KB금융그룹 | 국민의 평생 금융파트너

2025년 상반기 K-디지털 트레이닝

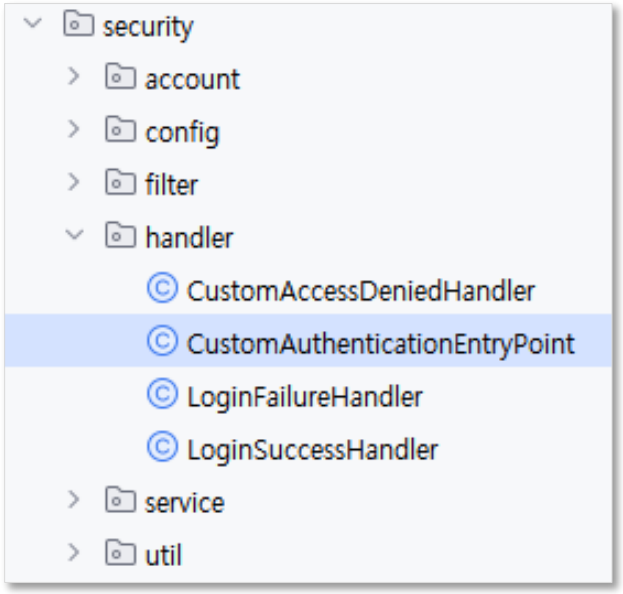
Spring Security JWT 인증2 (심화 1)

[KB] IT's Your Life

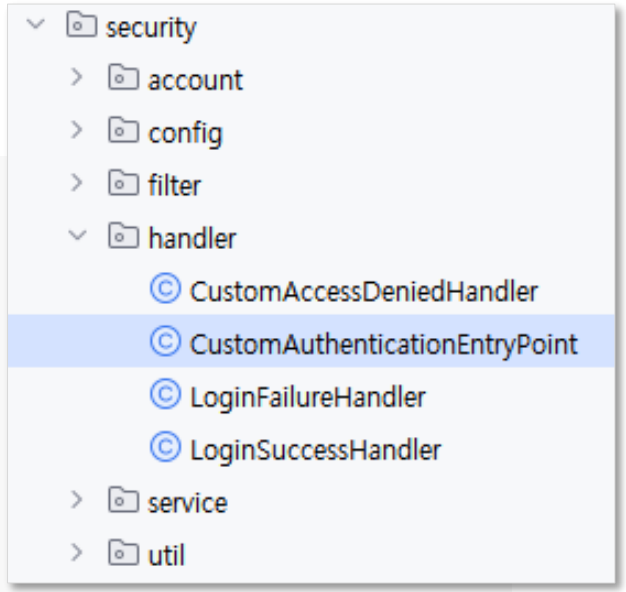
KB 국민은행

Spring Security JWT 인증2

- 인증 실패시 401 에러 처리를 하는 AuthenticationEntryPoint를 정의하세요.
  - JSON 응답 처리로 변경해야 함



Spring Security JWT 인증2

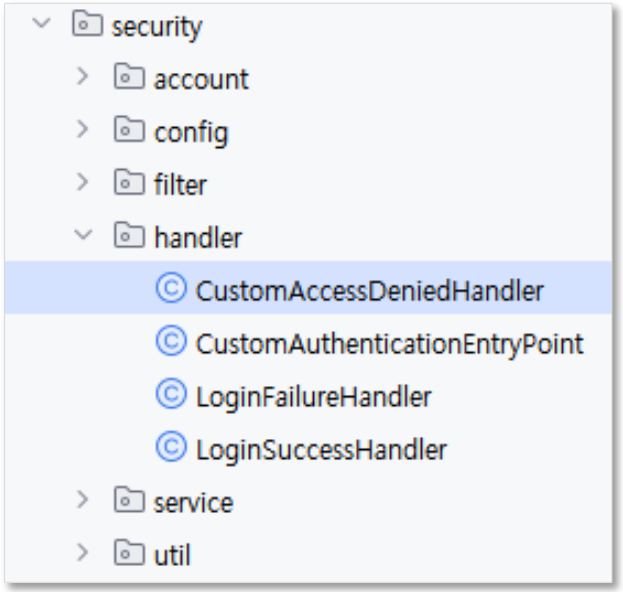


security.handler.CustomAuthenticationEntryPoint.java

```
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());
    }
}
```

