# [2023-1 Software Engineering]

클래스 정의서(CS)

제목:

제출일:     2023년   5월   31일

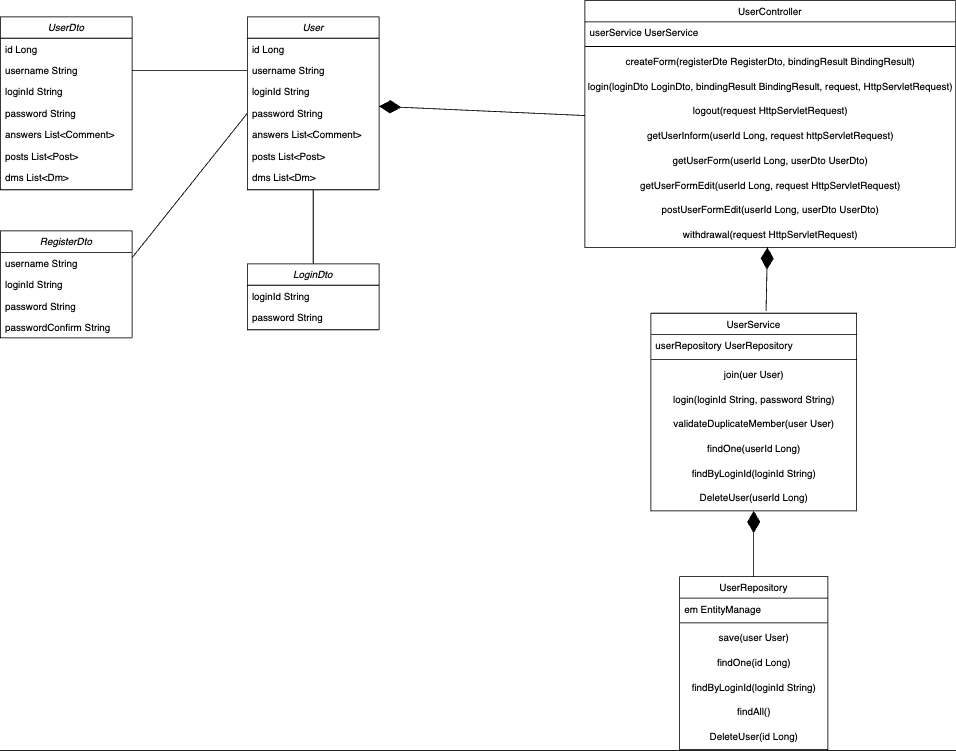
팀 명:서적집 막내아들

팀 원: 김재현, 김현성, 이재준, 정의엽, 조창민

1. 클래스 목록

| 구분 | 클래스 ID | 클래스 명 | 종류 | 설명 |
| --- | --- | --- | --- | --- |
| 유저 관리 | CL-UM-001 | User | entity | 데이터베이스에 실제로 저장되는 유저 객체이며, 저장, 수정, 삭제 등이 이루어지는 엔티티 클래스 |
| CL-UM-002 | UserDto | boundary | 데이터베이스에서 가져온 user 엔티티 정보를 외부로 노출시키지 않게 따로 정의해 둔 클래스 |
| CL-UM-003 | LoginDto | boundary | 외부에서 받은 로그인과 패스워드 정보를 직접 엔티티 클래스로 받지 않기 위해 정의해 둔 클래스 |
| CL-UM-004 | RegisterDto | boundary | 외부에서 받은 회원가입 정보를 직접 엔티티 클래스로 받지 않기 위해 정의해 둔 클래스 |
| CL-UM-005 | UserController | controller | 외부와 통신하기 위해 유저와 관련된 api를 정리한 컨트롤러 클래스 |
| CL-UM-006 | UserService | service | 컨트롤러로부터 요청되어 db에 접근해야 할 때 컨트롤러와, 직접 db에 접근하는 repository 사이에서 연결해주는 서비스 클래스 |
| CL-UM-007 | UseRepository | repository | 서비스 클래스에서 받은 요청에 따라 직접적으로 db에 정보를 추가, 수정, 삭제하는 리포지토리 클래스 |
| CL-UM-008 | UserExControllerAdvice | advice | 유저 관리에서 발생할 모든 예외를 처리하기 위한 클래스이다. 모두 RuntimeException을 상속받은 클래스들이고, ExceptionHandler를 통해 예외를 처리한다. |
| 게시판 관리 | CL-PM-001 | Post | entity | 데이터베이스에 실제로 저장되는 유저 객체이며, 저장, 수정, 삭제 등이 이루어지는 엔티티 클래스 |
| CL-PM-002 | PostDto | boundary | 데이터베이스에서 가져온 post 엔티티 정보를 외부로 노출시키지 않게 따로 정의해 둔 클래스 |
| CL-PM-003 | Comment | entity | 데이터베이스에 실제로 저장되는 댓글 객체이며, 저장, 수정, 삭제 등이 이루어지는 엔티티 클래스 |
| CL-PM-004 | CommentDto | boundary | 데이터베이스에서 가져온 comment 엔티티 정보를 외부로 노출시키지 않게 따로 정의해 둔 클래스 |
| CL-PM-005 | PostController | controller | 외부와 통신하기 위해 게시판과 관련된 api를 정리한 컨트롤러 클래스 |
| CL-PM-006 | PostService | service | 컨트롤러로부터 요청되어 db에 접근해야 할 때 컨트롤러와, 직접 db에 접근하는 repository 사이에서 연결해주는 서비스 클래스 |
| CL-PM-007 | PostRepostiroy | repository | 서비스 클래스에서 받은 요청에 따라 직접적으로 db에 정보를 추가, 수정, 삭제하는 리포지토리 클래스 |
| CL-PM-008 | BoardExControllerAdvice | advice | 게시판 관리에서 발생할 모든 예외를 처리하기 위한 클래스이다. 모두 RuntimeException을 상속받은 클래스들이고, ExceptionHandler를 통해 예외를 처리한다. |
| 서적 관리 | CL-BM-001 | Book | entity | 데이터베이스에 실제로 저장되는 책 객체이며, 저장, 수정, 삭제 등이 이루어지는 엔티티 클래스 |
| CL-BM-002 | BookDto | boundary | 데이터베이스에서 가져온 book 엔티티 정보를 외부로 노출시키지 않게 따로 정의해 둔 클래스 |
| CL-BM-003 | Author | entity | 데이터베이스에서 실제로 저장되는 책의 저자 객체이며, 저장, 수정, 삭제 등이 이루어지는 엔티티 클래스 |
| CL-BM-004 | AuthorDto | boundary | 데이터베이스에서 가져온 author 엔티티 정보를 외부로 노출시키지 않게 따로 정의해 둔 클래스 |
| CL-BM-005 | BookController | controller | 외부와 통신하기 위해 책과 관련된 api를 정리한 컨트롤러 클래스 |
| CL-BM-006 | BookService | service | 컨트롤러로부터 요청되어 db에 접근해야 할 때 컨트롤러와, 직접 db에 접근하는 repository 사이에서 연결해주는 서비스 클래스 |
| CL-BM-007 | BookRepository | repository | 서비스 클래스에서 받은 요청에 따라 직접적으로 db에 정보를 추가, 수정, 삭제하는 리포지토리 클래스 |
| CL-BM-008 | BookExControllerAdvice | advice | 책 관리에서 발생할 모든 예외를 처리하기 위한 클래스이다. 모두 RuntimeException을 상속받은 클래스들이고, ExceptionHandler를 통해 예외를 처리한다. |
| CL-BM-009 | BookInitializer | init | 서버를 시작하면 크롤링한 책의 정보를 모두 db에 저장시키기 위한 클래스 |
| 쪽지 관리 | CL-DM-001 | Dm | entity | 데이터베이스에 실제로 저장되는 쪽지 객체이며, 저장, 수정, 삭제 등이 이루어지는 엔티티 클래스 |
| CL-DM-002 | DmDto | boundary | 데이터베이스에서 가져온 dm 엔티티 정보를 외부로 노출시키지 않게 따로 정의해 둔 클래스 |
| CL-DM-003 | DmController | controller | 외부와 통신하기 위해 유저와 관련된 api를 정리한 컨트롤러 클래스 |
| CL-DM-004 | DmService | service | 컨트롤러로부터 요청되어 db에 접근해야 할 때 컨트롤러와, 직접 db에 접근하는 repository 사이에서 연결해주는 서비스 클래스 |
| CL-DM-005 | DmRepository | repository | 서비스 클래스에서 받은 요청에 따라 직접적으로 db에 정보를 추가, 수정, 삭제하는 리포지토리 클래스 |

