Univerzitet u Beogradu

Fakultet organizacionih nauka

Predmet: Cloud infrastruktura i servisi

Apache Kafka i NestJS:

Primena consumer producer paterna u ledger servisima

Mentor: Student:

Zorica Bogdanović Nikola Madić 4006/22

Beograd, 2023.

Sadržaj

[Sadržaj 2](#_Toc1223385675)

[Uvod 3](#_Toc140448789)

[Wallet-api 4](#_Toc141042002)

[Wallet-processor 5](#_Toc1298721225)

[Kafka 5](#_Toc677439060)

[Zookeeper 6](#_Toc1183741929)

[Kafdrop 6](#_Toc2111530801)

[Kafka kontejner 7](#_Toc1184525313)

# Uvod

U nastavku ce biti opisan domen na koji aplikacija primenom consumer producer paterna I kafka message broker-a se primenjuje:

Potrebno je kreirati sistem koji ce na optimalan nacin obradjivati veliki broj transakcija. Iz tog razloga jako je vazno da sistem bude zasticen od prevelikog broja transakcija u datom trenutku koje nije moguce obraditi.

Iz tih razloga sistem se sastoji iz 2 servisa koja medjusobno komuniciraju. Na taj nacin moguce je razdvajanje logike prihvatanja transakcije i stavljanja u red cekanja od same obrade.

To dovodi do mogucnosti kontrolisanja ‘stope obrade’, horizontalnim skaliranjem, konfiguracijom reda cekanja i slicno, dela aplikacije koji je zadužen za samu obradu transakcija.

Aplikacija se sastoji od sledećih elemenata:

1. Wallet-api
2. Wallet-processor
3. Apache Kafka
   1. zookeeper
   2. kafdrop
   3. kafka

# Wallet-api

Uloga wallet-api servisa je da obezbedi aplikacioni interfejs (API) pomoću koga je moguce poslati zahtev za obradom transakcija.

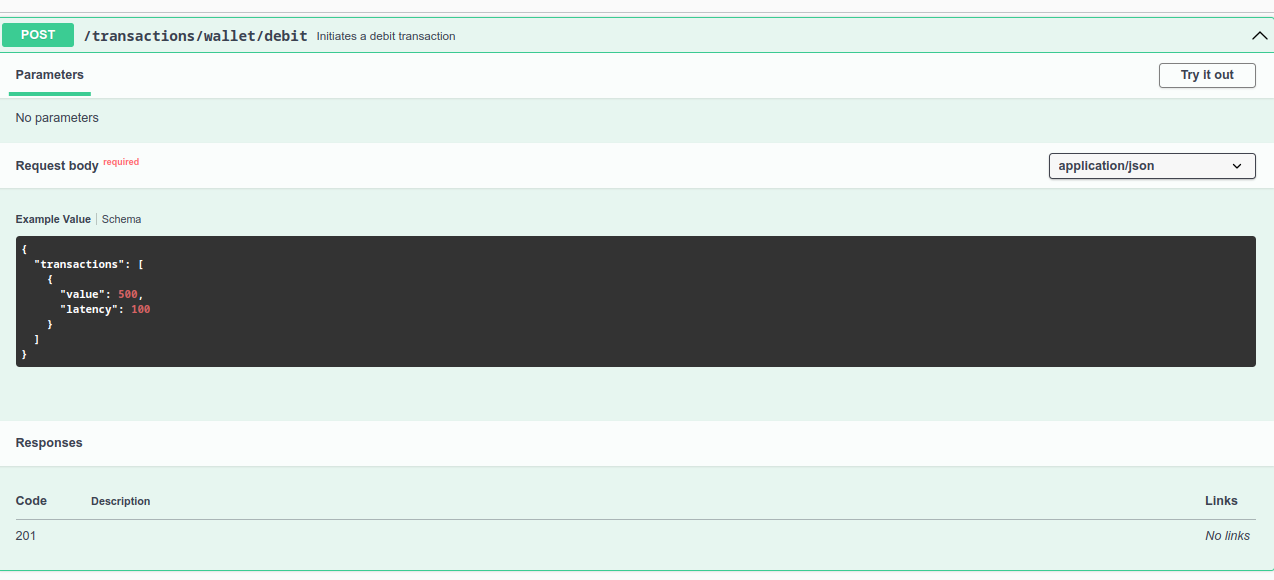
U konkretnom primeru korišćen je REST API.

