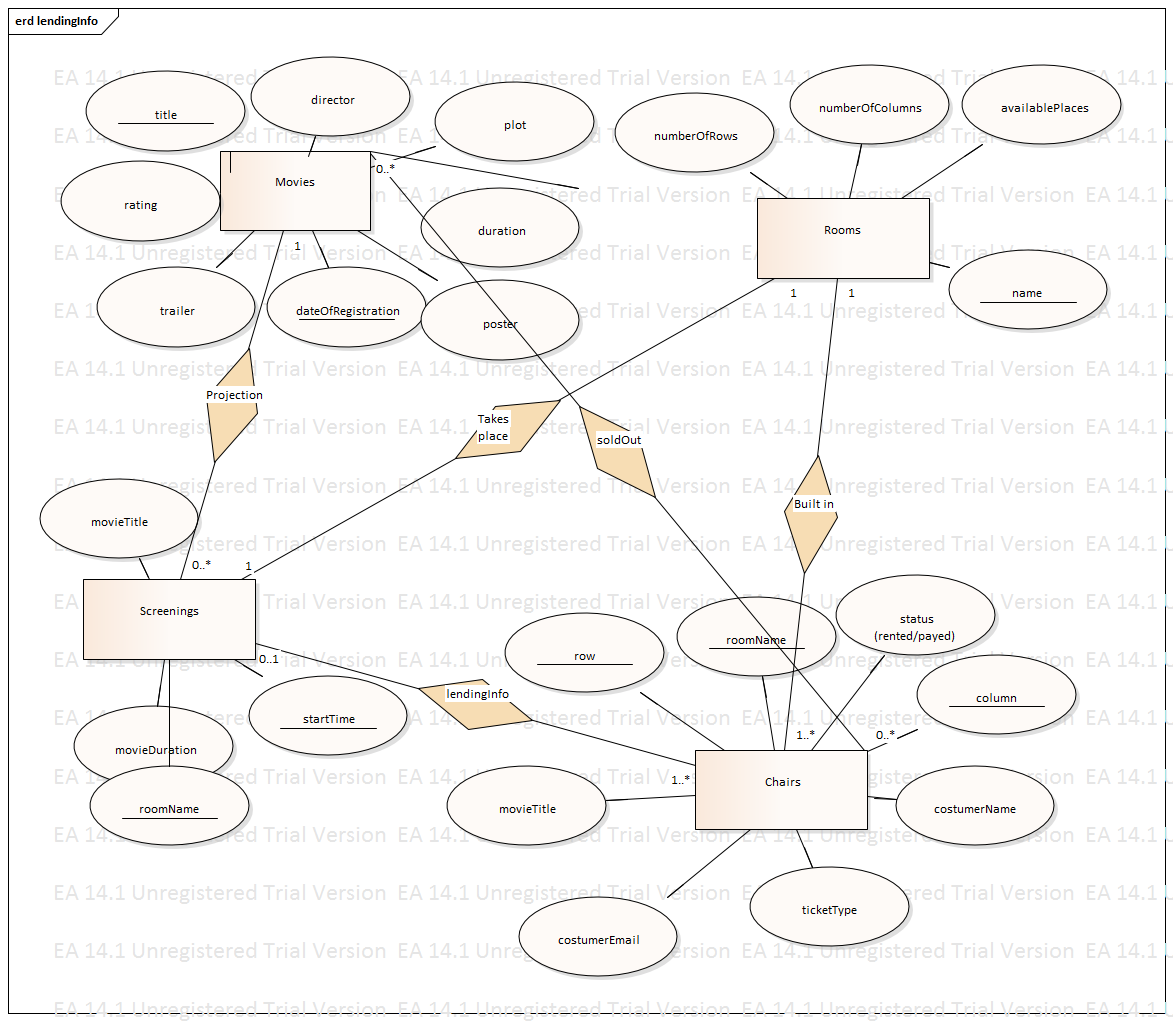
# Cinema – Dokumentáció

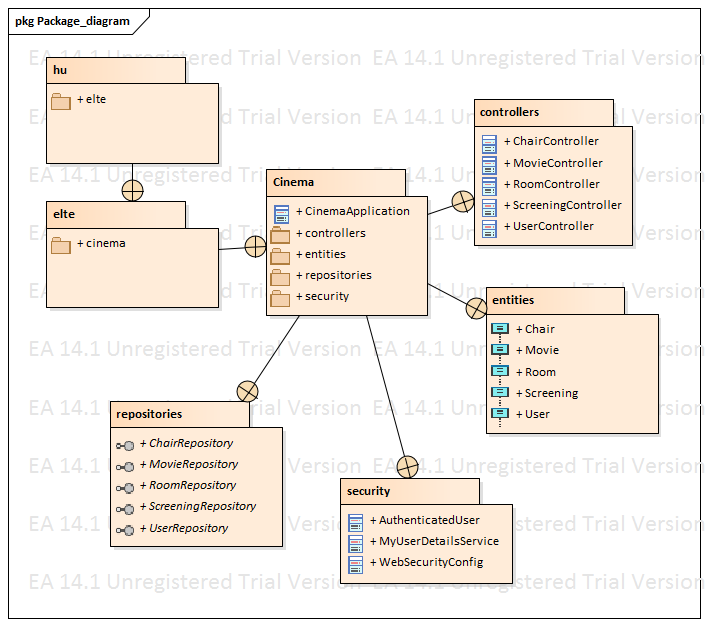
## A fejlesztői környezet és eszközök bemutatása:

A project a NetBeans nevű fejlesztői környezetben készült a Maven Project, mely egy olyan fejlesztői eszköz, amivel könnyen lehet felhasználni más modulokat/plug-in-eket, és leszedi az ehhez szükséges dependencyket(függőségeket), valamint a Spring Boot felhasználásával, mely egy olyan eszköz, amellyel könnyen és gyorsan lehet prototípusokat létrehozni. Továbbá felhasználtuk a h2 adatbázis motort is a projectben, mely Java SQL adatbázist vezényel, illetve a Lombok nevű plug-in-t, aminek köszönhetően nem kell gettereket, settereket írni, valamint összehasonlító(equals) metódusokat.

## Adatbázis terv:



## Alkalmazott könyvtárstruktúra bemutatása:



A forrás fájlok a cinema nevű főkönyvtárban találhatók, mely több alkönyvtárra ágazik szét:

* controllers
* security
* entities
* repositories

A fő könyvtárban található a CinemaApplication.java fájl mely a program belépési pontja (itt található a main metódus).

## Végpont-tervek és leírások:

A controllers package-ben lévő osztályok valósítják meg a végpontokat.

