

# 백엔드 개발자 과제 안내

안녕하세요 🙌

본 과제는 **KRAFTON Intra Platform Team** 신입 백엔드 개발자 채용 전형의 일환으로 진행됩니다.

과제를 통해 지원자의 코드 구조화, 테스트 작성 역량, 유지보수성, API 설계 감각, 그리고 **AI 활용 역량**을 평가합니다.

## 과제 개요

주제: 개인이 자주 방문하는 웹사이트를 관리할 수 있는 **북마크 관리 REST API** 서버를 개발하세요.

언어: Java

프레임워크: Spring Boot (권장)

제출 기한: 과제 수령 후 **3일 이내**

지원자는 자유롭게 설계하되, 아래의 **필수 요구사항**을 반드시 충족해야 합니다.

## 필수 요구사항

### 1) 주요 기능

북마크(Bookmark)를 관리하는 REST API를 구현합니다.

| 기능        | 설명              | 메서드    | 엔드포인트 예시        |
|-----------|-----------------|--------|-----------------|
| 북마크 등록    | 즐거찾기 사이트 추가     | POST   | /bookmarks      |
| 북마크 목록 조회 | 저장된 모든 북마크 조회   | GET    | /bookmarks      |
| 북마크 상세 조회 | 특정 북마크 상세 정보 조회 | GET    | /bookmarks/{id} |
| 북마크 수정    | 제목, URL 등 정보 수정 | PUT    | /bookmarks/{id} |
| 북마크 삭제    | 북마크 삭제          | DELETE | /bookmarks/{id} |

예시 필드: title , url , memo , createdAt , updatedAt

## 2) 테스트 코드

- JUnit + Mockito 사용 권장
- Service/Repository 레벨 중심 단위 테스트 작성
- 필요 시 MockMvc 또는 @SpringBootTest를 활용한 통합 테스트 일부 포함

## 3) API 명세

- Swagger(OpenAPI) 또는 Markdown( `api-docs.md` ) 형태로 작성
- 각 엔드포인트별 요청/응답 구조 및 상태 코드 포함

## 4) 실행 편의성

- 별도 설치 없이 실행 가능해야 합니다. (예: H2/in-memory DB)
- 아래 명령 중 하나로 실행 가능해야 합니다.

- Gradle: `./gradlew bootRun`
- Maven: `./mvnw spring-boot:run`
- 또는 빌드 후 실행: `java -jar build/libs/<jar파일명>.jar`

- README.md 에 다음 내용 포함:

- 프로젝트 구조 및 패키지 구성
- 빌드 / 실행 / 테스트 방법
- API 명세 확인 방법
- 주요 설계 이유 (3가지 정도)

예: "왜 H2 DB를 선택했는지", "왜 DTO를 따로 만들었는지", "예외 처리를 어떻게 통합했는지" 등

- 개선할 점 또는 아쉬운 점 (3가지 정도)

예: "시간이 더 있었다면 추가하고 싶었던 기능", "테스트 커버리지가 낮았던 이유" 등

---



## 선택(가산점) 요구사항

아래 항목은 선택 사항이며, 구현 시 가산점이 있습니다.

| 항목       | 설명                                  |
|----------|-------------------------------------|
| 태그 기능    | 북마크에 여러 태그를 추가하고, 태그별로 조회 가능        |
| 검색 기능    | 제목 또는 URL 키워드로 검색                   |
| 페이지네이션   | 목록 조회 시 페이지 단위 응답                   |
| 전역 예외 처리 | @ControllerAdvice 를 활용한 표준 예외 응답 구조 |
| 캐싱       | 자주 조회되는 데이터 캐싱 (예: Spring Cache)    |
| CI       | GitHub Actions를 활용한 자동 빌드/테스트 실행    |

### 선택(가산점): 인증 (Auth-Lite)

회원가입 / 로그인 / 로그아웃 기능을 추가하면 가산점이 있습니다.

다만, 이 기능은 예시 수준으로만 구현해도 충분합니다.

아래는 참고용 예시 엔드포인트입니다. 반드시 이 형태로 구현할 필요는 없습니다.

| 기능   | 설명                           | 예시 메서드 | 예시 엔드포인트     |
|------|------------------------------|--------|--------------|
| 회원가입 | 이메일과 비밀번호로 회원 생성             | POST   | /auth/signup |
| 로그인  | 로그인 성공 시 토큰 발급 (또는 세션 기반 처리) | POST   | /auth/login  |
| 로그아웃 | 토큰 무효화 또는 로그아웃 처리            | POST   | /auth/logout |

- 토큰 또는 세션 방식은 자유롭게 선택 가능합니다.
- 본인 계정의 북마크만 조회/수정/삭제 가능하도록 구현하면 좋습니다.
- Refresh 토큰, 이메일 인증 등은 불필요합니다.
- 단순한 인증 로직이라도 명확하게 구조화하고 예외 처리가 되어 있으면 충분히 가산점을 받을 수 있습니다.

# AI 활용 가이드라인 (가산점)

---

AI를 적극적으로 활용해 빠르고 효율적인 개발을 하는 것을 장려합니다.  
단, 이해와 투명성을 전제로 합니다.

## 1) 허용 범위

설계 아이디어, 테스트 케이스, 문서 초안, 반복 코드 생성 등에서 AI를 활용할 수 있습니다.

## 2) 요구 사항

프로젝트 루트에 `ai-notes.md` 파일을 추가하고 아래 항목을 간단히 작성하세요:

- 사용한 AI/도구 (예: ChatGPT, GitHub Copilot 등)
- 주요 프롬프트 또는 활용 방식 (핵심 3~5개)
- 생성 코드가 반영된 파일 또는 영역 (대략적인 범위)
- AI 생성물에 대해 직접 수정·검증한 부분 요약

## 3) 이해 검증

면접 시 본인이 제출한 모든 코드에 대해 직접 설명할 수 있어야 합니다.  
"AI가 만들어서 모른다"는 답변은 감점 사유가 됩니다.

## 4) 주의

- 비공개 소스나 사내 코드 업로드 금지
  - 라이선스 불명확한 코드 복사 금지
-

## 제출 방법

---

1. GitHub Repository에 업로드 후 링크 제출
  2. 반드시 아래 파일 포함
    - README.md
    - api-docs.md (또는 Swagger 명세)
    - ai-notes.md
  3. 과제 수령 후 **3일 이내 제출**
  4. 과제 제출은 아래 담당자 중 한 명에게 이메일로 보내주세요.
    - 홍성철 (schong@krafton.com)
    - 윤성현 (sunghyun.yoon@krafton.com)
- 

## 평가 관점

---

- 프로젝트 구조 및 코드 일관성
  - 가독성과 유지보수성
  - 테스트 코드의 품질
  - API 명세의 정확성과 실행 편의성
  - 선택 기능 구현 완성도
  - AI 활용의 투명성 및 자기 이해도
  - (선택 시) 인증 기능의 구조적 완성도
- 

## 문의

---

과제 진행 중 문의사항이 있는 경우 아래로 연락해 주세요.

- 윤성현 (sunghyun.yoon@krafton.com)
- 

## 참고

---

- 구현의 양보다 품질과 구조적 사고력을 중시합니다.
- 제출 후 코드 리뷰 인터뷰가 진행될 수 있습니다.