

Day → 2



- @EnableWebSecurity annotation tells that spring security don't go with default security configuration go with that configuration that I mentioned here.
- जैसे ही हम class पर @EnableWebSecurity annotation और SecurityFilterChain का Bean बना देते हैं चाहे हमने अभी उस Bean में कुछ लिखा तक भी नहीं फिर भी Default Configuration वाला login Page हट जायेगा अब नहीं आयेगा वो / क्योंकि CustomConfiguration enable हो गया।
- अब हम एक Project बनायेंगे फिर उससे working समझेंगे
- Create a new Project ⇒ DaysSpringSecurity-1

Dependencies ⇒

- spring web
- spring dev tools
- spring security

→ Create an entity class ⇒ Student

```
public class Student
```

```
{
```

```
    private int id;
```

```
    private String name;
```

```
    // Create getter & setter
```

```
    // Create Constructor
```

```
    // Create toString method
```

```
}
```

→

- Create a service class \Rightarrow StudentService

@Service

public class StudentService

{

public List<Student> students = new ArrayList<>()

List.of (

new Student(101, "Piyush"),

new Student(102, "Lokesh"),

new Student(103, "Jitu")

)

);

public List<Student> getAll()

{

return students;

}

public Student saveStudent(Student std)

{

students.add(std);

return std;

}

- go to application.properties \Rightarrow

spring.security.user.name = root

spring.security.user.password = root

\rightarrow next Page

→ Create a Controller class :- MyController

@RestController

public class MyController

{

 @Autowired

 private StudentService stdserv;

 @GetMapping("/getAll")

 public List<Student> getAll()

 {

 return stdserv.getAll();

 }

 @PostMapping("/save")

 public Student save(@RequestBody Student std)

 {

 return stdserv.saveStudent(std);

 }

}

• Run the Program → Open Postman

Get → localhost:8080/getAll

basicAuth → username → root

Password → root

then,

O/P → [id: 101
 name: Rajesh

 id: 102

 name: Lokesh

 id: 103

 name: Jithu

]



→ means हमने username and Password दे कर Data get करा तो हमें Data मिल गया।

“ पर अब हम Data को save करने के लिए POST mapping चलायेंगे तो 401 Unauthorized आएगा क्योंकि हम CSRF token नहीं भेज रहे हैं Data के साथ। ”

→ “ अब हम CSRF token generate करेंगे ”

और उस CSRF token को PostMapping के साथ भेजेंगे तो हमारा Data successfully save हो जायेगा।

Go to Controller :-

same code

```
@GetMapping("/csrf-Token")
public CsrfToken getToken (HttpServletRequest http)
{
    return (CsrfToken) http.getAttribute("-csrf");
}
```

→ Open Postman :- अब हम get करेंगे Token

Get → localhost : 8080 / csrf-Token

“ जिससे ही हम send पर click करेंगे तो एक Header और CSRF-Token o/p में आएगा उसे हम copy कर लेंगे। ”

→ अब हम PostMapping करेंगे।

Headers

Key : _____ : Value : _____
 ↓ ↓
 Paste Header Paste CSRF Token

send
Click

Data saved
successfully.

→ तो अब हम ये चाहते हैं की जो CSRF Token का concept है By default ये enable है तो हम इसे disable कर देंगे तो वो ही Previous Project वाले का concept use होगा।

→ Create a new class → SecurityConfig

@Configuration

@EnableWebSecurity

public class SecurityConfig

{

@Bean

public SecurityFilterChain func (HttpSecurity http) throws Exception

{

return http.csrf (csrf → csrf.disable())
 .build();

}

}

→ अब Data किता csrf token किता बिना जो Put हो जायगा वोकि हमने CSRF को disable कर दिया।

(ये CSRF Parameter में एक Interface होता है Customizer Interface में की एक functional Interface है इसलिए हम Lambda function का use कर पा रहे हैं csrf के Parameter में)

→ Same Code

return http.csrf (csrf → csrf.disable())

.authorizeHttpRequests (req → req.anyRequest().authenticated())
 .build();

→ ये लिखने से सारी request protect हो जायेगी की पहले आप authenticate करो।

localhost:9080/ getAPI

→ o/p → Access to localhost was denied

→ तो अब chrome पर login page आ ही नहीं रहा है तो username और password जल के authenticate कैसे करे की ताकि हम Page को access कर सके।

"Postman से भी नहीं होगा"

→ तो अब हम spring security को बोलेंगे थोड़ी सी तो हमारा customise default LoginPage लाना means हमारा अभी default configuration है।

