

## 5 DANA U OBLACIMA - CHALLENGE

### ZADATAK

Nalazite se u ulozi *software* inženjera kompanije Levi9 koja razvija platformu za menjačnicu namenjenu za trgovinu *crypto* valutama.

Zaduženi ste za izradu dela platforme koji vrši procesiranje novih BUY ili SELL *order*-a. Na vama je da implementirate API-je neophodne za uspešno funkcionisanje ovog dela platforme. Više detalja o zadatku možete videti u nastavku teksta.

## KAKO FUNKCIONIŠE MENJAČNICA?

Menjačnica pre svega funkcioniše na osnovu ponude i potražnje, u ovom slučaju za crypto valutama.

Npr. ukoliko želite da kupite 1 BTC (Bitcoin) po ceni od 20000 \$, kreiraćete jedan BUY order sa datim iznosom i količinom. Sistem, odnosno menjačnica, će pronaći postojeće SELL order-e kod kojih je tražena cena manja ili jednaka traženoj ceni iz BUY order-a i upariće BUY i SELL order za raspoloživu količinu i kreirati *trade* rekord za svaki upareni order (dok se tražena količina iz BUY order-a ne ispuni).

## ORDERBOOK

*Orderbook* predstavlja skup *buy* i *sell order*-a za određenu valutu.

Neke od karakteristika *orderbook*-a:

- *Sell order*-i predstavljaju ponude za prodaju valute. Sastoje se od cene i količine. Poređani su po ceni, od niže ka višoj ceni.
- *Buy order*-i predstavljaju ponude za kupovinu valute. Takođe se sastoje od cene i količine. Poređani su po ceni, od više ka nižoj ceni.
- Svaka ponuđena cena (*buy/sell*) sadrži količinu koja se za tu cenu prodaje/kupuje.
- Više *order*-a (*buy/sell*) mogu da imaju istu cenu (npr. više korisnika za istu cenu želi da proda ili kupi valutu). U tom slučaju *orderbook* će prikazati akumuliranu količinu od svih *order*-a.

		SELL	BUY		
AMOUNT	L9	PRICE RSD	RSD PRICE	RSD PRICE	L9 AMOUNT
100		9	7		250
50		10	6		800
1000		11	5		150
250		15	1		5000

## PRIMERI UPARIVANJA/REALIZACIJE BUY I SELL ORDER-A

Primeri ordera i stanje orderbook-a posle njihovog izvršenja su prikazani ispod. Svaki sledeći order se izvršava na stanje orderbook-a od prethodnog primera.

### Primer 1. Amount: 50, price: 9, type: BUY

Pošto je postojala prodajna ponuda po ceni od 9RSD i dovoljne količine (100), kupovina će se odmah izvršiti.

SELL		BUY	
AMOUNT L9	PRICE RSD	RSD PRICE	L9 AMOUNT
100	9	7	250
50	10	6	800
1000	11	5	150
250	15	1	5000

*Stanje pre novog BUY order-a*

SELL		BUY	
AMOUNT L9	PRICE RSD	RSD PRICE	L9 AMOUNT
50	9	7	250
50	10	6	800
1000	11	5	150
250	15	1	5000

*Stanje posle procesiranja BUY order-a*

### Primer 2. Amount: 100, price: 7, type: SELL

Pošto je postojala kupovna ponuda po ceni od 7RSD i dovoljne količine (250), prodaja će se odmah izvršiti.

SELL		BUY	
AMOUNT L9	PRICE RSD	RSD PRICE	L9 AMOUNT
50	9	7	250
50	10	6	800
1000	11	5	150
250	15	1	5000

*Stanje pre novog SELL order-a*

SELL		BUY	
AMOUNT L9	PRICE RSD	RSD PRICE	L9 AMOUNT
50	9	7	150
50	10	6	800
1000	11	5	150
250	15	1	5000

Stanje nakon procesiranja SELL order-a

### Primer 3. Amount: 70, price: 8, type: BUY

Pošto ne postoji ponuda za prodaju po ceni od 8RSD, kreira se nova ponuda za kupovinu po toj ceni i količini 70.

SELL		BUY	
AMOUNT L9	PRICE RSD	RSD PRICE	L9 AMOUNT
50	9	7	150
50	10	6	800
1000	11	5	150
250	15	1	5000

Stanje pre novog BUY order-a

SELL		BUY	
AMOUNT L9	PRICE RSD	RSD PRICE	L9 AMOUNT
50	9	8	70
50	10	7	150
1000	11	6	800
250	15	5	150
		1	5000

Stanje posle procesiranja BUY order-a

### Primer 4. Amount: 200, price: 6, type: SELL

Već postoje ponude za kupovinu i po višoj ceni pa će se one redom izvršavati. U ovom slučaju prvo će se isprazniti ponuda po ceni od 8 RSD (količina 70), pa onda ostatak po ceni od 7 RSD (količina 130). Korisnik koji je bio spreman da proda i po nižoj ceni je zapravo dobio višu cenu od njegove ponude. Slično bi se desilo i da je u pitanju BUY order, samo što bi se prvo praznili SELL orderi.

