T.C.
EGE ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

TICKET CLOUD MICROSERVICE FİNAL PROJESİ

MİKROSERVİSLER DERSİ

Hazırlayan

05170000804 - Burhan CABİROĞLU
burhancabiroglu97@gmail.com

Tarih

30 TEMMUZ 2021
İZMİR

TICKET CLOUD MICROSERVICE FİNAL PROJESİ

İÇİNDEKİLER

1. GİRİŞ:

Proje senaryosu; kullanıcı bilgilerini içeren bilet satış sisteminin gerçekleştirilmesi amaçlanmıştır. Buna göre sisteme elektronik posta, ad soyadı, şifre gibi bilgilerle kaydolabilen bir kullanıcının ardından bilet satış sisteminden bilet alabilmesi gerçekleştirilmiştir. Bilet almak isteyen kullanıcın sistemde numara kaydı bulunmak zorundadır. Eğer kayıt yoksa servislerimiz kullanıcıya hata uyarısında bulunur. Başarıyla kaydolmayı gerçekleştiren kullanıcının elektronik posta sistemine bilet bilgileri iletilir.

2. KULLANILAN TEKNOLOJİLER

Java:

Performans, etkinlik ve kullanım kolaylığı olarak en etkin programlama dilinin Java programlama dili olduğunu araştırdık. Bu sebeple projemizde Java programlama dili kullanmayı tercih ettik.

Spring Boot:

Spring, Java ve .NET için geliştirilmiş açık kaynak kodlu bir framework'tür. JavaEE uygulamalarını geliştirmeyi kolaylaştırır. Spring'i "frameworks of framework" olarak düşünebiliriz çünkü Struts, Hibernate, Tapestry, EJB, JSF gibi frameworklerin kullanımını destekleyen bir frameworktur.

Spring boot kullanım kolaylığı, hız ve performansı yüksek bir frameworktur. Günümüzde bankalar gibi güvenliğin önemli olduğu işletmeler, netflix gibi cloud stream platformları tarafından tercih edilmektedir. Önemli bir teknoloji olması ve kullanımının çok etkin olması sebebi ile tercih edilmiştir.

PostgreSql:

Oldukça popüler bir Relational Database Management System(RDMS) olmasından ve performansı yüksek olması sebebi ile projemizde bilet bilgilerini bu veri tabanı sisteminde tuttuk.

MongoDB:

NoSql veritabanı olarak üstün performanslı ve platform desteği çok güçlü olması sebebiyle projemizin kullanıcı verileri veri tabanında bu teknolojiyi kullandık.

RabbitMQ:

Open Source olan bir mesaj kuyruk sistemidir. Yani bir uygulamadan bir mesajı alıp bir başka uygulamaya sırası geldiğinde ileten sistemdir. Bu sistemi bir postane yahut kargo firması gibi düşünebiliriz. Nasıl ki bir mektup postaneye yahut bir ürün kargoya verildiğinde zamanı geldimi ilgili adrese teslim edilirlerse işte aynı işlemi RabbitMQ'da gerçekleştirmekte ve kendisine verilen mesajı doğru zamanda ilgili tüketiciye göndermektedir.

Spring Cloud Gateway:

Spring Framework'unun kullanıcıya sunmuş olduğu sistemlerden biri olan "Spring Cloud Gateway API" katmanının üzerinde hassas bir kontrol sağlayarak; güvenliği sağlama, yönlendirme, hizmetleri gizleme, yükü azaltma gibi birçok şeyle ilgilenir.

Netflix Eureka Server and Client:

Netflix Eureka ne işe yarar dediğimizde Eureka Servera Register olan Eureka clientler servislerin makina adı ve bağlantı noktasına ihtiyaç olmaksızın birbirleriyle iletişim kurmasını sağlar.

3. MICROSERVICE MİMARİSİ

Account Service:

Bu servisin portu 8001. Kullanıcı kayıt işlemlerini gerçekleştirir. Ayrıca ticket-service ile feign client üzerinden rest haberleşmesini gerçekleştirir. MongoDB veri tabanına bağlıdır. Kullanıcı tam adı, elektronik postası, şifresi, doğum yılı gibi sahalar ile sisteme kaydolabilir. Oluşturulma tarihi ve id sahaları otomatik olarak üretilir. Bu servisin dökümantasyonu aşağıdaki gibidir.

