçoit un **404** → le client WS ferme.

 **Fix (le plus important) : installer + activer Daphne**

Dans Django/Channels récents, c'est **Daphne** qui fournit le serveur ASGI (et le runserver compatible WS).

1. Installer daphne

```
pip install daphne
```

2. Modifier `INSTALLED_APPS` dans `settings.py` (important : daphne tout en haut, avant `staticfiles`)

```
INSTALLED_APPS = [  
    "daphne", # 📁 tout en haut  
  
    "django.contrib.admin",  
    "django.contrib.auth",  
    "django.contrib.contenttypes",  
    "django.contrib.sessions",  
    "django.contrib.messages",  
    "django.contrib.staticfiles",  
  
    "corsheaders",  
    "rest_framework",  
    "channels",  
  
    "chat",  
]
```

3. Vérifie aussi que tu as bien (tu l'as déjà normalement) :

```
ASGI_APPLICATION = "monsite.asgi.application"
```

4. Relance le serveur

```
py manage.py runserver
```

➡ Si c'est bon, tu devrais voir un démarrage lié à **ASGI/Daphne** (et quand un client WS se connecte, tu n'auras plus un 404 sur `/ws/...`).

 **Point important : ne teste pas le WS en tapant l'URL dans le navigateur**

`http://127.0.0.1:8000/ws/chat/general/` = requête HTTP → normal que Django URLconf te mette 404.

Le WS doit être testé via :

- ton `index.html` (JS WebSocket),
 - ou un client WS.
-

Test WebSocket propre (PowerShell)

Le plus simple : crée un fichier `ws_test.py` :

```
import asyncio, websockets

async def main():
    uri = "ws://127.0.0.1:8000/ws/chat/general/"
    async with websockets.connect(uri) as ws:
        await ws.send('{"author":"test","content":"hello"}')
        print("RECV:", await ws.recv())

asyncio.run(main())
```

Puis :

```
py ws_test.py
```

Bonus : ton “Room: general” existe (c’est OK)

Tu as bien créé la room via l’admin (on voit `/api/rooms/` renvoyer autre chose que `[]` ensuite). Donc la partie REST/DB est OK : il te manquait surtout **le serveur ASGI qui accepte l’upgrade WebSocket**.

Si après ça tu as encore “WS fermé”, colle-moi :

1. le **log complet** au démarrage après `runserver`
2. ton `chat/routing.py` et `chat/consumers.py` (juste le contenu) et je te dis exactement où ça bloque.

Room: general  WS connecté

Parfait 🎉 Ça veut dire que :

- **Daphne/ASGI + Channels** tournent bien
- Ton **routing WS** est pris en compte
- L'URL `ws://127.0.0.1:8000/ws/chat/general/` répond correctement

Maintenant, pour vérifier que l'envoi/réception marche vraiment

1. Ouvre **2 onglets** de ton `index.html`
2. Dans l'onglet A, écris un message → **Envoyer**
3. Tu dois voir le message apparaître dans l'onglet B (broadcast via le groupe Channels)

Si tu veux voir les logs côté serveur

Dans `ChatConsumer`, ajoute un `print` (ex) :

```
print("WS connect:", self.scope["url_route"]["kwargs"])
```

et dans `receive` :

```
print("WS receive:", text_data)
```

Tu me dis : quand tu “Envoyer”, tu vois quoi côté client (message affiché / erreur) et côté console serveur ?

```
venv\Lib\site-packages\redis\asyncio\multidb\config.py: health_check_url=database_config.health_check_url, venv\Lib\site-packages\redis\asyncio\multidb\config.py: database_config.weight, venv\Lib\site-packages\redis\asyncio\multidb\healthcheck.py: "Database health check url is not set. Please check DatabaseConfig for the current database." venv\Lib\site-packages\redis\auth\idp.py:class IdentityProviderConfigInterface(ABC): venv\Lib\site-packages\redis\auth\idp.py: Configuration class that provides a configured identity provider. venv\Lib\site-packages\redis\auth\token_manager.py:class TokenManagerConfig: venv\Lib\site-packages\redis\auth\token_manager.py: self, identity_provider: IdentityProviderInterface, config: TokenManagerConfig venv\Lib\site-packages\redis\auth\token_manager.py: self._config = config venv\Lib\site-packages\redis\auth\token_manager.py: if self._retries < self._config.get_retry_policy().get_max_attempts(): venv\Lib\site-packages\redis\auth\token_manager.py: sleep(self._config.get_retry_policy().get_delay_in_ms() / 1000) venv\Lib\site-packages\redis\auth\token_manager.py: if self._retries < self._config.get_retry_policy().get_max_attempts(): venv\Lib\site-packages\redis\auth\token_manager.py: self._config.get_retry_policy().get_delay_in_ms() / 1000 venv\Lib\site-packages\redis\auth\token_manager.py: - self._config.get_lower_refresh_bound_millis() venv\Lib\site-packages\redis\auth\token_manager.py: token_ttl * self._config.get_expiration_refresh_ratio() venv\Lib\site-packages\redis\cache.py:class
```

```

CacheConfigurationInterface(ABC): venv\Lib\site-packages\redis\cache.py: def config(self) -> CacheConfigurationInterface: venv\Lib\site-
packages\redis\cache.py: cache_config: CacheConfigurationInterface, venv\Lib\site-packages\redis\cache.py: self._cache_config = cache_config venv\Lib\site-
packages\redis\cache.py: self._eviction_policy = self._cache_config.get_eviction_policy().value() venv\Lib\site-packages\redis\cache.py: def config(self) ->
CacheConfigurationInterface: venv\Lib\site-packages\redis\cache.py: return self._cache_config venv\Lib\site-packages\redis\cache.py: if
self._cache_config.is_exceeds_max_size(len(self._cache)): venv\Lib\site-packages\redis\cache.py: return
self._cache_config.is_allowed_to_cache(key.command) venv\Lib\site-packages\redis\cache.py: class CacheConfig(CacheConfigurationInterface):
venv\Lib\site-packages\redis\cache.py: def __init__(self, cache_config: Optional[CacheConfig] = None): venv\Lib\site-packages\redis\cache.py: self._config =
cache_config venv\Lib\site-packages\redis\cache.py: if self._config is None: venv\Lib\site-packages\redis\cache.py: self._config = CacheConfig()
venv\Lib\site-packages\redis\cache.py: cache_class = self._config.get_cache_class() venv\Lib\site-packages\redis\cache.py: return
cache_class(cache_config=self._config) venv\Lib\site-packages\redis\client.py: from redis.cache import CacheConfig, CacheInterface venv\Lib\site-
packages\redis\client.py: MaintNotificationsConfig, venv\Lib\site-packages\redis\client.py: configuration, an instance will either use a ConnectionPool, or
venv\Lib\site-packages\redis\client.py: Return a Redis client object configured from the given URL venv\Lib\site-packages\redis\client.py: cache_config:
Optional[CacheConfig] = None, venv\Lib\site-packages\redis\client.py: maint_notifications_config: Optional[MaintNotificationsConfig] = None,
venv\Lib\site-packages\redis\client.py: 2. Define a Retry object with configured 'supported_errors' and set venv\Lib\site-packages\redis\client.py: When
'connection_pool' is provided - the retry configuration of the venv\Lib\site-packages\redis\client.py: maint_notifications_config: venv\Lib\site-
packages\redis\client.py: configuration the pool to support maintenance notifications - see venv\Lib\site-packages\redis\client.py:
redis.maint_notifications.MaintNotificationsConfig for details. venv\Lib\site-packages\redis\client.py: if (cache_config or cache) and protocol in [3,
"3"]: venv\Lib\site-packages\redis\client.py: "cache_config": cache_config, venv\Lib\site-packages\redis\client.py: maint_notifications_config and
maint_notifications_config.enabled venv\Lib\site-packages\redis\client.py: if maint_notifications_config: venv\Lib\site-packages\redis\client.py:
"maint_notifications_config": maint_notifications_config, venv\Lib\site-packages\redis\client.py: if (cache_config or cache) and
self.connection_pool.get_protocol() not in [ venv\Lib\site-packages\redis\cluster.py: from redis.cache import CacheConfig, CacheFactory,
CacheFactoryInterface, CacheInterface venv\Lib\site-packages\redis\cluster.py: from redis.maint_notifications import MaintNotificationsConfig venv\Lib\site-
packages\redis\cluster.py: "cache_config", venv\Lib\site-packages\redis\cluster.py: "CONFIG SET", venv\Lib\site-packages\redis\cluster.py: "CONFIG
REWRITE", venv\Lib\site-packages\redis\cluster.py: "CONFIG RESETSTAT", venv\Lib\site-packages\redis\cluster.py: "CLUSTER SET-CONFIG-EPOCH",
venv\Lib\site-packages\redis\cluster.py: "CONFIG GET", venv\Lib\site-packages\redis\cluster.py: "FT.CONFIG", venv\Lib\site-packages\redis\cluster.py:
"CONFIG SET", venv\Lib\site-packages\redis\cluster.py: "CONFIG REWRITE", venv\Lib\site-packages\redis\cluster.py: "CONFIG RESETSTAT",
venv\Lib\site-packages\redis\cluster.py: Return a Redis client object configured from the given URL venv\Lib\site-packages\redis\cluster.py: reason="Please
configure the 'load_balancing_strategy' instead", venv\Lib\site-packages\redis\cluster.py: reason="Please configure the 'retry' object instead", venv\Lib\site-
packages\redis\cluster.py: cache_config: Optional[CacheConfig] = None, venv\Lib\site-packages\redis\cluster.py: https://redis.io/topics/cluster-tutorial#redis-
cluster-configuration-parameters venv\Lib\site-packages\redis\cluster.py: @deprecated - Please configure the 'retry' object instead venv\Lib\site-
packages\redis\cluster.py: # and there we provide retry configuration without retries allowed. venv\Lib\site-packages\redis\cluster.py: if (cache_config or
cache) and protocol not in [3, "3"]: venv\Lib\site-packages\redis\cluster.py: cache_config=cache_config, venv\Lib\site-packages\redis\cluster.py: # Sending
READONLY command to server to configure connection as venv\Lib\site-packages\redis\cluster.py: # rc.cluster_save_config(rc.get_primaries())
venv\Lib\site-packages\redis\cluster.py: config option "self.retry" which defaults to 3 unless manually venv\Lib\site-packages\redis\cluster.py: configured.
venv\Lib\site-packages\redis\cluster.py: cache_config: Optional[CacheConfig] = None, venv\Lib\site-packages\redis\cluster.py: self._cache_config =
cache_config venv\Lib\site-packages\redis\cluster.py: # We are configuring the connection pool not to retry venv\Lib\site-packages\redis\cluster.py:
node_retry_config = Retry( venv\Lib\site-packages\redis\cluster.py: {"maint_notifications_config": MaintNotificationsConfig(enabled=False)} venv\Lib\site-

```


packages\redis\cluster.py: kwargs.update({"retry": node_retry_config}) venv\Lib\site-packages\redis\cluster.py: retry=node_retry_config, venv\Lib\site-packages\redis\cluster.py: if self._cache is None and self._cache_config is not None: venv\Lib\site-packages\redis\cluster.py: self._cache = CacheFactory(self._cache_config).get_cache() venv\Lib\site-packages\redis\cluster.py: reason="Please configure the 'retry' object instead", venv\Lib\site-packages\redis\cluster.py: "CONFIG GET", venv\Lib\site-packages\redis\cluster.py: "CONFIG RESETSTAT", venv\Lib\site-packages\redis\cluster.py: "CONFIG REWRITE", venv\Lib\site-packages\redis\cluster.py: "CONFIG SET", venv\Lib\site-packages\redis\cluster.py: "CONFIG", venv\Lib\site-packages\redis\cluster.py: the retries in config option "self.retry" venv\Lib\site-packages\redis\cluster.py: which defaults to 3 unless manually configured. venv\Lib\site-packages\redis\cluster.py: # rc.cluster_save_config(rc.get_primaries()) venv\Lib\site-packages\redis\commands\cluster.py: Get Cluster config for the node. venv\Lib\site-packages\redis\commands\cluster.py: Reconfigure a node as a slave of the specified master node venv\Lib\site-packages\redis\commands\cluster.py: def cluster_save_config(venv\Lib\site-packages\redis\commands\cluster.py: For more information see <https://redis.io/commands/cluster-saveconfig> venv\Lib\site-packages\redis\commands\cluster.py: return self.execute_command("CLUSTER SAVECONFIG", target_nodes=target_nodes) venv\Lib\site-packages\redis\commands\cluster.py: def cluster_set_config_epoch(venv\Lib\site-packages\redis\commands\cluster.py: Set the configuration epoch in a new node venv\Lib\site-packages\redis\commands\cluster.py: For more information see <https://redis.io/commands/cluster-set-config-epoch> venv\Lib\site-packages\redis\commands\cluster.py: "CLUSTER SET-CONFIG-EPOCH", epoch, target_nodes=target_nodes venv\Lib\site-packages\redis\commands\core.py: Load ACL rules from the configured `aclfile`. venv\Lib\site-packages\redis\commands\core.py: Note that the server must be configured with the `aclfile` venv\Lib\site-packages\redis\commands\core.py: Save ACL rules to the configured `aclfile`. venv\Lib\site-packages\redis\commands\core.py: Note that the server must be configured with the `aclfile` venv\Lib\site-packages\redis\commands\core.py: def config_get(venv\Lib\site-packages\redis\commands\core.py: Return a dictionary of configuration based on the `pattern` venv\Lib\site-packages\redis\commands\core.py: For more information, see <https://redis.io/commands/config-get> venv\Lib\site-packages\redis\commands\core.py: return self.execute_command("CONFIG GET", pattern, *args, **kwargs) venv\Lib\site-packages\redis\commands\core.py: def config_set(venv\Lib\site-packages\redis\commands\core.py: """Set config item `name` with value venv\Lib\site-packages\redis\commands\core.py: For more information, see <https://redis.io/commands/config-set> venv\Lib\site-packages\redis\commands\core.py: return self.execute_command("CONFIG SET", name, value, *args, **kwargs) venv\Lib\site-packages\redis\commands\core.py: def config_resetstat(self, **kwargs) -> ResponseT: venv\Lib\site-packages\redis\commands\core.py: For more information, see <https://redis.io/commands/config-resetstat> venv\Lib\site-packages\redis\commands\core.py: return self.execute_command("CONFIG RESETSTAT", **kwargs) venv\Lib\site-packages\redis\commands\core.py: def config_rewrite(self, **kwargs) -> ResponseT: venv\Lib\site-packages\redis\commands\core.py: Rewrite config file with the minimal change to reflect running config. venv\Lib\site-packages\redis\commands\core.py: For more information, see <https://redis.io/commands/config-rewrite> venv\Lib\site-packages\redis\commands\core.py: return self.execute_command("CONFIG REWRITE", **kwargs) venv\Lib\site-packages\redis\commands\core.py: """Shutdown the Redis server. If Redis has persistence configured, venv\Lib\site-packages\redis\commands\core.py: `save` will force a DB saving operation even if no save points are configured. venv\Lib\site-packages\redis\commands\core.py: are configured. venv\Lib\site-packages\redis\commands\core.py: """Shutdown the Redis server. If Redis has persistence configured, venv\Lib\site-packages\redis\commands\core.py: configured. If the "nosave" option is set, no data flush will be venv\Lib\site-packages\redis\commands\core.py: Loads a module from a dynamic library at runtime with configuration directives. venv\Lib\site-packages\redis\commands\core.py: pieces.append("CONFIG") venv\Lib\site-packages\redis\commands\search\commands.py: HybridPostProcessingConfig, venv\Lib\site-packages\redis\commands\search\commands.py: CONFIG_CMD = "FT.CONFIG" venv\Lib\site-packages\redis\commands\search\commands.py: def _parse_config_get(self, res, **kwargs): venv\Lib\site-packages\redis\commands\search\commands.py: post_processing: Optional[HybridPostProcessingConfig] = None, venv\Lib\site-packages\redis\commands\search\commands.py: - **post_processing**: HybridPostProcessingConfig object venv\Lib\site-packages\redis\commands\search\commands.py: Contains the post processing configuration venv\Lib\site-packages\redis\commands\search\commands.py: - **cursor**: HybridCursorQuery object - contains the cursor configuration venv\Lib\site-

packages\redis\commands\search\commands.py: reason="deprecated since Redis 8.0, call config_set from core module instead", venv\Lib\site-packages\redis\commands\search\commands.py: def config_set(self, option: str, value: str) -> bool: venv\Lib\site-packages\redis\commands\search\commands.py: """Set runtime configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: - **option**: the name of the configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: - **value**: a value for the configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: For more information see FT.CONFIG SET <<https://redis.io/commands/ft.config-set>>_. venv\Lib\site-packages\redis\commands\search\commands.py: cmd = [CONFIG_CMD, "SET", option, value] venv\Lib\site-packages\redis\commands\search\commands.py: reason="deprecated since Redis 8.0, call config_get from core module instead", venv\Lib\site-packages\redis\commands\search\commands.py: def config_get(self, option: str) -> str: venv\Lib\site-packages\redis\commands\search\commands.py: """Get runtime configuration option value. venv\Lib\site-packages\redis\commands\search\commands.py: - **option**: the name of the configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: For more information see FT.CONFIG GET <<https://redis.io/commands/ft.config-get>>_. venv\Lib\site-packages\redis\commands\search\commands.py: cmd = [CONFIG_CMD, "GET", option] venv\Lib\site-packages\redis\commands\search\commands.py: return self._parse_results(CONFIG_CMD, res) venv\Lib\site-packages\redis\commands\search\commands.py: post_processing: Optional[HybridPostProcessingConfig] = None, venv\Lib\site-packages\redis\commands\search\commands.py: - **post_processing**: HybridPostProcessingConfig object venv\Lib\site-packages\redis\commands\search\commands.py: Contains the post processing configuration venv\Lib\site-packages\redis\commands\search\commands.py: - **cursor**: HybridCursorQuery object - contains the cursor configuration venv\Lib\site-packages\redis\commands\search\commands.py: reason="deprecated since Redis 8.0, call config_set from core module instead", venv\Lib\site-packages\redis\commands\search\commands.py: async def config_set(self, option: str, value: str) -> bool: venv\Lib\site-packages\redis\commands\search\commands.py: """Set runtime configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: - **option**: the name of the configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: - **value**: a value for the configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: For more information see FT.CONFIG SET <<https://redis.io/commands/ft.config-set>>_. venv\Lib\site-packages\redis\commands\search\commands.py: cmd = [CONFIG_CMD, "SET", option, value] venv\Lib\site-packages\redis\commands\search\commands.py: reason="deprecated since Redis 8.0, call config_get from core module instead", venv\Lib\site-packages\redis\commands\search\commands.py: async def config_get(self, option: str) -> str: venv\Lib\site-packages\redis\commands\search\commands.py: """Get runtime configuration option value. venv\Lib\site-packages\redis\commands\search\commands.py: - **option**: the name of the configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: For more information see FT.CONFIG GET <<https://redis.io/commands/ft.config-get>>_. venv\Lib\site-packages\redis\commands\search\commands.py: cmd = [CONFIG_CMD, "GET", option] venv\Lib\site-packages\redis\commands\search\commands.py: return self._parse_results(CONFIG_CMD, res) venv\Lib\site-packages\redis\commands\search\hybrid_query.py: class HybridPostProcessingConfig: venv\Lib\site-packages\redis\commands\search\hybrid_query.py: Create a new hybrid post processing configuration object. venv\Lib\site-packages\redis\commands\search__init__.py: CONFIG_CMD, venv\Lib\site-packages\redis\commands\search__init__.py: CONFIG_CMD: self._parse_config_get, venv\Lib\site-packages\redis\commands\sentinel.py: configuration will be published so that the other Sentinels will venv\Lib\site-packages\redis\commands\sentinel.py: update their configurations). venv\Lib\site-packages\redis\commands\sentinel.py: Check if the current Sentinel configuration is able to reach the venv\Lib\site-packages\redis\commands\sentinel.py: def sentinel_flushconfig(self): venv\Lib\site-packages\redis\commands\sentinel.py: Force Sentinel to rewrite its configuration on disk, including the venv\Lib\site-packages\redis\commands\sentinel.py: Normally Sentinel rewrites the configuration every time something venv\Lib\site-packages\redis\commands\sentinel.py: However sometimes it is possible that the configuration file is lost venv\Lib\site-packages\redis\commands\sentinel.py: configuration managers. In those cases a way to force Sentinel to venv\Lib\site-packages\redis\commands\sentinel.py: rewrite the configuration file is handy. venv\Lib\site-

packages\redis\commands\sentinel.py: This command works even if the previous configuration file is venv\Lib\site-packages\redis\commands\sentinel.py:
 return self.execute_command("SENTINEL FLUSHCONFIG") venv\Lib\site-packages\redis\commands\timeseries\info.py:
https://redis.io/docs/latest/develop/data-types/timeseries/configuration/#duplicate_policy venv\Lib\site-packages\redis\connection.py:
 MaintNotificationsConfig, venv\Lib\site-packages\redis\connection.py: # the command name, e.g., 'CONFIG GET'. The Redis server expects these
 venv\Lib\site-packages\redis\connection.py: maint_notifications_config: Optional[MaintNotificationsConfig], venv\Lib\site-packages\redis\connection.py:
 maint_notifications_config (MaintNotificationsConfig): The configuration for maintenance notifications. venv\Lib\site-packages\redis\connection.py:
 self.maint_notifications_config = maint_notifications_config venv\Lib\site-packages\redis\connection.py: self._configure_maintenance_notifications(
 venv\Lib\site-packages\redis\connection.py: def _configure_maintenance_notifications(venv\Lib\site-packages\redis\connection.py: not
 self.maint_notifications_config venv\Lib\site-packages\redis\connection.py: or not self.maint_notifications_config.enabled venv\Lib\site-
 packages\redis\connection.py: "To configure maintenance notifications, a parser must be provided!" venv\Lib\site-packages\redis\connection.py:
 MaintNotificationsConnectionHandler(self, self.maint_notifications_config) venv\Lib\site-packages\redis\connection.py: self,
 maint_notifications_pool_handler.config venv\Lib\site-packages\redis\connection.py: self._maint_notifications_connection_handler.config = (venv\Lib\site-
 packages\redis\connection.py: maint_notifications_pool_handler.config venv\Lib\site-packages\redis\connection.py: # When the maint_notifications_config
 enabled mode is "auto", venv\Lib\site-packages\redis\connection.py: and self.maint_notifications_config venv\Lib\site-packages\redis\connection.py: and
 self.maint_notifications_config.enabled venv\Lib\site-packages\redis\connection.py: maint_notifications_config=self.maint_notifications_config, venv\Lib\site-
 packages\redis\connection.py: self, maint_notifications_config: MaintNotificationsConfig, check_health=True venv\Lib\site-packages\redis\connection.py:
 endpoint_type = maint_notifications_config.get_endpoint_type(host, self) venv\Lib\site-packages\redis\connection.py: and maint_notifications_config.enabled
 == "auto" venv\Lib\site-packages\redis\connection.py: maint_notifications_config: Optional[MaintNotificationsConfig] = None, venv\Lib\site-
 packages\redis\connection.py: maint_notifications_config, venv\Lib\site-packages\redis\connection.py: # if enabled in the configuration venv\Lib\site-
 packages\redis\connection.py: self._conn.maint_notifications_config, venv\Lib\site-packages\redis\connection.py: maint_notifications_config:
 Optional[MaintNotificationsConfig] = None, venv\Lib\site-packages\redis\connection.py: if maint_notifications_config is None and is_protocol_supported:
 venv\Lib\site-packages\redis\connection.py: maint_notifications_config = MaintNotificationsConfig() venv\Lib\site-packages\redis\connection.py: if
 maint_notifications_config and maint_notifications_config.enabled: venv\Lib\site-packages\redis\connection.py: self, maint_notifications_config venv\Lib\site-
 packages\redis\connection.py: The maintenance notifications config is stored in the pool handler. venv\Lib\site-packages\redis\connection.py:
 maint_notifications_config = (venv\Lib\site-packages\redis\connection.py: self._maint_notifications_pool_handler.config venv\Lib\site-
 packages\redis\connection.py: return maint_notifications_config and maint_notifications_config.enabled venv\Lib\site-packages\redis\connection.py: def
 update_maint_notifications_config(venv\Lib\site-packages\redis\connection.py: self, maint_notifications_config: MaintNotificationsConfig venv\Lib\site-
 packages\redis\connection.py: Updates the maintenance notifications configuration. venv\Lib\site-packages\redis\connection.py: and not
 maint_notifications_config.enabled venv\Lib\site-packages\redis\connection.py: self, maint_notifications_config venv\Lib\site-packages\redis\connection.py:
 self._maint_notifications_pool_handler.config = maint_notifications_config venv\Lib\site-packages\redis\connection.py:
 self._update_maint_notifications_configs_for_connections(venv\Lib\site-packages\redis\connection.py: "maint_notifications_config":
 maint_notifications_pool_handler.config, venv\Lib\site-packages\redis\connection.py: # configured the original values yet venv\Lib\site-
 packages\redis\connection.py: def _update_maint_notifications_configs_for_connections(venv\Lib\site-packages\redis\connection.py: """Update the
 maintenance notifications config for all connections in the pool.""" venv\Lib\site-packages\redis\connection.py: conn.maint_notifications_config = (
 venv\Lib\site-packages\redis\connection.py: maint_notifications_pool_handler.config venv\Lib\site-packages\redis\connection.py:
 conn.maint_notifications_config = (venv\Lib\site-packages\redis\connection.py: maint_notifications_pool_handler.config venv\Lib\site-
 packages\redis\connection.py: "connected_address", "configured_address", "notification_hash" venv\Lib\site-packages\redis\connection.py: elif

matching_pattern == "configured_address": venv\Lib\site-packages\redis\connection.py: "connected_address", "configured_address", "notification_hash"
venv\Lib\site-packages\redis\connection.py: If `maint_notifications_config` is provided, the connection pool will support venv\Lib\site-packages\redis\connection.py: If the `maint_notifications_config` is not provided but the protocol is 3, venv\Lib\site-packages\redis\connection.py: Return a connection pool configured from the given URL. venv\Lib\site-packages\redis\connection.py: `maint_notifications_config`: Optional[MaintNotificationsConfig] = None, venv\Lib\site-packages\redis\connection.py: if `connection_kwargs.get("cache_config")` or `connection_kwargs.get("cache")`: venv\Lib\site-packages\redis\connection.py: `self._connection_kwargs.get("cache_config")` venv\Lib\site-packages\redis\connection.py: `connection_kwargs.pop("cache_config", None)` venv\Lib\site-packages\redis\connection.py: `maint_notifications_config=maint_notifications_config`, venv\Lib\site-packages\redis\http\http_client.py: `retry`: Retry configuration for failed requests. venv\Lib\site-packages\redis\http\http_client.py: # Load client certificate for mTLS if configured venv\Lib\site-packages\redis\maint_notifications.py: class MaintNotificationsConfig: venv\Lib\site-packages\redis\maint_notifications.py: Configuration class for maintenance notifications handling behaviour. Notifications are received through venv\Lib\site-packages\redis\maint_notifications.py: Initialize a new MaintNotificationsConfig. venv\Lib\site-packages\redis\maint_notifications.py: If None, the endpoint type will be automatically determined based on the host and TLS configuration. venv\Lib\site-packages\redis\maint_notifications.py: `config`: MaintNotificationsConfig, venv\Lib\site-packages\redis\maint_notifications.py: `self.config = config` venv\Lib\site-packages\redis\maint_notifications.py: `copy = MaintNotificationsPoolHandler(self.pool, self.config)` venv\Lib\site-packages\redis\maint_notifications.py: `not self.config.proactive_reconnect` venv\Lib\site-packages\redis\maint_notifications.py: `and not self.config.is_relaxed_timeouts_enabled()` venv\Lib\site-packages\redis\maint_notifications.py: `self.config.proactive_reconnect` venv\Lib\site-packages\redis\maint_notifications.py: `or self.config.is_relaxed_timeouts_enabled()` venv\Lib\site-packages\redis\maint_notifications.py: `relaxed_timeout=self.config.relaxed_timeout`, venv\Lib\site-packages\redis\maint_notifications.py: if `self.config.proactive_reconnect`: venv\Lib\site-packages\redis\maint_notifications.py: # Update config for new connections: venv\Lib\site-packages\redis\maint_notifications.py: if `self.config.is_relaxed_timeouts_enabled()`: venv\Lib\site-packages\redis\maint_notifications.py: "socket_timeout": `self.config.relaxed_timeout`, venv\Lib\site-packages\redis\maint_notifications.py: "socket_connect_timeout": `self.config.relaxed_timeout`, venv\Lib\site-packages\redis\maint_notifications.py: `reset_relaxed_timeout = self.config.is_relaxed_timeouts_enabled()` venv\Lib\site-packages\redis\maint_notifications.py: `reset_host_address = self.config.proactive_reconnect` venv\Lib\site-packages\redis\maint_notifications.py: `config`: MaintNotificationsConfig, venv\Lib\site-packages\redis\maint_notifications.py: `self.config = config` venv\Lib\site-packages\redis\maint_notifications.py: `or not self.config.is_relaxed_timeouts_enabled()` venv\Lib\site-packages\redis\maint_notifications.py: `tmp_relaxed_timeout=self.config.relaxed_timeout` venv\Lib\site-packages\redis\maint_notifications.py: `self.connection.update_current_socket_timeout(self.config.relaxed_timeout)` venv\Lib\site-packages\redis\maint_notifications.py: `or not self.config.is_relaxed_timeouts_enabled()` venv\Lib\site-packages\redis\multidb\client.py: from `redis.multidb.config` import `DEFAULT_GRACE_PERIOD`, `MultiDbConfig` venv\Lib\site-packages\redis\multidb\client.py: `def __init__(self, config: MultiDbConfig)`: venv\Lib\site-packages\redis\multidb\client.py: `self._databases = config.databases()` venv\Lib\site-packages\redis\multidb\client.py: `config.default_health_checks()` venv\Lib\site-packages\redis\multidb\client.py: if `not config.health_checks` venv\Lib\site-packages\redis\multidb\client.py: else `config.health_checks` venv\Lib\site-packages\redis\multidb\client.py: `self._health_check_interval = config.health_check_interval` venv\Lib\site-packages\redis\multidb\client.py: `self._health_check_policy`: `HealthCheckPolicy = config.health_check_policy.value()` venv\Lib\site-packages\redis\multidb\client.py: `config.health_check_probes`, `config.health_check_probes_delay` venv\Lib\site-packages\redis\multidb\client.py: `config.default_failure_detectors()` venv\Lib\site-packages\redis\multidb\client.py: if `not config.failure_detectors` venv\Lib\site-packages\redis\multidb\client.py: else `config.failure_detectors` venv\Lib\site-packages\redis\multidb\client.py: `config.default_failover_strategy()` venv\Lib\site-packages\redis\multidb\client.py: if `config.failover_strategy` is None venv\Lib\site-packages\redis\multidb\client.py: else `config.failover_strategy` venv\Lib\site-packages\redis\multidb\client.py: `self._auto_fallback_interval = config.auto_fallback_interval` venv\Lib\site-packages\redis\multidb\client.py: `self._event_dispatcher = config.event_dispatcher`

venv\Lib\site-packages\redis\multidb\client.py: self._command_retry = config.command_retry venv\Lib\site-packages\redis\multidb\client.py: failover_attempts=config.failover_attempts, venv\Lib\site-packages\redis\multidb\client.py: failover_delay=config.failover_delay, venv\Lib\site-packages\redis\multidb\client.py: self._config = config venv\Lib\site-packages\redis\multidb\command_executor.py: from redis.multidb.config import DEFAULT_AUTO_FALLBACK_INTERVAL venv\Lib\site-packages\redis\multidb\config.py: class DatabaseConfig: venv\Lib\site-packages\redis\multidb\config.py: Dataclass representing the configuration for a database connection. venv\Lib\site-packages\redis\multidb\config.py: This class is used to store configuration settings for a database connection, venv\Lib\site-packages\redis\multidb\config.py: attributes and allows for the creation of customized configurations for various venv\Lib\site-packages\redis\multidb\config.py: from_pool (Optional[ConnectionPool]): A pre-configured connection pool to use. venv\Lib\site-packages\redis\multidb\config.py: class MultiDbConfig: venv\Lib\site-packages\redis\multidb\config.py: Configuration class for managing multiple database connections in a resilient and fail-safe manner. venv\Lib\site-packages\redis\multidb\config.py: databases_config: A list of database configurations. venv\Lib\site-packages\redis\multidb\config.py: Retrieves a collection of database clients managed by weighted configurations. venv\Lib\site-packages\redis\multidb\config.py: Initializes database clients based on the provided configuration and removes venv\Lib\site-packages\redis\multidb\config.py: with defined retry and backoff configurations. venv\Lib\site-packages\redis\multidb\config.py: databases_config: List[DatabaseConfig] venv\Lib\site-packages\redis\multidb\config.py: for database_config in self.databases_config: venv\Lib\site-packages\redis\multidb\config.py: database_config.client_kwargs.update(venv\Lib\site-packages\redis\multidb\config.py: if database_config.from_url: venv\Lib\site-packages\redis\multidb\config.py: database_config.from_url, **database_config.client_kwargs venv\Lib\site-packages\redis\multidb\config.py: elif database_config.from_pool: venv\Lib\site-packages\redis\multidb\config.py: database_config.from_pool.set_retry(venv\Lib\site-packages\redis\multidb\config.py: connection_pool=database_config.from_pool venv\Lib\site-packages\redis\multidb\config.py: client = self.client_class(**database_config.client_kwargs) venv\Lib\site-packages\redis\multidb\config.py: database_config.default_circuit_breaker() venv\Lib\site-packages\redis\multidb\config.py: if database_config.circuit is None venv\Lib\site-packages\redis\multidb\config.py: else database_config.circuit venv\Lib\site-packages\redis\multidb\config.py: weight=database_config.weight, venv\Lib\site-packages\redis\multidb\config.py: health_check_url=database_config.health_check_url, venv\Lib\site-packages\redis\multidb\config.py: database_config.weight, venv\Lib\site-packages\redis\multidb\healthcheck.py: "Database health check url is not set. Please check DatabaseConfig for the current database." venv\Lib\site-packages\redis_parsers\base.py: # user send an AUTH cmd to a server without authorization configured venv\Lib\site-packages\redis_parsers\base.py: "configured for the default user. Are you sure " venv\Lib\site-packages\redis_parsers\base.py: "your configuration is correct?": AuthenticationError, venv\Lib\site-packages\redis_parsers\helpers.py: "config-epoch": int, venv\Lib\site-packages\redis_parsers\helpers.py: def parse_config_get(response, **options): venv\Lib\site-packages\redis_parsers\helpers.py: "CLUSTER SAVECONFIG": bool_ok, venv\Lib\site-packages\redis_parsers\helpers.py: "CLUSTER SET-CONFIG-EPOCH": bool_ok, venv\Lib\site-packages\redis_parsers\helpers.py: "CONFIG RESETSTAT": bool_ok, venv\Lib\site-packages\redis_parsers\helpers.py: "CONFIG SET": bool_ok, venv\Lib\site-packages\redis_parsers\helpers.py: "SENTINEL FLUSHCONFIG": bool_ok, venv\Lib\site-packages\redis_parsers\helpers.py: "CONFIG GET": parse_config_get, venv\Lib\site-packages\redis_parsers\helpers.py: "CONFIG GET": lambda r: { venv\Lib\site-packages\rest_framework\apps.py: from django.apps import AppConfig venv\Lib\site-packages\rest_framework\apps.py: class RestFrameworkConfig(AppConfig): venv\Lib\site-packages\rest_framework\authtoken\apps.py: from django.apps import AppConfig venv\Lib\site-packages\rest_framework\authtoken\apps.py: class AuthTokenConfig(AppConfig): venv\Lib\site-packages\rest_framework\checks.py: def pagination_system_check(app_configs, **kwargs): venv\Lib\site-packages\rest_framework\compat.py: extension_configs = { venv\Lib\site-packages\rest_framework\compat.py: extensions=extensions, extension_configs=extension_configs venv\Lib\site-packages\rest_framework\fields.py: Quantize the decimal value to the configured precision. venv\Lib\site-packages\rest_framework\filters.py: from django.core.exceptions import FieldDoesNotExist, ImproperlyConfigured venv\Lib\site-packages\rest_framework\filters.py: raise ImproperlyConfigured(msg % self.__class__.__name__) venv\Lib\site-packages\rest_framework\generics.py: method if you want to apply the configured filtering backend to the venv\Lib\site-

```
packages\rest_framework\management\generateschema.py: help = "Generates configured API schema for project." venv\Lib\site-  
packages\rest_framework\pagination.py: page object, or None if pagination is not configured for this view. venv\Lib\site-  
packages\rest_framework\relations.py: from django.core.exceptions import ImproperlyConfigured, ObjectDoesNotExist venv\Lib\site-  
packages\rest_framework\relations.py: attributes are not configured to correctly match the URL conf. venv\Lib\site-packages\rest_framework\relations.py: #  
Return the hyperlink, or error if incorrectly configured. venv\Lib\site-packages\rest_framework\relations.py: 'model in your API, or incorrectly configured the '  
venv\Lib\site-packages\rest_framework\relations.py: raise ImproperlyConfigured(msg % self.view_name) venv\Lib\site-  
packages\rest_framework\renderers.py: from django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\renderers.py: raise  
ImproperlyConfigured( venv\Lib\site-packages\rest_framework\routers.py: from django.core.exceptions import ImproperlyConfigured venv\Lib\site-  
packages\rest_framework\routers.py: raise ImproperlyConfigured(msg) venv\Lib\site-packages\rest_framework\routers.py: raise ImproperlyConfigured(msg %  
, '.join(not_allowed)) venv\Lib\site-packages\rest_framework\serializers.py: from django.core.exceptions import FieldDoesNotExist, ImproperlyConfigured  
venv\Lib\site-packages\rest_framework\serializers.py: raise ImproperlyConfigured( venv\Lib\site-packages\rest_framework\settings.py: # View configuration  
venv\Lib\site-packages\rest_framework\test.py: from django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\test.py:  
raise ImproperlyConfigured('requests must be installed in order to use RequestsClient.') venv\Lib\site-packages\rest_framework\test.py: raise  
ImproperlyConfigured('coreapi must be installed in order to use CoreAPIClient.') venv\Lib\site-packages\rest_framework\throttling.py: from  
django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\throttling.py: raise ImproperlyConfigured(msg) venv\Lib\site-  
packages\rest_framework\throttling.py: raise ImproperlyConfigured(msg) venv\Lib\site-packages\rest_framework\views.py: # Filter out None values which  
may happen in case of config / rate venv\Lib\site-packages\sqlparse\lexer.py: """The Lexer supports configurable syntax. venv\Lib\site-  
packages\sqlparse\lexer.py: """Clear all syntax configurations. venv\Lib\site-packages\sqlparse\lexer.py: Useful if you want to load a reduced set of syntax  
configurations. venv\Lib\site-packages\tutorial\asgi.py: ASGI config for tutorial project. venv\Lib\site-packages\tutorial\quickstart\apps.py: class  
QuickstartConfig(AppConfig): venv\Lib\site-packages\tutorial\wsgi.py: WSGI config for tutorial project. (venv) PS D:\PYTHON DJANGO  
HOPES\Chat_API_DJANGO_mario> python manage.py makemigrations Migrations for 'chat': chat\migrations\0001_initial.py + Create model Room + Create  
model Message (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py migrate Operations to perform: Apply all  
migrations: admin, auth, chat, contenttypes, sessions Running migrations: Applying contenttypes.0001_initial... OK Applying auth.0001_initial... OK Applying  
admin.0001_initial... OK Applying admin.0002_logentry_remove_auto_add... OK Applying admin.0003_logentry_add_action_flag_choices... OK Applying  
contenttypes.0002_remove_content_type_name... OK Applying auth.0002_alter_permission_name_max_length... OK Applying  
auth.0003_alter_user_email_max_length... OK Applying auth.0004_alter_user_username_opts... OK Applying auth.0005_alter_user_last_login_null... OK  
Applying auth.0006_require_contenttypes_0002... OK Applying auth.0008_alter_user_username_max_length... OK Applying  
auth.0010_alter_group_name_max_length... OK Applying auth.0011_update_proxy_permissions... OK Applying  
auth.0012_alter_user_first_name_max_length... OK Applying chat.0001_initial... OK Applying sessions.0001_initial... OK (venv) PS D:\PYTHON DJANGO  
HOPES\Chat_API_DJANGO_mario> python -c "from django.core.asgi import get_asgi_application; print('django ok')" django ok (venv) PS D:\PYTHON  
DJANGO HOPES\Chat_API_DJANGO_mario> python -c "import chat.routing; print('routing ok')" Traceback (most recent call last): File "<string>", line 1, in  
<module> File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\routing.py", line 2, in <module> from .consumers import ChatConsumer  
File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\consumers.py", line 4, in <module> from .models import Room, Message File  
"D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\models.py", line 3, in <module> class Room(models.Model): File "D:\PYTHON DJANGO  
HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\db\models\base.py", line 131, in __new__ app_config =  
apps.get_containing_app_config(module) ~~~~~~ File "D:\PYTHON DJANGO  
HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\apps\registry.py", line 260, in get_containing_app_config self.check_apps_ready() File
```

