Neumann János Egyetem

GAMF Műszaki és Informatikai Kar

Java alkalmazások kurzus

Gyakorlati beadandó feladat dokumentációja

Készítette:

Balog Olivér

A20I86

Budai Zsolt

UB3RBV

2022.11.28.

**Tartalomjegyzék**

[**Feladatkiírás** 3](#_Toc120648932)

[**A program futtatásához szükséges gépi/nyelvi igények** 3](#_Toc120648933)

[**A program felépítése** 3](#_Toc120648934)

[A java mappa tartalma 3](#_Toc120648935)

[A vezérlők (Controller) 3](#_Toc120648936)

[Az ApiNotebookController osztály 3](#_Toc120648937)

[A RegisterController osztály 4](#_Toc120648938)

[A PagesController osztály 4](#_Toc120648939)

[Az NotebookNotFoundException osztály 4](#_Toc120648940)

[A modellek 5](#_Toc120648941)

[A Contact, a Gep, az Oprendszer, a Processzor, a Role, a User és a UserDTO osztályok: 5](#_Toc120648942)

[A resources/templates mappa tartalma 5](#_Toc120648943)

[Error mappa 5](#_Toc120648944)

[Fragments mappa 5](#_Toc120648945)

[**Továbbfejlesztési lehetőségek** 5](#_Toc120648946)

[**Képernyő képek** 6](#_Toc120648947)

# **Feladatkiírás**

Készítsen egy weboldalt egy fiktív cég, szervezet, alapítvány... részére, amely a következő követelményeknek tesz eleget:

# **A program futtatásához szükséges gépi/nyelvi igények**

A program futtatásához szükség van egy Java IDE-re, ami fordítja és futtatja az alkalmazást.

pl.: <https://www.jetbrains.com/idea/>

A src/main/resources/application.properties ki kell tölteni a megfelelő adatbázis értékekkel majd utána futtatni kell az alkalmazást.

Böngészőből a localhost:8080 link beírásával az oldal kezdőképernyője betöltődik, innen érhetjük el a program többi részét.

Internetes tárhelyen elhelyezett verzióhoz az alábbi linket kövesse:

TODO

A program elérhető GitHub-ról is a következő linkről: <https://github.com/oliverbalog/NotebookJavaBead>

# **A program felépítése**

A program felépítését megtekintve két fő mappával rendelkezik a src/main -ban a java és resources mappák.

## A java mappa tartalma

Ebben a mappában találhatóak a vezérlők (Controllers mappa), a repository-k (Repositories)a service-k (Services) és a modellek (Models).

A repository-k CrudInterface-k, amik az alap CRUD műveletekben segítenek.

A serviceben pedig UserDetailsService ami egyben egy RegisterUserService is.

## A vezérlők (Controller)

### Az ApiNotebookController osztály

A RESTful API kiszólásáért felel.

Az alábbi metódusok kerültek megvalósításra:

* store(): az megadott eszköz adatainak mentését végzi
* show(Integer $id): a megadott eszközt adja vissza
* update(Integer $id): egy eszköz módosítását végzi
* delete(Integer $id): a kiválasztott eszköz törlését végzi

### A RegisterController osztály

A felhasználó regisztrálásáért felel. A userRegistration függvény meghívja a userService registerNewUserAccount függvényét ami létrehozza a felhasználót egy user DTO (Data Transfer Object) alapján.

### A PagesController osztály

Az oldalak megjelenítésért felel. Metódusai:

* contactsListPage(Model model): kontaktok kilistázása
* contactPage(Model model): kapcsolati form betöltése
* contactSubmit(Model model): kapcsolati form validálása és mentése
* machinesPage(Model model): az eszközök kilistázása
* userPage(Model model): user jogosultság által látható oldal
* adminPage(Model model): admin jogosultság által látható oldal
* homePage(Model model): kezdőlap megjelenítése
* aboutPage(Model model): rólunk oldal megjelenítése
* loginPage(Model model): bejelentkezés oldal megjelenítése
* registerPage(Model model): regisztrációs oldal megjelenítése

### Az NotebookNotFoundException osztály

Egy eszköz nem találásának a kivétel kezelés osztálya.

## A modellek

A modellek az adatbázis táblákat reprezentálják egy-egy osztályal.

### A Contact, a Gep, az Oprendszer, a Processzor, a Role, a User és a UserDTO osztályok:

Get és set metódosukat tartalmaznak a privát adattagú mezőikre.

## A resources/templates mappa tartalma

Ebben a mappában kapott helyet a weboldal összes nézete kezdve a kezdőoldaltól, az adatbázist lekérő és megjelenítő oldalakon át, valamint a regisztrációs és belépési felület is.

A layout.html ami fontos mert ez fogja közre az összes aloldal tartalmát. Ez a fő html fájl amiben megtalálható a header és a footer is.

### Error mappa

A 404 és 500-as hibák megjelenítését tartalmazza.

### Fragments mappa

Ez a mappa tartalmazza az összes oldal tartalmát, amit a layout.html fog közre.

# **Továbbfejlesztési lehetőségek**

A legfontosabb tovább fejlesztési feladat az alap MVC fejlesztése, hogy még robusztusabb legyen és még több mindenben hasonlítson a nagytesójára, a Laravel-re. Pl.:

* Route-ok statikus deklarálása, put, patch, delete, options HTTP metódusokkal. RouteCollection háttérbe helyezése.
* Saját query builder, az absztract Modellen kívül.
* Hiba oldalnézetek.
* Logolás.
* Adatbázis migrációk.
* Adatbázis seederek.
* Factoryk.
* Több validációs lehetőség.
* Singleton app konténer.
* Automatikus dependency injection Reflection osztállyal.
* Middleware-ek.
* Konzolparancsok.
* .env (enviroment) fájl.

# **Képernyő képek**



