GET http://localhost:8001/account/findAll

GET http://localhost:8001/account/*:userid*

POST http://localhost:8001/account/

DELETE http://localhost:8001/account/*:userid*

PUT http://localhost:8001/account/*:userid*

“*userid*” alanları kullanıcı numarasıdır. POST Request, bazı body json parametreleri içerir. Bunlardan bir örnek aşağıdaki gibidir.

```
{
"id":"123456789",
"fullName":"BurhanCabiroglu",
"email":"geniusxburhan@gmail.com",
"password":"deneme",
"birthDate": "2021-02-02",
"createdAt": "2021-02-02"
}
```

Ticket Service:

Bu servisin portu 8002. Bu servis kullanıcıların bilet işlemlerinden sorumludur. Biletlerin kaydedilmesini PostgreSQL veri tabanı ile gerçekleştirir. Veri tabanı sahaları id, description, notes, accountId, priorityType, statusType gibi alanlardır.

Bileti kaydetmeden önce **feign client** ile account-service’e kullanıcı sorgusu yapar. Bu işlemi de **REST** haberleşmesi ile gerçekleştirir. Kayıt işlemi başarılı olduğunda **RabbitMQ** ile notification-service’e mesaj gönderir. Dökümantasyonu aşağıdaki gibidir.

GET http://localhost:8002/ticket/findAll

GET http://localhost:8002/ticket/*:userid*

POST http://localhost:8002/ticket/

DELETE http://localhost:8002/ticket/*:userid*

PUT http://localhost:8002/ticket/*:userid*

“userid” alanları ticket numarasıdır. POST Request, bazı body json parametreleri içerir. Bunlardan bir örnek aşağıdaki gibidir.

```
{
    "id":"9876543210",
    "description":"sample",
    "notes":"hello",
    "accountId":"123456789",
    "priorityType":"LOW"
}
```

Notification Service:

Servis portu 8003. Bu servisin amacı kullanıcılara bildirim göndermektir. RabbitMQ ile ticket-service'den mesaj alır. Bu mesajın içeriğini kullanıcının elektronik posta adresine gönderir.

Config Server:

Adından kolayla anlaşılabileceği servislerin port numaralarını ve bazı bilgilerini tutar. Açıkçası mikroservis mimarisine aykırı olduğunu düşünsem de spring boot geleneği olarak kullanmayı tercih ettim.

Communication Service:

Kendisi bir servis olmasa da servisler arası iletişimi sağlar. Özellikle ticket-service ve account-service arasındaki REST haberleşmesini gerçekleştirme üzere geliştirilmiştir. Feign Client adlı yapıdan türetilmiştir.

