

OAuth 소셜 로그인 2

구조

```
com/deeplify/tutorial/oauthlogin
├── OAuthLoginApplication.java
├── api
│   ├── controller
│   │   ├── auth
│   │   │   └── AuthController.java
│   │   └── user
│   │       └── UserController.java
│   ├── entity
│   │   ├── auth
│   │   │   └── AuthReqModel.java
│   │   └── user
│   │       ├── User.java
│   │       └── UserRefreshToken.java
│   ├── repository
│   │   ├── user
│   │   │   ├── UserRefreshTokenRepository.java
│   │   │   └── UserRepository.java
│   └── service
│       └── UserService.java
├── common
│   ├── ApiResponse.java
│   └── ApiResponseHeader.java
├── config
│   ├── properties
│   │   ├── AppProperties.java
│   │   └── CorsProperties.java
│   └── security
│       ├── JwtConfig.java
│       └── SecurityConfig.java
└── oauth
```

```

|   ├── entity
|   |   ├── ProviderType.java
|   |   ├── RoleType.java
|   |   └── UserPrincipal.java
|   ├── exception
|   |   ├── OAuthProviderMissMatchException.java
|   |   ├── RestAuthenticationEntryPoint.java
|   |   └── TokenValidFailedException.java
|   ├── filter
|   |   └── TokenAuthenticationFilter.java
|   ├── handler
|   |   ├── OAuth2AuthenticationFailureHandler.java
|   |   ├── OAuth2AuthenticationSuccessHandler.java
|   |   └── TokenAccessDeniedHandler.java
|   ├── info
|   |   ├── OAuth2UserInfo.java
|   |   ├── OAuth2UserInfoFactory.java
|   |   └── impl
|   |       ├── FacebookOAuth2UserInfo.java
|   |       ├── GoogleOAuth2UserInfo.java
|   |       ├── KakaoOAuth2UserInfo.java
|   |       └── NaverOAuth2UserInfo.java
|   ├── repository
|   |   └── OAuth2AuthorizationRequestBasedOnCookieRepository.java
|   ├── service
|   |   ├── CustomOAuth2UserService.java
|   |   └── CustomUserDetailsService.java
|   ├── token
|   |   ├── AuthToken.java
|   |   └── AuthTokenProvider.java
|   └── utils
|       ├── CookieUtil.java
|       └── HeaderUtil.java

```

각 클래스별 역할

api 디렉토리

Controller:

- `AuthController.java` : 로그인, 토큰 갱신 등 인증 관련 API 엔드포인트 처리
- `UserController.java` : 사용자 정보 조회, 수정 등 사용자 관련 API 엔드포인트 처리

Entity:

- `AuthReqModel.java` : 인증 요청 데이터를 담는 모델
- `User.java` : 사용자 정보를 저장하는 엔티티 (이름, 이메일, 프로필 등)
- `UserRefreshToken.java` : 리프레시 토큰 정보를 저장하는 엔티티

Repository:

- `UserRepository.java` : 사용자 정보 DB 접근 인터페이스
- `UserRefreshTokenRepository.java` : 리프레시 토큰 DB 접근 인터페이스

Service:

- `UserService.java` : 사용자 관련 비즈니스 로직 처리

common 디렉토리

- `ApiResponse.java` : API 응답의 공통 포맷 정의
- `ApiResponseHeader.java` : API 응답 헤더의 공통 포맷 정의

config 디렉토리

Properties:

- `AppProperties.java` : 애플리케이션 설정 값들 관리 (토큰 만료시간 등)
- `CorsProperties.java` : CORS 관련 설정 관리

Security:

- `JwtConfig.java` : JWT 관련 설정
- `SecurityConfig.java` : 스프링 시큐리티 설정 (보안 필터, 인증 핸들러 등)

oauth 디렉토리

Entity:

- `ProviderType.java` : 소셜 로그인 제공자 타입 열거형 (GOOGLE, FACEBOOK 등)
- `RoleType.java` : 사용자 권한 타입 열거형
- `UserPrincipal.java` : 인증된 사용자 정보를 담는 클래스

Exception:

- `OAuthProviderMissMatchException.java` : OAuth 제공자 불일치 예외
- `RestAuthenticationEntryPoint.java` : 인증되지 않은 요청 처리
- `TokenValidFailedException.java` : 토큰 검증 실패 예외

Filter:

- `TokenAuthenticationFilter.java` : 요청의 JWT 토큰을 검증하는 필터

Handler:

- `OAuth2AuthenticationSuccessHandler.java` : OAuth 인증 성공 처리
- `OAuth2AuthenticationFailureHandler.java` : OAuth 인증 실패 처리
- `TokenAccessDeniedHandler.java` : 접근 권한 없음 처리

Info:

- `OAuth2UserInfo.java` : OAuth 사용자 정보 추상 클래스
- `OAuth2UserInfoFactory.java` : 제공자별 OAuth 사용자 정보 객체 생성
- `impl` 패키지의 각 파일들: 각 소셜 로그인 제공자별 사용자 정보 처리 구현

Repository:

- `OAuth2AuthorizationRequestBasedOnCookieRepository.java` : OAuth 인증 요청 정보를 쿠키에 저장/관리

Service:

- `CustomOAuth2UserService.java` : OAuth 인증 처리 커스텀 서비스
- `CustomUserDetailsService.java` : 스프링 시큐리티의 `UserDetailsService` 구현

Token:

- `AuthToken.java` : JWT 토큰 클래스
- `AuthTokenProvider.java` : JWT 토큰 생성, 검증 등 처리

utils 디렉토리

- `CookieUtil.java` : 쿠키 생성, 조회, 삭제 등 유틸리티
- `HeaderUtil.java` : HTTP 헤더 처리 유틸리티