Spring Security JWT 인증2

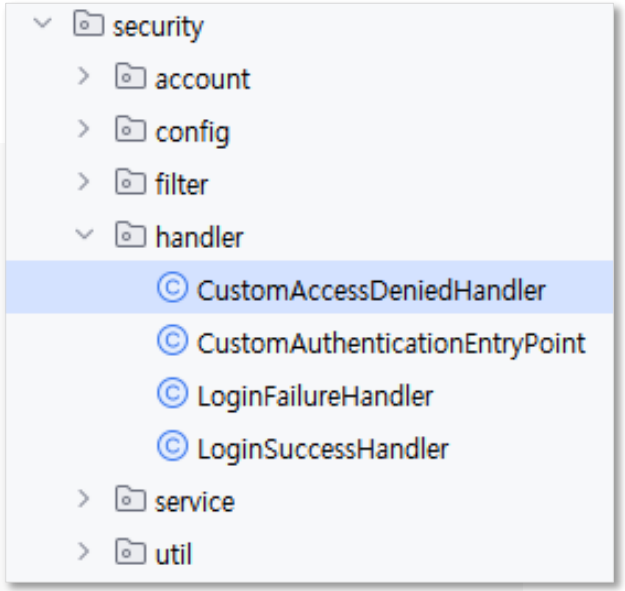
- **인증 실패시 403 에러 처리를 하는 CustomAccessDeniedHandler를 정의하세요.**
  - JSON 응답 처리로 변경해야 함



Spring Security JWT 인증2

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.");
  }
}
```



Spring Security JWT 인증2

- AuthenticationEntryPoint를 SecurityConfig에 설정하세요.

Spring Security JWT 인증2

security.config.SecurityConfig.java

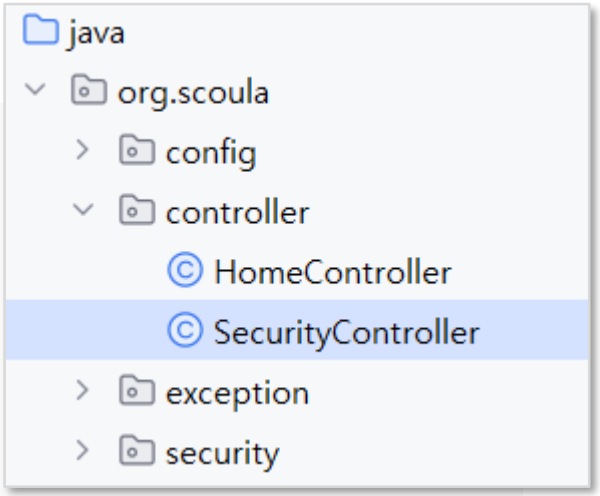
```
public class SecurityConfig extends WebSecurityConfigurerAdapter {
                                ...
  private final JwtUsernamePasswordAuthenticationFilter jwtUsernamePasswordAuthenticationFilter;

  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);
    ...
  }
```

Spring Security JWT 인증2

- **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_MILISECOND = 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);
  }
}
```

Spring Security JWT 인증2

- 앞에서 정의한 URL에 대해 SecurityConfig에 설정하세요.

