

# 기능 정의서

⓪ 도메인	⓪ 우선순위	≡ 메소드 번호	⓪ Http Method	≡ 엔드포인트	≡ 메서드명	Aa 기능명	≡ 상세 설명	≡ SQL	≡ Parameter
AI	상	A-2	POST		processCustomAISummary(String contents)	자체 AI 요약	크롤링한 데이터를 String으로 받아 자체 AI를 활용하여 요약  AI 응답을 가지고 U-1 (이미지 저장), U-6 (String을 JSON 형식으로 직렬화) 호출  호출 실패 시 Exception 발생 30초 이상 응답이 없으면 Exception 발생		String conter
AI	상	A-1	POST		processLilysAISummary(String url)	Lilys AI 요약	Lilys AI API를 호출 후 응답을 가지고 U-1 (이미지 저장), U-6 (String을 JSON 형식으로 직렬화) 호출  호출 실패 시 Exception 발생 30초 이상 응답이 없으면 Exception 발생		String URL
컨텐츠	상	C-9	GET	/api/contents/search	searchContents(String userId, String keyword)	컨텐츠 검색	userId와 keyword로Elastic Search에서 _search es에서 search한 cardId 리스트와 userId를 cards 테이블에서 SELECT		String userId String keywo
컨텐츠	상	C-5	POST		selectTargetBoard(String userId, String content)	컨텐츠 타겟 보드 선정	content를 가지고 U-3 (핵심 키워드 추출) 호출 userId로 cards 테이블에서 SELECT (미분류 보드 제외)  위의 두 응답을 가지고 벡터 값이 유사한 card가 존재하면 해당 card의 position 반환  존재하지 않으면 0반환 (0 = 미분류 보드)		String userId String conter
컨텐츠	하	C-8	POST	/api/contents/{cardId}/report	reportContent(String userId, String cardId)	AI 요약 신고	userId와 cardId로 cards 테이블에서 SELECT 변경 권한 및 카드 유무 확인 필요 값이 존재하지 않으면 Exception 발생  신고 card를 가져와서 해당 URL을 가지고 A-1 (Lilys AI 요약) 호출 A-1 (Lilys AI 요약) 호출 실패 시 영상(0)의 경우 U-7 (STT), 이외의 경우(1, 2) U-4 (크롤링) 호출 후 content 값 UPDATE  UPDATE 실패 시 Exception 발생  reports 테이블에 신고 받은 card의 이전 내용 INSERT 해당 INSERT는 트랜잭션을 분리하여 rollback 시 UPDATE에 영향을 주지 않도록 작성  성공 후 해당 card 반환	SELECT * FROM cards c JOIN boards b ON c.board_id = b.board_id WHERE c.card_id = cardId AND b.user_id = userId )  SELECT * FROM cards WHERE card_id = cardId  UPDATE cards SET content = contents WHERE card_id = cardId  INSERT INTO reports (card_id, content, original_url) VALUES (cardId, beforeContent, URL)	String userId String cardId
컨텐츠	상	C-4	PUT	/api/contents/{cardId}	updateContent(String userId, String cardId, String content)	컨텐츠 수정	userId와 cardId로 cards 테이블에서 SELECT 수정 권한 및 카드 유무 확인 필요 값이 존재하지 않으면 Exception 발생  UPDATE 실패 시 Exception 발생 성공 여부 응답을 true 로 반환	SELECT * FROM cards c JOIN boards b ON c.board_id = b.board_id WHERE c.card_id = cardId AND b.user_id = userId )  UPDATE cards SET content = content WHERE card_id = cardId	String userId String cardId String conter

㉠ 도메인	㉠ 우선순위	≡ 매소드 번호	㉠ Http Method	≡ 엔드포인트	≡ 메서드명	Aa 기능명	≡ 상세 설명	≡ SQL	≡ Parameter
