# Spring Security

Конфигурационный файл:

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11.  12.  13.  14.  15.  16.  17.  18.  19.  20.  21.  22.  23.  24.  25.  26.  27.  28.  29.  30.  31.  32.  33.  34.  35.  36.  37.  38.  39.  40.  41. | @EnableWebSecurity  public class SecurityConfig  extends WebSecurityConfigurerAdapter {  @Override  protected void configure(HttpSecurity http)  throws Exception {  http.authorizeRequests()  // авторизированные пользователи  .antMatchers("/authenticated/\*\*")  .authenticated()  // доступ по ролям  .antMatchers("/admin/\*\*")  .hasAnyRole("ADMIN", "SUPERADMIN")  // доступ по правам authority  .antMatchers("/profile/\*\*")  .hasAuthority()  .and()  // всплывающее окно  .httpBasic()  // своя форма логина  .formLogin()  // url страницы для входа  .loginProcessingUrl("/hellologin")  // ...  .successForwardUrl("/authenticated")  // страница успешного входа  .defaultSuccessUrl("/authenticated")  // обработчик успешной аутентификации  .successHandler()  .and()  // страница после выхода  .logout().logoutSuccessUrl("/");  }  } |

Стандартная форма входа:

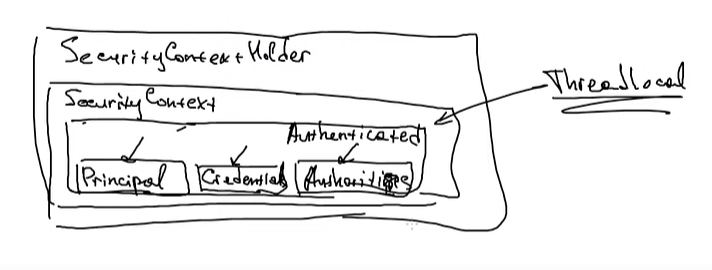
|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11.  12.  13.  14.  15.  16.  17.  18.  19.  20.  21.  22.  23.  24. | <form class="form-signin" method="post"  action="/security/login">  <h2 class="form-signin-heading">  Please sign in</h2>  <p>  <label for="username" class="sr-only">  Username</label>  <input type="text" id="username"  name="username" class="form-control"  placeholder="Username" required autofocus>  </p>  <p>  <label for="password" class="sr-only">  Password</label>  <input type="password" id="password"  name="password" class="form-control"  placeholder="Password" required>  </p>  <input name="\_csrf" type="hidden"  value="b65c8057-6296-44b9-af77-23158cccb80d"  />  <button class="btn btn-lg btn-primary btn-block"  type="submit">Sign in</button>  </form> |

Выделенные атрибуты name не стоит изменять. Их считывает Spring Security.

Для входа без БД можно воспользоваться логином user и паролем, сгенерированным в консоли.

Объект Principal можно заключить в параметры метода контроллера и получить информацию о пользователе:

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5. | @GetMapping("/authenticated")  public String pageForAuthenticatedUser(  Principal principal) {  return principal.getName();  } |



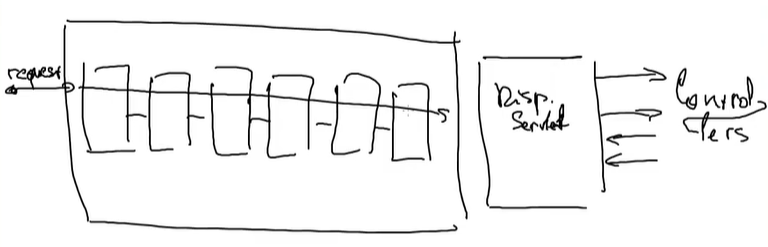
SecurityContextHolder — основное хранилище.

SecurityContext — хранилище данных, хранит данные в ThreadLocal переменной (для каждого потока свои данные)

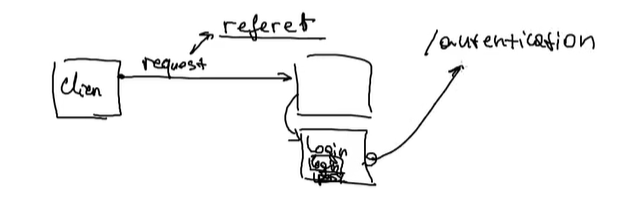
Authenticated: Principal (информация о пользователе), Credentials (пароль, который нужно проверить), Authorities (права доступа).

