

MySQL alapú nagyvállalati rendszer tapasztalatok

2017 HOUG szakmai nap

2017. 11. 08.

Ádám Károly

Világos Nándor



SZERENCSEJÁTÉK ZRT.

Tartalom

- Second Chance rendszer üzleti háttere
- Second chance rendszer projekt célkitűzései
- MySQL adatbázis réteg felépítése
- Alkalmazás réteg kialakítása
- Implementációs tapasztalatok



SECOND CHANCE RENDSZER ÜZLETI HÁTTERE



Second chance rendszer üzleti háttere

- Második esély a játékban
- Sorsjegyhez kapcsolódó akció
- Az akció nagyban segített a játkosokkal való kapcsolattartás megteremtésében
- A sikernek köszönhetően újabb két akció került hasonlóan megvalósításra (Okoslottó, Hírlevél)



SECOND CHANCE RENDSZER PROJEKT CÉLKITŰZÉSEI



Second chance projekt célkitűzései

- Fő cél a gyors megvalósítás volt
- Továbbá a gazdaságosság
- Fogadási rendszertől függetlenül került megvalósításra
- Az Oracle valósította meg az adatbázis és adatkezelő réteget, az ellenőrző logikát és adatvédelmi funkciókat



MYSQL ADATBÁZIS RÉTEG FELÉPÍTÉSE



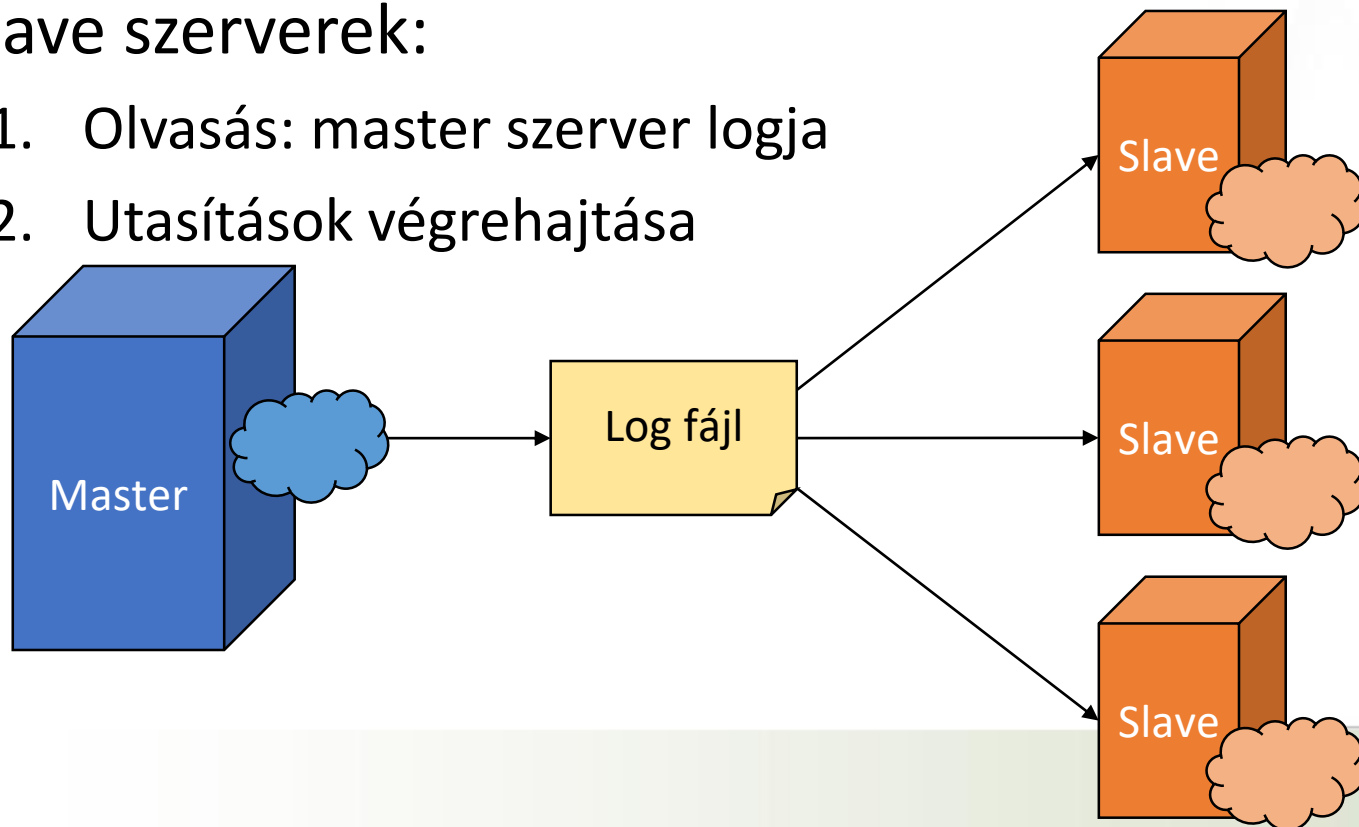
Alkalmazott verziók

- OS verzió:
