Gyakorlati feladatsor – Autentikáció és jogosultságkezelés

Ez a dokumentum két gyakorlati blokkot tartalmaz. A cél, hogy a hallgatók megértsék a session alapú (MVC) és a token/JWT alapú (REST) autentikáció közti különbségeket, és ezeket működés közben is megfigyeljék.

Github link: https://github.com/zbalogh/oe-internetes-alkalmazasok
Innen letölthető a teljes repository benne a szükséges projektek.

1. Gyakorlat – Spring Boot MVC: Session és Cookie

Projekt: **spring-boot-mvc-lab** a 10-es számú könyvtárban.

A **README.md** szerint leírtak alapján indítsd el a projektet, majd a böngészőben nyisd meg az alkalmazást. A gyakorlat során a Spring Security beépített form alapú login megoldását fogjuk használni.

Feladat 1 – Alkalmazás indítása

1. Indítsd el az alkalmazást.
2. Nyisd meg a böngészőben: http://localhost:8080
3. Figyeld meg, mi történik, ha nem vagy bejelentkezve.

Feladat 2 – Bejelentkezés és session létrejötte

1. Jelentkezz be a megadott felhasználóval.
2. Nyisd meg a böngésző DevTools eszközét.
3. Keresd meg a válaszban a Set-Cookie fejlécet.
4. Ellenőrizd, hogy a cookie minden kérésnél elküldésre kerül-e.

Feladat 3 – Védett oldal elérése

1. Jelentkezz ki az alkalmazásból.
2. Írd be közvetlenül a védett URL-t a böngésző címsorába.
3. Figyeld meg az alkalmazás viselkedését.

Feladat 4 – CRUD műveletek megfigyelése

1. Bejelentkezés után végezz el legalább két műveletet (create, update vagy delete).
2. Figyeld meg, hogy a session cookie minden kérésnél jelen van-e.

2. Gyakorlat – Spring Boot REST: JWT autentikáció

Projekt: **spring-boot-rest-lab** a 10-es számú könyvtárban.

Ebben a gyakorlatban JWT alapú stateless autentikációt fogunk tesztelni REST API-n keresztül.

Feladat 1 – Swagger elérése

1. A **README.md** szerint leírtak alapján indítsd el az alkalmazást
2. Olvasd el **JWT_TESTING_GUIDE.md** és a **SWAGGER_JWT_GUIDE.md** fájlokat.
Ezek részletes információt és segítséget adnak a feladatsor elvégzéséhez.
2. Nyisd meg a Swagger UI-t a böngészőben.
3. Nézd meg a publikus és védett végpontokat.

Feladat 2 – Védett végpont hívása token nélkül

1. Hívd meg a védett GET végpontot token nélkül.
2. Jegyezd fel a HTTP státuszkódot.

Feladat 3 – Token beszerzése (login)

1. Hívd meg az AuthController login végpontját helyes adatokkal.
2. Másold ki a kapott JWT tokent.
3. Próbáld ki hibás adatokkal is.

Feladat 4 – Védett végpont hívása tokennel

1. Add meg az Authorization: Bearer <token> fejlécet.
2. Hívd meg újra a védett végpontot: UserController
3. Figyeld meg a különbséget.

Feladat 5 – Swagger Authorize használata

1. Swagger-ben használd az Authorize gombot.
2. Illeszd be a JWT tokent.
3. Tesztelj egy védett végpontot.

A feladatokat elvégezhetők a **Swagger UI**-ban, miközben a böngésző **DevTools**-ban megfigyelhetők a HTTP kérések és válaszok.

A Swagger UI mellett használható a **Postman** vagy ehhez hasonló **REST kliens** is!

Megjegyzések:

1. Gyakorlat:

- A bejelentkezés után a szerver session-t hoz létre.
- A session azonosító cookie formájában kerül a klienshez.
- A böngésző minden kérésnél automatikusan elküldi a cookie-t.

2. Gyakorlat:

- Token nélkül a védett REST végpontok elutasítják a kérést.
- Sikeres login után JWT tokent kapunk.
- A tokent minden kérésnél explicit módon kell elküldeni.
- A REST API stateless módon működik.