

# Enhanced Chat API - 500 Error Fix Summary

---

**Date: October 7, 2025**

---

## Problem Identified

---

The `/api/ai/enhanced-chat` endpoint was returning a **500 Internal Server Error** when users tried to use the AI Assistant chat feature.

## Root Cause Analysis

---

After investigation, the issue was identified as:

1. **Ollama AI Service Not Running:** The primary cause was that Ollama (the local AI inference engine) was not installed or running on the system.
2. **Missing Model:** The required `llama3.2:3b` model was not downloaded.
3. **Connection Refused Error:** The API was trying to connect to `http://127.0.0.1:11434` but getting `ECONNREFUSED` errors.

## Investigation Process

---

1. **Server Status Check:** Verified the Next.js dev server was running
2. **API Route Verification:** Confirmed the route file exists at `src/app/api/ai/enhanced-chat/route.ts`
3. **Initial 404 Issues:** Discovered symlink-related routing issues (resolved by temporarily removing symlinks)
4. **Error Log Analysis:** Found the actual error: `Error: connect ECONNREFUSED 127.0.0.1:11434`
5. **Ollama Status:** Confirmed Ollama was not installed on the system

## Solution Implemented

---

### 1. Installed Ollama

```
curl -fsSL https://ollama.com/install.sh | sh
```

### 2. Started Ollama Service

```
nohup ollama serve > /tmp/ollama_serve.log 2>&1 &
```

### 3. Downloaded Required Model

```
ollama pull llama3.2:3b
```

### 4. Verified API Functionality

Tested the endpoint with multiple queries:

```
curl -X POST http://localhost:3000/api/ai/enhanced-chat \
-H "Content-Type: application/json" \
-d '{"message": "Hello, what can you help me with?"}'
```

## Current Status

✅ **RESOLVED** - The enhanced-chat API is now fully functional

## Test Results

**Test 1:** General greeting

- **Status:** ✅ Success
- **Response:** AI provided helpful context-aware response about Soundtrack integration
- **Context Used:** Yes (both codebase and knowledge base)

**Test 2:** TV control query

- **Status:** ✅ Success
- **Response:** AI provided detailed instructions on controlling TVs using DirecTV and bartender remotes
- **Context Used:** Yes (both codebase and knowledge base)

**Test 3:** Post-symlink restoration

- **Status:** ✅ Success
- **Response:** AI provided relevant guidance
- **Context Used:** Yes

## API Response Format

The API now returns proper JSON responses:

```
{
  "response": "AI-generated response text",
  "model": "llama3.2:3b",
  "usedContext": true,
  "usedCodebase": true,
  "usedKnowledge": true
}
```

## System Requirements Met

- ✅ Ollama installed and running on port 11434
- ✅ Model `llama3.2:3b` downloaded (2.0 GB)
- ✅ Next.js dev server running on port 3000
- ✅ API route properly configured
- ✅ Knowledge base integration working
- ✅ Codebase context integration working

## Notes About PR #104

PR #104 was created to add better error handling to the enhanced-chat endpoint. However, the current main branch code already has the route file and works correctly once Ollama is running. The PR

may still be useful for:

- Additional error handling improvements
- Better error messages for users
- Graceful degradation when Ollama is unavailable

## Recommendations

---

### For Production Deployment

1. **Ensure Ollama is Running:** Set up Ollama as a systemd service for automatic startup

```
bash
sudo systemctl enable ollama
sudo systemctl start ollama
```

2. **Monitor Ollama Status:** Add health checks to ensure Ollama is always available
3. **Error Handling:** Consider implementing the improvements from PR #104 for better user experience when Ollama is unavailable
4. **Model Persistence:** Ensure the `llama3.2:3b` model persists across system restarts

### For Development

1. **Start Ollama Before Dev Server:** Always ensure Ollama is running before starting the Next.js dev server

```
bash
ollama serve &
npm run dev
```

2. **Check Ollama Status:** Use `curl http://127.0.0.1:11434/api/tags` to verify Ollama is responding
3. **Model Management:** Keep track of downloaded models with `ollama list`






## Files Involved

---

- **API Route:** `src/app/api/ai/enhanced-chat/route.ts`
- **AI Knowledge Integration:** `lib/ai-knowledge-enhanced.ts`
- **Ollama Service:** Running on `http://127.0.0.1:11434`
- **Model:** `llama3.2:3b` (2.0 GB)

## Next Steps

---

1.  Enhanced-chat API is working
2.  Consider merging PR #104 for additional error handling
3.  Set up Ollama as a persistent service
4.  Add monitoring/alerting for Ollama service health
5.  Document Ollama setup in deployment guide

## Conclusion

---

The 500 error was successfully resolved by installing and configuring Ollama. The AI Assistant chat feature is now fully operational and providing context-aware responses using both the knowledge base and codebase information.