Spring Security JWT 인증2

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
   }
   ...
}
```

KB금융그룹 | 국민의 평생 금융파트너

2025년 상반기 K-디지털 트레이닝

Spring Security JWT 인증2 (심화 2)

[KB] IT's Your Life

KB 국민은행

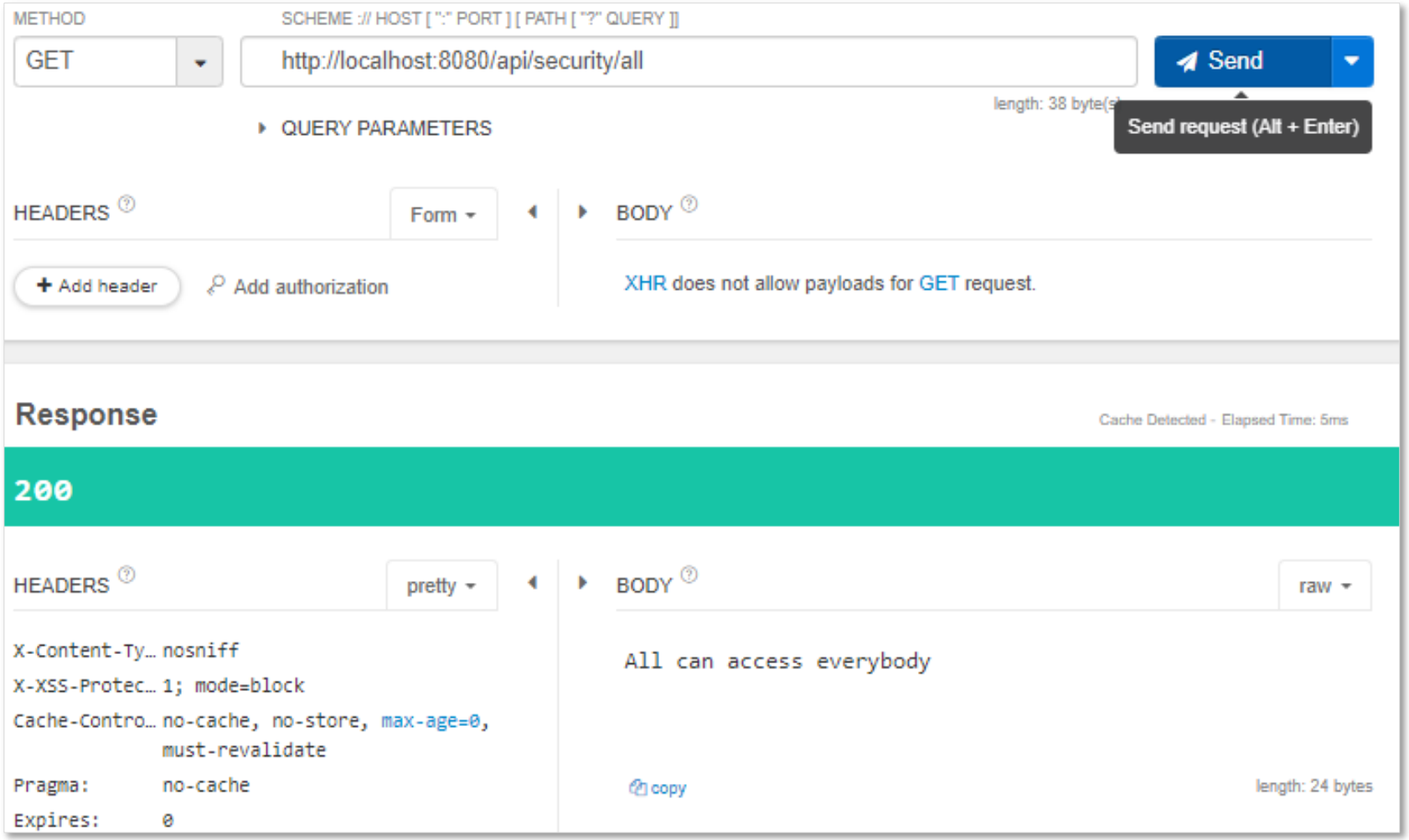
