

Stock Market App

A big trading company is using a mobile app to provide exchange services to their users. Each user is able to manage their orders and also see the top orders. The employees are able view market statistics.

On the server side at least the following details are maintained:

- Id - the internal order id. Integer value greater than zero.
- Name - the security name. A string of characters representing the security name.
- Price - the order price. An integer value.
- Quantity - the order quantity. An integer value.
- Type - the order type. A string of characters. Eg. "buy", "sell".
- Status - the status of the order. A string of characters. Eg. "open", "closed", etc.

The application should provide at least the following features:

- User Section (separate activity)
 - (1p) View open orders in a list sorted by type and value (price*quantity). Using **GET /orders** call, the user will retrieve the list of all open orders found in the system. If offline, the app will display an offline message and a way to retry the connection and the call. For each order the name, price, quantity and type are displayed.
 - (1p) Add an order. The user will be able add a new order, using a **POST /order** call, by sending all the order fields. Available online only.
 - (1p) Delete an order. By specifying the order id, using the **DELETE /order** call, the user will be able to remove an order. Available online only.
 - (1p) Close an order. By specifying the order object, using the **POST /close** call, the user will be able to close an order. Available online only.
 - (1p) See a list of all his closed orders. Everytime the user will close an order it will be persisted on the device. The list will display: name, value (price*quantity), type and status and will be sorted descending by value.
- Employee Section (separate activity)
 - (1p) The list of top 10 buy closed orders sorted ascending by price. The list will be retrieved using the **GET /buy** call, in this list along with the name, the app will display the price and quantity. Note that from the server you are retrieving an unsorted list containing all the buy orders.
 - (1p) The list of top 10 sell closed orders sorted ascending by price. The list will be retrieved using the **GET /sell** call, in this list along with the name, the app will display the value too. Note that from the server you are retrieving an unsorted list containing all the sell orders.

(1p) On the server side once a new order is added in the system, the server will send, using a websocket channel, a message to all the connected clients/applications with the new order object. Each application, that is connected, will display the received order details, in human form (not json text) using an in app "notification" (like snackbar or toast or a dialog or a message on the screen).

(0.5p) On all server operations a progress indicator will be displayed.

(0.5p) On all server interactions, if an error message is received, the app should display the error message using a toast or snackbar. On all interactions (server or db calls), a log message should be recorded.