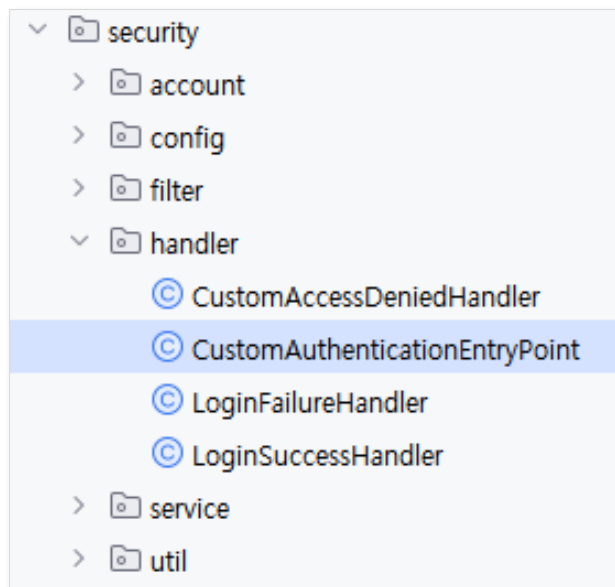


2025년 상반기 K-디지털 트레이닝

Spring Security JWT 인증2 (심화 1)

[KB] IT's Your Life

- ✓ 인증 실패시 401 에러 처리를 하는 `AuthenticationEntryPoint`를 정의하세요.
 - JSON 응답 처리로 변경해야 함



security.handler.CustomAuthenticationEntryPoint.java

```
package org.scoula.security.handler;
```

```
...
```

```
@Log4j2
```

```
@Component
```

```
public class CustomAuthenticationEntryPoint implements AuthenticationEntryPoint {
```

```
    @Override
```

```
    public void commence(HttpServletRequest request, HttpServletResponse response, AuthenticationException authException)
```

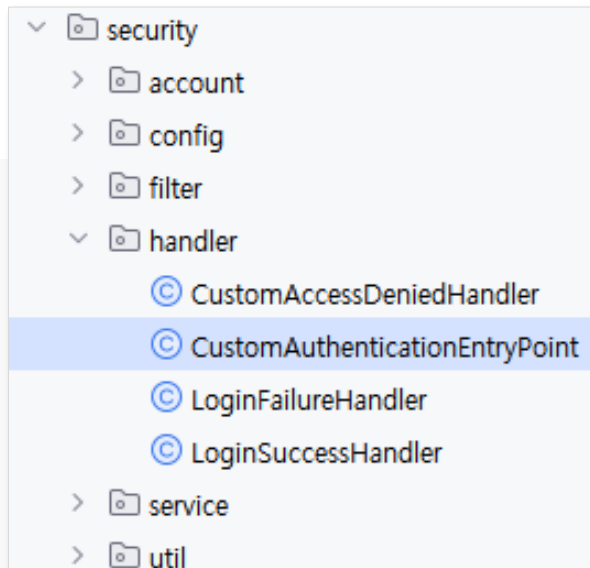
```
        throws IOException, ServletException {
```

```
        log.error("===== 인증 에러 =====");
```

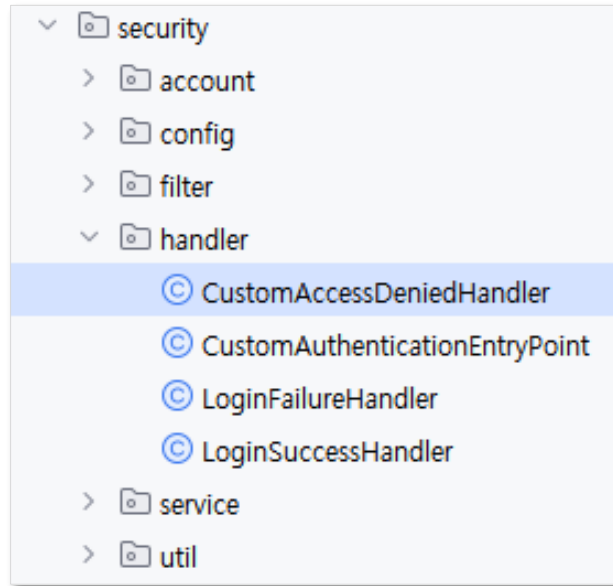
```
        JsonResponse.sendError(response, HttpStatus.UNAUTHORIZED, authException.getMessage());
```

```
    }
```

```
}
```



