```
# OpenMind.bot streamlines social interactions between personalized bots, representing users, media, and influencers, ensuring meaningful exchanges. It eliminates misunderstandings by using context-aware conversations, followed by summaries or audio recaps of these interactions for efficient communication.

import json
import datetime
import pytz

from flask import Flask, request, jsonify

app = Flask(__name__)


@app.route("/api/v1/conversations", methods=["POST"])
def create_conversation():
    # Create a new conversation
    conversation = {
        "user_id": request.json["user_id"],
        "bot_id": request.json["bot_id"],
        "messages": [],
    }

    # Save the conversation to the database
    with open("conversations.json", "w") as f:
        json.dump(conversation, f)
    return jsonify(conversation)



@app.route("/api/v1/conversations/<conversation_id>", methods=["GET"])
def get_conversation(conversation_id):
    # Get the conversation from the database
    with open("conversations.json", "r") as f:
        conversation = json.load(f)

    # Return the conversation
    return jsonify(conversation)



@app.route(
    "/api/v1/conversations/<conversation_id>/messages",
    methods=["POST"],
)
def create_message(conversation_id):
    # Create a new message
    message = {
        "user_id": request.json["user_id"],
        "bot_id": request.json["bot_id"],
        "text": request.json["text"],
        "timestamp": datetime.datetime.now(pytz.utc).isoformat(),
    }
    # Get the conversation from the database
    with open("conversations.json", "r") as f:
        conversation = json.load(f)

    # Add the message to the conversation
    conversation["messages"].append(message)

    # Save the conversation to the database
    with open("conversations.json", "w") as f:
        json.dump(conversation, f)

    return jsonify(message)


@app.route(
    "/api/v1/conversations/<conversation_id>/messages",
    methods=["GET"],
)
def get_messages(conversation_id):
    # Get the conversation from the database
    with open("conversations.json", "r") as f:
        conversation = json.load(f)

    # Return the messages
    return jsonify(conversation["messages"])
```

```
@app.route(
    "/api/v1/conversations/<conversation_id>/summary", methods=["GET"]
)
def get_summary(conversation_id):
    # Get the conversation from the database
    with open("conversations.json", "r") as f:
        conversation = json.load(f)

    # Create a summary of the conversation
    summary = ""
    for message in conversation["messages"]:
        summary += message["text"] + "\n"

    # Return the summary
    return jsonify(summary)


@app.route(
    "/api/v1/conversations/<conversation_id>/audio_recap",
    methods=["GET"],
)
def get_audio_recap(conversation_id):
    # Get the conversation from the database
    with open("conversations.json", "r") as f:
        conversation = json.load(f)

    # Create an audio recap of the conversation
    audio_recap = ""
    for message in conversation["messages"]:
        audio_recap += message["text"] + "\n"

    # Return the audio recap
    return jsonify(audio_recap)


if __name__ == "__main__":
    app.run()
```