1. 유저 관리
   1. 클래스 다이어그램



* 1. (CL-Um-001)User

1. 개요  
   데이터베이스에 실제로 저장되는 유저 객체이며, 저장, 수정, 삭제 등이 이루어지는 엔티티 클래스

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| id | private | Long | N/A | 유저의 db 고유 id |
| username | private | String | null | 유저의 닉네임 |
| loginId | private | String | null | 유저의 로그인 id |
| password | private | String | null | 유저의 로그인 패스워드 |
| answers | private | List<Comment> | null | 유저가 게시글에 달았던 댓글 |
| posts | private | List<Post> | null | 유저가 작성한 게시글 |
| dms | private | List<Dm> | null | 유저가 주고 받은 쪽지 |

C. (CL-UM-002)UserDto

1. 개요

데이터베이스에서 가져온 user 엔티티 정보를 외부로 노출시키지 않게 따로 정의해 둔 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| id | private | Long | N/A | 유저의 db 고유 id |
| username | private | String | null | 유저의 닉네임 |
| loginId | private | String | null | 유저의 로그인 id |
| password | private | String | null | 유저의 로그인 패스워드 |
| answers | private | List<Comment> | null | 유저가 게시글에 달았던 댓글 |
| posts | private | List<Post> | null | 유저가 작성한 게시글 |
| dms | private | List<Dml> | null | 유저가 주고 받은 쪽지 |

D. (CL-UM-003)LoginDto

1. 개요

외부에서 받은 로그인과 패스워드 정보를 직접 엔티티 클래스로 받지 않기 위해 정의해 둔 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| loginId | private | String | null | 입력받은 loginId |
| password | private | String | null | 입력받은 password |

E. (CL-UM-004)RegisterDto

1. 개요

외부에서 받은 회원가입 정보를 직접 엔티티 클래스로 받지 않기 위해 정의해 둔 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| loginId | private | String | null | 입력받은 loginId |
| password | private | String | null | 입력받은 password |
| passwordConfirm | private | String | null | 입력받은 passwordConfirm |
| username | private | String | null | 입력받은 username |

F. (CL-UM-005)UserController

1. 개요

외부와 통신하기 위해 유저와 관련된 api를 정리한 컨트롤러 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| userService | private | UserService | null | 컨트롤러에서 db로 요청되는 메시지를 전달하기 위한 클래스 |

