Систем за автентикација во Java Spring Boot

Проектот е изработен во Java Spring каде се обидов да креирам систем за автентикација без користење на веќе постоечката зависност од Spring наречена Spring Security.

Проектот е направен од повеќе слоја:

- Контролери
- Сервиси
- Репозитори
- Модели
- Интерцептори

Постојат 2 контролери од кои едниот е AuthController кој е всушност главниот контролер во оваа апликација тој ги врши сите потребни функции при автентикација на корисник. Вториот е WelcomeController односно тој има само една функција да ни прикажи страница за успешно најавен корисник и копче за Log out.

Да навлеземе подлабоко во функциите на AuthController:

- Прва и најглавна функција му е login() односно проверува дали корисникот има креирано профил, профилот е активиран и внесува точни информации при логирање. Ако корисникот успешно се логира тогаш серверот му испраќа cookie со токен кој мора да го испраќа при секое наредно барање за да си го потврди својот индентитет, каде серверот тој токен ќе го споредува со токенот зачуван во корисничката сесија.
- Имаме функција за регистрација на нови корисници. Таа прво проверува дали email адресата и одбраниот username веќе постојат во базата, ако не постојат, прво се проверува дали пасвордот одбран од корисникот е најмалку 8 знаци долг, и содржи барем една бројка. Ако се е тоа добро тогаш двата пасворди се хешираат на клиентска страна и се испраќаат до серверот, каде тој повторно проверува дали пасвордите се еднакви, ако е се исполнето според барањата, тогаш тој додава “сол” на веќе хешираниот пасворд и повторно го хешира заедно со новите знаци. Откако е се тоа исполнато, се испраќа email до корисничката адреса каде треба корисникот да го потврди креирањето на новиот профил, откако ќе го направи тоа тогаш може да се логира на страната, инаку се појавува порака дека корисничкиот профил не е активиран.
- Имаме функција која овозможува менување на пасвордот на корисникот. За да се мени пасвордот прво мора да се испрати токен за потврдување до email адресата на корисникот. Откако линкот на адресата е стиснат тогаш корисникот е редиректиран на страница за промена на својата лозинка каде важат истите стандарди како и при регистрација за одбирање на пасворд. Ако се е успешно тогаш повторно се додава сол, и се хешира пасвордот, и корисникот можи сега да пристапува со тој пасворд на неговиот профил.
- И за крај имаме logout функција односно таа само го трга корисникот од неговата сесија, каде серверот заборава дека тој е логиран, и за да пристапи повторно до некој ресурс тој мора повторно да се логира и да добие нов токен.

Сега да навлеземе подлабоко во сервисите:

- Го имаме AuthService односно тој е главниот сервис во оваа апликација. Тој има повеќе функции како loginUser() односно проверива дали се е исполнато за да се логира корисникот. CreateUser ја превзема работата од контролерот за проверка на точност на податоци и креирање на нов корисник, и тој е всушност кој го запишува корисникот во база. Имаме неколку функции за проверки како за Email Address и Username, кои се користени од самите сервиси и контролери. И ги имаме најважните функции односно функциите за проверка на токени при регистрација и менување на лозинка, ако тие не се исти како токените во базата тогаш корисникот не може да се логира/смени лозинка (односно тие осигураат дека друг корисник не може да потврди за вистинскиот корисник и да му наштети во самиот профил)
- Потоа го имаме CookieService односно тој сервис има само една улога и таа е креирање или одстранување на cookies за корисникот
- SaltService тој се користи за креирање на "сол" и хеширање на пасвордот на корисникот. Исто така тој е сервисот кој ја има функцијата за проверка на пасвордот внесен при логирање дали се совпаѓа со тој кој е запишан за корисникот во база заедно со солта.
- И за крај го имаме EmailService односно тој сервис само ги испраќа email пораките до корисниците со токените за регистрација или менување на пасворд.

Репозиторија која е вклучена во овој проект е само една тоа е само за User:

- Таа се користи за конектирање на серверот со базата, и за пребарување на корисници по нивниот username или email, и за запишивање на нови корисници во базата.

Апликацијата има само еден модел и неговите атрибути се:

- Id кое е клуч во базата
- Username I email кои можат да се користат при логирање на корисникот во апликацијата.
- Password односно тоа е нивната тајна лозинка која ја користат при логирање.
- 2 вида на токени, односно тие се користат при креирање на корисник или менување на лозинка, и двата стануваат null ако операцијата за која се потребни е успешно извршена за да се задржува помал мемориски простор во базата.
- Role односно овој атрибут ќе се користи подоцна кога ќе има потреба и за автентикација на корисниците според нивните улоги.

И за крај имаме Interceptor:

- Единствена улога на оваа класа е да го спречи корисникот до пристап на страната каде МОРА да биде логиран, ако не е логиран ќе го редиректира на /login