"D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\apps\registry.py", line 137, in check_apps_ready File
"D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\conf__init__.py", line 81, in __getattr__ self._setup(name) File
"D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\conf__init__.py", line 61, in _setup raise ImproperlyConfigured(
django.core.exceptions.ImproperlyConfigured: Requested setting INSTALLED_APPS, but settings are not configured. You must either define the environment
variable DJANGO_SETTINGS_MODULE or call settings.configure() before accessing settings. (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> python -c "import os; os.environ.setdefault('DJANGO_SETTINGS_MODULE','monsite.settings'); import django;
django.setup(); import chat.routing; print('routing ok')" routing ok (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py
makemigrations No changes detected (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py migrate Operations to
perform: Apply all migrations: admin, auth, chat, contenttypes, sessions Running migrations: No migrations to apply. (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> py manage.py runserver Watching for file changes with StatReloader Performing system checks... System check
identified no issues (0 silenced). February 07, 2026 - 16:54:06 Django version 5.2.8, using settings 'monsite.settings' Starting development server at
http://127.0.0.1:8000/ WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For
more information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/> Not Found: / [07/Feb/2026 16:54:39] "GET / HTTP/1.1"
404 2305 (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python -m pip show channels >> ^C (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> python -m pip show channels Name: channels Version: 4.1.0 Author: Django Software Foundation License: BSD
Location: D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages Requires: asgiref, Django Required-by: channels-redis (venv)
PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py makemigrations (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> python manage.py migrate Operations to perform: Apply all migrations: admin, auth, chat, contenttypes, sessions
Running migrations: (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py makemigrations (venv) PS D:\PYTHON
DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py migrate Operations to perform: Apply all migrations: admin, auth, chat, contenttypes,
sessions Running migrations: (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py makemigrations No changes
detected (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py migrate Operations to perform: Apply all migrations:
admin, auth, chat, contenttypes, sessions Running migrations: No migrations to apply. (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario>
py manage.py runserver Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07,
2026 - 17:03:21 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-
BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more
information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/> Not Found: / [07/Feb/2026 17:03:24] "GET / HTTP/1.1" 404
2305 Not Found: / [07/Feb/2026 17:03:25] "GET / HTTP/1.1" 404 2305 Not Found: / [07/Feb/2026 17:03:27] "GET / HTTP/1.1" 404 2305 [07/Feb/2026
17:04:21] "GET /admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:04:22] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 4252 [07/Feb/2026 17:04:22] "GET
/static/admin/css/nav_sidebar.css HTTP/1.1" 200 2810 [07/Feb/2026 17:04:22] "GET /static/admin/css/base.css HTTP/1.1" 200 22120 [07/Feb/2026 17:04:22]
"GET /static/admin/css/login.css HTTP/1.1" 200 951 [07/Feb/2026 17:04:22] "GET /static/admin/css/dark_mode.css HTTP/1.1" 200 2808 [07/Feb/2026
17:04:22] "GET /static/admin/css/responsive.css HTTP/1.1" 200 16565 [07/Feb/2026 17:04:22] "GET /static/admin/js/theme.js HTTP/1.1" 200 1653
[07/Feb/2026 17:04:22] "GET /static/admin/js/nav_sidebar.js HTTP/1.1" 200 3063 Not Found: /favicon.ico [07/Feb/2026 17:04:22] "GET /favicon.ico
HTTP/1.1" 404 2356 Not Found: /api/ [07/Feb/2026 17:04:37] "GET /api/ HTTP/1.1" 404 2739 Not Found: / [07/Feb/2026 17:04:48] "GET / HTTP/1.1" 404
2305 [07/Feb/2026 17:05:01] "GET /admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:05:01] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 4252 Not Found:
/api/ [07/Feb/2026 17:05:12] "GET /api/ HTTP/1.1" 404 2739 Not Found: / [07/Feb/2026 17:24:49] "GET /api/rooms/ HTTP/1.1" 200 2 Not Found:
/favicon.ico [07/Feb/2026 17:24:49] "GET /favicon.ico HTTP/1.1" 404 2356 (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> curl -X

```
POST http://127.0.0.1:8000/api/rooms/ -H "Content-Type: application/json" -d "{\"name\":\"general\"}" >> ^C (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> curl -X POST http://127.0.0.1:8000/api/rooms/ -H "Content-Type: application/json" -d "{\"name\":\"general\"}" Invoke-
WebRequest : Impossible de lier le paramètre «Headers». Impossible de convertir la valeur «Content-Type: application/json» du type «System.String» en type
«System.Collections.IDictionary». Au caractère Ligne:1 : 50 + ~~~~~ + CategoryInfo : InvalidArgument : (:) [Invoke-
WebRequest], ParameterBindingException + FullyQualifiedErrorId :
CannotConvertArgumentNoMessage,Microsoft.PowerShell.Commands.InvokeWebRequestCommand (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/" >> -Method Post >> -ContentType "application/json"
Invoke-RestMethod : Impossible de se connecter au serveur distant Au caractère Ligne:1 : 1 + Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/" +
~~~~~ + CategoryInfo : NotSpecified: (:) [Invoke-RestMethod], WebException +
FullyQualifiedErrorId : System.Net.WebException,Microsoft.PowerShell.Commands.InvokeRestMethodCommand (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> py manage.py runserver Watching for file
changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 17:30:20 Django version 5.2.8,
using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development
server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see:
https://docs.djangoproject.com/en/5.2/howto/deployment/ Not Found: / [07/Feb/2026 17:30:24] "GET / HTTP/1.1" 404 2305 Not Found: / [07/Feb/2026
17:30:25] "GET / HTTP/1.1" 404 2305 Not Found: /ws/chat/general/ [07/Feb/2026 17:31:01] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found:
/ws/chat/general/ [07/Feb/2026 17:31:04] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:31:24] "GET /api/rooms/ HTTP/1.1" 200 2 Not
Found: /ws/chat/general/ [07/Feb/2026 17:31:32] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:31:58] "GET
/ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:32:00] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found:
/ws/chat/general/ [07/Feb/2026 17:32:02] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:32:04] "GET
/ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:21] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found:
/ws/chat/general/ [07/Feb/2026 17:35:22] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:22] "GET
/ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:32] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found:
/ws/chat/general/ [07/Feb/2026 17:35:33] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:35] "GET
/ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /api/ \u0022 WS: ws://127.0.0.1:8000/ws/chat/<room>/ [07/Feb/2026 17:35:44] "GET
/api/%20%E2%80%A2%20WS:%20ws://127.0.0.1:8000/ws/chat/%3Croom%3E/ HTTP/1.1" 404 2899 Not Found: /api/ [07/Feb/2026 17:35:51] "GET /api/
HTTP/1.1" 404 2739 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:43] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/
[07/Feb/2026 17:36:44] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:44] "GET /ws/chat/general/ HTTP/1.1"
404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:45] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026
17:36:46] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:47] "GET /ws/chat/general/ HTTP/1.1" 404 2371
[07/Feb/2026 17:38:30] "GET /api/rooms/ HTTP/1.1" 200 2 Not Found: /favicon.ico [07/Feb/2026 17:38:30] "GET /favicon.ico HTTP/1.1" 404 2356 Not
Found: / [07/Feb/2026 17:38:52] "GET / HTTP/1.1" 404 2305 [07/Feb/2026 17:41:00] "GET /admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:41:00] "GET
/admin/login/?next=/admin/ HTTP/1.1" 200 4252 [07/Feb/2026 17:41:01] "GET /static/admin/css/dark_mode.css HTTP/1.1" 200 2808 [07/Feb/2026 17:41:01]
"GET /static/admin/css/base.css HTTP/1.1" 200 22120 [07/Feb/2026 17:41:01] "GET /static/admin/css/nav_sidebar.css HTTP/1.1" 200 2810 [07/Feb/2026
17:41:01] "GET /static/admin/js/theme.js HTTP/1.1" 200 1653 [07/Feb/2026 17:41:01] "GET /static/admin/js/nav_sidebar.js HTTP/1.1" 200 3063
[07/Feb/2026 17:41:01] "GET /static/admin/css/login.css HTTP/1.1" 200 951 [07/Feb/2026 17:41:01] "GET /static/admin/css/responsive.css HTTP/1.1" 200
16565 [07/Feb/2026 17:41:35] "POST /admin/login/?next=/admin/ HTTP/1.1" 200 4498 [07/Feb/2026 17:44:04] "POST /admin/login/?next=/admin/
```


HTTP/1.1" 302 0 [07/Feb/2026 17:44:04] "GET /admin/ HTTP/1.1" 200 7885 [07/Feb/2026 17:44:05] "GET /static/admin/css/dashboard.css HTTP/1.1" 200 441 [07/Feb/2026 17:44:05] "GET /static/admin/img/icon-changelink.svg HTTP/1.1" 200 380 [07/Feb/2026 17:44:05] "GET /static/admin/img/icon-addlink.svg HTTP/1.1" 200 331 [07/Feb/2026 17:44:27] "GET /admin/chat/room/add/ HTTP/1.1" 200 9637 [07/Feb/2026 17:44:27] "GET /static/admin/js/jquery.init.js HTTP/1.1" 200 347 [07/Feb/2026 17:44:27] "GET /static/admin/css/forms.css HTTP/1.1" 200 8525 [07/Feb/2026 17:44:27] "GET /static/admin/js/core.js HTTP/1.1" 200 6208 [07/Feb/2026 17:44:27] "GET /static/admin/js/urlify.js HTTP/1.1" 200 7887 [07/Feb/2026 17:44:27] "GET /static/admin/js/prepopulate.js HTTP/1.1" 200 1531 [07/Feb/2026 17:44:27] "GET /admin/jsi18n/ HTTP/1.1" 200 8634 [07/Feb/2026 17:44:27] "GET /static/admin/js/change_form.js HTTP/1.1" 200 606 [07/Feb/2026 17:44:27] "GET /static/admin/js/admin/RelatedObjectLookups.js HTTP/1.1" 200 9777 [07/Feb/2026 17:44:27] "GET /static/admin/js/actions.js HTTP/1.1" 200 8076 [07/Feb/2026 17:44:27] "GET /static/admin/js/prepopulate_init.js HTTP/1.1" 200 586 [07/Feb/2026 17:44:27] "GET /static/admin/css/widgets.css HTTP/1.1" 200 11973 [07/Feb/2026 17:44:28] "GET /static/admin/js/vendor/xregexp/xregexp.js HTTP/1.1" 200 325171 [07/Feb/2026 17:44:28] "GET /static/admin/js/vendor/jquery/jquery.js HTTP/1.1" 200 285314 [07/Feb/2026 17:44:44] "POST /admin/chat/room/add/ HTTP/1.1" 302 0 [07/Feb/2026 17:44:44] "GET /admin/chat/room/ HTTP/1.1" 200 10874 [07/Feb/2026 17:44:45] "GET /admin/jsi18n/ HTTP/1.1" 200 8634 [07/Feb/2026 17:44:45] "GET /static/admin/css/changelists.css HTTP/1.1" 200 6878 [07/Feb/2026 17:44:45] "GET /static/admin/js/filters.js HTTP/1.1" 200 978 [07/Feb/2026 17:44:45] "GET /static/admin/img/search.svg HTTP/1.1" 200 458 [07/Feb/2026 17:44:45] "GET /static/admin/img/tooltag-add.svg HTTP/1.1" 200 331 [07/Feb/2026 17:44:45] "GET /static/admin/img/icon-yes.svg HTTP/1.1" 200 436 [07/Feb/2026 17:44:58] "GET / HTTP/1.1" 404 2305 Not Found: / [07/Feb/2026 17:44:59] "GET / HTTP/1.1" 404 2305 Not Found: /ws/chat/general/ [07/Feb/2026 17:45:18] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:45:22] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:06] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:08] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:09] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:38] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:39] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:41] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:52] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:47:53] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 17:47:53] "GET /api/rooms/1/messages/ HTTP/1.1" 200 2 [07/Feb/2026 17:47:54] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 17:47:54] "GET /api/rooms/1/messages/ HTTP/1.1" 200 2 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:57] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:50:18] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /favicon.ico [07/Feb/2026 17:50:18] "GET /favicon.ico HTTP/1.1" 404 2356 [07/Feb/2026 17:50:21] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /favicon.ico [07/Feb/2026 17:50:21] "GET /favicon.ico HTTP/1.1" 404 2356 Not Found: /ws/chat/general/ [07/Feb/2026 17:53:33] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:53:34] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:53:34] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:55:20] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /favicon.ico [07/Feb/2026 17:55:21] "GET /favicon.ico HTTP/1.1" 404 2356 [07/Feb/2026 17:56:52] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 17:57:19] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2467 Not Found: /ws/chat/general/ [07/Feb/2026 17:58:15] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 17:59:50] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2467 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:09] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:11] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:12] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:18] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:25] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:25] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:26] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 18:00:27] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 18:00:27] "GET /api/rooms/1/messages/

HTTP/1.1" 200 2 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:28] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:29] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:07:52] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:07:53] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:08:00] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:08:00] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:08:01] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:08:28] "GET /ws/chat/general/ HTTP/1.1" 404 2371 D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\views.py changed, reloading. Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 18:24:04 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/> D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\views.py changed, reloading. Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 18:24:10 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/> D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\monsite\urls.py changed, reloading. Watching for file changes with StatReloader Performing system checks... Exception in thread django-main-thread: Traceback (most recent call last): File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\urls\resolvers.py", line 720, in url_patterns iter(patterns) TypeError: 'module' object is not iterable The above exception was the direct cause of the following exception: Traceback (most recent call last): File "C:\Program Files\Python3_12_4\Lib\threading.py", line 1073, in _bootstrap_inner self.run() File "C:\Program Files\Python3_12_4\Lib\threading.py", line 1010, in run self._target(*self._args, **self._kwargs) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\utils\autoreload.py", line 64, in wrapper fn(*args, **kwargs) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\management\commands\runserver.py", line 134, in inner_run self.check(**check_kwargs) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\management\base.py", line 492, in check_all_issues = checks.run_checks() File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\checks\registry.py", line 89, in run_checks new_errors = check(app_configs=app_configs, databases=databases) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\checks\urls.py", line 16, in check_url_config return check_resolver(resolver) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\checks\urls.py", line 26, in check_resolver return check_method() File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\urls\resolvers.py", line 531, in check_for_pattern in self.url_patterns: File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\utils\functional.py", line 47, in __get__ res = instance.__dict__[self.name] = self.func(instance) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\urls\resolvers.py", line 728, in url_patterns raise ImproperlyConfigured(msg.format(name=self.urlconf_name)) from e django.core.exceptions.ImproperlyConfigured: The included URLconf 'monsite.urls' does not appear to have any patterns in it. If you see the 'urlpatterns' variable with valid patterns in the file then the issue is probably caused by a circular import. D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\monsite\urls.py changed, reloading. Watching for file changes with StatReloader Performing system checks... Exception in thread django-main-thread: Traceback (most recent call last): File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\urls\resolvers.py", line 720, in url_patterns iter(patterns) TypeError: 'module' object is not iterable The above exception was the direct cause of the following exception: Traceback (most recent call last): File "C:\Program Files\Python3_12_4\Lib\threading.py", line 1073, in _bootstrap_inner self.run()

[illegible]

Downloading cbor2-5.8.0-cp312-cp312-win_amd64.whl.metadata (5.6 kB) Collecting cffi>=2.0.0 (from autobahn>=22.4.2->daphne) Using cached cffi-2.0.0-cp312-cp312-win_amd64.whl.metadata (2.6 kB) Collecting cryptography>=3.4.6 (from autobahn>=22.4.2->daphne) Downloading cryptography-46.0.4-cp311-abi3-win_amd64.whl.metadata (5.7 kB) Collecting hyperlink>=21.0.0 (from autobahn>=22.4.2->daphne) Downloading hyperlink-21.0.0-py2.py3-none-any.whl.metadata (1.5 kB) Requirement already satisfied: msgpack>=1.0.2 in d:\python django hopes\chat_api_django_mario\venv\lib\site-packages (from autobahn>=22.4.2->daphne) (1.1.2) Collecting py-ubjson>=0.16.1 (from autobahn>=22.4.2->daphne) Downloading py-ubjson-0.16.1.tar.gz (50 kB)

50.3/50.3 kB 284.2 kB/s eta 0:00:00 Installing build dependencies ... done Getting requirements to build wheel ... done Preparing metadata (pyproject.toml) ... done Collecting txaio>=25.12.2 (from autobahn>=22.4.2->daphne) Downloading txaio-25.12.2-py3-none-any.whl.metadata (7.0 kB) Collecting ujson>=4.0.2 (from autobahn>=22.4.2->daphne) Downloading ujson-5.11.0-cp312-cp312-win_amd64.whl.metadata (9.6 kB) Collecting attrs>=22.2.0 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading attrs-25.4.0-py3-none-any.whl.metadata (10 kB) Collecting automat>=24.8.0 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading automat-25.4.16-py3-none-any.whl.metadata (8.4 kB) Collecting constantly>=15.1 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading constantly-23.10.4-py3-none-any.whl.metadata (1.8 kB) Collecting incremental>=24.7.0 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading incremental-24.11.0-py3-none-any.whl.metadata (9.5 kB) Collecting typing-extensions>=4.2.0 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Using cached typing_extensions-4.15.0-py3-none-any.whl.metadata (3.3 kB) Collecting zope-interface>=5 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading zope_interface-8.2-cp312-cp312-win_amd64.whl.metadata (46 kB)

46.7/46.7 kB 331.3 kB/s eta 0:00:00 Collecting idna>=2.4 (from twisted[tls]>=22.4->daphne) Downloading idna-3.11-py3-none-any.whl.metadata (8.4 kB) Collecting pyopenssl>=21.0.0 (from twisted[tls]>=22.4->daphne) Downloading pyopenssl-25.3.0-py3-none-any.whl.metadata (17 kB) Collecting service-identity>=18.1.0 (from twisted[tls]>=22.4->daphne) Downloading service_identity-24.2.0-py3-none-any.whl.metadata (5.1 kB) Collecting pycparser (from cffi>=2.0.0->autobahn>=22.4.2->daphne) Downloading pycparser-3.0-py3-none-any.whl.metadata (8.2 kB) Collecting packaging>=17.0 (from incremental>=24.7.0->twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading packaging-26.0-py3-none-any.whl.metadata (3.3 kB) Collecting pyasn1 (from service-identity>=18.1.0->twisted[tls]>=22.4->daphne) Downloading pyasn1-0.6.2-py3-none-any.whl.metadata (8.4 kB) Collecting pyasn1-modules (from service-identity>=18.1.0->twisted[tls]>=22.4->daphne) Using cached pyasn1_modules-0.4.2-py3-none-any.whl.metadata (3.5 kB) Downloading daphne-4.2.1-py3-none-any.whl (29 kB) Downloading autobahn-25.12.2-cp312-cp312-win_amd64.whl (2.2 MB)

2.2/2.2 MB 1.2 MB/s eta 0:00:00 Downloading twisted-25.5.0-py3-none-any.whl (3.2 MB)

3.2/3.2 MB 1.5 MB/s eta 0:00:00 Downloading attrs-25.4.0-py3-none-any.whl (67 kB)

67.6/67.6 kB 459.0 kB/s eta 0:00:00 Downloading automat-25.4.16-py3-none-any.whl (42 kB)

42.8/42.8 kB 299.1 kB/s eta 0:00:00 Downloading cbor2-5.8.0-cp312-cp312-win_amd64.whl (69 kB)

69.8/69.8 kB 546.6 kB/s eta 0:00:00 Using cached cffi-2.0.0-cp312-cp312-win_amd64.whl (183 kB) Downloading constantly-23.10.4-py3-none-any.whl (13 kB) Downloading cryptography-46.0.4-cp311-abi3-win_amd64.whl (3.5 MB)

3.5/3.5 MB 844.1 kB/s eta 0:00:00 Downloading hyperlink-21.0.0-py2.py3-none-any.whl (74 kB)

74.6/74.6 kB 825.6 kB/s eta 0:00:00 Downloading idna-3.11-py3-none-any.whl (71 kB)

71.0/71.0 kB 129.6 kB/s eta 0:00:00 Downloading incremental-24.11.0-py3-none-any.whl (21 kB) Downloading pyopenssl-25.3.0-py3-none-any.whl (57 kB)

57.3/57.3 kB 505.7 kB/s eta 0:00:00 Downloading service_identity-24.2.0-py3-none-any.whl (11

```

kB) Downloading txaio-25.12.2-py3-none-any.whl (31 kB) Using cached typing_extensions-4.15.0-py3-none-any.whl (44 kB) Downloading ujson-5.11.0-
cp312-cp312-win_amd64.whl (43 kB) ----- 43.9/43.9 kB 357.6
kB/s eta 0:00:00 Downloading zope_interface-8.2-cp312-cp312-win_amd64.whl (212 kB) -----
----- 212.4/212.4 kB 1.2 MB/s eta 0:00:00 Downloading packaging-26.0-py3-none-any.whl (74 kB) -----
----- 74.4/74.4 kB 372.8 kB/s eta 0:00:00 Downloading pyasn1-0.6.2-py3-none-any.whl (83
kB) ----- 83.4/83.4 kB 778.0 kB/s eta 0:00:00 Using cached
pyasn1_modules-0.4.2-py3-none-any.whl (181 kB) Downloading pycparser-3.0-py3-none-any.whl (48 kB) -----
----- 48.2/48.2 kB 403.5 kB/s eta 0:00:00 Building wheels for collected packages: py-ubjson Building wheel for py-
ubjson (pyproject.toml) ... done Created wheel for py-ubjson: filename=py_ubjson-0.16.1-cp312-cp312-win_amd64.whl size=19356
sha256=7b3fc1d543eefd1d8e1994a85b47c17a024a1a95741dc0ced87344b32adfb966 Stored in directory:
c:\users\spsse02men\appdata\local\pip\cache\wheels\c0\8\73\e1931c0c1125bc84ae955eeb3d8ba9fe23ecd95ee337095267 Successfully built py-ubjson
Installing collected packages: py-ubjson, zope-interface, ujson, typing-extensions, txaio, pycparser, pyasn1, packaging, idna, constantly, cbor2, automat, attrs,
pyasn1-modules, incremental, hyperlink, cffi, twisted, cryptography, service-identity, pyopenssl, Successfully installed attrs-25.4.0 autobahn-25.12.2 automat-
25.4.16 cbor2-5.8.0 cffi-2.0.0 constantly-23.10.4 cryptography-46.0.4 daphne-4.2.1 hyperlink-21.0.0 idna-3.11 incremental-24.11.0 packaging-26.0 py-ubjson-
0.16.1 pyasn1-0.6.2 pyasn1-modules-0.4.2 pycparser-3.0 pyopenssl-25.3.0 service-identity-24.2.0 twisted-25.5.0 txaio-25.12.2 typing-extensions-4.15.0 ujson-
5.11.0 zope-interface-8.2 [notice] A new release of pip is available: 24.0 -> 26.0.1 [notice] To update, run: python.exe -m pip install --upgrade pip (venv) PS
D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> py manage.py
runserver Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 -
18:34:56 Django version 5.2.8, using settings 'monsite.settings' Starting ASGI/Daphne version 4.2.1 development server at http://127.0.0.1:8000/ Quit the
server with CTRL-BREAK. Not Found: /ws/chat/general/ HTTP GET /ws/chat/general/ 404 [0.06, 127.0.0.1:22100] Not Found: /ws/chat/general/ HTTP GET
/ws/chat/general/ 404 [0.02, 127.0.0.1:22100] HTTP GET / 200 [0.01, 127.0.0.1:22100] WebSocket HANDSHAKING /ws/chat/general/ [127.0.0.1:22107]
WebSocket CONNECT /ws/chat/general/ [127.0.0.1:22107] WebSocket DISCONNECT /ws/chat/general/ [127.0.0.1:22107] HTTP GET / 200 [0.01,
127.0.0.1:22100] WebSocket HANDSHAKING /ws/chat/general/ [127.0.0.1:22110] WebSocket CONNECT /ws/chat/general/ [127.0.0.1:22110]

```

Je vois exactement ce qui se passe 📌

1) Pourquoi tu as des 404 /ws/chat/general/ et des logs “WebSocket CONNECT” ?

Parce que tu testes 2 choses différentes :

- **HTTP GET** sur /ws/chat/general/ (ex: tu colles l’URL dans le navigateur) → **ça fait un 404**, normal : c’est **pas** une route HTTP.
- **WebSocket** sur ws://127.0.0.1:8000/ws/chat/general/ (via JS `new WebSocket(...)`) → là on voit :
 - WebSocket HANDSHAKING
 - WebSocket CONNECT

✅ donc le WS marche.

👉 Conclusion : **ignore les 404 HTTP** sur `/ws/...` si tu n'es pas en train d'ouvrir un vrai WebSocket.

2) Ton souci principal maintenant : “Envoyer → WS fermé / erreur”

Ça arrive souvent quand :

- tu cliques “Envoyer” alors que `socket.readyState !== OPEN`
- ou ton `receive()` plante côté serveur (exception → Django ferme le socket)
- ou tu n'as pas de `channel layer` (Redis) et `group_send` échoue

Comme tu as `channels-redis` installé (vu dans `pip show channels` “Required-by: channels-redis”), mais on ne voit pas Redis tourner, il y a un risque.

3) Test ultra simple pour valider sans Redis (pas de group)

Dans ton `ChatConsumer`, fais un echo pour commencer :

```
async def receive(self, text_data):  
    await self.send(text_data=text_data)
```

👉 Si ça renvoie le message dans le même onglet → **ton WS + JS sont OK.**

Ensuite seulement tu remets `group_send`.

4) Si tu veux le broadcast (multi-onglets), il faut un Channel Layer

Option A (dev rapide, sans Redis) : InMemory (1 process)

Dans `settings.py` :

```
CHANNEL_LAYERS = {  
    "default": {"BACKEND": "channels.layers.InMemoryChannelLayer"}
```

```
}
```

✅ marche pour tests **sur une seule instance** (pas prod)

Option B (recommandé) : Redis

Toujours dans `settings.py` :

```
CHANNEL_LAYERS = {
    "default": {
        "BACKEND": "channels_redis.core.RedisChannelLayer",
        "CONFIG": {"hosts": [("127.0.0.1", 6379)]},
    }
}
```

Et il faut **Redis lancé** (service/docker).

5) Petit détail important côté client

Ton log montre :

- CONNECT
- puis parfois DISCONNECT vite

Donc avant d'envoyer, fais :

```
if (socket.readyState === WebSocket.OPEN) {
    socket.send(JSON.stringify({ message: msg }));
} else {
    console.log("WS pas ouvert:", socket.readyState);
}
```

Ce que je te conseille là tout de suite

1. Mets le `receive` en **echo** (sans groupe) pour voir si l'envoi marche.

2. Si OK → ajoute `InMemoryChannelLayer` pour tester le broadcast.
3. Si tu veux Redis → on le branche après.

Si tu colles ici ton `chat/consumers.py` (connect/receive/disconnect) + le JS `socket.send(...)`, je te dis exactement où ça casse.

```
venv\Lib\site-packages\redis\asyncio\multidb\config.py: health_check_url=database_config.health_check_url, venv\Lib\site-
packages\redis\asyncio\multidb\config.py: database_config.weight, venv\Lib\site-packages\redis\asyncio\multidb\healthcheck.py: "Database health check url is
not set. Please check DatabaseConfig for the current database." venv\Lib\site-packages\redis\auth\idp.py:class IdentityProviderConfigInterface(ABC):
venv\Lib\site-packages\redis\auth\idp.py: Configuration class that provides a configured identity provider. venv\Lib\site-
packages\redis\auth\token_manager.py:class TokenManagerConfig: venv\Lib\site-packages\redis\auth\token_manager.py: self, identity_provider:
IdentityProviderInterface, config: TokenManagerConfig venv\Lib\site-packages\redis\auth\token_manager.py: self._config = config venv\Lib\site-
packages\redis\auth\token_manager.py: if self._retries < self._config.get_retry_policy().get_max_attempts(): venv\Lib\site-
packages\redis\auth\token_manager.py: sleep(self._config.get_retry_policy().get_delay_in_ms() / 1000) venv\Lib\site-packages\redis\auth\token_manager.py:
if self._retries < self._config.get_retry_policy().get_max_attempts(): venv\Lib\site-packages\redis\auth\token_manager.py:
self._config.get_retry_policy().get_delay_in_ms() / 1000 venv\Lib\site-packages\redis\auth\token_manager.py: - self._config.get_lower_refresh_bound_millis()
venv\Lib\site-packages\redis\auth\token_manager.py: token_ttl * self._config.get_expiration_refresh_ratio() venv\Lib\site-packages\redis\cache.py:class
CacheConfigurationInterface(ABC): venv\Lib\site-packages\redis\cache.py: def config(self) -> CacheConfigurationInterface: venv\Lib\site-
packages\redis\cache.py: cache_config: CacheConfigurationInterface, venv\Lib\site-packages\redis\cache.py: self. cache_config = cache_config venv\Lib\site-
packages\redis\cache.py: self._eviction_policy = self._cache_config.get_eviction_policy().value() venv\Lib\site-packages\redis\cache.py: def config(self) ->
CacheConfigurationInterface: venv\Lib\site-packages\redis\cache.py: return self._cache_config venv\Lib\site-packages\redis\cache.py: if
self._cache_config.is_exceeds_max_size(len(self._cache)): venv\Lib\site-packages\redis\cache.py: return
self._cache_config.is_allowed_to_cache(key.command) venv\Lib\site-packages\redis\cache.py:class CacheConfig(CacheConfigurationInterface):
venv\Lib\site-packages\redis\cache.py: def __init__(self, cache_config: Optional[CacheConfig] = None): venv\Lib\site-packages\redis\cache.py: self._config =
cache_config venv\Lib\site-packages\redis\cache.py: if self._config is None: venv\Lib\site-packages\redis\cache.py: self._config = CacheConfig()
venv\Lib\site-packages\redis\cache.py: cache_class = self._config.get_cache_class() venv\Lib\site-packages\redis\cache.py: return
cache_class(cache_config=self._config) venv\Lib\site-packages\redis\client.py:from redis.cache import CacheConfig, CacheInterface venv\Lib\site-
packages\redis\client.py: MaintNotificationsConfig, venv\Lib\site-packages\redis\client.py: configuration, an instance will either use a ConnectionPool, or
venv\Lib\site-packages\redis\client.py: Return a Redis client object configured from the given URL venv\Lib\site-packages\redis\client.py: cache_config:
Optional[CacheConfig] = None, venv\Lib\site-packages\redis\client.py: maint_notifications_config: Optional[MaintNotificationsConfig] = None,
venv\Lib\site-packages\redis\client.py: 2. Define a Retry object with configured 'supported_errors' and set venv\Lib\site-packages\redis\client.py: When
'connection_pool' is provided - the retry configuration of the venv\Lib\site-packages\redis\client.py: maint_notifications_config: venv\Lib\site-
packages\redis\client.py: configuration the pool to support maintenance notifications - see venv\Lib\site-packages\redis\client.py:
redis.maint_notifications.MaintNotificationsConfig for details. venv\Lib\site-packages\redis\client.py: if (cache_config or cache) and protocol in [3,
"3"]: venv\Lib\site-packages\redis\client.py: "cache_config": cache_config, venv\Lib\site-packages\redis\client.py: maint_notifications_config and
maint_notifications_config.enabled venv\Lib\site-packages\redis\client.py: if maint_notifications_config: venv\Lib\site-packages\redis\client.py:
"maint_notifications_config": maint_notifications_config, venv\Lib\site-packages\redis\client.py: if (cache_config or cache) and
self.connection_pool.get_protocol() not in [ venv\Lib\site-packages\redis\cluster.py:from redis.cache import CacheConfig, CacheFactory,
```


CacheFactoryInterface, CacheInterface venv\Lib\site-packages\redis\cluster.py:from redis.maint_notifications import MaintNotificationsConfig venv\Lib\site-packages\redis\cluster.py: "cache_config", venv\Lib\site-packages\redis\cluster.py: "CONFIG SET", venv\Lib\site-packages\redis\cluster.py: "CONFIG REWRITE", venv\Lib\site-packages\redis\cluster.py: "CONFIG RESETSTAT", venv\Lib\site-packages\redis\cluster.py: "CLUSTER SET-CONFIG-EPOCH", venv\Lib\site-packages\redis\cluster.py: "CONFIG GET", venv\Lib\site-packages\redis\cluster.py: "FT.CONFIG", venv\Lib\site-packages\redis\cluster.py: "CONFIG SET", venv\Lib\site-packages\redis\cluster.py: "CONFIG REWRITE", venv\Lib\site-packages\redis\cluster.py: "CONFIG RESETSTAT", venv\Lib\site-packages\redis\cluster.py: Return a Redis client object configured from the given URL venv\Lib\site-packages\redis\cluster.py: reason="Please configure the 'load_balancing_strategy' instead", venv\Lib\site-packages\redis\cluster.py: reason="Please configure the 'retry' object instead", venv\Lib\site-packages\redis\cluster.py: cache_config: Optional[CacheConfig] = None, venv\Lib\site-packages\redis\cluster.py: <https://redis.io/topics/cluster-tutorial#redis-cluster-configuration-parameters> venv\Lib\site-packages\redis\cluster.py: @deprecated - Please configure the 'retry' object instead venv\Lib\site-packages\redis\cluster.py: # and there we provide retry configuration without retries allowed. venv\Lib\site-packages\redis\cluster.py: if (cache_config or cache) and protocol not in [3, "3"]: venv\Lib\site-packages\redis\cluster.py: cache_config=cache_config, venv\Lib\site-packages\redis\cluster.py: # Sending READONLY command to server to configure connection as venv\Lib\site-packages\redis\cluster.py: # rc.cluster_save_config(rc.get_primaries()) venv\Lib\site-packages\redis\cluster.py: config option "self.retry" which defaults to 3 unless manually venv\Lib\site-packages\redis\cluster.py: configured. venv\Lib\site-packages\redis\cluster.py: cache_config: Optional[CacheConfig] = None, venv\Lib\site-packages\redis\cluster.py: self._cache_config = cache_config venv\Lib\site-packages\redis\cluster.py: # We are configuring the connection pool not to retry venv\Lib\site-packages\redis\cluster.py: node_retry_config = Retry(venv\Lib\site-packages\redis\cluster.py: {"maint_notifications_config": MaintNotificationsConfig(enabled=False)} venv\Lib\site-packages\redis\cluster.py: kwargs.update({"retry": node_retry_config}) venv\Lib\site-packages\redis\cluster.py: retry=node_retry_config, venv\Lib\site-packages\redis\cluster.py: if self._cache is None and self._cache_config is not None: venv\Lib\site-packages\redis\cluster.py: self._cache = CacheFactory(self._cache_config).get_cache() venv\Lib\site-packages\redis\cluster.py: reason="Please configure the 'retry' object instead", venv\Lib\site-packages\redis\cluster.py: "CONFIG GET", venv\Lib\site-packages\redis\cluster.py: "CONFIG RESETSTAT", venv\Lib\site-packages\redis\cluster.py: "CONFIG REWRITE", venv\Lib\site-packages\redis\cluster.py: "CONFIG SET", venv\Lib\site-packages\redis\cluster.py: "CONFIG", venv\Lib\site-packages\redis\cluster.py: the retries in config option "self.retry" venv\Lib\site-packages\redis\cluster.py: which defaults to 3 unless manually configured. venv\Lib\site-packages\redis\cluster.py: # rc.cluster_save_config(rc.get_primaries()) venv\Lib\site-packages\redis\commands\cluster.py: Get Cluster config for the node. venv\Lib\site-packages\redis\commands\cluster.py: Reconfigure a node as a slave of the specified master node venv\Lib\site-packages\redis\commands\cluster.py: def cluster_save_config(venv\Lib\site-packages\redis\commands\cluster.py: For more information see <https://redis.io/commands/cluster-saveconfig> venv\Lib\site-packages\redis\commands\cluster.py: return self.execute_command("CLUSTER SAVECONFIG", target_nodes=target_nodes) venv\Lib\site-packages\redis\commands\cluster.py: def cluster_set_config_epoch(venv\Lib\site-packages\redis\commands\cluster.py: Set the configuration epoch in a new node venv\Lib\site-packages\redis\commands\cluster.py: For more information see <https://redis.io/commands/cluster-set-config-epoch> venv\Lib\site-packages\redis\commands\cluster.py: "CLUSTER SET-CONFIG-EPOCH", epoch, target_nodes=target_nodes venv\Lib\site-packages\redis\commands\core.py: Load ACL rules from the configured 'aclfile'. venv\Lib\site-packages\redis\commands\core.py: Note that the server must be configured with the 'aclfile' venv\Lib\site-packages\redis\commands\core.py: Save ACL rules to the configured 'aclfile'. venv\Lib\site-packages\redis\commands\core.py: Note that the server must be configured with the 'aclfile' venv\Lib\site-packages\redis\commands\core.py: def config_get(venv\Lib\site-packages\redis\commands\core.py: Return a dictionary of configuration based on the 'pattern' venv\Lib\site-packages\redis\commands\core.py: For more information, see <https://redis.io/commands/config-get> venv\Lib\site-packages\redis\commands\core.py: return self.execute_command("CONFIG GET", pattern, *args, **kwargs) venv\Lib\site-packages\redis\commands\core.py: def config_set(venv\Lib\site-packages\redis\commands\core.py: """Set config item 'name' with value venv\Lib\site-packages\redis\commands\core.py: For more information, see <https://redis.io/commands/config-set> venv\Lib\site-packages\redis\commands\core.py: return self.execute_command("CONFIG SET",

name, value, *args, **kwargs) venv\Lib\site-packages\redis\commands\core.py: def config_resetstat(self, **kwargs) -> ResponseT: venv\Lib\site-packages\redis\commands\core.py: For more information, see <https://redis.io/commands/config-resetstat> venv\Lib\site-packages\redis\commands\core.py: return self.execute_command("CONFIG RESETSTAT", **kwargs) venv\Lib\site-packages\redis\commands\core.py: def config_rewrite(self, **kwargs) -> ResponseT: venv\Lib\site-packages\redis\commands\core.py: Rewrite config file with the minimal change to reflect running config. venv\Lib\site-packages\redis\commands\core.py: For more information, see <https://redis.io/commands/config-rewrite> venv\Lib\site-packages\redis\commands\core.py: return self.execute_command("CONFIG REWRITE", **kwargs) venv\Lib\site-packages\redis\commands\core.py: """Shutdown the Redis server. If Redis has persistence configured, venv\Lib\site-packages\redis\commands\core.py: `save` will force a DB saving operation even if no save points are configured. venv\Lib\site-packages\redis\commands\core.py: are configured. venv\Lib\site-packages\redis\commands\core.py: """Shutdown the Redis server. If Redis has persistence configured, venv\Lib\site-packages\redis\commands\core.py: configured. If the "nosave" option is set, no data flush will be venv\Lib\site-packages\redis\commands\core.py: Loads a module from a dynamic library at runtime with configuration directives. venv\Lib\site-packages\redis\commands\core.py: pieces.append("CONFIG") venv\Lib\site-packages\redis\commands\search\commands.py: HybridPostProcessingConfig, venv\Lib\site-packages\redis\commands\search\commands.py: CONFIG_CMD = "FT.CONFIG" venv\Lib\site-packages\redis\commands\search\commands.py: def _parse_config_get(self, res, **kwargs): venv\Lib\site-packages\redis\commands\search\commands.py: post_processing: Optional[HybridPostProcessingConfig] = None, venv\Lib\site-packages\redis\commands\search\commands.py: - **post_processing**: HybridPostProcessingConfig object venv\Lib\site-packages\redis\commands\search\commands.py: Contains the post processing configuration venv\Lib\site-packages\redis\commands\search\commands.py: - **cursor**: HybridCursorQuery object - contains the cursor configuration venv\Lib\site-packages\redis\commands\search\commands.py: reason="deprecated since Redis 8.0, call config_set from core module instead", venv\Lib\site-packages\redis\commands\search\commands.py: def config_set(self, option: str, value: str) -> bool: venv\Lib\site-packages\redis\commands\search\commands.py: """Set runtime configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: - **option**: the name of the configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: - **value**: a value for the configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: For more information see `FT.CONFIG SET` <<https://redis.io/commands/ft.config-set>>_. venv\Lib\site-packages\redis\commands\search\commands.py: cmd = [CONFIG_CMD, "SET", option, value] venv\Lib\site-packages\redis\commands\search\commands.py: reason="deprecated since Redis 8.0, call config_get from core module instead", venv\Lib\site-packages\redis\commands\search\commands.py: def config_get(self, option: str) -> str: venv\Lib\site-packages\redis\commands\search\commands.py: """Get runtime configuration option value. venv\Lib\site-packages\redis\commands\search\commands.py: - **option**: the name of the configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: For more information see `FT.CONFIG GET` <<https://redis.io/commands/ft.config-get>>_. venv\Lib\site-packages\redis\commands\search\commands.py: cmd = [CONFIG_CMD, "GET", option] venv\Lib\site-packages\redis\commands\search\commands.py: return self._parse_results(CONFIG_CMD, res) venv\Lib\site-packages\redis\commands\search\commands.py: post_processing: Optional[HybridPostProcessingConfig] = None, venv\Lib\site-packages\redis\commands\search\commands.py: - **post_processing**: HybridPostProcessingConfig object venv\Lib\site-packages\redis\commands\search\commands.py: Contains the post processing configuration venv\Lib\site-packages\redis\commands\search\commands.py: - **cursor**: HybridCursorQuery object - contains the cursor configuration venv\Lib\site-packages\redis\commands\search\commands.py: reason="deprecated since Redis 8.0, call config_set from core module instead", venv\Lib\site-packages\redis\commands\search\commands.py: async def config_set(self, option: str, value: str) -> bool: venv\Lib\site-packages\redis\commands\search\commands.py: """Set runtime configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: - **option**: the name of the configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: - **value**: a value for the configuration option. venv\Lib\site-packages\redis\commands\search\commands.py: For more information see `FT.CONFIG SET` <<https://redis.io/commands/ft.config-set>>_.

