# 개발주석표준 가이드

## 📋 개요

이 문서는 CMS 프로젝트의 JavaDoc 주석 작성 표준을 정의합니다. 일관된 주석 스타일을 통해 코드의 가독성과 유지보수성을 향상시키는 것을 목표로 합니다.

## 🎯 주석 작성 원칙

### 1. \*\*기본 원칙\*\*

- \*\*명확성\*\*: 코드의 의도와 동작을 명확하게 설명

- \*\*일관성\*\*: 프로젝트 전체에서 동일한 스타일 적용

- \*\*완전성\*\*: 모든 public 메서드와 클래스에 주석 작성

- \*\*최신성\*\*: 코드 변경 시 주석도 함께 업데이트

### 2. \*\*주석 대상\*\*

- \*\*필수\*\*: public 클래스, 인터페이스, 메서드

- \*\*권장\*\*: protected 메서드, 중요한 필드

- \*\*선택\*\*: private 메서드 (복잡한 로직인 경우)

## 📝 JavaDoc 기본 문법

### 1. \*\*기본 구조\*\*

```java

/\*\*

\* 클래스나 메서드의 간단한 설명 (첫 번째 문장)

\*

\* <p>상세한 설명이 필요한 경우 여기에 작성합니다.</p>

\*

\* @param paramName 파라미터 설명

\* @return 반환값 설명

\* @throws ExceptionType 예외 상황 설명

\* @since 1.0

\* @author 작성자명

\* @see 관련클래스#메서드

\*/

```

### 2. \*\*주요 JavaDoc 태그\*\*

| 태그 | 설명 | 사용 위치 |

|------|------|-----------|

| `@param` | 메서드 파라미터 설명 | 메서드 |

| `@return` | 반환값 설명 | 메서드 |

| `@throws` | 예외 상황 설명 | 메서드 |

| `@since` | 도입 버전 | 클래스, 메서드 |

| `@author` | 작성자 | 클래스 |

| `@version` | 버전 정보 | 클래스 |

| `@see` | 관련 항목 참조 | 모든 곳 |

| `@apiNote` | API 사용 시 주의사항 | 메서드 |

| `@implNote` | 구현 관련 참고사항 | 메서드 |

## 🎨 HTML 태그 활용 가이드

### 1. \*\*HTML 태그 사용 이유\*\*

JavaDoc은 소스코드 주석을 파싱하여 \*\*HTML 형태의 API 문서\*\*를 자동 생성합니다. HTML 태그를 사용하면:

- \*\*가독성 향상\*\*: 구조화된 문서로 이해하기 쉬움

- \*\*자동 문서화\*\*: JavaDoc 도구가 예쁜 HTML 문서 생성

- \*\*IDE 지원\*\*: 개발 도구에서 리치 텍스트로 표시

- \*\*표준 준수\*\*: Oracle JavaDoc 스타일 가이드 준수

### 2. \*\*자주 사용되는 HTML 태그\*\*

#### \*\*텍스트 서식\*\*

```java

/\*\*

\* <p>단락을 구분할 때 사용합니다.</p>

\* <b>중요한 내용</b>을 강조할 때 사용합니다.

\* <i>이탤릭체</i>로 표시할 때 사용합니다.

\* <code>method()</code>와 같은 코드를 표시할 때 사용합니다.

\*/

```

#### \*\*목록 작성\*\*

```java

/\*\*

\* <p><b>주요 기능:</b></p>

\* <ul>

\* <li>첫 번째 기능</li>

\* <li>두 번째 기능</li>

\* <li>세 번째 기능</li>

\* </ul>

\*

\* <p><b>처리 과정:</b></p>

\* <ol>

\* <li>첫 번째 단계</li>

\* <li>두 번째 단계</li>

\* <li>세 번째 단계</li>

\* </ol>

\*/

```

#### \*\*코드 블록\*\*

```java

/\*\*

\* <p>사용 예시:</p>

\* <pre>

\* AuthController controller = new AuthController();

\* LoginRequest request = new LoginRequest("user", "password");

\* ResponseEntity&lt;LoginResponse&gt; response = controller.login(request);

\* </pre>

\*/

```

#### \*\*테이블\*\*

```java

/\*\*

\* <p><b>HTTP 상태 코드:</b></p>

\* <table>

\* <tr><th>상태 코드</th><th>설명</th></tr>

\* <tr><td>200</td><td>성공</td></tr>

\* <tr><td>400</td><td>잘못된 요청</td></tr>

\* <tr><td>401</td><td>인증 실패</td></tr>

\* </table>

\*/