SELL		BUY	
AMOUNT L9	PRICE RSD	RSD PRICE	L9 AMOUNT
50	9	8	70
50	10	7	150
1000	11	6	800
250	15	5	150
		1	5000

Stanje pre novog SELL order-a

SELL		BUY	
AMOUNT L9	PRICE RSD	RSD PRICE	L9 AMOUNT
50	9	7	20
50	10	6	800
1000	11	5	150
250	15	1	5000

Stanje posle procesiranja SELL order-a

#### Primer 5. Amount: 80, price: 9, type: BUY

Već postoji ponuda za prodaju po istoj ceni ali u nedovoljnoj količini. U ovom slučaju će se isprazniti ponuda za prodaju po ceni od 9 RSD i kreiraće se ponuda za kupovinu za ostatak (količina 30)

SELL		BUY	
AMOUNT L9	PRICE RSD	RSD PRICE	L9 AMOUNT
50	9	7	20
50	10	6	800
1000	11	5	150
250	15	1	5000

Stanje pre novog BUY order-a

SELL		BUY	
AMOUNT L9	PRICE RSD	RSD PRICE	L9 AMOUNT
50	10	9	30
1000	11	7	20
250	15	6	800
		5	150
		1	5000

Stanje posle procesiranja BUY order-a

## DOMEN

Osnovni domenski objekti su order, trade i orderbook. Ispod je specificirana njihova struktura, a takođe više detalja se nalazi kod opisa API-ja.

**Order** se sastoji od sledećih atributa:

- Id – identifikator order-a
- CurrencyPair – valuta na koju se odnosi trgovina. Fiksna vrednost BTCUSD (Bitcoin prema US dolaru)
- CreatedDateTime - vreme kreiranja order-a
- Type – tip order-a koji može biti BUY ili SELL
- Price – cena izražena u decimalnoj vrednosti (preciznost 2 decimalna mesta)
- Quantity - tražena količina izražena u decimalnoj vrednosti (preciznost 2 decimalna mesta)
- FilledQuantity - realizovana količina izražena u decimalnoj vrednosti (preciznost 2 decimalna mesta)
- Status – status order-a koji može biti OPEN ili CLOSED
- Trades – lista realizovanih trade-ova

**Trade** se sastoji od sledećih atributa:

- Id – identifikator trade-a
- BuyOrderId – identifikator buy order-a
- SellOrderId – identifikator sell order-a
- CreatedDateTime - vreme kreiranja trade-a
- Price – cena realizacije trade-a izražena u decimalnoj vrednosti (preciznost 2 decimalna mesta)
- Quantity - realizovana količina izražena u decimalnoj vrednosti (preciznost 2 decimalna mesta)

**Orderbook** se sastoji od sledećih atributa:

- BuyOrders – lista akumuliranih količina po svakoj ceni (za svaku unikatnu cenu iz BUY order-a koji su u OPEN statusu), sortirani po ceni opadajuće.
- SellOrders – lista akumuliranih količina po svakoj ceni (za svaku unikatnu cenu iz SELL order-a koji su u OPEN statusu), sortirani po ceni rastuće.

## PRAVILA ALGORITMA ZA REALIZACIJU ORDER-A

Sledeća pravila moraju biti ispunjena pri realizaciji order-a:

- Svi order-i su u OPEN statusu dok se ne realizuje ukupna količina. Nakon realizacije ukupne količine, order prelazi u CLOSED status.
- Za svaki upareni BUY i SELL order (ukupna ili delimična količina) kreira se trade rekord sa realizovanom količinom.
- Za nove BUY *order-e*, uparivanje se vrši sa SELL *order-ima* koji:
  - imaju OPEN status, tj realizovana količina (filledQuantity) je manja od tražene količine (quantity)
  - imaju cenu nižu ili jednaku od tražene cene iz BUY order-a
  - prvo će biti realizovani (upareni) SELL order-i sa najnižom cenom
    - ukoliko ima više SELL *order-a* sa istom cenom, prvo će se realizovati najstariji *order* (na osnovu *timestamp-a*)
- Za nove SELL *order-e*, uparivanje se vrši sa BUY *order-ima* koji:
  - imaju OPEN status, tj realizovana količina (filledQuantity) je manja od tražene količine (quantity)
  - imaju cenu višu ili jednaku od tražene cene iz SELL order-a
  - prvo će biti realizovani (upareni) BUY *order-i* sa najvišom cenom
    - ukoliko ima više BUY *order-a* sa istom cenom, prvo će se realizovati najstariji *order* (na osnovu *timestamp-a*)

## OPIS ZADATKA

Kao što je spomenuto na početku dokumenta, vaš zadatak je da napravite aplikaciju zaduženu za procesiranje BUY/SELL order-a.