```

venv\Lib\site-packages\redis\commands\search\commands.py: cmd = [CONFIG_CMD, "SET", option, value] venv\Lib\site-
packages\redis\commands\search\commands.py: reason="deprecated since Redis 8.0, call config_get from core module instead", venv\Lib\site-
packages\redis\commands\search\commands.py: async def config_get(self, option: str) -> str: venv\Lib\site-packages\redis\commands\search\commands.py:
""""Get runtime configuration option value. venv\Lib\site-packages\redis\commands\search\commands.py: - **option**: the name of the configuration option.
venv\Lib\site-packages\redis\commands\search\commands.py: For more information see FT.CONFIG GET <https://redis.io/commands/ft.config-get>_.
venv\Lib\site-packages\redis\commands\search\commands.py: cmd = [CONFIG_CMD, "GET", option] venv\Lib\site-
packages\redis\commands\search\commands.py: return self._parse_results(CONFIG_CMD, res) venv\Lib\site-
packages\redis\commands\search\hybrid_query.py:class HybridPostProcessingConfig: venv\Lib\site-packages\redis\commands\search\hybrid_query.py: Create
a new hybrid post processing configuration object. venv\Lib\site-packages\redis\commands\search\__init__.py: CONFIG_CMD, venv\Lib\site-
packages\redis\commands\search\__init__.py: CONFIG_CMD: self._parse_config_get, venv\Lib\site-packages\redis\commands\sentinel.py: configuration will
be published so that the other Sentinels will venv\Lib\site-packages\redis\commands\sentinel.py: update their configurations). venv\Lib\site-
packages\redis\commands\sentinel.py: Check if the current Sentinel configuration is able to reach the venv\Lib\site-packages\redis\commands\sentinel.py: def
sentinel_flushconfig(self): venv\Lib\site-packages\redis\commands\sentinel.py: Force Sentinel to rewrite its configuration on disk, including the venv\Lib\site-
packages\redis\commands\sentinel.py: Normally Sentinel rewrites the configuration every time something venv\Lib\site-packages\redis\commands\sentinel.py:
However sometimes it is possible that the configuration file is lost venv\Lib\site-packages\redis\commands\sentinel.py: configuration managers. In those cases
a way to to force Sentinel to venv\Lib\site-packages\redis\commands\sentinel.py: rewrite the configuration file is handy. venv\Lib\site-
packages\redis\commands\sentinel.py: This command works even if the previous configuration file is venv\Lib\site-packages\redis\commands\sentinel.py:
return self.execute_command("SENTINEL FLUSHCONFIG") venv\Lib\site-packages\redis\commands\timeseries\info.py:
https://redis.io/docs/latest/develop/data-types/timeseries/configuration/#duplicate\_policy venv\Lib\site-packages\redis\connection.py:
MaintNotificationsConfig, venv\Lib\site-packages\redis\connection.py: # the command name, e.g., 'CONFIG GET'. The Redis server expects these
venv\Lib\site-packages\redis\connection.py: maint_notifications_config: Optional[MaintNotificationsConfig], venv\Lib\site-packages\redis\connection.py:
maint_notifications_config (MaintNotificationsConfig): The configuration for maintenance notifications. venv\Lib\site-packages\redis\connection.py:
self.maint_notifications_config = maint_notifications_config venv\Lib\site-packages\redis\connection.py: self._configure_maintenance_notifications(
venv\Lib\site-packages\redis\connection.py: def _configure_maintenance_notifications( venv\Lib\site-packages\redis\connection.py: not
self.maint_notifications_config venv\Lib\site-packages\redis\connection.py: or not self.maint_notifications_config.enabled venv\Lib\site-
packages\redis\connection.py: "To configure maintenance notifications, a parser must be provided!" venv\Lib\site-packages\redis\connection.py:
MaintNotificationsConnectionHandler(self, self.maint_notifications_config) venv\Lib\site-packages\redis\connection.py: self,
maint_notifications_pool_handler.config venv\Lib\site-packages\redis\connection.py: self.maint_notifications_connection_handler.config = ( venv\Lib\site-
packages\redis\connection.py: maint_notifications_pool_handler.config venv\Lib\site-packages\redis\connection.py: # When the maint_notifications_config
enabled mode is "auto", venv\Lib\site-packages\redis\connection.py: and self.maint_notifications_config venv\Lib\site-packages\redis\connection.py: and
self.maint_notifications_config.enabled venv\Lib\site-packages\redis\connection.py: maint_notifications_config=self.maint_notifications_config, venv\Lib\site-
packages\redis\connection.py: self, maint_notifications_config: MaintNotificationsConfig, check_health=True venv\Lib\site-packages\redis\connection.py:
endpoint_type = maint_notifications_config.get_endpoint_type(host, self) venv\Lib\site-packages\redis\connection.py: and maint_notifications_config.enabled
== "auto" venv\Lib\site-packages\redis\connection.py: maint_notifications_config: Optional[MaintNotificationsConfig] = None, venv\Lib\site-
packages\redis\connection.py: maint_notifications_config, venv\Lib\site-packages\redis\connection.py: # if enabled in the configuration venv\Lib\site-
packages\redis\connection.py: self._conn.maint_notifications_config, venv\Lib\site-packages\redis\connection.py: maint_notifications_config:
Optional[MaintNotificationsConfig] = None, venv\Lib\site-packages\redis\connection.py: if maint_notifications_config is None and is_protocol_supported:

```

```

venv\Lib\site-packages\redis\connection.py: maint_notifications_config = MaintNotificationsConfig() venv\Lib\site-packages\redis\connection.py: if
maint_notifications_config and maint_notifications_config.enabled: venv\Lib\site-packages\redis\connection.py: self, maint_notifications_config venv\Lib\site-
packages\redis\connection.py: The maintenance notifications config is stored in the pool handler. venv\Lib\site-packages\redis\connection.py:
maint_notifications_config = ( venv\Lib\site-packages\redis\connection.py: self._maint_notifications_pool_handler.config venv\Lib\site-
packages\redis\connection.py: return maint_notifications_config and maint_notifications_config.enabled venv\Lib\site-packages\redis\connection.py: def
update_maint_notifications_config( venv\Lib\site-packages\redis\connection.py: self, maint_notifications_config: MaintNotificationsConfig venv\Lib\site-
packages\redis\connection.py: Updates the maintenance notifications configuration. venv\Lib\site-packages\redis\connection.py: and not
maint_notifications_config.enabled venv\Lib\site-packages\redis\connection.py: self, maint_notifications_config venv\Lib\site-packages\redis\connection.py:
self._maint_notifications_pool_handler.config = maint_notifications_config venv\Lib\site-packages\redis\connection.py:
self._update_maint_notifications_configs_for_connections( venv\Lib\site-packages\redis\connection.py: "maint_notifications_config":
maint_notifications_pool_handler.config, venv\Lib\site-packages\redis\connection.py: # configured the original values yet venv\Lib\site-
packages\redis\connection.py: def _update_maint_notifications_configs_for_connections( venv\Lib\site-packages\redis\connection.py: """Update the
maintenance notifications config for all connections in the pool.""" venv\Lib\site-packages\redis\connection.py: conn.maint_notifications_config = (
venv\Lib\site-packages\redis\connection.py: maint_notifications_pool_handler.config venv\Lib\site-packages\redis\connection.py:
conn.maint_notifications_config = ( venv\Lib\site-packages\redis\connection.py: maint_notifications_pool_handler.config venv\Lib\site-
packages\redis\connection.py: "connected_address", "configured_address", "notification_hash" venv\Lib\site-packages\redis\connection.py: elif
matching_pattern == "configured_address": venv\Lib\site-packages\redis\connection.py: "connected_address", "configured_address", "notification_hash"
venv\Lib\site-packages\redis\connection.py: If `maint_notifications_config` is provided, the connection pool will support venv\Lib\site-
packages\redis\connection.py: If the `maint_notifications_config` is not provided but the protocol is 3, venv\Lib\site-
packages\redis\connection.py: Return a connection pool configured from the given URL. venv\Lib\site-packages\redis\connection.py:
maint_notifications_config: Optional[MaintNotificationsConfig] = None, venv\Lib\site-packages\redis\connection.py: if
connection_kwargs.get("cache_config") or connection_kwargs.get("cache"): venv\Lib\site-packages\redis\connection.py:
self._connection_kwargs.get("cache_config") venv\Lib\site-packages\redis\connection.py: connection_kwargs.pop("cache_config", None) venv\Lib\site-
packages\redis\connection.py: maint_notifications_config=maint_notifications_config, venv\Lib\site-packages\redis\http\http_client.py: retry: Retry
configuration for failed requests. venv\Lib\site-packages\redis\http\http_client.py: # Load client certificate for mTLS if configured venv\Lib\site-
packages\redis\maint_notifications.py:class MaintNotificationsConfig: venv\Lib\site-packages\redis\maint_notifications.py: Configuration class for
maintenance notifications handling behaviour. Notifications are received through venv\Lib\site-packages\redis\maint_notifications.py: Initialize a new
MaintNotificationsConfig. venv\Lib\site-packages\redis\maint_notifications.py: If None, the endpoint type will be automatically determined based on the host
and TLS configuration. venv\Lib\site-packages\redis\maint_notifications.py: config: MaintNotificationsConfig, venv\Lib\site-
packages\redis\maint_notifications.py: self.config = config venv\Lib\site-packages\redis\maint_notifications.py: copy =
MaintNotificationsPoolHandler(self.pool, self.config) venv\Lib\site-packages\redis\maint_notifications.py: not self.config.proactive_reconnect venv\Lib\site-
packages\redis\maint_notifications.py: and not self.config.is_relaxed_timeouts_enabled() venv\Lib\site-packages\redis\maint_notifications.py:
self.config.proactive_reconnect venv\Lib\site-packages\redis\maint_notifications.py: or self.config.is_relaxed_timeouts_enabled() venv\Lib\site-
packages\redis\maint_notifications.py: relaxed_timeout=self.config.relaxed_timeout, venv\Lib\site-packages\redis\maint_notifications.py: if
self.config.proactive_reconnect: venv\Lib\site-packages\redis\maint_notifications.py: # Update config for new connections: venv\Lib\site-
packages\redis\maint_notifications.py: if self.config.is_relaxed_timeouts_enabled(): venv\Lib\site-packages\redis\maint_notifications.py: "socket_timeout":
self.config.relaxed_timeout, venv\Lib\site-packages\redis\maint_notifications.py: "socket_connect_timeout": self.config.relaxed_timeout, venv\Lib\site-

```

packages\redis\maint_notifications.py: reset_relaxed_timeout = self.config.is_relaxed_timeouts_enabled() venv\Lib\site-packages\redis\maint_notifications.py: reset_host_address = self.config.proactive_reconnect venv\Lib\site-packages\redis\maint_notifications.py: config: MaintNotificationsConfig, venv\Lib\site-packages\redis\maint_notifications.py: self.config = config venv\Lib\site-packages\redis\maint_notifications.py: or not self.config.is_relaxed_timeouts_enabled() venv\Lib\site-packages\redis\maint_notifications.py: tmp_relaxed_timeout=self.config.relaxed_timeout venv\Lib\site-packages\redis\maint_notifications.py: self.connection.update_current_socket_timeout(self.config.relaxed_timeout) venv\Lib\site-packages\redis\maint_notifications.py: or not self.config.is_relaxed_timeouts_enabled() venv\Lib\site-packages\redis\multidb\client.py: from redis.multidb.config import DEFAULT_GRACE_PERIOD, MultiDbConfig venv\Lib\site-packages\redis\multidb\client.py: def __init__(self, config: MultiDbConfig): venv\Lib\site-packages\redis\multidb\client.py: self._databases = config.databases() venv\Lib\site-packages\redis\multidb\client.py: config.default_health_checks() venv\Lib\site-packages\redis\multidb\client.py: if not config.health_checks venv\Lib\site-packages\redis\multidb\client.py: else config.health_checks venv\Lib\site-packages\redis\multidb\client.py: self._health_check_interval = config.health_check_interval venv\Lib\site-packages\redis\multidb\client.py: self._health_check_policy: HealthCheckPolicy = config.health_check_policy.value(venv\Lib\site-packages\redis\multidb\client.py: config.health_check_probes, config.health_check_probes_delay venv\Lib\site-packages\redis\multidb\client.py: config.default_failure_detectors() venv\Lib\site-packages\redis\multidb\client.py: if not config.failure_detectors venv\Lib\site-packages\redis\multidb\client.py: else config.failure_detectors venv\Lib\site-packages\redis\multidb\client.py: config.default_failover_strategy() venv\Lib\site-packages\redis\multidb\client.py: if config.failover_strategy is None venv\Lib\site-packages\redis\multidb\client.py: else config.failover_strategy venv\Lib\site-packages\redis\multidb\client.py: self._auto_fallback_interval = config.auto_fallback_interval venv\Lib\site-packages\redis\multidb\client.py: self._event_dispatcher = config.event_dispatcher venv\Lib\site-packages\redis\multidb\client.py: self._command_retry = config.command_retry venv\Lib\site-packages\redis\multidb\client.py: failover_attempts=config.failover_attempts, venv\Lib\site-packages\redis\multidb\client.py: failover_delay=config.failover_delay, venv\Lib\site-packages\redis\multidb\client.py: self._config = config venv\Lib\site-packages\redis\multidb\command_executor.py: from redis.multidb.config import DEFAULT_AUTO_FALLBACK_INTERVAL venv\Lib\site-packages\redis\multidb\config.py: class DatabaseConfig: venv\Lib\site-packages\redis\multidb\config.py: Dataclass representing the configuration for a database connection. venv\Lib\site-packages\redis\multidb\config.py: This class is used to store configuration settings for a database connection, venv\Lib\site-packages\redis\multidb\config.py: attributes and allows for the creation of customized configurations for various venv\Lib\site-packages\redis\multidb\config.py: from_pool (Optional[ConnectionPool]): A pre-configured connection pool to use. venv\Lib\site-packages\redis\multidb\config.py: class MultiDbConfig: venv\Lib\site-packages\redis\multidb\config.py: Configuration class for managing multiple database connections in a resilient and fail-safe manner. venv\Lib\site-packages\redis\multidb\config.py: databases_config: A list of database configurations. venv\Lib\site-packages\redis\multidb\config.py: Retrieves a collection of database clients managed by weighted configurations. venv\Lib\site-packages\redis\multidb\config.py: Initializes database clients based on the provided configuration and removes venv\Lib\site-packages\redis\multidb\config.py: with defined retry and backoff configurations. venv\Lib\site-packages\redis\multidb\config.py: databases_config: List[DatabaseConfig] venv\Lib\site-packages\redis\multidb\config.py: for database_config in self.databases_config: venv\Lib\site-packages\redis\multidb\config.py: database_config.client_kwargs.update(venv\Lib\site-packages\redis\multidb\config.py: if database_config.from_url: venv\Lib\site-packages\redis\multidb\config.py: database_config.from_url, **database_config.client_kwargs venv\Lib\site-packages\redis\multidb\config.py: elif database_config.from_pool: venv\Lib\site-packages\redis\multidb\config.py: database_config.from_pool.set_retry(venv\Lib\site-packages\redis\multidb\config.py: connection_pool=database_config.from_pool venv\Lib\site-packages\redis\multidb\config.py: client = self.client_class(**database_config.client_kwargs) venv\Lib\site-packages\redis\multidb\config.py: database_config.default_circuit_breaker() venv\Lib\site-packages\redis\multidb\config.py: if database_config.circuit is None venv\Lib\site-packages\redis\multidb\config.py: else database_config.circuit venv\Lib\site-packages\redis\multidb\config.py: weight=database_config.weight, venv\Lib\site-packages\redis\multidb\config.py: health_check_url=database_config.health_check_url, venv\Lib\site-packages\redis\multidb\config.py: database_config.weight, venv\Lib\site-

packages\redis\multidb\healthcheck.py: "Database health check url is not set. Please check DatabaseConfig for the current database." venv\Lib\site-packages\redis_parsers\base.py:# user send an AUTH cmd to a server without authorization configured venv\Lib\site-packages\redis_parsers\base.py: "configured for the default user. Are you sure " venv\Lib\site-packages\redis_parsers\base.py: "your configuration is correct?": AuthenticationError, venv\Lib\site-packages\redis_parsers\helpers.py: "config-epoch": int, venv\Lib\site-packages\redis_parsers\helpers.py:def parse_config_get(response, **options): venv\Lib\site-packages\redis_parsers\helpers.py: "CLUSTER SAVECONFIG": bool_ok, venv\Lib\site-packages\redis_parsers\helpers.py: "CLUSTER SET-CONFIG-EPOCH": bool_ok, venv\Lib\site-packages\redis_parsers\helpers.py: "CONFIG RESETSTAT": bool_ok, venv\Lib\site-packages\redis_parsers\helpers.py: "CONFIG SET": bool_ok, venv\Lib\site-packages\redis_parsers\helpers.py: "SENTINEL FLUSHCONFIG": bool_ok, venv\Lib\site-packages\redis_parsers\helpers.py: "CONFIG GET": parse_config_get, venv\Lib\site-packages\redis_parsers\helpers.py: "CONFIG GET": lambda r: { venv\Lib\site-packages\rest_framework\apps.py:from django.apps import AppConfig venv\Lib\site-packages\rest_framework\apps.py:class RestFrameworkConfig(AppConfig): venv\Lib\site-packages\rest_framework\authtoken\apps.py:from django.apps import AppConfig venv\Lib\site-packages\rest_framework\authtoken\apps.py:class AuthTokenConfig(AppConfig): venv\Lib\site-packages\rest_framework\checks.py:def pagination_system_check(app_configs, **kwargs): venv\Lib\site-packages\rest_framework\compat.py: extension_configs = { venv\Lib\site-packages\rest_framework\compat.py: extensions=extensions, extension_configs=extension_configs venv\Lib\site-packages\rest_framework\fields.py: Quantize the decimal value to the configured precision. venv\Lib\site-packages\rest_framework\filters.py:from django.core.exceptions import FieldDoesNotExist, ImproperlyConfigured venv\Lib\site-packages\rest_framework\filters.py: raise ImproperlyConfigured(msg % self.__class__.__name__) venv\Lib\site-packages\rest_framework\generics.py: method if you want to apply the configured filtering backend to the venv\Lib\site-packages\rest_framework\management\commands\generateschema.py: help = "Generates configured API schema for project." venv\Lib\site-packages\rest_framework\pagination.py: page object, or None if pagination is not configured for this view. venv\Lib\site-packages\rest_framework\relations.py:from django.core.exceptions import ImproperlyConfigured, ObjectDoesNotExist venv\Lib\site-packages\rest_framework\relations.py: attributes are not configured to correctly match the URL conf. venv\Lib\site-packages\rest_framework\relations.py: # Return the hyperlink, or error if incorrectly configured. venv\Lib\site-packages\rest_framework\relations.py: 'model in your API, or incorrectly configured the ' venv\Lib\site-packages\rest_framework\relations.py: raise ImproperlyConfigured(msg % self.view_name) venv\Lib\site-packages\rest_framework\renderers.py:from django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\renderers.py: raise ImproperlyConfigured(venv\Lib\site-packages\rest_framework\routers.py:from django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\routers.py: raise ImproperlyConfigured(msg) venv\Lib\site-packages\rest_framework\routers.py: raise ImproperlyConfigured(msg % ', '.join(not_allowed)) venv\Lib\site-packages\rest_framework\serializers.py:from django.core.exceptions import FieldDoesNotExist, ImproperlyConfigured venv\Lib\site-packages\rest_framework\serializers.py: raise ImproperlyConfigured(venv\Lib\site-packages\rest_framework\settings.py: # View configuration venv\Lib\site-packages\rest_framework\test.py:from django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\test.py: raise ImproperlyConfigured('requests must be installed in order to use RequestsClient.') venv\Lib\site-packages\rest_framework\test.py: raise ImproperlyConfigured('coreapi must be installed in order to use CoreAPIClient.') venv\Lib\site-packages\rest_framework\throttling.py:from django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\throttling.py: raise ImproperlyConfigured(msg) venv\Lib\site-packages\rest_framework\throttling.py: raise ImproperlyConfigured(msg) venv\Lib\site-packages\rest_framework\views.py: # Filter out None values which may happen in case of config / rate venv\Lib\site-packages\sqlparse\lexer.py: """The Lexer supports configurable syntax. venv\Lib\site-packages\sqlparse\lexer.py: """Clear all syntax configurations. venv\Lib\site-packages\sqlparse\lexer.py: Useful if you want to load a reduced set of syntax configurations. venv\Lib\site-packages\tutorial\asgi.py:ASGI config for tutorial project. venv\Lib\site-packages\tutorial\quickstart\apps.py:class QuickstartConfig(AppConfig): venv\Lib\site-packages\tutorial\wsgi.py:WSGI config for tutorial project. (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py makemigrations Migrations for 'chat': chat\migrations\0001_initial.py + Create model Room + Create

[illegible]

detected (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py migrate Operations to perform: Apply all migrations: admin, auth, chat, contenttypes, sessions Running migrations: No migrations to apply. (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> py manage.py runserver Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 17:03:21 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/> Not Found: / [07/Feb/2026 17:03:24] "GET / HTTP/1.1" 404 2305 Not Found: / [07/Feb/2026 17:03:25] "GET / HTTP/1.1" 404 2305 Not Found: / [07/Feb/2026 17:03:27] "GET / HTTP/1.1" 404 2305 [07/Feb/2026 17:04:21] "GET /admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:04:22] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 4252 [07/Feb/2026 17:04:22] "GET /static/admin/css/nav_sidebar.css HTTP/1.1" 200 2810 [07/Feb/2026 17:04:22] "GET /static/admin/css/base.css HTTP/1.1" 200 22120 [07/Feb/2026 17:04:22] "GET /static/admin/css/login.css HTTP/1.1" 200 951 [07/Feb/2026 17:04:22] "GET /static/admin/css/dark_mode.css HTTP/1.1" 200 2808 [07/Feb/2026 17:04:22] "GET /static/admin/css/responsive.css HTTP/1.1" 200 16565 [07/Feb/2026 17:04:22] "GET /static/admin/js/theme.js HTTP/1.1" 200 1653 [07/Feb/2026 17:04:22] "GET /static/admin/js/nav_sidebar.js HTTP/1.1" 200 3063 Not Found: /favicon.ico [07/Feb/2026 17:04:22] "GET /favicon.ico HTTP/1.1" 404 2356 Not Found: /api/ [07/Feb/2026 17:04:37] "GET /api/ HTTP/1.1" 404 2739 Not Found: / [07/Feb/2026 17:04:48] "GET / HTTP/1.1" 404 2305 [07/Feb/2026 17:05:01] "GET /admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:05:01] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 4252 Not Found: /api/ [07/Feb/2026 17:05:12] "GET /api/ HTTP/1.1" 404 2739 Not Found: / [07/Feb/2026 17:24:49] "GET /api/rooms/ HTTP/1.1" 200 2 Not Found: /favicon.ico [07/Feb/2026 17:24:49] "GET /favicon.ico HTTP/1.1" 404 2356 (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> curl -X POST http://127.0.0.1:8000/api/rooms/ -H "Content-Type: application/json" -d '{"name":"general"}' >> ^C (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> curl -X POST http://127.0.0.1:8000/api/rooms/ -H "Content-Type: application/json" -d '{"name":"general"}' InvokeWebRequest : Impossible de lier le paramètre «Headers». Impossible de convertir la valeur «Content-Type: application/json» du type «System.String» en type «System.Collections.IDictionary». Au caractère Ligne:1 : 50 + ~~~~~ + CategoryInfo : InvalidArgument : (:) [InvokeWebRequest], ParameterBindingException + FullyQualifiedErrorId : CannotConvertArgumentNoMessage,Microsoft.PowerShell.Commands.InvokeWebRequestCommand (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/" >> -Method Post >> -ContentType "application/json" Invoke-RestMethod : Impossible de se connecter au serveur distant Au caractère Ligne:1 : 1 + Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/" + ~~~~~ + CategoryInfo : NotSpecified : (:) [Invoke-RestMethod], WebException + FullyQualifiedErrorId : System.Net.WebException,Microsoft.PowerShell.Commands.InvokeRestMethodCommand (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> py manage.py runserver Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 17:30:20 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/> Not Found: / [07/Feb/2026 17:30:24] "GET / HTTP/1.1" 404 2305 Not Found: / [07/Feb/2026 17:30:25] "GET / HTTP/1.1" 404 2305 Not Found: /ws/chat/general/ [07/Feb/2026 17:31:01] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:31:04] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:31:24] "GET /api/rooms/ HTTP/1.1" 200 2 Not Found: /ws/chat/general/ [07/Feb/2026 17:31:32] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:31:58] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:32:00] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:32:02] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:32:04] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:21] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found:

/ws/chat/general/ [07/Feb/2026 17:35:22] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:22] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:32] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:33] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:35] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /api/ \u0022 WS: ws://127.0.0.1:8000/ws/chat/<room>/ [07/Feb/2026 17:35:44] "GET /api/%20%E2%80%A2%20WS:%20ws://127.0.0.1:8000/ws/chat/%3Croom%3E/ HTTP/1.1" 404 2899 Not Found: /api/ [07/Feb/2026 17:35:51] "GET /api/ HTTP/1.1" 404 2739 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:43] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:44] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:44] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:45] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:46] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:47] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:38:30] "GET /api/rooms/ HTTP/1.1" 200 2 Not Found: /favicon.ico [07/Feb/2026 17:38:30] "GET /favicon.ico HTTP/1.1" 404 2356 Not Found: / [07/Feb/2026 17:38:52] "GET / HTTP/1.1" 404 2305 [07/Feb/2026 17:41:00] "GET /admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:41:00] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 4252 [07/Feb/2026 17:41:01] "GET /static/admin/css/dark_mode.css HTTP/1.1" 200 2808 [07/Feb/2026 17:41:01] "GET /static/admin/css/base.css HTTP/1.1" 200 22120 [07/Feb/2026 17:41:01] "GET /static/admin/css/nav_sidebar.css HTTP/1.1" 200 2810 [07/Feb/2026 17:41:01] "GET /static/admin/js/theme.js HTTP/1.1" 200 1653 [07/Feb/2026 17:41:01] "GET /static/admin/js/nav_sidebar.js HTTP/1.1" 200 3063 [07/Feb/2026 17:41:01] "GET /static/admin/css/login.css HTTP/1.1" 200 951 [07/Feb/2026 17:41:01] "GET /static/admin/css/responsive.css HTTP/1.1" 200 16565 [07/Feb/2026 17:41:35] "POST /admin/login/?next=/admin/ HTTP/1.1" 200 4498 [07/Feb/2026 17:44:04] "POST /admin/login/?next=/admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:44:04] "GET /admin/ HTTP/1.1" 200 7885 [07/Feb/2026 17:44:05] "GET /static/admin/css/dashboard.css HTTP/1.1" 200 441 [07/Feb/2026 17:44:05] "GET /static/admin/img/icon-changelink.svg HTTP/1.1" 200 380 [07/Feb/2026 17:44:05] "GET /static/admin/img/icon-addlink.svg HTTP/1.1" 200 331 [07/Feb/2026 17:44:27] "GET /admin/chat/room/add/ HTTP/1.1" 200 9637 [07/Feb/2026 17:44:27] "GET /static/admin/js/jquery.init.js HTTP/1.1" 200 347 [07/Feb/2026 17:44:27] "GET /static/admin/css/forms.css HTTP/1.1" 200 8525 [07/Feb/2026 17:44:27] "GET /static/admin/js/core.js HTTP/1.1" 200 6208 [07/Feb/2026 17:44:27] "GET /static/admin/js/urlify.js HTTP/1.1" 200 7887 [07/Feb/2026 17:44:27] "GET /static/admin/js/prepopulate.js HTTP/1.1" 200 1531 [07/Feb/2026 17:44:27] "GET /admin/jsi18n/ HTTP/1.1" 200 8634 [07/Feb/2026 17:44:27] "GET /static/admin/js/change_form.js HTTP/1.1" 200 606 [07/Feb/2026 17:44:27] "GET /static/admin/js/admin/RelatedObjectLookups.js HTTP/1.1" 200 9777 [07/Feb/2026 17:44:27] "GET /static/admin/js/actions.js HTTP/1.1" 200 8076 [07/Feb/2026 17:44:27] "GET /static/admin/js/prepopulate_init.js HTTP/1.1" 200 586 [07/Feb/2026 17:44:27] "GET /static/admin/css/widgets.css HTTP/1.1" 200 11973 [07/Feb/2026 17:44:28] "GET /static/admin/js/vendor/xregexp/xregexp.js HTTP/1.1" 200 325171 [07/Feb/2026 17:44:28] "GET /static/admin/js/vendor/jquery/jquery.js HTTP/1.1" 200 285314 [07/Feb/2026 17:44:44] "POST /admin/chat/room/add/ HTTP/1.1" 302 0 [07/Feb/2026 17:44:44] "GET /admin/chat/room/ HTTP/1.1" 200 10874 [07/Feb/2026 17:44:45] "GET /admin/jsi18n/ HTTP/1.1" 200 8634 [07/Feb/2026 17:44:45] "GET /static/admin/css/changelists.css HTTP/1.1" 200 6878 [07/Feb/2026 17:44:45] "GET /static/admin/js/filters.js HTTP/1.1" 200 978 [07/Feb/2026 17:44:45] "GET /static/admin/img/search.svg HTTP/1.1" 200 458 [07/Feb/2026 17:44:45] "GET /static/admin/img/tooltag-add.svg HTTP/1.1" 200 331 [07/Feb/2026 17:44:45] "GET /static/admin/img/icon-yes.svg HTTP/1.1" 200 436 Not Found: / [07/Feb/2026 17:44:58] "GET / HTTP/1.1" 404 2305 Not Found: / [07/Feb/2026 17:44:59] "GET / HTTP/1.1" 404 2305 Not Found: /ws/chat/general/ [07/Feb/2026 17:45:18] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:45:22] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:06] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:08] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:09] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:38] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:39] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:41] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:52] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026

17:47:53] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 17:47:53] "GET /api/rooms/1/messages/ HTTP/1.1" 200 2 [07/Feb/2026 17:47:54] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 17:47:54] "GET /api/rooms/1/messages/ HTTP/1.1" 200 2 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:57] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:50:18] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /favicon.ico [07/Feb/2026 17:50:18] "GET /favicon.ico HTTP/1.1" 404 2356 [07/Feb/2026 17:50:21] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /favicon.ico [07/Feb/2026 17:50:21] "GET /favicon.ico HTTP/1.1" 404 2356 Not Found: /ws/chat/general/ [07/Feb/2026 17:53:33] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:53:34] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:53:34] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:55:20] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /favicon.ico [07/Feb/2026 17:55:21] "GET /favicon.ico HTTP/1.1" 404 2356 [07/Feb/2026 17:56:52] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 17:57:19] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2467 Not Found: /ws/chat/general/ [07/Feb/2026 17:58:15] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 17:59:50] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2467 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:09] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:11] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:12] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:18] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:25] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:25] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:26] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 18:00:27] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 18:00:27] "GET /api/rooms/1/messages/ HTTP/1.1" 200 2 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:28] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:29] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:07:52] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:07:53] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:08:00] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:08:00] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:08:01] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:08:28] "GET /ws/chat/general/ HTTP/1.1" 404 2371 D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\views.py changed, reloading. Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 18:24:04 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/> D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\views.py changed, reloading. Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 18:24:10 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/> D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\monsite\urls.py changed, reloading. Watching for file changes with StatReloader Performing system checks... Exception in thread django-main-thread: Traceback (most recent call last): File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\urls\resolvers.py", line 720, in url_patterns iter(patterns) TypeError: 'module' object is not iterable The above exception was the direct cause of the following exception: Traceback (most recent call last): File "C:\Program Files\Python3_12_4\Lib\threading.py", line 1073, in _bootstrap_inner self.run() File "C:\Program Files\Python3_12_4\Lib\threading.py", line 1010, in run self._target(*self._args, **self._kwargs) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\utils\autoreload.py", line 64, in wrapper fn(*args, **kwargs) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\management\commands\runserver.py", line 134, in inner_run self.check(**check_kwargs) File "D:\PYTHON DJANGO

[illegible]

18:28:57] "GET / HTTP/1.1" 200 776 Not Found: /ws/chat/general/ [07/Feb/2026 18:28:57] "GET /ws/chat/general/ HTTP/1.1" 404 2514 [07/Feb/2026 18:29:05] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:05] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2610 [07/Feb/2026 18:29:06] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:06] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2610 [07/Feb/2026 18:29:07] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:07] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2610 [07/Feb/2026 18:29:07] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:07] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2610 [07/Feb/2026 18:29:07] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:07] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2610 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:08] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2610 [07/Feb/2026 18:29:09] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:09] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2610 (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> pip install daphne >> ^C (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> pip install daphne Collecting daphne Downloading daphne-4.2.1-py3-none-any.whl.metadata (1.7 kB) Requirement already satisfied: asgiref<4,>=3.5.2 in d:\python django hopes\chat_api_django_mario\venv\lib\site-packages (from daphne) (3.11.1) Collecting autobahn>=22.4.2 (from daphne) Downloading autobahn-25.12.2-cp312-cp312-win_amd64.whl.metadata (32 kB) Collecting twisted>=22.4 (from twisted[tls]>=22.4->daphne) Downloading twisted-25.5.0-py3-none-any.whl.metadata (22 kB) Collecting cbor2>=5.2.0 (from autobahn>=22.4.2->daphne) Downloading cbor2-5.8.0-cp312-cp312-win_amd64.whl.metadata (5.6 kB) Collecting cffi>=2.0.0 (from autobahn>=22.4.2->daphne) Using cached cffi-2.0.0-cp312-cp312-win_amd64.whl.metadata (2.6 kB) Collecting cryptography>=3.4.6 (from autobahn>=22.4.2->daphne) Downloading cryptography-46.0.4-cp311-abi3-win_amd64.whl.metadata (5.7 kB) Collecting hyperlink>=21.0.0 (from autobahn>=22.4.2->daphne) Downloading hyperlink-21.0.0-py2.py3-none-any.whl.metadata (1.5 kB) Requirement already satisfied: msgpack>=1.0.2 in d:\python django hopes\chat_api_django_mario\venv\lib\site-packages (from autobahn>=22.4.2->daphne) (1.1.2) Collecting py-ubjson>=0.16.1 (from autobahn>=22.4.2->daphne) Downloading py-ubjson-0.16.1.tar.gz (50 kB) ----- 50.3/50.3 kB 284.2 kB/s eta 0:00:00 Installing build dependencies ... done Getting requirements to build wheel ... done Preparing metadata (pyproject.toml) ... done Collecting txai0>=25.12.2 (from autobahn>=22.4.2->daphne) Downloading txai0-25.12.2-py3-none-any.whl.metadata (7.0 kB) Collecting ujson>=4.0.2 (from autobahn>=22.4.2->daphne) Downloading ujson-5.11.0-cp312-cp312-win_amd64.whl.metadata (9.6 kB) Collecting attrs>=22.2.0 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading attrs-25.4.0-py3-none-any.whl.metadata (10 kB) Collecting automat>=24.8.0 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading automat-25.4.16-py3-none-any.whl.metadata (8.4 kB) Collecting constantly>=15.1 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading constantly-23.10.4-py3-none-any.whl.metadata (1.8 kB) Collecting incremental>=24.7.0 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading incremental-24.11.0-py3-none-any.whl.metadata (9.5 kB) Collecting typing-extensions>=4.2.0 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Using cached typing_extensions-4.15.0-py3-none-any.whl.metadata (3.3 kB) Collecting zope-interface>=5 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading zope_interface-8.2-cp312-cp312-win_amd64.whl.metadata (46 kB) ----- 46.7/46.7 kB 331.3 kB/s eta 0:00:00 Collecting idna>=2.4 (from twisted[tls]>=22.4->daphne) Downloading idna-3.11-py3-none-any.whl.metadata (8.4 kB) Collecting pyopenssl>=21.0.0 (from twisted[tls]>=22.4->daphne) Downloading pyopenssl-25.3.0-py3-none-any.whl.metadata (17 kB) Collecting service-identity>=18.1.0 (from twisted[tls]>=22.4->daphne) Downloading service_identity-24.2.0-py3-none-any.whl.metadata (5.1 kB) Collecting pycparser (from cffi>=2.0.0->autobahn>=22.4.2->daphne) Downloading pycparser-3.0-py3-none-any.whl.metadata (8.2 kB) Collecting packaging>=17.0 (from incremental>=24.7.0->twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading packaging-26.0-py3-none-any.whl.metadata (3.3 kB) Collecting pyasn1 (from service-identity>=18.1.0->twisted[tls]>=22.4->daphne) Downloading pyasn1-0.6.2-py3-none-any.whl.metadata (8.4 kB) Collecting pyasn1-modules (from service-

identity>=18.1.0->twisted[tls]>=22.4->daphne) Using cached pyasn1_modules-0.4.2-py3-none-any.whl.metadata (3.5 kB) Downloading daphne-4.2.1-py3-none-any.whl (29 kB) Downloading autobahn-25.12.2-cp312-cp312-win_amd64.whl (2.2 MB) —————
2.2/2.2 MB 1.2 MB/s eta 0:00:00 Downloading twisted-25.5.0-py3-none-any.whl (3.2 MB) —————
3.2/3.2 MB 1.5 MB/s eta 0:00:00 Downloading attrs-25.4.0-py3-none-any.whl (67 kB) —————
67.6/67.6 kB 459.0 kB/s eta 0:00:00 Downloading automat-25.4.16-py3-none-any.whl (42 kB) —————
42.8/42.8 kB 299.1 kB/s eta 0:00:00 Downloading cbor2-5.8.0-cp312-cp312-win_amd64.whl (69 kB) —————
69.8/69.8 kB 546.6 kB/s eta 0:00:00 Using cached cffi-2.0.0-cp312-cp312-win_amd64.whl (183 kB) Downloading constantly-23.10.4-py3-none-any.whl (13 kB) Downloading cryptography-46.0.4-cp311-abi3-win_amd64.whl (3.5 MB) —————
3.5/3.5 MB 844.1 kB/s eta 0:00:00 Downloading hyperlink-21.0.0-py2.py3-none-any.whl (74 kB) —————
74.6/74.6 kB 825.6 kB/s eta 0:00:00 Downloading idna-3.11-py3-none-any.whl (71 kB) —————
71.0/71.0 kB 129.6 kB/s eta 0:00:00 Downloading incremental-24.11.0-py3-none-any.whl (21 kB) Downloading pyopenssl-25.3.0-py3-none-any.whl (57 kB) —————
57.3/57.3 kB 505.7 kB/s eta 0:00:00 Downloading service_identity-24.2.0-py3-none-any.whl (11 kB) Downloading txaio-25.12.2-py3-none-any.whl (31 kB) Using cached typing_extensions-4.15.0-py3-none-any.whl (44 kB) Downloading ujson-5.11.0-cp312-cp312-win_amd64.whl (43 kB) —————
43.9/43.9 kB 357.6 kB/s eta 0:00:00 Downloading zope_interface-8.2-cp312-cp312-win_amd64.whl (212 kB) —————
212.4/212.4 kB 1.2 MB/s eta 0:00:00 Downloading packaging-26.0-py3-none-any.whl (74 kB) —————
74.4/74.4 kB 372.8 kB/s eta 0:00:00 Downloading pyasn1-0.6.2-py3-none-any.whl (83 kB) —————
83.4/83.4 kB 778.0 kB/s eta 0:00:00 Using cached pyasn1_modules-0.4.2-py3-none-any.whl (181 kB) Downloading pycparser-3.0-py3-none-any.whl (48 kB) —————
48.2/48.2 kB 403.5 kB/s eta 0:00:00 Building wheels for collected packages: py-ubjson Building wheel for py-ubjson (pyproject.toml) ... done Created wheel for py-ubjson: filename=py_ubjson-0.16.1-cp312-cp312-win_amd64.whl size=19356 sha256=7b3fc1d543eefd1d8e1994a85b47c17a024a1a95741dc0ced87344b32adfb966 Stored in directory: c:\users\spsse02men\appdata\local\pip\cache\wheels\c0\8\73\e1931c0c1125bc84ae955eeb3d8ba9fe23ecd95ee337095267 Successfully built py-ubjson Installing collected packages: py-ubjson, zope-interface, ujson, typing-extensions, txaio, pycparser, pyasn1, packaging, idna, constantly, cbor2, automat, attrs, pyasn1-modules, incremental, hyperlink, cffi, twisted, cryptography, service-identity, pyopenssl, Successfully installed attrs-25.4.0 autobahn-25.12.2 automat-25.4.16 cbor2-5.8.0 cffi-2.0.0 constantly-23.10.4 cryptography-46.0.4 daphne-4.2.1 hyperlink-21.0.0 idna-3.11 incremental-24.11.0 packaging-26.0 py-ubjson-0.16.1 pyasn1-0.6.2 pyasn1-modules-0.4.2 pycparser-3.0 pyopenssl-25.3.0 service-identity-24.2.0 twisted-25.5.0 txaio-25.12.2 typing-extensions-4.15.0 ujson-5.11.0 zope-interface-8.2 [notice] A new release of pip is available: 24.0 -> 26.0.1 [notice] To update, run: python.exe -m pip install --upgrade pip (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> py manage.py runserver Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 18:34:56 Django version 5.2.8, using settings 'monsie.settings' Starting ASGI/Daphne version 4.2.1 development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. Not Found: /ws/chat/general/ HTTP GET /ws/chat/general/ 404 [0.06, 127.0.0.1:22100] Not Found: /ws/chat/general/ HTTP GET /ws/chat/general/ 404 [0.02, 127.0.0.1:22100] HTTP GET / 200 [0.01, 127.0.0.1:22100] WebSocket HANDSHAKING /ws/chat/general/ [127.0.0.1:22107]

WebSocket CONNECT /ws/chat/general/ [127.0.0.1:22107] WebSocket DISCONNECT /ws/chat/general/ [127.0.0.1:22107] HTTP GET / 200 [0.01, 127.0.0.1:22100] WebSocket HANDSHAKING /ws/chat/general/ [127.0.0.1:22110] WebSocket CONNECT /ws/chat/general/ [127.0.0.1:22110]
Réflexion durant 8s

You're basically **there** — your logs show the important clue:

- HTTP GET /ws/chat/general/ 404  **normal**
- WebSocket HANDSHAKING ... CONNECT ...  **good** (this means your Channels + Daphne path is working)

Why you see 404 on `/ws/chat/general/`

`/ws/chat/general/` is a **WebSocket endpoint**, not a normal HTTP page.

So if you type it in the browser address bar (HTTP GET), Django will answer **404**. That's expected.

To “open” it, you must use a WebSocket client (JS, Postman WS, websocat, etc.).

1) Quick test (browser console)

Open `http://127.0.0.1:8000/` (your page returns 200 now), then press **F12** → **Console** and run:

```
const ws = new WebSocket("ws://127.0.0.1:8000/ws/chat/general/");
ws.onopen = () => { console.log("WS OPEN"); ws.send(JSON.stringify({message:"hello"})); };
ws.onmessage = (e) => console.log("WS MESSAGE:", e.data);
ws.onclose = () => console.log("WS CLOSED");
ws.onerror = (e) => console.log("WS ERROR", e);
```

If your consumer echoes/broadcasts, you'll see messages.

2) Make sure your ASGI routing is correct

These are the **canonical** minimal files.

`monsite/asgi.py`

```

import os
from channels.routing import ProtocolTypeRouter, URLRouter
from django.core.asgi import get_asgi_application
from channels.auth import AuthMiddlewareStack
import chat.routing

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "monsite.settings")

django_asgi_app = get_asgi_application()

application = ProtocolTypeRouter({
    "http": django_asgi_app,
    "websocket": AuthMiddlewareStack(
        URLRouter(chat.routing.websocket_urlpatterns)
    ),
})

```

chat/routing.py

```

from django.urls import re_path
from .consumers import ChatConsumer

websocket_urlpatterns = [
    re_path(r"ws/chat/(?P<room_name>\w+)/$", ChatConsumer.as_asgi()),
]

```



Your test `python -c "... django.setup(); import chat.routing"` already succeeded, so this structure is likely close.

3) Your settings must include Channels

monsite/settings.py

```

INSTALLED_APPS = [
    # ...
    "channels",
    "chat",
    # ...
]

ASGI_APPLICATION = "monsite.asgi.application"

```

```
# Dev option (no Redis needed):  
CHANNEL_LAYERS = {  
    "default": {"BACKEND": "channels.layers.InMemoryChannelLayer"}  
}
```

If you want Redis later, switch backend to `channels_redis.core.RedisChannelLayer`.

4) Daphne message you're seeing is a good sign

This part confirms it:

- Starting ASGI/Daphne ...
- WebSocket HANDSHAKING ...
- WebSocket CONNECT ...

So the main remaining “problem” is just: **don't expect `/ws/...` to be a browsable page.**

5) PowerShell “curl” issue (your POST to `/api/rooms/`)

In PowerShell, `curl` is an alias for `Invoke-WebRequest`, so `-H "Content-Type: ..."` breaks.

Use **either**:

Option A: real curl

```
curl.exe -X POST "http://127.0.0.1:8000/api/rooms/" -H "Content-Type: application/json" -d '{"name":"general"}"
```

Option B: Invoke-RestMethod properly

```
Invoke-RestMethod `
    -Uri "http://127.0.0.1:8000/api/rooms/" `
    -Method Post `
    -ContentType "application/json" `
    -Body '{"name":"general"}'
```



If you get “Impossible de se connecter...”, it simply means **the server wasn’t running at that moment**.

The only thing I’d change based on your logs

Stop testing `/ws/chat/general/` by opening it like a normal URL. Test it via WebSocket (console snippet above), and you’re good.

If you paste your `monsite/asgi.py`, `monsite/settings.py` (Channels parts), and `chat/consumers.py`, I can spot-check in one pass.

Chat en ligne (Django + Channels) REST: `http://127.0.0.1:8000/api/` • WS: `ws://127.0.0.1:8000/ws/chat/<room>/` system • 07/02/2026 18:39:50 Connecté au salon "general" system • 07/02/2026 18:39:57 Déconnecté system • 07/02/2026 18:39:57 Connecté au salon "general" system • 07/02/2026 18:39:59 Déconnecté system • 07/02/2026 18:39:59 Connecté au salon "general"

Parfait — ça veut dire que **ton WebSocket marche**  : ton client arrive à se connecter à `ws://127.0.0.1:8000/ws/chat/general/`.


Le seul souci visible maintenant, c’est le **cycle “connecté → déconnecté → reconnecté”** (souvent dû au front qui recrée la socket, à un refresh, ou à une fermeture côté serveur après une erreur).

Voici les causes **les plus fréquentes** + comment vérifier rapidement.

1) Tu ouvres plusieurs connexions (double “Connect”)

Si ton JS fait un `connect()` :

- au chargement (`DOMContentLoaded`)
- **et** quand tu cliques “Join”
- **et** aussi dans un “reconnect loop”
→ tu verras exactement ce pattern.

 Check : dans le code JS, cherche `new WebSocket()` et assure-toi qu’il n’est appelé qu’une fois, ou qu’avant de reconnecter tu fermes l’ancien.

Bon pattern :

```
if (window.ws && (ws.readyState === WebSocket.OPEN || ws.readyState === WebSocket.CONNECTING)) {
    return; // déjà connecté
}
window.ws = new WebSocket(url);
```

2) Le client ferme lui-même (socket.close / navigation)

Si tu changes de page, reload, ou si ton code appelle `ws.close()` dans un handler, tu verras “Déconnecté”.

✅ Check : cherche `close()` et aussi un éventuel `beforeunload` :

```
window.addEventListener("beforeunload", () => ws?.close());
```

3) Erreur dans le Consumer → déconnexion immédiate

Quand le consumer plante, Channels ferme souvent la socket.

✅ Check : regarde ton terminal `runserver` :

- tu devrais voir une `stacktrace` si `receive()` ou `connect()` plante.
- et côté navigateur : onglet Network → WS → Frames / Messages, ou Console `ws.onerror`.

Consumer minimal stable (echo) :

```
import json
from channels.generic.websocket import AsyncWebsocketConsumer

class ChatConsumer(AsyncWebsocketConsumer):
    async def connect(self):
        self.room_name = self.scope["url_route"]["kwargs"]["room_name"]
        self.group_name = f"chat_{self.room_name}"
        await self.channel_layer.group_add(self.group_name, self.channel_name)
        await self.accept()

    async def disconnect(self, close_code):
        await self.channel_layer.group_discard(self.group_name, self.channel_name)
```

```
async def receive(self, text_data):
    data = json.loads(text_data)
    msg = data.get("message", "")
    await self.channel_layer.group_send(
        self.group_name,
        {"type": "chat.message", "message": msg}
    )

async def chat_message(self, event):
    await self.send(text_data=json.dumps({"message": event["message"]}))
```

4) Tu utilises HTTP au lieu de WS par moment

Si ton front essaie d'atteindre `/ws/chat/general/` par **fetch** ou en mettant l'URL dans un lien → ça fera des 404 / comportements bizarres.

✅ Check : Network → tu dois voir une requête **WebSocket** (status 101 Switching Protocols), pas un GET normal.

5) Reconnect automatique trop agressif

Si ton JS fait :

```
ws.onclose = () => connect();
```

ça reconnecte immédiatement et donne exactement tes logs.

✅ Bon pattern :

```
ws.onclose = () => setTimeout(connect, 1000);
```

Le test le plus simple

1. Ouvre la page
2. Dans la console, affiche les events :

```
ws.onopen = () => console.log("open");
ws.onclose = (e) => console.log("close", e.code, e.reason);
ws.onerror = (e) => console.log("error", e);
```

➡ Le **code close** aide énormément :

- 1000 = fermeture normale (souvent ton code / reload)
- 1006 = fermeture anormale (souvent crash / réseau)

Si tu me colles **ton JS de connexion WebSocket** (la partie `new WebSocket, onclose, onmessage`) + ton `ChatConsumer`, je te dis exactement pourquoi ça déconnecte et comment le stabiliser.

Pouvez vous donner le fichier corrigé ?

Oui — je peux te donner **les fichiers corrigés** “prêts à coller” pour que :

- `/ws/chat/<room>/` fonctionne **sans 404**
- la connexion WS soit stable (pas de loop “connecté/déconnecté” côté serveur)
- ton projet Channels soit correctement câblé (ASGI + routing)

Comme je n’ai pas ton code exact, je te donne une **version corrigée standard** (celle qui marche dans 99% des setups Django 5 + Channels 4). Tu remplaces/alignes tes fichiers avec ça.

1) `monsite/asgi.py` (corrigé)

```
import os
from django.core.asgi import get_asgi_application

from channels.auth import AuthMiddlewareStack
from channels.routing import ProtocolTypeRouter, URLRouter

import chat.routing  # IMPORTANT: importe le routing websocket

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "monsite.settings")
```

```
django_asgi_app = get_asgi_application()

application = ProtocolTypeRouter(
    {
        "http": django_asgi_app,
        "websocket": AuthMiddlewareStack(
            URLRouter(chat.routing.websocket_urlpatterns)
        ),
    }
)
```

2) chat/routing.py (corrigé)

```
from django.urls import re_path
from .consumers import ChatConsumer

websocket_urlpatterns = [
    re_path(r"ws/chat/(?P<room_name>\w+)/$", ChatConsumer.as_asgi()),
]
```



Avec ça, ws://127.0.0.1:8000/ws/chat/general/ match bien.

3) chat/consumers.py (corrigé, stable)

```
import json
from channels.generic.websocket import AsyncWebsocketConsumer

class ChatConsumer(AsyncWebsocketConsumer):
    async def connect(self):
        self.room_name = self.scope["url_route"]["kwargs"]["room_name"]
        self.room_group_name = f"chat_{self.room_name}"

        await self.channel_layer.group_add(
            self.room_group_name,
            self.channel_name
        )
```

```

    await self.accept()

    # message système optionnel
    await self.send(text_data=json.dumps({
        "type": "system",
        "message": f"Connecté au salon \"{self.room_name}\""
    }))

    async def disconnect(self, close_code):
        await self.channel_layer.group_discard(
            self.room_group_name,
            self.channel_name
        )

    async def receive(self, text_data):
        data = json.loads(text_data)
        message = data.get("message", "")

        await self.channel_layer.group_send(
            self.room_group_name,
            {
                "type": "chat_message",
                "message": message,
            }
        )

    async def chat_message(self, event):
        await self.send(text_data=json.dumps({
            "type": "chat",
            "message": event["message"]
        }))

```

4) monsite/settings.py (les lignes importantes)

Assure-toi d'avoir :

```

INSTALLED_APPS = [
    # ...
    "channels",
    "chat",
    # ...
]

```

```
ASGI_APPLICATION = "monsite.asgi.application"

CHANNEL_LAYERS = {
    "default": {
        "BACKEND": "channels.layers.InMemoryChannelLayer",
    }
}
```

✅ InMemory suffit pour tester en local (pas besoin de Redis).

5) monsite/urls.py (évite l’erreur “urlpatterns module not iterable”)

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path("admin/", admin.site.urls),
    path("api/", include("chat.urls")), # si tu as chat/urls.py pour REST
]
```

Bonus : pourquoi tu avais 404 avant ?

