**Stock Trading Application Documentation**

**Introduction**

The Stock Trading Application is a web-based system designed to facilitate the management of trades and orders for a stock trading platform. It provides functionalities to create, update, and delete trades, as well as create orders based on existing trades. This document outlines the features, architecture, and usage instructions for the application.

**Features**

1. **Trade Management**:
   * Add, update, and delete trades.
   * View a list of existing trades with details like stock name, quantity, and listing price.
2. **Order Management**:
   * Create orders based on existing trades.
   * View a list of confirmed orders with details like stock name, quantity, price per unit, and type.

**Architecture**

* **Frontend**: Developed using ReactJS, a popular JavaScript library for building user interfaces.
* **Backend**: Built with Spring Boot, a framework for creating Java-based applications, including Spring MVC for handling RESTful APIs and Spring Data JPA for database access.
* **Database**: MySQL database is used to store trade and order data.

**Frontend Components**

1. **TradeListing**: Displays a list of trades and provides links to create orders for each trade.
2. **CreateTrade**: Allows users to add new trades by entering stock name, quantity, and listing price.
3. **CreateOrder**: Enables users to create orders based on selected trades by specifying quantity, price per unit, and type (buy/sell).
4. **OrderListing**: Displays a paginated grid view of confirmed orders.

**Backend Endpoints**

* **GET /trades**: Retrieves a list of existing trades.
* **POST /trades**: Creates a new trade.
* **GET /orders**: Retrieves a list of confirmed orders.
* **POST /orders/{tradeId}**: Creates a new order based on the specified trade ID.

**Usage**

1. **Installation**:
   * Clone the repository: **git clone <repository-url>**
   * Install frontend dependencies: **npm install**
   * Install backend dependencies (if applicable): **mvn install**
2. **Configuration**:
   * Configure the backend server URL in the frontend code (**axios** requests) to point to the correct backend endpoints.
3. **Running the Application**:
   * Start the frontend development server: **npm start**
   * Start the backend server (if applicable).
4. **Accessing the Application**:
   * Open a web browser and navigate to the URL where the frontend server is running (usually **http://localhost:3000**).
5. **Functionality**:
   * Use the Trade Listing page to view existing trades and create orders for each trade.
   * Use the Create Trade page to add new trades.
   * Use the Create Order page to create orders based on selected trades.
   * View confirmed orders on the Order Listing page.

**Additional Considerations**

* Ensure that the backend server is running and accessible before using the frontend application.
* Implement proper error handling and validation to enhance the user experience.
* Deploy the application to a production environment following security best practices.