

기능 정의서

☉ 도메인	☉ 우선순위	≡ 메시지 번호	Aa 기능명	≡ 상세 설명	≡ SQL	≡ Parameter	≡ Return
AI	상	A-2	<u>자체 AI 요약</u>	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 contents	String
AI	상	A-1	<u>Lilys AI 요약</u>	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	String
컨텐츠	상	C-5	<u>컨텐츠 타겟 보드 선정</u>	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)	SELECT * FROM cards c JOIN boards b ON c.board_id = b.board_id WHERE b.user_id = userId	String userId String content	Integer
컨텐츠	상	C-4	<u>컨텐츠 수정</u>	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 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 content = content WHERE card_id = cardId	String userId String cardId String content	Boolean
컨텐츠	상	C-3	<u>컨텐츠 삭제</u>	Soft Delete 방식 사용 userId와 cardId로 cards 테이블에서 SELECT 삭제 권한 및 카드 유무 확인 필요 값이 존재하지 않으면 Exception 발생	SELECT EXISTS (SELECT 1 FROM cards c JOIN boards b ON c.board_id = b.board_id WHERE c.card_id = cardId	String userId String cardId	Boolean

☉ 도메인	☉ 우선순위	≡ 메시지 번호	Aa 기능명	≡ 상세 설명	≡ SQL	≡ Parameter	≡ Return
				cardId에 해당하는 card에 delete_date 를 NOW()로 갱신 UPDATE 실패 시 Exception 발생 성공 여부 응답을 true 로 반환	AND b.user_id = userId) UPDATE cards SET delete_date = NOW() WHERE card_id = cardId		
컨텐츠	상	C-2	컨텐츠 추가	userId와 URL로 cards 테이블에서 SELECT 값이 존재하면 Exception 발생 (User별 해당 URL 중복 확인) U-5 (URL 별 콘텐츠 종류 분석) 호출 후 type 값 선정 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 U-6 (String을 JSON 형식으로 직렬화) 호출 후 content 값 선정 C-5 (컨텐츠 타겟 보드 선정) 호출 후 position으로 boards 테이블에 board_id를 SELECT card가 들어갈 board의 status 변경 삽입 보드(board_id)와 내용(JSON), 콘텐츠 종류(0: YouTube, 1: Blog, 2: News), URL INSERT INSERT 실패 시 Exception 발생 성공 여부 응답을 true 반환	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 board_id FROM boards WHERE user_id = userId AND position = position UPDATE boards SET status = true WHERE board_id= boardId INSERT INTO cards (board_id, content, type, original_url) VALUES (String, JSON, Integer, URL)	String userId String URL	Boolean
컨텐츠	상	C-1	컨텐츠 리스트	userId와 boardId로 boards 테이블에서 SELECT 권한 및 보드 유무 확인 필요 값이 존재하지 않으면 Exception 발생 boardId로 card 테이블에서 모든 요소를 SELECT 후 List로 반환	SELECT EXISTS (SELECT 1 FROM boards WHERE board_id = boardId AND user_id = userId) SELECT * FROM cards WHERE board_id = boardId	String userId String boardId	List<String>
컨텐츠	중	C-7	컨텐츠 복사	userId와 URL로 cards 테이블에서 SELECT 값이 존재하면 Exception 발생 (User별 해당 URL 중복 확인) userId와 boardId로 boards 테이블에서 SELECT	SELECT EXISTS (SELECT 1 FROM cards c JOIN boards b ON c.board_id = b.board_id WHERE c.original_url	String userId String cardId String boardId	Boolean

☉ 도메인	☉ 우선순위	≡ 메소드 번호	Aa 기능명	≡ 상세 설명	≡ SQL	≡ Parameter	≡ Return
				<p>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>	<p>= URL AND b.user_id = userId)</p> <p>SELECT EXISTS (SELECT 1 FROM boards WHERE board_id = boardId AND user_id = userId)</p> <p>SELECT * FROM cards WHERE card_id = cardId</p> <p>INSERT INTO cards (board_id, content, type, original_url) VALUES (boardId, JSON, Integer, URL)</p>		
컨텐츠	중	C-6	<u>소속 보드 변경</u>	<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>	<p>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)</p> <p>SELECT EXISTS (SELECT 1 FROM boards WHERE board_id = boardId AND user_id = userId)</p> <p>UPDATE cards SET board_id= boardId WHERE card_id = cardId</p>	String userId String cardId String boardId	Boolean
보드	상	B-3	<u>보드 제거</u>	<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>SELECT EXISTS (SELECT 1 FROM boards WHERE board_id = boardId AND user_id = userId)</p> <p>DELETE FROM boards WHERE board_id =</p>	String userId Integer boardId Integer targetPosition	Boolean