**ChairController:**

A „/chairs” végponton le lehet kérni az összes széket az adatbázisban, illetve új széket adhatunk hozzá. Ha a User admin, akkor az összeset, ha felhasználó, akkor pedig a hozzá tartozó székeket.

A „/chairs/{id}” végponton azonosító szerint lehet lekérni az adott széket , lehet módosítani, illetve törölni.

**MovieController:**

A „/movies” végponton le lehet kérni az összes filmet az adatbázisban, illetve új filmet lehet hozzáaadni.

A „/movies/{id}” végponton azonosító szerint lehet lekérni az adott filmet, lehet módosítani, illetve törölni.

A „/movies/{id}/screenings” végponton le lehet kérni az adott filmhez tartozó összes vetítést, illetve hozzá lehet adni új vetítést az adott filmhez.

A „/movies/{id}/chairs” végponton le lehet kérni az adott filmhez tartozó összes lefoglalt széket (tehát más szóval az összes foglalást az összes vetítésre), illetve hozzá lehet adni új széket (=foglalást) az adott filmhez, továbbá módosítani lehet a székeket (pl.: foglalt-ból eladott-at csinálni).

**RoomController:**

A „/rooms” végponton le lehet kérni az összes termet az adatbázisban, illetve hozzá lehet új termet adni.

A „/rooms/{id}” végponton az adott azonosítójú termet kérhetjük le, módosíthatjuk, illetve törölhetjük.

