

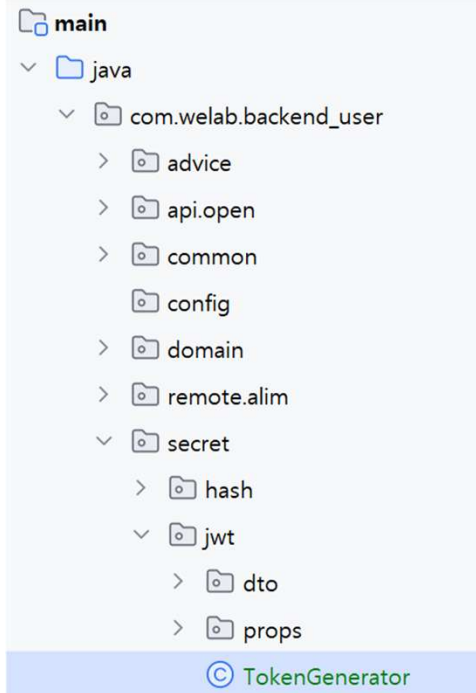
서비스 로직 작성

토큰 리프레시

AccessToken의 만료기간이 지난 경우,
RefreshToken을 통해 AccessToken 새로 발급

User 프로젝트 코드 작성 – TokenGenerator 메소드 추가

토큰 검증 메소드 추가



```
public String validateJwtToken(String refreshToken) {  
    String userId = null;  
    final Claims claims = this.verifyAndGetClaims(refreshToken);  
    if (claims == null) {  
        return null;  
    }  
  
    Date expirationDate = claims.getExpiration();  
    if (expirationDate == null || expirationDate.before(new Date())) {  
        return null;  
    }  
  
    userId = claims.get("userId", String.class);  
  
    String tokenType = claims.get("tokenType", String.class);  
    if (!"refresh".equals(tokenType)) {  
        return null;  
    }  
  
    return userId;  
}
```

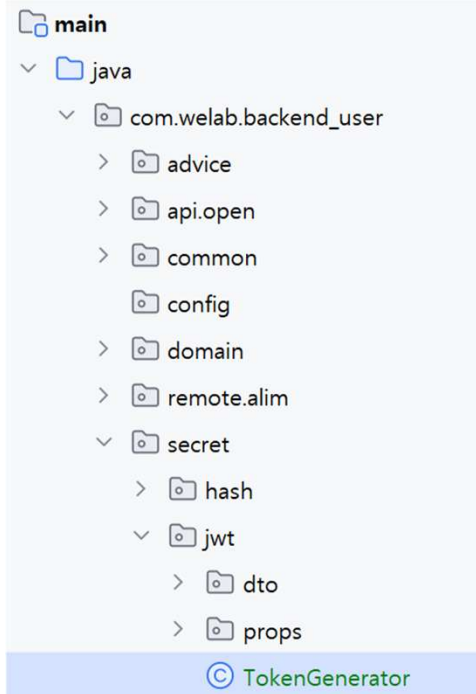
payload 가

가

가

[
tokenType null
NullPointerException

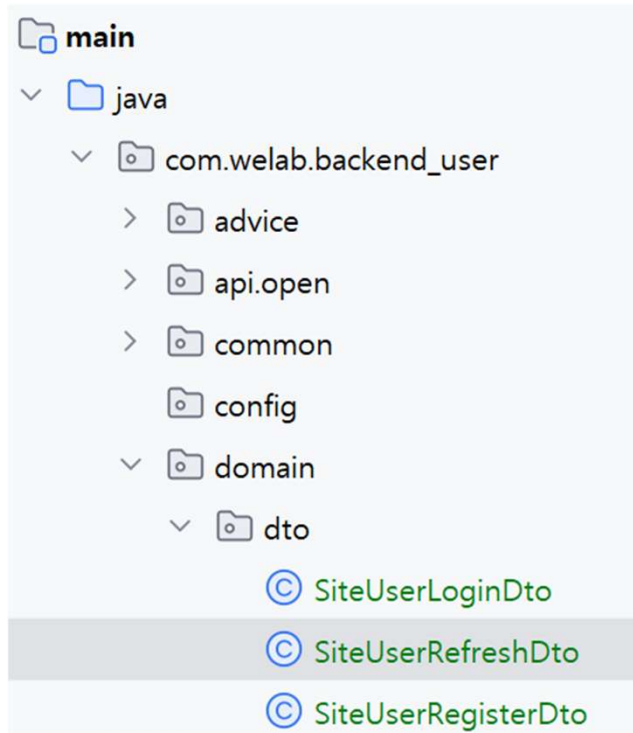
User 프로젝트 코드 작성 – TokenGenerator 메소드 추가 (계속)