컨텐츠	중	C-7	POST	/api/contents/{cardId}/copy	copyContent(String userId, String cardId, String boardId)	컨텐츠 복사	<p>userId와 URL로 cards 테이블에서 SELECT 값이 존재하면 Exception 발생 (User별 해당 URL 중복 확인)</p> <p>userId와 boardId로 boards 테이블에서 SELECT 권한 및 보드 유무 확인 필요 값이 존재하지 않으면 Exception 발생</p> <p>cardId로 cards 테이블에서 SELECT 값이 존재하지 않으면 Exception 발생</p> <p>boardId로 cards 테이블에 Card INSERT</p> <p>INSERT 실패 시 Exception 발생 성공 여부 응답을 true 로 반환</p>	<pre>SELECT EXISTS ( SELECT 1 FROM cards c JOIN boards b ON c.board_id = b.board_id WHERE c.original_url = URL AND b.user_id = userId )  SELECT EXISTS ( SELECT 1 FROM boards WHERE board_id = boardId AND user_id = userId )  SELECT * FROM cards WHERE card_id = cardId  INSERT INTO cards (board_id, content, type, original_url) VALUES (boardId, JSON, Integer, URL)</pre>	String userId String cardId String boardId
컨텐츠	상	C-3	DELETE	/api/contents/{cardId}	deleteContent(String userId, String cardId)	컨텐츠 삭제	<p>Soft Delete 방식 사용</p> <p>userId와 cardId로 cards 테이블에서 SELECT 삭제 권한 및 카드 유무 확인 필요 값이 존재하지 않으면 Exception 발생</p> <p>cardId에 해당하는 card에 delete_date 를 NOW() 로 갱신</p> <p>UPDATE 실패 시 Exception 발생 성공 여부 응답을 true 로 반환</p>	<pre>SELECT EXISTS ( SELECT 1 FROM cards c JOIN boards b ON c.board_id = b.board_id WHERE c.card_id = cardId AND b.user_id = userId )  UPDATE cards SET delete_date = NOW() WHERE card_id = cardId</pre>	String userId String cardId
컨텐츠	중	C-6	PUT	/api/contents/{cardId}/board	changeBoard(String userId, String cardId, String boardId)	소속 보드 변경	<p>userId와 cardId로 cards 테이블에서 SELECT 변경 권한 및 카드 유무 확인 필요 값이 존재하지 않으면 Exception 발생</p> <p>userId와 boardId로 boards 테이블에서 SELECT 권한 및 보드 유무 확인 필요 값이 존재하지 않으면 Exception 발생</p> <p>UPDATE 실패 시 Exception 발생 성공 여부 응답을 true 로 반환</p>	<pre>SELECT EXISTS ( SELECT 1 FROM cards c JOIN boards b ON c.board_id = b.board_id WHERE c.card_id = cardId AND b.user_id = userId )  SELECT EXISTS ( SELECT 1 FROM boards WHERE board_id = boardId AND user_id = userId )  UPDATE cards SET board_id=</pre>	String userId String cardId String boardId

⓪ 도메인	⓪ 우 선순 위	≡ 매 소 드 번 호	⓪ Http Method	≡ 엔드포인트	≡ 메서드명	Aa 기 능명	≡ 상세 설명	≡ SQL	≡ Paramete
								boardId WHERE card_id = cardId	
컨텐츠	상	C-2	POST	/api/contents	addContent(String userId, String url)	컨텐츠 추가	<p>userId와 URL로 cards 테이블에서 SELECT 값이 존재하면 Exception 발생 (User별 해당 URL 중복 확인)</p> <p>U-5 (URL별 콘텐츠 종류 분석) 호출 후 type 값 선정</p> <p>URL 요청으로 A-1 (Lilys AI 요약) 호출 A-1 (Lilys AI 요약) 호출 실패 시 영상(0)의 경우 U-7 (STT), 이외의 경우(1, 2) U-4 (크롤링) 호출 후 content 값 선정</p> <p>C-5 (컨텐츠 타겟 보드 선정) 호출 후 position으로 boards 테이블에 board_id를 SELECT</p> <p>card가 들어갈 board의 status 변경</p> <p>삽입 보드(board_id)와 내용(JSON), 콘텐츠 종류(0: YouTube, 1: Blog, 2: News), URL INSERT</p> <p>INSERT 실패 시 Exception 발생 성공 여부 응답을 true 반환</p>	<p>SELECT EXISTS ( SELECT 1 FROM cards c JOIN boards b ON c.board_id = b.board_id WHERE c.originalUrl = URL AND b.user_id = userId )</p> <p>SELECT board_id FROM boards WHERE user_id = userId AND position = position</p> <p>UPDATE boards SET status = true WHERE board_id= boardId</p> <p>INSERT INTO cards (board_id, content, type, originalUrl) VALUES (String, JSON, Integer, URL)</p>	String userId String URL