- ✓ 인증 실패시 403 에러 처리를 하는 CustomAccessDeniedHandler를 정의하세요.
 - JSON 응답 처리로 변경해야 함

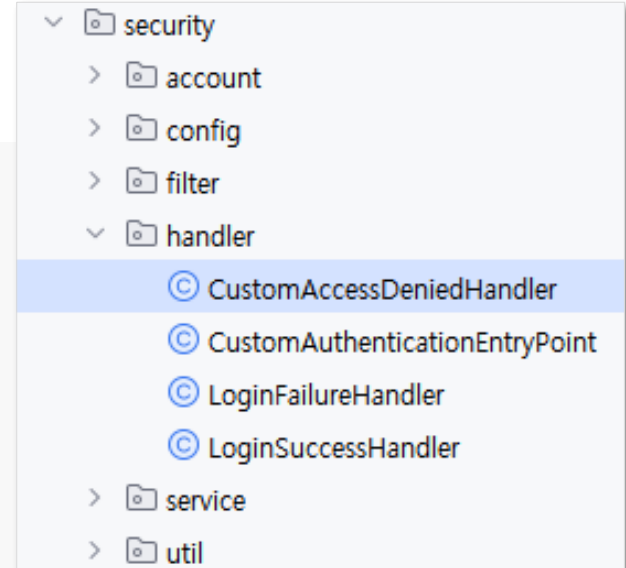


CustomAccessDeniedHandler.java

```
package org.scoula.security.handler;
...

@Component
@Log4j2
public class CustomAccessDeniedHandler implements AccessDeniedHandler {

    @Override
    public void handle(HttpServletRequest request, HttpServletResponse response,
        AccessDeniedException accessDeniedException) throws IOException, ServletException {
        log.error("===== 인가 에러 =====");
        JsonResponse.sendError(response, HttpStatus.FORBIDDEN, "권한이 부족합니다.");
    }
}
```



- ✓ **AuthenticationEntryPoint를 SecurityConfig에 설정하세요.**

security.config.SecurityConfig.java

```
public class SecurityConfig extends WebSecurityConfigurerAdapter {  
    ...  
  
    private final JwtUsernamePasswordAuthenticationFilter jwtUsernamePasswordAuthenticationFilter;  
  
    private final CustomAccessDeniedHandler accessDeniedHandler;  
    private final CustomAuthenticationEntryPoint authenticationEntryPoint;  
    ...  
  
    @Override  
    public void configure(HttpSecurity http) throws Exception {  
        // 한글 인코딩 필터 설정  
        http.addFilterBefore(encodingFilter(), CsrfFilter.class)  
            ...;  
        // 예외 처리 설정  
        http  
            .exceptionHandling()  
            .authenticationEntryPoint(authenticationEntryPoint)  
            .accessDeniedHandler(accessDeniedHandler);  
        ...  
    }  
}
```

✔ jwt 인증 테스트를 위한 Rest 컨트롤러를 정의하세요.