REST (Representational State Transfer) je arhitekturni stil za projektiranje mrežnih aplikacija koji se oslanja na HTTP protokol. REST API omogućava komunikaciju između različitih softverskih sistema putem standardnih HTTP metoda poput GET, POST, PUT i DELETE, koristeći resurse (kao što su URL-ovi) za predstavljanje podataka. Osnovni principi REST-a uključuju jednostavnost, skalabilnost, nezavisnost od stanja i resursno usmjeravanje, što ga čini popularnim izborom za razvoj web usluga i aplikacija.

REST API wallet-api se sastoji iz jedne rute:

* Transactions/wallet/debit

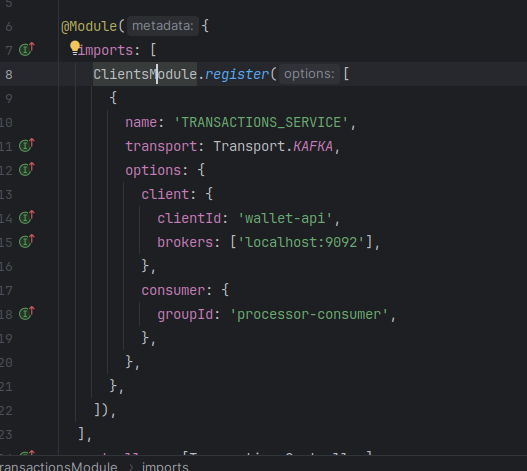
Kao sto se može videti iz OPEN API specifikacije:



Wallet-api zaduzen je za validaciju korisnickih zahteva, kao i stavljanje validnih zahteva na red cekanja za daljom obradom, za koju je zaduzen wallet-processor servis. Dakle wallet-api je ‘producer’, odnosno komponenta koja je zaduzena za ‘punjenje’ reda cekanja.

Wallet-api takodje ima zadatak da uzavisnosti od vrednosti i vremena potrebnog za obradu transakcije je na optimalan nacin rasporedi i upakuje u pakete, pre nego sto ih posalje dalje na obradu.

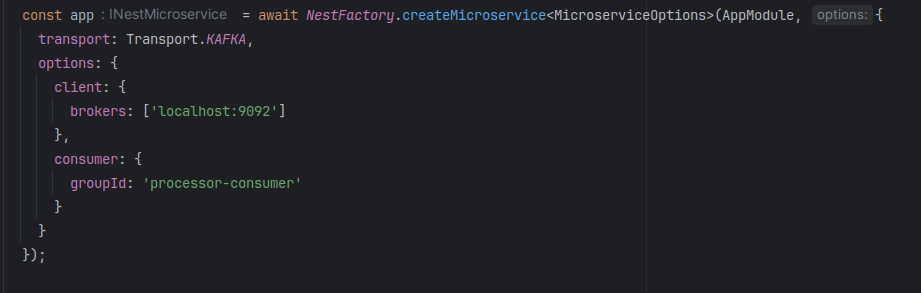
Konfiguracija wallet-api komunikacije sa kafka-om je realizovana na sledeci način:

* Korišćen je NestJS framework kao i ClientsModule iz '@nestjs/microservices' paketa
* Potrebno je proslediti kofiguraciju za ClientsModule kao na prikazanoj slici:

# Wallet-processor

Uloga wallet-processor servisa je da sama obrada transakcija. Dakle wallet-processor predstavlja ‘consumer’. Samo vreme obrade je simulirano timeout funkcijom i predstavlja potrebne kalkulacije, vreme upisavanja u bazu ili poziva ka ostalim servisima koji su potrebni da bi se transakcija obradila.

Povezivanje wallet-processor-a na sistem uradjeno je na sledeci nacin:

I 

Koriscen je NestJS framework, kao i njegova podrška za mikroservisne komponente.

Dovoljno je proslediti opcije `createMicroservice` fukciji kao na prikazanoj slici.