컨텐츠	상	C-1	GET	/api/contents	getContentList(String userId, String boardId, String searchCondition)	컨텐츠 리스트	<p>userId와 boardId, 그리고searchCondition으로 boards 테이블에서 SELECT 권한 및 보드 유무 확인 필요 값이 존재하지 않으면 Exception 발생</p> <p>확인한 boardId의 board status 변경</p> <p>boardId로 card 테이블에서 모든 요소를 SELECT 후 List로 반환</p> <p>- searchCondition 컨텐츠 종류, 정렬방식</p>	<p>SELECT EXISTS ( SELECT 1 FROM boards WHERE board_id = boardId AND user_id = userId )</p> <p>UPDATE boards SET status = false WHERE board_id= boardId</p> <p>SELECT * FROM cards WHERE board_id = boardId AND 옵션들 ORDER BY 정렬 방식</p>	String userId String boardId
보드	하	B-5	PUT	/api/boards/{boardId}/name	updateBoardName(String userId, String boardId, String boardName)	보드명 변경	<p>userId와 boardId로 boards 테이블에서 SELECT 수정 권한 및 보드 유무 확인 필요 값이 존재하지 않으면 Exception 발생</p> <p>userId와 boardId가 일치하는 board의 board_name UPDATE</p> <p>UPDATE 실패 시 Exception 발생 성공 여부 응답을 true 로 반환</p>	<p>SELECT EXISTS ( SELECT 1 FROM boards WHERE board_id = boardId AND user_id = userId )</p> <p>UPDATE boards SET board_name = boardName WHERE</p>	String userId String boardId String boardName

⓪ 도메인	⓪ 우선순위	≡ 매소드 번호	⓪ Http Method	≡ 엔드포인트	≡ 메서드명	Aa 기능명	≡ 상세 설명	≡ SQL	≡ Parameter
								<pre> user_id = userid AND board_id = boardId </pre>	
보드	상	B-3	DELETE	/api/boards/{boardId}	deleteBoard(String userId, Integer boardId, Integer targetPosition)	보드 제거	<p>userId와 boardId로 boards 테이블에서 SELECT 삭제 권한 및 보드 유무 확인 필요 값이 존재하지 않으면 Exception 발생</p> <p>값이 존재하면 해당 보드 DELETE 진행 DELETE 실패 시 Exception 발생</p> <p>targetPosition 이후의 모든 보드의 순서 -1 UPDATE 실패 시 Exception 발생 성공 여부 응답을 true 로 반환</p>	<pre> SELECT EXISTS ( SELECT 1 FROM boards WHERE board_id = boardId AND user_id = userid )  DELETE FROM boards WHERE board_id = boardId AND user_id = userid  UPDATE boards SET position = position - 1 WHERE user_id = userId AND position &gt; targetPosition </pre>	String userId Integer boardId Integer targetPosition
보드	상	B-2	POST	/api/boards	createBoard(String userId, String boardName)	보드 생성	<p>userId로 boards 테이블에서 총 board의 count 확인</p> <p>userId와 count로 boards 테이블에 INSERT (board position의 index는 0부터 시작)</p> <p>INSERT 실패 시 Exception 발생 성공 여부 응답을 true 로 반환</p>	<pre> SELECT COUNT(*) FROM boards WHERE user_id = userId  INSERT INTO boards (user_id, board_name, position) VALUES (userId, boardName, count) </pre>	String userId String boardName