 - Red Hat 6.x
- MySQL verzió:
 - 5.7.18 Community Edition



Magas rendelkezésre állású MySQL kiépítés

- Replikációs megoldás:
 - Bináris naplófájl alapján
 - Slave szerverek:
 1. Olvasás: master szerver logja
 2. Utasítások végrehajtása



MySQL monitorozás

- Az egyes komponensek külön fájlba
- Monitorozott komponensek:
 - my.conf fájl
 - MySQL kapcsolatok
 - Error napló
 - Replikációs lemaradás
 - Replikációs hibák
 - mysql szolgáltatás



MySQL mentés

- Két külön mentés
 - Teljes mentés
 - Naponta
 - Eszköz: mysqldump
 - Régi mentések, naplók törlése
 - Tranzakciós napló mentés
 - Óránként
 - Eszköz: mysqlbinlog
 - Utolsó mentett naplóbejegyzés után



ALKALMAZÁS RÉTEG KIALAKÍTÁSA



Fejlesztői eszközök

- Maven
 - Projektmenedzsment eszköz
 - Függőségek egységesítése
 - Deployment generálás leírása
 - Unit tesztek
 - Stb.
- Jenkins
 - Automatizált build + tesztek
 - Integrációs tesztek

Maven™

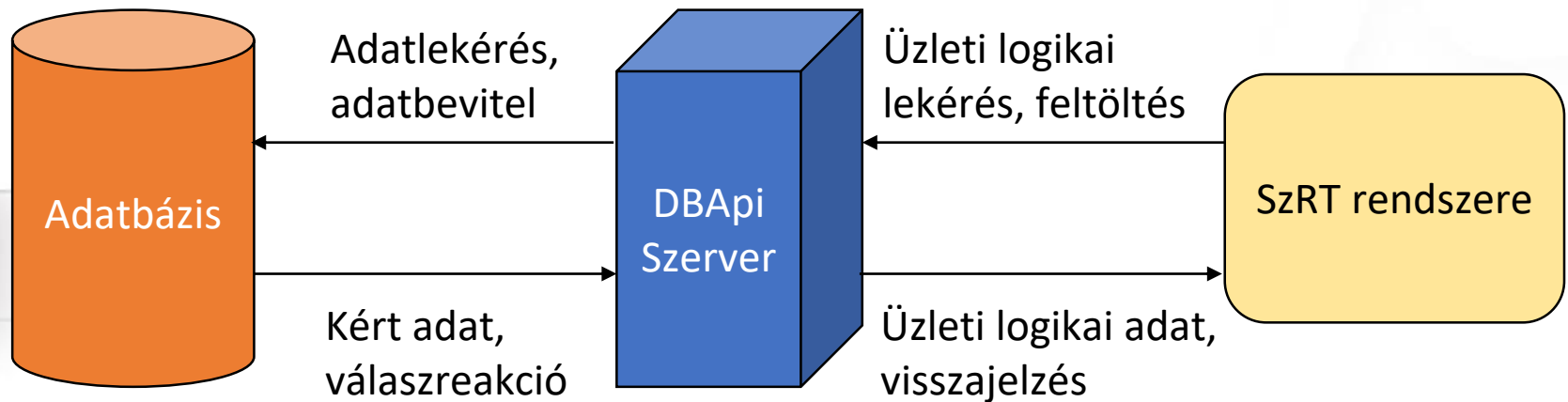


Fejlesztői eszközök

- Git
 - Verziókövetés
 - Fejlesztés összehangolása
 - Gitflow workflow: szabványos fejlesztési folyamat



