

KRİPTO PARA FİYAT UYGULAMASI

STAJ PROJESİ

Muhammet Ali BAKINÇ
Opnovo
Kocaeli Üniversitesi
Bilgisayar Mühendisliği Bölümü

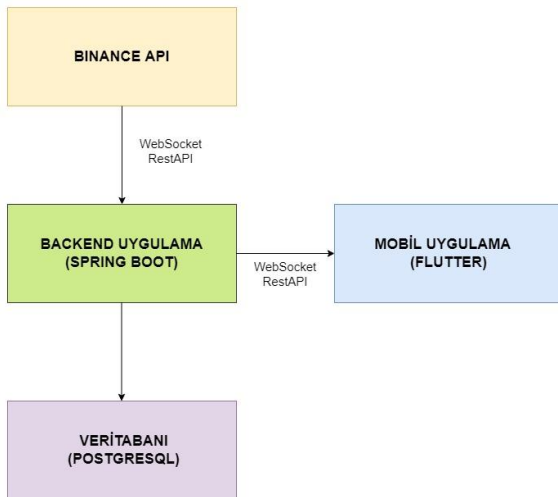
Özet

Bu doküman Staj süresince geliştirdiğim proje için çözümümü açıklamaya yönelik oluşturulmuştur. Dokümanda projenin tanımı, çözüme yönelik yapılan araştırmalar, kullanılan yöntemler, proje hazırlanırken kullanılan geliştirme ortamı, kod bilgisi ve kurulum gibi programın oluşumunu açıklayan başlıklara yer verilmiştir. Doküman sonunda projemi hazırlarken kullandığım kaynaklar ve proje derlenirken dikkat edilmesi gereken hususlar bulunmaktadır.

1. Proje Tanımı

Projede planlanan kripto para verilerini listeleyen ve belirli aralıklarla güncelleyen bir mobil uygulama ve verileri gönderen bir backend uygulaması geliştirmektir. Server üzerinde çalışan Spring Boot uygulaması Binance API'den aldığı verileri Flutter mobil arayüze göndermektedir.

2. Çözüm Mimarisi



Projede Spring Boot ile yazılan uygulama, Binance API'nin sağladığı restApi ve websocket bağlantıları ile güncel verileri almaktır. Flutter ile geliştirilen mobil uygulama arayüzünden backend uygulamasına yapılan istekler sonucu backend verileri gönderir ve istek sonuçlarını PostgreSQL üzerine kaydeder. Backend uygulaması web üzerinde bir Linux hostingte servis olarak çalışır.

3. Geliştirme Ortamı

Projemi Windows ve Linux sistemlerde, *IntelliJ Idea* ve *Android Studio* üzerinde geliştirip derledim.

4. Kod Bilgisi

4.1. Kullanılan Teknolojiler

Spring Boot - Java

Backend uygulamasının temelini Java dili ve Spring Boot framework ile geliştirdim.

JPA - PostgreSQL

Java JPA teknolojisi ile veri tabanına kaydedilecek ve okunacak nesnelerin kolayca eşleştirilmesini sağladım. İşlemleri veri tabanı komutlarına ihtiyaç duymadan fonksiyonlar vasıtasıyla rahat ve okunabilir şekilde yapmış oldum.

Slf4j - Logback

Yapılan işlemlerin ve olası hataların hem test hem üretim aşamalarında kayıtlarının tutulması için kullandım. Slf4j ile Logback

işlemlerinin daha kolay ve okunabilir olmasını sağlamış oldum.

Flutter - SockJS

Mobil arayüz için kullandığım framework'tür. Backend üzerinde sorgu yapmak için http client'ler ve websocket bağlantısı için SockJS kütüphanesini kullandım.

4.2. Class'lar ve Görevleri

WsConfig

Backend ile mobil uygulama arasında veri akışını sağlayacak websocket bağlantısının konfigürasyon bilgilerini içerir.

ApiController

RestApi ile yapılan sorguları gerekli işlemlerin yapılacağı sınıfları kullanarak sonuçlarını döndüren sınıftır.

BinanceApiService

Binance restApi ile yapılacak sorguları yapan, sorgu sonuçlarını veritabanına kaydeden sınıftır.

BinanceWebSocketService

Binance websocket ile gelen verileri işleyen, sonuçlarını veritabanına kaydeden sınıftır. Aynı zamanda anlık verileri backend uygulamasının websocket'ine göndererek mobil uygulamaya ulaşmasını sağlar.

5. Kurulum

Programın backend kaynak kodunda bulunan application-properties'in local, test ve prod profilleri için gerekli veritabanı bilgileriniz ile güncelleyiniz. İsteğe bağlı Logback konfigürasyon dosyasını güncelleyiniz. Uygulamanın jar paketini oluşturunuz. Linux hostingine WinSCP vb. bir tool ile paketi yükleyiniz. Paket ile aynı klasör içerisine ana application-properties dosyanızı istediğiniz profili seçerek kaydediniz. Bir sonraki bölümdeki adımları uygulayarak uygulamayı sürekli servis olarak çalıştırabilirsiniz.

Mobil uygulamanın kaynak kodunda bulunan api linkini kendinizinkiyle değiştirin ve APK olarak çıkartın.

5.1. Uygulamanın Servis Olarak Çalıştırılması

Linux üzerinde /etc/systemd/system adresine gidiniz. Servisinizin adı ile .service uzantılı bir dosya oluşturun ve aşağıdaki metni düzenleyerek içine yapıştırınız.

```
[Unit]
Description=Employee Spring Boot
application service
```

```
[Service]
User=muhammet
ExecStart=/home/muhammet/software/api/cryptoApp.jar
SuccessExitStatus=143
```

```
TimeoutStopSec=10
Restart=on-failure
RestartSec=5
```

```
[Install]
WantedBy=multi-user.target
```

ExecStart kısmına jar paketini yüklediğiniz adresi veriniz. User kısmına da kullanıcı adınızı giriniz. Ardından aşağıdaki komutları kullanarak servisi başlatma ve durdurma işlemlerini yapabilirsiniz.

```
sudo systemctl start app.service
sudo systemctl stop app.service
sudo systemctl restart app.service
```