Tu avais **Daphne** (donc WS OK), mais si **ASGI_APPLICATION** / `asgi.py` n’était pas câblé avec `ProtocolTypeRouter` + `URLRouter`, alors Django traitait `/ws/...` comme du HTTP normal ⇒ **404**.

Si tu veux que je “corrige TES fichiers exacts”

Colle ici le contenu de :

- monsite/asgi.py
- monsite/settings.py (parties Channels seulement)
- chat/routing.py

- chat/consumers.py
- monsite/urls.py

Et je te renvoie **la version corrigée exacte** adaptée à ton projet (sans hypothèses).

asgi.py

Python

monsite/asgi.py

Réflexion durant 9s

Voici une version **corrigée / propre** de monsite/asgi.py (l'ordre est important : on configure DJANGO_SETTINGS_MODULE, on initialise Django via get_asgi_application(), **puis** on importe chat.routing pour éviter l'erreur "settings are not configured").

asgi

```
import os

from django.core.asgi import get_asgi_application
from channels.auth import AuthMiddlewareStack
from channels.routing import ProtocolTypeRouter, URLRouter

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "monsite.settings")

# Initialise Django (charge settings/apps)
django_asgi_app = get_asgi_application()

# IMPORTANT : importer après l'init Django pour éviter ImproperlyConfigured
import chat.routing

application = ProtocolTypeRouter({
    "http": django_asgi_app,
    "websocket": AuthMiddlewareStack(
        URLRouter(chat.routing.websocket_urlpatterns)
    ),
})
```

Si tu remplaces ton fichier par celui-ci, ton endpoint WebSocket /ws/chat/<room>/ sera bien géré par Channels (et ne tombera plus en 404 "HTTP" quand la connexion WebSocket est faite correctement).

Sources
settings.py
Python
monsite/settings.py
Réflexion durant 36s

Voici une version **corrigée et plus robuste** de `monsite/asgi.py` (le point clé : **faire `django.setup()` avant d'importer `chat.routing`**, sinon tu peux avoir des erreurs/instabilités quand `routing/consumers/models` sont importés trop tôt).

```
# monsite/asgi.py
import os
import django

from channels.auth import AuthMiddlewareStack
from channels.routing import ProtocolTypeRouter, URLRouter
from django.core.asgi import get_asgi_application

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "monsite.settings")

# IMPORTANT: initialise Django AVANT d'importer routing/consumers/models
django.setup()

import chat.routing  # noqa: E402

django_asgi_app = get_asgi_application()

application = ProtocolTypeRouter(
    {
        "http": django_asgi_app,
        "websocket": AuthMiddlewareStack(
            URLRouter(chat.routing.websocket_urlpatterns)
        ),
    }
)
```



Tu peux remplacer ton fichier `monsite/asgi.py` par celui-ci, puis relancer :

```
py manage.py runserver
```

(Si tu continues à voir “Connecté / Déconnecté” en boucle côté front, ce sera plutôt lié au code JS client ou au `ChatConsumer`, mais au moins la config ASGI/Channels sera propre.)

routing.py
Python
chat/routing.py
Réflexion durant 26s

Voici des versions **propres/corrigées** de tes 3 fichiers pour que **Django Channels + WebSocket** fonctionnent correctement avec l'URL :

- REST : `http://127.0.0.1:8000/api/`
- WS : `ws://127.0.0.1:8000/ws/chat/<room>/`



monsite/asgi.py

```
import os

from channels.auth import AuthMiddlewareStack
from channels.routing import ProtocolTypeRouter, URLRouter
from django.core.asgi import get_asgi_application

import chat.routing

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "monsite.settings")

django_asgi_app = get_asgi_application()

application = ProtocolTypeRouter(
    {
        "http": django_asgi_app,
        "websocket": AuthMiddlewareStack(
            URLRouter(chat.routing.websocket_urlpatterns)
        ),
    }
)
```



monsite/settings.py (parties importantes)

Tu peux copier/coller ces blocs dans ton fichier, ou remplacer directement si ton settings est petit.


```


INSTALLED_APPS = [
    "django.contrib.admin",
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",

    # apps tierces
    "rest_framework",
    "channels",

    # ton app
    "chat",
]

ASGI_APPLICATION = "monsite.asgi.application"

#  Option 1 (simple) : InMemory (OK en dev, 1 seul process)
CHANNEL_LAYERS = {
    "default": {"BACKEND": "channels.layers.InMemoryChannelLayer"}
}

#  Option 2 (recommandé) : Redis (si tu utilises channels_redis)
# CHANNEL_LAYERS = {
#     "default": {
#         "BACKEND": "channels_redis.core.RedisChannelLayer",
#         "CONFIG": {"hosts": [("127.0.0.1", 6379)]},
#     }
# }

```



chat/routing.py

```

from django.urls import re_path
from .consumers import ChatConsumer

websocket_urlpatterns = [
    re_path(r"^ws/chat/(?P<room_name>[/]+)/$", ChatConsumer.as_asgi()),
]

```

⚠ Important (cause fréquente de “connecté / déconnecté”)

Si ton navigateur fait un **GET HTTP** vers `/ws/chat/general/`, tu verras du **404** (normal). Il faut bien ouvrir via **WebSocket** côté JS :

```
const ws = new WebSocket("ws://127.0.0.1:8000/ws/chat/general/");
```

Si tu veux, envoie aussi `chat/consumers.py` (le fichier le plus critique). C’est souvent là que se cache la raison des déconnexions immédiates (exception silencieuse, `group_name` invalide, JSON mal géré, etc.).

consumers.py
Python
chat/consumers.py
Réflexion durant 22s

Oui — voici les **versions corrigées** des fichiers que vous avez envoyés (ASGI + Channels). Copiez/collez exactement.

monsite/asgi.py

```
import os

from django.core.asgi import get_asgi_application
from channels.routing import ProtocolTypeRouter, URLRouter
from channels.auth import AuthMiddlewareStack

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "monsite.settings")

django_asgi_app = get_asgi_application()

# IMPORTANT: importer routing après l'init Django
import chat.routing

application = ProtocolTypeRouter({
    "http": django_asgi_app,
    "websocket": AuthMiddlewareStack(
        URLRouter(chat.routing.websocket_urlpatterns)
    ),
})
```

)

monsite/settings.py

```
from pathlib import Path

BASE_DIR = Path(__file__).resolve().parent.parent

SECRET_KEY = "dev-secret-key-change-me"
DEBUG = True
ALLOWED_HOSTS = ["127.0.0.1", "localhost"]

INSTALLED_APPS = [
    "daphne", # <- important: avant tout pour runserver ASGI

    "django.contrib.admin",
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",

    "corsheaders",
    "rest_framework",
    "channels",

    "chat",
]

MIDDLEWARE = [
    "corsheaders.middleware.CorsMiddleware",
    "django.middleware.security.SecurityMiddleware",
    "django.contrib.sessions.middleware.SessionMiddleware",
    "django.middleware.common.CommonMiddleware",
    "django.middleware.csrf.CsrfViewMiddleware",
    "django.contrib.auth.middleware.AuthenticationMiddleware",
    "django.contrib.messages.middleware.MessageMiddleware",
    "django.middleware.clickjacking.XFrameOptionsMiddleware",
]

ROOT_URLCONF = "monsite.urls"

TEMPLATES = [
    {
```

```

    "BACKEND": "django.template.backends.django.DjangoTemplates",
    "DIRS": [],
    "APP_DIRS": True,
    "OPTIONS": {
        "context_processors": [
            "django.template.context_processors.request",
            "django.contrib.auth.context_processors.auth",
            "django.contrib.messages.context_processors.messages",
        ],
    },
],

WSGI_APPLICATION = "monsite.wsgi.application"
ASGI_APPLICATION = "monsite.asgi.application"

DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.sqlite3",
        "NAME": BASE_DIR / "db.sqlite3",
    }
}

AUTH_PASSWORD_VALIDATORS = [
    {"NAME": "django.contrib.auth.password_validation.UserAttributeSimilarityValidator"},
    {"NAME": "django.contrib.auth.password_validation.MinimumLengthValidator"},
    {"NAME": "django.contrib.auth.password_validation.CommonPasswordValidator"},
    {"NAME": "django.contrib.auth.password_validation.NumericPasswordValidator"},
]

LANGUAGE_CODE = "fr-fr"
TIME_ZONE = "Indian/Antananarivo"
USE_I18N = True
USE_TZ = True

STATIC_URL = "static/"
DEFAULT_AUTO_FIELD = "django.db.models.BigAutoField"

# Dev CORS
CORS_ALLOW_ALL_ORIGINS = True

# Channels (DEV): InMemory
CHANNEL_LAYERS = {
    "default": {
        "BACKEND": "channels.layers.InMemoryChannelLayer",
    }
}

```

```
}

REST_FRAMEWORK = {
    "DEFAULT_RENDERER_CLASSES": ["rest_framework.renderers.JSONRenderer"],
}
```

chat/routing.py

```
from django.urls import re_path
from .consumers import ChatConsumer

websocket_urlpatterns = [
    re_path(r"ws/chat/(?P<room_name>[^/]+)/$", ChatConsumer.as_asgi()),
]
```

chat/consumers.py

```
import json

from channels.generic.websocket import AsyncWebsocketConsumer
from channels.db import database_sync_to_async

from .models import Room, Message

class ChatConsumer(AsyncWebsocketConsumer):
    async def connect(self):
        self.room_name = self.scope["url_route"]["kwargs"]["room_name"]
        self.group_name = f"chat_{self.room_name}"

        await self.channel_layer.group_add(self.group_name, self.channel_name)
        await self.accept()

        await self.get_or_create_room(self.room_name)

    async def disconnect(self, close_code):
        await self.channel_layer.group_discard(self.group_name, self.channel_name)

    async def receive(self, text_data=None, bytes_data=None):
        if not text_data:
            return
```

```

data = json.loads(text_data)
author = (data.get("author") or "anonymous").strip()[:80]
content = (data.get("content") or "").strip()
if not content:
    return

msg = await self.create_message(self.room_name, author, content)

await self.channel_layer.group_send(
    self.group_name,
    {
        "type": "chat_message",
        "id": msg["id"],
        "author": msg["author"],
        "content": msg["content"],
        "created_at": msg["created_at"],
    },
)

async def chat_message(self, event):
    await self.send(text_data=json.dumps(event))

@database_sync_to_async
def get_or_create_room(self, name: str):
    return Room.objects.get_or_create(name=name)

@database_sync_to_async
def create_message(self, room_name: str, author: str, content: str):
    room = Room.objects.get(name=room_name)
    m = Message.objects.create(room=room, author=author, content=content)
    return {
        "id": m.id,
        "author": m.author,
        "content": m.content,
        "created_at": m.created_at.isoformat(),
    }

```

Important (cause fréquente du “connecté / déconnecté”)

- L’URL WebSocket doit **correspondre exactement** au routing (avec **slash final**) :
ws://127.0.0.1:8000/ws/chat/general/
- Ouvrir **http://127.0.0.1:8000/ws/chat/general/** dans le navigateur donne souvent un **404 HTTP** normal (ce n’est pas une requête WebSocket).

Si vous me donnez aussi votre **monsite/urls.py** + le fichier **front (HTML/JS)** qui ouvre le WS, je peux corriger la partie qui provoque les déconnexions en boucle.

urls.py
Python
monsite/urls.py
Réflexion durant 25s

Voici une version **corrigée et “propre”** des fichiers que vous avez envoyés (compatible **Django 5.2 + Channels 4 + Daphne**).

✅ Objectif : avoir **HTTP OK + WebSocket OK** sur `ws://127.0.0.1:8000/ws/chat/<room>/`

monsite/asgi.py

```
import os

from django.core.asgi import get_asgi_application
from channels.routing import ProtocolTypeRouter, URLRouter
from channels.auth import AuthMiddlewareStack

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "monsite.settings")

django_asgi_app = get_asgi_application()

# Import AFTER Django init (évite ImproperlyConfigured/circular imports)
from chat.routing import websocket_urlpatterns

application = ProtocolTypeRouter(
    {
        "http": django_asgi_app,
        "websocket": AuthMiddlewareStack(
            URLRouter(websocket_urlpatterns)
        ),
    }
)
```

monsite/settings.py

```
from pathlib import Path
```

```
BASE_DIR = Path(__file__).resolve().parent.parent

SECRET_KEY = "dev-secret-key-change-me"
DEBUG = True
ALLOWED_HOSTS = ["127.0.0.1", "localhost"]

INSTALLED_APPS = [
    "daphne", # important pour runserver ASGI
    "django.contrib.admin",
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",

    "corsheaders",
    "rest_framework",
    "channels",

    "chat",
]

MIDDLEWARE = [
    "corsheaders.middleware.CorsMiddleware",
    "django.middleware.security.SecurityMiddleware",
    "django.contrib.sessions.middleware.SessionMiddleware",
    "django.middleware.common.CommonMiddleware",
    "django.middleware.csrf.CsrfViewMiddleware",
    "django.contrib.auth.middleware.AuthenticationMiddleware",
    "django.contrib.messages.middleware.MessageMiddleware",
    "django.middleware.clickjacking.XFrameOptionsMiddleware",
]

ROOT_URLCONF = "monsite.urls"

TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
        "DIRS": [],
        "APP_DIRS": True,
        "OPTIONS": {
            "context_processors": [
                "django.template.context_processors.request",
                "django.contrib.auth.context_processors.auth",
                "django.contrib.messages.context_processors.messages",
            ],
        },
    },
]
```

```

        },
    },
]

WSGI_APPLICATION = "monsite.wsgi.application"
ASGI_APPLICATION = "monsite.asgi.application"

DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.sqlite3",
        "NAME": BASE_DIR / "db.sqlite3",
    }
}

AUTH_PASSWORD_VALIDATORS = [
    {"NAME": "django.contrib.auth.password_validation.UserAttributeSimilarityValidator"},
    {"NAME": "django.contrib.auth.password_validation.MinimumLengthValidator"},
    {"NAME": "django.contrib.auth.password_validation.CommonPasswordValidator"},
    {"NAME": "django.contrib.auth.password_validation.NumericPasswordValidator"},
]

LANGUAGE_CODE = "fr-fr"
TIME_ZONE = "Indian/Antananarivo"
USE_I18N = True
USE_TZ = True

STATIC_URL = "static/"
DEFAULT_AUTO_FIELD = "django.db.models.BigAutoField"

CORS_ALLOW_ALL_ORIGINS = True

# DEV: InMemory layer (OK en local)
CHANNEL_LAYERS = {
    "default": {
        "BACKEND": "channels.layers.InMemoryChannelLayer",
    }
}

REST_FRAMEWORK = {
    "DEFAULT_RENDERER_CLASSES": ["rest_framework.renderers.JSONRenderer"],
}

```

chat/routing.py

```
from django.urls import re_path
from .consumers import ChatConsumer

websocket_urlpatterns = [
    re_path(r"^ws/chat/(?P<room_name>[/+]+)/$", ChatConsumer.as_asgi()),
]
```

chat/consumers.py

```
import json
from channels.generic.websocket import AsyncWebsocketConsumer
from channels.db import database_sync_to_async
from .models import Room, Message

class ChatConsumer(AsyncWebsocketConsumer):
    async def connect(self):
        self.room_name = self.scope["url_route"]["kwargs"]["room_name"]
        self.group_name = f"chat_{self.room_name}"

        await self.channel_layer.group_add(self.group_name, self.channel_name)
        await self.accept()

        await self.get_or_create_room(self.room_name)

        # (optionnel) petit message d'accueil
        await self.send(text_data=json.dumps({
            "system": True,
            "message": f"Connecté au salon \"{self.room_name}\""
        }))

    async def disconnect(self, close_code):
        await self.channel_layer.group_discard(self.group_name, self.channel_name)

    async def receive(self, text_data=None, bytes_data=None):
        if not text_data:
            return

        try:
            data = json.loads(text_data)
        except json.JSONDecodeError:
            return

        author = (data.get("author") or "anonymous").strip()[:80]
```

```

content = (data.get("content") or "").strip()
if not content:
    return

msg = await self.create_message(self.room_name, author, content)

await self.channel_layer.group_send(
    self.group_name,
    {
        "type": "chat_message",
        "id": msg["id"],
        "author": msg["author"],
        "content": msg["content"],
        "created_at": msg["created_at"],
    },
)

async def chat_message(self, event):
    # On renvoie au client uniquement les champs utiles
    payload = {
        "id": event.get("id"),
        "author": event.get("author"),
        "content": event.get("content"),
        "created_at": event.get("created_at"),
    }
    await self.send(text_data=json.dumps(payload))

@database_sync_to_async
def get_or_create_room(self, name: str):
    return Room.objects.get_or_create(name=name)

@database_sync_to_async
def create_message(self, room_name: str, author: str, content: str):
    room = Room.objects.get(name=room_name)
    m = Message.objects.create(room=room, author=author, content=content)
    return {
        "id": m.id,
        "author": m.author,
        "content": m.content,
        "created_at": m.created_at.isoformat(),
    }

```

```
from django.contrib import admin
from django.urls import path, include
from chat import views

urlpatterns = [
    path("", views.index),
    path("admin/", admin.site.urls),
    path("api/", include("chat.urls")),
]
```

Remarque importante (pour votre symptôme “connecté puis déconnecté”)

Le correctif principal ici est dans `consumers.py` :

- gestion `JSONDecodeError`
- ne pas renvoyer `"type": "chat_message"` au navigateur (certains front ferment/bug si format inattendu)
- envoyer un petit message “system” après `accept()` (utile pour debug)

Si vous voulez, envoyez aussi votre **fichier front (HTML/JS)** ou `chat/views.py` (juste le contenu), et je l’aligne exactement avec le format JSON renvoyé par le WebSocket.

views.py
Python
chat/views.py
Réflexion durant 1m 8s

Oui — voici une version “propre” et stable des fichiers (avec **routing en `path()`**, un **ASGI sûr** et un **consumer plus robuste** pour éviter les déconnexions en cas de souci DB).

1) `monsite/asgi.py`

```
import os

from channels.auth import AuthMiddlewareStack
from channels.routing import ProtocolTypeRouter, URLRouter
from django.core.asgi import get_asgi_application

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "monsite.settings")
```

```
# Initialise Django ASGI application early to ensure the AppRegistry is populated.
django_asgi_app = get_asgi_application()

# Import after Django setup (avoid "settings are not configured"/AppRegistry errors)
from chat.routing import websocket_urlpatterns # noqa: E402

application = ProtocolTypeRouter(
    {
        "http": django_asgi_app,
        "websocket": AuthMiddlewareStack(URLRouter(websocket_urlpatterns)),
    }
)
```

2) chat/routing.py

```
from django.urls import path

from .consumers import ChatConsumer

websocket_urlpatterns = [
    path("ws/chat/<str:room_name>/", ChatConsumer.as_asgi()),
]
```

3) chat/consumers.py

```
import json

from channels.db import database_sync_to_async
from channels.generic.websocket import AsyncWebsocketConsumer

from .models import Message, Room

class ChatConsumer(AsyncWebsocketConsumer):
    async def connect(self):
        self.room_name = self.scope["url_route"]["kwargs"]["room_name"]
        self.group_name = f"chat_{self.room_name}"

        # Ensure the room exists before accepting the socket.
        try:
            await self.get_or_create_room(self.room_name)
        except Exception:
            # If DB isn't ready, refuse the connection cleanly.
```

```

        await self.close(code=1011)
        return

    await self.channel_layer.group_add(self.group_name, self.channel_name)
    await self.accept()

    # Optional: confirm connection to client
    await self.send(
        text_data=json.dumps(
            {"type": "system", "message": f'Connecté au salon "{self.room_name}"'}
        )
    )

async def disconnect(self, close_code):
    group = getattr(self, "group_name", None)
    if group:
        await self.channel_layer.group_discard(group, self.channel_name)

async def receive(self, text_data):
    data = json.loads(text_data or "{}")
    author = (data.get("author") or "anonymous").strip()[:80]
    content = (data.get("content") or "").strip()

    if not content:
        return

    msg = await self.create_message(self.room_name, author, content)

    await self.channel_layer.group_send(
        self.group_name,
        {
            "type": "chat.message",
            "id": msg["id"],
            "author": msg["author"],
            "content": msg["content"],
            "created_at": msg["created_at"],
        },
    )

async def chat_message(self, event):
    payload = {
        "type": "message",
        "id": event.get("id"),
        "author": event.get("author"),
        "content": event.get("content"),
        "created_at": event.get("created_at"),
    }

```



```

    }
    await self.send(text_data=json.dumps(payload))

@database_sync_to_async
def get_or_create_room(self, name: str):
    return Room.objects.get_or_create(name=name)

@database_sync_to_async
def create_message(self, room_name: str, author: str, content: str):
    room, _ = Room.objects.get_or_create(name=room_name)
    m = Message.objects.create(room=room, author=author, content=content)
    return {
        "id": m.id,
        "author": m.author,
        "content": m.content,
        "created_at": m.created_at.isoformat(),
    }

```

4) monsite/settings.py

```

from pathlib import Path

BASE_DIR = Path(__file__).resolve().parent.parent

SECRET_KEY = "dev-secret-key-change-me"
DEBUG = True
ALLOWED_HOSTS = ["127.0.0.1", "localhost"]

INSTALLED_APPS = [
    # When Daphne is installed, Django uses it automatically for runserver.
    "daphne",

    "django.contrib.admin",
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",

    "corsheaders",
    "rest_framework",
    "channels",

    "chat",

```

```

]

MIDDLEWARE = [
    "corsheaders.middleware.CorsMiddleware",
    "django.middleware.security.SecurityMiddleware",
    "django.contrib.sessions.middleware.SessionMiddleware",
    "django.middleware.common.CommonMiddleware",
    "django.middleware.csrf.CsrfViewMiddleware",
    "django.contrib.auth.middleware.AuthenticationMiddleware",
    "django.contrib.messages.middleware.MessageMiddleware",
    "django.middleware.clickjacking.XFrameOptionsMiddleware",
]

ROOT_URLCONF = "monsite.urls"

TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
        "DIRS": [],
        "APP_DIRS": True,
        "OPTIONS": {
            "context_processors": [
                "django.template.context_processors.request",
                "django.contrib.auth.context_processors.auth",
                "django.contrib.messages.context_processors.messages",
            ],
        },
    ],
]

WSGI_APPLICATION = "monsite.wsgi.application"
ASGI_APPLICATION = "monsite.asgi.application"

DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.sqlite3",
        "NAME": BASE_DIR / "db.sqlite3",
    }
}

AUTH_PASSWORD_VALIDATORS = [
    {"NAME": "django.contrib.auth.password_validation.UserAttributeSimilarityValidator"},
    {"NAME": "django.contrib.auth.password_validation.MinimumLengthValidator"},
    {"NAME": "django.contrib.auth.password_validation.CommonPasswordValidator"},
    {"NAME": "django.contrib.auth.password_validation.NumericPasswordValidator"},
]

```

```

LANGUAGE_CODE = "fr-fr"
TIME_ZONE = "Indian/Antananarivo"
USE_I18N = True
USE_TZ = True

STATIC_URL = "static/"
DEFAULT_AUTO_FIELD = "django.db.models.BigAutoField"

# Dev CORS (frontend ouvert en fichier local / autre port)
CORS_ALLOW_ALL_ORIGINS = True

# Channels: in-memory for dev. If you run Redis, switch to channels_redis.
CHANNEL_LAYERS = {
    "default": {"BACKEND": "channels.layers.InMemoryChannelLayer"}
}

REST_FRAMEWORK = {
    "DEFAULT_RENDERER_CLASSES": ["rest_framework.renderers.JSONRenderer"],
}

```

5) monsite/urls.py

```

from django.contrib import admin
from django.urls import path, include

from chat.views import index

urlpatterns = [
    path("", index, name="index"),
    path("admin/", admin.site.urls),
    path("api/", include("chat.urls")),
]

```

6) chat/views.py

```

from django.shortcuts import render
from rest_framework import generics

from .models import Message, Room
from .serializers import MessageSerializer, RoomSerializer

```

```
def index(request):
    return render(request, "chat/index.html")

class RoomListCreateView(generics.ListCreateAPIView):
    queryset = Room.objects.all().order_by("name")
    serializer_class = RoomSerializer

class MessageListView(generics.ListAPIView):
    serializer_class = MessageSerializer

    def get_queryset(self):
        room_id = self.kwargs["room_id"]
        return Message.objects.filter(room_id=room_id).order_by("created_at")[:200]
```

✅ Fichiers prêts dans le dossier “corrected”

J’ai aussi écrit ces versions dans votre espace ici :

- `sandbox:/mnt/data/corrected/asgi.py`
- `sandbox:/mnt/data/corrected/settings.py`
- `sandbox:/mnt/data/corrected/urls.py`
- `sandbox:/mnt/data/corrected/routing.py`
- `sandbox:/mnt/data/corrected/consumers.py`
- `sandbox:/mnt/data/corrected/views.py`

Si vous voulez, collez aussi votre **template** `chat/index.html` (ou votre JS frontend), et je corrige la partie client qui provoque les “connecté/déconnecté” en boucle.

`index.html`

Fichier

Réflexion durant 1m 7s

Oui — le **problème principal** vient de `chat/consumers.py` : tu utilises l’ORM Django (`Room.objects...`, `Message.objects...`) **directement dans des méthodes async**, ce qui provoque souvent une exception en arrière-plan et une **déconnexion immédiate du WebSocket**.

Voici le **fichier corrigé** (`chat/consumers.py`) en utilisant `database_sync_to_async`.

```

import json

from channels.generic.websocket import AsyncWebsocketConsumer
from channels.db import database_sync_to_async

from .models import Room, Message

class ChatConsumer(AsyncWebsocketConsumer):
    async def connect(self):
        self.room_name = self.scope["url_route"]["kwargs"]["room_name"]
        self.room_group_name = f"chat_{self.room_name}"

        # ✅ ORM via database_sync_to_async
        self.room, _ = await self.get_or_create_room(self.room_name)

        await self.channel_layer.group_add(self.room_group_name, self.channel_name)
        await self.accept()

        # Optionnel: message système (utile pour debug côté client)
        await self.send(text_data=json.dumps({
            "type": "system",
            "message": f'Connecté au salon "{self.room_name}"'
        }))

    async def disconnect(self, close_code):
        await self.channel_layer.group_discard(self.room_group_name, self.channel_name)

    async def receive(self, text_data=None, bytes_data=None):
        if not text_data:
            return

        data = json.loads(text_data)
        message = (data.get("message") or "").strip()
        if not message:
            return

        # ✅ ORM via database_sync_to_async
        saved = await self.create_message(self.room, message)

        await self.channel_layer.group_send(
            self.room_group_name,
            {
                "type": "chat_message",
                "message": saved.content,
            }
        )

```

```

        },
    )

    async def chat_message(self, event):
        await self.send(text_data=json.dumps({
            "type": "chat",
            "message": event["message"],
        }))

# -----
# DB helpers (sync -> async)
# -----
@database_sync_to_async
def get_or_create_room(self, name: str):
    return Room.objects.get_or_create(name=name)

@database_sync_to_async
def create_message(self, room: Room, content: str):
    return Message.objects.create(room=room, content=content)

```

Pourquoi ça corrige ?

- Django ORM est **synchrone**.
- Dans Channels, ton consumer est **async**.
- Donc il faut passer par `database_sync_to_async` pour éviter les erreurs du type :
“You cannot call this from an async context” → ce qui entraîne souvent une **fermeture du WS**.

Si tu veux, je peux aussi te renvoyer une version améliorée qui **renvoie l’historique des messages** à la connexion (ex: les 50 derniers) — mais avec ce correctif, tu devrais déjà ne plus être déconnecté instantanément.

Pouvez vous mettre tous ces fichiers corrigés dans un fichier zip

Réflexion durant 39s

Voici le fichier zip contenant tous les fichiers corrigés (avec l’arborescence `monsite/...` et `chat/...`, et `index.html` placé dans `chat/templates/chat/index.html`) :

Télécharger le zip

venv\Lib\site-packages\redis\cluster.py: config option "self.retry" which defaults to 3 unless manually venv\Lib\site-packages\redis\cluster.py: configured.
 venv\Lib\site-packages\redis\cluster.py: cache_config: Optional[CacheConfig] = None, venv\Lib\site-packages\redis\cluster.py: self._cache_config =
 cache_config venv\Lib\site-packages\redis\cluster.py: # We are configuring the connection pool not to retry venv\Lib\site-packages\redis\cluster.py:

```

node_retry_config = Retry( venv\Lib\site-packages\redis\cluster.py: {"maint_notifications_config": MaintNotificationsConfig(enabled=False)} venv\Lib\site-
packages\redis\cluster.py: kwargs.update({"retry": node_retry_config}) venv\Lib\site-packages\redis\cluster.py: retry=node_retry_config, venv\Lib\site-
packages\redis\cluster.py: if self._cache is None and self._cache_config is not None: venv\Lib\site-packages\redis\cluster.py: self._cache =
CacheFactory(self._cache_config).get_cache() venv\Lib\site-packages\redis\cluster.py: reason="Please configure the 'retry' object instead", venv\Lib\site-
packages\redis\cluster.py: "CONFIG GET", venv\Lib\site-packages\redis\cluster.py: "CONFIG RESETSTAT", venv\Lib\site-packages\redis\cluster.py:
"CONFIG REWRITE", venv\Lib\site-packages\redis\cluster.py: "CONFIG SET", venv\Lib\site-packages\redis\cluster.py: "CONFIG", venv\Lib\site-
packages\redis\cluster.py: the retries in config option "self.retry" venv\Lib\site-packages\redis\cluster.py: which defaults to 3 unless manually configured.
venv\Lib\site-packages\redis\cluster.py: # rc.cluster_save_config(rc.get_primaries()) venv\Lib\site-packages\redis\commands\cluster.py: Get Cluster config for
the node. venv\Lib\site-packages\redis\commands\cluster.py: Reconfigure a node as a slave of the specified master node venv\Lib\site-
packages\redis\commands\cluster.py: def cluster_save_config( venv\Lib\site-packages\redis\commands\cluster.py: For more information see
https://redis.io/commands/cluster-saveconfig venv\Lib\site-packages\redis\commands\cluster.py: return self.execute_command("CLUSTER SAVECONFIG",
target_nodes=target_nodes) venv\Lib\site-packages\redis\commands\cluster.py: def cluster_set_config_epoch( venv\Lib\site-
packages\redis\commands\cluster.py: Set the configuration epoch in a new node venv\Lib\site-packages\redis\commands\cluster.py: For more information see
https://redis.io/commands/cluster-set-config-epoch venv\Lib\site-packages\redis\commands\cluster.py: "CLUSTER SET-CONFIG-EPOCH", epoch,
target_nodes=target_nodes venv\Lib\site-packages\redis\commands\core.py: Load ACL rules from the configured 'aclfile'. venv\Lib\site-
packages\redis\commands\core.py: Note that the server must be configured with the 'aclfile' venv\Lib\site-packages\redis\commands\core.py: Save ACL
rules to the configured 'aclfile'. venv\Lib\site-packages\redis\commands\core.py: Note that the server must be configured with the 'aclfile' venv\Lib\site-
packages\redis\commands\core.py: def config_get( venv\Lib\site-packages\redis\commands\core.py: Return a dictionary of configuration based on the
'pattern' venv\Lib\site-packages\redis\commands\core.py: For more information, see https://redis.io/commands/config-get venv\Lib\site-
packages\redis\commands\core.py: return self.execute_command("CONFIG GET", pattern, *args, **kwargs) venv\Lib\site-packages\redis\commands\core.py:
def config_set( venv\Lib\site-packages\redis\commands\core.py: ""Set config item 'name' with value venv\Lib\site-packages\redis\commands\core.py: For
more information, see https://redis.io/commands/config-set venv\Lib\site-packages\redis\commands\core.py: return self.execute_command("CONFIG SET",
name, value, *args, **kwargs) venv\Lib\site-packages\redis\commands\core.py: def config_resetstat(self, **kwargs) -> ResponseT: venv\Lib\site-
packages\redis\commands\core.py: For more information, see https://redis.io/commands/config-resetstat venv\Lib\site-packages\redis\commands\core.py:
return self.execute_command("CONFIG RESETSTAT", **kwargs) venv\Lib\site-packages\redis\commands\core.py: def config_rewrite(self, **kwargs) ->
ResponseT: venv\Lib\site-packages\redis\commands\core.py: Rewrite config file with the minimal change to reflect running config. venv\Lib\site-
packages\redis\commands\core.py: For more information, see https://redis.io/commands/config-rewrite venv\Lib\site-packages\redis\commands\core.py: return
self.execute_command("CONFIG REWRITE", **kwargs) venv\Lib\site-packages\redis\commands\core.py: ""Shutdown the Redis server. If Redis has
persistence configured, venv\Lib\site-packages\redis\commands\core.py: 'save' will force a DB saving operation even if no save points are
configured. venv\Lib\site-packages\redis\commands\core.py: are configured. venv\Lib\site-packages\redis\commands\core.py: ""Shutdown the Redis
server. If Redis has persistence configured, venv\Lib\site-packages\redis\commands\core.py: configured. If the "nosave" option is set, no data flush will be
venv\Lib\site-packages\redis\commands\core.py: Loads a module from a dynamic library at runtime with configuration directives. venv\Lib\site-
packages\redis\commands\core.py: pieces.append("CONFIG") venv\Lib\site-packages\redis\commands\search\commands.py: HybridPostProcessingConfig,
venv\Lib\site-packages\redis\commands\search\commands.py: CONFIG_CMD = "FT.CONFIG" venv\Lib\site-packages\redis\commands\search\commands.py:
def _parse_config_get(self, res, **kwargs): venv\Lib\site-packages\redis\commands\search\commands.py: post_processing:
Optional[HybridPostProcessingConfig] = None, venv\Lib\site-packages\redis\commands\search\commands.py: - **post_processing**:
HybridPostProcessingConfig object venv\Lib\site-packages\redis\commands\search\commands.py: Contains the post processing configuration venv\Lib\site-

