Fejlesztői teszt

Imaginer

Készítette:

Horváth Ádám

Verzió: 1.0

2016.12.06

Tartalom

[1. Dokumentum története 3](#_Toc468804594)

[2. Bevezetés 4](#_Toc468804595)

[3. Alkalmazás 4](#_Toc468804596)

[3.1. Felhasznált technológiák 4](#_Toc468804597)

[3.2. Adattárolás 4](#_Toc468804598)

[3.2.1. Felhasználói adatok 4](#_Toc468804599)

[3.2.2. Galériák és képek 5](#_Toc468804600)

[3.3. Authentikáció 5](#_Toc468804601)

[3.4. Authorizáció 5](#_Toc468804602)

[3.5. Riport 6](#_Toc468804603)

[3.6. Telepítés 6](#_Toc468804604)

# 1. Dokumentum története

|  |  |  |  |
| --- | --- | --- | --- |
| **Változat** | **Dátum** | **Módosító** | **Változás rövid leírása** |
| 1.0 | 2016-12-06 | Horváth Ádám | Dokumentum létrehozása |

# 2. Bevezetés

Jelen dokumentum a fejlesztői teszt során elkészítendő alkalmazással kapcsolatban tartalmaz információkat.

# 3. Alkalmazás

## 3.1. Felhasznált technológiák

* Java 1.8
* Spring Framework 4.3
* JSF 2.2
* Servlet 3.1
* Primefaces 6.0
* Omnifaces 2.5

### 3.2. Adattárolás

### 3.2.1. Felhasználói adatok

A felhasználói adatok tárolása (felhasználónév, jelszó, email cím stb.) egy MySQL relációs adatbázisba történik (***user*** tábla). A felolvasásuk JPA (Hibernate implementáció), valamint Spring Data segítségével történik.

Az adatbázis kapcsolat kiépítése JNDI lookup segítségével történik.

***JNDI név***: jdbc/imaginer-ds

### 3.2.2. Galériák és képek

A tárolásuk Content Repository-ba történik (Apache Jackrabbit).

**Adattárolás struktúra:**

* root
  + users
    - felhasználónév1
      * galérianév1 (tulajdonságok: létrehozás, módosítás dátum, ikon bináris)
        + képnév1 (tulajdonságok: kép bináris, formátum)
        + képnév2 (tulajdonságok: kép bináris, formátum)
        + …
      * galérianév2
      * …
    - felhasználónév2
    - …

A kapcsolat kiépítése szintén JDNI lookup segítségével történik.

***JNDI név***: jackrabbitrepository

## 3.3. Authentikáció

Az authentikációt a Spring Security modul biztosítja. A megadott felhasználónév és a jelszó a ***user*** táblába kerül lementésre. A jelszó titkosítása Bcrypt segítségével történik. Az így kapott hash-elt jelszó kerül lementésre az adatbázisba.

### 3.4. Authorizáció

A felhasználókhoz jogok vannak kötve. Ezen jogok alapján érhetik el az alkalmazás egyes funkcióit. A jogok tárolása a ***user\_group***, a felhasználókhoz való hozzárendelése a ***user\_group\_to\_user*** táblába történik.

**Két alapvető jogot kezel a rendszer:**

* ***Normál felhasználó (USER)***: tud galériát létrehozni, képeket feltölteni és megtekinteni
* ***Riport készítő felhasználó (REPORT):*** létező felhasználó, csak a riportok oldalt látja

A rendszer lehetővé teszi, hogy egy felhasználó akár mindkét jogkörrel rendelkezzen.

## 3.5. Riport

A riporthoz tárolva vannak a felhasználók bejelentkezési dátumai. Ezek a rekordok a ***user\_login\_history*** táblába kerülnek. Riport generálást a REPORT joggal rendelkező felhasználók indíthatnak. A bejegyzések egy CSV állományba kerülnek lementésre.

## 3.6. Telepítés

Az alkalmazás építéséhez Maven 3 és 1.8-as Java szükséges.

***mvn clean install***

dist/verziószám mappába kerül a kész artifact.

A fejlesztéshez Weblogic 12C alkalmazásszervet használtam. Az ***alkalmazás szerver konfigurációja*** a következő:

* Fel kell venni egy ***MySQL-es datasource***-t a fentebb említett JNDI névvel ***(jdbc/imaginer-ds)***. Az alkalmazás ezt a JNDI nevet az alkalmazásszerver domain home-jában található app-config/database-config.properties fájlból olvassa fel. A fájl megtalálható a projekt/ domain-properties mappában. A hbm2ddl modul az alkalmazás indítása során létrehozza a megfelelő táblákat. Szükséges, hogy a projekt core moduljában (src/main/resources) található import.sql-t lefuttassuk a user\_group-ok felvételéhez.
* Létre kell hozni egy ***Jackrabbit repository-t***. Ehhez a Jackrabbit webalkalmazást használtam (jackrabbit-webapp-2.4.1.war). A repository konfigurációját a bootstrap.properties állomány tartalmazza. Ez a Jackrabbit repository gyökerében található. Az webalkalmazás által létrehozott bootstrap.properties-ben található értékeket felül kell írni a projekt/ domain-properties mappában található azonos nevű fájl tartalmával. Az alkalmazás a kapcsolódáshoz szükséges információkat a zalkalmazásszerver domain home-jában található app-config/ imaginer\_jcr.properties és jackrabbit.properties fájlból olvassák fel. Mindkét fájl megtalálható a projekt/ domain-properties mappában.

***Megjegyzés***: Szükséges, hogy a jcr-2.0.jar fent legyen a szerver classpath-án.

* A validációhoz használt reguláris kifejezéseket a validation.properties fájl tartalmazza. Szinten a projekt/ domain-properties mappában található. Ezt a fájl is be kell másolni az alkalmazásszerver domain home-jában található app-config mappába.
* log4j2.xml fájt az alkalmazásszerver domain home-jába kell másolni.