- /api/security/all
 - 인증 없이 접근 가능
- /api/security/member
 - ROLE_MEMBER가 있어야 접근 가능
- /api/security/admin
 - ROLE_ADMIN이 있어야 접근 가능

- 토큰 유효 시간을 10분으로 설정
 - JwtProcessor
 - `TOKEN_VALID_MILLISECOND = 1000L * 60 * 10;`

controller.SecurityController.java

```
package org.scoula.controller;
```

```
...
```

```
@Log4j2
```

```
@RequestMapping("/api/security")
```

```
@RestController
```

```
public class SecurityController {
```

```
    @GetMapping("/all")
```

```
    public ResponseEntity<String> doAll() {
```

```
        log.info("do all can access everybody");
```

```
        return ResponseEntity.ok("All can access everybody");
```

```
    }
```

```
    @GetMapping("/member")
```

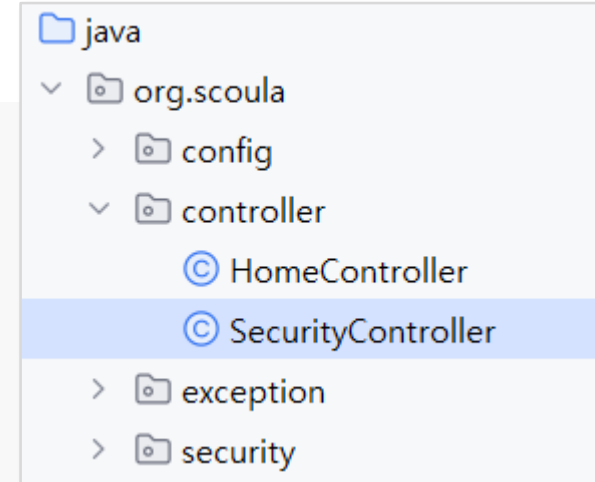
```
    public ResponseEntity<String> doMember(Authentication authentication) {
```

```
        UserDetails userDetails = (UserDetails)authentication.getPrincipal();
```

```
        log.info("username = " + userDetails.getUsername());
```

```
        return ResponseEntity.ok(userDetails.getUsername());
```

```
    }
```



controller.SecurityController.java

```
@GetMapping("/admin")
public ResponseEntity<MemberVO> doAdmin(@AuthenticationPrincipal CustomUser customUser) {
    MemberVO member = customUser.getMember();
    log.info("username = " + member);
    return ResponseEntity.ok(member);
}
```

- ✔ 앞에서 정의한 URL에 대해 SecurityConfig에 설정하세요.

security.config.SecurityConfig.java

```
public class SecurityConfig extends WebSecurityConfigurerAdapter {  
    ...  
  
    @Override  
    public void configure(HttpSecurity http) throws Exception {  
        ...  
  
        http  
            .authorizeRequests() // 경로별 접근 권한 설정  
            .antMatchers(HttpMethod.OPTIONS).permitAll()  
            .antMatchers("/api/security/all").permitAll() // 모두 허용  
            .antMatchers("/api/security/member").access("hasRole('ROLE_MEMBER')") // ROLE_MEMBER 이상 접근 허용  
            .antMatchers("/api/security/admin").access("hasRole('ROLE_ADMIN')") // ROLE_ADMIN 이상 접근 허용  
            .anyRequest().authenticated(); // 나머지는 로그인 된 경우 모두 허용  
    }  
    ...  
}
```

2025년 상반기 K-디지털 트레이닝

Spring Security JWT 인증2 (심화 2)

[KB] IT's Your Life

✓ 각 상황을 Talend API Tester로 테스트 하세요.

- 로그인 안 한 상태로 접근하기
 - `http://localhost:8080/api/security/all`
 - `http://localhost:8080/api/security/member`
- 로그인 하기
- 로그인 한 상태로 접근하기
 - `http://localhost:8080/api/security/member`
 - `http://localhost:8080/api/security/admin`
- 10분 경과, 토큰 유효 시간 이후 접근
 - `http://localhost:8080/api/security/admin`

