



Geek Soft builds
REAL business.



Frontend Developer

Zadanie

Utwórz prostą aplikację, która będzie wyświetlała tabelę ze zleceniami przy użyciu frameworka Angular (najlepiej z TypeScript).

Wizualizacja tabeli ze zwiniętymi szczegółami zleceń:

Symbol	Order ID	Side	Size	Open Time	Open Price	Swap	Profit
▼ BTCUSD	6		0.4		31123.97	-0.0875	0.2141
▼ ETHUSD	2		0.02		3538.875	0	-0.0002
▼ TTWO.US	1		1		162.08	0	-0.00004

Wizualizacja tabeli z rozwiniętymi poszczególnymi grupami zleceń:

Symbol	Order ID	Side	Size	Open Time	Open Price	Swap	Profit
▲ BTCUSD	6		0.4		31123.97	-0.0875	0.2140
	1202940	BUY	0.01	04.08.2023 20:11:10	29172	-0.0023	0.0056
	1203380	BUY	0.2	10.08.2023 16:42:40	29470.83	-0.0463	0.1120
	1203382	BUY	0.1	10.08.2023 16:43:18	29454.28	-0.0231	0.0560
▲ ETHUSD	2		0.02		3538.875	0	-0.0002
	1235467	BUY	0.01	11.06.2024 13:15:22	3539.43	0	-0.0001
	1235468	BUY	0.01	11.06.2024 13:15:26	3538.32	0	-0.0001
▼ TTWO.US	1		1		162.08	0	-0.00004

Wymagania:

1. dane do tabeli powinny być ściągane za pomocą requestu HTTP z url:
<https://geeksoft.pl/assets/order-data.json>
2. tabela powinna grupować zlecenia wg pola "symbol"
3. na początku tabela powinna prezentować wyłącznie grupy symboli jako wiersze (3 widoczne wiersze: BTCUSD, ETHUSD, TTWO.US)
4. wiersz z informacją o grupie symboli powinien prezentować tylko wartości w kolumnach: symbol (oraz liczba zleceń z danego symbolu), open price, swap, profit, size (wartości powinny być zsumowane, open price oraz profit uśredniony)
5. po kliknięciu na dowolny wiersz z nazwą grupy rozwijamy pod spodem wiersze prezentujące pełne dane (w kolumnach) zleceń z danego symbolu (openTime powinien prezentować datę w formacie: 20.06.2024 18:30:22)
6. na końcu każdego wiersza powinien być button (lub ikona zamknięcie) który spowoduje usunięcie danego wiersza lub grupy zleceń



Geek Soft builds REAL business.



7. po kliknięciu na button/ikonkę zamknięcia powinien zostać wyświetlony komunikat o treści: "Zamknięto zlecenie nr xxx" (gdzie xxx to numer zlecenia lub jeśli zamknięto grupę orderów - numery zleceń kolejno po przecinku)
8. komunikat może być dowolny w postaci alertu/snackbara, można wykorzystać dowolną bibliotekę np: <https://material.angular.io/components/snack-bar/overview>
9. kolumna profit powinna być wyliczana ze wzoru:
 $(closePrice - openPrice) * multiplier * sideMultiplier / 100$, gdzie
 - multiplier: to potęga z podstawy 10 oraz wykładnika (2 dla symbolu BTCUSD, 3 dla ETHUSD oraz 1 dla TTWO.US)
 - sideMultiplier: 1 dla zlecenia z side równym BUY, w przeciwnym wypadku -1
10. (opcjonalnie) WebSocket - wartość "profit" można przeliczać zastępując we wzorze z pkt. 9 closePrice aktualną wartością/ceną (pole 'b') na podstawie danych z WebSocketów: <wss://webquotes.geeksoft.pl/websocket/quotes>

Aby otrzymywać aktualne ceny należy zasubskrybować się na zmiany wysyłając event według wzoru:

```
{
  'p': '/subscribe/addlist',
  'd': SYMBOLS_ARRAY
}
```

Serwer zwraca dane wg. schematu (w polu 'b' znajduje się cena, którą należy wykorzystać w przeliczeniach):

```
{
  "p": "/quotes/subscribed",
  "d": [
    {"s": "BTCUSD", "b": 67213.0, "a": 67258.85, "t": 1718112011},
    {"s": "ETHUSD", "b": 3538.45, "a": 3542.87, "t": 1718112011}
  ]
}
```

Aby zakończyć utrzymywanie nowych cen dla danego symbolu należy odsubskrybować się eventem:

```
{
  'p': '/subscribe/removelist',
  'd': SYMBOLS_ARRAY
}
```

Projekt graficzny:

- jest pełna dowolność w przygotowaniu html/css dla tabeli
- prosimy tylko o zastosowanie poniższej palety kolorów dla szablonu jasnego + ciemnego
- najlepiej gdyby aplikacja rozpoznała motyw urządzenia i przełączyła tabelę w odpowiedni motyw ciemny lub jasny



Geek Soft builds
REAL business.



Motyw jasny:

- główny kolor tła: `rgb(233, 237, 241)`
- kolor czcionek: `rgb(14, 15, 26)`
- kolor tła wiersza: `rgb(220, 225, 229)`
- kolor tła wiersza (stan hover): `rgb(201, 209, 216)`
- kolor profit: jeśli jest większy od 0 to: `rgb(60, 193, 149)` gdy jest mniejszy od 0 to: `rgb(249, 76, 76)`

Motyw ciemny

- główny kolor tła: `rgb(42, 56, 71)`
- kolor czcionek: `rgb(198, 210, 219)`
- kolor tła wiersza: `rgb(14, 15, 26, .25)`
- kolor tła wiersza (stan hover): `rgba(53, 71, 89, .5)`
- kolor profit: jeśli jest większy od 0 to: `rgb(60, 193, 149)` gdy jest mniejszy od 0 to: `rgb(249, 76, 76)`

Realizacja zadania musi być kompatybilna z najnowszymi wersjami przeglądarek Chrome, Opera, Safari.

Projekt będzie oceniany pod względem modułowości, poprawności pracy z frameworkiem Angular, użytych metodologii (wykorzystanie Typescript, sposób konstruowania klas itd.), kwestie wizualne nie będą oceniane.