# Kafka

Apache Kafka je visoko skalabilan i distribuirani message broker sistem koji se koristi za obradu velike količine podataka u stvarnom vremenu. Kafka omogućava slanje, primanje i obradu poruka između različitih aplikacija i sistema te podržava velik broj klijenata na različitim platformama. Temelji se na publish-subscribe modelu i često se koristi za rješavanje problema povezanih s obradom logova, praćenjem događaja u stvarnom vremenu te sinhronizacijom i integracijom podataka iz različitih izvora. Kafka je postao ključna tehnologija za velike infrastrukture tokova podataka i streaming analitiku u mnogim organizacijama.

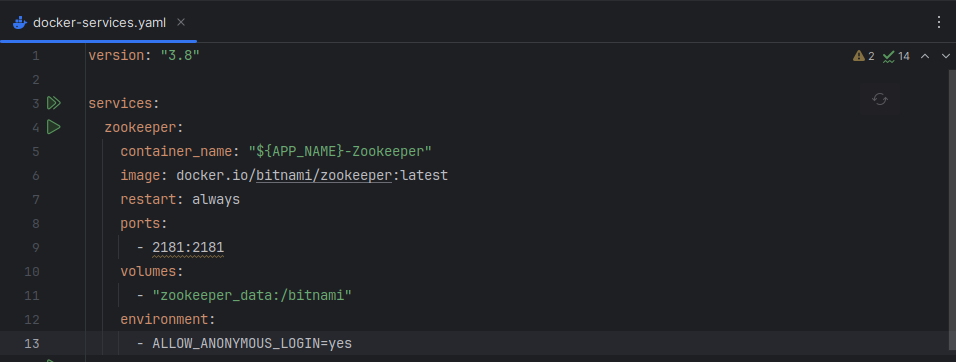
Kafka servis je omogucen primenom docker alata kao i docker-compose fajla na sledeci nacin:

Potrebne komponente:

## Zookeeper

ZooKeeper je ključna komponenta Apache Kafka ekosistema koja se koristi za upravljanje i održavanje distribuiranog Kafka sistema. Ova centralizirana usluga omogućava Kafka klasterskim članovima da koordiniraju svoje aktivnosti, održavaju konfiguraciju, praćenje statusa čvorova i izbore lidera.

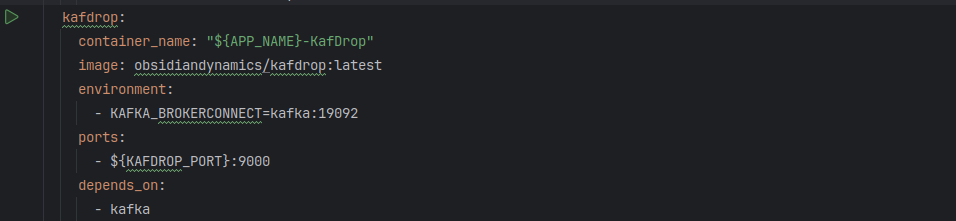
Konkretno, ZooKeeper se koristi za održavanje metapodataka o Kafka klasteru, poput informacija o temama, particijama i liderima particija. Također pomaže u otkrivanju i upravljanju čvorovima u klasteru te osigurava konzistentnost i integritet podataka. Zahvaljujući ZooKeeperu, Kafka može pružiti visoku dostupnost i pouzdanost u distribuiranim okruženjima, čineći ga pouzdanim rješenjem za streaming podataka i obradu događaja u stvarnom vremenu.



## Kafdrop

Kafdrop je alat za upravljanje i nadzor Apache Kafka klastera koji pruža grafički interfejs za pregled, praćenje i upravljanje Kafka temama, particijama i porukama. Ovaj alat omogućava korisnicima da vizualizuju stanje Kafka klastera, pregledaju i traže poruke te prate performanse Kafka brokera.

Kafdrop je koristan za brži uvid u Kafka klaster i olakšava administraciju Kafka okruženja. Omogućuje korisnicima da pregledavaju metapodatke o temama, pratite consumer-e i producer-e, te istražuju poruke u stvarnom vremenu. Ovo grafički interfejs često olakšava rad s Kafka klasterom za timove koji nisu upoznati s komandnom linijom ili traže brži način za pregled i praćenje Kafka sistema.



## Kafka kontejner