```

packages\redis\commands\search\commands.py: - ****cursor****: HybridCursorQuery object - contains the cursor configuration
 packages\redis\commands\search\commands.py: reason="deprecated since Redis 8.0, call config_set from core module instead",
 packages\redis\commands\search\commands.py: def config_set(self, option: str, value: str) -> bool:
 packages\redis\commands\search\commands.py: **"""Set runtime configuration option.**
****option****: the name of the configuration option.
****value****: a value for the configuration option.
 packages\redis\commands\search\commands.py: For more information see `FT.CONFIG SET`
<https://redis.io/commands/ft.config-set>
 packages\redis\commands\search\commands.py: cmd = [CONFIG_CMD, "SET", option, value]
 packages\redis\commands\search\commands.py: reason="deprecated since Redis 8.0, call config_get from core module instead",
 packages\redis\commands\search\commands.py: def config_get(self, option: str) -> str:
 packages\redis\commands\search\commands.py: **"""Get runtime configuration option value.**
****option****: the name of the configuration option.
 packages\redis\commands\search\commands.py: For more information see `FT.CONFIG GET`
<https://redis.io/commands/ft.config-get>
 packages\redis\commands\search\commands.py: cmd = [CONFIG_CMD, "GET", option]
 packages\redis\commands\search\commands.py: return self._parse_results(CONFIG_CMD, res)
 packages\redis\commands\search\commands.py: post_processing: Optional[HybridPostProcessingConfig] = None,
 packages\redis\commands\search\commands.py: - ****post_processing****: HybridPostProcessingConfig object
 packages\redis\commands\search\commands.py: Contains the post processing configuration
 packages\redis\commands\search\commands.py: - ****cursor****: HybridCursorQuery object - contains the cursor configuration
 packages\redis\commands\search\commands.py: reason="deprecated since Redis 8.0, call config_set from core module instead",
 packages\redis\commands\search\commands.py: async def config_set(self, option: str, value: str) -> bool:
 packages\redis\commands\search\commands.py: **"""Set runtime configuration option.**
 packages\redis\commands\search\commands.py: - ****option****: the name of the configuration option.
 packages\redis\commands\search\commands.py: - ****value****: a value for the configuration option.
 packages\redis\commands\search\commands.py: For more information see `FT.CONFIG SET`
<https://redis.io/commands/ft.config-set>
 packages\redis\commands\search\commands.py: cmd = [CONFIG_CMD, "SET", option, value]
 packages\redis\commands\search\commands.py: reason="deprecated since Redis 8.0, call config_get from core module instead",
 packages\redis\commands\search\commands.py: async def config_get(self, option: str) -> str:
 packages\redis\commands\search\commands.py: **"""Get runtime configuration option value.**
 packages\redis\commands\search\commands.py: - ****option****: the name of the configuration option.
 packages\redis\commands\search\commands.py: For more information see `FT.CONFIG GET`
<https://redis.io/commands/ft.config-get>
 packages\redis\commands\search\commands.py: cmd = [CONFIG_CMD, "GET", option]
 packages\redis\commands\search\commands.py: return self._parse_results(CONFIG_CMD, res)
 packages\redis\commands\search\hybrid_query.py: class HybridPostProcessingConfig:
 packages\redis\commands\search\hybrid_query.py: Create a new hybrid post processing configuration object.
 packages\redis\commands\search__init__.py: CONFIG_CMD, packages\redis\commands\sentinel.py: configuration will be published so that the other Sentinels will
 packages\redis\commands\sentinel.py: update their configurations).
 packages\redis\commands\sentinel.py: Check if the current Sentinel configuration is able to reach the
 packages\redis\commands\sentinel.py: def sentinel_flushconfig(self):
 packages\redis\commands\sentinel.py: Force Sentinel to rewrite its configuration on disk, including the
 packages\redis\commands\sentinel.py: Normally Sentinel rewrites the configuration every time something
 packages\redis\commands\sentinel.py: However sometimes it is possible that the configuration file is lost
 packages\redis\commands\sentinel.py: configuration managers. In those cases

a way to force Sentinel to `venv\Lib\site-packages\redis\commands\sentinel.py`: rewrite the configuration file is handy. `venv\Lib\site-packages\redis\commands\sentinel.py`: This command works even if the previous configuration file is `venv\Lib\site-packages\redis\commands\sentinel.py`: `return self.execute_command("SENTINEL FLUSHCONFIG")` `venv\Lib\site-packages\redis\commands\timeseries\info.py`: https://redis.io/docs/latest/develop/data-types/timeseries/configuration/#duplicate_policy `venv\Lib\site-packages\redis\connection.py`: `MaintNotificationsConfig`, `venv\Lib\site-packages\redis\connection.py`: # the command name, e.g., 'CONFIG GET'. The Redis server expects these `venv\Lib\site-packages\redis\connection.py`: `maint_notifications_config`: Optional[`MaintNotificationsConfig`], `venv\Lib\site-packages\redis\connection.py`: `maint_notifications_config` (`MaintNotificationsConfig`): The configuration for maintenance notifications. `venv\Lib\site-packages\redis\connection.py`: `self.maint_notifications_config = maint_notifications_config` `venv\Lib\site-packages\redis\connection.py`: `self._configure_maintenance_notifications`(`venv\Lib\site-packages\redis\connection.py`: `def _configure_maintenance_notifications`(`venv\Lib\site-packages\redis\connection.py`: not `self.maint_notifications_config` `venv\Lib\site-packages\redis\connection.py`: or not `self.maint_notifications_config.enabled` `venv\Lib\site-packages\redis\connection.py`: "To configure maintenance notifications, a parser must be provided!" `venv\Lib\site-packages\redis\connection.py`: `MaintNotificationsConnectionHandler`(`self`, `self.maint_notifications_config`) `venv\Lib\site-packages\redis\connection.py`: `self`, `maint_notifications_pool_handler.config` `venv\Lib\site-packages\redis\connection.py`: `self._maint_notifications_connection_handler.config = (` `venv\Lib\site-packages\redis\connection.py`: `maint_notifications_pool_handler.config` `venv\Lib\site-packages\redis\connection.py`: # When the `maint_notifications_config` enabled mode is "auto", `venv\Lib\site-packages\redis\connection.py`: and `self.maint_notifications_config` `venv\Lib\site-packages\redis\connection.py`: and `self.maint_notifications_config.enabled` `venv\Lib\site-packages\redis\connection.py`: `maint_notifications_config=``self.maint_notifications_config`, `venv\Lib\site-packages\redis\connection.py`: `self`, `maint_notifications_config`: `MaintNotificationsConfig`, `check_health=True` `venv\Lib\site-packages\redis\connection.py`: `endpoint_type = maint_notifications_config.get_endpoint_type(host, self)` `venv\Lib\site-packages\redis\connection.py`: and `maint_notifications_config.enabled == "auto"` `venv\Lib\site-packages\redis\connection.py`: `maint_notifications_config`: Optional[`MaintNotificationsConfig`] = `None`, `venv\Lib\site-packages\redis\connection.py`: `maint_notifications_config`, `venv\Lib\site-packages\redis\connection.py`: # if enabled in the configuration `venv\Lib\site-packages\redis\connection.py`: `self._conn.maint_notifications_config`, `venv\Lib\site-packages\redis\connection.py`: `maint_notifications_config`: Optional[`MaintNotificationsConfig`] = `None`, `venv\Lib\site-packages\redis\connection.py`: if `maint_notifications_config` is `None` and `is_protocol_supported`: `venv\Lib\site-packages\redis\connection.py`: `maint_notifications_config = MaintNotificationsConfig()` `venv\Lib\site-packages\redis\connection.py`: if `maint_notifications_config` and `maint_notifications_config.enabled`: `venv\Lib\site-packages\redis\connection.py`: `self`, `maint_notifications_config` `venv\Lib\site-packages\redis\connection.py`: The maintenance notifications config is stored in the pool handler. `venv\Lib\site-packages\redis\connection.py`: `maint_notifications_config = (` `venv\Lib\site-packages\redis\connection.py`: `self._maint_notifications_pool_handler.config` `venv\Lib\site-packages\redis\connection.py`: `return maint_notifications_config` and `maint_notifications_config.enabled` `venv\Lib\site-packages\redis\connection.py`: `def update_maint_notifications_config`(`venv\Lib\site-packages\redis\connection.py`: `self`, `maint_notifications_config`: `MaintNotificationsConfig` `venv\Lib\site-packages\redis\connection.py`: Updates the maintenance notifications configuration. `venv\Lib\site-packages\redis\connection.py`: and not `maint_notifications_config.enabled` `venv\Lib\site-packages\redis\connection.py`: `self`, `maint_notifications_config` `venv\Lib\site-packages\redis\connection.py`: `self._maint_notifications_pool_handler.config = maint_notifications_config` `venv\Lib\site-packages\redis\connection.py`: `self._update_maint_notifications_configs_for_connections`(`venv\Lib\site-packages\redis\connection.py`: "maint_notifications_config": `maint_notifications_pool_handler.config`, `venv\Lib\site-packages\redis\connection.py`: # configured the original values yet `venv\Lib\site-packages\redis\connection.py`: `def _update_maint_notifications_configs_for_connections`(`venv\Lib\site-packages\redis\connection.py`: """Update the maintenance notifications config for all connections in the pool.""" `venv\Lib\site-packages\redis\connection.py`: `conn.maint_notifications_config = (` `venv\Lib\site-packages\redis\connection.py`: `maint_notifications_pool_handler.config` `venv\Lib\site-packages\redis\connection.py`: `conn.maint_notifications_config = (` `venv\Lib\site-packages\redis\connection.py`: `maint_notifications_pool_handler.config` `venv\Lib\site-`

packages\redis\connection.py: "connected_address", "configured_address", "notification_hash" venv\Lib\site-packages\redis\connection.py: elif
matching_pattern == "configured_address": venv\Lib\site-packages\redis\connection.py: "connected_address", "configured_address", "notification_hash"
venv\Lib\site-packages\redis\connection.py: If `maint_notifications_config` is provided, the connection pool will support venv\Lib\site-
packages\redis\connection.py: If the `maint_notifications_config` is not provided but the protocol is 3, venv\Lib\site-
packages\redis\connection.py: Return a connection pool configured from the given URL. venv\Lib\site-packages\redis\connection.py:
maint_notifications_config: Optional[MaintNotificationsConfig] = None, venv\Lib\site-packages\redis\connection.py: if
connection_kwargs.get("cache_config") or connection_kwargs.get("cache"): venv\Lib\site-packages\redis\connection.py:
self._connection_kwargs.get("cache_config") venv\Lib\site-packages\redis\connection.py: connection_kwargs.pop("cache_config", None) venv\Lib\site-
packages\redis\connection.py: maint_notifications_config=maint_notifications_config, venv\Lib\site-packages\redis\http\http_client.py: retry: Retry
configuration for failed requests. venv\Lib\site-packages\redis\http\http_client.py: # Load client certificate for mTLS if configured venv\Lib\site-
packages\redis\maint_notifications.py:class MaintNotificationsConfig: venv\Lib\site-packages\redis\maint_notifications.py: Configuration class for
maintenance notifications handling behaviour. Notifications are received through venv\Lib\site-packages\redis\maint_notifications.py: Initialize a new
MaintNotificationsConfig. venv\Lib\site-packages\redis\maint_notifications.py: If None, the endpoint type will be automatically determined based on the host
and TLS configuration. venv\Lib\site-packages\redis\maint_notifications.py: config: MaintNotificationsConfig, venv\Lib\site-
packages\redis\maint_notifications.py: self.config = config venv\Lib\site-packages\redis\maint_notifications.py: copy =
MaintNotificationsPoolHandler(self.pool, self.config) venv\Lib\site-packages\redis\maint_notifications.py: not self.config.proactive_reconnect venv\Lib\site-
packages\redis\maint_notifications.py: and not self.config.is_relaxed_timeouts_enabled() venv\Lib\site-packages\redis\maint_notifications.py:
self.config.proactive_reconnect venv\Lib\site-packages\redis\maint_notifications.py: or self.config.is_relaxed_timeouts_enabled() venv\Lib\site-
packages\redis\maint_notifications.py: relaxed_timeout=self.config.relaxed_timeout, venv\Lib\site-packages\redis\maint_notifications.py: if
self.config.proactive_reconnect: venv\Lib\site-packages\redis\maint_notifications.py: # Update config for new connections: venv\Lib\site-
packages\redis\maint_notifications.py: if self.config.is_relaxed_timeouts_enabled(): venv\Lib\site-packages\redis\maint_notifications.py: "socket_timeout":
self.config.relaxed_timeout, venv\Lib\site-packages\redis\maint_notifications.py: "socket_connect_timeout": self.config.relaxed_timeout, venv\Lib\site-
packages\redis\maint_notifications.py: reset_relaxed_timeout = self.config.is_relaxed_timeouts_enabled() venv\Lib\site-packages\redis\maint_notifications.py:
reset_host_address = self.config.proactive_reconnect venv\Lib\site-packages\redis\maint_notifications.py: config: MaintNotificationsConfig, venv\Lib\site-
packages\redis\maint_notifications.py: self.config = config venv\Lib\site-packages\redis\maint_notifications.py: or not
self.config.is_relaxed_timeouts_enabled() venv\Lib\site-packages\redis\maint_notifications.py: tmp_relaxed_timeout=self.config.relaxed_timeout
venv\Lib\site-packages\redis\maint_notifications.py: self.connection.update_current_socket_timeout(self.config.relaxed_timeout) venv\Lib\site-
packages\redis\maint_notifications.py: or not self.config.is_relaxed_timeouts_enabled() venv\Lib\site-packages\redis\multidb\client.py: from
redis.multidb.config import DEFAULT_GRACE_PERIOD, MultiDbConfig venv\Lib\site-packages\redis\multidb\client.py: def __init__(self, config:
MultiDbConfig): venv\Lib\site-packages\redis\multidb\client.py: self._databases = config.databases() venv\Lib\site-packages\redis\multidb\client.py:
config.default_health_checks() venv\Lib\site-packages\redis\multidb\client.py: if not config.health_checks venv\Lib\site-packages\redis\multidb\client.py: else
config.health_checks venv\Lib\site-packages\redis\multidb\client.py: self._health_check_interval = config.health_check_interval venv\Lib\site-
packages\redis\multidb\client.py: self._health_check_policy: HealthCheckPolicy = config.health_check_policy.value(venv\Lib\site-
packages\redis\multidb\client.py: config.health_check_probes, config.health_check_probes_delay venv\Lib\site-packages\redis\multidb\client.py:
config.default_failure_detectors() venv\Lib\site-packages\redis\multidb\client.py: if not config.failure_detectors venv\Lib\site-packages\redis\multidb\client.py:
else config.failure_detectors venv\Lib\site-packages\redis\multidb\client.py: config.default_failover_strategy() venv\Lib\site-packages\redis\multidb\client.py:
if config.failover_strategy is None venv\Lib\site-packages\redis\multidb\client.py: else config.failover_strategy venv\Lib\site-packages\redis\multidb\client.py:

```

self._auto_fallback_interval = config.auto_fallback_interval
venv\Lib\site-packages\redis\multidb\client.py: self._event_dispatcher = config.event_dispatcher
venv\Lib\site-packages\redis\multidb\client.py: self._command_retry = config.command_retry
venv\Lib\site-packages\redis\multidb\client.py: failover_attempts=config.failover_attempts, venv\Lib\site-packages\redis\multidb\client.py: failover_delay=config.failover_delay, venv\Lib\site-
packages\redis\multidb\client.py: self._config = config
venv\Lib\site-packages\redis\multidb\command_executor.py: from redis.multidb.config import
DEFAULT_AUTO_FALLBACK_INTERVAL
venv\Lib\site-packages\redis\multidb\config.py: class DatabaseConfig:
venv\Lib\site-
packages\redis\multidb\config.py: Dataclass representing the configuration for a database connection.
venv\Lib\site-packages\redis\multidb\config.py: This
class is used to store configuration settings for a database connection,
venv\Lib\site-packages\redis\multidb\config.py: attributes and allows for the creation of
customized configurations for various
venv\Lib\site-packages\redis\multidb\config.py: from_pool (Optional[ConnectionPool]): A pre-configured connection
pool to use.
venv\Lib\site-packages\redis\multidb\config.py: class MultiDbConfig:
venv\Lib\site-packages\redis\multidb\config.py: Configuration class for
managing multiple database connections in a resilient and fail-safe manner.
venv\Lib\site-packages\redis\multidb\config.py: databases_config: A list of
database configurations.
venv\Lib\site-packages\redis\multidb\config.py: Retrieves a collection of database clients managed by weighted configurations.
venv\Lib\site-packages\redis\multidb\config.py: Initializes database clients based on the provided configuration and removes
venv\Lib\site-
packages\redis\multidb\config.py: with defined retry and backoff configurations.
venv\Lib\site-packages\redis\multidb\config.py: databases_config:
List[DatabaseConfig]
venv\Lib\site-packages\redis\multidb\config.py: for database_config in self.databases_config:
venv\Lib\site-
packages\redis\multidb\config.py: database_config.client_kwargs.update(
venv\Lib\site-packages\redis\multidb\config.py: if database_config.from_url:
venv\Lib\site-packages\redis\multidb\config.py: database_config.from_url, **database_config.client_kwargs
venv\Lib\site-packages\redis\multidb\config.py: elif database_config.from_pool:
venv\Lib\site-packages\redis\multidb\config.py: database_config.from_pool.set_retry(
venv\Lib\site-
packages\redis\multidb\config.py: connection_pool=database_config.from_pool
venv\Lib\site-packages\redis\multidb\config.py: client =
self.client_class(**database_config.client_kwargs)
venv\Lib\site-packages\redis\multidb\config.py: database_config.default_circuit_breaker()
venv\Lib\site-
packages\redis\multidb\config.py: if database_config.circuit is None
venv\Lib\site-packages\redis\multidb\config.py: else database_config.circuit
venv\Lib\site-
packages\redis\multidb\config.py: weight=database_config.weight,
venv\Lib\site-packages\redis\multidb\config.py: health_check_url=database_config.health_check_url,
venv\Lib\site-packages\redis\multidb\config.py: database_config.weight,
venv\Lib\site-
packages\redis\multidb\healthcheck.py: "Database health check url is not set. Please check DatabaseConfig for the current database."
venv\Lib\site-
packages\redis\_parsers\base.py: # user send an AUTH cmd to a server without authorization configured
venv\Lib\site-packages\redis\_parsers\base.py: "configured for the default user. Are you sure "
venv\Lib\site-packages\redis\_parsers\base.py: "your configuration is correct?": AuthenticationError,
venv\Lib\site-packages\redis\_parsers\helpers.py: "config-epoch": int,
venv\Lib\site-packages\redis\_parsers\helpers.py: def parse_config_get(response,
**options):
venv\Lib\site-packages\redis\_parsers\helpers.py: "CLUSTER SAVECONFIG": bool_ok,
venv\Lib\site-packages\redis\_parsers\helpers.py: "CLUSTER SET-CONFIG-EPOCH": bool_ok,
venv\Lib\site-packages\redis\_parsers\helpers.py: "CONFIG RESETSTAT": bool_ok,
venv\Lib\site-
packages\redis\_parsers\helpers.py: "CONFIG SET": bool_ok,
venv\Lib\site-packages\redis\_parsers\helpers.py: "SENTINEL FLUSHCONFIG": bool_ok,
venv\Lib\site-packages\redis\_parsers\helpers.py: "CONFIG GET": parse_config_get,
venv\Lib\site-packages\redis\_parsers\helpers.py: "CONFIG GET":
lambda r: {
venv\Lib\site-packages\rest_framework\apps.py: from django.apps import AppConfig
venv\Lib\site-packages\rest_framework\apps.py: class
RestFrameworkConfig(AppConfig):
venv\Lib\site-packages\rest_framework\authtoken\apps.py: from django.apps import AppConfig
venv\Lib\site-
packages\rest_framework\authtoken\apps.py: class AuthTokenConfig(AppConfig):
venv\Lib\site-packages\rest_framework\checks.py: def
pagination_system_check(app_configs, **kwargs):
venv\Lib\site-packages\rest_framework\compat.py: extension_configs = {
venv\Lib\site-
packages\rest_framework\compat.py: extensions=extensions, extension_configs=extension_configs
venv\Lib\site-packages\rest_framework\fields.py: Quantize
the decimal value to the configured precision.
venv\Lib\site-packages\rest_framework\filters.py: from django.core.exceptions import FieldDoesNotExist,
ImproperlyConfigured
venv\Lib\site-packages\rest_framework\filters.py: raise ImproperlyConfigured(msg % self.__class__.__name__)
venv\Lib\site-

```

```

packages\rest_framework\generics.py: if you want to apply the configured filtering backend to the venv\Lib\site-
packages\rest_framework\management\commands\generateschema.py: help = "Generates configured API schema for project." venv\Lib\site-
packages\rest_framework\pagination.py: page object, or None if pagination is not configured for this view. venv\Lib\site-
packages\rest_framework\relations.py: from django.core.exceptions import ImproperlyConfigured, ObjectDoesNotExist venv\Lib\site-
packages\rest_framework\relations.py: attributes are not configured to correctly match the URL conf. venv\Lib\site-packages\rest_framework\relations.py: #
Return the hyperlink, or error if incorrectly configured. venv\Lib\site-packages\rest_framework\relations.py: 'model in your API, or incorrectly configured the '
venv\Lib\site-packages\rest_framework\relations.py: raise ImproperlyConfigured(msg % self.view_name) venv\Lib\site-
packages\rest_framework\renderers.py: from django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\renderers.py: raise
ImproperlyConfigured( venv\Lib\site-packages\rest_framework\routers.py: from django.core.exceptions import ImproperlyConfigured venv\Lib\site-
packages\rest_framework\routers.py: raise ImproperlyConfigured(msg) venv\Lib\site-packages\rest_framework\routers.py: raise ImproperlyConfigured(msg %
', '.join(not_allowed)) venv\Lib\site-packages\rest_framework\serializers.py: from django.core.exceptions import FieldDoesNotExist, ImproperlyConfigured
venv\Lib\site-packages\rest_framework\serializers.py: raise ImproperlyConfigured( venv\Lib\site-packages\rest_framework\settings.py: # View configuration
venv\Lib\site-packages\rest_framework\test.py: from django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\test.py:
raise ImproperlyConfigured('requests must be installed in order to use RequestsClient.') venv\Lib\site-packages\rest_framework\test.py: raise
ImproperlyConfigured('coreapi must be installed in order to use CoreAPIClient.') venv\Lib\site-packages\rest_framework\throttling.py: from
django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\throttling.py: raise ImproperlyConfigured(msg) venv\Lib\site-
packages\rest_framework\throttling.py: raise ImproperlyConfigured(msg) venv\Lib\site-packages\rest_framework\views.py: # Filter out None values which
may happen in case of config / rate venv\Lib\site-packages\sqlparse\lexer.py: """The Lexer supports configurable syntax. venv\Lib\site-
packages\sqlparse\lexer.py: """Clear all syntax configurations. venv\Lib\site-packages\sqlparse\lexer.py: Useful if you want to load a reduced set of syntax
configurations. venv\Lib\site-packages\tutorial\asgi.py: ASGI config for tutorial project. venv\Lib\site-packages\tutorial\quickstart\apps.py: class
QuickstartConfig(AppConfig): venv\Lib\site-packages\tutorial\wsgi.py: WSGI config for tutorial project. (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> python manage.py makemigrations Migrations for 'chat': chat\migrations\0001_initial.py + Create model Room + Create
model Message (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py migrate Operations to perform: Apply all
migrations: admin, auth, chat, contenttypes, sessions Running migrations: Applying contenttypes.0001_initial... OK Applying auth.0001_initial... OK Applying
admin.0001_initial... OK Applying admin.0002_logentry_remove_auto_add... OK Applying admin.0003_logentry_add_action_flag_choices... OK Applying
contenttypes.0002_remove_content_type_name... OK Applying auth.0002_alter_permission_name_max_length... OK Applying
auth.0003_alter_user_email_max_length... OK Applying auth.0004_alter_user_username_opts... OK Applying auth.0005_alter_user_last_login_null... OK
Applying auth.0006_require_contenttypes_0002... OK Applying auth.0008_alter_user_username_max_length... OK Applying
auth.0010_alter_group_name_max_length... OK Applying auth.0011_update_proxy_permissions... OK Applying
auth.0012_alter_user_first_name_max_length... OK Applying chat.0001_initial... OK Applying sessions.0001_initial... OK (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> python -c "from django.core.asgi import get_asgi_application; print('django ok')" django ok (venv) PS D:\PYTHON
DJANGO HOPES\Chat_API_DJANGO_mario> python -c "import chat.routing; print('routing ok')" Traceback (most recent call last): File "<string>", line 1, in
<module> File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\routing.py", line 2, in <module> from .consumers import ChatConsumer
File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\consumers.py", line 4, in <module> from .models import Room, Message File
"D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\models.py", line 3, in <module> class Room(models.Model): File "D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\db\models\base.py", line 131, in __new__ app_config =
apps.get_containing_app_config(module) ~~~~~~ File "D:\PYTHON DJANGO

```

HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\apps\registry.py", line 260, in get_containing_app_config self.check_apps_ready() File "D:\PYTHON_DJANGO_HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\apps\registry.py", line 137, in check_apps_ready File "D:\PYTHON_DJANGO_HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\conf__init__.py", line 81, in __getattr__ self.setup(name) File "D:\PYTHON_DJANGO_HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\conf__init__.py", line 61, in _setup raise ImproperlyConfigured(django.core.exceptions.ImproperlyConfigured: Requested setting INSTALLED_APPS, but settings are not configured. You must either define the environment variable DJANGO_SETTINGS_MODULE or call settings.configure() before accessing settings. (venv) PS D:\PYTHON_DJANGO_HOPES\Chat_API_DJANGO_mario> python -c "import os; os.environ.setdefault('DJANGO_SETTINGS_MODULE','monsite.settings'); import django; django.setup(); import chat.routing; print('routing ok')" routing ok (venv) PS D:\PYTHON_DJANGO_HOPES\Chat_API_DJANGO_mario> python manage.py makemigrations No changes detected (venv) PS D:\PYTHON_DJANGO_HOPES\Chat_API_DJANGO_mario> python manage.py migrate Operations to perform: Apply all migrations: admin, auth, chat, contenttypes, sessions Running migrations: No migrations to apply. (venv) PS D:\PYTHON_DJANGO_HOPES\Chat_API_DJANGO_mario> py manage.py runserver Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 16:54:06 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: https://docs.djangoproject.com/en/5.2/howto/deployment/ Not Found: / [07/Feb/2026 16:54:39] "GET / HTTP/1.1" 404 2305 (venv) PS D:\PYTHON_DJANGO_HOPES\Chat_API_DJANGO_mario> python -m pip show channels >> ^C (venv) PS D:\PYTHON_DJANGO_HOPES\Chat_API_DJANGO_mario> python -m pip show channels Name: channels Version: 4.1.0 Author: Django Software Foundation License: BSD Location: D:\PYTHON_DJANGO_HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages Requires: asgiref, Django Required-by: channels-redis (venv) PS D:\PYTHON_DJANGO_HOPES\Chat_API_DJANGO_mario> python manage.py makemigrations (venv) PS D:\PYTHON_DJANGO_HOPES\Chat_API_DJANGO_mario> python manage.py migrate Operations to perform: Apply all migrations: admin, auth, chat, contenttypes, sessions Running migrations: (venv) PS D:\PYTHON_DJANGO_HOPES\Chat_API_DJANGO_mario> python manage.py makemigrations (venv) PS D:\PYTHON_DJANGO_HOPES\Chat_API_DJANGO_mario> python manage.py migrate Operations to perform: Apply all migrations: admin, auth, chat, contenttypes, sessions Running migrations: (venv) PS D:\PYTHON_DJANGO_HOPES\Chat_API_DJANGO_mario> python manage.py makemigrations No changes detected (venv) PS D:\PYTHON_DJANGO_HOPES\Chat_API_DJANGO_mario> python manage.py migrate Operations to perform: Apply all migrations: admin, auth, chat, contenttypes, sessions Running migrations: No migrations to apply. (venv) PS D:\PYTHON_DJANGO_HOPES\Chat_API_DJANGO_mario> py manage.py runserver Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 17:03:21 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: https://docs.djangoproject.com/en/5.2/howto/deployment/ Not Found: / [07/Feb/2026 17:03:24] "GET / HTTP/1.1" 404 2305 Not Found: / [07/Feb/2026 17:03:25] "GET / HTTP/1.1" 404 2305 Not Found: / [07/Feb/2026 17:03:27] "GET / HTTP/1.1" 404 2305 [07/Feb/2026 17:04:21] "GET /admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:04:22] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 4252 [07/Feb/2026 17:04:22] "GET /static/admin/css/nav_sidebar.css HTTP/1.1" 200 2810 [07/Feb/2026 17:04:22] "GET /static/admin/css/base.css HTTP/1.1" 200 22120 [07/Feb/2026 17:04:22] "GET /static/admin/css/login.css HTTP/1.1" 200 951 [07/Feb/2026 17:04:22] "GET /static/admin/css/dark_mode.css HTTP/1.1" 200 2808 [07/Feb/2026 17:04:22] "GET /static/admin/css/responsive.css HTTP/1.1" 200 16565 [07/Feb/2026 17:04:22] "GET /static/admin/js/theme.js HTTP/1.1" 200 1653 [07/Feb/2026 17:04:22] "GET /static/admin/js/nav_sidebar.js HTTP/1.1" 200 3063 Not Found: /favicon.ico [07/Feb/2026 17:04:22] "GET /favicon.ico HTTP/1.1" 404 2356 Not Found: /api/ [07/Feb/2026 17:04:37] "GET /api/ HTTP/1.1" 404 2739 Not Found: / [07/Feb/2026 17:04:48] "GET / HTTP/1.1" 404 2305 [07/Feb/2026 17:05:01] "GET /admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:05:01] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 4252 Not Found: /api/ [07/Feb/2026 17:05:12] "GET /api/ HTTP/1.1" 404 2739 Not Found: / [07/Feb/2026 17:24:49] "GET /api/rooms/ HTTP/1.1" 200 2 Not Found:

```
/favicon.ico [07/Feb/2026 17:24:49] "GET /favicon.ico HTTP/1.1" 404 2356 (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> curl -X
POST http://127.0.0.1:8000/api/rooms/ -H "Content-Type: application/json" -d '{"name":"general"}' >> ^C (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> curl -X POST http://127.0.0.1:8000/api/rooms/ -H "Content-Type: application/json" -d '{"name":"general"}' Invoke-
WebRequest : Impossible de lier le paramètre «Headers». Impossible de convertir la valeur «Content-Type: application/json» du type «System.String» en type
«System.Collections.IDictionary». Au caractère Ligne:1 : 50 + ~~~~~ + CategoryInfo : InvalidArgument : (:) [Invoke-
WebRequest], ParameterBindingException + FullyQualifiedErrorId :
CannotConvertArgumentNoMessage,Microsoft.PowerShell.Commands.InvokeWebRequestCommand (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/" >> -Method Post >> -ContentType "application/json"
Invoke-RestMethod : Impossible de se connecter au serveur distant Au caractère Ligne:1 : 1 + Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/" +
~~~~~ + CategoryInfo : NotSpecified : (:) [Invoke-RestMethod], WebException +
FullyQualifiedErrorId : System.Net.WebException,Microsoft.PowerShell.Commands.InvokeRestMethodCommand (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> py manage.py runserver Watching for file
changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 17:30:20 Django version 5.2.8,
using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development
server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see:
https://docs.djangoproject.com/en/5.2/howto/deployment/ Not Found: / [07/Feb/2026 17:30:24] "GET / HTTP/1.1" 404 2305 Not Found: / [07/Feb/2026
17:30:25] "GET / HTTP/1.1" 404 2305 Not Found: /ws/chat/general/ [07/Feb/2026 17:31:01] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found:
/ws/chat/general/ [07/Feb/2026 17:31:04] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:31:24] "GET /api/rooms/ HTTP/1.1" 200 2 Not
Found: /ws/chat/general/ [07/Feb/2026 17:31:32] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:31:58] "GET
/ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:32:00] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found:
/ws/chat/general/ [07/Feb/2026 17:32:02] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:32:04] "GET
/ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:21] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found:
/ws/chat/general/ [07/Feb/2026 17:35:22] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:22] "GET
/ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:32] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found:
/ws/chat/general/ [07/Feb/2026 17:35:33] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:35] "GET
/ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /api/ \u2022 WS: ws://127.0.0.1:8000/ws/chat/<room>/ [07/Feb/2026 17:35:44] "GET
/api/%20%E2%80%A2%20WS:%20ws://127.0.0.1:8000/ws/chat/%3Croom%3E/ HTTP/1.1" 404 2899 Not Found: /api/ [07/Feb/2026 17:35:51] "GET /api/
HTTP/1.1" 404 2739 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:43] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/
[07/Feb/2026 17:36:44] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:44] "GET /ws/chat/general/ HTTP/1.1"
404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:45] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026
17:36:46] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:47] "GET /ws/chat/general/ HTTP/1.1" 404 2371
[07/Feb/2026 17:38:30] "GET /api/rooms/ HTTP/1.1" 200 2 Not Found: /favicon.ico [07/Feb/2026 17:38:30] "GET /favicon.ico HTTP/1.1" 404 2356 Not
Found: / [07/Feb/2026 17:38:52] "GET / HTTP/1.1" 404 2305 [07/Feb/2026 17:41:00] "GET /admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:41:00] "GET
/admin/login/?next=/admin/ HTTP/1.1" 200 4252 [07/Feb/2026 17:41:01] "GET /static/admin/css/dark_mode.css HTTP/1.1" 200 2808 [07/Feb/2026 17:41:01]
"GET /static/admin/css/base.css HTTP/1.1" 200 22120 [07/Feb/2026 17:41:01] "GET /static/admin/css/nav_sidebar.css HTTP/1.1" 200 2810 [07/Feb/2026
17:41:01] "GET /static/admin/js/theme.js HTTP/1.1" 200 1653 [07/Feb/2026 17:41:01] "GET /static/admin/js/nav_sidebar.js HTTP/1.1" 200 3063
[07/Feb/2026 17:41:01] "GET /static/admin/css/login.css HTTP/1.1" 200 951 [07/Feb/2026 17:41:01] "GET /static/admin/css/responsive.css HTTP/1.1" 200
```

16565 [07/Feb/2026 17:41:35] "POST /admin/login/?next=/admin/ HTTP/1.1" 200 4498 [07/Feb/2026 17:44:04] "POST /admin/login/?next=/admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:44:04] "GET /admin/ HTTP/1.1" 200 7885 [07/Feb/2026 17:44:05] "GET /static/admin/css/dashboard.css HTTP/1.1" 200 441 [07/Feb/2026 17:44:05] "GET /static/admin/img/icon-changelink.svg HTTP/1.1" 200 380 [07/Feb/2026 17:44:05] "GET /static/admin/img/icon-addlink.svg HTTP/1.1" 200 331 [07/Feb/2026 17:44:27] "GET /admin/chat/room/add/ HTTP/1.1" 200 9637 [07/Feb/2026 17:44:27] "GET /static/admin/js/jquery.init.js HTTP/1.1" 200 347 [07/Feb/2026 17:44:27] "GET /static/admin/css/forms.css HTTP/1.1" 200 8525 [07/Feb/2026 17:44:27] "GET /static/admin/js/core.js HTTP/1.1" 200 6208 [07/Feb/2026 17:44:27] "GET /static/admin/js/urlify.js HTTP/1.1" 200 7887 [07/Feb/2026 17:44:27] "GET /static/admin/js/prepopulate.js HTTP/1.1" 200 1531 [07/Feb/2026 17:44:27] "GET /admin/jsi18n/ HTTP/1.1" 200 8634 [07/Feb/2026 17:44:27] "GET /static/admin/js/change_form.js HTTP/1.1" 200 606 [07/Feb/2026 17:44:27] "GET /static/admin/js/admin/RelatedObjectLookups.js HTTP/1.1" 200 9777 [07/Feb/2026 17:44:27] "GET /static/admin/js/actions.js HTTP/1.1" 200 8076 [07/Feb/2026 17:44:27] "GET /static/admin/js/prepopulate_init.js HTTP/1.1" 200 586 [07/Feb/2026 17:44:27] "GET /static/admin/css/widgets.css HTTP/1.1" 200 11973 [07/Feb/2026 17:44:28] "GET /static/admin/js/vendor/xregexp/xregexp.js HTTP/1.1" 200 325171 [07/Feb/2026 17:44:28] "GET /static/admin/js/vendor/jquery/jquery.js HTTP/1.1" 200 285314 [07/Feb/2026 17:44:44] "POST /admin/chat/room/add/ HTTP/1.1" 302 0 [07/Feb/2026 17:44:44] "GET /admin/chat/room/ HTTP/1.1" 200 10874 [07/Feb/2026 17:44:45] "GET /admin/jsi18n/ HTTP/1.1" 200 8634 [07/Feb/2026 17:44:45] "GET /static/admin/css/changelists.css HTTP/1.1" 200 6878 [07/Feb/2026 17:44:45] "GET /static/admin/js/filters.js HTTP/1.1" 200 978 [07/Feb/2026 17:44:45] "GET /static/admin/img/search.svg HTTP/1.1" 200 458 [07/Feb/2026 17:44:45] "GET /static/admin/img/tooltag-add.svg HTTP/1.1" 200 331 [07/Feb/2026 17:44:45] "GET /static/admin/img/icon-yes.svg HTTP/1.1" 200 436 [07/Feb/2026 17:44:58] "GET / HTTP/1.1" 404 2305 [07/Feb/2026 17:44:59] "GET / HTTP/1.1" 404 2305 [07/Feb/2026 17:45:18] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:45:22] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:47:06] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:47:08] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:47:09] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:47:38] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:47:39] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:47:41] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:47:52] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:47:53] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 17:47:53] "GET /api/rooms/1/messages/ HTTP/1.1" 200 2 [07/Feb/2026 17:47:54] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 17:47:54] "GET /api/rooms/1/messages/ HTTP/1.1" 200 2 [07/Feb/2026 17:47:57] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:50:18] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 17:50:18] "GET /favicon.ico HTTP/1.1" 404 2356 [07/Feb/2026 17:50:21] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 17:50:21] "GET /favicon.ico HTTP/1.1" 404 2356 [07/Feb/2026 17:53:33] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:53:34] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:55:20] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 17:55:21] "GET /favicon.ico HTTP/1.1" 404 2356 [07/Feb/2026 17:56:52] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 17:57:19] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2467 [07/Feb/2026 17:58:15] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:59:50] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2467 [07/Feb/2026 18:00:09] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 18:00:11] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 18:00:12] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 18:00:18] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 18:00:25] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 18:00:25] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 18:00:26] "GET

/ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 18:00:27] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 18:00:27] "GET /api/rooms/1/messages/ HTTP/1.1" 200 2 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:28] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:29] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:07:52] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:07:53] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:08:00] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:08:00] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:08:01] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:08:28] "GET /ws/chat/general/ HTTP/1.1" 404 2371 D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\views.py changed, reloading. Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 18:24:04 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/> D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\views.py changed, reloading. Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 18:24:10 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/> D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\monsite\urls.py changed, reloading. Watching for file changes with StatReloader Performing system checks... Exception in thread django-main-thread: Traceback (most recent call last): File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\urls\resolvers.py", line 720, in url_patterns iter(patterns) TypeError: 'module' object is not iterable The above exception was the direct cause of the following exception: Traceback (most recent call last): File "C:\Program Files\Python3_12_4\Lib\threading.py", line 1073, in _bootstrap_inner self.run() File "C:\Program Files\Python3_12_4\Lib\threading.py", line 1010, in run self._target(*self._args, **self._kwargs) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\utils\autoreload.py", line 64, in wrapper fn(*args, **kwargs) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\management\commands\runserver.py", line 134, in inner_run self.check(**check_kwargs) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\management\base.py", line 492, in check_all_issues = checks.run_checks() File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\checks\registry.py", line 89, in run_checks new_errors = check(app_configs=app_configs, databases=databases) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\checks\urls.py", line 16, in check_url_config return check_resolver(resolver) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\checks\urls.py", line 26, in check_resolver return check_method() File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\urls\resolvers.py", line 531, in check_for_pattern in self.url_patterns: File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\utils\functional.py", line 47, in __get__ res = instance.__dict__[self.name] = self.func(instance) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\urls\resolvers.py", line 728, in url_patterns raise ImproperlyConfigured(msg.format(name=self.urlconf_name)) from e django.core.exceptions.ImproperlyConfigured: The included URLconf 'monsite.urls' does not appear to have any patterns in it. If you see the 'urlpatterns' variable with valid patterns in the file then the issue is probably caused by a circular import. D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\monsite\urls.py changed, reloading. Watching for file changes with StatReloader Performing system checks... Exception in thread django-main-thread: Traceback (most recent call last): File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\urls\resolvers.py", line 720, in url_patterns iter(patterns) TypeError: 'module' object is not iterable The above exception was the direct cause of

[illegible]

twisted[tls]>=22.4->daphne) Downloading twisted-25.5.0-py3-none-any.whl.metadata (22 kB) Collecting cbor2>=5.2.0 (from autobahn>=22.4.2->daphne)
Downloading cbor2-5.8.0-cp312-cp312-win_amd64.whl.metadata (5.6 kB) Collecting cffi>=2.0.0 (from autobahn>=22.4.2->daphne) Using cached cffi-2.0.0-
cp312-cp312-win_amd64.whl.metadata (2.6 kB) Collecting cryptography>=3.4.6 (from autobahn>=22.4.2->daphne) Downloading cryptography-46.0.4-cp311-
abi3-win_amd64.whl.metadata (5.7 kB) Collecting hyperlink>=21.0.0 (from autobahn>=22.4.2->daphne) Downloading hyperlink-21.0.0-py2.py3-none-
any.whl.metadata (1.5 kB) Requirement already satisfied: msgpack>=1.0.2 in d:\python django hopes\chat_api_django_mario\venv\lib\site-packages (from
autobahn>=22.4.2->daphne) (1.1.2) Collecting py-ubjson>=0.16.1 (from autobahn>=22.4.2->daphne) Downloading py-ubjson-0.16.1.tar.gz (50 kB) —————
50.3/50.3 kB 284.2 kB/s eta 0:00:00 Installing build dependencies ...
done Getting requirements to build wheel ... done Preparing metadata (pyproject.toml) ... done Collecting txai>=25.12.2 (from autobahn>=22.4.2->daphne)
Downloading txai-25.12.2-py3-none-any.whl.metadata (7.0 kB) Collecting ujson>=4.0.2 (from autobahn>=22.4.2->daphne) Downloading ujson-5.11.0-
cp312-cp312-win_amd64.whl.metadata (9.6 kB) Collecting attrs>=22.2.0 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading attrs-25.4.0-py3-
none-any.whl.metadata (10 kB) Collecting automat>=24.8.0 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading automat-25.4.16-py3-none-
any.whl.metadata (8.4 kB) Collecting constantly>=15.1 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading constantly-23.10.4-py3-none-
any.whl.metadata (1.8 kB) Collecting incremental>=24.7.0 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading incremental-24.11.0-py3-none-
any.whl.metadata (9.5 kB) Collecting typing-extensions>=4.2.0 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Using cached typing_extensions-4.15.0-
py3-none-any.whl.metadata (3.3 kB) Collecting zope-interface>=5 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading zope_interface-8.2-
cp312-cp312-win_amd64.whl.metadata (46 kB) ————— 46.7/46.7
kB 331.3 kB/s eta 0:00:00 Collecting idna>=2.4 (from twisted[tls]>=22.4->daphne) Downloading idna-3.11-py3-none-any.whl.metadata (8.4 kB) Collecting
pyopenssl>=21.0.0 (from twisted[tls]>=22.4->daphne) Downloading pyopenssl-25.3.0-py3-none-any.whl.metadata (17 kB) Collecting service-identity>=18.1.0
(from twisted[tls]>=22.4->daphne) Downloading service_identity-24.2.0-py3-none-any.whl.metadata (5.1 kB) Collecting pycparser (from cffi>=2.0.0-
>autobahn>=22.4.2->daphne) Downloading pycparser-3.0-py3-none-any.whl.metadata (8.2 kB) Collecting packaging>=17.0 (from incremental>=24.7.0-
>twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading packaging-26.0-py3-none-any.whl.metadata (3.3 kB) Collecting pyasn1 (from service-
identity>=18.1.0->twisted[tls]>=22.4->daphne) Downloading pyasn1-0.6.2-py3-none-any.whl.metadata (8.4 kB) Collecting pyasn1-modules (from service-
identity>=18.1.0->twisted[tls]>=22.4->daphne) Using cached pyasn1_modules-0.4.2-py3-none-any.whl.metadata (3.5 kB) Downloading daphne-4.2.1-py3-
none-any.whl (29 kB) Downloading autobahn-25.12.2-cp312-cp312-win_amd64.whl (2.2 MB) —————
2.2/2.2 MB 1.2 MB/s eta 0:00:00 Downloading twisted-25.5.0-py3-none-any.whl (3.2 MB) —————
3.2/3.2 MB 1.5 MB/s eta 0:00:00 Downloading attrs-25.4.0-py3-none-any.whl (67 kB)
————— 67.6/67.6 kB 459.0 kB/s eta 0:00:00 Downloading
automat-25.4.16-py3-none-any.whl (42 kB) ————— 42.8/42.8 kB
299.1 kB/s eta 0:00:00 Downloading cbor2-5.8.0-cp312-cp312-win_amd64.whl (69 kB) —————
69.8/69.8 kB 546.6 kB/s eta 0:00:00 Using cached cffi-2.0.0-cp312-cp312-win_amd64.whl (183 kB) Downloading constantly-
23.10.4-py3-none-any.whl (13 kB) Downloading cryptography-46.0.4-cp311-abi3-win_amd64.whl (3.5 MB) —————
3.5/3.5 MB 844.1 kB/s eta 0:00:00 Downloading hyperlink-21.0.0-py2.py3-none-any.whl (74 kB) —————
74.6/74.6 kB 825.6 kB/s eta 0:00:00 Downloading idna-3.11-py3-
none-any.whl (71 kB) ————— 71.0/71.0 kB 129.6 kB/s eta 0:00:00
Downloading incremental-24.11.0-py3-none-any.whl (21 kB) Downloading pyopenssl-25.3.0-py3-none-any.whl (57 kB) —————

57.3/57.3 kB 505.7 kB/s eta 0:00:00 Downloading service_identity-24.2.0-py3-none-any.whl (11 kB) Downloading txaio-25.12.2-py3-none-any.whl (31 kB) Using cached typing_extensions-4.15.0-py3-none-any.whl (44 kB) Downloading ujson-5.11.0-cp312-cp312-win_amd64.whl (43 kB) 43.9/43.9 kB 357.6 kB/s eta 0:00:00 Downloading zope_interface-8.2-cp312-cp312-win_amd64.whl (212 kB) 212.4/212.4 kB 1.2 MB/s eta 0:00:00 Downloading packaging-26.0-py3-none-any.whl (74 kB) 74.4/74.4 kB 372.8 kB/s eta 0:00:00 Downloading pyasn1-0.6.2-py3-none-any.whl (83 kB) 83.4/83.4 kB 778.0 kB/s eta 0:00:00 Using cached pyasn1_modules-0.4.2-py3-none-any.whl (181 kB) Downloading pycparser-3.0-py3-none-any.whl (48 kB) 48.2/48.2 kB 403.5 kB/s eta 0:00:00 Building wheels for collected packages: py-ubjson Building wheel for py-ubjson (pyproject.toml) ... done Created wheel for py-ubjson: filename=py_ubjson-0.16.1-cp312-cp312-win_amd64.whl size=19356 sha256=7b3fc1d543eefd1d8e1994a85b47c17a024a1a95741dc0ced87344b32adfb966 Stored in directory: c:\users\spsse02men\appdata\local\pip\cache\wheels\c0\8\73\e1931c0c1125bc84ae955eeb3d8ba9fe23ecd95ee337095267 Successfully built py-ubjson Installing collected packages: py-ubjson, zope-interface, ujson, typing-extensions, txaio, pycparser, pyasn1, packaging, idna, constantly, cbor2, automat, attrs, pyasn1-modules, incremental, hyperlink, cffi, twisted, cryptography, service-identity, pyopenssl, Successfully installed attrs-25.4.0 autobahn-25.12.2 automat-25.4.16 cbor2-5.8.0 cffi-2.0.0 constantly-23.10.4 cryptography-46.0.4 daphne-4.2.1 hyperlink-21.0.0 idna-3.11 incremental-24.11.0 packaging-26.0 py-ubjson-0.16.1 pyasn1-0.6.2 pyasn1-modules-0.4.2 pycparser-3.0 pyopenssl-25.3.0 service-identity-24.2.0 twisted-25.5.0 txaio-25.12.2 typing-extensions-4.15.0 ujson-5.11.0 zope-interface-8.2 [notice] A new release of pip is available: 24.0 -> 26.0.1 [notice] To update, run: python.exe -m pip install --upgrade pip (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> py manage.py runserver Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 18:34:56 Django version 5.2.8, using settings 'monsite.settings' Starting ASGI/Daphne version 4.2.1 development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. Not Found: /ws/chat/general/ HTTP GET /ws/chat/general/ 404 [0.06, 127.0.0.1:22100] Not Found: /ws/chat/general/ HTTP GET /ws/chat/general/ 404 [0.02, 127.0.0.1:22100] HTTP GET / 200 [0.01, 127.0.0.1:22100] WebSocket HANDSHAKING /ws/chat/general/ [127.0.0.1:22107] WebSocket CONNECT /ws/chat/general/ [127.0.0.1:22107] WebSocket DISCONNECT /ws/chat/general/ [127.0.0.1:22107] HTTP GET / 200 [0.01, 127.0.0.1:22100] WebSocket HANDSHAKING /ws/chat/general/ [127.0.0.1:22110] WebSocket CONNECT /ws/chat/general/ [127.0.0.1:22110] HTTP GET /api/rooms/ 200 [0.01, 127.0.0.1:22239] Not Found: /favicon.ico HTTP GET /favicon.ico 404 [0.03, 127.0.0.1:22240] WebSocket HANDSHAKING /ws/chat/general/ [127.0.0.1:22244] WebSocket CONNECT /ws/chat/general/ [127.0.0.1:22244] WebSocket DISCONNECT /ws/chat/general/ [127.0.0.1:22244] WebSocket HANDSHAKING /ws/chat/general/ [127.0.0.1:22250] WebSocket CONNECT /ws/chat/general/ [127.0.0.1:22250] WebSocket DISCONNECT /ws/chat/general/ [127.0.0.1:22250] WebSocket HANDSHAKING /ws/chat/general/ [127.0.0.1:22252] WebSocket CONNECT /ws/chat/general/ [127.0.0.1:22252] Not Found: /ws/chat/general/ HTTP GET /ws/chat/general/ 404 [0.02, 127.0.0.1:22651] Not Found: /ws/chat/general/ HTTP GET /ws/chat/general/ 404 [0.02, 127.0.0.1:22651] Not Found: /ws/chat/general/ HTTP GET /ws/chat/general/ 404 [0.02, 127.0.0.1:22651] Not Found: /ws/chat/general/ HTTP GET /ws/chat/general/ 404 [0.00, 127.0.0.1:22651] Not Found: /ws/chat/general/ HTTP GET /ws/chat/general/ 404 [0.02, 127.0.0.1:22651] WebSocket DISCONNECT /ws/chat/general/ [127.0.0.1:22252] D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\monsite\asgi.py changed, reloading. Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 18:55:06 Django version 5.2.8, using settings 'monsite.settings' Starting ASGI/Daphne version 4.2.1 development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. Exception in thread django-main-thread: Traceback (most recent call last): File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\daphne\management\commands\runserver.py", line 33, in


