

# React CLI Application Documentation

## Overview

This document serves as a comprehensive guide to the React CLI application, detailing its structure, functionalities, and usage instructions. It is intended for developers and users who wish to understand and interact with the application effectively.

## Application Hierarchy

```
.
├── Backend
│   ├── dist
│   │   └── app.js
│   ├── package.json
│   └── tsconfig.json
├── cli-app
│   ├── .gitignore
│   ├── package.json
│   ├── public
│   │   ├── index.html
│   │   └── manifest.json
│   └── src
│       ├── App.css
│       ├── App.js
│       ├── components
│       │   ├── terminal.css
│       │   └── terminal.js
│       ├── index.css
│       ├── index.js
│       ├── services
│       │   ├── commandService.js
│       │   └── chartService.js
│       └── setupTests.js
└── README.md
```

# User Guide:

- **Executing Commands**

- The React CLI allows users to execute various commands. Here's how to use them:

- ❖ **help**

- Description: Displays available commands.
- Usage: Type **help** in the terminal.

- ❖ **about**

- Description: Shows information about the CLI.
- Usage: Type **about** in the terminal.

- ❖ **fetch-price [coin]**

- Description: Fetches the current price of a specified cryptocurrency.
- Usage: **fetch-price btcusdt**

- ❖ **upload**

- Description: Opens a file explorer to upload CSV files.
- Usage: Type **upload** in the terminal.

- ❖ **draw [file] [columns]**

- Description: Draws a chart based on specified columns of a file.
- Usage: **draw Data.csv Price,Low**

- ❖ **delete [file]**

- Description: Deletes a specified file.
- Usage: **delete Data.csv**

- **Error Handling**

- In case of errors, the CLI will provide feedback. For instance, if a command is not recognized or if there's an issue with file processing, an error message will be displayed.

## Backend Structure:

The backend is a Node.js application.

### **app.js**

- Location: **Backend/dist/app.js**
- Purpose: Serves as the main entry point of the backend application, handling file uploads, data processing, and API requests.

### **Configuration Files**

- **package.json**: Defines the project's metadata and dependencies.
- **tsconfig.json**: Configuration for TypeScript compilation.

### **CLI-App Structure**

This section houses the front-end React application.

### **App.js**

- Location: **cli-app/src/App.js**
- Purpose: Main React component, acting as the root of the application.

### **Components**

#### **terminal.js**

- Location: **cli-app/src/components/terminal.js**
- Function: Implements the CLI interface, processing user inputs and displaying outputs.

### **CSS Files**

- **App.css, terminal.css, index.css**: Provide styling for the application.

### **Services**

#### **commandService.js**

- Location: **cli-app/src/services/commandService.**

- **Function:** Handles the logic for processing various CLI commands.

#### **chartService.js**

- **Location:** `cli-app/src/services/chartService.js`
- **Function:** Manages the rendering of charts based on provided data.

#### **Miscellaneous Files**

- **.gitignore:** Specifies files to be ignored in Git.
- **public/index.html:** The main HTML file.
- **public/manifest.json:** Metadata for the web application.
- **index.js:** Entry point for React application.
- **setupTests.js:** Setup file for tests.

## **Front-End Code Overview:**

### **❖ Terminal.js**

- **Location:** `cli-app/src/components/terminal.js`
- **Purpose:** Acts as the command-line interface for the application.
- **Functionality:**
  - **Input Handling:** Processes user input commands.
  - **Command Execution:** Interprets and executes commands like **help**, **about**, **fetch-price**, **upload**, **draw**, and **delete**.
  - **Display Output:** Presents the results or feedback from executed commands to the user.

### **❖ FileUploader.js (File Upload Functionality)**

- **Location:** `cli-app/src/FileUploader.js`
- **Purpose:** Manages the file upload process in the application.
- **Functionality:**
  - **File Selection:** Allows users to select files from their system using a file input element.

- **File Validation:** Ensures the selected file meets certain criteria (e.g., file type, size).
- **File Upload Handling:** Once a file is selected, it triggers a callback to process the file (like passing it to the server for further actions).

### ❖ **commandService.js (Command Processing)**

- **Location:** `cli-app/src/services/commandService.js`
- **Purpose:** Processes the commands entered in the terminal.

#### **Functionality:**

- **Command Parsing:** Breaks down the user input into command and arguments.
- **Command Switching:** Based on the command, it routes to the appropriate functionality like fetching cryptocurrency prices, uploading files, drawing charts, or deleting files.
- **Error Handling:** Provides error messages for unrecognized commands or invalid inputs.

### ❖ **chartService.js (Chart Drawing Functionality)**

- **Location:** `cli-app/src/services/chartService.js`
- **Purpose:** Handles the generation of charts based on CSV file data.

#### **Functionality:**

- **Data Processing:** Reads and processes the CSV file data.
- **Chart Rendering:** Utilizes the **recharts** library to render line charts based on the processed data.
- **Dynamic Chart Creation:** Generates charts dynamically based on the selected columns from the CSV file.

## **Summary**

Each component and service in the **cli-app** directory plays a specific role in facilitating the user interaction with the CLI application. From handling file uploads to processing commands and displaying charts, these files collectively enable a smooth and functional user experience in the CLI environment. The separation of concerns across different

files (like **terminal.js**, **FileUploader.js**, **commandService.js**, **chartService.js**) ensures a modular and maintainable code structure.