Spring Security JWT 인증2

- **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

Spring Security JWT 인증2

로그인 안한 상태

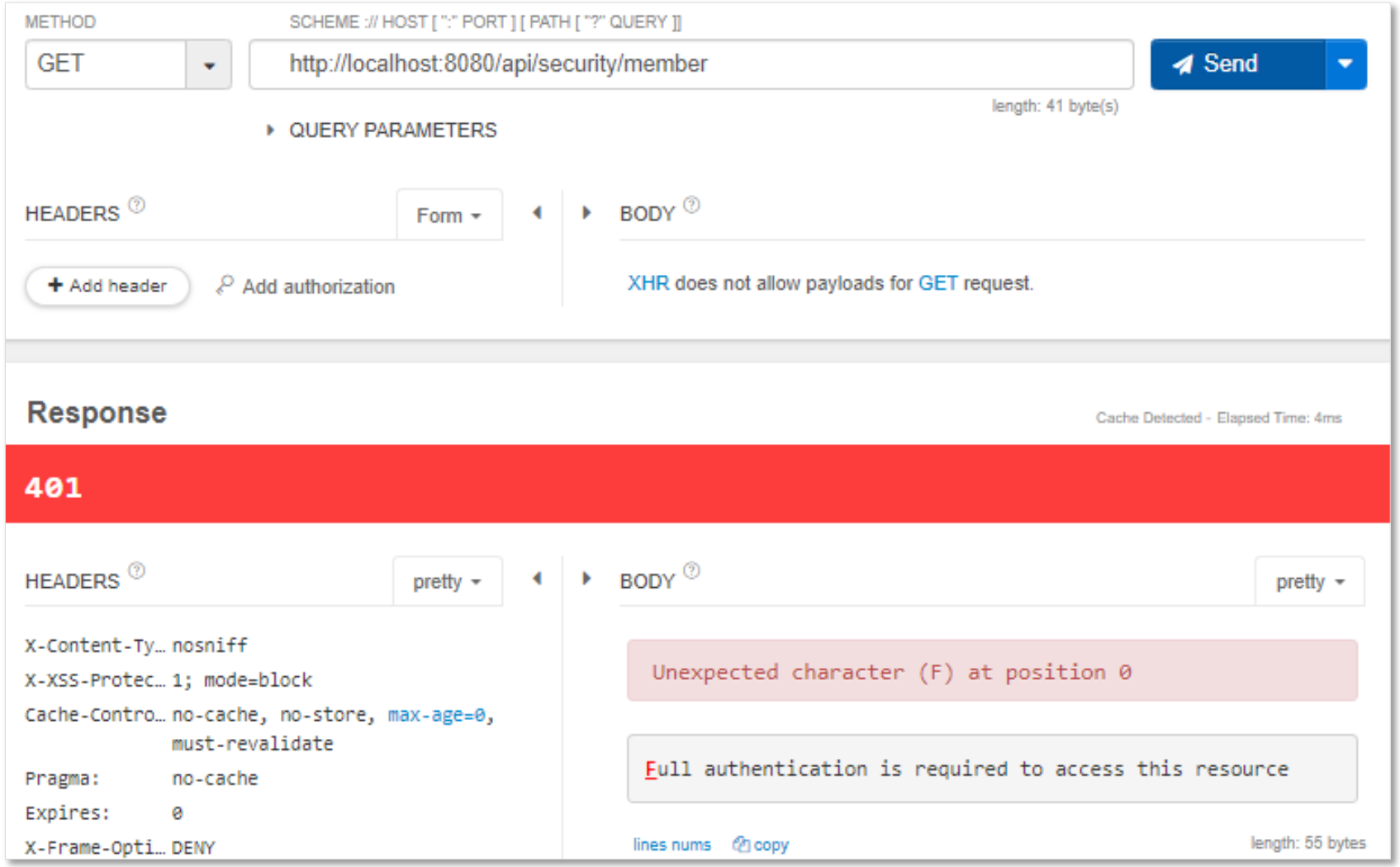
- http://localhost:8080/api/security/all



Spring Security JWT 인증2

로그인 안한 상태