보드	중	B-4	PUT	/api/boards/{boardId}/position	changeBoardPosition(String userId, String boardId, Integer sourcePosition, Integer targetPosition)	보드 순서 변경	<p>userId와 boardId로 boards 테이블에서 SELECT 편집 권한 및 보드 유무 확인 필요 값이 존재하지 않으면 Exception 발생</p> <p>sourcePosition &lt; targetPosition 인 경우와 sourcePosition &gt; targetPosition 인 경우에 대해 다른 예외도 적용</p> <p>위치가 변경되는 범위 내의 모든 board의 position UPDATE</p> <p>이동한 board의 position을 targetPosition으로 UPDATE</p> <p>UPDATE 실패 시 Exception 발생 성공 여부 응답을 true 로 반환</p>	<pre> SELECT EXISTS ( SELECT 1 FROM boards WHERE board_id = boardId AND user_id = userId )  - if sourcePosition &lt; targetPosition  UPDATE boards SET position = position - 1 WHERE user_id = userId AND position &gt; sourcePosition AND position &lt;= targetPosition  - else if sourcePosition &gt; targetPosition  UPDATE boards SET position = position + 1 WHERE </pre>	String userId String boardId Integer sourcePosition Integer targetPosition

⊙ 도메인	⊙ 우선순위	≡ 매소드 번호	⊙ Http Method	≡ 엔드포인트	≡ 메서드명	Aa 기능명	≡ 상세 설명	≡ SQL	≡ Parameter
								<pre> user_id = userid AND position &gt;= targetPosition AND position &lt; sourcePosition  UPDATE boards SET position = targetPosition WHERE board_id = boardid </pre>	
보드	상	B-1	GET	/api/boards	getBoardList(String userId)	보드 리스트	userId로 boards 테이블에서 모든 요소를 SELECT 후 List로 반환	<pre> SELECT * FROM boards WHERE user_id = userid </pre>	String userId
유틸	상	U-9	POST		hashPassword(String password)	해싱	org.apache.commons.codec.digest.DigestUtils를 import하여 구현 입력으로 받은 password를 hashing 하여 return		String passw
유틸	상	U-8	POST		saveImages(List<MultipartFile> fileList)	이미지 다중 저장	S3에 이미지 파일을 모두 저장 저장 후 생성된 URL을 입력 순서와 동일하게 List로 반환		List<MultipartfileList
유틸	상	U-6	POST		serializeToJson(String content)	String을 JSON 형식으로 직렬화	원시 String content를 JSON 형식으로 직렬화  JSON 형식 예시 (String으로 구현) { "title": "제목", "content": [ { "type": "paragraph_title", "text": "첫 번째 문단 제목" }, { "type": "paragraph", "text": "첫 번째 문단 내용" }, { "type": "image", "url": "이미지 URL", }, { "type": "paragraph_title", "text": "두 번째 문단 제목" } ] }		String conter
유틸	상	U-5	POST		analyzeContentType(String url)	URL 별 콘텐츠 종류 분석	URL 분석 후 콘텐츠 종류 반환  0: YouTube, 1: Blog, 2: News, 3: Image, 4: Shorts, 5: 미분류		String URL
유틸	상	U-3	POST		extractKeywords(String contents)	핵심 키워드 추출	pynini(유사 api 가능)으로 핵심 키워드 뽑기 핵심 키워드 List 반환		String conter
유틸	중	U-10	POST		refreshToken	JWT 토큰			
유틸	상	U-2	POST		generateUUID()	UUID 생성	다양한 PK에 활용될 UUID를 반환하도록 구현  java.util.UUID를 import하여 구현 UUID 객체를 발급 받고, toString() 메소드를 이용하여 String으로 변환 후 반환 UUID는 하이픈을 제외한 길이가 32자로 고정		