☉ 도메인	☉ 우선순위	≡ 메소드 번호	Aa 기능명	≡ 상세 설명	≡ SQL	≡ Parameter	≡ Return
				targetPosition 이후의 모든 보드의 순서 -1 UPDATE 실패 시 Exception 발생 성공 여부 응답을 true 로 반환	boardId AND user_id = userId UPDATE boards SET position = position - 1 WHERE user_id = userId AND position > targetPosition		
보드	상	B-2	<u>보드 생성</u>	userId로 boards 테이블에서 총 board의 count 확인 userId와 count로 boards 테이블에 INSERT (board position의 index는 0부터 시작) INSERT 실패 시 Exception 발생 성공 여부 응답을 true 로 반환	SELECT COUNT(*) FROM boards WHERE user_id = userId INSERT INTO boards (user_id, board_name, position) VALUES (userId, boardName, count)	String userId String boardName	Boolean
보드	상	B-1	<u>보드 리스트</u>	userId로 boards 테이블에서 모든 요소를 SELECT 후 List로 반환	SELECT * FROM boards WHERE user_id = userId	String userId	List<String>
보드	중	B-4	<u>보드 순서 변경</u>	userId와 boardId로 boards 테이블에서 SELECT 편집 권한 및 보드 유무 확인 필요 값이 존재하지 않으면 Exception 발생 sourcePosition < targetPosition 인 경우와 sourcePosition > targetPosition 인 경우에 대해 다른 메소드 적용 위치가 변경되는 범위 내의 모든 board의 position UPDATE 이동한 board의 position을 targetPosition으로 UPDATE UPDATE 실패 시 Exception 발생 성공 여부 응답을 true 로 반환	SELECT EXISTS (SELECT 1 FROM boards WHERE board_id = boardId AND user_id = userId) - if sourcePosition < targetPosition UPDATE boards SET position = position - 1 WHERE user_id = userId AND position > sourcePosition AND position <= targetPosition - else if sourcePosition > targetPosition UPDATE boards SET position = position + 1 WHERE user_id = userId AND position >= targetPosition AND position < sourcePosition	String userId String boardId Integer sourcePosition Integer targetPosition	Boolean

☉ 도메인	☉ 우선순위	≡ 메소드 번호	Aa 기능명	≡ 상세 설명	≡ SQL	≡ Parameter	≡ Return
					UPDATE boards SET position = targetPosition WHERE board_id = boardId		
보드	하	B-5	<u>보드명 변경</u>	userId와 boardId로 boards 테이블에서 SELECT 수정 권한 및 보드 유무 확인 필요 값이 존재하지 않으면 Exception 발생 userId와 boardId가 일치하는 board의 board_name UPDATE UPDATE 실패 시 Exception 발생 성공 여부 응답을 true 로 반환	SELECT EXISTS (SELECT 1 FROM boards WHERE board_id = boardId AND user_id = userId) UPDATE boards SET board_name = boardName WHERE user_id = userId AND board_id = boardId	String userId String boardId String boardName	Boolean
유틸	상	U-6	<u>String을 JSON 형식으로 직렬화</u>	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 content	String
유틸	상	U-5	<u>URL별 콘텐츠 종류 분석</u>	URL 분석 후 콘텐츠 종류 반환 0: YouTube, 1: Blog, 2: News, 3: Image, 4: Shorts, 5: 미분류		String URL	Integer
유틸	상	U-3	<u>핵심 키워드 추출</u>	pynini(유사 api 가능)으로 핵심 키워드 뽑기 핵심 키워드 List 반환		String contents	List<String>
유틸	상	U-2	<u>UUID 생성</u>	다양한 PK에 활용될 UUID를 반환하도록 구현			String

☉ 도메인	☉ 우선순위	≡ 메소드 번호	Aa 기능명	≡ 상세 설명	≡ SQL	≡ Parameter	≡ Return
				<p>java.util.UUID를 import하여 구현</p> <p>UUID 객체를 발급 받고, toString() 메소드를 이용하여 String으로 변환 후 반환</p> <p>UUID는 하이픈을 제외한 길이가 32자로 고정</p>			
유틸	상	U-1	<u>이미지 저장</u>	<p>S3에 이미지 파일을 저장</p> <p>저장 후 생성된 URL을 String으로 반환</p>		MultipartFile file	String
유틸	상	U-4	<u>크롤링</u>	<p>org.jsoup.Jsoup를 import하여 구현</p> <p>크롤링한 데이터를 String으로 파싱 후 A-2 호출 후 반환</p> <p>실패 시 Exception 발생</p>		String URL	String
유틸	하	U-7	<u>STT</u>	<p>Fast Wisper API를 사용하여 구현</p> <p>크롤링한 데이터를 String으로 파싱 후 자체 AI 요약 메소드 호출</p> <p>실패 시 Exception 발생</p>		String URL	String
폐기	상		<u>유저 정보</u>	요청한 사용자의 프로필 정보 주기 (프로필 이미지, 닉네임, 팔로워 수, 팔로잉 수)			
폐기	중		<u>JWT 토큰</u>				
폐기	중		<u>개인 정보 수정</u>	닉네임, 비밀번호, 프로필 이미지 변경			
폐기	중	B-7	<u>티원 초대</u>				
폐기	중		<u>팔로잉 리스트</u>	해당 사용자가 팔로잉한 유저 목록			
폐기	중		<u>팔로워 리스트</u>	해당 사용자를 팔로잉한 유저 목록			
폐기	중		<u>팔로우/언팔로우</u>	다른 사용자에게 대한 팔로잉 상태 변환 (팔로잉 ↔ 언팔로잉)			
폐기	중	B-6	<u>즐거찾기 등록/취소</u>	상대방의 보드인 경우, 해당 보드 즐겨찾기 등록 및 취소			
폐기	중		<u>소셜 로그인</u>	<p>소셜(카카오, 구글, 네이버)을 활용해 서비스에 사용자 인증을 받는 기능</p> <p>DB에 존재하지 않는 정보의 경우, DB에 정보 저장 후 로그인</p>			
폐기	하		<u>회원탈퇴</u>	서비스에 저장된 사용자 계정을 만료하기 위한 기능			
폐기	하		<u>회원가입</u>	서비스에 가입을 하기 위한 기능			

⦿ 도메인	⦿ 우선순위	≡ 메소드 번호	Aa 기능명	≡ 상세 설명	≡ SQL	≡ Parameter	≡ Return
폐기	하		<u>로그인</u>	사용자 인증을 받아 권한을 주는 기능			
폐기			<u>알림</u>				
폐기			<u>시 요약 신고</u>				