

3/8 개발일지

1. 소셜 로그인 개발작업 진행

▼ 소셜로그인 구현 위한 로직 생각

3/7 개발일지엔 소셜로그인 구현을 전체적으로 이해하지 못하여 복붙단계에 지나지 않았다.

따라서 오늘은 전체적인 흐름을 이해할 예정이다.

일단 Spring Security 에 익숙하지 않아서 이 라이브러리에 익숙해져야겠다는 생각을 했다.

공부해보니 전체적인 흐름은 이렇다고 한다.

- SecurityConfig (제일 앞단에서 흐름구현)

- 아래 코드 참고

```
oauth2Login()  
    .loginPage("http://localhost:3000/login")  
    .defaultSuccessUrl("http://localhost:3000/")  
    .userInfoEndpoint()  
    .userService(oAuth2MemberService)  
    .and()  
    .successHandler(new CustomAuthenticationSuccessHandler())
```

- oauth2login() 즉 소셜로그인이 감지되면 로그인 하지 못하면 .loginPage로 가고
 - 성공하면 .defaultSuccessUrl로 리다이렉트
 - 그 이후에 .userService로 기능구현
 - 만약 또 그 이후에 어떤 이벤트가 필요하면 handler를 이용하여 이후 이벤트 구현

- OAuthController (컨트롤러 → url의 흐름 관장)

- 아래 코드 참고

```

package web.web1.Oauth.controller;

import web.web1.Oauth.domain.Member;
import web.web1.Oauth.domain.MemberRepository;
import lombok.RequiredArgsConstructor;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;

@Controller
@RequiredArgsConstructor
public class OAuthController {
    private final BCryptPasswordEncoder encoder;
    private final MemberRepository memberRepository;
    @GetMapping("/loginForm")
    public String home() {
        return "loginForm";
    }
    @GetMapping("/joinForm")
    public String joinForm() {
        return "joinForm";
    }
    @PostMapping("/join")
    public String join(Member member) {
        String rawPwd = member.getPassword();
        System.out.println("member = " + member);
        member.setRole("ROLE_USER");
        member.setPassword(encoder.encode(rawPwd));
        memberRepository.save(member);
        return "redirect:/loginForm";
    }

    @GetMapping("/private")
    public String privatePage() {
        return "privatePage";
    }
}

```

```

    @GetMapping("/admin")
    public String adminPage() {
        return "adminPage";
    }
}

```

- SecurityConfig에 위치한 url들의 미세한 움직임을 조정한다고 이해했다.
- OAuth2MemberService.java
 - 아래 코드 참고

```

package web.web1.Oauth.domain;

import web.web1.Oauth.oauth2.*;
import lombok.RequiredArgsConstructor;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.oauth2.client.userinfo.DefaultOAuth2UserService;
import org.springframework.security.oauth2.core.OAuth2AuthenticatedUser;
import org.springframework.security.oauth2.core.user.OAuth2User;
import org.springframework.stereotype.Service;

import java.util.Map;
import java.util.Optional;

@Service
@RequiredArgsConstructor
public class OAuth2MemberService extends DefaultOAuth2UserService {
    private final BCryptPasswordEncoder encoder;
    private final MemberRepository memberRepository;

    @Override
    public OAuth2User loadUser(OAuth2UserRequest userRequest) {
        OAuth2User oAuth2User = super.loadUser(userRequest);
        OAuth2MemberInfo memberInfo = null;
        // System.out.println(application.properties);
        System.out.println(userRequest.getClientRegistration());
    }
}

```

```

String registrationId = userRequest.getClientRe
System.out.println("registrationId = " + regist
if (registrationId.equals("google")) {
    memberInfo = new GoogleMemberInfo(oAuth2Use
} else if (registrationId.equals("naver")) {
    memberInfo = new NaverMemberInfo((Map)oAuth
} else if (registrationId.equals("kakao")) {
    memberInfo = new KakaoMemberInfo(oAuth2User
System.out.println("-----
System.out.println(memberInfo);
} else {
    System.out.println("로그인 실패");
}
String provider = memberInfo.getProvider();
String providerId = memberInfo.getProviderId();
String oauth2Id = provider + "_" + providerId;
String username = memberInfo.getName();
String email = memberInfo.getEmail();
String role = "ROLE_ADMIN"; //일반 유저
System.out.println("----asdfasdf----");
System.out.println(oAuth2User.getAttributes());
Optional<Member> findMember = memberRepository.
Member member=null;
if (findMember.isEmpty()) { //찾지 못했다면
    member = Member.builder()
        .oauth2Id(oauth2Id)
        .name(username)
        .email(email)
        .password(encoder.encode("password")
        .role(role)
        .provider(provider)
        .providerId(providerId).build();
    memberRepository.save(member);
}
else{
    member=findMember.get();
}
PrincipalDetails principalDetails = new Princip