1. 오퍼레이션

| 오퍼레이션명 | 가시성 | 반환타입 | 기본값 | | 설명 |
| --- | --- | --- | --- | --- | --- |
| 명칭 | 설명 |
| createForm | public | UserDto | RegisterDto: registerDto | 회원가입 폼 | /register  회원가입 |
| BindingResult:bindingResult | form에서 빈 칸 확인 |
| login | public | LoginDto | LoginDto: loginDto | 로그인 폼 | /login  로그인 |
| BindingResult: bindingResult | form에서 빈 칸 확인 |
| HttpServletRequest: request | 리퀘스트(세션 확인) |
| logout | public | void | HttpServletRequest: request | 리퀘스트(세션 확인) | /logout  로그아웃 |
| getUserInform | public | userDto | Long: userId | 유저의 db 고유 id | /my\_page/{userId}  마이페이지 유저 정보 보여줌 |
| HttpServletRequest: request | 리퀘스트(세션 확인) |
| getUserFormEdit | public | userDto | Long: userId | 유저의 db 고유 id | /my\_page/{userId}/edit  유저의 정보를 보내줌 |
| HttpServletRequest: request | 리퀘스트(세션 확인) |
| postUserFormEdit | public | userDto | Long: userId | 유저의 db 고유 id | /my\_page/{userId}/edit  수정된 유저의 정보를 보내줌 |
| UserDto: userDto | 수정된 user의 정보 |
| withdrawal | public | void | HttpServletRequest: request | 리퀘스트(세션 확인) | /withdrawal  회원탈퇴 |

G. (CL-UM-006)UserService

1. 개요

컨트롤러로부터 요청되어 db에 접근해야 할 때 컨트롤러와, 직접 db에 접근하는 repository 사이에서 연결해주는 서비스 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| userRepository | private | UserRepository | null | 서비스에서 db로 요청되는 메시지를 직접적으로 처리하기 위한 클래스 |

1. 오퍼레이션

| 오퍼레이션명 | 가시성 | 반환타입 | 기본값 | | 설명 |
| --- | --- | --- | --- | --- | --- |
| 명칭 | 설명 |
| join | public | Long | User: user | db에 저장할 유저 | 들어온 user를 db에 저장한다. |
| login | public | User | String: loginId | 입력된 loginId | 들어온 loginId와 password를 통해 로그인을 진행한다. |
| String: password | 입력된 password |
| validateDuplicateMember | private | void | User: user | db에 중복 유저가 있는지 확인 | 회원가입 과정에서 중복 회원을 확인한다. |
| findOne | public | User | Long: userId | 찾을 유저의 id | userId를 통해 user의 정보를 반환한다. |
| findByLoginId | public | User | String loginId | 찾을 유저의 loginId | loginId를 통해 user의 정보를 반환한다. |
| DeleteUser | public | void | Long: userId | 삭제할 유저의 userId | userId를 통해 db에서 user 정보를 삭제한다. |

H. (CL-UM-007)UserRepository

1. 개요

서비스 클래스에서 받은 요청에 따라 직접적으로 db에 정보를 추가, 수정, 삭제하는 리포지토리 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| em | private | EntityManage | null | db에 저장된 엔티티를 관리하기 위한 클래스 |

1. 오퍼레이션

| 오퍼레이션명 | 가시성 | 반환타입 | 기본값 | | 설명 |
| --- | --- | --- | --- | --- | --- |
| 명칭 | 설명 |
| save | public | void | User: user | db에 저장할 user 정보 | 전달된 user 정보를 db에 저장한다. |
| findOne | public | User | Long: id | db로부터 찾아야 할 userId | 전달된 userId를 db에서 찾아서 user 정보를 반환한다. |
| findByLoginId | public | Optional<User> | String: loginId | 입력받은 loginId로 로그인 시도 | 전달된 loginId와 일치하는 password가 있다면 로그인을 성공시키고 user 정보를 반환한다. |
| findAll | public | List<User> | N/A | | db에 저장된 모든 user 정보를 반환한다. |
| DeleteUser | public | void | Long: id | db에 저장된 user의 id | 전달된 id를 가진 user를 db에서 삭제시킨다. |

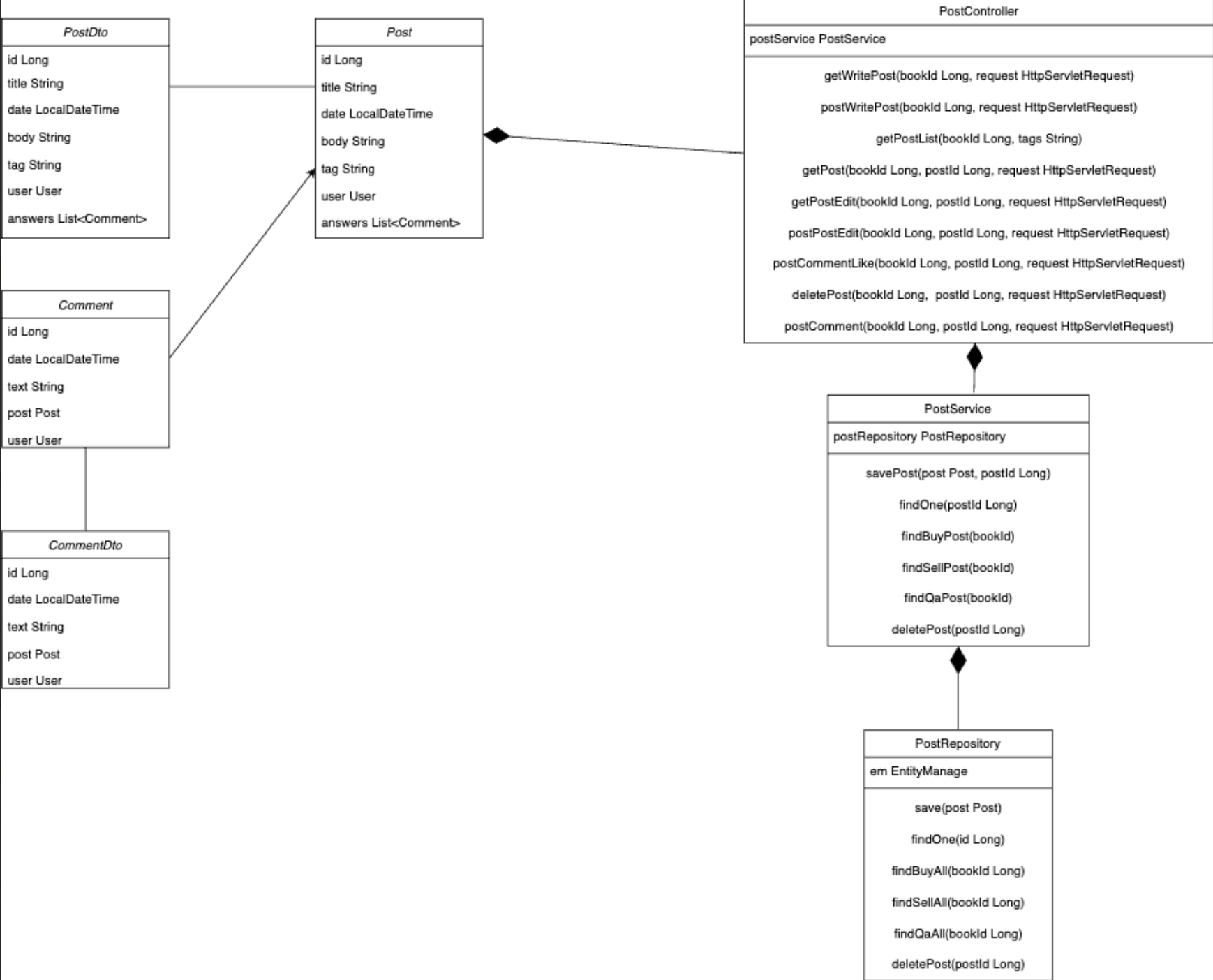
1. (CL-UM-008)UserExControllerAdvice
2. 개요