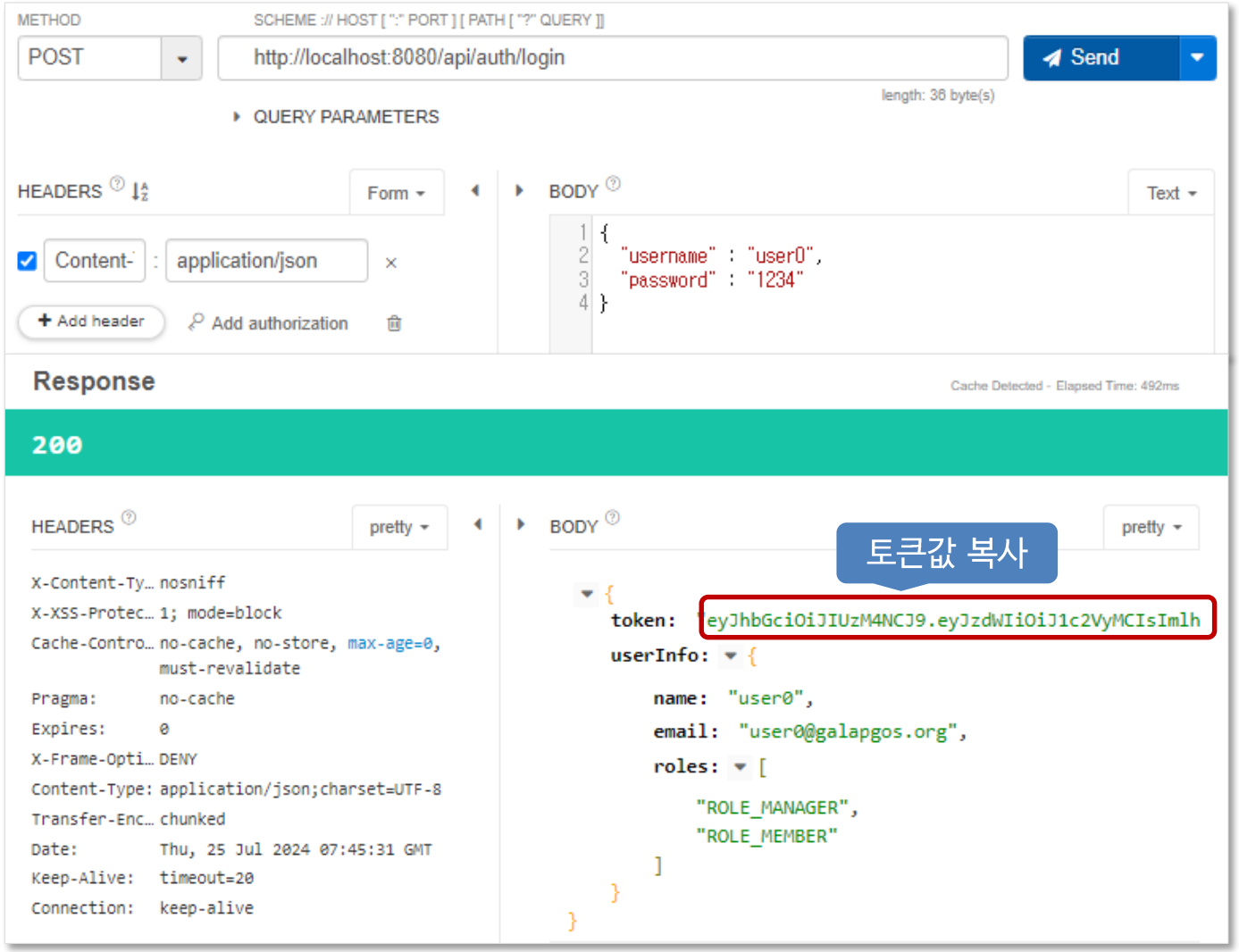
- http://localhost:8080/api/security/member



Spring Security JWT 인증2

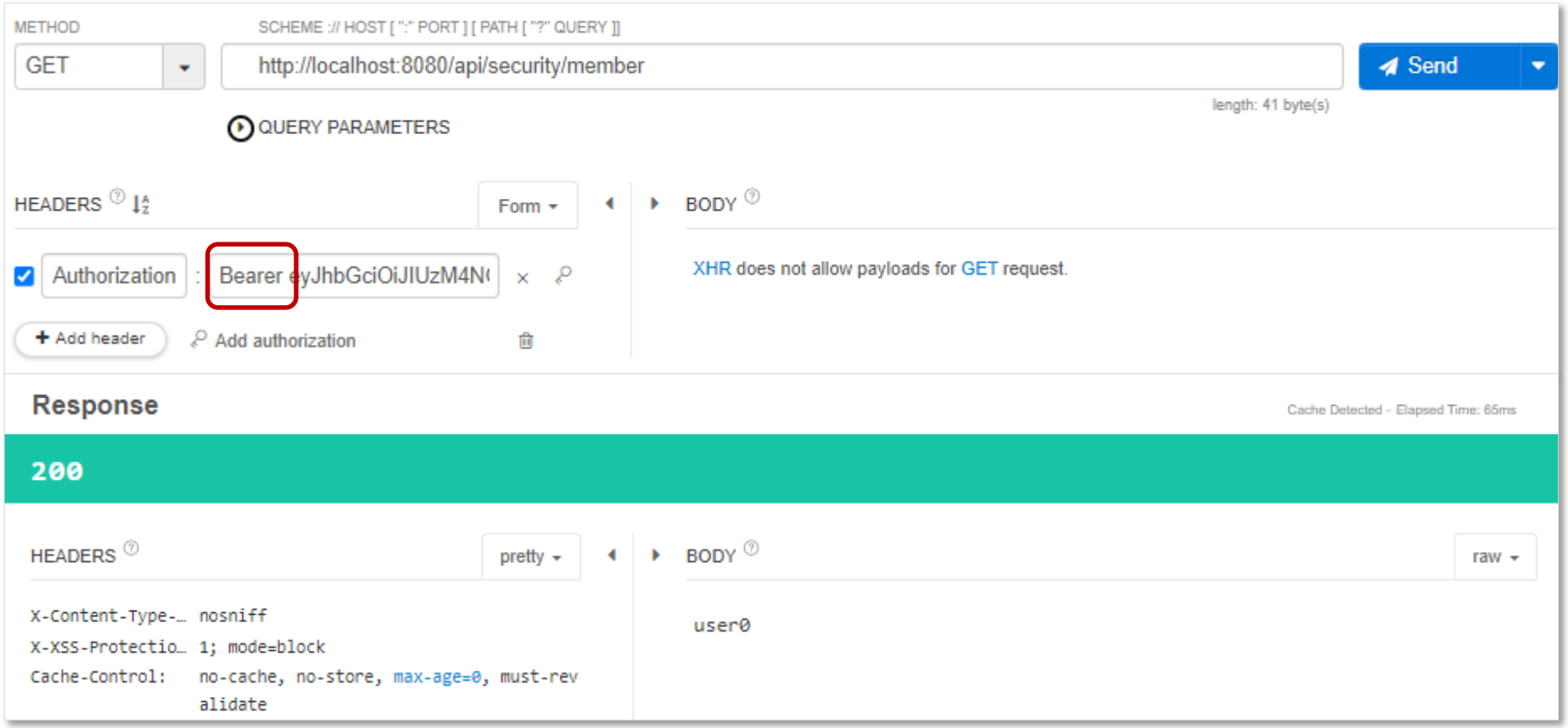
로그인

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



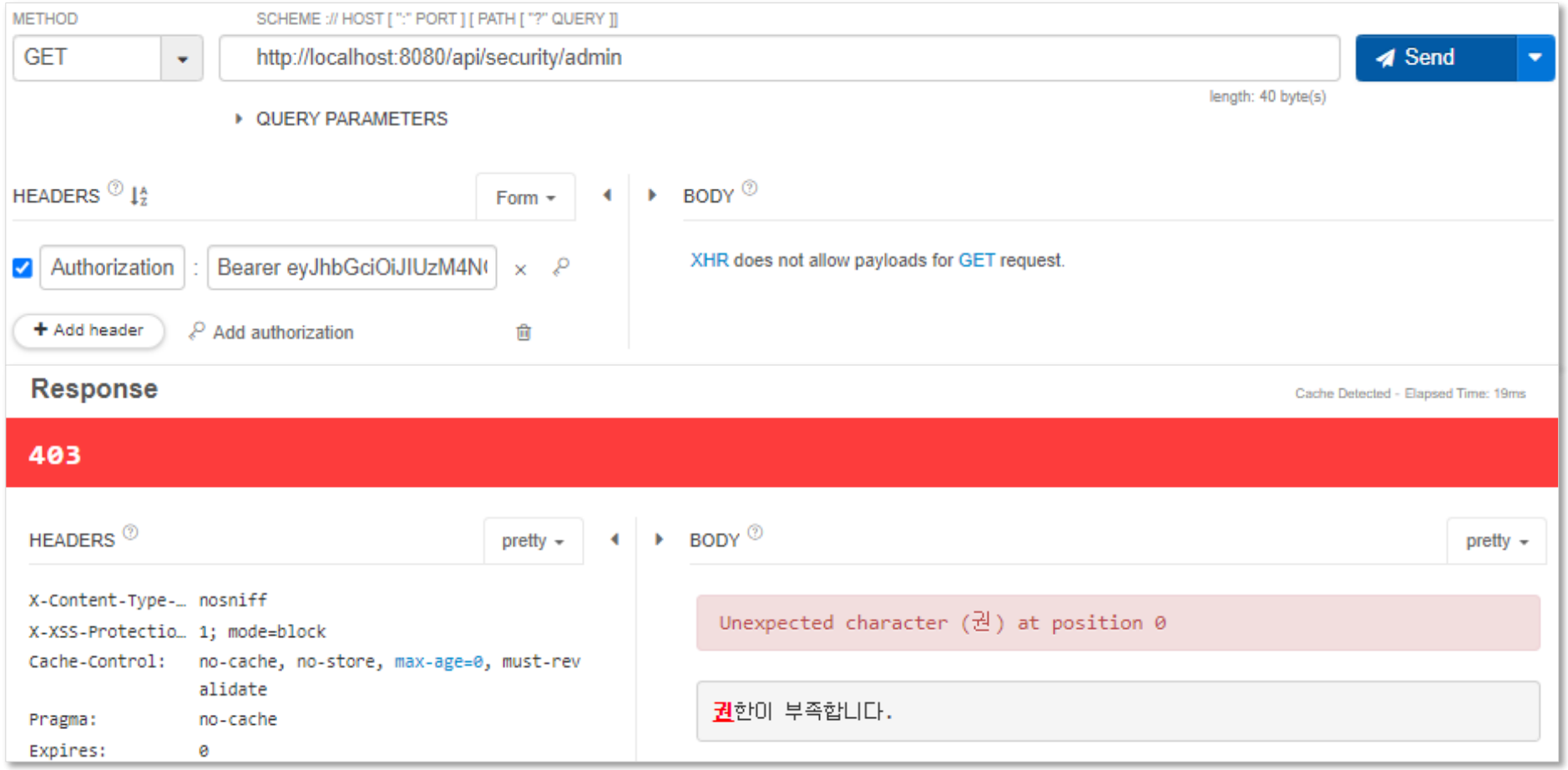