```

```

        System.out.println("principalDetails = " + prin
        return principalDetails;
    }
}

```

- Member 엔티티와의 이벤트 담당 및 다음 단계로 넘어가기 위한 객체 구현 (PrincipalDetails)를 이용하여
- PrincipalDetails와 PrincipalDetailsService는 principalDetails객체를 만들기 위한 조각조각에 해당한다.
 - PrincipalDetails.java

```

package web.web1.Oauth.domain;

import lombok.Getter;
import org.springframework.security.core.GrantedAuthority;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.oauth2.core.user.OAuth2User;

import java.util.ArrayList;
import java.util.Collection;
import java.util.Map;
@Getter
public class PrincipalDetails implements OAuth2User {
    private Member member;
    private Map<String, Object> attributes;

    public PrincipalDetails(Member member) {
        this.member=member;
    }

    public PrincipalDetails(Member member, Map<String, Object> attributes) {
        this.member=member;
        this.attributes=attributes;
    }

    @Override

```

```

public Collection<? extends GrantedAuthority> ge
    Collection<GrantedAuthority> collect = new A
collect.add(new GrantedAuthority() {
    @Override
    public String getAuthority() {
        return member.getRole();
    }
});
return collect;
}

@Override
public String getPassword() {
    return member.getPassword();
}

@Override
public String getUsername() {
    return member.getName();
}

@Override
public boolean isAccountNonExpired() {
    return true;
}

@Override
public boolean isAccountNonLocked() {
    return true;
}

@Override
public boolean isCredentialsNonExpired() {
    return true;
}

@Override
public boolean isEnabled() {

```

```

        return true;
    }

    @Override
    public String getName() {
        return "name";
    }

    @Override
    public String toString() {
        return "PrincipalDetails{" +
            "member=" + member +
            ", attributes=" + attributes +
            '}';
    }
}

```

- PrincipalDetailsService.java (엔티티와의 연계구현)

```

package web.web1.0auth.domain;

import lombok.RequiredArgsConstructor;
import org.springframework.security.core.userdetails
import org.springframework.security.core.userdetails
import org.springframework.security.core.userdetails
import org.springframework.stereotype.Service;

import java.util.Optional;

// 시큐리티 설정에서 loginProcessingUrl("/login");
// login 요청이 오면 자동으로 UserDetails Service 타입으로
@Service
@RequiredArgsConstructor
public class PrincipalDetailsService implements User
    private final MemberRepository memberRepository;

```

```

@Override
public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {

    Optional<Member> member = memberRepository.findOne(username);
    if (member.isPresent()) {
        System.out.println("member = " + member.get());
        return new PrincipalDetails(member.get());
    }
    return null;
}
}

```

▼ 앞으로의 로직

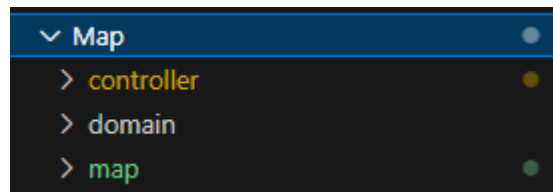
현재 인증코드를 받아오는것까진 완료.

이젠 인증코드를 토큰화 시켜야된다.

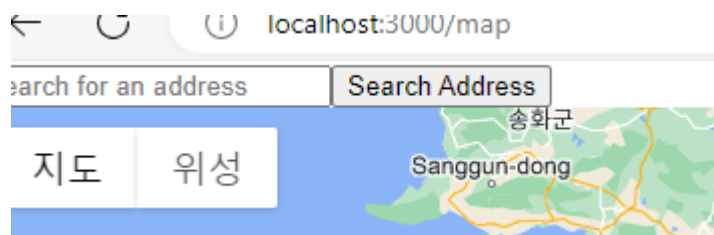
2. 도로명 주소검색 및 지도 마커 기능구현 (초기단계)