```
        URLRouter(chat.routing.websocket_urlpatterns)
    },
)
)
```

Important : dans `settings.py`, vous devez avoir :

```
ASGI_APPLICATION = "monsite.asgi.application"
```

2) You must set `settings.ALLOWED_HOSTS` if `DEBUG` is `False`.

Vous avez mis `DEBUG = False` dans `monsite/settings.py`, et dans ce cas Django exige `ALLOWED_HOSTS`.



Solution simple en dev :

```
DEBUG = True
ALLOWED_HOSTS = []
```



Si vous voulez garder `DEBUG = False` :

```
DEBUG = False
ALLOWED_HOSTS = ["127.0.0.1", "localhost"]
```

À propos du “404 /ws/chat/general/”

- Si vous tapez **dans le navigateur** : `http://127.0.0.1:8000/ws/chat/general/` → c’est normal d’avoir **404** (c’est une URL WebSocket, pas une page HTTP).
- Votre page doit être : `http://127.0.0.1:8000/`
- Et elle se connecte en WebSocket sur : `ws://127.0.0.1:8000/ws/chat/general/`



Quand Daphne est actif, on voit bien dans votre log :

WebSocket HANDSHAKING puis CONNECT → donc c’est OK.



ZIP des fichiers corrigés

Voici à nouveau le ZIP contenant les fichiers corrigés (dont `asgi.py` avec `application`) :

Download the zip

Room: general  WS connecté  {"type": "chat_message", "id": 1, "author": "Mario", "content": "salut", "created_at": "2026-02-07T16:24:50.528337+00:00"}

Parfait — ça veut dire que **tout fonctionne côté WebSocket** :

-  **WS connecté** → le handshake + routing Channels OK
-  **message reçu** avec `type: "chat_message"` → votre `ChatConsumer` envoie bien l'événement
- Vous recevez aussi `id`, `author`, `content`, `created_at` → donc la sérialisation / DB (ou création) marche aussi.

Si vous voulez vérifier le “vrai temps réel” :

1. Ouvrez **2 onglets** sur `http://127.0.0.1:8000/`
2. Connectez les deux au salon `general`
3. Envoyez un message dans l'onglet A → il doit apparaître immédiatement dans l'onglet B.

Optionnel (petit plus utile) : si vous voyez parfois `created_at` en UTC (+00:00) et vous voulez l'heure locale, je peux vous montrer comment formater ça côté front (JS) ou côté API (DRF).

oui côté API (DRF).

OK — côté **API DRF**, le plus propre est de **laisser la DB en UTC** et de **formater à l'affichage** (ou de renvoyer aussi une version locale).

Voici 3 options (de la plus simple à la plus “pro”) :

Option A — DRF renvoie ISO8601, et DRF applique votre timezone (simple)

1. Dans `settings.py` :

```
USE_TZ = True
TIME_ZONE = "Indian/Antananarivo" # ou "Africa/Nairobi" si vous êtes à Madagascar, à confirmer
```

2. Dans settings.py (DRF) :

```
REST_FRAMEWORK = {  
    "DATETIME_FORMAT": "%Y-%m-%d %H:%M:%S",  
}
```

➡ Résultat : `created_at` sera renvoyé en "2026-02-07 19:24:50" (sans +00:00).

Option B — garder ISO8601 + ajouter un champ “local” (recommandé)

Dans votre serializer `MessageSerializer` :

```
from django.utils import timezone  
from rest_framework import serializers  
from .models import Message  
  
class MessageSerializer(serializers.ModelSerializer):  
    created_at_local = serializers.SerializerMethodField()  
  
    class Meta:  
        model = Message  
        fields = ["id", "author", "content", "created_at", "created_at_local"]  
  
    def get_created_at_local(self, obj):  
        if not obj.created_at:  
            return None  
        local_dt = timezone.localtime(obj.created_at)  
        return local_dt.strftime("%Y-%m-%d %H:%M:%S")
```

➡ Résultat :

- `created_at` reste ISO8601 (pratique pour machines)
 - `created_at_local` est lisible pour humains
-

Option C — ISO8601 en timezone locale (pro, standard)

Dans `settings.py` :

```
USE_TZ = True
TIME_ZONE = "Indian/Antananarivo"
REST_FRAMEWORK = {
    "DATETIME_FORMAT": None, # garde ISO8601
}
```

DRF renverra toujours ISO8601, mais si vos datetimes sont “aware”, `timezone.localtime()` (option B) reste le meilleur moyen d’être certain de la timezone présentée.

Petit check important

Votre `created_at` actuel est ...+00:00 (UTC). C’est normal.

Avec l’option B, vous gardez l’UTC pour la logique, et vous affichez la version locale proprement.

Si vous me collez votre `serializers.py` (ou le code du serializer actuel), je vous le réécris directement dans votre structure exacte.

serializers.py
Python
Réflexion durant 5s

Parfait 👍 Côté API (DRF), ton `serializers.py` est **bon** : il sérialise bien `Room` et `Message`, et protège `id/created_at` en lecture seule pour `Message`.

serializers

Tester rapidement l’API (Windows / PowerShell)

1) Lister les rooms

```
Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/" -Method Get
```

2) Créer une room (POST)

En PowerShell, `curl` est souvent un alias de `Invoke-WebRequest`, donc ta commande “`curl -H ...`” casse. Utilise **Invoke-RestMethod** comme ci-dessous :


```
$body = @{ name = "general" } | ConvertTo-Json
Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/" -Method Post -ContentType "application/json" -Body $body
```

(Alternative) Si tu veux absolument `curl`, utilise `curl.exe` :

```
curl.exe -X POST "http://127.0.0.1:8000/api/rooms/" -H "Content-Type: application/json" -d '{"name":"general"}"
```

3) Lister les messages d'une room

```
Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/1/messages/" -Method Get
```

4) Poster un message via l'API (si ton endpoint l'autorise)

```
$body = @{ room = 1; author = "Mario"; content = "salut via API" } | ConvertTo-Json
Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/messages/" -Method Post -ContentType "application/json" -Body $body
```

Si tu veux, je peux te dire exactement quelles routes DRF sont exposées

Envoie-moi ton `chat/urls.py` (ou le fichier où tu declares tes routes DRF / router), et je te confirme la liste exacte (`/api/rooms/`, `/api/messages/`, `/api/rooms/<id>/messages/`, etc.).


Sources

`urls.py`

Python

`chat/urls.py`

Réflexion durant 1m 0s

OK  J'ai corrigé la partie **API (DRF)** pour que :

- `POST /api/rooms/` crée un salon
- `GET /api/rooms/` liste les salons
- `GET /api/rooms/<room_id>/messages/` liste les messages d'un salon
- `POST /api/rooms/<room_id>/messages/` crée un message (le `room` est imposé par l'URL, donc pas besoin de l'envoyer dans le body)



Voici le zip avec les fichiers corrigés (incluant `chat/serializers.py`, `chat/views.py`, `chat/urls.py` + les autres fichiers du projet déjà partagés) :
[Download the ZIP](#)

Commandes PowerShell utiles (POST JSON)






Créer une room :

```
Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/" `
-Method Post `
-ContentType "application/json" `
-Body '{"name":"general"}'
```


Créer un message dans la room 1 :


```
Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/1/messages/" `
-Method Post `
-ContentType "application/json" `
-Body '{"author":"Mario","content":"salut"}'
```

Si tu veux aussi une route `POST /api/rooms/<room_id>/messages/` qui accepte *en plus* room dans le body (optionnel), je peux te l'ajouter.

Room: general  WS connecté  {"type": "chat_message", "id": 2, "author": "Mario", "content": "chateau", "created_at": "2026-02-07T16:29:24.751189+00:00"}  {"type": "chat_message", "id": 3, "author": "Mario", "content": "hibou", "created_at": "2026-02-07T16:30:07.003948+00:00"}  WS erreur  WS fermé
Réflexion durant 1m 43s

Ton WS se ferme très souvent quand **une exception non gérée arrive dans le consumer** (ex: message vide, JSON invalide, room introuvable, erreur DB...). Dans ce cas Channels/Daphne coupe la connexion → côté navigateur tu vois  WS erreur puis  WS fermé.

 Je t'ai préparé une version **plus robuste** de `chat/consumers.py` (avec `try/except`, ignore des messages vides, création automatique de la room, et retour d'erreur JSON au lieu de crasher), + `monsite/urls.py` et `chat/urls.py` cohérents.

 Téléchargement du ZIP (tous fichiers corrigés) :
Download `fichiers_corriges_ws_stable.zip`

À faire après remplacement

1. Remplace les fichiers par ceux du ZIP (mêmes chemins).
2. Redémarre le serveur :

```
py manage.py runserver
```

3. Ouvre la page / et reteste le chat.

Si ça se ferme encore

Regarde **la trace exacte côté serveur** au moment de la fermeture (console `runserver`) : c'est elle qui dira la vraie cause (DB, modèle, champ non-null, etc.).

Pouvez vous donner les fichiers complets corrigés

Réflexion durant 1m 28s

Voici **les fichiers complets corrigés** + un **ZIP** prêt à copier dans votre projet.



Télécharger le ZIP : fichiers_corriges_complets_final.zip

monsite/asgi.py

```
"""ASGI config for monsite project.

It exposes the ASGI callable as a module-level variable named ``application``.

This file wires Django + Channels (WebSocket) together.
"""

import os

from django.core.asgi import get_asgi_application
from channels.auth import AuthMiddlewareStack
from channels.routing import ProtocolTypeRouter, URLRouter

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "monsite.settings")

django_asgi_app = get_asgi_application()

import chat.routing  # noqa: E402

application = ProtocolTypeRouter(
    {
        "http": django_asgi_app,
        "websocket": AuthMiddlewareStack(URLRouter(chat.routing.websocket_urlpatterns)),
    }
)
```

monsite/settings.py

```
from pathlib import Path

BASE_DIR = Path(__file__).resolve().parent.parent

SECRET_KEY = "dev-secret-key-change-me"
DEBUG = True

ALLOWED_HOSTS = ["127.0.0.1", "localhost"]

INSTALLED_APPS = [
    "django.contrib.admin",
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",

    # third-party
    "corsheaders",
    "rest_framework",
    "channels",

    # local apps
    "chat",
]

MIDDLEWARE = [
    "corsheaders.middleware.CorsMiddleware",
    "django.middleware.security.SecurityMiddleware",
    "django.contrib.sessions.middleware.SessionMiddleware",
    "django.middleware.common.CommonMiddleware",
    "django.middleware.csrf.CsrfViewMiddleware",
    "django.contrib.auth.middleware.AuthenticationMiddleware",
    "django.contrib.messages.middleware.MessageMiddleware",
    "django.middleware.clickjacking.XFrameOptionsMiddleware",
]

ROOT_URLCONF = "monsite.urls"

TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
        "DIRS": [BASE_DIR / "templates"],
    },
]
```

```

        "APP_DIRS": True,
        "OPTIONS": {
            "context_processors": [
                "django.template.context_processors.request",
                "django.contrib.auth.context_processors.auth",
                "django.contrib.messages.context_processors.messages",
            ],
        },
    ],

WSGI_APPLICATION = "monsite.wsgi.application"
ASGI_APPLICATION = "monsite.asgi.application"

DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.sqlite3",
        "NAME": BASE_DIR / "db.sqlite3",
    }
}

AUTH_PASSWORD_VALIDATORS = [
    {"NAME": "django.contrib.auth.password_validation.UserAttributeSimilarityValidator"},
    {"NAME": "django.contrib.auth.password_validation.MinimumLengthValidator"},
    {"NAME": "django.contrib.auth.password_validation.CommonPasswordValidator"},
    {"NAME": "django.contrib.auth.password_validation.NumericPasswordValidator"},
]

LANGUAGE_CODE = "fr-fr"
TIME_ZONE = "UTC"
USE_I18N = True
USE_TZ = True

STATIC_URL = "static/"
DEFAULT_AUTO_FIELD = "django.db.models.BigAutoField"

# CORS (useful if your frontend is on another port/domain)
CORS_ALLOW_ALL_ORIGINS = True

# Channels: for local/dev we keep it simple and stable with in-memory layer.
# If you want Redis later:
#     pip install channels-redis
#     CHANNEL_LAYERS = {"default": {"BACKEND": "channels_redis.core.RedisChannelLayer",
#                                     "CONFIG": {"hosts": [("127.0.0.1", 6379)]}}}
#
CHANNEL_LAYERS = {
    "default": {

```

```
        "BACKEND": "channels.layers.InMemoryChannelLayer",
    }
}
```

monsite/urls.py

```
from django.contrib import admin
from django.urls import path, include
from chat import views as chat_views

urlpatterns = [
    path("admin/", admin.site.urls),
    path("", chat_views.index, name="index"),
    path("api/", include("chat.urls")),
]
```

chat/routing.py

```
from django.urls import re_path
from .consumers import ChatConsumer

websocket_urlpatterns = [
    re_path(r"^ws/chat/(?P<room_name>[^/]+)/$", ChatConsumer.as_asgi()),
]
```

chat/consumers.py

```
import json
import logging

from channels.db import database_sync_to_async
from channels.generic.websocket import AsyncWebsocketConsumer

from .models import Room, Message

logger = logging.getLogger(__name__)

class ChatConsumer(AsyncWebsocketConsumer):
    async def connect(self):
        self.room_name = self.scope["url_route"]["kwargs"]["room_name"]
        self.group_name = f"chat_{self.room_name}"

        await self.channel_layer.group_add(self.group_name, self.channel_name)
```

```

    await self.accept()

    # Ensure the room exists
    await self.get_or_create_room(self.room_name)

async def disconnect(self, close_code):
    await self.channel_layer.group_discard(self.group_name, self.channel_name)

async def receive(self, text_data=None, bytes_data=None):
    # Be tolerant: never crash the consumer on bad payloads
    try:
        payload = json.loads(text_data or "{}")
    except json.JSONDecodeError:
        await self.send_json({"type": "error", "error": "Invalid JSON"})
        return

    author = (payload.get("author") or "anonymous").strip()[:64]
    content = (payload.get("content") or "").strip()

    if not content:
        await self.send_json({"type": "error", "error": "Empty message"})
        return

    try:
        msg = await self.create_message(self.room_name, author, content)
    except Exception:
        logger.exception("Failed to create message")
        await self.send_json({"type": "error", "error": "Server error saving message"})
        return

    await self.channel_layer.group_send(
        self.group_name,
        [
            {
                "type": "chat_message",
                "id": msg["id"],
                "author": msg["author"],
                "content": msg["content"],
                "created_at": msg["created_at"],
            },
        ],
    )

async def chat_message(self, event):
    await self.send_json(event)

async def send_json(self, data: dict):
    await self.send(text_data=json.dumps(data, ensure_ascii=False))

```

```

@database_sync_to_async
def get_or_create_room(self, name: str):
    Room.objects.get_or_create(name=name)

@database_sync_to_async
def create_message(self, room_name: str, author: str, content: str):
    room, _ = Room.objects.get_or_create(name=room_name)
    msg = Message.objects.create(room=room, author=author, content=content)
    return {
        "id": msg.id,
        "author": msg.author,
        "content": msg.content,
        "created_at": msg.created_at.isoformat(),
    }

```

chat/views.py

```

from django.shortcuts import render
from rest_framework import generics

from .models import Room, Message
from .serializers import RoomSerializer, MessageSerializer

def index(request):
    return render(request, "chat/index.html")

class RoomListCreateView(generics.ListCreateAPIView):
    queryset = Room.objects.all().order_by("name")
    serializer_class = RoomSerializer

class MessageListView(generics.ListAPIView):
    serializer_class = MessageSerializer

    def get_queryset(self):
        room_id = self.kwargs["room_id"]
        return Message.objects.filter(room_id=room_id).order_by("created_at")[:200]

```

chat/serializers.py

```

from rest_framework import serializers
from .models import Room, Message

```



```

class RoomSerializer(serializers.ModelSerializer):
    class Meta:
        model = Room
        fields = ["id", "name"]

class MessageSerializer(serializers.ModelSerializer):
    class Meta:
        model = Message
        fields = ["id", "room", "author", "content", "created_at"]
        read_only_fields = ["id", "created_at"]

```

chat/urls.py

```

from django.urls import path
from . import views

urlpatterns = [
    path("rooms/", views.RoomListCreateView.as_view(), name="room-list"),
    path("rooms/<int:room_id>/messages/", views.MessageListView.as_view(), name="message-list"),
]

```

templates/chat/index.html

```

<!doctype html>
<html lang="fr">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Chat (Django + Channels)</title>
    <style>
        body { font-family: system-ui, sans-serif; margin: 24px; }
        .row { display: flex; gap: 8px; flex-wrap: wrap; }
        input, button { padding: 10px 12px; font-size: 14px; }
        input { min-width: 220px; }
        pre { background: #f5f5f5; padding: 12px; border-radius: 8px; overflow: auto; }
        .badge { display:inline-block; padding: 2px 8px; border-radius: 999px; background:#eee; }
    </style>
</head>
<body>
    <h1>Chat</h1>

    <div class="row">

```

```

<label>Room:
  <input id="room" value="general" />
</label>

<label>Author:
  <input id="author" value="Mario" />
</label>

  <span class="badge" id="status">offline</span>
</div>

<div class="row" style="margin-top: 12px;">
  <input id="msg" placeholder="message..." />
  <button id="send">Envoyer</button>
  <button id="reconnect">Reconnect</button>
</div>

<pre id="log" style="margin-top: 12px; height: 360px;"></pre>

<script>
  const elLog = document.getElementById("log");
  const elStatus = document.getElementById("status");

  const log = (t) => {
    elLog.textContent += t + "\n";
    elLog.scrollTop = elLog.scrollHeight;
  };

  let ws = null;
  let reconnectDelayMs = 500;
  let reconnectTimer = null;

  const setStatus = (s) => {
    elStatus.textContent = s;
  };

  function wsUrl() {
    const room = document.getElementById("room").value.trim() || "general";
    const proto = (location.protocol === "https:") ? "wss" : "ws";
    return `${proto}://${location.host}/ws/chat/${encodeURIComponent(room)}`;
  }

  function connect() {
    if (reconnectTimer) {
      clearTimeout(reconnectTimer);
      reconnectTimer = null;
    }
  }

```

```

}

if (ws && (ws.readyState === WebSocket.OPEN || ws.readyState === WebSocket.CONNECTING)) {
  return;
}

setStatus("connecting");
const url = wsUrl();
log("🔌 " + url);

ws = new WebSocket(url);

ws.onopen = () => {
  reconnectDelayMs = 500;
  setStatus("online");
  log("✅ WS connecté");
};

ws.onmessage = (e) => {
  log("📄 " + e.data);
};

ws.onerror = () => {
  log("⚠️ WS erreur");
};

ws.onclose = () => {
  setStatus("offline");
  log("❌ WS fermé");

  // Auto-reconnect (useful if the dev server restarts)
  reconnectTimer = setTimeout(() => {
    reconnectDelayMs = Math.min(8000, reconnectDelayMs * 2);
    connect();
  }, reconnectDelayMs);
};
}

function sendMsg() {
  if (!ws || ws.readyState !== WebSocket.OPEN) {
    log("⚠️ Pas connecté (message non envoyé)");
    return;
  }
  const author = document.getElementById("author").value.trim() || "anonymous";

```

```

    const content = document.getElementById("msg").value;
    ws.send(JSON.stringify({ author, content }));
    document.getElementById("msg").value = "";
}

document.getElementById("send").onclick = sendMsg;
document.getElementById("msg").addEventListener("keydown", (e) => {
    if (e.key === "Enter") sendMsg();
});

document.getElementById("reconnect").onclick = () => {
    try { ws && ws.close(); } catch(e) {}
    connect();
};

// Start
connect();
</script>
</body>
</html>

```

Si vous me donnez aussi votre `chat/models.py`, je peux vérifier que les champs `Room` / `Message` correspondent bien aux `serializers/consumer` (et ajuster si besoin).

`models.py`

Python

Room: Author: offline 🖱️ ws://127.0.0.1:5500/ws/chat/general/ ⚠️ Pas connecté (message non envoyé) ⚠️ Pas connecté (message non envoyé) 🖱️

ws://127.0.0.1:5500/ws/chat/general/ ⚠️ WS erreur ❌ WS fermé

Réflexion durant 1m 3s

Votre log montre que votre page **essaie d'ouvrir le WebSocket sur le port 5500** :

ws://127.0.0.1:5500/ws/chat/general/

➡️ **C'est la cause du "offline / pas connecté"** : le port **5500** (Live Server / VSCode) ne sert **pas** Django/Channels, donc l'URL WS doit pointer vers **Django (8000)**.



Correctif (front)

Si vous ouvrez `index.html` via Live Server (5500), gardez quand même :

- API → `http://127.0.0.1:8000/api/...`
- WS → `ws://127.0.0.1:8000/ws/chat/<room>/`

Je vous ai généré un **index.html corrigé** (il force : 8000 même si la page est servie en : 5500) :

👉 **Télécharger le zip contenant le index.html corrigé :**

Download `index_fixed.zip`

Notes importantes

- Quand vous tapez `/ws/chat/general/` dans le navigateur (en HTTP), c'est normal d'avoir un 404 : c'est une **route WebSocket**, pas une page HTTP.
- Assurez-vous que **Django tourne bien sur 8000** (`py manage.py runserver` ou daphne).

Si vous voulez, collez ici le contenu actuel de votre `index.html` (celui utilisé avec 5500) et je vous dis exactement la ligne à remplacer.

Room: general ⚠ WS erreur ❌ WS fermé
Réflexion durant 43s

Votre erreur vient du fait que votre page est servie sur **127.0.0.1:5500** (Live Server), donc si le JS construit l'URL du WebSocket avec `window.location.host`, il essaye de se connecter à **5500**... alors que **Django/Channels écoute sur 8000**.

✅ **Il faut forcer le WebSocket vers 127.0.0.1:8000** (même si le HTML est ouvert sur 5500).

Je vous ai préparé un **index.html corrigé** (WS toujours sur 8000, et code de fermeture affiché) dans un zip :


👉 **Télécharger index_ws_port_8000.zip**




À vérifier côté exécution


- Lancez Django : `py manage.py runserver` (ou `py manage.py runserver` avec daphne installé, c'est OK)
- Ouvrez soit :
 - <http://127.0.0.1:8000/> (recommandé, servi par Django)
 - ou votre Live Server **5500**, mais le **WS doit rester sur 8000** (corrigé dans le fichier fourni)


Pourquoi vous voyiez ça

Quand vous aviez :

 `ws://127.0.0.1:5500/ws/chat/general/`

 c'est normal que ça finisse par  WS erreur /  WS fermé, car il n'y a **pas** Channels sur 5500.


Si après ce correctif ça se ferme encore, regardez le **code** affiché :  WS fermé (code=xxxx) et la console Django (souvent un 1011 avec une exception côté consumer).

Chat Room Author offline Logs  `ws://127.0.0.1:8000/ws/chat/general/`  `ws://127.0.0.1:8000/ws/chat/general/`  WS erreur  WS fermé (code=1006)

 WS erreur  WS fermé (code=1006)

Réflexion durant 1m 23s

Le code **1006** vient très souvent d'une **exception côté serveur** (le consumer plante → fermeture “anormale”). Dans ton cas, il y a un **race condition** : tu fais `accept()` **avant** de créer la room en base, donc si le navigateur envoie un message immédiatement après `onopen`, `Room.objects.get(...)` peut échouer → crash → close 1006.

 J'ai corrigé ça en **créant la room AVANT `accept()`** + en rendant `create_message()` robuste (`get_or_create`) + en ajoutant une protection `JSONDecodeError`.

J'ai aussi corrigé/complété les fichiers “API + WS + index”.

 **Télécharge le zip complet corrigé :**

Download `fichiers_corriges_ws_1006_fix.zip`

Contenu du zip :

- `monsite/asgi.py`
- `monsite/settings.py`
- `monsite/urls.py`
- `chat/routing.py`
- `chat/consumers.py` (**fix 1006**)
- `chat/models.py`
- `chat/serializers.py`
- `chat/views.py`
- `chat/urls.py` (DRF)
- `chat/templates/chat/index.html` (client WS robuste)

Si tu vois encore une fermeture, copie/colle juste la trace côté serveur (console runserver/daphne) au moment du 1006 : on la corrigera au millimètre.

```
venv\Lib\site-packages\redis\multidb\client.py: config.default_failover_strategy() venv\Lib\site-packages\redis\multidb\client.py: if config.failover_strategy is
None venv\Lib\site-packages\redis\multidb\client.py: else config.failover_strategy venv\Lib\site-packages\redis\multidb\client.py: self._auto_fallback_interval
= config.auto_fallback_interval venv\Lib\site-packages\redis\multidb\client.py: self._event_dispatcher = config.event_dispatcher venv\Lib\site-
packages\redis\multidb\client.py: self._command_retry = config.command_retry venv\Lib\site-packages\redis\multidb\client.py:
failover_attempts=config.failover_attempts, venv\Lib\site-packages\redis\multidb\client.py: failover_delay=config.failover_delay, venv\Lib\site-
packages\redis\multidb\client.py: self._config = config venv\Lib\site-packages\redis\multidb\command_executor.py:from redis.multidb.config import
DEFAULT_AUTO_FALLBACK_INTERVAL venv\Lib\site-packages\redis\multidb\config.py:class DatabaseConfig: venv\Lib\site-
packages\redis\multidb\config.py: Dataclass representing the configuration for a database connection. venv\Lib\site-packages\redis\multidb\config.py: This
class is used to store configuration settings for a database connection, venv\Lib\site-packages\redis\multidb\config.py: attributes and allows for the creation of
customized configurations for various venv\Lib\site-packages\redis\multidb\config.py: from_pool (Optional[ConnectionPool]): A pre-configured connection
pool to use. venv\Lib\site-packages\redis\multidb\config.py:class MultiDbConfig: venv\Lib\site-packages\redis\multidb\config.py: Configuration class for
managing multiple database connections in a resilient and fail-safe manner. venv\Lib\site-packages\redis\multidb\config.py: databases_config: A list of
database configurations. venv\Lib\site-packages\redis\multidb\config.py: Retrieves a collection of database clients managed by weighted configurations.
venv\Lib\site-packages\redis\multidb\config.py: Initializes database clients based on the provided configuration and removes venv\Lib\site-
packages\redis\multidb\config.py: with defined retry and backoff configurations. venv\Lib\site-packages\redis\multidb\config.py: databases_config:
List[DatabaseConfig] venv\Lib\site-packages\redis\multidb\config.py: for database_config in self.databases_config: venv\Lib\site-
packages\redis\multidb\config.py: database_config.client_kwargs.update( venv\Lib\site-packages\redis\multidb\config.py: if database_config.from_url:
venv\Lib\site-packages\redis\multidb\config.py: database_config.from_url, **database_config.client_kwargs venv\Lib\site-packages\redis\multidb\config.py:
elif database_config.from_pool: venv\Lib\site-packages\redis\multidb\config.py: database_config.from_pool.set_retry( venv\Lib\site-
packages\redis\multidb\config.py: connection_pool=database_config.from_pool venv\Lib\site-packages\redis\multidb\config.py: client =
self.client_class(**database_config.client_kwargs) venv\Lib\site-packages\redis\multidb\config.py: database_config.default_circuit_breaker() venv\Lib\site-
packages\redis\multidb\config.py: if database_config.circuit is None venv\Lib\site-packages\redis\multidb\config.py: else database_config.circuit venv\Lib\site-
packages\redis\multidb\config.py: weight=database_config.weight, venv\Lib\site-packages\redis\multidb\config.py:
health_check_url=database_config.health_check_url, venv\Lib\site-packages\redis\multidb\config.py: database_config.weight, venv\Lib\site-
packages\redis\multidb\healthcheck.py: "Database health check url is not set. Please check DatabaseConfig for the current database." venv\Lib\site-
packages\redis\_parsers\base.py:# user send an AUTH cmd to a server without authorization configured venv\Lib\site-packages\redis\_parsers\base.py:
"configured for the default user. Are you sure " venv\Lib\site-packages\redis\_parsers\base.py: "your configuration is correct?": AuthenticationError,
venv\Lib\site-packages\redis\_parsers\helpers.py: "config-epoch": int, venv\Lib\site-packages\redis\_parsers\helpers.py:def parse_config_get(response,
**options): venv\Lib\site-packages\redis\_parsers\helpers.py: "CLUSTER SAVECONFIG": bool_ok, venv\Lib\site-packages\redis\_parsers\helpers.py:
"CLUSTER SET-CONFIG-EPOCH": bool_ok, venv\Lib\site-packages\redis\_parsers\helpers.py: "CONFIG RESETSTAT": bool_ok, venv\Lib\site-
packages\redis\_parsers\helpers.py: "CONFIG SET": bool_ok, venv\Lib\site-packages\redis\_parsers\helpers.py: "SENTINEL FLUSHCONFIG": bool_ok,
venv\Lib\site-packages\redis\_parsers\helpers.py: "CONFIG GET": parse_config_get, venv\Lib\site-packages\redis\_parsers\helpers.py: "CONFIG GET":
lambda r: { venv\Lib\site-packages\rest_framework\apps.py:from django.apps import AppConfig venv\Lib\site-packages\rest_framework\apps.py:class
RestFrameworkConfig(AppConfig): venv\Lib\site-packages\rest_framework\authtoken\apps.py:from django.apps import AppConfig venv\Lib\site-
packages\rest_framework\authtoken\apps.py:class AuthTokenConfig(AppConfig): venv\Lib\site-packages\rest_framework\checks.py:def
```

```

pagination_system_check(app_configs, **kwargs): venv\Lib\site-packages\rest_framework\compat.py: extension_configs = { venv\Lib\site-
packages\rest_framework\compat.py: extensions=extensions, extension_configs=extension_configs venv\Lib\site-packages\rest_framework\fields.py: Quantize
the decimal value to the configured precision. venv\Lib\site-packages\rest_framework\filters.py: from django.core.exceptions import FieldDoesNotExist,
ImproperlyConfigured venv\Lib\site-packages\rest_framework\filters.py: raise ImproperlyConfigured(msg % self.__class__.__name__) venv\Lib\site-
packages\rest_framework\generics.py: method if you want to apply the configured filtering backend to the venv\Lib\site-
packages\rest_framework\management\commands\generateschema.py: help = "Generates configured API schema for project." venv\Lib\site-
packages\rest_framework\pagination.py: page object, or None if pagination is not configured for this view. venv\Lib\site-
packages\rest_framework\relations.py: from django.core.exceptions import ImproperlyConfigured, ObjectDoesNotExist venv\Lib\site-
packages\rest_framework\relations.py: attributes are not configured to correctly match the URL conf. venv\Lib\site-packages\rest_framework\relations.py: #
Return the hyperlink, or error if incorrectly configured. venv\Lib\site-packages\rest_framework\relations.py: 'model in your API, or incorrectly configured the '
venv\Lib\site-packages\rest_framework\relations.py: raise ImproperlyConfigured(msg % self.view_name) venv\Lib\site-
packages\rest_framework\renderers.py: from django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\renderers.py: raise
ImproperlyConfigured( venv\Lib\site-packages\rest_framework\routers.py: from django.core.exceptions import ImproperlyConfigured venv\Lib\site-
packages\rest_framework\routers.py: raise ImproperlyConfigured(msg) venv\Lib\site-packages\rest_framework\routers.py: raise ImproperlyConfigured(msg %
', '.join(not_allowed)) venv\Lib\site-packages\rest_framework\serializers.py: from django.core.exceptions import FieldDoesNotExist, ImproperlyConfigured
venv\Lib\site-packages\rest_framework\serializers.py: raise ImproperlyConfigured( venv\Lib\site-packages\rest_framework\settings.py: # View configuration
venv\Lib\site-packages\rest_framework\test.py: from django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\test.py:
raise ImproperlyConfigured('requests must be installed in order to use RequestsClient.') venv\Lib\site-packages\rest_framework\test.py: raise
ImproperlyConfigured('coreapi must be installed in order to use CoreAPIClient.') venv\Lib\site-packages\rest_framework\throttling.py: from
django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\throttling.py: raise ImproperlyConfigured(msg) venv\Lib\site-
packages\rest_framework\throttling.py: raise ImproperlyConfigured(msg) venv\Lib\site-packages\rest_framework\views.py: # Filter out None values which
may happen in case of config / rate venv\Lib\site-packages\sqlparse\lexer.py: """The Lexer supports configurable syntax. venv\Lib\site-
packages\sqlparse\lexer.py: """Clear all syntax configurations. venv\Lib\site-packages\sqlparse\lexer.py: Useful if you want to load a reduced set of syntax
configurations. venv\Lib\site-packages\tutorial\asgi.py: ASGI config for tutorial project. venv\Lib\site-packages\tutorial\quickstart\apps.py: class
QuickstartConfig(AppConfig): venv\Lib\site-packages\tutorial\wsgi.py: WSGI config for tutorial project. (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> python manage.py makemigrations Migrations for 'chat': chat\migrations\0001_initial.py + Create model Room + Create
model Message (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py migrate Operations to perform: Apply all
migrations: admin, auth, chat, contenttypes, sessions Running migrations: Applying contenttypes.0001_initial... OK Applying auth.0001_initial... OK Applying
admin.0001_initial... OK Applying admin.0002_logentry_remove_auto_add... OK Applying admin.0003_logentry_add_action_flag_choices... OK Applying
contenttypes.0002_remove_content_type_name... OK Applying auth.0002_alter_permission_name_max_length... OK Applying
auth.0003_alter_user_email_max_length... OK Applying auth.0004_alter_user_username_opts... OK Applying auth.0005_alter_user_last_login_null... OK
Applying auth.0006_require_contenttypes_0002... OK Applying auth.0008_alter_user_username_max_length... OK Applying
auth.0010_alter_group_name_max_length... OK Applying auth.0011_update_proxy_permissions... OK Applying
auth.0012_alter_user_first_name_max_length... OK Applying chat.0001_initial... OK Applying sessions.0001_initial... OK (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> python -c "from django.core.asgi import get_asgi_application; print('django ok')" django ok (venv) PS D:\PYTHON
DJANGO HOPES\Chat_API_DJANGO_mario> python -c "import chat.routing; print('routing ok')" Traceback (most recent call last): File "<string>", line 1, in
<module> File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\routing.py", line 2, in <module> from .consumers import ChatConsumer

```


[illegible]

[07/Feb/2026 17:04:22] "GET /static/admin/js/nav_sidebar.js HTTP/1.1" 200 3063 Not Found: /favicon.ico [07/Feb/2026 17:04:22] "GET /favicon.ico HTTP/1.1" 404 2356 Not Found: /api/ [07/Feb/2026 17:04:37] "GET /api/ HTTP/1.1" 404 2739 Not Found: / [07/Feb/2026 17:04:48] "GET / HTTP/1.1" 404 2305 [07/Feb/2026 17:05:01] "GET /admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:05:01] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 4252 Not Found: /api/ [07/Feb/2026 17:05:12] "GET /api/ HTTP/1.1" 404 2739 Not Found: / [07/Feb/2026 17:24:49] "GET /api/rooms/ HTTP/1.1" 200 2 Not Found: /favicon.ico [07/Feb/2026 17:24:49] "GET /favicon.ico HTTP/1.1" 404 2356 (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> curl -X POST http://127.0.0.1:8000/api/rooms/ -H "Content-Type: application/json" -d '{"name":"general"}' >> ^C (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> curl -X POST http://127.0.0.1:8000/api/rooms/ -H "Content-Type: application/json" -d '{"name":"general"}' Invoke-WebRequest : Impossible de lier le paramètre «Headers». Impossible de convertir la valeur «Content-Type: application/json» du type «System.String» en type «System.Collections.IDictionary». Au caractère Ligne:1 : 50 + ~~~~~ + CategoryInfo : InvalidArgument : (:) [Invoke-WebRequest], ParameterBindingException + FullyQualifiedErrorId : CannotConvertArgumentNoMessage,Microsoft.PowerShell.Commands.InvokeWebRequestCommand (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/" >> -Method Post >> -ContentType "application/json" Invoke-RestMethod : Impossible de se connecter au serveur distant Au caractère Ligne:1 : 1 + Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/" + ~~~~~ + CategoryInfo : NotSpecified: (:) [Invoke-RestMethod], WebException + FullyQualifiedErrorId : System.Net.WebException,Microsoft.PowerShell.Commands.InvokeRestMethodCommand (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> py manage.py runserver Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 17:30:20 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: https://docs.djangoproject.com/en/5.2/howto/deployment/ Not Found: / [07/Feb/2026 17:30:24] "GET / HTTP/1.1" 404 2305 Not Found: / [07/Feb/2026 17:30:25] "GET / HTTP/1.1" 404 2305 Not Found: /ws/chat/general/ [07/Feb/2026 17:31:01] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:31:04] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:31:24] "GET /api/rooms/ HTTP/1.1" 200 2 Not Found: /ws/chat/general/ [07/Feb/2026 17:31:32] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:31:58] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:32:00] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:32:02] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:32:04] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:21] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:22] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:22] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:32] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:33] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:35] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /api/ \u0222 WS: ws://127.0.0.1:8000/ws/chat/<room>/ [07/Feb/2026 17:35:44] "GET /api/%20%E2%80%A2%20WS:%20ws://127.0.0.1:8000/ws/chat/%3Croom%3E/ HTTP/1.1" 404 2899 Not Found: /api/ [07/Feb/2026 17:35:51] "GET /api/ HTTP/1.1" 404 2739 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:43] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:44] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:44] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:45] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:46] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:47] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:38:30] "GET /api/rooms/ HTTP/1.1" 200 2 Not Found: /favicon.ico [07/Feb/2026 17:38:30] "GET /favicon.ico HTTP/1.1" 404 2356 Not Found: / [07/Feb/2026 17:38:52] "GET / HTTP/1.1" 404 2305 [07/Feb/2026 17:41:00] "GET /admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:41:00] "GET

```

/admin/login/?next=/admin/ HTTP/1.1" 200 4252 [07/Feb/2026 17:41:01] "GET /static/admin/css/dark_
"GET /static/admin/css/base.css HTTP/1.1" 200 22120 [07/Feb/2026 17:41:01] "GET /static/admin/css/nav_sidebar.css HTTP/1.1" 200 2810 [07/Feb/2026 17:41:01] "GET /static/admin/js/theme.js HTTP/1.1" 200 1653 [07/Feb/2026 17:41:01] "GET /static/admin/js/nav_sidebar.js HTTP/1.1" 200 3063 [07/Feb/2026 17:41:01] "GET /static/admin/css/login.css HTTP/1.1" 200 951 [07/Feb/2026 17:41:01] "GET /static/admin/css/responsive.css HTTP/1.1" 200 16565 [07/Feb/2026 17:41:35] "POST /admin/login/?next=/admin/ HTTP/1.1" 200 4498 [07/Feb/2026 17:44:04] "POST /admin/login/?next=/admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:44:04] "GET /admin/ HTTP/1.1" 200 7885 [07/Feb/2026 17:44:05] "GET /static/admin/css/dashboard.css HTTP/1.1" 200 441 [07/Feb/2026 17:44:05] "GET /static/admin/img/icon-changelink.svg HTTP/1.1" 200 380 [07/Feb/2026 17:44:05] "GET /static/admin/img/icon-addlink.svg HTTP/1.1" 200 331 [07/Feb/2026 17:44:27] "GET /admin/chat/room/add/ HTTP/1.1" 200 9637 [07/Feb/2026 17:44:27] "GET /static/admin/js/jquery.init.js HTTP/1.1" 200 347 [07/Feb/2026 17:44:27] "GET /static/admin/css/forms.css HTTP/1.1" 200 8525 [07/Feb/2026 17:44:27] "GET /static/admin/js/core.js HTTP/1.1" 200 6208 [07/Feb/2026 17:44:27] "GET /static/admin/js/urlify.js HTTP/1.1" 200 7887 [07/Feb/2026 17:44:27] "GET /static/admin/js/prepopulate.js HTTP/1.1" 200 1531 [07/Feb/2026 17:44:27] "GET /admin/jsi18n/ HTTP/1.1" 200 8634 [07/Feb/2026 17:44:27] "GET /static/admin/js/change_form.js HTTP/1.1" 200 606 [07/Feb/2026 17:44:27] "GET /static/admin/js/admin/RelatedObjectLookups.js HTTP/1.1" 200 9777 [07/Feb/2026 17:44:27] "GET /static/admin/js/actions.js HTTP/1.1" 200 8076 [07/Feb/2026 17:44:27] "GET /static/admin/js/prepopulate_init.js HTTP/1.1" 200 586 [07/Feb/2026 17:44:27] "GET /static/admin/css/widgets.css HTTP/1.1" 200 11973 [07/Feb/2026 17:44:28] "GET /static/admin/js/vendor/xregexp/xregexp.js HTTP/1.1" 200 325171 [07/Feb/2026 17:44:28] "GET /static/admin/js/vendor/jquery/jquery.js HTTP/1.1" 200 285314 [07/Feb/2026 17:44:44] "POST /admin/chat/room/add/ HTTP/1.1" 302 0 [07/Feb/2026 17:44:44] "GET /admin/chat/room/ HTTP/1.1" 200 10874 [07/Feb/2026 17:44:45] "GET /admin/jsi18n/ HTTP/1.1" 200 8634 [07/Feb/2026 17:44:45] "GET /static/admin/css/changelists.css HTTP/1.1" 200 6878 [07/Feb/2026 17:44:45] "GET /static/admin/js/filters.js HTTP/1.1" 200 978 [07/Feb/2026 17:44:45] "GET /static/admin/img/search.svg HTTP/1.1" 200 458 [07/Feb/2026 17:44:45] "GET /static/admin/img/tooltag-add.svg HTTP/1.1" 200 331 [07/Feb/2026 17:44:45] "GET /static/admin/img/icon-yes.svg HTTP/1.1" 200 436 [07/Feb/2026 17:44:58] "GET / HTTP/1.1" 404 2305 [07/Feb/2026 17:44:59] "GET / HTTP/1.1" 404 2305 [07/Feb/2026 17:45:18] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:45:22] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:47:06] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:47:08] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:47:09] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:47:38] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:47:39] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:47:41] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:47:52] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:47:53] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 17:47:53] "GET /api/rooms/1/messages/ HTTP/1.1" 200 2 [07/Feb/2026 17:47:54] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 17:47:54] "GET /api/rooms/1/messages/ HTTP/1.1" 200 2 [07/Feb/2026 17:47:57] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:50:18] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 17:50:18] "GET /favicon.ico HTTP/1.1" 404 2356 [07/Feb/2026 17:50:21] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 17:50:21] "GET /favicon.ico HTTP/1.1" 404 2356 [07/Feb/2026 17:53:33] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:53:34] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:55:20] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 17:55:21] "GET /favicon.ico HTTP/1.1" 404 2356 [07/Feb/2026 17:56:52] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 17:57:19] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2467 [07/Feb/2026 17:58:15] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:59:50] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2467 [07/Feb/2026 18:00:09]

```

[illegible]

variable with valid patterns in the file then the issue is probably caused by a circular import. D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\monsite\urls.py changed, reloading. Watching for file changes with StatReloader Performing system checks... Exception in thread django-main-thread: Traceback (most recent call last): File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\urls\resolvers.py", line 720, in url_patterns iter(patterns) TypeError: 'module' object is not iterable The above exception was the direct cause of the following exception: Traceback (most recent call last): File "C:\Program Files\Python3_12_4\Lib\threading.py", line 1073, in _bootstrap_inner self.run() File "C:\Program Files\Python3_12_4\Lib\threading.py", line 1010, in run self._target(*self._args, **self._kwargs) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\utils\autoreload.py", line 64, in wrapper fn(*args, **kwargs) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\management\commands\runserver.py", line 134, in inner_run self.check(**check_kwargs) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\management\base.py", line 492, in check all_issues = checks.run_checks(File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\checks\registry.py", line 89, in run_checks new_errors = check(app_configs=app_configs, databases=databases) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\checks\urls.py", line 16, in check_url_config return check_resolver(resolver) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\checks\urls.py", line 26, in check_resolver return check_method() File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\urls\resolvers.py", line 531, in check for pattern in self.url_patterns: File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\utils\functional.py", line 47, in __get__ res = instance.__dict__[self.name] = self.func(instance) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\urls\resolvers.py", line 728, in url_patterns raise ImproperlyConfigured(msg.format(name=self.urlconf_name)) from e django.core.exceptions.ImproperlyConfigured: The included URLconf 'monsite.urls' does not appear to have any patterns in it. If you see the 'urlpatterns' variable with valid patterns in the file then the issue is probably caused by a circular import. (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> py manage.py runserver Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 18:26:55 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/> [07/Feb/2026 18:27:08] "GET / HTTP/1.1" 200 776 Not Found: /ws/chat/general/ [07/Feb/2026 18:27:09] "GET /ws/chat/general/ HTTP/1.1" 404 2514 [07/Feb/2026 18:28:54] "GET / HTTP/1.1" 200 776 Not Found: /ws/chat/general/ [07/Feb/2026 18:28:54] "GET /ws/chat/general/ HTTP/1.1" 404 2514 [07/Feb/2026 18:28:57] "GET / HTTP/1.1" 200 776 Not Found: /ws/chat/general/ [07/Feb/2026 18:28:57] "GET /ws/chat/general/ HTTP/1.1" 404 2514 [07/Feb/2026 18:29:05] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:05] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2610 [07/Feb/2026 18:29:06] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:06] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2610 [07/Feb/2026 18:29:07] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:07] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2610 [07/Feb/2026 18:29:07] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:07] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2610 [07/Feb/2026 18:29:07] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:07] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2610 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:08] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2610 [07/Feb/2026 18:29:09] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:09] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404