Spring Security JWT 인증2

- 로그인 상태(user0, ROLE_MANAGER)
  - http://localhost:8080/api/security/member
  - 요청 헤더에 토큰 직접 설정 Authorization: Bearer 토큰_문자열



Spring Security JWT 인증2

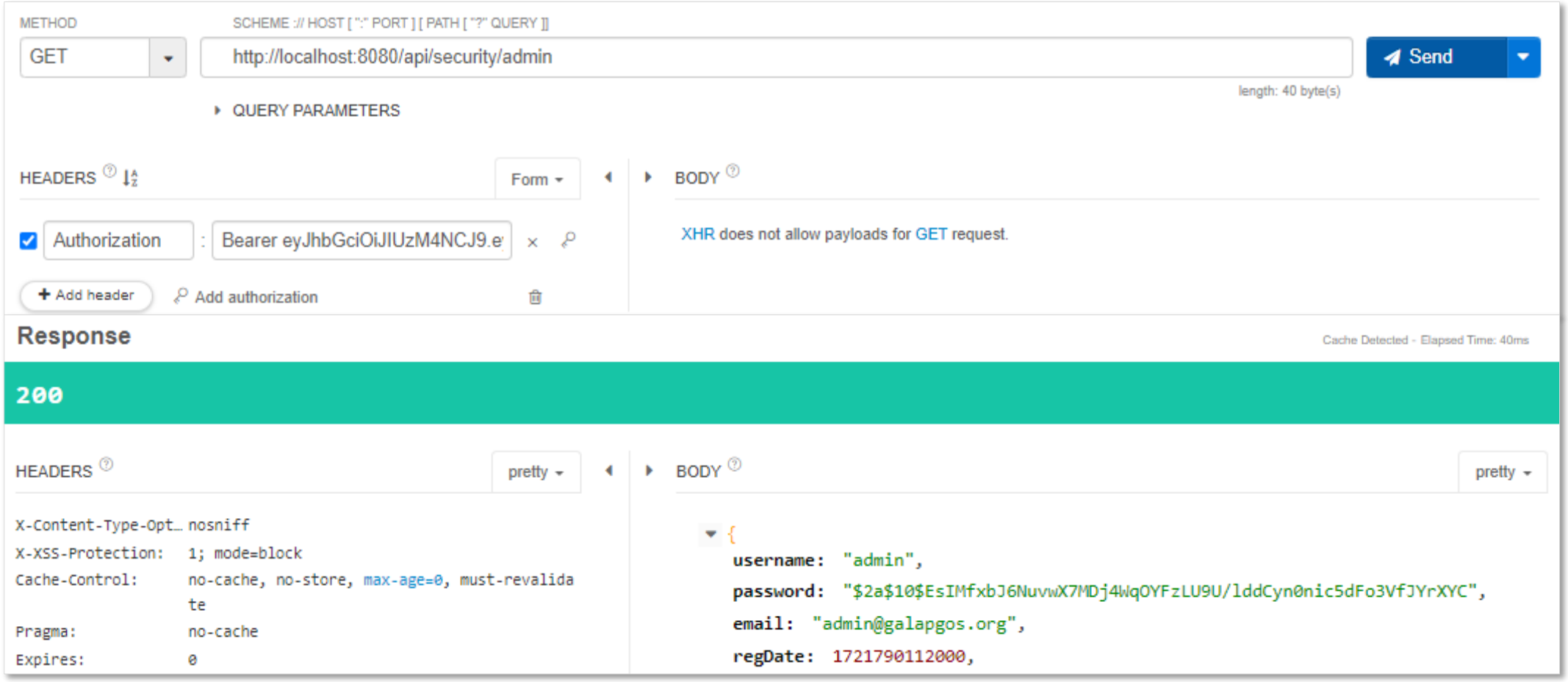
- 로그인 상태(user0, ROLE_MANAGER)