유저 관리에서 발생할 모든 예외를 처리하기 위한 클래스이다. 모두 RuntimeException을 상속받은 클래스들이고, ExceptionHandler를 통해 예외를 처리한다.

1. 오퍼레이션

| 오퍼레이션명 | 가시성 | 반환타입 | 기본값 | | 설명 |
| --- | --- | --- | --- | --- | --- |
| 명칭 | 설명 |
| DuplicateLoginIdException | public | ErrorResult | N/A | | /register에서 유저가 입력한 loginId가 이미 db에 있으면 발생할 예외를 잡는다. |
| NotExistUserException | public | ErrorResult | /login에서 유저가 입력한 loginId가 db에 없으면 발생할 예외를 잡는다. |
| NotMatchUserExceptionException | public | ErrorResult | /login에서 유저가 입력한 비밀번호가 맞지 않으면 발생할 예외를 잡는다. |
| NotLoginUserException | public | ErrorResult | 특정 url에 로그인을 하지 않은 유저가 접근하면 발생할 예외를 잡는다. |

1. 게시판 관리
   1. 클래스 다이어그램



* 1. (CL-PM-001)Post

1. 개요

데이터베이스에 실제로 저장되는 유저 객체이며, 저장, 수정, 삭제 등이 이루어지는 엔티티 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| id | private | Long | N/A | 게시글의 db 고유 id |
| title | private | String | null | 게시글의 제목 |
| date | private | LocalDateTime | null | 게시글이 생성, 수정된 날짜 |
| body | private | String | null | 게시글의 본문 내용 |
| tag | private | String | null | 게시글의 종류 (삽니다, 팝니다, 문제풀이) |
| user | private | User | null | 게시글을 작성한 유저 |
| book | private | Book | null | 게시글에 관련된 책 |
| answers | private | List<Comment> | null | 게시글에 달린 댓글 |

* 1. (CL-PM-002)PostDto

1. 개요

데이터베이스에서 가져온 post 엔티티 정보를 외부로 노출시키지 않게 따로 정의해 둔 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| id | private | Long | N/A | 게시글의 db 고유 id |
| title | private | String | null | 게시글의 제목 |
| date | private | LocalDateTime | null | 게시글이 생성, 수정된 날짜 |
| body | private | String | null | 게시글의 본문 내용 |
| tag | private | String | null | 게시글의 종류 (삽니다, 팝니다, 문제풀이) |
| user | private | User | null | 게시글을 작성한 유저 |
| book | private | Book | null | 게시글에 관련된 책 |
| answers | private | List<Comment> | null | 게시글에 달린 댓글 |

* 1. (CL-PM-003)Comment

1. 개요

데이터베이스에 실제로 저장되는 댓글 객체이며, 저장, 수정, 삭제 등이 이루어지는 엔티티 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| id | private | Long | N/A | 댓글의 db 고유 id |
| date | private | LocalDataTime | null | 댓글이 생성, 수정된 날짜 |
| text | private | String | null | 댓글의 내용 |
| post | private | Post | null | 댓글을 단 게시글 |
| user | private | User | null | 댓글을 단 유저 |

* 1. (CL-PM-004)CommentDto

1. 개요

데이터베이스에서 가져온 comment 엔티티 정보를 외부로 노출시키지 않게 따로 정의해 둔 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| id | private | Long | N/A | 댓글의 db 고유 id |
| date | private | LocalDataTime | null | 댓글이 생성, 수정된 날짜 |
| text | private | String | null | 댓글의 내용 |
| post | private | Post | null | 댓글을 단 게시글 |
| user | private | User | null | 댓글을 단 유저 |

* 1. (CL-PM-005)PostController

1. 개요

외부와 통신하기 위해 게시판과 관련된 api를 정리한 컨트롤러 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| postService | private | PostService | null | 컨트롤러에서 db로 요청되는 메시지를 전달하기 위한 클래스 |

