

```
!pip install fastapi uvicorn pyngrok nest_asyncio
```

```
Requirement already satisfied: fastapi in /usr/local/lib/python3.12/dist-packages (0.118.0)
Requirement already satisfied: uvicorn in /usr/local/lib/python3.12/dist-packages (0.37.0)
Collecting pyngrok
  Downloading pyngrok-7.4.0-py3-none-any.whl.metadata (8.1 kB)
Requirement already satisfied: nest_asyncio in /usr/local/lib/python3.12/dist-packages (1.6.0)
Requirement already satisfied: starlette<0.49.0,>=0.40.0 in /usr/local/lib/python3.12/dist-packages (from fastapi) (0.48.0)
Requirement already satisfied: pydantic!=1.8,!<1.8.1,!<2.0.0,!<2.0.1,!<2.1.0,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/dist-pa
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.12/dist-packages (from fastapi) (4.15.0)
Requirement already satisfied: click>=7.0 in /usr/local/lib/python3.12/dist-packages (from uvicorn) (8.3.0)
Requirement already satisfied: h11>=0.8 in /usr/local/lib/python3.12/dist-packages (from uvicorn) (0.16.0)
Requirement already satisfied: PyYAML>=5.1 in /usr/local/lib/python3.12/dist-packages (from pyngrok) (6.0.3)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!<1.8.1,!<
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!<1.8.1,!<
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!<1.8.1,!<
Requirement already satisfied: anyio<5,>=3.6.2 in /usr/local/lib/python3.12/dist-packages (from starlette<0.49.0,>=0.40.0->fast
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.12/dist-packages (from anyio<5,>=3.6.2->starlette<0.49.0,>=0.
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.12/dist-packages (from anyio<5,>=3.6.2->starlette<0.49.0,>=0.
Downloading pyngrok-7.4.0-py3-none-any.whl (25 kB)
Installing collected packages: pyngrok
Successfully installed pyngrok-7.4.0
```

```
!ngrok config add-authtoken 33gIBhr6wU4dYniiWVpjGBqdEbt_3ZdmNGK2V4Rswr8K3bAYm
```

```
Authtoken saved to configuration file: /root/.config/ngrok/ngrok.yml
```

```
import nest_asyncio
from pyngrok import ngrok
from fastapi import FastAPI
from pydantic import BaseModel
from typing import List
import uvicorn
import asyncio
```

```
!pip install fastapi uvicorn pyngrok nest_asyncio
```

```
import nest_asyncio
import uvicorn
import asyncio
from fastapi import FastAPI, HTTPException
from pydantic import BaseModel
from typing import List, Optional
from pyngrok import ngrok
from datetime import datetime
import uuid

# Allow nested event loops
nest_asyncio.apply()

# Configure ngrok with your authtoken
NGROK_AUTHTOKEN = "33gIBhr6wU4dYniiWVpjGBqdEbt_3ZdmNGK2V4Rswr8K3bAYm"
ngrok.set_auth_token(NGROK_AUTHTOKEN)
print("✅ Ngrok authtoken configured successfully!")