✓ 로그인 안한 상태

- <http://localhost:8080/api/security/all>

The screenshot shows a web browser's developer tools interface. At the top, the 'METHOD' dropdown is set to 'GET' and the 'URL' field contains 'http://localhost:8080/api/security/all'. A 'Send' button is visible next to the URL. Below the URL, the 'QUERY PARAMETERS' section is empty. The 'HEADERS' section is set to 'Form' view, showing '+ Add header' and 'Add authorization' buttons. The 'BODY' section shows a message: 'XHR does not allow payloads for GET request.' Below this, the 'Response' section is displayed, showing a status code of '200'. The 'Response' section has a 'Cache Detected - Elapsed Time: 5ms' indicator. The 'HEADERS' section is set to 'pretty' view, showing the following headers: 'X-Content-Type: nosniff', 'X-XSS-Protection: 1; mode=block', 'Cache-Control: no-cache, no-store, max-age=0, must-revalidate', 'Pragma: no-cache', and 'Expires: 0'. The 'BODY' section is set to 'raw' view, showing the text 'All can access everybody'. A 'copy' button is visible next to the body text. The length of the response is indicated as 'length: 24 bytes'.

✓ 로그인 안한 상태

- <http://localhost:8080/api/security/member>

The screenshot shows a web browser's developer tools interface. At the top, the 'METHOD' is set to 'GET' and the 'URL' is 'http://localhost:8080/api/security/member'. The 'Send' button is visible. Below the URL bar, the 'QUERY PARAMETERS' section is empty. The 'HEADERS' section is expanded, showing a 'Form' dropdown and buttons for '+ Add header' and 'Add authorization'. The 'BODY' section is also expanded, displaying the message 'XHR does not allow payloads for GET request.' Below this, the 'Response' section is highlighted with a red background, showing a '401' status code. The 'Response' section also indicates 'Cache Detected - Elapsed Time: 4ms'. Below the status code, the 'HEADERS' and 'BODY' sections are expanded. The 'HEADERS' section shows the following headers: 'X-Content-Type: nosniff', 'X-XSS-Protection: 1; mode=block', 'Cache-Control: no-cache, no-store, max-age=0, must-revalidate', 'Pragma: no-cache', 'Expires: 0', and 'X-Frame-Options: DENY'. The 'BODY' section shows the message 'Unexpected character (F) at position 0' and 'Full authentication is required to access this resource'. The 'length' of the response is 55 bytes.

METHOD: GET
SCHEME :// HOST [":" PORT] [PATH ["?" QUERY]]
http://localhost:8080/api/security/member
length: 41 byte(s)
Send

QUERY PARAMETERS

HEADERS ?
Form
+ Add header
Add authorization

BODY ?
XHR does not allow payloads for GET request.

Response
Cache Detected - Elapsed Time: 4ms
401

HEADERS ?
pretty
X-Content-Type: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY

BODY ?
pretty
Unexpected character (F) at position 0
Full authentication is required to access this resource
lines nums copy
length: 55 bytes

- <http://localhost:8080/api/auth/login>

토큰값 복사

✓ 로그인 상태(user0, ROLE_MANAGER)

- <http://localhost:8080/api/security/member>
- 요청 헤더에 토큰 직접 설정 Authorization: **Bearer** 토큰_문자열

The screenshot shows a REST client interface. The top section displays the request details: METHOD is GET, and the URL is `http://localhost:8080/api/security/member`. The length of the request is 41 byte(s). Below the URL bar, the 'HEADERS' tab is active, showing a single header: `Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTJ5In0...`. The 'BODY' tab is also visible, showing a message: `XHR does not allow payloads for GET request.`. The bottom section shows the 'Response' status: `200`. The response headers include `X-Content-Type-Options: nosniff`, `X-XSS-Protection: 1; mode=block`, and `Cache-Control: no-cache, no-store, max-age=0, must-revalidate`. The response body is `user0`.

✓ 로그인 상태(user0, ROLE_MANAGER)

- <http://localhost:8080/api/security/admin>
- 요청 헤더에 토큰 직접 설정 Authorization: Bearer 토큰_문자열

METHOD: GET
SCHEME :// HOST [":" PORT] [PATH ["?" QUERY]]
http://localhost:8080/api/security/admin
length: 40 byte(s)
Send

QUERY PARAMETERS

HEADERS
Authorization: Bearer eyJhbGciOiJIUzM4Ni...

