

אז ישנו role ויש לנו permission בעצם הrole זה השם של ההגדרה והpermission זה ההרשאות שיש לו.

נגיד יהיה לי admin שיוכל לקרוא ולכתוב להרבה דברים ויהיה לי student שיוכל לקרוא וכתוב להגדרות שלו אבל לקרוא רק את הקורסים לשו בלי יכולות לכתוב.

אז איך בונים רולים באפליקציה שלנו?

יצרנו enum חדש בשם ApplicationUserPermission שבו יגדרו כל ההרשאות

```
package com.example.demo.security;

public enum ApplicationUserPermission {
    STUDENT_READ("student:read"),
    STUDENT_WRITE("student:write"),
    COURSE_READ("course:read"),
    COURSE_WRITE("course:write");

    private final String permission;

    ApplicationUserPermission(String permission) {
        this.permission = permission;
    }

    public String getPermission() {
        return permission;
    }
}
```

אחרי זה הוספנו את התלות הזאת

```
<dependency>
  <groupId>com.google.guava</groupId>
  <artifactId>guava</artifactId>
  <version>28.1-jre</version>
</dependency>
```

```

ApplicationSecurityConfig.java × applicationUserRole.java × ApplicationUserPermission.java ×
1 package com.example.demo.security;
2
3 import com.google.common.collect.Sets;
4
5 import java.util.Set;
6 import static com.example.demo.security.ApplicationUserPermission.*;
7 public enum applicationUserRole {
8     STUDENT(Sets.newHashSet()),
9     ADMIN(Sets.newHashSet(COURSE_READ, COURSE_WRITE, STUDENT_READ, STUDENT_WRITE));
10
11     private final Set<ApplicationUserPermission> permissions;
12
13     applicationUserRole(Set<ApplicationUserPermission> permissions) {
14         this.permissions = permissions;
15     }
16
17     public Set<ApplicationUserPermission> getPermissions() {
18         return permissions;
19     }

```

אז אלו יהיו המשתמשים באפליקציה שלנו

משתמש admin עם role של ADMIN

משתמש student עם role של STUDENT

הנה rolen ישמרו בצורה של ROLE_ADMIN -> כלומר יתווסף להם _ROLE בהתחלה

```

@Override
@Bean
protected UserDetailsService userDetailsService() {
    UserDetails rootUser = User.builder()
        .username("admin")
        .password(passwordEncoder.encode(rawPassword: "admin"))
        .roles(ADMIN.name())
        .build();

    UserDetails studentUser = User.builder()
        .username("student")
        .password(passwordEncoder.encode(rawPassword: "1324"))
        .roles(STUDENT.name())
        .build();

    return new InMemoryUserDetailsManager(rootUser, studentUser);
}

```

עשינו את הimport בצורה סטטית (אפשר לעמוד מאיפה שמביאים אותו לעשות

```
    .password(passwordEncoder.encode(password))
    .roles(ApplicationUserRole.ADMIN.name())
    .build();

// Details studentUser = User.builder()
//     .username("student")
//     .password(passwordEncoder.encode(password))
//     .roles(ApplicationUserRole.STUDENT.name())
//     .build();
```

Split into declaration and assignment

>

Expand method call chain into call sequence

>

Add static import for 'com.example.demo.security.ApplicationUserRole.STUDENT'

>

Wrap vararg arguments with explicit array creation

>

Press Ctrl+Shift+I to open preview

```
import static com.example.demo.security.ApplicationUserRole.*;
```

ואז למעלה לעשות *