**Authorization**

Authorization is to check if a request has the authority to perform what it needs to do.  
  
Now here we will take a Spring boot application with a couple of APIs and we will learn how to enable or disable access the APIs depending on who the logged in user is.   
  
We will learn the way to configure Authorization!!

Let’s take three APIs:

1. “/” 🡪 All(unauthenticated) – to be accessed by all.
2. “/user” 🡪 only authenticated user/admins
3. “/admin” 🡪 only authenticated admins, authenticated users shall not be allowed to access the APIs.  
     
     
   PROGRAM:

Create a controller:

@RestController

**public** **class** SpringSecurityController {

@GetMapping("/")

**public** String home() {

**return** "<h1>Welcome to your home page </h1>";

}

@GetMapping("/user")

**public** String user() {

**return** "<h1>Welcome User</h1>";

}

@GetMapping("/admin")

**public** String admin() {

**return** "<h1>Welcome Admin</h1>";

}

}

The way to achieve Authorization is to create a HttpSecurity object. Override configure(HttpSecurity http)

@Override

**protected** **void** configure(HttpSecurity http) **throws** Exception {

http.authorizeRequests()

.antMatchers("/admin")

.hasRole("ADMIN")

.antMatchers("/user")

.hasAnyRole("USER", "ADMIN")

.antMatchers("/")

.permitAll()

.and()

.formLogin();

}