

Tutorial 8

Mengimplementasikan Web Security Pada Spring Boot Framework

Review

Pada tutorial 5 ini mengimplementasikan web security. Dimana dalam mengembangkan sistem perlu dibuat aturan otentikasi dan otorisasi. Otentikasi diperlukan untuk melakukan verifikasi bahwa pengguna mempunyai akses untuk masuk ke dalam sistem. Otorisasi merupakan pengaturan hak pengguna yang telah memiliki otentikasi. Pengimplementasiannya pun sesuai dengan tingkatan pengguna. Pengguna dengan tingkat Admin memiliki hak otorisasi yang berbeda dengan staf biasa. Otorisasi menentukan fitur apa saja yang dapat diakses oleh seorang pengguna yang telah diotentikasi.

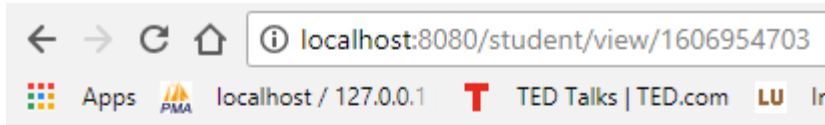
Latihan

1. Buatlah hal yang sama dengan menambahkan tombol Sign Out dan cetak nama user untuk halaman student/view/{npm}.

Pada view.html tambahkan code seperti di bawah ini yaitu untuk membuat button sign out.

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <title>View Student by NPM</title>
  </head>
  <body>
    <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
    <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
    <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
  </body>
  <h2 th:text="'Login as ' + ${#httpServletRequest.remoteUser}" >Login as</h2>
  <form th:action="@{/logout}" method="post">
    <input type="submit" value="Sign Out"/>
  </form>
</html>
```

Ini adalah tampilan outputnya.



NPM = 1606954703

Name = Aisyah Sadiyah

GPA = 3.0

Login as admin

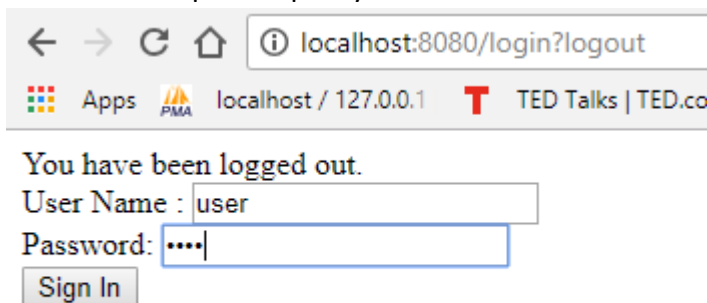
Sign Out

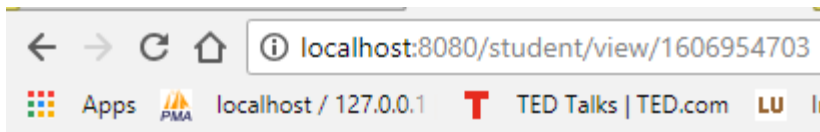
2. Buatlah user baru dengan username "user", password "user", dan role "USER". Coba apakah bisa login dengan akun tersebut.

Pada WebSecurityConfig.java tambahkan code seperti di bawah ini. Untuk membuat user baru dengan username, password, dan role user.