▼ 도로명 주소검색 기능구현

- 필요한 디렉토리 구성



- controller → 프론트엔드부터의 백엔드로의 요청 수행
 - getAddrApi (void)
 - /map/getaddress
 - 사용자가 프론트에서 도로명 주소를 검색하면, POST



- ```

로영 주수 받기 시작
e48117d8fWegDI0EPwDE0fJy0TEs0US3tc=
0
공포구 토정로
son
url?url=https://business.juso.go.kr/addrLink/addrLinkApi.do?currentPage=1&countPerPage=10&keyword=3EBA788B8E1873CAEAB5AC1E18A85A0ECAA0757E8BA130C&confKey=devU1T0FWEgDI0T0H0wDE0fJy0TEs0D
&resultType=json

```

- | 마포구 토정로                     | Search Address                     |       |                        |
|-----------------------------|------------------------------------|-------|------------------------|
| 도로명 주소                      | 지번 주소                              | 우편번호  | 버튼                     |
| 서울특별시 마포구 토정로 2 (합정동)       | 서울특별시 마포구 합정동 139-21 양화진공원주차장      | 04085 | <a href="#">Select</a> |
| 서울특별시 마포구 토정로 4 (합정동)       | 서울특별시 마포구 합정동 97-1 천주교서울대교구구무실교회관  | 04085 | <a href="#">Select</a> |
| 서울특별시 마포구 토정로 6 (합정동)       | 서울특별시 마포구 합정동 96-1 절두산순교기념박물관      | 04085 | <a href="#">Select</a> |
| 서울특별시 마포구 토정로 6-1 (합정동)     | 서울특별시 마포구 합정동 194-1 절두산 순교박물관 영상교육 | 04085 | <a href="#">Select</a> |
| 서울특별시 마포구 토정로 8-1 (합정동)     | 서울특별시 마포구 합정동 185-13               | 04085 | <a href="#">Select</a> |
| 서울특별시 마포구 토정로 8-7 (합정동)     | 서울특별시 마포구 합정동 185-11               | 04085 | <a href="#">Select</a> |
| 서울특별시 마포구 토정로 8-11 (합정동)    | 서울특별시 마포구 합정동 196-3                | 04085 | <a href="#">Select</a> |
| 서울특별시 마포구 토정로 8-19 (합정동)    | 서울특별시 마포구 합정동 94-3                 | 04085 | <a href="#">Select</a> |
| 서울특별시 마포구 토정로 8-41 (합정동)    | 서울특별시 마포구 합정동 89-2                 | 04085 | <a href="#">Select</a> |
| 서울특별시 마포구 토정로 9 (합정동, 두산빌라) | 서울특별시 마포구 합정동 376-24 두산빌라          | 04083 | <a href="#">Select</a> |

- /map/getgeocoding

- | 마포구 토정로                     | Search Address | 지번 주소                              | 우편번호  | 버튼     |
|-----------------------------|----------------|------------------------------------|-------|--------|
| 서울특별시 마포구 토정로 2 (합정동)       |                | 서울특별시 마포구 합정동 139-21 양화진공영주차장      | 04085 | Select |
| 서울특별시 마포구 토정로 4 (합정동)       |                | 서울특별시 마포구 합정동 97-1 천주교서울대교구구르실료회관  | 04085 | Select |
| 서울특별시 마포구 토정로 6 (합정동)       |                | 서울특별시 마포구 합정동 96-1 절두산순교기념박물관      | 04085 | Select |
| 서울특별시 마포구 토정로 6-1 (합정동)     |                | 서울특별시 마포구 합정동 194-1 절두산 순교박물관 영상교육 | 04085 | Select |
| 서울특별시 마포구 토정로 8-1 (합정동)     |                | 서울특별시 마포구 합정동 185-13               | 04085 | Select |
| 서울특별시 마포구 토정로 8-7 (합정동)     |                | 서울특별시 마포구 합정동 185-11               | 04085 | Select |
| 서울특별시 마포구 토정로 8-11 (합정동)    |                | 서울특별시 마포구 합정동 196-3                | 04085 | Select |
| 서울특별시 마포구 토정로 8-19 (합정동)    |                | 서울특별시 마포구 합정동 94-3                 | 04085 | Select |
| 서울특별시 마포구 토정로 8-41 (합정동)    |                | 서울특별시 마포구 합정동 89-2                 | 04085 | Select |
| 서울특별시 마포구 토정로 9 (합정동, 두산빌라) |                | 서울특별시 마포구 합정동 376-24 두산빌라          | 04083 | Select |

- 지물분석기 마포구 토종초 2 (잠정동)

- 구글맵이 해당 위치에 마커를 자동으로 찍어준다.