```

### 3. \*\*HTML 태그 사용 규칙\*\*

- \*\*&lt;와 &gt;\*\*: 제네릭 타입 표현 시 `&lt;T&gt;` 사용

- \*\*줄바꿈\*\*: `<br>` 태그 사용 (가급적 `<p>` 태그 권장)

- \*\*링크\*\*: `{@link ClassName#methodName}` 사용

- \*\*인라인 코드\*\*: `{@code variableName}` 또는 `<code>` 사용

## 📋 클래스 주석 표준

### 1. \*\*클래스 주석 템플릿\*\*

```java

/\*\*

\* 클래스의 간단한 설명

\*

\* <p>클래스의 상세한 설명과 목적을 여기에 작성합니다.</p>

\*

\* <p><b>주요 기능:</b></p>

\* <ul>

\* <li>기능 1</li>

\* <li>기능 2</li>

\* <li>기능 3</li>

\* </ul>

\*

\* <p><b>사용 예시:</b></p>

\* <pre>

\* ClassName instance = new ClassName();

\* instance.someMethod();

\* </pre>

\*

\* <p><b>주의사항:</b></p>

\* <ul>

\* <li>주의사항 1</li>

\* <li>주의사항 2</li>

\* </ul>

\*

\* @author 작성자명

\* @version 1.0

\* @since 2024-01-01

\*

\* @see 관련클래스1

\* @see 관련클래스2

\*/

public class ClassName {

// ...

}

```

### 2. \*\*Controller 클래스 주석 예시\*\*

```java

/\*\*

\* 사용자 관리 API 컨트롤러

\*

\* <p>사용자 등록, 조회, 수정, 삭제 기능을 제공하는 REST API 컨트롤러입니다.</p>

\*

\* <p><b>주요 기능:</b></p>

\* <ul>

\* <li>사용자 등록 및 인증</li>

\* <li>사용자 정보 조회 및 수정</li>

\* <li>사용자 권한 관리</li>

\* </ul>

\*

\* <p><b>API 엔드포인트:</b></p>

\* <ul>

\* <li>POST /api/users - 사용자 등록</li>

\* <li>GET /api/users/{id} - 사용자 조회</li>

\* <li>PUT /api/users/{id} - 사용자 수정</li>

\* <li>DELETE /api/users/{id} - 사용자 삭제</li>

\* </ul>

\*

\* <p><b>보안 고려사항:</b></p>

\* <ul>

\* <li>모든 엔드포인트는 인증이 필요합니다</li>

\* <li>개인정보 처리 시 데이터 마스킹 적용</li>

\* <li>입력값 검증 및 XSS 방지</li>

\* </ul>

\*

\* @author cms-team

\* @version 1.0

\* @since 2024-01-01

\*

\* @see UserService

\* @see UserRepository

\* @see UserDTO

\*/

@RestController

@RequestMapping("/api/users")

public class UserController {

// ...

}

```

## 📋 메서드 주석 표준

### 1. \*\*메서드 주석 템플릿\*\*

```java

/\*\*

\* 메서드의 간단한 설명

\*

\* <p>메서드의 상세한 설명과 동작을 여기에 작성합니다.</p>

\*

\* <p><b>처리 과정:</b></p>

\* <ol>

\* <li>첫 번째 단계</li>

\* <li>두 번째 단계</li>

\* <li>세 번째 단계</li>

\* </ol>

\*

\* <p><b>주의사항:</b></p>

\* <ul>

\* <li>주의사항 1</li>

\* <li>주의사항 2</li>

\* </ul>

\*

\* @param param1 첫 번째 파라미터 설명

\* - 추가 설명이 필요한 경우

\* @param param2 두 번째 파라미터 설명

\*

\* @return 반환값 설명

\* - 성공 시: 설명

\* - 실패 시: 설명

\*

\* @throws ExceptionType1 예외 상황 1 설명

\* @throws ExceptionType2 예외 상황 2 설명

\*

\* @apiNote API 사용 시 주의사항

\*

\* @see 관련메서드1

\* @see 관련메서드2

\*

\* @since 1.0

\*/

public ReturnType methodName(Type param1, Type param2) {

// ...

}