  - http://localhost:8080/api/security/admin
  - 요청 헤더에 토큰 직접 설정 Authorization: Bearer 토큰_문자열



Spring Security JWT 인증2

로그인 상태(admin, ROLE_ADMIN)

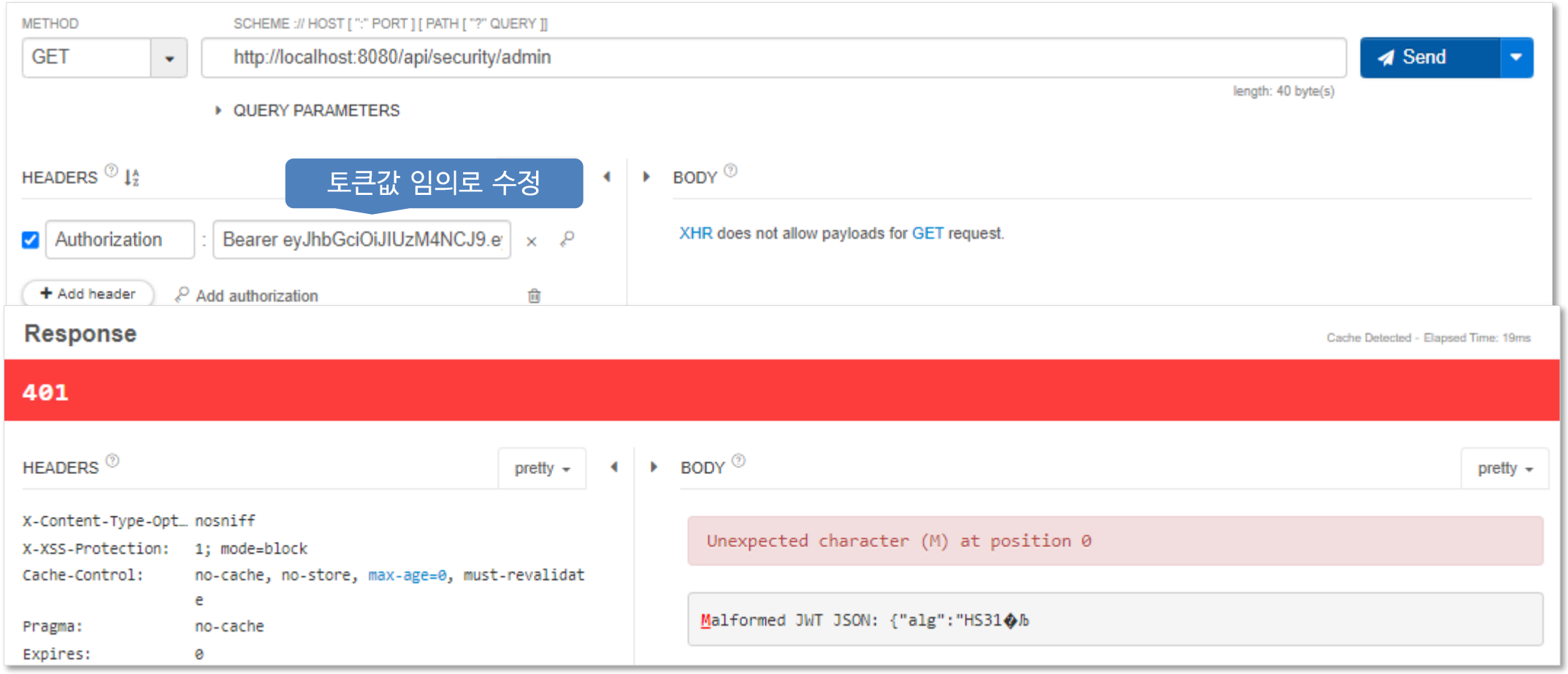
- http://localhost:8080/api/security/admin
- 요청 헤더에 토큰 직접 설정 Authorization: Bearer 토큰_문자열



Spring Security JWT 인증2

로그인 상태(admin, ROLE_ADMIN)

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



Spring Security JWT 인증2

- 10분 경과, 토큰 유효 시간 이후 접근
  - http://localhost:8080/api/security/admin