

Property, AuthenticationPrincipal 사용하기?

설정파일을 읽고 현재 요청에 대한 사용자 정보를 조회하는 방법을 알아봅니다.

API를 호출 할 때 내 정보를 json등으로 같이 넘기는 것은 client에서 위변조를 할 가능성이 있습니다.

현재 서버가 내어 준 session정보를 토대로 하여 사용자 정보가 무엇인지를 가져오는 것이 좋은 방법입니다!

1. Controller 구현하기

```
public class PropertyController {

    private Logger LOGGER = LoggerFactory.getLogger(PropertyController.class);

    @Value("${logging.path}")
    private String logging_path;

    // 앞서 adapter에서 ignore 해놓은 openapi경로의 URL이다.
    @RequestMapping("/openapi/getProperty")
    public @ResponseBody String getProperty(@AuthenticationPrincipal SecurityMember securityMember) {

        StringBuffer sb = new StringBuffer();

        // 바로 호출할경우 login을 거치지 않기 때문에 SecurityMember 객체를 받아오지 못한다.
        if(securityMember != null) {
            sb.append("securityMember.getIp()=")
              .append(securityMember.getIp())
              .append(" / ");
        }else {
            sb.append("securityMember is null / ");
        }
        sb.append("logging_path=")
          .append(logging_path);

        LOGGER.debug(sb.toString());

        return sb.toString();
    }

    // 로그인 이후 접근할 수 있는 getMember URL
    @RequestMapping("/getMember")
    // @AuthenticationPrincipal annotation을 사용하여 SecurityMember를 자동으로 Autowired받아 사용한다.
    public @ResponseBody String getMember(@AuthenticationPrincipal SecurityMember securityMember) {

        StringBuffer sb = new StringBuffer();

        // 로그인 이후 접근할 수 있으므로 정상적으로 받아온다.
        if(securityMember != null) {
            sb.append("securityMember.getIp()=")
              .append(securityMember.getIp());
        }

        // log를 남긴다.
        LOGGER.debug(sb.toString());

        // sb -> Member정보를 return한다. ResponseBody annotation을 사용했기 때문에 값을 화면에 보여준다.
        return sb.toString();
    }
}
```

1-1. /openapi/~ 경로

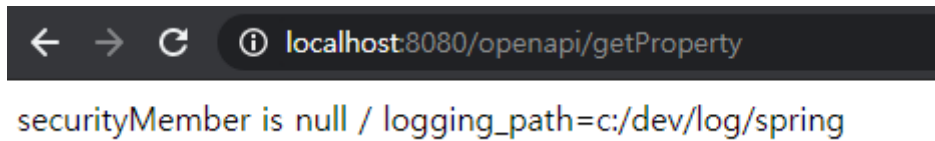
@ResponseBody annotation을 사용해 String 형태로 답을 줍니다.
즉! 화면 이동없이 API 호출처럼 그대로 문자를 response해줍니다.

상단의 /openapi/~ 경로는 앞서 Spring security의 adapter에서 ignore해주었습니다.
따라서 로그인을 거치지 않고 접근할 수 있습니다.

@AuthenticationPrincipal annotation을 통해 SecurityConfig에서 User를 상속받은 객체를
자동으로 autowired해줍니다.

getIp 메소드를 통해 securityMember의 Ip정보를 가져오고
application.yml에 정의된 대로 logging.path를 가져옵니다.

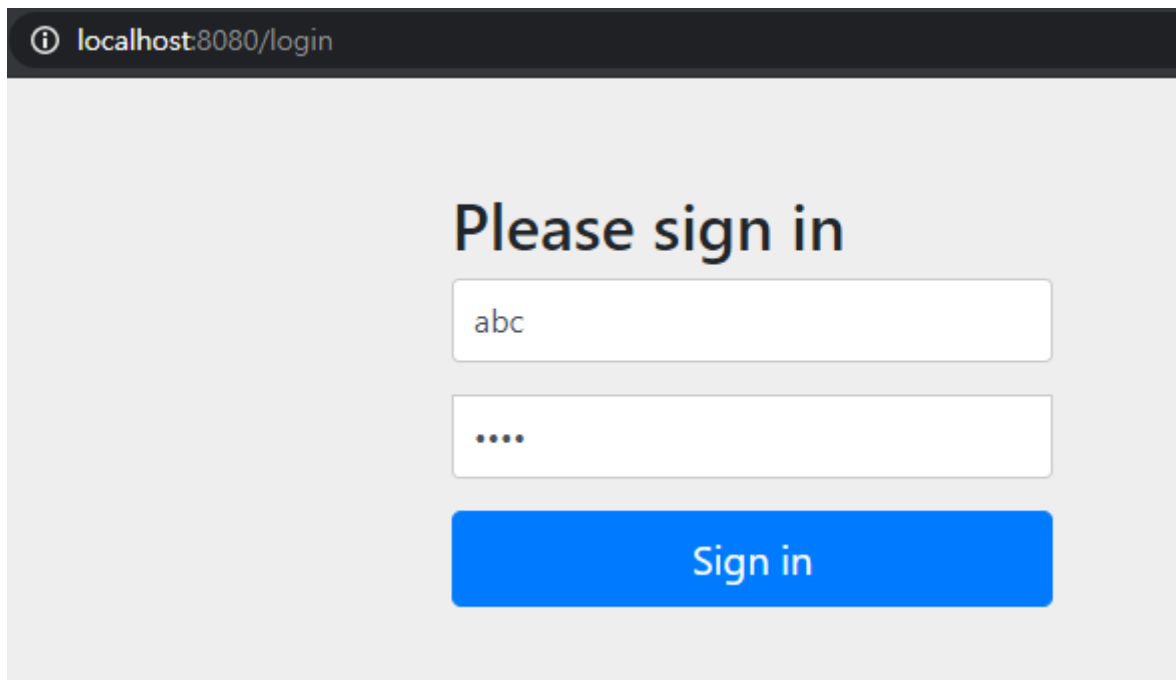
결과적으로는 정상적으로 결과값을 가져오지 못합니다!!
앞서 설명한대로 ignore처리해서 로그인을 거치지 않아 Member정보가 없기 때문!!



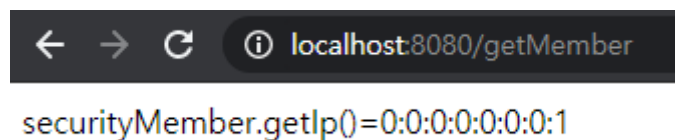
login 화면을 거치지 않고 바로 출력되며 SecurityMember는 가져오지 못했습니다.

1-2. /getMember 경로

위와 다르게 ignore처리되지 않은 경로이므로 login UI를 먼저 거칩니다.
따라서 SecurityMember 객체에 정보를 채웁니다.



로그인을 거친 후 접속하면



이와 같은 값을 반환합니다.

IP주소가 이상한데요??

-> 아닙니다. **IPv4**가 아닌 **IPv6**형태로 값을 반환합니다.

IPv4로는 127.0.0.1 입니다. 따라서 정상적으로 반환됨을 확인할 수 있습니다.

Made by 장경석