```

### 2. \*\*API 메서드 주석 예시\*\*

```java

/\*\*

\* 사용자 로그인 처리

\*

\* <p>사용자 ID와 비밀번호를 검증하여 JWT 토큰을 생성합니다.</p>

\*

\* <p><b>처리 과정:</b></p>

\* <ol>

\* <li>입력값 검증 (@Valid 어노테이션)</li>

\* <li>사용자 인증 정보 확인</li>

\* <li>JWT 토큰 생성</li>

\* <li>사용자 세션 시작</li>

\* <li>로그인 메트릭 업데이트</li>

\* </ol>

\*

\* <p><b>보안 고려사항:</b></p>

\* <ul>

\* <li>비밀번호는 BCrypt로 암호화된 값과 비교</li>

\* <li>로그인 실패 시 상세 정보 노출 방지</li>

\* <li>로그인 시도 로그 기록</li>

\* </ul>

\*

\* @param request 로그인 요청 정보

\* - username: 사용자 ID (3-50자, 필수)

\* - password: 비밀번호 (6-100자, 필수)

\*

\* @return ResponseEntity&lt;LoginResponse&gt; 로그인 응답 정보

\* - 성공 시: 200 OK + JWT 토큰, 사용자 역할, 만료 시간

\* - 실패 시: 401 Unauthorized + 에러 메시지

\*

\* @throws InvalidCredentialsException 인증 정보가 올바르지 않은 경우

\* @throws RuntimeException 로그인 처리 중 시스템 오류 발생 시

\*

\* @apiNote 보안상 이유로 로그인 실패 시 구체적인 실패 사유를 반환하지 않습니다.

\*

\* @see AuthService#login(LoginRequest)

\* @see UserSessionService#startSession(String, String, java.time.Instant)

\*

\* @since 1.0

\*/

@PostMapping("/login")

public ResponseEntity<LoginResponse> login(@Valid @RequestBody LoginRequest request) {

// ...

}

```

## 📋 필드 주석 표준

### 1. \*\*필드 주석 템플릿\*\*

```java

/\*\*

\* 필드의 간단한 설명

\*

\* <p>필드의 상세한 설명과 용도를 여기에 작성합니다.</p>

\*

\* <p><b>주의사항:</b></p>

\* <ul>

\* <li>주의사항 1</li>

\* <li>주의사항 2</li>

\* </ul>

\*

\* @since 1.0

\*/

private final ServiceType serviceName;

```

### 2. \*\*의존성 주입 필드 예시\*\*

```java

/\*\*

\* 인증 서비스

\*

\* <p>사용자 로그인/로그아웃 처리 및 JWT 토큰 관리를 담당합니다.</p>

\*

\* <p><b>주요 기능:</b></p>

\* <ul>

\* <li>사용자 인증 정보 검증</li>

\* <li>JWT 토큰 생성 및 검증</li>

\* <li>로그인/로그아웃 처리</li>

\* </ul>

\*

\* @see AuthService

\* @since 1.0

\*/

private final AuthService authService;

```

## 📋 예외 클래스 주석 표준

```java

/\*\*

\* 사용자 인증 실패 예외

\*

\* <p>로그인 시 사용자 ID 또는 비밀번호가 올바르지 않을 때 발생하는 예외입니다.</p>

\*

\* <p><b>발생 상황:</b></p>

\* <ul>

\* <li>존재하지 않는 사용자 ID</li>

\* <li>잘못된 비밀번호</li>

\* <li>계정 잠금 상태</li>

\* </ul>

\*

\* <p><b>처리 방법:</b></p>

\* <ul>

\* <li>사용자에게 일반적인 오류 메시지 표시</li>

\* <li>보안 로그에 상세 정보 기록</li>

\* <li>브루트 포스 공격 방지 로직 실행</li>

\* </ul>

\*

\* @author cms-team

\* @version 1.0

\* @since 2024-01-01

\*

\* @see AuthService

\* @see SecurityException

\*/

public class InvalidCredentialsException extends RuntimeException {

// ...

}

