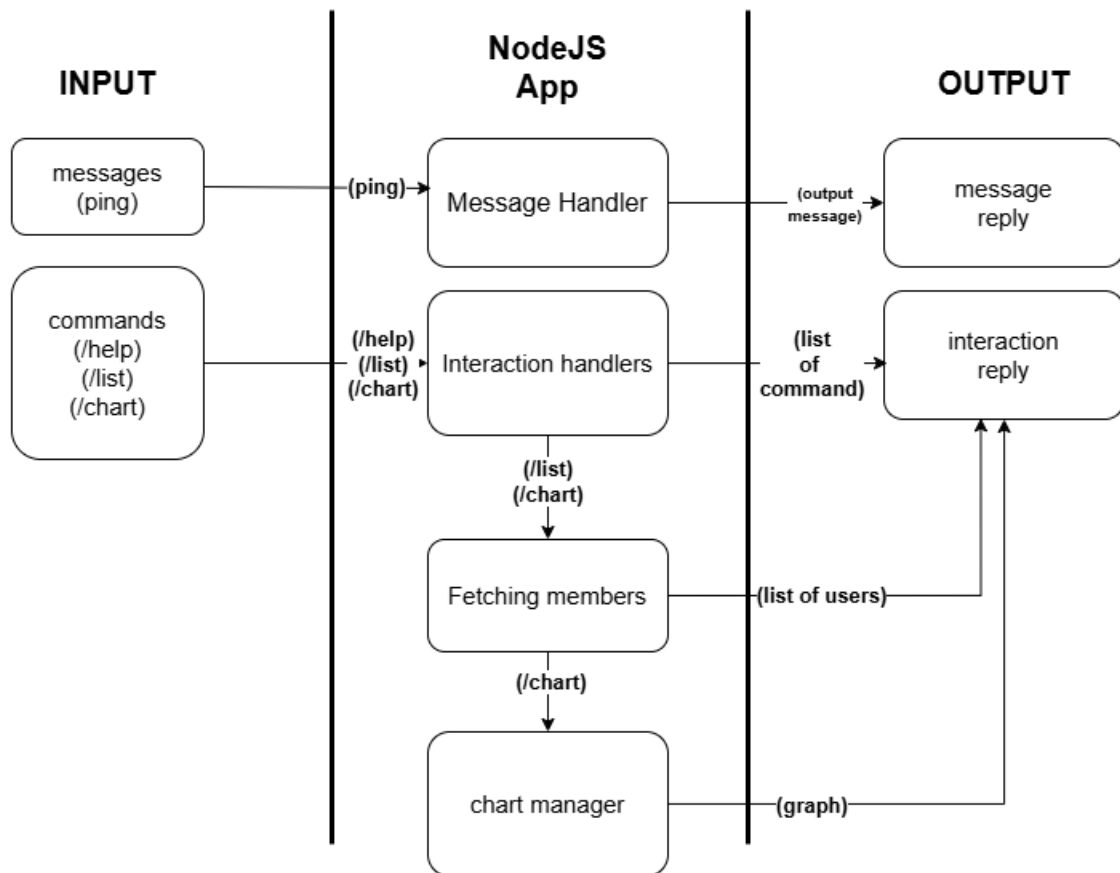


Introduction

This documentation outlines the workflow for a Discord bot built with Node.js using the Discord.js library. The bot handles specific user commands and interacts with the server to provide responses. The workflow includes commands such as ping, /help, /list, and /chart, each providing different functionalities such as server status, available commands, user lists based on status, and dynamic chart generation.



Workflow

1. Ping message

- Trigger: User sends a message with the word "ping".
- Action: The bot receives the message and triggers a response from the server.
- Response: The server replies with the message: "Node.js server is up and running", confirming that the bot's server is active and functioning.

2. Help Command

- Trigger: User sends the /help command.
- Action: The bot processes the command and fetches a list of available commands from the system.
- Response: The bot replies with a list of commands it can execute, such as /ping, /list, and /chart. This allows the user to understand what commands are available to interact with the bot.

3. List Command

- Trigger: User sends the /list command with an optional parameter to filter users by their status (e.g., "online", "offline", "don't disturb", or "idle").
- Action: The bot retrieves and filters the list of users in the server based on their status.
- Response: The bot sends a list of users whose status matches the filter. The list is presented in the chat for the user to see which users are currently online or offline, depending on the option provided.

4. Chart Command

- Trigger: User sends the /chart command with an option to choose the chart type (either "bar" or "pie").
- Action: The bot processes the command, gathers the number of users in each status (e.g., online, offline), and creates a chart using Chart.js.
- Response: The bot generates a chart, either a bar or pie chart, depending on the user's selection. The chart displays the number of users in each status category. The bot sends the generated chart as an image or visual output in the chat.