토큰 검증 후 payload 가져오기

```
private Claims verifyAndGetClaims(String token) {  
    Claims claims;  
    try {  
        claims = Jwts.parser().verifyWith(getSecretKey()).build().parseSignedClaims(token).getPayload();  
    } catch (Exception e) {  
        claims = null;  
    }  
    return claims;  
}
```

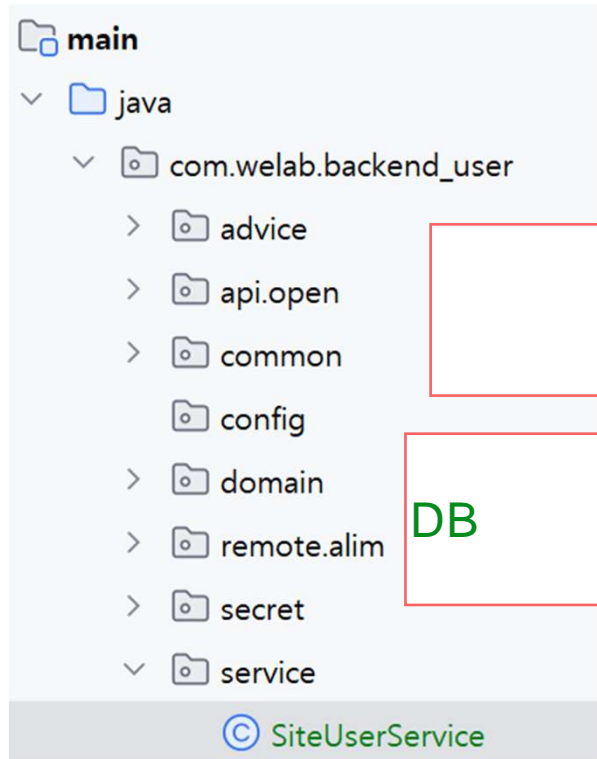
User 프로젝트 코드 작성 – SiteUserRefreshDto



```
@Getter
@Setter
public class SiteUserRefreshDto {
    @NotBlank(message = "리프레시 토큰을 입력하세요.")
    private String token;
}
```

token 안에서 userId를 추출 가능

User 프로젝트 코드 작성 – SiteUserService 기능 추가



DB

```
@Transactional(readonly = true)
public TokenDto.AccessToken refresh(SiteUserRefreshDto refreshDto) {
    String userId = tokenGenerator.validateJwtToken(refreshDto.getToken());
    if (userId == null) {
        throw new BadParameter("토큰이 유효하지 않습니다.");
    }

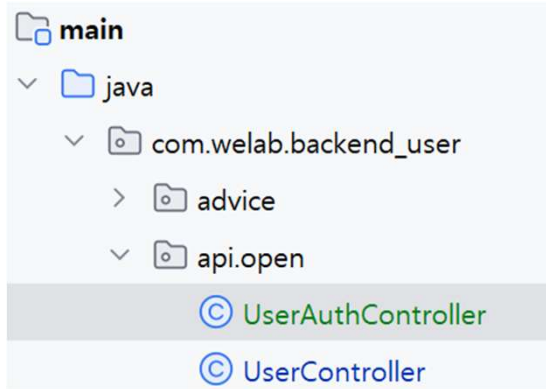
    SiteUser user = siteUserRepository.findById(userId);
    if (user == null) {
        throw new NotFound("사용자를 찾을 수 없습니다.");
    }

    return tokenGenerator.generateAccessToken(userId, "WEB");
}
```

DB
readOnly

가

User 프로젝트 코드 작성 – UserAuthController API 추가



```
@PostMapping(value = "/refresh")
public ApiResponseDto<TokenDto.AccessToken> refresh(@RequestBody @Valid SiteUserRefreshDto refreshDto) {
    TokenDto.AccessToken token = siteUserService.refresh(refreshDto);
    return ApiResponseDto.createOk(token);
}
```