

Dmoat Chat App Documentation

Introduction

DMoat is a ReactJS web platform where users can engage in conversations with *AI personas* inspired by some of the most renowned historical figures. By selecting from a diverse array of meticulously prompt engineered personas, users can immerse themselves in discussions that offer both historical insight and speculative projections.

The pivotal strength of the app lies in the Openai's GPT-4 API. It harnesses the API's capabilities to fetch dynamic AI responses based on user queries, ensuring a fluid conversational experience. The design ethos of the app ensures adaptability across devices, offering seamless operation on desktops, tablets, and smartphones.

This is a full-stack introductory demo, and although GPT-4 is still far too limited, it was a fun and interesting experiment for exploring the concept of anthropomorphic AI and how humans interact with it. It served also as an important professional growth journey and it showcases my product design and development skills as well as my conceptual thinking about cutting edge technologies.

Dependencies:

The app implements the following third party libraries at the front-end: **React** and **React-dom** for the development architecture. It also uses additional and complementary packages like **react-responsive** for monitoring device size, **autosize** for managing text-areas size, **dotenv** is used for environmental variables and finally **git** for version control.

Backend support is built using **Express**, and it utilises **axios** for making API requests, as well as **CORS** and **body-parser** for handling request and response middleware.

The Code

Architecture: The platform revolves around the app component where most variables and methods are defined (and managed). Due to its limited size there is no code splitting and this allows a more straightforward readability of the main code.

File Organization: All the app components are modularized and stored neatly in the Components folder. A separate folder at the root contains the back-end components.

Comments: The code incorporates comments to elucidate its functionality and implementation.

Naming Conventions: Adherence to camelCase for variable and function naming while functional components and methods are named using PascalCase.

Programming Practices: The app predominantly utilises functional components accompanied by React hooks. Furthermore, due to its limited size, it implements only props for transferring data and variables between components(instead of using react context or third party libraries like Redux).

The Infrastructure

Backend Routes & Functionalities: GPT Personas Route (/api/gptPersonas): An asynchronous route that fetches the GPT personas from a local JSON file named gptpersonas.json. The server simulates an asynchronous action using setTimeout before sending the personas' data to the client.

Chat Route (/chat): This route accepts POST requests and forwards them to the GPT-4 API. The server expects a payload in the request. The actual interaction with the OpenAI API happens using axios. The GPT model version in use is gpt-4.

Environment Variables: DMOAT_API_KEY: This environment variable stores the API key required to authenticate with the GPT-4 API.

Middleware:

CORS: Configured to accept requests from http://localhost:3000 and http://192.168.8.105:3000.

bodyParser.json(): Used for parsing incoming request bodies in a middleware.

express.json(): Built-in middleware in Express to parse incoming requests with JSON payloads.

Useful Links & External Services

API Documentation: <https://platform.openai.com/docs/api-reference>

Notes

The app currently supports dynamic features like theme selection, managing chat interactions, seamless session management, and GPT Persona selection.

For a nuanced understanding or any ambiguities, it's advisable to directly consult the source code and any related comments. Remember, this documentation serves as a high-level guide to understanding the application's infrastructure and functionalities.

Developer contact: hello@joscoyne.co.uk