1. 오퍼레이션

| 오퍼레이션명 | 가시성 | 반환타입 | 기본값 | | 설명 |
| --- | --- | --- | --- | --- | --- |
| 명칭 | 설명 |
| getWritePost | public | BookDto | Long: bookId | 책의 db 고유 id | /{bookId}/write 에서 bookId에 부합한 책에 대한 게시글 작성을 위한 클래스 |
| HttpServletRequest: request | 리퀘스트(세션 확인) |
| postWritePost | public | PostDto | Long: bookId | 책의 db 고유 id | /{bookId}/write 에서 유저가 게시글을 작성하고 등록을 누르면 게시글이 db에 저장 |
| HttpServletRequest: request | 리퀘스트(세션 확인) |
| getPostList | public | List<PostDto> | Long: bookId | 책의 db 고유 id | /{bookId}/{tags} 에서 pathvariable에 부합한 게시판의 정보들을 제공 |
| String: tags | 게시글의 종류 |
| getPost | public | PostDto | Long: bookId | 책의 db 고유 id | /{bookId}/{postId} 에서 pathvariable에 부합한 게시글의 정보를 제공 |
| Long: postId | 게시글의 db 고유 id |
| HttpServletRequest: request | 리퀘스트(세션 확인) |
| getPostEdit | public | PostDto | Long: bookId | 책의 db 고유 id | /{bookId}/{postId}/edit  에서 pathvariable 에 부합한 게시글 정보를 수정을 위한 클래스 |
| Long: postId | 게시글의 db 고유 id |
| HttpServletRequest: request | 리퀘스트(세션 확인) |
| postPostEdit | public | PostDto | Long: bookId | 책의 db 고유 id | /{bookId}/{postId}/edit  에서 pathvariable 에 부합한 게시글 정보를 수정 |
| Long: postId | 게시글의 db 고유 id |
| HttpServletRequest: request | 리퀘스트(세션 확인) |
| postCommentLike | public | PostDto | Long: bookId | 책의 db 고유 id | /{bookId}/{postId} 에서 댓글 옆에 좋아요 버튼을 누르면 좋아요 개수가 올라감 |
| Long: postId | 게시글의 db 고유 id |
| HttpServletRequest: request | 리퀘스트(세션 확인) |
| deletePost | public | void | Long: bookId | 책의 db 고유 id | /{bookId}/{postId}/delete 에 get method가 들어오면 pathvariable을 확인한 후 게시글 삭제 |
| Long: postId | 게시글의 db 고유 id |
| HttpServletRequest: request | 리퀘스트(세션 확인) |
| postComment | public | CommentDto | Long: bookId | 책의 db 고유 id | /{bookId}/{postId} 에서 댓글을 작성 |
| Long: postId | 게시글의 db 고유 id |
| HttpServletRequest: request | 리퀘스트(세션 확인) |

* 1. (CL-PM-006)PostService

1. 개요

컨트롤러로부터 요청되어 db에 접근해야 할 때 컨트롤러와, 직접 db에 접근하는 repository 사이에서 연결해주는 서비스 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| postRepository | private | PostRepository | null | 컨트롤러에서 db로 요청되는 메시지를 전달하기 위한 클래스 |

1. 오퍼레이션

| 오퍼레이션명 | 가시성 | 반환타입 | 기본값 | | 설명 |
| --- | --- | --- | --- | --- | --- |
| 명칭 | 설명 |
| savePost | public | Long | Post: post | db에 저장할 게시글 | 들어온 post를 db에 저장한다. |
| findOne | public | Post | Long: postId | 찾을 게시글의 id | postId를 통해 post의 정보를 반환한다. |
| findBuyPost | public | List<Post> | Long: bookId | 게시글이 작성된 책 | bookId를 통해 book을 찾고 book에 달린 게시글 중 tags가 buy인 post만 반환한다. |
| findSellPost | public | List<Post> | Long: bookId | 게시글이 작성된 책 | bookId를 통해 book을 찾고 book에 달린 게시글 중 tags가 sell인 post만 반환한다. |
| findQaPost | public | List<Post> | Long: bookId | 게시글이 작성된 책 | bookId를 통해 book을 찾고 book에 달린 게시글 중 tags가 qa인 post만 반환한다. |
| deletPost | public | void | Long: postId | 삭제할 게시글의 postId | postId를 통해 db에서 post 정보를 삭제한다. |

* 1. (CL-PM-007)PostRepository

1. 개요

서비스 클래스에서 받은 요청에 따라 직접적으로 db에 정보를 추가, 수정, 삭제하는 리포지토리 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| em | private | EntityManage | null | db에 저장된 엔티티를 관리하기 위한 클래스 |

1. 오퍼레이션

| 오퍼레이션명 | 가시성 | 반환타입 | 기본값 | | 설명 |
| --- | --- | --- | --- | --- | --- |
| 명칭 | 설명 |
| save | public | Long | Post: post | db에 저장할 post 정보 | 전달된 post 정보를 db에 저장한다. |
| findOne | public | Post | Long: postId | db로부터 찾아야 할 postId | 전달된 postId를 db에서 찾아서 post 정보를 반환한다. |
| findBuyAll | public | List<Post> | Long: bookId | 게시글이 작성된 책 | bookId에 부합한 책에 게시글 중 tags가 buy인 게시글만 반환한다. |
| findSellAll | public | List<Post> | Long: bookId | 게시글이 작성된 책 | bookId에 부합한 책에 게시글 중 tags가 sell인 게시글만 반환한다. |
| findQaAll | public | List<Post> | Long: bookId | 게시글이 작성된 책 | bookId에 부합한 책에 게시글 중 tags가 qa인 게시글만 반환한다. |
| deletePost | public | void | Long: postId | db에 저장된 post의 id | 전달된 postId를 가진 post를 db에서 삭제시킨다. |