```
@Autowired
public void configureGlobal (AuthenticationManagerBuilder auth) throws Exception
{
    auth.inMemoryAuthentication()
        .withUser("admin").password("admin")
        .roles("ADMIN");
    auth.inMemoryAuthentication()
        .withUser("user").password("user")
        .roles("USER");
}
```

Ini adalah tampilan outputnya.





NPM = 1606954703

Name = Aisyah Sadiyah

GPA = 3.0

Login as user

Sign Out

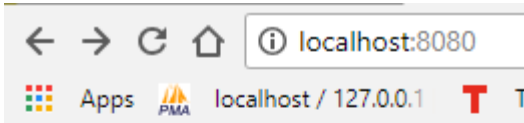
3. Tampilkan menu view student by npm beserta input field dan button untuk memasukan nomor npm student yang ingin dicari pada halaman index user dengan role USER. Pada index.html tambahkan code di bawah ini untuk menampilkan view student by npm dengan form input NPM dan button lihat.

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org" >
  <head>
    <title>Index</title>
  </head>
  <body>
    <h2>Welcome</h2>
    <a href="/login">Login</a>
    <a th:href="${#httpServletRequest.isUserInRole('ADMIN')}" href="/student/viewall">View All Student</a><br/>
    <div th:if="${#httpServletRequest.isUserInRole('ADMIN')} || #httpServletRequest.isUserInRole('USER')}">
      <form th:action="@{/student/view/}" method="post">
        <h3>view student by npm</h3>
        <div class="form-row">
          <div class="form-group col-md-4">
            <label for="npm">NPM</label>
          </div>
          <div class="form-group col-md-4">
            <input type="text" name="npm" class="form-control"/>
          </div>
          <div class="col-md-4">
            <button type="submit" class="btn btn-primary">Lihat</button>
          </div>
        </div>
      </form>
    </div>
  </body>
</html>
```

Pada WebSecurityConfig.java ditambahkan rolenya seperti di bawah ini.

```
@Configuration
@EnableWebSecurity
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
    @Override
    protected void configure(HttpSecurity http) throws Exception
    {
        http
            .authorizeRequests()
            .antMatchers("/").permitAll()
            .antMatchers("/student/viewall").hasRole("ADMIN")
            .antMatchers("/student/view/**").hasAnyRole("ADMIN", "USER")
            .anyRequest().authenticated()
            .and()
            .formLogin()
            .loginPage("/login")
            .permitAll()
            .and()
            .logout()
            .permitAll();
    }
}
```

Ini adalah tampilan outputnya.



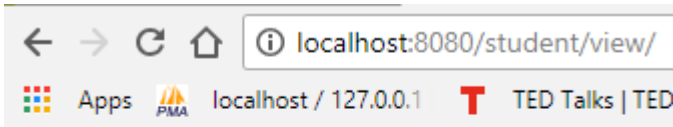
Welcome

[Login](#)

view student by npm

NPM

Lihat



NPM = 1606954703

Name = Aisyah Sadiyah

GPA = 3.0

Login as user

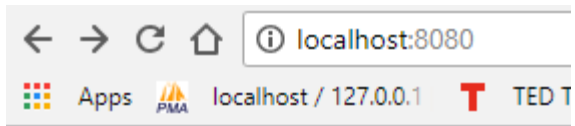
Sign Out

4. Buat agar user dengan role ADMIN dapat melihat view all student dan view student by npm juga.

Pada index.html ditambahkan code seperti di bawah ini untuk role admin dapat melihat view all student.

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org" >
  <head>
    <title>Index</title>
  </head>
  <body>
    <h2>Welcome</h2>
    <a href="/login">Login</a>
    <a th:if="${#httpServletRequest.isUserInRole('ADMIN')}" href="/student/viewall">View All Student</a><br/>
    <div th:if="${#httpServletRequest.isUserInRole('ADMIN')} || #httpServletRequest.isUserInRole('USER')}">
      <form th:action="@{/student/view/}" method="post">
        <h3>view student by npm</h3>
        <div class="form-row">
          <div class="form-group col-md-4">
            <label for="npm">NPM</label>
          </div>
          <div class="form-group col-md-4">
            <input type="text" name="npm" class="form-control"/>
          </div>
          <div class="col-md-4">
            <button type="submit" class="btn btn-primary">Lihat</button>
          </div>
        </div>
      </form>
    </div>
  </body>
</html>
```

Ini adalah tampilan outputnya.



Welcome

[Login](#) [View All Student](#)

view student by npm

NPM

Lihat