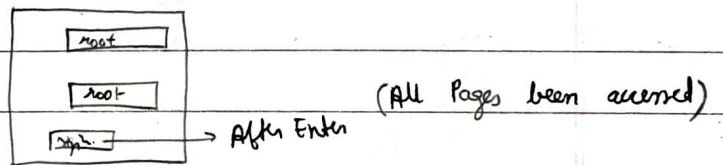
same code

```
return http.csrf (csrf -> csrf.disable())
```

- authorizeHttpRequests (req -> req.anyRequest().authenticated())
 - formLogin (Customizer.withDefaults())
 - build();
- ~~class~~ interface → static method

}

Open chrome :- localhost:8080/



→ पर हम same अब Postman से करेंगे तो Response में पूरा HTML Page आ जाएगा Default Login वाला student का data नहीं आ रहा।

```
return http.csrf (csrf -> csrf.disable())
```

- authorizeHttpRequests (req -> req.anyRequest().authenticated())
 - formLogin (Customizer.withDefaults()) → ये chrome के लिए
 - httpBasic (Customizer.withDefaults())
 - build();
- ये Postman के लिए।

}

"Most Important Concept"

→ अभी तक यह हो रहा था कि हम Application Properties file में एक user declare कर रहे थे व password के साथ और जब हम Page access कर रहे थे तो spring security के login page पर हम वही दे देते username & password तो हमें access मिल जाता।

“ पर यदि अब मैं चाहता हूँ कि ये ये user login हो सके ”

① username → Pijush

Password → root

② username → Lokesh

Password → root123

means मैं चाहता हूँ की एक से ज्यादा users access कर सकें।

“ अभी हम ये same Database से करेंगे पर अभी हम Database पर नहीं जा रहे हैं ”

→ So, go to our Configuration class

Same Code

@Bean

public SecurityFilterChain func(HttpSecurity http) throws Exception {

return http.csrf(csrf -> csrf.disable());

- authorizeHttpRequests (req -> req.anyRequest().authenticated());
- formLogin (Customizer.withDefaults());
- build();

}

@Bean

public UserDetailsService func1() {
UserDetails userDetails = User.withDefaultPasswordEncoder()

Interface

ये User का reference इमेजिस्ट कर पा रहा है क्योंकि ये User class हम interface को implement करती है।

• username ("Pijush")

• password ("123")

• roles ("USER")

• build();

Interface (which contain only one method)
↳ UserDetails loadByUsername (String s)

Class

```
UserDetails userDto2 = User.withDefaultPasswordEncoder()
    .username("lucky")
    .password("root")
    .roles("USER")
    .build();
```

```
return new InMemoryUserDetailsManager(userDtls1, userDtls2);
}
```

इस UserDetailsService का object तो नहीं बना सकते तो
यह class उसकी implementing class है।

→ Complete working of Above Code \Rightarrow ^{on} Architecture
"First you see Internal Diagram of Spring security"

हमने above code में func1. बनाया है जिसमें UserDetailsService का Implementation है। जब भी हम UserDetailsService का implementation करते हैं तो Spring security internally इस method को call करता है authentication process के दौरान।

Spring security ने बीना की तुम मुझे UserDetailsService का Object दे ले तो Internally बाकी सब से कर लूँगा।
तो UserDetailsService तो एक Interface है तो हमने उसकी Implementing class InMemoryUserDetailsManager का object बनाके return कर दिया। पर वही class के पास एक Parameterized constructor है जो Parameter में UserDetails लेता है (यह एक Interface है)

Parameter में UserDetails होता है।
 ✓ User इसकी Implementing class है तो हम उसमें username, Password, role में सब Define कर देंगे और UserDetails का reference variable इसको hold कर सकता है क्योंकि "super class का reference variable sub class के object को hold कर सकता है"।
 इस UserDetails object को InMemoryUserDetailsManager के Parameter में Pass कर देंगे।

→ Suppose हमने sign Page पर username & Password डाला like Pujush and root तो Spring security के अंदर internally वो UserDetailsService interface की loadByUsername() method को call करेगा और उसके Parameter में हमारा Pujush username डाल देगा। फिर loadByUsername() method से internally check करेगी क्या इस नाम का user exist कर रहा है यदि कर रहा है तो UserDetailsService का fully Object return करेगी (means implementing class का) अब ~~ह~~ ये check होगी की जो इस object के अंदर Password है वो match हो रहा है क्या जो user ने Password डाला है उसे यदि match हो गया तो हमें access मिल जायेगा।