Form

BODY
XHR does not allow payloads for GET request.

Response
Cache Detected - Elapsed Time: 19ms

403

HEADERS
X-Content-Type-... nosniff
X-XSS-Protection... 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0

pretty

BODY
Unexpected character (권) at position 0
권한이 부족합니다.

✓ 로그인 상태(admin, ROLE_ADMIN)

- <http://localhost:8080/api/security/admin>
- 요청 헤더에 토큰 직접 설정 Authorization: **Bearer** 토큰_문자열

METHOD: GET | SCHEME :// HOST [":" PORT] [PATH ["?" QUERY]] | http://localhost:8080/api/security/admin | length: 40 byte(s) | Send

QUERY PARAMETERS

HEADERS: ☒ Authorization : Bearer eyJhbGciOiJIUzI4NCJ9.e | + Add header | Add authorization

BODY: XHR does not allow payloads for GET request.

Response | Cache Detected - Elapsed Time: 40ms

200

HEADERS: X-Content-Type-Opt... nosniff, X-XSS-Protection: 1; mode=block, Cache-Control: no-cache, no-store, max-age=0, must-revalidate, Pragma: no-cache, Expires: 0

BODY: {
 username: "admin",
 password: "\$2a\$10\$EsIMfxbJ6NuvwX7MDj4WqOYFzLU9U/lddCyn0nic5dFo3VfJYrXYC",
 email: "admin@galapgos.org",
 regDate: 1721790112000,

✓ 로그인 상태(admin, ROLE_ADMIN)

- <http://localhost:8080/api/security/admin>
- 요청 헤더에 잘못된 토큰 직접 설정 Authorization: Bearer 토큰_문자열x

The screenshot shows a REST client interface with the following details:

- Request:**
 - METHOD: GET
 - URL: `http://localhost:8080/api/security/admin`
 - QUERY PARAMETERS: (empty)
 - HEADERS:
 - Authorization: Bearer eyJhbGciOiJIUzI4NCJ9.e
 - BODY: XHR does not allow payloads for GET request.
- Response:**
 - Status: 401
 - HEADERS:
 - X-Content-Type-Options: nosniff
 - X-XSS-Protection: 1; mode=block
 - Cache-Control: no-cache, no-store, max-age=0, must-revalidate
 - Pragma: no-cache
 - Expires: 0
 - BODY:
 - Unexpected character (M) at position 0
 - Malformed JWT JSON: {"alg":"HS31

✓ 10분 경과, 토큰 유효 시간 이후 접근

○ <http://localhost:8080/api/security/admin>

The screenshot shows a web browser's developer tools interface. The top section is for the request, showing a GET method to the URL `http://localhost:8080/api/security/admin`. The request headers include an Authorization header with the value `Bearer eyJhbGciOiJIUzI4NCJ9.e`. The response section shows a 401 status code. The response headers include `X-Content-Type-Options: nosniff`, `X-XSS-Protection: 1; mode=block`, `Cache-Control: no-cache, no-store, max-age=0, must-revalidate`, `Pragma: no-cache`, and `Expires: 0`. The response body contains the message `Unexpected character (토큰) at position 0` and `토큰의 유효시간이 지났습니다.` (Token's validity period has expired).

METHOD: GET
SCHEME :// HOST [":" PORT] [PATH ["?" QUERY]]
`http://localhost:8080/api/security/admin`
length: 40 byte(s)
Send

QUERY PARAMETERS

HEADERS: Authorization : Bearer eyJhbGciOiJIUzI4NCJ9.e
BODY: XHR does not allow payloads for GET request.

Response
Cache Detected - Elapsed Time: 116ms

401

HEADERS: X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0

BODY: Unexpected character (토큰) at position 0
토큰의 유효시간이 지났습니다.