유틸	상	U-1	POST		saveImage(MultipartFile file)	이미지 저장	S3에 이미지 파일을 저장 저장 후 생성된 URL을 String으로 반환		MultipartFile
유틸	하	U-7	POST		convertSpeechToText(String url)	STT	Fast Wisper API를 사용하여 구현  크롤링한 데이터를 String으로 파싱 후 자체 AI 요약 메소드 호출		String URL
유틸	상	U-4	POST		crawlData(String url)	크롤링	실때 시 Exception 발생  org.jsoup.Jsoup를 import하여 구현  크롤링한 데이터를 String으로 파싱 후 A-2 (자체 AI 요약) 호출 후 반환  실때 시 Exception 발생		String URL

⊙ 도메인	⊙ 우선순위	≡ 매소드 번호	⊙ Http Method	≡ 엔드포인트	≡ 메서드명	Aa 기능명	≡ 상세 설명	≡ SQL	≡ Parameter
유저	하	H-5	GET	/api/user/logout	logout	로그아웃			
유저	중	H-2	PUT	/api/user/{userId}	updateUserProfile	개인 정보 수정	email과 password로 users 테이블에서 SELECT 유저 정보 확인 필요 값이 존재하지 않으면 Exception 발생  닉네임, 비밀번호, 프로필 이미지 변경	SELECT EXISTS ( SELECT 1 FROM users WHERE email = email AND password = hashPassword )	String email String password User user
유저	상	H-6	GET	/api/user/{userId}	getUser	유저 정보	요청한 사용자의 프로필 정보 주기 (프로필 이미지, 닉네임, 팔로워 수, 팔로잉 수)		
유저	하	H-4	DELETE	/api/user/{userId}	deleteUser	회원탈퇴	Soft Delete 방식 사용  email과 password로 users 테이블에서 SELECT 유저 정보 확인 필요 값이 존재하지 않으면 Exception 발생  email과 password로 users 테이블에서 deleted_date UPDATE UPDATE 실패 시 Exception 발생 성공 여부 응답을 true 로 반환	SELECT EXISTS ( SELECT 1 FROM users WHERE email = email AND password = hashPassword )  UPDATE users SET deleted_date = NOW() WHERE user_id = userId AND password = hashPassword	String email String password String verificationCode
유저	중		POST	/api/user/socialLogin	socialLogin	소셜 로그인	소셜(카카오, 구글, 네이버)을 활용해 서비스에 사용자 인증을 받는 기능 DB에 존재하지 않는 정보의 경우, DB에 정보 저장 후 로그인		
유저	하	H-3	POST	/api/user/signup	signup	회원가입	email로 users 테이블에서 SELECT 이메일 중복 확인 필요 값이 존재하면 Exception 발생  U-9 (해싱) 으로 비밀번호 해싱 처리  users 테이블에 새로운 user 정보 INSERT INSERT 실패 시 Exception 발생  성공 여부 응답을 true 로 반환	SELECT EXISTS ( SELECT 1 FROM users WHERE email = email )  INSERT INTO users (email, password) VALUES (email, hashPassword)	String email String password
유저	하	H-1	POST	/api/user/login	login	로그인	U-9 (해싱) 으로 비밀번호 해싱 처리  email과 password로 users 테이블에서 SELECT 유저 정보 확인 필요 값이 존재하지 않으면 Exception 발생  값이 존재하면 U-10 (refreshToken) 호출 후 토큰 발급	SELECT EXISTS ( SELECT 1 FROM users WHERE email = email AND password = hashPassword )	String email String password
폐기						통계 보기			
폐기						알림			
폐기	중	B-7				팀원 초대			
폐기	중					팔로잉 리스트	해당 사용자가 팔로잉한 유저 목록		
폐기	중					팔로워 리스트	해당 사용자를 팔로잉한 유저 목록		
폐기	중					팔로우/언팔로우	다른 사용자에 대한 팔로잉 상태 변환 (팔로잉 ↔ 언팔로잉)		
폐기	중	B-6				즐거찾 기록/취소	상대방의 보드인 경우, 해당 보드 즐겨찾기 등록 및 취소		
						제목 없음			

㉠ 도메인	㉠ 우선순위	≡ 매소드 번호	㉠ Http Method	≡ 엔드포인트	≡ 메서드명	Aa 기능명	≡ 상세 설명	≡ SQL	≡ Paramete
						제목 없음			
						제목 없음			