* 1. (CL-PM-008)PostExControllerAdvice

1. 개요

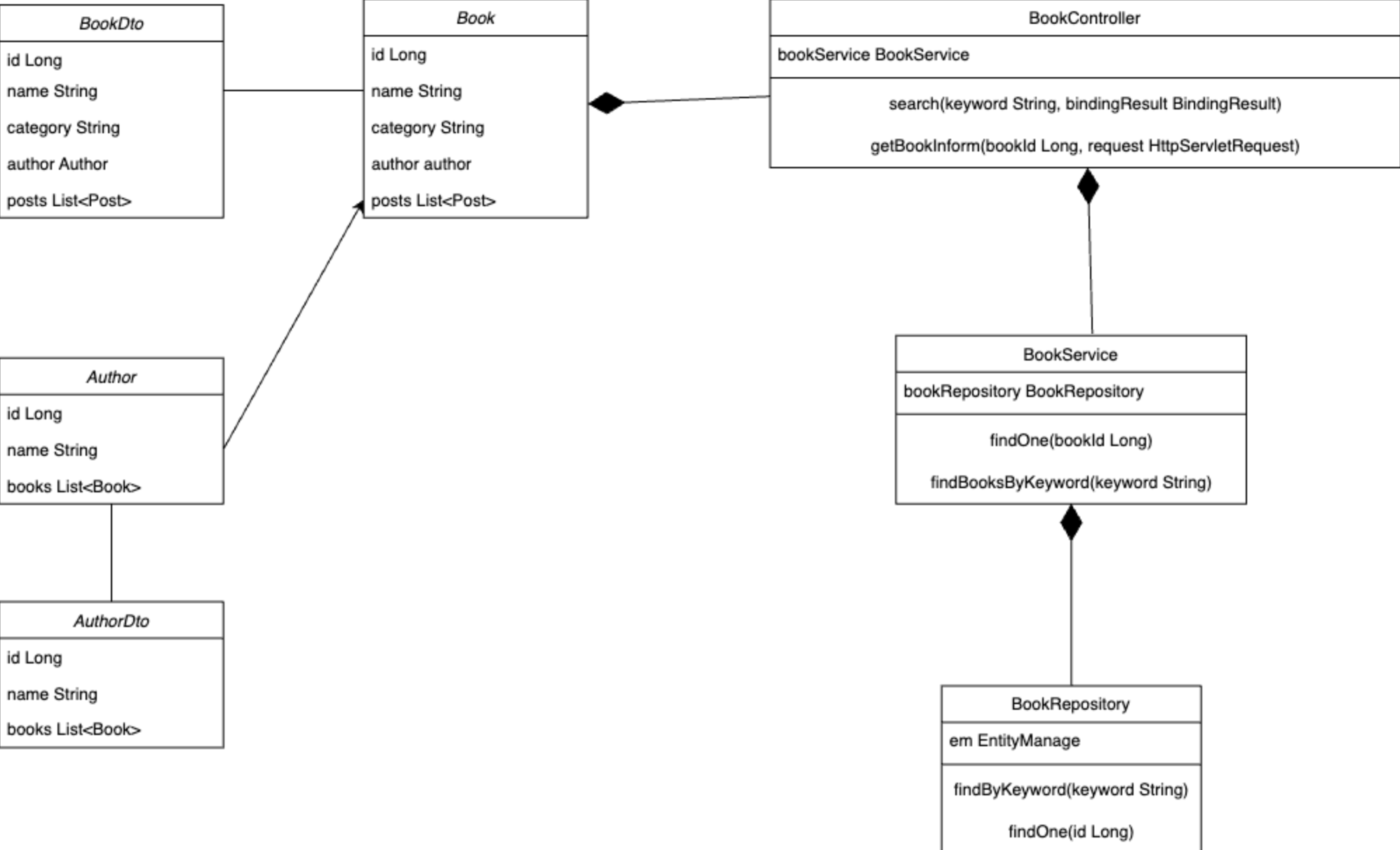
게시판 관리에서 발생할 모든 예외를 처리하기 위한 클래스이다. 모두 RuntimeException을 상속받은 클래스들이고, ExceptionHandler를 통해 예외를 처리한다.

1. 오퍼레이션

| 오퍼레이션명 | 가시성 | 반환타입 | 기본값 | | 설명 |
| --- | --- | --- | --- | --- | --- |
| 명칭 | 설명 |
| invalidUserExHandle | public | ErrorResult | N/A | | 게시글을 작성하지 않은 유저가 edit이나, delete를 시도할 때 발생할 예외를 잡는다. |
| notExistPostExHandle | public | ErrorResult | 존재하지 않는 게시글의 정보로 접근했을 때 발생할 예외를 잡는다. |

4. 서적 관리

1. 클래스 다이어그램



1. (CL-BM-001)Book
2. 개요

데이터베이스에 실제로 저장되는 책 객체이며, 저장, 수정, 삭제 등이 이루어지는 엔티티 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| id | private | Long | N/A | 책의 db 고유 id |
| name | private | String | null | 책의 닉네임 |
| category | private | String | null | 책의 종류 |
| author | private | String | null | 책의 저자 |
| posts | private | List<Post> | null | 책에 등록된 게시글 |

1. (CL-BM-002)BookDto
2. 개요

데이터베이스에서 가져온 book 엔티티 정보를 외부로 노출시키지 않게 따로 정의해 둔 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| id | private | Long | N/A | 책의 db 고유 id |
| name | private | String | null | 책의 닉네임 |
| category | private | String | null | 책의 종류 |
| author | private | String | null | 책의 저자 |
| posts | private | List<Post> | null | 책에 등록된 게시글 |