# FastAPI app
app = FastAPI(
    title="Chat App",
    description="A simple real-time chat application",
    version="1.0.0"
)

# Enhanced message storage
chat_history = []

class Message(BaseModel):
    sender: str
    text: str
    message_id: Optional[str] = None
    timestamp: Optional[str] = None

@app.post("/chat", response_model=dict)
```

```

async def send_message(message: Message):
    """Send a new message to the chat"""
    try:
        # Add metadata
        message_dict = message.dict()
        message_dict["message_id"] = str(uuid.uuid4())
        message_dict["timestamp"] = datetime.now().isoformat()

        chat_history.append(message_dict)

        # Keep only last 100 messages to prevent memory issues
        if len(chat_history) > 100:
            chat_history.pop(0)

    return {
        "status": "Message received",
        "message_id": message_dict["message_id"],
        "timestamp": message_dict["timestamp"]
    }
    except Exception as e:
        raise HTTPException(status_code=500, detail=f"Error sending message: {str(e)}")

@app.get("/history", response_model=List[Message])
async def get_history(limit: Optional[int] = None):
    """Get chat history with optional limit"""
    try:
        if limit and limit > 0:
            return chat_history[-limit:]
        return chat_history
    except Exception as e:
        raise HTTPException(status_code=500, detail=f"Error retrieving history: {str(e)}")

@app.delete("/clear")
async def clear_history():
    """Clear all chat history"""
    try:
        chat_history.clear()
        return {"status": "Chat history cleared"}
    except Exception as e:
        raise HTTPException(status_code=500, detail=f"Error clearing history: {str(e)}")

@app.get("/")
async def root():
    return {
        "message": "Chat API is running!",
        "docs": "/docs",
        "endpoints": {
            "send_message": "POST /chat",
            "get_history": "GET /history",
            "clear_history": "DELETE /clear"
        }
    }

# Enhanced server setup
try:
    # Kill existing tunnels
    ngrok.kill()

    # Create tunnel with authenticated account
    public_url = ngrok.connect(8000, proto="http").public_url
    print(f"🔴 Public URL: {public_url}")
    print(f"📘 Swagger Docs: {public_url}/docs")
    print(f"🔗 Direct API: {public_url}/chat")
    print("\n💡 Try these commands:")
    print(f'curl -X POST "{public_url}/chat" -H "Content-Type: application/json" -d \'{{"sender": "User1", "text": "Hello World"}}
```

```

print(f'curl "{public_url}/history"')
print(f'curl -X DELETE "{public_url}/clear"')

except Exception as e:
    print(f"🔴 Ngrok error: {e}")
    print("⚡️ Trying to run without public URL...")
```

```

# Run FastAPI server
try:
    config = uvicorn.Config(
        app=app,
        host="0.0.0.0",
```

```
        port=8000,
        log_level="info",
        access_log=True
    )
server = uvicorn.Server(config)

print("✅ Server starting... Press Ctrl+C to stop")
loop = asyncio.get_event_loop()
loop.create_task(server.serve())

except Exception as e:
    print(f"❌ Server error: {e}")

Requirement already satisfied: fastapi in /usr/local/lib/python3.12/dist-packages (0.118.0)
Requirement already satisfied: uvicorn in /usr/local/lib/python3.12/dist-packages (0.37.0)
Requirement already satisfied: pyngrok in /usr/local/lib/python3.12/dist-packages (7.4.0)
Requirement already satisfied: nest_asyncio in /usr/local/lib/python3.12/dist-packages (1.6.0)
Requirement already satisfied: starlette<0.49.0,>=0.40.0 in /usr/local/lib/python3.12/dist-packages (from fastapi) (0.48.0)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/dist-pa
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.12/dist-packages (from fastapi) (4.15.0)
Requirement already satisfied: click>=7.0 in /usr/local/lib/python3.12/dist-packages (from uvicorn) (8.3.0)
Requirement already satisfied: h11>=0.8 in /usr/local/lib/python3.12/dist-packages (from uvicorn) (0.16.0)
Requirement already satisfied: PyYAML>=5.1 in /usr/local/lib/python3.12/dist-packages (from pyngrok) (6.0.3)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,!=2
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,!=2.
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,!=2.
Requirement already satisfied: anyio<5,>=3.6.2 in /usr/local/lib/python3.12/dist-packages (from starlette<0.49.0,>=0.40.0->fast
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.12/dist-packages (from anyio<5,>=3.6.2->starlette<0.49.0,>=0.
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.12/dist-packages (from anyio<5,>=3.6.2->starlette<0.49.0,>
✅ Ngrok authoken configured successfully!
WARNING:pyngrok.process.ngrok:t=2025-10-06T06:54:11+0000 lvl=warn msg="failed to start tunnel" pg=/api/tunnels id=468d5e8a08ac28
❌ Ngrok error: ngrok client exception, API returned 502: {"error_code":103,"status_code":502,"msg":"failed to start tunnel","de

    ✅ Trying to run without public URL...
    ✅ Server starting... Press Ctrl+C to stop
```

Start coding or generate with AI.