```
2610 (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> pip install daphne >> ^C (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> pip install daphne Collecting daphne Downloading daphne-4.2.1-py3-none-any.whl.metadata (1.7 kB) Requirement
already satisfied: asgiref<4,>=3.5.2 in d:\python django hopes\chat_api_django_mario\venv\lib\site-packages (from daphne) (3.11.1) Collecting
autobahn>=22.4.2 (from daphne) Downloading autobahn-25.12.2-cp312-cp312-win_amd64.whl.metadata (32 kB) Collecting twisted>=22.4 (from
twisted[tls]>=22.4->daphne) Downloading twisted-25.5.0-py3-none-any.whl.metadata (22 kB) Collecting cbor2>=5.2.0 (from autobahn>=22.4.2->daphne)
Downloading cbor2-5.8.0-cp312-cp312-win_amd64.whl.metadata (5.6 kB) Collecting cffi>=2.0.0 (from autobahn>=22.4.2->daphne) Using cached cffi-2.0.0-
cp312-cp312-win_amd64.whl.metadata (2.6 kB) Collecting cryptography>=3.4.6 (from autobahn>=22.4.2->daphne) Downloading cryptography-46.0.4-cp311-
abi3-win_amd64.whl.metadata (5.7 kB) Collecting hyperlink>=21.0.0 (from autobahn>=22.4.2->daphne) Downloading hyperlink-21.0.0-py2.py3-none-
any.whl.metadata (1.5 kB) Requirement already satisfied: msgpack>=1.0.2 in d:\python django hopes\chat_api_django_mario\venv\lib\site-packages (from
autobahn>=22.4.2->daphne) (1.1.2) Collecting py-ubjson>=0.16.1 (from autobahn>=22.4.2->daphne) Downloading py-ubjson-0.16.1.tar.gz (50 kB)
-----
50.3/50.3 kB 284.2 kB/s eta 0:00:00 Installing build dependencies ...
done Getting requirements to build wheel ... done Preparing metadata (pyproject.toml) ... done Collecting txai>=25.12.2 (from autobahn>=22.4.2->daphne)
Downloading txai-25.12.2-py3-none-any.whl.metadata (7.0 kB) Collecting ujson>=4.0.2 (from autobahn>=22.4.2->daphne) Downloading ujson-5.11.0-
cp312-cp312-win_amd64.whl.metadata (9.6 kB) Collecting attrs>=22.2.0 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading attrs-25.4.0-py3-
none-any.whl.metadata (10 kB) Collecting automat>=24.8.0 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading automat-25.4.16-py3-none-
any.whl.metadata (8.4 kB) Collecting constantly>=15.1 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading constantly-23.10.4-py3-none-
any.whl.metadata (1.8 kB) Collecting incremental>=24.7.0 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading incremental-24.11.0-py3-none-
any.whl.metadata (9.5 kB) Collecting typing-extensions>=4.2.0 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Using cached typing_extensions-4.15.0-
py3-none-any.whl.metadata (3.3 kB) Collecting zope-interface>=5 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading zope_interface-8.2-
cp312-cp312-win_amd64.whl.metadata (46 kB)
-----
46.7/46.7
kB 331.3 kB/s eta 0:00:00 Collecting idna>=2.4 (from twisted[tls]>=22.4->daphne) Downloading idna-3.11-py3-none-any.whl.metadata (8.4 kB) Collecting
pyopenssl>=21.0.0 (from twisted[tls]>=22.4->daphne) Downloading pyopenssl-25.3.0-py3-none-any.whl.metadata (17 kB) Collecting service-identity>=18.1.0
(from twisted[tls]>=22.4->daphne) Downloading service_identity-24.2.0-py3-none-any.whl.metadata (5.1 kB) Collecting pycparser (from cffi>=2.0.0-
>autobahn>=22.4.2->daphne) Downloading pycparser-3.0-py3-none-any.whl.metadata (8.2 kB) Collecting packaging>=17.0 (from incremental>=24.7.0-
>twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading packaging-26.0-py3-none-any.whl.metadata (3.3 kB) Collecting pyasn1 (from service-
identity>=18.1.0->twisted[tls]>=22.4->daphne) Downloading pyasn1-0.6.2-py3-none-any.whl.metadata (8.4 kB) Collecting pyasn1-modules (from service-
identity>=18.1.0->twisted[tls]>=22.4->daphne) Using cached pyasn1_modules-0.4.2-py3-none-any.whl.metadata (3.5 kB) Downloading daphne-4.2.1-py3-
none-any.whl (29 kB) Downloading autobahn-25.12.2-cp312-cp312-win_amd64.whl (2.2 MB)
-----
2.2/2.2 MB 1.2 MB/s eta 0:00:00 Downloading twisted-25.5.0-py3-none-any.whl (3.2 MB)
-----
3.2/3.2 MB 1.5 MB/s eta 0:00:00 Downloading attrs-25.4.0-py3-none-any.whl (67 kB)
-----
67.6/67.6 kB 459.0 kB/s eta 0:00:00 Downloading
automat-25.4.16-py3-none-any.whl (42 kB)
-----
42.8/42.8 kB
299.1 kB/s eta 0:00:00 Downloading cbor2-5.8.0-cp312-cp312-win_amd64.whl (69 kB)
-----
69.8/69.8 kB 546.6 kB/s eta 0:00:00 Using cached cffi-2.0.0-cp312-cp312-win_amd64.whl (183 kB) Downloading constantly-
23.10.4-py3-none-any.whl (13 kB) Downloading cryptography-46.0.4-cp311-abi3-win_amd64.whl (3.5 MB)
-----
3.5/3.5 MB 844.1 kB/s eta 0:00:00 Downloading hyperlink-21.0.0-py2.py3-none-any.whl (74 kB)
-----
```

74.6/74.6 kB 825.6 kB/s eta 0:00:00 Downloading idna-3.11-py3-none-any.whl (71 kB) 71.0/71.0 kB 129.6 kB/s eta 0:00:00
Downloading incremental-24.11.0-py3-none-any.whl (21 kB) Downloading pyopenssl-25.3.0-py3-none-any.whl (57 kB) 57.3/57.3 kB 505.7 kB/s eta 0:00:00 Downloading service_identity-24.2.0-py3-none-any.whl (11 kB) Downloading txai-25.12.2-py3-none-any.whl (31 kB) Using cached typing_extensions-4.15.0-py3-none-any.whl (44 kB) Downloading ujson-5.11.0-cp312-cp312-win_amd64.whl (43 kB) 43.9/43.9 kB 357.6 kB/s eta 0:00:00 Downloading zope_interface-8.2-cp312-cp312-win_amd64.whl (212 kB) 212.4/212.4 kB 1.2 MB/s eta 0:00:00 Downloading packaging-26.0-py3-none-any.whl (74 kB) 74.4/74.4 kB 372.8 kB/s eta 0:00:00 Downloading pyasn1-0.6.2-py3-none-any.whl (83 kB) 83.4/83.4 kB 778.0 kB/s eta 0:00:00 Using cached pyasn1_modules-0.4.2-py3-none-any.whl (181 kB) Downloading pyparser-3.0-py3-none-any.whl (48 kB) 48.2/48.2 kB 403.5 kB/s eta 0:00:00 Building wheels for collected packages: py-ubjson Building wheel for py-ubjson (pyproject.toml) ... done Created wheel for py-ubjson: filename=py_ubjson-0.16.1-cp312-cp312-win_amd64.whl size=19356 sha256=7b3fc1d543eefd1d8e1994a85b47c17a024a1a95741dc0ced87344b32adfb966 Stored in directory: c:\users\spse02men\appdata\local\pip\cache\wheels\c0\87\3e\1931c0c1125bc84ae955eeb3d8ba9fe23ecd95ee337095267 Successfully built py-ubjson
Installing collected packages: py-ubjson, zope-interface, ujson, typing-extensions, txai, pyparser, pyasn1, packaging, idna, constantly, cbor2, automat, attrs, pyasn1-modules, incremental, hyperlink, cffi, twisted, cryptography, service-identity, pyopenssl, Successfully installed attrs-25.4.0 autobahn-25.12.2 automat-25.4.16 cbor2-5.8.0 cffi-2.0.0 constantly-23.10.4 cryptography-46.0.4 daphne-4.2.1 hyperlink-21.0.0 idna-3.11 incremental-24.11.0 packaging-26.0 py-ubjson-0.16.1 pyasn1-0.6.2 pyasn1-modules-0.4.2 pyparser-3.0 pyopenssl-25.3.0 service-identity-24.2.0 twisted-25.5.0 txai-25.12.2 typing-extensions-4.15.0 ujson-5.11.0 zope-interface-8.2 [notice] A new release of pip is available: 24.0 -> 26.0.1 [notice] To update, run: python.exe -m pip install --upgrade pip (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> py manage.py runserver Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 18:34:56 Django version 5.2.8, using settings 'monsite.settings' Starting ASGI/Daphne version 4.2.1 development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. Not Found: /ws/chat/general/ HTTP GET /ws/chat/general/ 404 [0.06, 127.0.0.1:22100] Not Found: /ws/chat/general/ HTTP GET /ws/chat/general/ 404 [0.02, 127.0.0.1:22100] HTTP GET / 200 [0.01, 127.0.0.1:22100] WebSocket HANDSHAKING /ws/chat/general/ [127.0.0.1:22107] WebSocket CONNECT /ws/chat/general/ [127.0.0.1:22107] WebSocket DISCONNECT /ws/chat/general/ [127.0.0.1:22107] HTTP GET / 200 [0.01, 127.0.0.1:22100] WebSocket HANDSHAKING /ws/chat/general/ [127.0.0.1:22110] WebSocket CONNECT /ws/chat/general/ [127.0.0.1:22110] HTTP GET /api/rooms/ 200 [0.01, 127.0.0.1:22239] Not Found: /favicon.ico HTTP GET /favicon.ico 404 [0.03, 127.0.0.1:22240] WebSocket HANDSHAKING /ws/chat/general/ [127.0.0.1:22244] WebSocket CONNECT /ws/chat/general/ [127.0.0.1:22244] WebSocket DISCONNECT /ws/chat/general/ [127.0.0.1:22244] WebSocket HANDSHAKING /ws/chat/general/ [127.0.0.1:22250] WebSocket CONNECT /ws/chat/general/ [127.0.0.1:22250] WebSocket DISCONNECT /ws/chat/general/ [127.0.0.1:22250] WebSocket HANDSHAKING /ws/chat/general/ [127.0.0.1:22252] WebSocket CONNECT /ws/chat/general/ [127.0.0.1:22252] Not Found: /ws/chat/general/ HTTP GET /ws/chat/general/ 404 [0.02, 127.0.0.1:22651] Not Found: /ws/chat/general/ HTTP GET /ws/chat/general/ 404 [0.02, 127.0.0.1:22651] Not Found: /ws/chat/general/ HTTP GET /ws/chat/general/ 404 [0.02, 127.0.0.1:22651] Not Found: /ws/chat/general/ HTTP GET /ws/chat/general/ 404 [0.00, 127.0.0.1:22651] Not Found: /ws/chat/general/ HTTP GET /ws/chat/general/ 404 [0.02, 127.0.0.1:22651] WebSocket DISCONNECT /ws/chat/general/ [127.0.0.1:22252] D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\monsite\asgi.py changed, reloading. Watching for file changes with StatReloader Performing system checks... System

[illegible]


```
[07/Feb/2026 19:44:38] "GET / HTTP/1.1" 200 776 [07/Feb/2026 19:44:38] "GET / HTTP/1.1" 200 776 Not Found: /ws/chat/general/ [07/Feb/2026 19:44:39] "GET /ws/chat/general/ HTTP/1.1" 404 2514 [07/Feb/2026 19:44:57] "GET / HTTP/1.1" 200 776 Not Found: /ws/chat/general/ [07/Feb/2026 19:44:57] "GET /ws/chat/general/ HTTP/1.1" 404 2514 [07/Feb/2026 19:44:57] "GET / HTTP/1.1" 200 776 Not Found: /ws/chat/general/ [07/Feb/2026 19:44:59] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /favicon.ico [07/Feb/2026 19:45:07] "GET /favicon.ico HTTP/1.1" 404 2499 [07/Feb/2026 19:45:09] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /favicon.ico [07/Feb/2026 19:45:09] "GET /favicon.ico HTTP/1.1" 404 2499 Not Found: /ws/chat/general/ [07/Feb/2026 19:52:58] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 19:52:59] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 19:53:00] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 19:53:01] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 19:53:02] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 19:54:12] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 19:54:19] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 19:55:03] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 19:55:39] "GET /ws/chat/general/ HTTP/1.1" 404 2514 D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario\monsite\settings.py changed, reloading. Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 19:55:41 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: https://docs.djangoproject.com/en/5.2/howto/deployment/ Not Found: /ws/chat/general/ [07/Feb/2026 19:55:43] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 19:55:45] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 19:55:47] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Internal Server Error: / Traceback (most recent call last): File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\handlers\Exception.py", line 55, in inner_response = get_response(request) ~~~~~~ File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\handlers\base.py", line 197, in _get_response response = wrapped_callback(request, *callback_args, **callback_kwargs) ~~~~~~ File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\views.py", line 8, in index return render(request, "chat/index.html") ~~~~~~ File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\shortcuts.py", line 25, in render content = loader.render_to_string(template_name, context, request, using=using) ~~~~~~ File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\template\loader.py", line 61, in render_to_string template = get_template(template_name, using=using) ~~~~~~ File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\template\loader.py", line 19, in get_template raise TemplateDoesNotExist(template_name, chain=chain) django.template.exceptions.TemplateDoesNotExist: chat/index.html [07/Feb/2026 20:00:02] "GET / HTTP/1.1" 500 83462 Not Found: /ws/chat/general/ [07/Feb/2026 20:00:47] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 20:00:48] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 20:00:48] "GET /ws/chat/general/ HTTP/1.1" 404 2514 (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> py manage.py runserver Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 20:10:52 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: https://docs.djangoproject.com/en/5.2/howto/deployment/ Internal Server Error: / Traceback (most recent call last): File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\handlers\Exception.py", line 55, in inner_response = get_response(request) ~~~~~~ File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\handlers\base.py", line 197, in
```

```
_get_response response = wrapped_callback(request, *callback_args, **callback_kwargs)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\views.py", line
9, in index return render(request, "chat/index.html") ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ File "D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django.shortcuts.py", line 25, in render content = loader.render_to_string(template_name, context,
request, using=using) ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ File "D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django.template.loader.py", line 61, in render_to_string template = get_template(template_name,
using=using) ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-
packages\django\template\loader.py", line 19, in get_template raise TemplateDoesNotExist(template_name, chain=chain)
django.template.exceptions.TemplateDoesNotExist: chat/index.html [07/Feb/2026 20:11:13] "GET / HTTP/1.1" 500 83128 Not Found: /ws/chat/general/
[07/Feb/2026 20:15:15] "GET /ws/chat/general/ HTTP/1.1" 404 2528 Not Found: /ws/chat/general/ [07/Feb/2026 20:15:17] "GET /ws/chat/general/ HTTP/1.1"
404 2528 Not Found: /ws/chat/general/ [07/Feb/2026 20:15:19] "GET /ws/chat/general/ HTTP/1.1" 404 2528 Not Found: /ws/chat/general/ [07/Feb/2026
20:18:26] "GET /ws/chat/general/ HTTP/1.1" 404 2528 Not Found: /ws/chat/general/ [07/Feb/2026 20:18:27] "GET /ws/chat/general/ HTTP/1.1" 404 2528
D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\urls.py changed, reloading. Watching for file changes with StatReloader Performing system
checks... System check identified no issues (0 silenced). February 07, 2026 - 20:21:41 Django version 5.2.8, using settings 'monsite.settings' Starting
development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production
setting. Use a production WSGI or ASGI server instead. For more information on production servers see:
https://docs.djangoproject.com/en/5.2/howto/deployment/ D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\monsite\urls.py changed, reloading.
Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 20:22:12 Django
version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a
development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see:
https://docs.djangoproject.com/en/5.2/howto/deployment/ Not Found: /ws/chat/general/ [07/Feb/2026 20:22:42] "GET /ws/chat/general/ HTTP/1.1" 404 2528
Not Found: /ws/chat/general/ [07/Feb/2026 20:23:08] "GET /ws/chat/general/ HTTP/1.1" 404 2528
Réflexion durant 1m 14s
```

Pourquoi vous avez GET /ws/chat/general/ 404 ?




Dans vos logs on voit parfois `Starting development server... (mode WSGI)` au lieu de `Starting ASGI/Daphne... (mode ASGI)`.

Vérification rapide

Quand vous lancez :

```
python manage.py runserver
```

vous devez voir :

-  Starting ASGI/Daphne ...
et ensuite, quand votre front se connecte :
-  WebSocket HANDSHAKING /ws/chat/general/
-  WebSocket CONNECT /ws/chat/general/

Note : si vous tapez une URL `ws://...` directement dans le navigateur, il peut faire un **HTTP GET** (donc 404). Ce qui compte c’est la **connexion WebSocket** via `new WebSocket(...)`.

Commandes

- Lancer en mode ASGI (via runserver, grâce à daphne dans INSTALLED_APPS) :

```
python manage.py runserver
```

- Alternative directe :

```
daphne -b 127.0.0.1 -p 8000 monsite.asgi:application
```

Si vous me dites ce que vous voyez exactement dans la console au lancement (la ligne “Starting ... server ...”), je vous dis immédiatement si vous êtes bien en ASGI ou retombé en WSGI.

`venv\Lib\site-packages\redis\multidb\config.py: from_pool (Optional[ConnectionPool]):` A pre-configured connection pool to use. `venv\Lib\site-packages\redis\multidb\config.py:class MultiDbConfig:` `venv\Lib\site-packages\redis\multidb\config.py:` Configuration class for managing multiple database connections in a resilient and fail-safe manner. `venv\Lib\site-packages\redis\multidb\config.py: databases_config:` A list of database configurations. `venv\Lib\site-packages\redis\multidb\config.py:` Retrieves a collection of database clients managed by weighted configurations. `venv\Lib\site-packages\redis\multidb\config.py:` Initializes database clients based on the provided configuration and removes `venv\Lib\site-packages\redis\multidb\config.py:` with defined retry and backoff configurations. `venv\Lib\site-packages\redis\multidb\config.py: databases_config:` `List[DatabaseConfig]` `venv\Lib\site-packages\redis\multidb\config.py:` for `database_config` in `self.databases_config:` `venv\Lib\site-packages\redis\multidb\config.py:`

database_config.client_kwargs.update(venv\Lib\site-packages\redis\multidb\config.py: if database_config.from_url: venv\Lib\site-packages\redis\multidb\config.py: database_config.from_url, **database_config.client_kwargs venv\Lib\site-packages\redis\multidb\config.py: elif database_config.from_pool: venv\Lib\site-packages\redis\multidb\config.py: database_config.from_pool.set_retry(venv\Lib\site-packages\redis\multidb\config.py: connection_pool=database_config.from_pool venv\Lib\site-packages\redis\multidb\config.py: client = self.client_class(**database_config.client_kwargs) venv\Lib\site-packages\redis\multidb\config.py: database_config.default_circuit_breaker() venv\Lib\site-packages\redis\multidb\config.py: if database_config.circuit is None venv\Lib\site-packages\redis\multidb\config.py: else database_config.circuit venv\Lib\site-packages\redis\multidb\config.py: weight=database_config.weight, venv\Lib\site-packages\redis\multidb\config.py: health_check_url=database_config.health_check_url, venv\Lib\site-packages\redis\multidb\config.py: database_config.weight, venv\Lib\site-packages\redis\multidb\healthcheck.py: "Database health check url is not set. Please check DatabaseConfig for the current database." venv\Lib\site-packages\redis_parsers\base.py:# user send an AUTH cmd to a server without authorization configured venv\Lib\site-packages\redis_parsers\base.py: "configured for the default user. Are you sure " venv\Lib\site-packages\redis_parsers\base.py: "your configuration is correct?": AuthenticationError, venv\Lib\site-packages\redis_parsers\helpers.py: "config-epoch": int, venv\Lib\site-packages\redis_parsers\helpers.py:def parse_config_get(response, **options): venv\Lib\site-packages\redis_parsers\helpers.py: "CLUSTER SAVECONFIG": bool_ok, venv\Lib\site-packages\redis_parsers\helpers.py: "CLUSTER SET-CONFIG-EPOCH": bool_ok, venv\Lib\site-packages\redis_parsers\helpers.py: "CONFIG RESETSTAT": bool_ok, venv\Lib\site-packages\redis_parsers\helpers.py: "CONFIG SET": bool_ok, venv\Lib\site-packages\redis_parsers\helpers.py: "SENTINEL FLUSHCONFIG": bool_ok, venv\Lib\site-packages\redis_parsers\helpers.py: "CONFIG GET": parse_config_get, venv\Lib\site-packages\redis_parsers\helpers.py: "CONFIG GET": lambda r: { venv\Lib\site-packages\rest_framework\apps.py:from django.apps import AppConfig venv\Lib\site-packages\rest_framework\apps.py:class RestFrameworkConfig(AppConfig): venv\Lib\site-packages\rest_framework\authtoken\apps.py:from django.apps import AppConfig venv\Lib\site-packages\rest_framework\authtoken\apps.py:class AuthTokenConfig(AppConfig): venv\Lib\site-packages\rest_framework\checks.py:def pagination_system_check(app_configs, **kwargs): venv\Lib\site-packages\rest_framework\compat.py: extension_configs = { venv\Lib\site-packages\rest_framework\compat.py: extensions=extensions, extension_configs=extension_configs venv\Lib\site-packages\rest_framework\fields.py: Quantize the decimal value to the configured precision. venv\Lib\site-packages\rest_framework\filters.py:from django.core.exceptions import FieldDoesNotExist, ImproperlyConfigured venv\Lib\site-packages\rest_framework\filters.py: raise ImproperlyConfigured(msg % self.__class__.__name__) venv\Lib\site-packages\rest_framework\generics.py: method if you want to apply the configured filtering backend to the venv\Lib\site-packages\rest_framework\management\commands\generateschema.py: help = "Generates configured API schema for project." venv\Lib\site-packages\rest_framework\pagination.py: page object, or None if pagination is not configured for this view. venv\Lib\site-packages\rest_framework\relations.py:from django.core.exceptions import ImproperlyConfigured, ObjectDoesNotExist venv\Lib\site-packages\rest_framework\relations.py: attributes are not configured to correctly match the URL conf. venv\Lib\site-packages\rest_framework\relations.py: # Return the hyperlink, or error if incorrectly configured. venv\Lib\site-packages\rest_framework\relations.py: 'model in your API, or incorrectly configured the ' venv\Lib\site-packages\rest_framework\relations.py: raise ImproperlyConfigured(msg % self.view_name) venv\Lib\site-packages\rest_framework\renderers.py:from django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\renderers.py: raise ImproperlyConfigured(venv\Lib\site-packages\rest_framework\routers.py:from django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\routers.py: raise ImproperlyConfigured(msg) venv\Lib\site-packages\rest_framework\routers.py: raise ImproperlyConfigured(msg % ', '.join(not_allowed)) venv\Lib\site-packages\rest_framework\serializers.py:from django.core.exceptions import FieldDoesNotExist, ImproperlyConfigured venv\Lib\site-packages\rest_framework\serializers.py: raise ImproperlyConfigured(venv\Lib\site-packages\rest_framework\settings.py: # View configuration venv\Lib\site-packages\rest_framework\test.py:from django.core.exceptions import ImproperlyConfigured venv\Lib\site-packages\rest_framework\test.py: raise ImproperlyConfigured('requests must be installed in order to use RequestsClient.') venv\Lib\site-packages\rest_framework\test.py: raise

[illegible]

```
404 2305 (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python -m pip show channels >> ^C (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> python -m pip show channels Name: channels Version: 4.1.0 Author: Django Software Foundation License: BSD
Location: D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages Requires: asgiref, Django Required-by: channels-redis (venv)
PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py makemigrations (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> python manage.py migrate Operations to perform: Apply all migrations: admin, auth, chat, contenttypes, sessions
Running migrations: (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py makemigrations (venv) PS D:\PYTHON
DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py migrate Operations to perform: Apply all migrations: admin, auth, chat, contenttypes,
sessions Running migrations: (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py makemigrations No changes
detected (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> python manage.py migrate Operations to perform: Apply all migrations:
admin, auth, chat, contenttypes, sessions Running migrations: No migrations to apply. (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario>
py manage.py runserver Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07,
2026 - 17:03:21 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-
BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more
information on production servers see: https://docs.djangoproject.com/en/5.2/howto/deployment/ Not Found: / [07/Feb/2026 17:03:24] "GET / HTTP/1.1" 404
2305 Not Found: / [07/Feb/2026 17:03:25] "GET / HTTP/1.1" 404 2305 Not Found: / [07/Feb/2026 17:03:27] "GET / HTTP/1.1" 404 2305 [07/Feb/2026
17:04:21] "GET /admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:04:22] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 4252 [07/Feb/2026 17:04:22] "GET
/static/admin/css/nav_sidebar.css HTTP/1.1" 200 2810 [07/Feb/2026 17:04:22] "GET /static/admin/css/base.css HTTP/1.1" 200 22120 [07/Feb/2026 17:04:22]
"GET /static/admin/css/login.css HTTP/1.1" 200 951 [07/Feb/2026 17:04:22] "GET /static/admin/css/dark_mode.css HTTP/1.1" 200 2808 [07/Feb/2026
17:04:22] "GET /static/admin/css/responsive.css HTTP/1.1" 200 16565 [07/Feb/2026 17:04:22] "GET /static/admin/js/theme.js HTTP/1.1" 200 1653
[07/Feb/2026 17:04:22] "GET /static/admin/js/nav_sidebar.js HTTP/1.1" 200 3063 Not Found: /favicon.ico [07/Feb/2026 17:04:22] "GET /favicon.ico
HTTP/1.1" 404 2356 Not Found: /api/ [07/Feb/2026 17:04:37] "GET /api/ HTTP/1.1" 404 2739 Not Found: / [07/Feb/2026 17:04:48] "GET / HTTP/1.1" 404
2305 [07/Feb/2026 17:05:01] "GET /admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:05:01] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 4252 Not Found:
/api/ [07/Feb/2026 17:05:12] "GET /api/ HTTP/1.1" 404 2739 Not Found: / [07/Feb/2026 17:24:49] "GET /api/rooms/ HTTP/1.1" 200 2 Not Found:
/favicon.ico [07/Feb/2026 17:24:49] "GET /favicon.ico HTTP/1.1" 404 2356 (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> curl -X
POST http://127.0.0.1:8000/api/rooms/ -H "Content-Type: application/json" -d '{"name":"general"}' >> ^C (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> curl -X POST http://127.0.0.1:8000/api/rooms/ -H "Content-Type: application/json" -d '{"name":"general"}' Invoke-
WebRequest : Impossible de lier le paramètre «Headers». Impossible de convertir la valeur «Content-Type: application/json» du type «System.String» en type
«System.Collections.IDictionary». Au caractère Ligne:1 : 50 + ~~~~~ + CategoryInfo : InvalidArgument : (:) [Invoke-
WebRequest], ParameterBindingException + FullyQualifiedErrorId :
CannotConvertArgumentNoMessage,Microsoft.PowerShell.Commands.InvokeWebRequestCommand (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/" >> -Method Post >> -ContentType "application/json"
Invoke-RestMethod : Impossible de se connecter au serveur distant Au caractère Ligne:1 : 1 + Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/" +
~~~~~ + CategoryInfo : NotSpecified: (:) [Invoke-RestMethod], WebException +
FullyQualifiedErrorId : System.Net.WebException,Microsoft.PowerShell.Commands.InvokeRestMethodCommand (venv) PS D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario> (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> py manage.py runserver Watching for file
changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 17:30:20 Django version 5.2.8,
using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development
```

server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see:

<https://docs.djangoproject.com/en/5.2/howto/deployment/> Not Found: / [07/Feb/2026 17:30:24] "GET / HTTP/1.1" 404 2305 Not Found: / [07/Feb/2026 17:30:25] "GET / HTTP/1.1" 404 2305 Not Found: /ws/chat/general/ [07/Feb/2026 17:31:01] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:31:04] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:31:24] "GET /api/rooms/ HTTP/1.1" 200 2 Not Found: /ws/chat/general/ [07/Feb/2026 17:31:32] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:31:58] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:32:00] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:32:02] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:32:04] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:21] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:22] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:22] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:32] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:33] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:35:35] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /api/ \u2022 WS: ws://127.0.0.1:8000/ws/chat/<room>/ [07/Feb/2026 17:35:44] "GET /api/%20%E2%80%A2%20WS:%20ws://127.0.0.1:8000/ws/chat/%3Croom%3E/ HTTP/1.1" 404 2899 Not Found: /api/ [07/Feb/2026 17:35:51] "GET /api/ HTTP/1.1" 404 2739 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:43] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:44] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:44] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:45] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:46] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:36:47] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:38:30] "GET /api/rooms/ HTTP/1.1" 200 2 Not Found: /favicon.ico [07/Feb/2026 17:38:30] "GET /favicon.ico HTTP/1.1" 404 2356 Not Found: / [07/Feb/2026 17:38:52] "GET / HTTP/1.1" 404 2305 [07/Feb/2026 17:41:00] "GET /admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:41:00] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 4252 [07/Feb/2026 17:41:01] "GET /static/admin/css/dark_mode.css HTTP/1.1" 200 2808 [07/Feb/2026 17:41:01] "GET /static/admin/css/base.css HTTP/1.1" 200 22120 [07/Feb/2026 17:41:01] "GET /static/admin/css/nav_sidebar.css HTTP/1.1" 200 2810 [07/Feb/2026 17:41:01] "GET /static/admin/js/theme.js HTTP/1.1" 200 1653 [07/Feb/2026 17:41:01] "GET /static/admin/js/nav_sidebar.js HTTP/1.1" 200 3063 [07/Feb/2026 17:41:01] "GET /static/admin/css/login.css HTTP/1.1" 200 951 [07/Feb/2026 17:41:01] "GET /static/admin/css/responsive.css HTTP/1.1" 200 16565 [07/Feb/2026 17:41:35] "POST /admin/login/?next=/admin/ HTTP/1.1" 200 4498 [07/Feb/2026 17:44:04] "POST /admin/login/?next=/admin/ HTTP/1.1" 302 0 [07/Feb/2026 17:44:04] "GET /admin/ HTTP/1.1" 200 7885 [07/Feb/2026 17:44:05] "GET /static/admin/css/dashboard.css HTTP/1.1" 200 441 [07/Feb/2026 17:44:05] "GET /static/admin/img/icon-changelink.svg HTTP/1.1" 200 380 [07/Feb/2026 17:44:05] "GET /static/admin/img/icon-addlink.svg HTTP/1.1" 200 331 [07/Feb/2026 17:44:27] "GET /admin/chat/room/add/ HTTP/1.1" 200 9637 [07/Feb/2026 17:44:27] "GET /static/admin/js/jquery.init.js HTTP/1.1" 200 347 [07/Feb/2026 17:44:27] "GET /static/admin/css/forms.css HTTP/1.1" 200 8525 [07/Feb/2026 17:44:27] "GET /static/admin/js/core.js HTTP/1.1" 200 6208 [07/Feb/2026 17:44:27] "GET /static/admin/js/urlify.js HTTP/1.1" 200 7887 [07/Feb/2026 17:44:27] "GET /static/admin/js/prepopulate.js HTTP/1.1" 200 1531 [07/Feb/2026 17:44:27] "GET /admin/jsi18n/ HTTP/1.1" 200 8634 [07/Feb/2026 17:44:27] "GET /static/admin/js/change_form.js HTTP/1.1" 200 606 [07/Feb/2026 17:44:27] "GET /static/admin/js/admin/RelatedObjectLookups.js HTTP/1.1" 200 9777 [07/Feb/2026 17:44:27] "GET /static/admin/js/actions.js HTTP/1.1" 200 8076 [07/Feb/2026 17:44:27] "GET /static/admin/js/prepopulate_init.js HTTP/1.1" 200 586 [07/Feb/2026 17:44:27] "GET /static/admin/css/widgets.css HTTP/1.1" 200 11973 [07/Feb/2026 17:44:28] "GET /static/admin/js/vendor/xregexp/xregexp.js HTTP/1.1" 200 325171 [07/Feb/2026 17:44:28] "GET /static/admin/js/vendor/jquery/jquery.js HTTP/1.1" 200 285314 [07/Feb/2026 17:44:44] "POST /admin/chat/room/add/ HTTP/1.1" 302 0 [07/Feb/2026 17:44:44] "GET /admin/chat/room/ HTTP/1.1" 200 10874 [07/Feb/2026 17:44:45] "GET /admin/jsi18n/ HTTP/1.1" 200 8634 [07/Feb/2026 17:44:45] "GET /static/admin/css/changelists.css HTTP/1.1" 200 6878 [07/Feb/2026 17:44:45] "GET /static/admin/js/filters.js HTTP/1.1" 200 978 [07/Feb/2026 17:44:45] "GET /static/admin/img/search.svg HTTP/1.1" 200 458

[07/Feb/2026 17:44:45] "GET /static/admin/img/tooltag-add.svg HTTP/1.1" 200 331 [07/Feb/2026 17:44:45] "GET /static/admin/img/icon-yes.svg HTTP/1.1" 200 436 Not Found: / [07/Feb/2026 17:44:58] "GET / HTTP/1.1" 404 2305 Not Found: / [07/Feb/2026 17:44:59] "GET / HTTP/1.1" 404 2305 Not Found: /ws/chat/general/ [07/Feb/2026 17:45:18] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:45:22] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:06] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:08] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:09] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:38] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:39] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:41] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:52] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:47:53] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 17:47:53] "GET /api/rooms/1/messages/ HTTP/1.1" 200 2 [07/Feb/2026 17:47:54] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 17:47:54] "GET /api/rooms/1/messages/ HTTP/1.1" 200 2 Not Found: /ws/chat/general/ [07/Feb/2026 17:47:57] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:50:18] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /favicon.ico [07/Feb/2026 17:50:18] "GET /favicon.ico HTTP/1.1" 404 2356 [07/Feb/2026 17:50:21] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /favicon.ico [07/Feb/2026 17:50:21] "GET /favicon.ico HTTP/1.1" 404 2356 Not Found: /ws/chat/general/ [07/Feb/2026 17:53:33] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:53:34] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 17:53:34] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 17:55:20] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /favicon.ico [07/Feb/2026 17:55:21] "GET /favicon.ico HTTP/1.1" 404 2356 [07/Feb/2026 17:56:52] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 17:57:19] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2467 Not Found: /ws/chat/general/ [07/Feb/2026 17:58:15] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 17:59:50] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2467 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:09] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:11] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:12] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:18] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:25] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:25] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:26] "GET /ws/chat/general/ HTTP/1.1" 404 2371 [07/Feb/2026 18:00:27] "GET /api/rooms/ HTTP/1.1" 200 27 [07/Feb/2026 18:00:27] "GET /api/rooms/1/messages/ HTTP/1.1" 200 2 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:28] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:00:29] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:07:52] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:07:53] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:08:00] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:08:00] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:08:01] "GET /ws/chat/general/ HTTP/1.1" 404 2371 Not Found: /ws/chat/general/ [07/Feb/2026 18:08:28] "GET /ws/chat/general/ HTTP/1.1" 404 2371 D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\views.py changed, reloading. Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 18:24:04 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/> D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\views.py changed, reloading. Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 18:24:10 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see:

<https://docs.djangoproject.com/en/5.2/howto/deployment/> D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\monsite\urls.py changed, reloading. Watching for file changes with StatReloader Performing system checks... Exception in thread django-main-thread: Traceback (most recent call last): File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\urls\resolvers.py", line 720, in url_patterns iter(patterns) TypeError: 'module' object is not iterable The above exception was the direct cause of the following exception: Traceback (most recent call last): File "C:\Program Files\Python3_12_4\Lib\threading.py", line 1073, in _bootstrap_inner self.run() File "C:\Program Files\Python3_12_4\Lib\threading.py", line 1010, in run self._target(*self._args, **self._kwargs) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\utils\autoreload.py", line 64, in wrapper fn(*args, **kwargs) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\management\commands\runserver.py", line 134, in inner_run self.check(**check_kwargs) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\management\base.py", line 492, in check_all_issues = checks.run_checks(File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\checks\registry.py", line 89, in run_checks new_errors = check(app_configs=app_configs, databases=databases) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\checks\urls.py", line 16, in check_url_config return check_resolver(resolver) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\checks\urls.py", line 26, in check_resolver return check_method() File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\urls\resolvers.py", line 531, in check for pattern in self.url_patterns: File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\utils\functional.py", line 47, in __get__ res = instance.__dict__[self.name] = self.func(instance) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\urls\resolvers.py", line 728, in url_patterns raise ImproperlyConfigured(msg.format(name=self.urlconf_name)) from e django.core.exceptions.ImproperlyConfigured: The included URLconf 'monsite.urls' does not appear to have any patterns in it. If you see the 'urlpatterns' variable with valid patterns in the file then the issue is probably caused by a circular import. D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\monsite\urls.py changed, reloading. Watching for file changes with StatReloader Performing system checks... Exception in thread django-main-thread: Traceback (most recent call last): File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\urls\resolvers.py", line 720, in url_patterns iter(patterns) TypeError: 'module' object is not iterable The above exception was the direct cause of the following exception: Traceback (most recent call last): File "C:\Program Files\Python3_12_4\Lib\threading.py", line 1073, in _bootstrap_inner self.run() File "C:\Program Files\Python3_12_4\Lib\threading.py", line 1010, in run self._target(*self._args, **self._kwargs) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\utils\autoreload.py", line 64, in wrapper fn(*args, **kwargs) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\management\commands\runserver.py", line 134, in inner_run self.check(**check_kwargs) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\management\base.py", line 492, in check_all_issues = checks.run_checks(File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\checks\registry.py", line 89, in run_checks new_errors = check(app_configs=app_configs, databases=databases) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\checks\urls.py", line 16, in check_url_config return check_resolver(resolver) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\checks\urls.py", line 26, in check_resolver return check_method() File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\urls\resolvers.py", line 531, in check for pattern in self.url_patterns: File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\utils\functional.py", line 47, in __get__ res = instance.__dict__[self.name] = self.func(instance) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\urls\resolvers.py", line 728, in url_patterns raise ImproperlyConfigured(msg.format(name=self.urlconf_name)) from e

django.core.exceptions.ImproperlyConfigured: The included URLconf 'monsite.urls' does not appear to have any patterns in it. If you see the 'urlpatterns' variable with valid patterns in the file then the issue is probably caused by a circular import. (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> py manage.py runserver Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 18:26:55 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/> [07/Feb/2026 18:27:08] "GET / HTTP/1.1" 200 776 Not Found: /ws/chat/general/ [07/Feb/2026 18:27:09] "GET /ws/chat/general/ HTTP/1.1" 404 2514 [07/Feb/2026 18:28:54] "GET / HTTP/1.1" 200 776 Not Found: /ws/chat/general/ [07/Feb/2026 18:28:54] "GET /ws/chat/general/ HTTP/1.1" 404 2514 [07/Feb/2026 18:28:57] "GET / HTTP/1.1" 200 776 Not Found: /ws/chat/general/ [07/Feb/2026 18:28:57] "GET /ws/chat/general/ HTTP/1.1" 404 2514 [07/Feb/2026 18:29:05] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:05] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2610 [07/Feb/2026 18:29:06] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:06] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2610 [07/Feb/2026 18:29:07] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:07] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2610 [07/Feb/2026 18:29:07] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:07] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2610 [07/Feb/2026 18:29:07] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:07] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2610 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:08] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2610 [07/Feb/2026 18:29:09] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /.well-known/appspecific/com.chrome.devtools.json [07/Feb/2026 18:29:09] "GET /.well-known/appspecific/com.chrome.devtools.json HTTP/1.1" 404 2610 (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> pip install daphne >> ^C (venv) PS D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> pip install daphne Collecting daphne Downloading daphne-4.2.1-py3-none-any.whl.metadata (1.7 kB) Requirement already satisfied: asgiref<4,>=3.5.2 in d:\python django hopes\chat_api_django_mario\venv\lib\site-packages (from daphne) (3.11.1) Collecting autobahn>=22.4.2 (from daphne) Downloading autobahn-25.12.2-cp312-cp312-win_amd64.whl.metadata (32 kB) Collecting twisted>=22.4 (from twisted[tls]>=22.4->daphne) Downloading twisted-25.5.0-py3-none-any.whl.metadata (22 kB) Collecting cbor2>=5.2.0 (from autobahn>=22.4.2->daphne) Downloading cbor2-5.8.0-cp312-cp312-win_amd64.whl.metadata (5.6 kB) Collecting cffi>=2.0.0 (from autobahn>=22.4.2->daphne) Using cached cffi-2.0.0-cp312-cp312-win_amd64.whl.metadata (2.6 kB) Collecting cryptography>=3.4.6 (from autobahn>=22.4.2->daphne) Downloading cryptography-46.0.4-cp311-abi3-win_amd64.whl.metadata (5.7 kB) Collecting hyperlink>=21.0.0 (from autobahn>=22.4.2->daphne) Downloading hyperlink-21.0.0-py2.py3-none-any.whl.metadata (1.5 kB) Requirement already satisfied: msgpack>=1.0.2 in d:\python django hopes\chat_api_django_mario\venv\lib\site-packages (from autobahn>=22.4.2->daphne) (1.1.2) Collecting py-ubjson>=0.16.1 (from autobahn>=22.4.2->daphne) Downloading py-ubjson-0.16.1.tar.gz (50 kB)