1. (CL-BM-003)Author
2. 개요

데이터베이스에서 실제로 저장되는 책의 저자 객체이며, 저장, 수정, 삭제 등이 이루어지는 엔티티 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| id | private | Long | N/A | 저자의 db 고유 id |
| name | private | String | null | 저자의 닉네임 |
| books | private | List<Book> | null | 저자의 책 |

1. (CL-BM-004)AuthorDto
2. 개요

데이터베이스에서 가져온 author 엔티티 정보를 외부로 노출시키지 않게 따로 정의해 둔 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| id | private | Long | N/A | 저자의 db 고유 id |
| name | private | String | null | 저자의 닉네임 |
| books | private | List<Book> | null | 저자의 책 |

1. (CL-BM-005)BookController
2. 개요

외부와 통신하기 위해 책과 관련된 api를 정리한 컨트롤러 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| bookService | private | BookService | null | 컨트롤러에서 db로 요청되는 메시지를 전달하기 위한 클래스 |

1. 오퍼레이션

| 오퍼레이션명 | 가시성 | 반환타입 | 기본값 | | 설명 |
| --- | --- | --- | --- | --- | --- |
| 명칭 | 설명 |
| search | public | List<BookDto> | String: keyword | 검색한 내용 | /search 에서 유저가 입력한 책 정보를 통해 알맞은 책의 정보를 반환한다. |
| BindingResult:bindingResult | form에서 빈 칸 확인 |
| getBookInform | public | BookDto | Long: bookId | 책의 db 고유 id | bookId에 부합한 책의 정보를 반환 |
| HttpServletRequest: request | 리퀘스트(세션 확인) |

1. (CL-BM-006)BookService
2. 개요

컨트롤러로부터 요청되어 db에 접근해야 할 때 컨트롤러와, 직접 db에 접근하는 repository 사이에서 연결해주는 서비스 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| bookRepository | private | BookRepository | null | 서비스에서 db로 요청되는 메시지를 직접적으로 처리하기 위한 클래스 |

1. 오퍼레이션

| 오퍼레이션명 | 가시성 | 반환타입 | 기본값 | | 설명 |
| --- | --- | --- | --- | --- | --- |
| 명칭 | 설명 |
| findOne | public | void | Long: bookId | 찾을 책의 id | bookId를 통해 book의 정보를 반환한다. |
| findBooksByKeyword | public | List<Book> | String keyword | 유저가 검색한 내용 | keyword가 들어간 책의 정보를 반환한다. |

1. (CL-BM-007)BookRepository
2. 개요

서비스 클래스에서 받은 요청에 따라 직접적으로 db에 정보를 추가, 수정, 삭제하는 리포지토리 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| em | private | EntityManage | 0 | db에 저장된 엔티티를 관리하기 위한 클래스 |

1. 오퍼레이션

| 오퍼레이션명 | 가시성 | 반환타입 | 기본값 | | 설명 |
| --- | --- | --- | --- | --- | --- |
| 명칭 | 설명 |
| findByKeyword | public | List<Book> | String: keyword | 유저가 검색한 내용 | 전달된 keyword를 포함한 db에서 제목, 저자, 카테고리에 따라 book 정보를 반환한다. |
| findOne | public | Book | Long: id | db로부터 찾아야 할 bookId | 전달된 bookId를 db에서 찾아서 book 정보를 반환한다. |

1. (CL-BM-008)BookExControllerAdvice
2. 개요

책 관리에서 발생할 모든 예외를 처리하기 위한 클래스이다. 모두 RuntimeException을 상속받은 클래스들이고, ExceptionHandler를 통해 예외를 처리한다.

1. 오퍼레이션

| 오퍼레이션명 | 가시성 | 반환타입 | 기본값 | | 설명 |
| --- | --- | --- | --- | --- | --- |
| 명칭 | 설명 |
| notExistBookExHandle | public | ErrorResult | N/A | | 존재하지 않는 책의 정보로 접근했을 때 발생할 예외를 잡는다. |