Alkalmazás architektúra és a DBApi szerepe



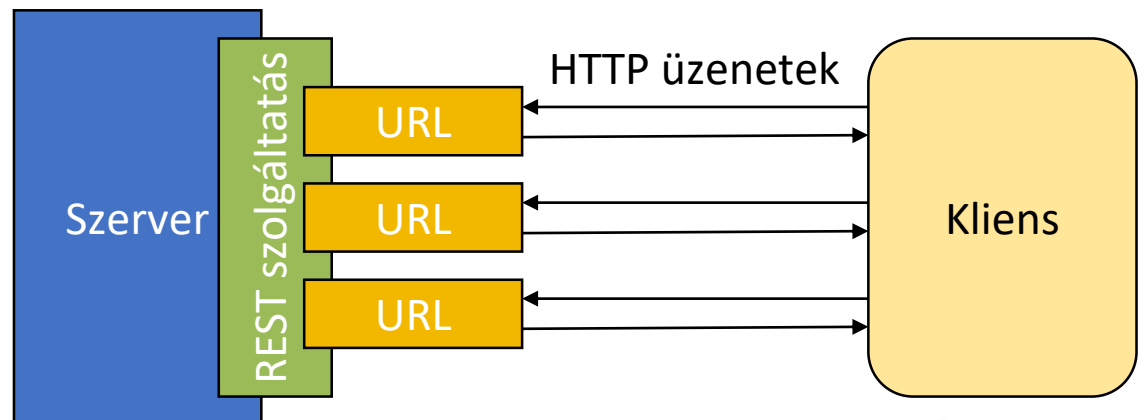
DBApi működése és alkalmazott technológiák

- RESTful (Representational state transfer) webes szolgáltatás
 - Kommunikáció: HTTP üzenetek segítségével
 - Műveletek: HTTP műveletek
 - GET, PUT, POST, DELETE, ...
 - Szükséges információk: URL-ben
 - Használható entitás
 - Pl. host/clients
 - Paraméterek
 - Stb.



DBApi működése és alkalmazott technológiák

- RESTful (Representational state transfer) webes szolgáltatás
 - Szabványos interfész
 - Lehetőség: automatizált dokumentáció
 - Hiba esetén: HTTP státusz kód + szöveges hibaüzenet



DBApi működése és alkalmazott technológiák

- Alkalmazott technológiák
 - Java API for RESTful Web Services (**JAX-RS**)
 - RESTful szolgáltatás biztosítása
 - **Enunciate**
 - Automatizált REST interfész dokumentáció
 - Java Persistence API (**JPA**)
 - Adatbázis kezelése
 - **JUnit**
 - Automatizált tesztelés



Performancia megfontolások és alkalmazott megoldások

- Adatátvitel: szöveges (JSON)
 - Kicsi overhead
- Named query VS. JPQL VS. Criteria API

	Named Query	JPQL	Criteria API
Lekérdezési lehetőség	Statikus	Dinamikus	Dinamikus
Fordítási idő	Előfordított	Futási időben fordított	Futási időben fordított (cache előfordulhat)
Bővíthetőség	Nem lehetséges	Megoldható	Könnyű
Olvashatóság	Könnyű	Könnyű	Nehéz
Lekérdezés felépítése	Szöveges	Leginkább szöveges	Objektum-orientált
Szintaktikai hiba detektálása	Futási időben	Futási időben	Fordítási időben



Biztonsági megfontolások és alkalmazott megoldások

- Paraméterek validálása
 - Védelem: nem rendeltetésszerű használat, végrehajtás csökkentése
- API kulcs
 - A HTTP header tartalmazza
 - A felhasználót azonosítja
 - Csak adott felhasználók használhatják a szolgáltatást



Biztonsági megfontolások és alkalmazott megoldások

- Infrastruktúrához kapcsolódó biztonsági megoldások
 - Pl. csak belső hálózatról érhető el
 - Nem a fejlesztés része



IMPLEMENTÁCIÓS TAPASZTALATOK



Implementációs tapasztalatok

- MySQL
 - Karakterkódolási probléma: hibás tárolás
 - Adatbázis: UTF-8, DBApi: nem definiált
 - MySQL driver bug
 - Első, timestamp típusú mező:
 - minden UPDATE-nél frissül



Implementációs tapasztalatok

- Fejlesztés
 - Java SE → Java EE Web profile
 - Maven: elengedhetetlen
 - Automatikus tesztelés: fontos
 - Automatikus dokumentáció generálás: hosszútávon előnyös



Kérdések



KÖSZÖNJÜK A FIGYELMET!

Szerző: Ádám Károly
e-mali: adam.karoly@szerencsejatek.hu

Szerző: Világos Nándor
e-mali: nandor.vilagos@oracle.com