Credentials чистится сразу после проверки пароля. Principal не хранит в себе пароля. Сделано в целях безопасности.

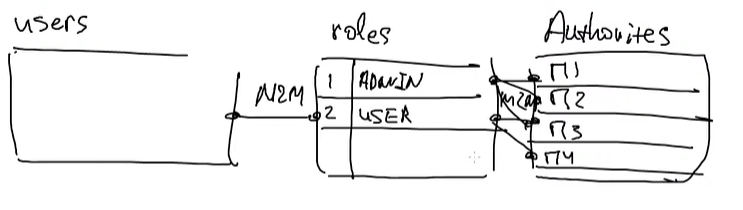
Данные храниться во ThreadLocal переменной тоже в целях безопасности. Пользователь в своем потоке работает, и только о себе информацию знает.



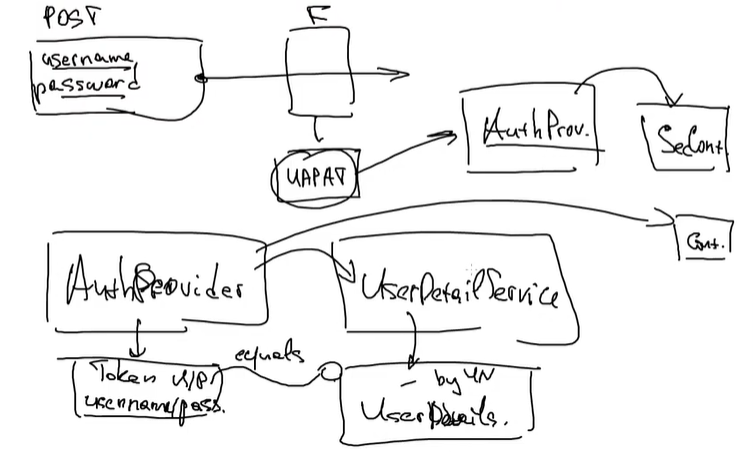
Процесс аутентификация происходит до диспетчера сервлета и обрабатывается множеством фильтров.



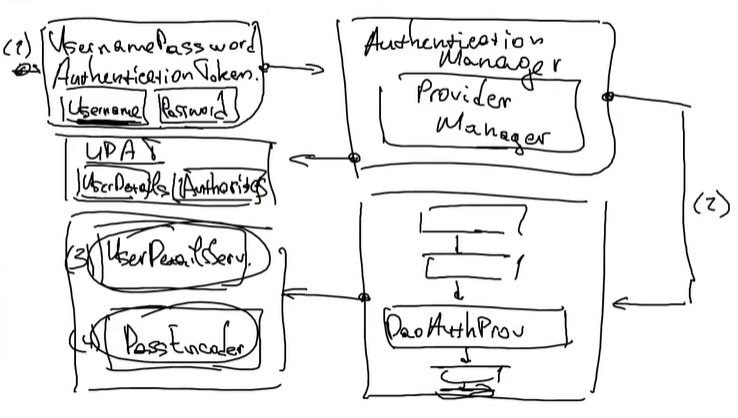
Asd



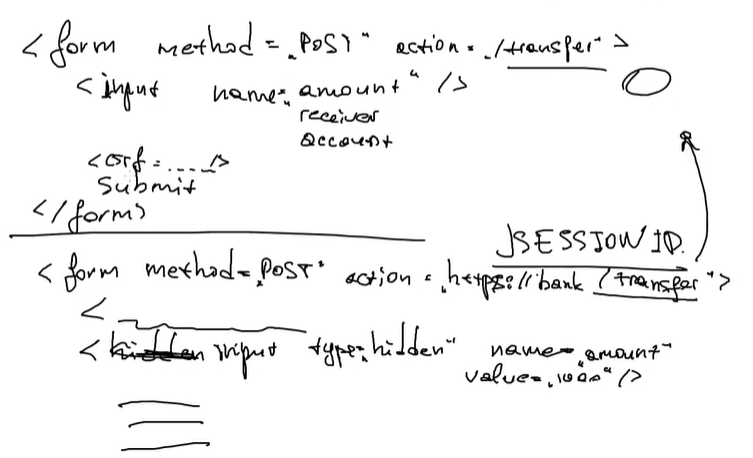
asd



Asd



Asdf



увфы