```

## 📋 인터페이스 주석 표준

```java

/\*\*

\* 사용자 인증 서비스 인터페이스

\*

\* <p>사용자 로그인, 로그아웃, 토큰 관리 등 인증 관련 기능을 정의합니다.</p>

\*

\* <p><b>구현 클래스:</b></p>

\* <ul>

\* <li>{@link AuthServiceImpl} - 기본 구현체</li>

\* <li>{@link LdapAuthService} - LDAP 연동 구현체</li>

\* </ul>

\*

\* <p><b>주요 기능:</b></p>

\* <ul>

\* <li>사용자 인증 및 토큰 생성</li>

\* <li>로그아웃 및 토큰 무효화</li>

\* <li>토큰 유효성 검증</li>

\* </ul>

\*

\* @author cms-team

\* @version 1.0

\* @since 2024-01-01

\*

\* @see AuthServiceImpl

\* @see TokenProvider

\*/

public interface AuthService {

// ...

}

```

## 📋 Enum 주석 표준

```java

/\*\*

\* 사용자 역할 열거형

\*

\* <p>시스템 사용자의 권한 수준을 정의합니다.</p>

\*

\* <p><b>권한 계층:</b></p>

\* <ol>

\* <li>{@link #ADMIN} - 최고 관리자 권한</li>

\* <li>{@link #MANAGER} - 관리자 권한</li>

\* <li>{@link #USER} - 일반 사용자 권한</li>

\* </ol>

\*

\* @author cms-team

\* @version 1.0

\* @since 2024-01-01

\*/

public enum Role {

/\*\*

\* 시스템 관리자

\*

\* <p>모든 시스템 기능에 대한 접근 권한을 가집니다.</p>

\*

\* <p><b>주요 권한:</b></p>

\* <ul>

\* <li>사용자 관리</li>

\* <li>시스템 설정</li>

\* <li>모든 데이터 접근</li>

\* </ul>

\*/

ADMIN,

/\*\*

\* 관리자

\*

\* <p>제한된 관리 기능에 대한 접근 권한을 가집니다.</p>

\*/

MANAGER,

/\*\*

\* 일반 사용자

\*

\* <p>기본적인 시스템 기능에 대한 접근 권한을 가집니다.</p>

\*/

USER

}

```

## 📋 주석 작성 체크리스트

### ✅ \*\*클래스 주석 체크리스트\*\*

- [ ] 클래스의 목적과 역할 명시

- [ ] 주요 기능 목록 작성

- [ ] 사용 예시 코드 포함

- [ ] 주의사항 및 제약사항 명시

- [ ] @author, @version, @since 태그 작성

- [ ] 관련 클래스 @see 태그 추가

### ✅ \*\*메서드 주석 체크리스트\*\*

- [ ] 메서드의 목적과 동작 설명

- [ ] 처리 과정 단계별 설명

- [ ] 모든 @param 태그 작성

- [ ] @return 태그 작성 (void 메서드 제외)

- [ ] 가능한 예외 @throws 태그 작성

- [ ] 보안 관련 주의사항 명시

- [ ] 관련 메서드 @see 태그 추가

### ✅ \*\*HTML 태그 체크리스트\*\*

- [ ] 목록은 `<ul>`, `<ol>`, `<li>` 태그 사용

- [ ] 강조는 `<b>` 태그 사용

- [ ] 단락 구분은 `<p>` 태그 사용

- [ ] 코드는 `<code>` 또는 `<pre>` 태그 사용

- [ ] 제네릭 타입은 `&lt;`, `&gt;` 사용

## 📋 도구 및 설정

### 1. \*\*JavaDoc 생성\*\*

```bash

# Gradle 프로젝트

./gradlew javadoc

# Maven 프로젝트

mvn javadoc:javadoc

# 직접 실행

javadoc -d docs -cp classpath src/\*\*/\*.java

```

### 2. \*\*IDE 설정\*\*

- \*\*IntelliJ IDEA\*\*: Settings > Editor > Code Style > Java > JavaDoc

- \*\*Eclipse\*\*: Window > Preferences > Java > Code Style > Code Templates

- \*\*VS Code\*\*: Java Extension Pack + JavaDoc 플러그인

### 3. \*\*정적 분석 도구\*\*

- \*\*Checkstyle\*\*: JavaDoc 누락 검사

- \*\*SpotBugs\*\*: 주석 품질 검사

- \*\*SonarQube\*\*: 문서화 품질 메트릭

## 📋 참고 자료

- [Oracle JavaDoc 가이드](https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html)

- [Google Java Style Guide](https://google.github.io/styleguide/javaguide.html)

- [Spring Framework 주석 스타일](https://github.com/spring-projects/spring-framework/wiki/Code-Style)

---