A „/rooms/{id}/chairs” végponton az adott azonosítójú teremnek kérhetjük le az összes székét (hogy foglalt-e/eladott és, hogy ki foglalta le stb.), illetve, ha lefoglalnak új széket, akkor itt is hozzá lehet adni.

A „/rooms/{id}/screenings” végponton megkaphatjuk az összes vetítést az adott azonosítójú teremhez, illetve újabb vetítést adhatunk hozzá.

**ScreeningController:**

A „/screenings” végponton le lehet kérni az összes vetítést az adatbázisban, illetve hozzá lehet új vetítést adni.

A „/screenings/{id}” végponton az adott azonosítójú vetítést kérhetjük le, módosíthatjuk, illetve törölhetjük.

A „/screenings/{id}/chairs” végponton az adott azonosítójú vetÍtéshez kérhetjük le a székeket (foglalási adatok), illetve adhatunk hozzá székeket (új foglalást).

A „/screenings/{id}/rooms” végponton az adott azonosítójú vetÍtéshez kérhetjük le a hozzá tartozó termet.

**UserController:**

A „/users/register” végponton új felhasználót tudunk hozzáadni.

A „/users/login” végponton pedig bejelentkezéssel azonosítani tudjuk a felhasználót.

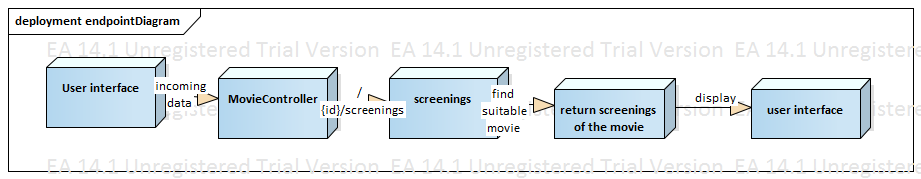
## 1db végpont működésének leírása:

Vegyük például a MovieController osztályt, és az abban található „movies/{id}/screenings” végpontot. Ehhez 2 metódus tartozik a screenings (a vetítések listázása) és az insertScreening metódus.

A „movies/{id]/screenings” végpontra érkezik a kérés, ezt a Controller elkapja, és a paraméterek alapján beazonosítja, hogy a screenings vagy az insertScreening metódust kell-e használnia.

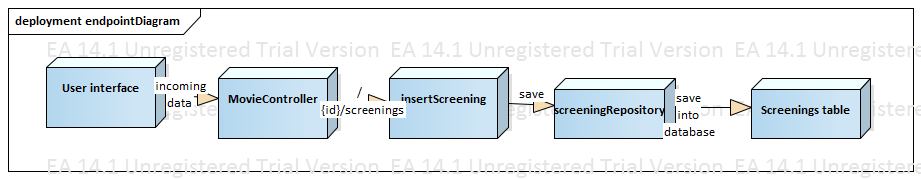
*screenings metódus:*

Megkeresi a filmet a film azonosítója (id) szerint, ha talált az adott azonosítóval filmet, akkor, a ResponseEntity ok metódusának segítségével visszaküldi a filmhez tartozó vetítéseket, ha nem, akkor pedig egy nem talált válasz entitást fog buildelni, amit visszaküld a ResponseEntity notFound metódusával.



*insertScreening metódus:*

Megkeresi a filmet a film azonosítója (id) szerint, ha talált az adott azonosítóval filmet, akkor a paraméterben kapott Screening objekutmnak beállítja az adott adott azonosítójú filmet, és a ResponseEntity ok metódusának segítségével elmenti a screeningRepository-ba (tehát a Screenings táblába) az adott vetítést, ha nincs az adott azonosítóval film, akkor pedig a ResponseEntity notFound metódusával egy nem talált válasz entitást fog buildelni.



## Fontosabb Specifikumok:

A chairs táblában az összes adatot csak az admin tudja megnézni, movies táblába adatot felvinni, törölni és módosítani is csak az admin tud, illetve a screenings és a rooms táblába felvinni, módosítani és törölni is csak az admin tud.