

1. Prezentarea proiectului

Scopul proiectului este gestionarea evenimentelor și cumpărarea biletelor pentru spectacolele de la operă. Acest proiect are o arhitectură bazată pe microservicii, constând în componente esențiale ***ticket-service***, ***order-service*** și ***stock-service***. Pentru a facilita comunicarea între aceste microservicii, am creat un serviciu de discovery și anume ***discovery-server***. Aceasta este ca o instanță centrală care indică fiecărui microserviciu locația celorlalte în cadrul sistemului. De asemenea, am implementat un microserviciu numit ***api-gateway***. Acesta preia requesturile de la utilizatori și le direcționează către microserviciul corespunzător. Astfel, api-gateway și discovery-server contribuie la o integrare și o interacțiune eficiente între microservicii în cadrul proiectului. Pentru monitorizarea și urmărirea performanței sistemului am utilizat ***Zipkin***.

Fiecare microserviciu comunica cu propria sa bază de date. De exemplu, ***ticket-service*** utilizează o instanță de ***MongoDB***, în timp ce ***order-service*** și ***stock-service*** se conectează la instanțe distințe de ***Postgresql***.

Am folosit ***Thymeleaf*** pentru frontend-ul fiecărui microserviciu principal, reusind astfel să integrez și să afișez datele în paginile web.

2. Prezentarea microserviciilor principale

- ticket-service

Administratorul de sistem poate să introducă noi tipuri de bilete în sistem, alături de detalii despre evenimentul pentru care este alocat biletul, tipul de bilet, descrierea locului ales și prețul unui astfel de tip.

- stock-service

Administratorul de sistem poate să introducă noi stocuri de bilete în sistem pentru evenimentele găzduite de operă, alături de detalii despre evenimentul, tipul de bilet și numărul de bilete pus în vânzare.

- order-service

Prin intermediul acestuia se pot plasa comenzi de bilete pentru evenimentele disponibile în sistem, alegând pentru ce eveniment se cumpără bilet, care este tipul de bilet dorit și câte astfel de bilete vor fi cumpărare.

Acest microserviciu comunica cu stock-service pentru a lista evenimentele pentru care există bielete disponibile în sistem, precum și pentru a actualiza stock-ul în urma cumpărării unui anumit număr de bielete.

Acest microserviciu comunica cu ticket-service pentru a obține prețul aferent unui anumit tip de bilet pentru un anumit spectacol.

Am utilizat WebClient.Builder pentru a facilita comunicarea între microservicii. Un exemplu concret este prezent în metoda de mai jos, care obține toate evenimentele disponibile în sistem. Am construit un obiect WebClient pentru a face o cerere GET către endpoint-ul "http://stock-service/api/stock/all". Rezultatul este apoi transformat într-un array de obiecte StockResponse.

```
1 usage
public List<StockResponse> getAllEvents() {
    StockResponse[] stockResponseArray = webClientBuilder.build().get() RequestHeadersUriSpec<capture of ?>
        .uri( uri: "http://stock-service/api/stock/all" ) capture of ?
        .accept(MediaType.APPLICATION_JSON)
        .retrieve() ResponseSpec
        .onStatus(HttpStatus::isError, clientResponse -> {
            return clientResponse.bodyToMono(String.class)
                .flatMap(errorBody -> Mono.error(new RuntimeException("Error response: " + errorBody)));
        })
        .bodyToMono(StockResponse[].class) Mono<StockResponse[]>
        .onErrorResume(Throwable -> {
            System.err.println("Error during HTTP request: " + throwable.getMessage());
            return Mono.just(new StockResponse[0]);
        })
        .block();

    return stockResponseArray != null ? Arrays.asList(stockResponseArray) : new ArrayList<>();
}
```

3. Api-gateway:

Configurația API Gateway indică modul în care acesta direcționează traficul către microservicii în funcție de rute specifice. Pentru exemplul de mai jos avem *id rută*: stock-service , *uri destinație* care redirecționează cererile către instante ale serviciului „stock-service” folosind load balance, și condiția de rutare: se aplică doar cererilor cu prefixul „/api/stock/**”.

```
## Stock Service Route
spring.cloud.gateway.routes[4].id=stock-service
spring.cloud.gateway.routes[4].uri=lb://stock-service
spring.cloud.gateway.routes[4].predicates[0]=Path=/api/stock/**
```

4. Discovery-server

În imaginea următoare este evidențiat modul în care Eureka face gestionarea și descoperirea serviciilor.

Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - DESKTOP-192.168.1.10:api-gateway
ORDER-SERVICE	n/a (1)	(1)	UP (1) - DESKTOP-192.168.1.10:order-service:8081
STOCK-SERVICE	n/a (1)	(1)	UP (1) - DESKTOP-192.168.1.10:stock-service:8090
TICKET-SERVICE	n/a (1)	(1)	UP (1) - DESKTOP-192.168.1.10:ticket-service:8083

General Info

Name	Value
total-avail-memory	78mb
num-of-cpus	4
current-memory-usage	65mb (83%)
server-upptime	00:05
registered-replicas	
unavailable-replicas	
available-replicas	

Instance Info

Name	Value
------	-------

5. Plasarea unei comenzi de bilete

Event Code	Ticket Type	Availability	Buy tickets
Lacul Lebedelor	A	true	<button>Buy tickets</button>
Lacul Lebedelor	B	true	<button>Buy tickets</button>
Lacul Lebedelor	C	true	<button>Buy tickets</button>
Spargatorul de nuci	B	true	<button>Buy tickets</button>
Traviata	C	false	<button>Sold Out</button>
Traviata	B	true	<button>Buy tickets</button>
Spargatorul de nuci	A	true	<button>Buy tickets</button>

'As an art form, opera is a rare and remarkable creation.' - Luciano Pavarotti

Bucharest, Romania

New Stock Info

Event Code:

Ticket Type:

Quantity:

Add order

Coiculescu Veronica-Ionela Grupa 505

Events - Buy a ticket All Orders

Available Orders

Order Number	Event Code	Ticket Type	Quantity	Total Price (RON)
4	Traviata	C	2	240.00
5	Traviata	B	3	300.00
6	Traviata	B	40	4000.00
7	Spargatorul de nuci	A	2	500.00
8	Lacul Lebedelor	C	8	1200.00

'As an art form, opera is a rare and remarkable creation.' - Luciano Pavarotti

📍 Bucharest, Romania

6. Adaugarea unui tip de bilet in sistem

Tickets Add New Ticket

Available Tickets

Ticket Id	Ticket Type	Event Code	Details	Price	Delete
65a2dc8904195a21ae541b2f	A	Lacul Lebedelor	Cele mai bune locuri pentru vizionarea spectacolelor	250	<button>Delete</button>
65a2f2a07448926727f0c115	B	Lacul Lebedelor	Aveti acces la locuri bune	200	<button>Delete</button>
65a2f2bb7448926727f0c116	C	Lacul Lebedelor	Aveti acces la locuri bune	150	<button>Delete</button>
65a2f2d67448926727f0c117	A	Spargatorul de nuci	Cele mai bune locuri pentru vizionarea spectacolelor	250	<button>Delete</button>
65aa8bba6e0a635b5a72cbbb	A	Oedip	Aveti acces la locuri bune	150	<button>Delete</button>
65aa91eed114b90248d77f3a	B	Oedip	Aveti acces la locuri bune	230	<button>Delete</button>

Add new ticket

'As an art form, opera is a rare and remarkable creation.' - Luciano Pavarotti

📍 Bucharest, Romania

7. Adaugarea unui nou stock de bilete in sistem

Stocks Add New Stock

Available Stocks

Stock Id	Event Code	Ticket Type	Quantity	Delete
1	Lacul Lebedelor	A	50	<button>Delete</button>
2	Lacul Lebedelor	B	100	<button>Delete</button>
6	Spargatorul de nuci	B	50	<button>Delete</button>
10	Traviata	C	0	<button>Delete</button>
11	Traviata	B	57	<button>Delete</button>
5	Spargatorul de nuci	A	98	<button>Delete</button>
12	Traviata	A	120	<button>Delete</button>
3	Lacul Lebedelor	C	92	<button>Delete</button>

'As an art form, opera is a rare and remarkable creation.' - Luciano Pavarotti

📍 Bucharest, Romania