Potrebno je implementirati API koji sadrži sledeće endpoint-e:

1. Procesiranje novih *order-a*
2. Preuzimanje *orderbook-a*
3. Preuzimanje *order-a* preko id-ja
4. Brisanje svih *order-a* i *trade-ova*

## SPECIFIKACIJA API-IJA

### 1. Endpoint za procesiranje novih order-a

Ovaj endpoint prihvata novi order i vrši procesiranje po algoritmu koji je opisan u dokumentu. Više detalja je takođe opisano u primerima uparivanja-realizacije order-a.

Path	/order
Request method	POST
Request body	<pre>{   "id": 1,   "currencyPair": "BTCUSD",   "type": "BUY",   "price": 20150.00,   "quantity": 0.3 }</pre>
Response status code	201 – Order uspešno kreiran 400 – Neuspešna validacija (nevalidan currency, type, negativne vrednosti za cene ili količinu)
Response body	<pre>{   "id": 1,   "createdDateTime": "2022-09-28T14:44:04.92656",   "currencyPair": "BTCUSD",   "type": "BUY",   "price": 20150.00,   "quantity": 0.3,   "filledQuantity": 0,   "orderStatus": "OPEN",   "trades": [] }</pre>

### 2. Endpoint za preuzimanje orderbook-a

Ovaj endpoint vraća trenutni orderbook kao što je opisano iznad u dokumentu.

Path	/orderbook
Request method	GET
Response status code	200
Response body	<pre>{   "buyOrders": [     {       "price": 2067.12,       "quantity": 22     },     {</pre>



	<pre>         "price": 2032.98,         "quantity": 66       },     ],     "sellOrders": [       {         "price": 2078.12,         "quantity": 20       },       {         "price": 2099.36,         "quantity": 23       }     ]   ] }</pre>
--	---

### 3. Endpoint za preuzimanje detalja o order-u preko id-ija

Detalji o order-u takođe sadrže i listu realizovanih trade-ova, vezanih za ovaj order.

Path	/order/\$id
Request method	GET
Response status code	200
Response body	<pre> {   "id": 1,   "createdDateTime": "2022-09-28T14:44:04.92656",   "currencyPair": "BTCUSD",   "type": "BUY",   "price": 20150.00,   "quantity": 0.3,   "filledQuantity": 0.3,   "status": "CLOSED",   "trades": [     {       "id": 10,       "buyOrderId": 1,       "sellOrderId": 5,       "timestamp": 1662638820000,       "price": 20150.00,       "quantity": 0.2     },     {       "id": 11,       "buyOrderID": 1,       "sellOrderID": 6,       "timestamp": 1662638820000,       "price": 20150.00,       "quantity": 0.1     }   ] }</pre>

#### 4. Endpoint za reset-ovanje stanja

Ovaj endpoint treba da obriše sve *order* i *trade* rekorde.

Path	/order/all
Request method	DELETE
Response status code	200

## TEHNIČKI ZAHTEVI REŠENJA

Tehnologije:

Obavezno: Java ili Javascript/Typescript

Obavezno: korišćenje build alata ili package manager-a  
(Maven/Gradle/NPM/Yarn/...)

Obavezno: rešenje zadatka mora da radi sa opensource free software rešenjima

Rešenje treba da sadrži:

- Izvorni kod
- Fajlove korišćene za testiranje
- Dokumentaciju u Readme.md fajlu:
  - Opis okruženja potrebnog da se uradi build
  - Kako se radi build
  - Primer kako se aplikacija pokreće
  - Listu korišćenih tehnologija sa kratkim opisom

Svaki prijavljeni takmičar treba kreirati nalog na GitHub platformi i kreirati repozitorijum koji će sadržati rešenje zadatka. Na repozitorijum je neophodno dodati korisnika mradojcic kao kolaboratora sa admin role-om (podešavanje se dodaje preko Settings / Collaborators and Teams / Add people).

Rešenjem se smatra sadržaj kreiranog Git repozitorijuma 17. novembra u 9 časova.

# PRAVILA BODOVANJA

- Tehničko rešenje – 40%
  - Subjektivni kvalitet rešenja i koda - 30%
  - Unit testovi - 5%
  - Readme - 5% – objašnjenje na koji način se builda i pokreće projekat
- Funkcionalni zahtevi - 60%
  - Test case 1 – 20% (javni test case)
    - Prikazan je u Gherkin tekstu ispod
  - Ostali test case-ovi - 40% (tajni)

Javni test case:

```
Scenario: Trade is created and BUY order is in status OPEN when BUY order finds SELL order with same price and less amount
  Given SELL order with amount of 100 and price 1045.6 USD is created
  When BUY order with amount of 105 and price 1045.6 USD is created
  Then BUY order will be in status OPEN and one trade will be created
  And SELL order will be in status CLOSED
  And Orderbook will contain BUY order with quantity 5 and price 1045.6
```