50.3/50.3 kB 284.2 kB/s eta 0:00:00 Installing build dependencies ... done Getting requirements to build wheel ... done Preparing metadata (pyproject.toml) ... done Collecting txai0>=25.12.2 (from autobahn>=22.4.2->daphne) Downloading txai0-25.12.2-py3-none-any.whl.metadata (7.0 kB) Collecting ujson>=4.0.2 (from autobahn>=22.4.2->daphne) Downloading ujson-5.11.0-cp312-cp312-win_amd64.whl.metadata (9.6 kB) Collecting attrs>=22.2.0 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading attrs-25.4.0-py3-none-any.whl.metadata (10 kB) Collecting automat>=24.8.0 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading automat-25.4.16-py3-none-any.whl.metadata (8.4 kB) Collecting constantly>=15.1 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading constantly-23.10.4-py3-none-any.whl.metadata (1.8 kB) Collecting incremental>=24.7.0 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading incremental-24.11.0-py3-none-any.whl.metadata (9.5 kB) Collecting typing-extensions>=4.2.0 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Using cached typing_extensions-4.15.0-

py3-none-any.whl.metadata (3.3 kB) Collecting zope-interface>=5 (from twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading zope_interface-8.2-cp312-cp312-win_amd64.whl.metadata (46 kB) 46.7/46.7 kB 331.3 kB/s eta 0:00:00 Collecting idna>=2.4 (from twisted[tls]>=22.4->daphne) Downloading idna-3.11-py3-none-any.whl.metadata (8.4 kB) Collecting pyopenssl>=21.0.0 (from twisted[tls]>=22.4->daphne) Downloading pyopenssl-25.3.0-py3-none-any.whl.metadata (17 kB) Collecting service-identity>=18.1.0 (from twisted[tls]>=22.4->daphne) Downloading service_identity-24.2.0-py3-none-any.whl.metadata (5.1 kB) Collecting pycparser (from cffi>=2.0.0->autobahn>=22.4.2->daphne) Downloading pycparser-3.0-py3-none-any.whl.metadata (8.2 kB) Collecting packaging>=17.0 (from incremental>=24.7.0->twisted>=22.4->twisted[tls]>=22.4->daphne) Downloading packaging-26.0-py3-none-any.whl.metadata (3.3 kB) Collecting pyasn1 (from service-identity>=18.1.0->twisted[tls]>=22.4->daphne) Downloading pyasn1-0.6.2-py3-none-any.whl.metadata (8.4 kB) Collecting pyasn1-modules (from service-identity>=18.1.0->twisted[tls]>=22.4->daphne) Using cached pyasn1_modules-0.4.2-py3-none-any.whl.metadata (3.5 kB) Downloading daphne-4.2.1-py3-none-any.whl (29 kB) Downloading autobahn-25.12.2-cp312-cp312-win_amd64.whl (2.2 MB) 2.2/2.2 MB 1.2 MB/s eta 0:00:00 Downloading twisted-25.5.0-py3-none-any.whl (3.2 MB) 3.2/3.2 MB 1.5 MB/s eta 0:00:00 Downloading attrs-25.4.0-py3-none-any.whl (67 kB) 67.6/67.6 kB 459.0 kB/s eta 0:00:00 Downloading automat-25.4.16-py3-none-any.whl (42 kB) 42.8/42.8 kB 299.1 kB/s eta 0:00:00 Downloading cbor2-5.8.0-cp312-cp312-win_amd64.whl (69 kB) 69.8/69.8 kB 546.6 kB/s eta 0:00:00 Using cached cffi-2.0.0-cp312-cp312-win_amd64.whl (183 kB) Downloading constantly-23.10.4-py3-none-any.whl (13 kB) Downloading cryptography-46.0.4-cp311-abi3-win_amd64.whl (3.5 MB) 3.5/3.5 MB 844.1 kB/s eta 0:00:00 Downloading hyperlink-21.0.0-py2.py3-none-any.whl (74 kB) 74.6/74.6 kB 825.6 kB/s eta 0:00:00 Downloading idna-3.11-py3-none-any.whl (71 kB) 71.0/71.0 kB 129.6 kB/s eta 0:00:00 Downloading incremental-24.11.0-py3-none-any.whl (21 kB) Downloading pyopenssl-25.3.0-py3-none-any.whl (57 kB) 57.3/57.3 kB 505.7 kB/s eta 0:00:00 Downloading service_identity-24.2.0-py3-none-any.whl (11 kB) Downloading txai-25.12.2-py3-none-any.whl (31 kB) Using cached typing_extensions-4.15.0-py3-none-any.whl (44 kB) Downloading ujson-5.11.0-cp312-cp312-win_amd64.whl (43 kB) 43.9/43.9 kB 357.6 kB/s eta 0:00:00 Downloading zope_interface-8.2-cp312-cp312-win_amd64.whl (212 kB) 212.4/212.4 kB 1.2 MB/s eta 0:00:00 Downloading packaging-26.0-py3-none-any.whl (74 kB) 74.4/74.4 kB 372.8 kB/s eta 0:00:00 Downloading pyasn1-0.6.2-py3-none-any.whl (83 kB) 83.4/83.4 kB 778.0 kB/s eta 0:00:00 Using cached pyasn1_modules-0.4.2-py3-none-any.whl (181 kB) Downloading pycparser-3.0-py3-none-any.whl (48 kB) 48.2/48.2 kB 403.5 kB/s eta 0:00:00 Building wheels for collected packages: py-ubjson Building wheel for py-ubjson (pyproject.toml) ... done Created wheel for py-ubjson: filename=py_ubjson-0.16.1-cp312-cp312-win_amd64.whl size=19356 sha256=7b3fc1d543eefd1d8e1994a85b47c17a024a1a95741dc0ced87344b32adfb966 Stored in directory: c:\users\spsse02men\appdata\local\pip\cache\wheels\c0\8\73\1e1931c0c1125bc84ae955eeb3d8ba9fe23ecd95ee337095267 Successfully built py-ubjson Installing collected packages: py-ubjson, zope-interface, ujson, typing-extensions, txai, pycparser, pyasn1, packaging, idna, constantly, cbor2, automat, attrs, pyasn1-modules, incremental, hyperlink, cffi, twisted, cryptography, service-identity, pyopenssl, Successfully installed attrs-25.4.0 autobahn-25.12.2 automat-

[illegible]

system checks... System check identified no issues (0 silenced). February 07, 2026 - 19:21:52 Django version 5.2.8, using settings 'monsite.settings' Starting ASGI/Daphne version 4.2.1 development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. HTTP GET /api/rooms/ 200 [0.02, 127.0.0.1:23635] Not Found: /favicon.ico HTTP GET /favicon.ico 404 [0.05, 127.0.0.1:23636] HTTP GET / 200 [0.01, 127.0.0.1:23735] WebSocket HANDSHAKING /ws/chat/general/ [127.0.0.1:23736] WebSocket CONNECT /ws/chat/general/ [127.0.0.1:23736] Not Found: /ws/chat/general/ HTTP GET /ws/chat/general/ 404 [0.02, 127.0.0.1:23735] Not Found: /ws/chat/general/ HTTP GET /ws/chat/general/ 404 [0.02, 127.0.0.1:23804] HTTP GET / 200 [0.01, 127.0.0.1:23836] WebSocket HANDSHAKING /ws/chat/general/ [127.0.0.1:23837] WebSocket CONNECT /ws/chat/general/ [127.0.0.1:23837] HTTP GET / 200 [0.01, 127.0.0.1:23836] WebSocket HANDSHAKING /ws/chat/general/ [127.0.0.1:23841] WebSocket CONNECT /ws/chat/general/ [127.0.0.1:23841] WebSocket DISCONNECT /ws/chat/general/ [127.0.0.1:23841] HTTP GET / 200 [0.02, 127.0.0.1:23836] WebSocket HANDSHAKING /ws/chat/general/ [127.0.0.1:23861] WebSocket CONNECT /ws/chat/general/ [127.0.0.1:23861] D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\serializers.py changed, reloading. Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 19:40:09 Django version 5.2.8, using settings 'monsite.settings' Starting ASGI/Daphne version 4.2.1 development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\monsite\settings.py changed, reloading. Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 19:43:10 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/> Not Found: /ws/chat/general/ [07/Feb/2026 19:44:11] "GET /ws/chat/general/ HTTP/1.1" 404 2514 [07/Feb/2026 19:44:35] "GET / HTTP/1.1" 200 776 Not Found: /ws/chat/general/ [07/Feb/2026 19:44:36] "GET /ws/chat/general/ HTTP/1.1" 404 2514 [07/Feb/2026 19:44:37] "GET / HTTP/1.1" 200 776 Not Found: /ws/chat/general/ [07/Feb/2026 19:44:37] "GET /ws/chat/general/ HTTP/1.1" 404 2514 [07/Feb/2026 19:44:38] "GET / HTTP/1.1" 200 776 Not Found: /ws/chat/general/ [07/Feb/2026 19:44:38] "GET /ws/chat/general/ HTTP/1.1" 404 2514 [07/Feb/2026 19:44:38] "GET / HTTP/1.1" 200 776 [07/Feb/2026 19:44:38] "GET / HTTP/1.1" 200 776 Not Found: /ws/chat/general/ [07/Feb/2026 19:44:39] "GET /ws/chat/general/ HTTP/1.1" 404 2514 [07/Feb/2026 19:44:57] "GET / HTTP/1.1" 200 776 Not Found: /ws/chat/general/ [07/Feb/2026 19:44:57] "GET /ws/chat/general/ HTTP/1.1" 404 2514 [07/Feb/2026 19:44:57] "GET / HTTP/1.1" 200 776 Not Found: /ws/chat/general/ [07/Feb/2026 19:44:59] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /favicon.ico [07/Feb/2026 19:45:07] "GET /favicon.ico HTTP/1.1" 404 2499 [07/Feb/2026 19:45:09] "GET /api/rooms/ HTTP/1.1" 200 27 Not Found: /favicon.ico [07/Feb/2026 19:45:09] "GET /favicon.ico HTTP/1.1" 404 2499 Not Found: /ws/chat/general/ [07/Feb/2026 19:52:58] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 19:52:59] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 19:53:00] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 19:53:01] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 19:53:02] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 19:54:12] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 19:54:19] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 19:55:03] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 19:55:39] "GET /ws/chat/general/ HTTP/1.1" 404 2514 D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\monsite\settings.py changed, reloading. Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 19:55:41 Django version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/> Not Found: /ws/chat/general/ [07/Feb/2026 19:55:43] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 19:55:45] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 19:55:47] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Internal Server Error: / Traceback (most recent call last): File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-

```
packages\django\core\handlers\exception.py", line 55, in inner response = get_response(request) File "D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\handlers\base.py", line 197, in _get_response response = wrapped_callback(request,
*callback_args, **callback_kwargs) File "D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario\chat\views.py", line 8, in index return render(request, "chat/index.html") File
"D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\shortcuts.py", line 25, in render content =
loader.render_to_string(template_name, context, request, using=using) File
"D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\template\loader.py", line 61, in render_to_string template =
get_template(template_name, using=using) File "D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\template\loader.py", line 19, in get_template raise TemplateDoesNotExist(template_name,
chain=chain) django.template.exceptions.TemplateDoesNotExist: chat/index.html [07/Feb/2026 20:00:02] "GET / HTTP/1.1" 500 83462 Not Found:
/ws/chat/general/ [07/Feb/2026 20:00:47] "GET /ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 20:00:48] "GET
/ws/chat/general/ HTTP/1.1" 404 2514 Not Found: /ws/chat/general/ [07/Feb/2026 20:00:48] "GET /ws/chat/general/ HTTP/1.1" 404 2514 (venv) PS
D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario> py manage.py runserver Watching for file changes with StatReloader Performing system
checks... System check identified no issues (0 silenced). February 07, 2026 - 20:10:52 Django version 5.2.8, using settings 'monsite.settings' Starting
development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production
setting. Use a production WSGI or ASGI server instead. For more information on production servers see:
https://docs.djangoproject.com/en/5.2/howto/deployment/ Internal Server Error: / Traceback (most recent call last): File "D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\handlers\exception.py", line 55, in inner response = get_response(request)
File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\core\handlers\base.py", line 197, in
_get_response response = wrapped_callback(request, *callback_args, **callback_kwargs)
File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\views.py", line
9, in index return render(request, "chat/index.html") File "D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\shortcuts.py", line 25, in render content = loader.render_to_string(template_name, context,
request, using=using) File "D:\PYTHON DJANGO
HOPES\Chat_API_DJANGO_mario\venv\Lib\site-packages\django\template\loader.py", line 61, in render_to_string template = get_template(template_name,
using=using) File "D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\venv\Lib\site-
packages\django\template\loader.py", line 19, in get_template raise TemplateDoesNotExist(template_name, chain=chain)
django.template.exceptions.TemplateDoesNotExist: chat/index.html [07/Feb/2026 20:11:13] "GET / HTTP/1.1" 500 83128 Not Found: /ws/chat/general/
[07/Feb/2026 20:15:15] "GET /ws/chat/general/ HTTP/1.1" 404 2528 Not Found: /ws/chat/general/ [07/Feb/2026 20:15:17] "GET /ws/chat/general/ HTTP/1.1"
404 2528 Not Found: /ws/chat/general/ [07/Feb/2026 20:15:19] "GET /ws/chat/general/ HTTP/1.1" 404 2528 Not Found: /ws/chat/general/ [07/Feb/2026
20:18:26] "GET /ws/chat/general/ HTTP/1.1" 404 2528 Not Found: /ws/chat/general/ [07/Feb/2026 20:18:27] "GET /ws/chat/general/ HTTP/1.1" 404 2528
D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\urls.py changed, reloading. Watching for file changes with StatReloader Performing system
checks... System check identified no issues (0 silenced). February 07, 2026 - 20:21:41 Django version 5.2.8, using settings 'monsite.settings' Starting
development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production
setting. Use a production WSGI or ASGI server instead. For more information on production servers see:
https://docs.djangoproject.com/en/5.2/howto/deployment/ D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\monsite\urls.py changed, reloading.
Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 20:22:12 Django
```

version 5.2.8, using settings 'monsite.settings' Starting development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead. For more information on production servers see: <https://docs.djangoproject.com/en/5.2/howto/deployment/> Not Found: /ws/chat/general/ [07/Feb/2026 20:22:42] "GET /ws/chat/general/ HTTP/1.1" 404 2528 Not Found: /ws/chat/general/ [07/Feb/2026 20:23:08] "GET /ws/chat/general/ HTTP/1.1" 404 2528 D:\PYTHON DJANGO HOPES\Chat_API_DJANGO_mario\chat\serializers.py changed, reloading. Watching for file changes with StatReloader Performing system checks... System check identified no issues (0 silenced). February 07, 2026 - 20:26:50 Django version 5.2.8, using settings 'monsite.settings' Starting ASGI/Daphne version 4.2.1 development server at http://127.0.0.1:8000/ Quit the server with CTRL-BREAK. WebSocket HANDSHAKING /ws/chat/general/ [127.0.0.1:25685] WebSocket CONNECT /ws/chat/general/ [127.0.0.1:25685] WebSocket DISCONNECT /ws/chat/general/ [127.0.0.1:25685] WebSocket HANDSHAKING /ws/chat/general/ [127.0.0.1:25698] WebSocket CONNECT /ws/chat/general/ [127.0.0.1:25698]

OK là tu as fait **le plus dur** 📌

Tes logs montrent exactement ce qui se passe :

1) Quand tu vois “Starting ASGI/Daphne ...”



WebSockets OK (on voit `WebSocket HANDSHAKING` puis `CONNECT`)

Donc **Channels + routing fonctionnent**.

2) Pourquoi tu as encore `HTTP GET /ws/chat/general/ 404` parfois ?

Parce que ce n'est pas un **WebSocket**, c'est un **HTTP GET** sur l'URL `/ws/...`

Ça arrive quand :

- tu tapes `/ws/chat/general/` dans le navigateur → ça fait un GET HTTP
- ton front utilise `fetch("http://.../ws/...")` au lieu de `new WebSocket("ws://...")`
- ou tu as un outil / extension qui ping cette URL en HTTP



Conclusion : un GET HTTP sur `/ws/...` retournera souvent **404** (normal).

Ce qui compte c'est : `WebSocket HANDSHAKING/CONNECT`.

Le vrai problème qui te reste : tu retombes parfois en “Starting development server ...”

À plusieurs reprises tu repasses en WSGI (Django runserver normal), donc WS cassé et tu vois 404.

Ça arrive typiquement quand :

- tu as retiré daphne de `INSTALLED_APPS`
- ou tu as une erreur dans `ASGI_APPLICATION` / `asgi.py` et Django tombe sur WSGI
- ou tu as mis `DEBUG=False` sans `ALLOWED_HOSTS` (tu l’as eu)

✅ La règle : si c’est ASGI, la ligne doit dire ASGI/Daphne.

Fix ultra clair (copie/colle)

A) `monsite/settings.py`

Assure-toi d’avoir ça :

```
INSTALLED_APPS = [  
    "daphne",                # IMPORTANT: doit être avant django.contrib.staticfiles  
    "django.contrib.admin",  
    "django.contrib.auth",  
    "django.contrib.contenttypes",  
    "django.contrib.sessions",  
    "django.contrib.messages",  
    "django.contrib.staticfiles",  
  
    "rest_framework",  
    "channels",  
    "chat",  
]  
  
ASGI_APPLICATION = "monsite.asgi.application"
```



```
CHANNEL_LAYERS = {
    "default": {
        "BACKEND": "channels.layers.InMemoryChannelLayer",
    }
}

DEBUG = True
ALLOWED_HOSTS = ["127.0.0.1", "localhost"]
```

Quand tu voudras Redis plus tard on change CHANNEL_LAYERS, mais pour l'instant **InMemory** suffit.

B) monsite/asgi.py

Ton erreur Cannot find 'application' in ASGI_APPLICATION module monsite.asgi vient de là : ton fichier a été modifié et ne contenait plus application.

Il doit ressembler à ça :

```
import os
from django.core.asgi import get_asgi_application
from channels.routing import ProtocolTypeRouter, URLRouter
from channels.auth import AuthMiddlewareStack
import chat.routing

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "monsite.settings")

django_asgi_app = get_asgi_application()

application = ProtocolTypeRouter({
    "http": django_asgi_app,
    "websocket": AuthMiddlewareStack(
        URLRouter(chat.routing.websocket_urlpatterns)
    ),
})
```

C) chat/routing.py

```
from django.urls import re_path
from .consumers import ChatConsumer
```

```
websocket_urlpatterns = [
    re_path(r"ws/chat/(?P<room_name>\w+)/$", ChatConsumer.as_asgi()),
]
```

Test “propre” pour vérifier que tout est bon

1. Relance serveur :

```
py manage.py runserver
```

Tu dois voir :

✅ Starting ASGI/Daphne ...

2. Ouvre ton navigateur sur / (page HTTP) ok.
3. Test WS via JS console du navigateur (F12 > Console) :

```
const ws = new WebSocket("ws://127.0.0.1:8000/ws/chat/general/");
ws.onopen = () => console.log("OPEN");
ws.onmessage = (e) => console.log("MSG", e.data);
ws.onclose = () => console.log("CLOSE");
ws.onerror = (e) => console.log("ERR", e);
```

Si tu vois OPEN, c’est bon.

Dernier point : `TemplateDoesNotExist: chat/index.html`

Ça c’est indépendant des WebSockets.

Si tu veux garder une page / avec un template, il faut créer :

```
chat/templates/chat/index.html
```

ou alors supprime ta view `index()` / route / si tu n’en as pas besoin (API-only).

En 1 phrase : où tu en es ?

✅ WebSockets OK quand tu es en ASGI (tes logs le prouvent).

Le “404” sur `/ws/...` est **normal** quand c’est un GET HTTP, et le reste vient du fait que tu **repasses parfois en WSGI** après reload.

Si tu veux, colle-moi **ton `monsite/settings.py` + `monsite/asgi.py` actuels** (juste ces 2 fichiers) et je te dis exactement pourquoi tu retombes en “Starting development server ...” après certains reload.

`settings.py`

Python

`asgi.py`

Python

Réflexion durant 6s

Tu es **presque** au bout : ton Channels/ASGI est bien câblé, mais tu mélanges parfois **HTTP GET** et **WebSocket**, et tu as aussi un souci de **template**.

1) Pourquoi tu vois `Not Found: /ws/chat/general/ (404)` alors que le WS “CONNECT” apparaît ?

- Quand tu tapes dans le navigateur `http://127.0.0.1:8000/ws/chat/general/`, c’est un **HTTP GET** → Django répond **404** (normal).
- Un WebSocket doit se faire en **`ws://`** (ou **`wss://`**), pas en **`http://`**.
- Ton `asgi.py` est correct : il route bien `"websocket"` vers `chat.routing.websocket_urlpatterns`.

`asgi`

✅ Donc : **le 404 est normal** si tu testes `/ws/...` en URL HTTP.

👉 Il faut tester via un vrai client WebSocket (JS dans une page, Postman WS, websocketking, etc.).

2) Vérifie que ton URL WS côté front utilise bien **`ws://`** et pas **`http://`**

Exemple (dans ton `index.html` ou un script) :

```
const roomName = "general";
const socket = new WebSocket(`ws://${window.location.host}/ws/chat/${roomName}/`);
```

```
socket.onopen = () => console.log("WS open");
socket.onmessage = (e) => console.log("WS message", e.data);
```

```
socket.onclose = () => console.log("WS close");
socket.onerror = (e) => console.log("WS error", e);
```

Si tu fais `fetch("/ws/chat/general/")` ou tu ouvres `http://.../ws/chat/general/` → tu auras **404** (normal).

3) Ton `settings.py` : ok pour Channels (in-memory layer)

Tu as bien :

- daphne + channels installés et listés dans `INSTALLED_APPS`
- `ASGI_APPLICATION = "monsite.asgi.application"`
- `CHANNEL_LAYERS` en `InMemoryChannelLayer` (parfait pour dev)

`settings`

👉 Donc, côté settings, c'est bon.

4) Erreur `TemplateDoesNotExist: chat/index.html`

Tu as `TEMPLATES["DIRS"] = [BASE_DIR / "templates"]` et `APP_DIRS=True`.

`settings`

Donc tu as 2 options correctes :

Option A (recommandée) : template dans l'app

Crée :

- `chat/templates/chat/index.html`

Option B : template global

Crée :

- `templates/chat/index.html` (à la racine du projet, au même niveau que `manage.py`)

Ensuite dans ta vue :

```
return render(request, "chat/index.html")
```

...marchera.

5) Ton test POST PowerShell (erreur `curl` / Headers)

Sous Windows PowerShell, `curl` appelle souvent `Invoke-WebRequest` (alias), d'où ton erreur.



Utilise **`curl.exe`** :

```
curl.exe -X POST "http://127.0.0.1:8000/api/rooms/" -H "Content-Type: application/json" -d '{"name\":"general\"}'
```



Ou `Invoke-RestMethod` correctement :

```
Invoke-RestMethod -Uri "http://127.0.0.1:8000/api/rooms/" `
    -Method Post `
    -Headers @{ "Content-Type"="application/json" } `
    -Body '{"name":"general"}'
```

6) Si tu veux un test WS rapide (sans front)

Installe un client WS python :

```
pip install websockets
```

Puis teste :

```
import asyncio, websockets
```

```
async def main():
    async with websockets.connect("ws://127.0.0.1:8000/ws/chat/general/") as ws:
        await ws.send("hello")
        print(await ws.recv())
```

```
asyncio.run(main())
```

Si tu colles ici ton `chat/routing.py` et `chat/consumers.py`, je te dis exactement pourquoi tu vois parfois “CONNECT puis DISCONNECT” (souvent c’est un `accept()` manquant, une exception dans `connect()`, ou un souci de nom de route).

Sources
`routing.py`
Python
`consumers.py`
Python
Réflexion durant 16s

Tu es **quasi bon** 👍

D’après tes logs, quand tu lances avec **Daphne/ASGI**, on voit bien :

- `WebSocket HANDSHAKING /ws/chat/general/`
- `WebSocket CONNECT /ws/chat/general/`

➡ Donc le **WebSocket** fonctionne.

1) Pourquoi tu vois encore `Not Found: /ws/chat/general/ (404)` ?

Parce que tu tapes l’URL dans le navigateur, donc le navigateur fait un **HTTP GET** :

- `http://127.0.0.1:8000/ws/chat/general/` → **HTTP** → 404 normal
- `ws://127.0.0.1:8000/ws/chat/general/` → **WebSocket** → OK (handshake/connect)

✅ Conclusion : `/ws/...` n’est pas une route Django (`urls.py`), c’est une route Channels (`routing.py`). En HTTP ça doit répondre 404, c’est attendu.

2) Tes fichiers Channels sont corrects (ceux que tu as envoyés)

- `ASGI_APPLICATION = "monsite.asgi.application"` ✅
- `asgi.py` contient bien `application = ProtocolTypeRouter({...})` ✅

- `routing.py: ^ws/chat/<room>/` 

Donc côté config, rien de “cassé”.

3) Ce qu’il te manque : une page HTML qui ouvre le WebSocket

Au lieu d’aller sur `/ws/chat/general/` dans la barre d’adresse, tu vas sur une page (ex: `/`) qui exécute du JS :

Crée ce fichier :

chat/templates/chat/index.html

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="utf-8" />
  <title>Chat WS</title>
</head>
<body>
  <h1>Room: general</h1>

  <div id="log" style="border:1px solid #ccc; padding:10px; height:200px; overflow:auto;"></div>

  <input id="msg" placeholder="Ton message..." />
  <button id="send">Envoyer</button>

  <script>
    const roomName = "general";

    const wsScheme = (window.location.protocol === "https:") ? "wss" : "ws";
    const socket = new WebSocket(`${wsScheme}://${window.location.host}/ws/chat/${roomName}/`);

    const log = (t) => {
      const el = document.getElementById("log");
      el.innerHTML += `<div>${t}</div>`;
      el.scrollTop = el.scrollHeight;
    };

    socket.onopen = () => log("🟢 WS connecté");
```

```

socket.onclose = () => log("❌ WS fermé");
socket.onerror = (e) => log("⚠️ WS erreur");
socket.onmessage = (e) => log("📬 " + e.data);

document.getElementById("send").onclick = () => {
  const content = document.getElementById("msg").value;
  socket.send(JSON.stringify({ author: "mario", content }));
  document.getElementById("msg").value = "";
};
</script>
</body>
</html>

```

Et assure-toi d'avoir une vue + url HTTP pour servir cette page.

chat/views.py

```

from django.shortcuts import render

def index(request):
    return render(request, "chat/index.html")

```

chat/urls.py

```

from django.urls import path
from .views import index

urlpatterns = [
    path("", index, name="index"),
]

```

monsite/urls.py

```

from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path("admin/", admin.site.urls),
    path("", include("chat.urls")),
]

```

Ensuite :

- `py manage.py runserver`
 - ouvre `http://127.0.0.1:8000/`
 - tu verras “WS connecté” + tu pourras envoyer des messages.
-

4) Ton erreur PowerShell “curl -H ...” (headers)

Sous PowerShell, `curl` appelle souvent **Invoke-WebRequest**, donc `-H` n’est pas interprété comme sur Linux.

Utilise plutôt :

```
Invoke-RestMethod `
  -Uri "http://127.0.0.1:8000/api/rooms/" `
  -Method Post `
  -ContentType "application/json" `
  -Body '{"name":"general"}'
```

Ou force le vrai curl :

```
curl.exe -X POST "http://127.0.0.1:8000/api/rooms/" -H "Content-Type: application/json" -d "{\"name\":\"general\"}"
```

“Impossible de se connecter au serveur distant” arrivait juste quand ton serveur n’était pas en train de tourner (ou a crash/reload au même moment).

5) Si tu veux un check rapide

Quand tu lances `runserver`, si tu vois des lignes comme :

- `WebSocket HANDSHAKING ...`
- `WebSocket CONNECT ...`



C’est que **Channels** marche.

Le 404 sur `/ws/...` en HTTP n’est pas un problème.

Si tu veux, envoie ton `monsite/urls.py` + `chat/urls.py` actuels (copier/coller) et je te dis exactement pourquoi / marche / ne marche pas chez toi (et où tu dois inclure tes routes API).

🖱️ `ws://127.0.0.1:8000/ws/chat/general/` ✅ WS connecté 📧 {"type": "error", "error": "Empty message"} 🖱️ `ws://127.0.0.1:8000/ws/chat/general/` ❌ WS fermé ✅ WS connecté 🖱️ `ws://127.0.0.1:8000/ws/chat/general/` ❌ WS fermé ✅ WS connecté 🖱️ `ws://127.0.0.1:8000/ws/chat/general/` ❌ WS fermé ✅ WS connecté 🖱️ `ws://127.0.0.1:8000/ws/chat/general/` ❌ WS fermé ✅ WS connecté 🖱️ `ws://127.0.0.1:8000/ws/chat/general/` ❌ WS fermé ✅ WS connecté 📧 {"type": "chat_message", "id": 8, "author": "Mario", "content": "Dragon", "created_at": "2026-02-07T17:37:20.143797+00:00"} 📧 {"type": "chat_message", "id": 9, "author": "Mario", "content": "Turbo", "created_at": "2026-02-07T17:37:31.534315+00:00"} 📧 {"type": "chat_message", "id": 10, "author": "Mario", "content": "Danis", "created_at": "2026-02-07T17:37:40.033565+00:00"}

Parfait — là on voit que **tout marche**, et tu as juste 2 petits comportements à corriger :

1) {"type": "error", "error": "Empty message"}

Ça veut dire que ton **client a envoyé un message vide** (souvent "" ou " "), et ton `ChatConsumer.receive()` a une validation qui refuse ça.

✅ Fix côté **front (JS)** : ne pas envoyer si vide

```
document.getElementById("send").onclick = () => {
  const input = document.getElementById("msg");
  const content = input.value.trim();
  if (!content) return; // <-- empêche message vide
  socket.send(JSON.stringify({ author: "Mario", content }));
  input.value = "";
};
```

✅ Fix côté **backend** (si tu veux être tolérant) : ignorer au lieu de renvoyer erreur/fermer

Dans `consumers.py`, dans `receive()` :

```
content = (data.get("content") or "").strip()
if not content:
    await self.send_json({"type": "error", "error": "Empty message"})
    return # <-- ne pas close
```

2) Pourquoi tu vois plein de ❌ WS fermé puis ✅ WS connecté

Ça arrive typiquement quand :

- tu rafraîchis la page (F5),
- tu as **plusieurs onglets** ouverts sur la même page,
- le navigateur recrée la connexion (ou un plugin / live-reload),
- ou ton code JS **crée plusieurs WebSocket** (script chargé plusieurs fois).

✅ Check simple :

- Ferme tous les onglets du site, rouvre **un seul** onglet.
- Ouvre la console et log le `socket.readyState` si besoin.

✅ Petite sécurité côté JS : fermer proprement quand tu quittes la page

```
window.addEventListener("beforeunload", () => {  
  try { socket.close(); } catch(e) {}  
});
```

3) Le plus important : tes messages “Dragon / Turbo / Danis” arrivent ✅


Donc :

- handshake OK
- routing OK
- consumer OK
- save DB OK
- broadcast OK

Si tu me colles ton `receive()` actuel dans `consumers.py`, je te propose la version “solide” (gestion JSON invalide, message vide, etc.) sans faire tomber la connexion.


consumers.py
Python
Réflexion durant 11s

Tu es **bien connecté en WebSocket**, et le message :

 {"type":"error","error":"Empty message"}

vient **directement** de ton `ChatConsumer.receive()` : si `content` est vide (ou si tu envoies `""` / `{}` / rien), le serveur répond avec cette erreur.

consumers

Ensuite, le fait que tu vois “ **WS fermé**” juste après est presque toujours côté **client** : ton script ferme la socket après avoir reçu une erreur (ou il envoie un message vide au moment du connect, reçoit l’erreur, puis se ferme/reconnecte).

Correctifs recommandés

1) Côté serveur : ignorer silencieusement les “messages vides” (au lieu de renvoyer une erreur)


Dans `receive()`, remplace le bloc “Empty message” par un simple `return` :

```
if not content:
    # Ignore (ne pas spammer d'erreur, ne pas casser le client)
    return
```

Ça évite que ton client interprète ça comme une erreur “fatale” et ferme la connexion.

consumers

2) Côté client : ne rien envoyer au connect

Beaucoup de clients (ou ton code) font un `ws.send("")` / `ws.send("{}")` juste après “ connecté”. Assure-toi que **tu n’appelles pas send** tant que l’utilisateur n’a pas tapé un vrai message.

Exemple (logique) :

- `onopen` → afficher “connecté” (mais **ne pas send**)
- bouton “Send” → si `message.trim() === ""` → **ne rien faire**

3) Option robuste : gérer un “ping” ou des messages non-chat

Si tu veux envoyer des messages techniques depuis le client, utilise un champ `type` :

Client :

```
{"type": "ping"}
```

Serveur :

```
msg_type = payload.get("type")
if msg_type == "ping":
    await self.send_json({"type": "pong"})
    return
```

Pourquoi tu reçois quand même des `chat_message` ensuite ?

Parce qu’une fois que tu envoies un payload non vide (`{"author": "Mario", "content": "Dragon"}`), ton consumer sauvegarde puis broadcast correctement via `group_send`, et tu reçois bien les events `chat_message`.

consumers

Si tu veux, colle ici **le code JS** (ou ton client Python) qui fait la connexion WS : je te dirai exactement où il envoie le message vide / pourquoi il ferme la socket.

Sources

quel est le corrigé du fichier `consumers.py`

Réflexion durant 29s

Voici une version **corrigée / robuste** de `consumers.py` (compatible Django Channels 4.x) : elle :

- accepte la connexion + crée la room si elle n’existe pas,
- gère **JSON invalide** / **message vide** sans faire planter le WS,
- écrit en base via `database_sync_to_async`,
- broadcast le message au groupe.

```
import json
import logging

from channels.db import database_sync_to_async
from channels.generic.websocket import AsyncWebsocketConsumer
```

```

from .models import Room, Message

logger = logging.getLogger(__name__)

class ChatConsumer(AsyncWebsocketConsumer):
    async def connect(self):
        self.room_name = self.scope["url_route"]["kwargs"]["room_name"]
        self.group_name = f"chat_{self.room_name}"

        await self.channel_layer.group_add(self.group_name, self.channel_name)
        await self.accept()

        # S'assure que la room existe
        await self.get_or_create_room(self.room_name)

    async def disconnect(self, close_code):
        await self.channel_layer.group_discard(self.group_name, self.channel_name)

    async def receive(self, text_data=None, bytes_data=None):
        # Tolérant: ne crash jamais sur un payload invalide
        try:
            payload = json.loads(text_data or "{}")
        except json.JSONDecodeError:
            await self.send_json({"type": "error", "error": "Invalid JSON"})
            return

        author = (payload.get("author") or "anonymous").strip()[:64]
        content = (payload.get("content") or "").strip()

        if not content:
            await self.send_json({"type": "error", "error": "Empty message"})
            return

        try:
            msg = await self.create_message(self.room_name, author, content)
        except Exception:
            logger.exception("Failed to create message")
            await self.send_json({"type": "error", "error": "Server error saving message"})
            return

        await self.channel_layer.group_send(
            self.group_name,
            {
                "type": "chat_message",
                "id": msg["id"],
            }
        )

```

```

        "author": msg["author"],
        "content": msg["content"],
        "created_at": msg["created_at"],
    },
)

async def chat_message(self, event):
    await self.send_json(event)

async def send_json(self, data: dict):
    await self.send(text_data=json.dumps(data, ensure_ascii=False))

@database_sync_to_async
def get_or_create_room(self, name: str):
    Room.objects.get_or_create(name=name)

@database_sync_to_async
def create_message(self, room_name: str, author: str, content: str):
    room, _ = Room.objects.get_or_create(name=room_name)
    msg = Message.objects.create(room=room, author=author, content=content)
    return {
        "id": msg.id,
        "author": msg.author,
        "content": msg.content,
        "created_at": msg.created_at.isoformat(),
    }

```

✅ **Important** : ton log `{"type": "error", "error": "Empty message"}` vient presque toujours du **client** qui envoie "" ou {} à l’ouverture. Ce consumer répond juste avec une erreur **sans fermer** la connexion. Si tu veux, colle ici ton code JS (ou Postman/websocket client) et je te dis exactement pourquoi tu vois “WS fermé” côté client.

Pouvez vous lister tous les API utilisé ici

Oui — dans ce `consumers.py`, les “API” utilisées sont essentiellement celles de **Django Channels**, **channel layer**, **Channels DB**, et **Django ORM** (plus JSON/logging). Voici la liste complète, classée par module.

1) Python standard library

- `json.loads(...)`
- `json.dumps(..., ensure_ascii=False)`
- `logging.getLogger(__name__)`
- `logger.exception(...)`

2) Django Channels (WebSocket)

Depuis `channels.generic.websocket.AsyncWebsocketConsumer` :

- `AsyncWebsocketConsumer.connect(self)` (*méthode que tu overrides*)
- `AsyncWebsocketConsumer.disconnect(self, close_code)` (*override*)
- `AsyncWebsocketConsumer.receive(self, text_data=None, bytes_data=None)` (*override*)
- `self.accept()`
- `self.send(text_data=...)`

Accès à la route / paramètres :

- `self.scope["url_route"]["kwargs"]["room_name"]`

3) Channel Layer (groupes + broadcast)

Via `self.channel_layer` :

- `self.channel_layer.group_add(group, channel_name)`
- `self.channel_layer.group_discard(group, channel_name)`
- `self.channel_layer.group_send(group, event_dict)`

Et la propriété Channels :

- `self.channel_name`

4) Channels DB (bridge async ↔ sync)

Depuis `channels.db` :

- `@database_sync_to_async` (décorateur)

5) Django ORM (modèles Room / Message)

Depuis `Room` et `Message` :

- `Room.objects.get_or_create(name=...)`
- `Message.objects.create(room=..., author=..., content=...)`
- accès aux champs du modèle :
 - `msg.id`
 - `msg.author`
 - `msg.content`
 - `msg.created_at.isoformat()`

6) “API” interne du consumer (helpers que j’ai ajoutés)

- `send_json(self, data: dict)`
- `get_or_create_room(self, name: str)`
- `create_message(self, room_name: str, author: str, content: str)`
- `chat_message(self, event)` (*handler appelé automatiquement quand type: "chat_message" est envoyé par group_send*)

Si tu veux, je peux aussi te lister les **API côté URLs/routing/asgi** qui participent au WS (`ProtocolTypeRouter`, `URLRouter`, `AuthMiddlewareStack`, `re_path`, etc.) en regardant tes fichiers `asgi.py` et `routing.py` que tu as uploadés.

Je voulais dire tous les API du genre `http://127.0.0.1:8000/api/rooms/`
Réflexion durant 42s

Voici **toutes les routes “API”** (dans ton projet actuel) du style `http://127.0.0.1:8000/api/...` — avec les **méthodes HTTP** supportées.

Endpoints HTTP

1) Rooms

- **GET** `http://127.0.0.1:8000/api/rooms/`
→ liste des rooms
- **POST** `http://127.0.0.1:8000/api/rooms/`
→ crée une room (JSON: `{"name": "general"}`)

Exemple PowerShell (POST)

```
Invoke-RestMethod `
  -Uri "http://127.0.0.1:8000/api/rooms/" `
  -Method Post `
  -ContentType "application/json" `
```

```
-Body '{"name":"general"}'
```

2) Messages d'une room

- **GET** `http://127.0.0.1:8000/api/rooms/<room_id>/messages/`
→ liste des messages de la room

Exemple

- `http://127.0.0.1:8000/api/rooms/1/messages/`

⚠ Dans ton code actuel, cet endpoint est un **ListAPIView** → donc **GET uniquement** (pas de POST ici).

Autres routes HTTP (non “/api”)

- **GET** `http://127.0.0.1:8000/` (page index)
 - **GET** `http://127.0.0.1:8000/admin/` (admin Django)
-

(Bonus) Endpoint WebSocket

Même si ce n'est pas du HTTP, ton endpoint WS est :

- `ws://127.0.0.1:8000/ws/chat/<room_name>/`
Ex: `ws://127.0.0.1:8000/ws/chat/general/`

Si tu veux, je peux aussi te lister **les formats JSON exacts** attendus/émis par le WebSocket (types de messages, champs, erreurs) à partir de ton `consumers.py`.

oui

Réflexion durant 1m 3s

Voici **le corrigé de `consumers.py`** (version plus robuste) + **la liste des endpoints HTTP `/api/`...** de ton projet.

Corrigé `chat/consumers.py`

Points corrigés/améliorés :

- si le message est vide → renvoie `{"type": "error", "error": "Empty message"}` (comme tu as vu dans ton client)
- gestion JSON invalide
- validation `author/content`
- `Room.get_or_create()` aussi côté création de message (évite un `DoesNotExist` en cas de course)
- envoie un format cohérent + broadcast au groupe

```
import json

from channels.generic.websocket import AsyncWebsocketConsumer
from channels.db import database_sync_to_async

from .models import Room, Message

class ChatConsumer(AsyncWebsocketConsumer):
    async def connect(self):
        self.room_name = self.scope["url_route"]["kwargs"]["room_name"]
        self.room_group_name = f"chat_{self.room_name}"

        # Rejoindre le groupe WS
        await self.channel_layer.group_add(self.room_group_name, self.channel_name)
        await self.accept()

        # Créer la room si elle n'existe pas
        await self.get_or_create_room(self.room_name)

    async def disconnect(self, close_code):
        await self.channel_layer.group_discard(self.room_group_name, self.channel_name)

    async def receive(self, text_data=None, bytes_data=None):
        # On ignore les messages binaires
        if text_data is None:
            return

        # JSON parsing
        try:
```

```

        data = json.loads(text_data)
    except json.JSONDecodeError:
        await self.send(text_data=json.dumps({"type": "error", "error": "Invalid JSON"}))
        return

    author = (data.get("author") or "").strip()
    content = (data.get("content") or "").strip()

    if not author:
        await self.send(text_data=json.dumps({"type": "error", "error": "Missing author"}))
        return

    if not content:
        await self.send(text_data=json.dumps({"type": "error", "error": "Empty message"}))
        return

    # DB: sauvegarde
    message = await self.create_message(self.room_name, author, content)

    # Broadcast au groupe
    await self.channel_layer.group_send(
        self.room_group_name,
        {
            "type": "chat_message",
            "id": message.id,
            "author": message.author,
            "content": message.content,
            "created_at": message.created_at.isoformat(),
        },
    )

async def chat_message(self, event):
    # Renvoi au client
    await self.send(text_data=json.dumps(event))

@database_sync_to_async
def get_or_create_room(self, name):
    room, _ = Room.objects.get_or_create(name=name)
    return room

@database_sync_to_async
def create_message(self, room_name, author, content):
    room, _ = Room.objects.get_or_create(name=room_name)
    return Message.objects.create(room=room, author=author, content=content)

```

Liste des API HTTP `/api/...` dans ton projet


D'après tes fichiers `monsite/urls.py`, `chat/urls.py` et `chat/views.py`, tu as :

1) Rooms

GET `/api/rooms/`

 liste toutes les rooms

POST `/api/rooms/`


 crée une room

Body JSON :

```
{"name": "general"}
```

2) Messages d'une room

GET `/api/rooms/<room_id>/messages/`

 liste les messages de la room (les 200 derniers max)

Exemple :

- `GET /api/rooms/1/messages/`

(Bonus) WebSocket (ce n'est pas du HTTP API, mais utile)

ws `://127.0.0.1:8000/ws/chat/<room_name>/`

Exemple :

- `ws://127.0.0.1:8000/ws/chat/general/`
-

Si tu veux, colle-moi aussi ton `chat/serializers.py` (ou upload) et je te dis exactement quels champs sont exigés côté `/api/rooms/` et `/api/rooms/<id>/messages/`.