

# Házi feladat specifikáció

Üzleti Intelligencia

Félév: 2025 ősz

**Kriptovaluta      és      részvény**  
**adatok elemzése**

Sőrés Barna - (CASI6C)

[sores.barni@gmail.com](mailto:sores.barni@gmail.com)

## Bemutatás

A pénzügyi piacok napjainkban jelentős szerepet töltenek be a gazdaságban, legyen szó digitális eszközökről vagy hagyományos részvényekről. A házi feladat kriptovaluta és részvény adatokat dolgoz fel annak érdekében, hogy hasznos és könnyen értelmezhető információkat nyerjünk, például az árfolyamok alakulásáról, a legnagyobb napi változásokról, a forgalomról és a volatilitásról. Az elemzések célja, hogy áttekintést adjanak a piacok mozgásairól és a különböző eszközök teljesítményéről.

## Főbb funkciók

Az alkalmazás funkcióinak bemutatása részletesen:

### Választott adatforrás(ok):

- Kriptó árfolyamok (OHLC, volume): CoinGecko vagy CoinPaprika API.
- Részvény árfolyamok (napi OHLC, volume): Twelve Data API vagy Yahoo Finance (yfinance Python csomag segítségével).

### Adattárolás megvalósításának leírása:

Az adatgyűjtés és tárolás több rétegen történik, igazodva az üzleti intelligencia rendszer feldolgozási folyamataihoz.

#### • RAW réteg

Az API-kból érkező kriptovaluta és részvény árfolyam adatok eredeti JSON formában kerülnek mentésre egy objektumtárra (MinIO).

Az adatok naponta, forrásonként külön mappákban kerülnek mentésre JSON formátumban, így biztosítva a visszakereshetőséget és az eredeti állapot megőrzését.

#### • STAGE réteg

A nyers adatok előfeldolgozása (tisztítás, struktúrlás, típuskezelés, alap transzformációk) történik.

Ezek az ideiglenes, részben normalizált adatok PostgreSQL-ben kerülnek tárolásra egy külön stage sémában.

#### • DWH réteg

A végeleges, egységesített, elemzésre előkészített adatok PostgreSQL-ben találhatók a dwh sémában.

Itt kerülnek kialakításra a lekérdezéshez szükséges táblák és aggregációk, például napi zároár, hozam, forgalom és volatilitás.

#### • Cache réteg

A legfrissebb piaci adatok gyors eléréséhez Redis kerül felhasználásra, rövid ideig tárolva az aktuális árfolyamokat.

Az adatbetöltést és rétegek közötti mozgatást automatizált ETL folyamat végzi Apache NiFi segítségével.

### **Megvalósítandó ETL/ELT job-ok:**

1. ETL: Kriptovaluta- és részvényadatok lekérése az API-król, majd a nyers JSON adatok mentése a RAW rétegbe (MinIO).  
A folyamat paraméterezhető (pl. eszközlista, dátumtartomány), és óránként automatikusan fut az aktuális árfolyamok frissítésére.
2. ETL: Az adatok tisztítása és előfeldolgozása, például típuskonverziók, hiányzó értékek kezelése és duplikációk eltávolítása.  
A transzformáció célja az adatok egységes formátumra hozása és a hibás rekordok kiszűrése a további feldolgozás előtt.
3. ETL: A kripto- és részvényadatok egységes formátumba hozása és összekapcsolása, hogy összehasonlíthatóak legyenek az elemzésekben.  
Az adatösszekapcsolás napi ütemezéssel történik, azonos kulcs (ticker/symbol és dátum) alapján.
4. ETL: Időszakos (napi, heti, havi) aggregált mutatók előállítása, például hozam, forgalom és volatilitás számítása.  
A transzformációk SQL-alapúak, automatikusan futnak a frissített adatok betöltése után.
5. ETL: A feldolgozott adatok betöltése az adattárházba (PostgreSQL), ahol a kimutatásokhoz és vizualizációkhöz szükséges táblák találhatók.  
A betöltés ütemezett (napi vagy heti) folyamat, amely frissíti az elemzési táblákat és előkészíti az adatokat a riportokhoz.
6. ETL: A legfrissebb árfolyamadatok cache-elése Redis-ben a gyors lekérdezésekhez és az aktuális értékek megjelenítéséhez.  
Az adatok óránként frissülnek, rövid ideig tárolódnak, így biztosítva a valós idejű megjelenítést a dashboardokon.

### **Megjelenítési réteg:**

Az adatok megjelenítése Grafana dashboardokon történik, ahol a feldolgozott és aggregált adatok különböző nézetekben vizualizálhatók.

A riportok dinamikusak, lehetőséget biztosítanak szűrésre, rendezésre és lefúrásra az eszközök, időszakok vagy adatforrások alapján.

Reportok:

- Piaci áttekintő dashboard: fő KPI-k megjelenítése (napi hozam, forgalom, volatilitás, árfolyamváltozás).
- Top Movers riport: a legnagyobb napi árfolyam-emelkedéseket és eséseket mutató eszközök listája, szűrhető kategória szerint (részvény, kriptó).
- Eszközprofil riport: egy kiválasztott eszköz részletes időszoros árfolyamgrafikonja, forgalma és hozama, lefúrási lehetőséggel különböző időablakokra (1 hétközött, 1 hónap, 3 hónap).
- Összehasonlító nézet: több eszköz egymás melletti teljesítményének és korrelációjának megjelenítése.

### **Speciális funkciók:**

A rendszer képes az árfolyamadatokból egyszerű trend és anomália elemzést végezni.

A mozgóátlagok segítségével megjeleníthetők a rövid és hosszú távú piaci trendek, míg az anomália detektálás a hirtelen árfolyam változásokat emeli ki.

Az eredmények vizuálisan is megjelennek a dashboardokon, így a felhasználó gyorsan felismerheti a rendkívüli piaci eseményeket és irányváltozásokat.

**Készülnek-e tesztek, ha igen, milyenek:** nem lesznek

## Technológiák

### Adatbázisok:

- MinIO: objektumtároló, ahol az API-kból lekért nyers JSON adatok kerülnek mentésre (RAW réteg).
- PostgreSQL: relációs adatbázis, amely a feldolgozott és aggregált adatokat tárolja a STAGE és DWH rétegen.
- Redis: memória alapú cache, amely a legfrissebb kriptovaluta- és részvényárfolyamokat tárolja a gyors lekérdezésekhez.

### ETL:

- Apache NiFi: az adatbetöltési és átalakítási folyamatokat vezéri. Automatizálja az API-lekéréseket, az adattisztítást, az adattárházba történő betöltést, valamint az ütemezett frissítéseket.

### Egyedi kódok, script-ek:

- Az API-hívások és adatlekérésök Python nyelven készülnek, egyszerű script formájában (pl. CoinPaprika, Yahoo Finance). Ezek a folyamatok integrálhatók a NiFi-be.

### Reporting motor:

- Grafana: a riportok és dashboardok megjelenítésére szolgál. A rendszer fő KPI-jei, trendjei és elemzései itt kerülnek vizualizálásra dinamikus szűrési és lefűrásí lehetőségekkel.

### Egyéb technológiák:

- Docker Compose: a fejlesztési környezet és komponensek (NiFi, PostgreSQL, Redis, Grafana) konténerizált futtatására, egyszerű telepítés és hordozhatóság érdekében.