Spring MVC2 12 [서블릿 필터]

스프링 MVC 2편 - 백엔드 웹 개발 활용 기술

김영한

2022.10.25

서블릿 필터

만약 로그인 한 사용자만 상품 관리 페이지에 들어갈 수 있어야 한다면 ?

필터등의 조건을 적용시켜놓지 않으면 로그인을 하지 않아도 URL로 바로 접속이 가능하다.

이 문제는 로그인 체크 여부를 확인하는 로직을 각각 컨트롤러에 모두 넣어주면 해결 가능하다.

단순한 해결법이지만 반복되는 기능, 만약 로직이 변경된다면? 하는 문제들을
효과적으로 해결 할 필요가 있다.

공통 관심사 (Cross-cutting concern) 를 처리하는 방식으로 스프링 AOP를 사용해도 되지만 웹 관련된 공통 관심사를 처리할 때는 서블릿 필터나 스프링 인터셉터를 사용하는 것이 좋다.

서블릿 필터?

서블릿이 지원하는 수문장 역할을 함

흐름

```
요청 → WAS → 필터 → 서블릿 → 컨트롤러
```

(스프링에서는 서블릿 = 디스패쳐 서블릿 이라 볼 수 있다)

필터를 체인으로 구성되어 중간에 필터를 자유롭게 추가할 수 있다.
요청 → WAS → 필터1 → 필터2 → - - - → 서블릿 → 컨트롤러

서블릿 필터 - 요청 로그 확인

```
@Slf4j
public class LogFilter implements Filter {
   @Override
   public void init(FilterConfig filterConfig) throws ServletException {
   } → 필터가 등록되었을 때 실행

   @Override
   public void doFilter(ServletRequest request, ServletResponse response,
FilterChain chain) throws IOException, ServletException {
   } → 필터가 진행되었을 때 실행

   @Override
   public void destroy() {
   } → 필터가 종료되었을 때 실행
}
```

로그 남기는 로직으로 **doFilter** 수정

```
@Override
public void doFilter(ServletRequest request, ServletResponse response, FilterChain
chain) throws IOException, ServletException {
   log.info("log filter doFilter");
   HttpServletRequest httpRequest = (HttpServletRequest) request;
   String requestURI = httpRequest.getRequestURI();

   String uuid = UUID.randomUUID().toString();
   try {
       log.info("REQUEST [{}][{}]", uuid, requestURI);
       chain.doFilter(request, response);
 → 다음 필터로 넘어감 없으면 그대로 진행.
   } catch (Exception e) {
       throw e;
   } finally {
       log.info("RESPONSE [{}][{}]", uuid, requestURI);
   }
}
```

chain.doFilter : 다음 필터가 있으면 필터가 호출되고 필터가 없으면 서블릿을 호출한다.

→ 이것이 없으면 진행 자체가 안된다.

필터를 적용하는 **Configration**

```
@Configuration
public class WebConfig {

   @Bean
   public FilterRegistrationBean logFilter(){
       FilterRegistrationBean<Filter> filterFilterRegistrationBean = new
FilterRegistrationBean<>();
       filterFilterRegistrationBean.setFilter(new LogFilter());
       filterFilterRegistrationBean.setOrder(1);
            → 순서가 낮을 수록 먼저 실행
       filterFilterRegistrationBean.addUrlPatterns("/*");

       return filterFilterRegistrationBean;
   }
}
```

[logback mdc 검색] - 자동으로 동일한 요청에서 원하는 문구를 넣을 수 있다.

로그인 인증 체크 필터 개발

```
@Slf4j
public class LoginCheckFilter implements Filter {

   //로그인이 필요없는 페이지들
   private static final String[] whitelist = {"/", "/members/add", "/login",
"/css/*"};

   @Override
   public void doFilter(ServletRequest request, ServletResponse response,
FilterChain chain) throws IOException, ServletException {

       HttpServletRequest httpRequest = (HttpServletRequest) request;
       HttpServletResponse httpResponse = (HttpServletResponse) response;

       String requestURI = httpRequest.getRequestURI();

       try {
           log.info("인증 체크 필터 시작{}", requestURI);
```

```
            if(isLoginCheckPath(requestURI)){
                log.info("인증 체크 로직 실행 {} ", requestURI);
                HttpSession session = httpRequest.getSession(false);
                if(session == null ||
                   session.getAttribute(SessionConst.LOGIN_MEMBER) == null){

                    log.info("미인증 사용자 요청 {}", requestURI);
                    //로그인 redirect --> 로그인 시에
                    //다시 돌아오도록 redirectURL을 넣어준다. (로직 개발 필요하다.)
                    httpResponse.sendRedirect("/login?redirectURL=" + requestURI);
                    return;
                }
            }

            chain.doFilter(request, response);

        } catch (Exception e) {
            //예외 로깅이 가능 하지만, 톰캣까지 예외를 보내줘야 한다. 안그러면 작동한다
            throw e;
        } finally {
            log.info("인증 체크 필터 종료 [{}]", requestURI);
        }

    }

    /**
     * 화이트 리스트의 경우 인증 체크X
     */

    private boolean isLoginCheckPath(String requestURI){
        return !PatternMatchUtils.simpleMatch(whitelist, requestURI);
    }

}
```

물론 /login 컨트롤러에서 로그인 성공시 해당 경로로 이동하는 기능은 추가로 개발해야 한다.

`"/login?redirectURL=" + requestURI` → 이부분

로그인 컨트롤러 파라미터에 추가해준 후에 redirect 를 지정하면 된다.

```
, @RequestParam(defaultValue = "/") String redirectURL)
return "redirect:" + redirectURL;
```