

1. Sebesség összehasonlítás:

C/C++

- **Leggyorsabb** - natív kód, nincs virtuális gép
- De: **nehezebb fejleszteni**, több kód kell ugyanahhoz

Kotlin/Java

- **JVM optimalizáció** miatt nagyon gyors (sokszor közel C++ sebességhez)
- **Spring Boot** rengeteg optimalizációval
- **Könnyebb fejleszteni** - kevesebb kód

Valóság:

- **Nagy terhelésű rendszereknél** (Netflix, LinkedIn) Java/Kotlin tökéletesen megállja a helyét
- **Kis projektekhez** a különbség elhanyagolható
- **Fejlesztési sebesség** gyakran fontosabb mint a futtatási sebesség

2. JSON fájl generálás:

Most a szerver **futás közben** szolgálja ki a JSON-t. Ha **statikus JSON fájlt** akarsz:### Vagy hozzáadhatod a szerverhez egy endpoint-ot:

```
@GetMapping("/export")
fun exportToFile(): String {
    val greetings = index() // meglévő lista
    val mapper = jacksonObjectMapper()
    val json = mapper.writerWithDefaultPrettyPrinter()
        .writeValueAsString(greetings)

    File("exported_greetings.json").writeText(json)
    return "JSON fájl létrehozva!"
}
```

3. Miért jobb a "live" szerver?

Statikus JSON fájl:

- ❌ **Nem frissül** automatikusan
- ❌ **Nincs logika** (szűrés, rendezés)
- ❌ **Nem skálázható**

Live szerver:

- ✅ **Dinamikus** adatok (adatbázisból)
- ✅ **API funkciók** (keresés, módosítás)
- ✅ **Authentikáció**, jogosultságok
- ✅ **Több kliens** tud csatlakozni

4. Hibrid megoldás:

```
@GetMapping("/cached")
fun getCachedData(): List<Greeting> {
    // Cache-elt adat, ami 5 percenként frissül
    return cachedGreetings
}
```