Configurações do LM Studio

Projeto desenvolvido para suporte a um sistema de Help Desk, via Watsapp integrado com IA.

Ferramentas utilizadas . Node.js, Nexts 14 e LM Studio como ferramenta de integração.

[LM Studio REST API (beta) | LM Studio Docs](https://lmstudio.ai/docs/app/api/endpoints/rest)

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Após : executar o comando no terminal lms server start

Projeto :

mkdir Nome do projeto

cd Nome do projeto

npm init -y

npm install @whiskeysockets/baileys express axios

npm install ejs

importar npm install mime-types axios

npm install whatsapp-web.js qrcode-terminal

**Index.js**

const { Client, LocalAuth } = require('whatsapp-web.js');

const qrcode = require('qrcode-terminal');

const sessions = require('./sessions');

const { perguntarDeepSeek } = require('./lmstudio');

const client = new Client({

authStrategy: new LocalAuth()

});

client.on('qr', qr => {

qrcode.generate(qr, { small: true });

});

client.on('ready', () => {

console.log('✅ WhatsApp conectado!');

});

client.on('message', async msg => {

const chatId = msg.from;

const texto = msg.body.trim();

const sessao = sessions.get(chatId);

*// Caso esteja em modo DeepSeek*

if (sessao?.modo === 'deepseek') {

    const resposta = await perguntarDeepSeek(texto);

    client.sendMessage(chatId, resposta);

    return;

}

switch (texto) {

    case '/start':

    client.sendMessage(chatId, `👋 Olá! Como podemos ajudar?

1️⃣ Fazer uma chamada com um atendente

2️⃣ Falar com nosso suporte automático 🤖

3️⃣ Enviar um e-mail ao suporte

Digite o número da opção desejada.`);

    break;

    case '1':

    client.sendMessage(chatId, '📞 Ligue para nosso suporte: (11) 1234-5678');

    break;

    case '2':

    sessions.set(chatId, { modo: 'deepseek' });

    client.sendMessage(chatId, '🤖 Você está agora falando com o assistente. Envie sua dúvida!');

    break;

    case '3':

    client.sendMessage(chatId, '📧 Envie um e-mail para suporte@empresa.com');

    break;

    case '/sair':

    sessions.remove(chatId);

    client.sendMessage(chatId, '🛑 Sessão encerrada. Digite `/start` para começar novamente.');

    break;

    default:

    client.sendMessage(chatId, '❌ Comando não reconhecido. Digite `/start` para ver o menu.');

    break;

}

});

client.initialize();

**sessions.js**

const sessions = {};

module.exports = {

get: (id) => sessions[id],

set: (id, data) => { sessions[id] = data },

remove: (id) => { delete sessions[id] },

};

**Lmstudio.js** 🡪 Responsável pela conexão com o lmstudio.

const axios = require('axios');

async function perguntarDeepSeek(mensagem) {

try {

    const res = await axios.post(

    'http://localhost:1234/v1/chat/completions',

    {

        model: "phi-2", //moledo utilizado para comunicação com a IA escolhida.

        stream: true,

        messages: [

            { role: "system", content: "Você é um assistente que responde sempre em \*\*português do Brasil\*\*, mesmo que a pergunta esteja em outro idioma." },

            { role: "user", content: mensagem }

        ]

    },

    {

        responseType: 'stream'

    }

    );

    return await new Promise((resolve, reject) => {

    let resposta = '';

    res.data.on('data', (chunk) => {

        const linhas = chunk.toString().split('\n');

        for (const linha of linhas) {

        if (linha.startsWith('data: ')) {

            const json = linha.replace('data: ', '').trim();

            if (json === '[DONE]') return;

            try {

            const delta = JSON.parse(json);

            const content = delta.choices?.[0]?.delta?.content;

            if (content) resposta += content;

            } catch (e) {

            console.error('Erro ao processar JSON da stream:', e.message);

            }

        }

        }

    });

    res.data.on('end', () => {

        resolve(resposta || '⚠️ Sem resposta do modelo.');

    });

    res.data.on('error', (err) => {

        reject('⚠️ Erro na stream: ' + err.message);

    });

    });

} catch (err) {

    console.error('Erro ao chamar DeepSeek:', err.message || err);

    return '⚠️ Desculpe, houve um erro ao acessar o suporte automático.';

}

}

module.exports = {

perguntarDeepSeek

};