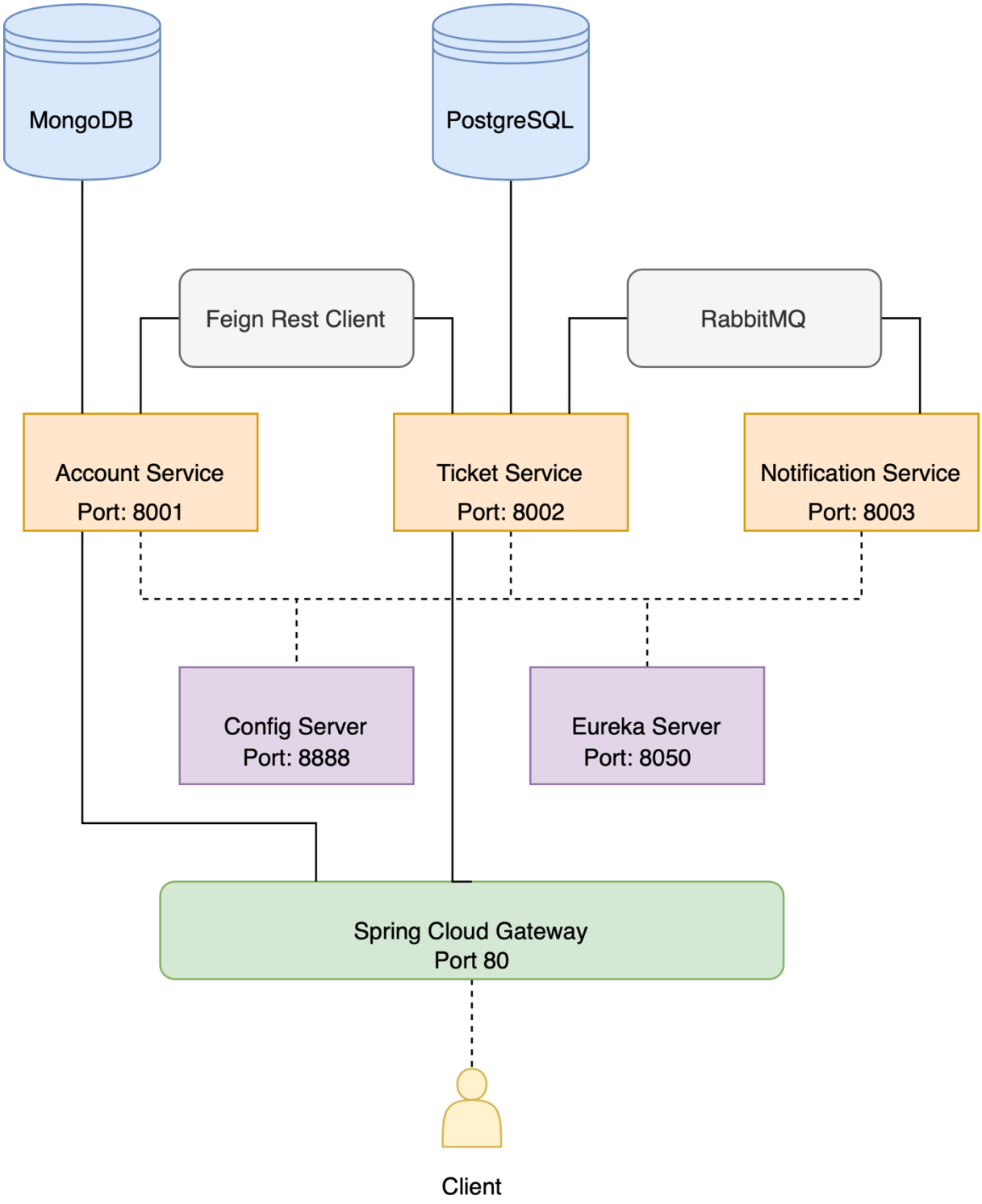
Eureka Server:

Netflix ekibin ürettiği eureka serverdan türetilmiş bir server yapısıdır. Bağlanan servislerin görüntülenmesi, kolayca yönetimine katkıda bulunmak gibi işleri gerçekleştirir. Port numarası 8050'dir.

Spring Cloud Gateway:

Servislerin dış dünya ile haberleşmesini, yönetimini ve güvenliğini sağlayan yapıdır. Projemizde sadece ticket-service ve account-service'i dış dünya ile 80 portu yani varsayılan internet portu ile bağladık.

MİMARİNİN GENEL GÖRÜNÜMÜ



Şekil 1. Ticket Cloud Mikroservis Mimarisi

KAYNAKLAR

- **Rapor Kaynakları:**

https://www.gencayyildiz.com/blog/rabbitmq-nedir-ne-amaca-hizmet-etmektedir/
https://medium.com/@buseodaci/spring-framework-nedir-fe46c9ce3456
https://www.argenova.com.tr/spring-nedir
https://metinalniacik.medium.com/spring-cloud-spring-boot-ve-eureka-server-kullanarak-microservice-örneği-1775cf220b2d

- **Proje Kaynakları:**

https://spring.io
https://youtu.be/k0A5bRfk0QM
https://youtu.be/M27XQA_OgXI
https://youtu.be/jOujw6bfii8
http://celikozgur.net/netflix-eureka-server-client/
https://youtu.be/9SGDpanrc8U