1. (CL-BM-009)BookDataInitializer
2. 개요

서버를 시작하면 크롤링한 책의 정보를 모두 db에 저장시키기 위한 클래스

1. 속성

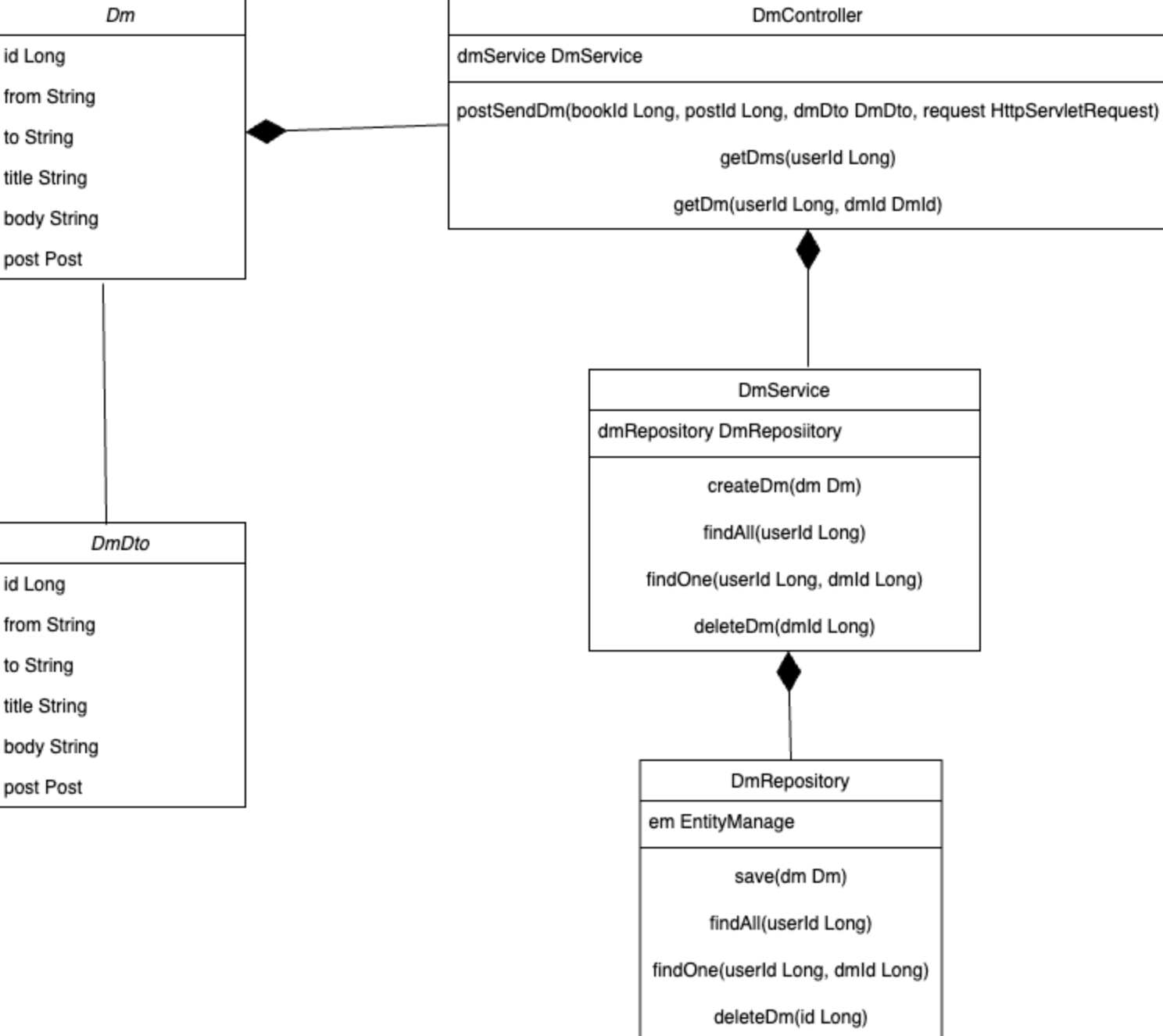
| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| bookService | private | BookService | null | 엑셀에 있는 정보를 db에 저장하기 위해 요청되는 메시지를 전달하기 위한 클래스 |

1. 오퍼레이션

| 오퍼레이션명 | 가시성 | 반환타입 | 기본값 | | 설명 |
| --- | --- | --- | --- | --- | --- |
| 명칭 | 설명 |
| init | public | void | N/A | | 엑셀에 있는 책의 정보들을 db에 저장시킨다. |

5. 쪽지 관리

1. 클래스 다이어그램



1. (CL-DM-001)Dm
2. 개요

데이터베이스에 실제로 저장되는 쪽지 객체이며, 저장, 수정, 삭제 등이 이루어지는 엔티티 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| id | private | Long | N/A | 쪽지의 db 고유 id |
| from | private | String | null | 쪽지를 보낸 유저 |
| to | private | String | null | 쪽지를 받은 유저 |
| title | private | String | null | 쪽지의 제목 |
| body | private | String | null | 쪽지의 본문 내용 |
| post | private | Post | null | 쪽지를 보낸 게시글 |

1. (CL-DM-002)DmDto
2. 개요

데이터베이스에서 가져온 dm 엔티티 정보를 외부로 노출시키지 않게 따로 정의해 둔 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| id | private | Long | N/A | 쪽지의 db 고유 id |
| from | private | String | null | 쪽지를 보낸 유저 |
| to | private | String | null | 쪽지를 받은 유저 |
| title | private | String | null | 쪽지의 제목 |
| body | private | List<Comment> | null | 쪽지의 본문 내용 |
| post | private | Post | null | 쪽지를 보낸 게시글 |

1. (CL-DM-003)DmController
2. 개요

외부와 통신하기 위해 유저와 관련된 api를 정리한 컨트롤러 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| dmService | private | DmService | null | 컨트롤러에서 db로 요청되는 메시지를 전달하기 위한 클래스 |

1. 오퍼레이션

| 오퍼레이션명 | 가시성 | 반환타입 | 기본값 | | 설명 |
| --- | --- | --- | --- | --- | --- |
| 명칭 | 설명 |
| postSendDm | public | DmDto | Long: bookId | 책의 db 고유 id | /{bookId}/{postId}/dm  에서 사용자로부터 받은 입력을 db에 쪽지 entity를 저장한다. |
| Long: postId | 게시글의 db 고유 id |
| DmDto: dmDto | 외부로부터 받은 쪽지 정보 |
| BindingResult:bindingResult | form에서 빈 칸 확인 |
| HttpSercvletRequest: request | 리퀘스트(세션 확인) |
| getDms | public | List<DmDto> | Long: userId | 유저의 db 고유 id | /my\_page/dm 에서 자신이 보낸, 받은 쪽지 정보를 확인할 수 있다. |
| getDm | public | DmDto | Long: userId | 유저의 db 고유 id | 특정 쪽지 정보 하나를 가져온다. |
| Long: dmId | 쪽지의 db 고유 id |

1. (CL-DM-004)DmService
2. 개요

컨트롤러로부터 요청되어 db에 접근해야 할 때 컨트롤러와, 직접 db에 접근하는 repository 사이에서 연결해주는 서비스 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| dmRepository | private | DmRepository | null | 서비스에서 db로 요청되는 메시지를 직접적으로 처리하기 위한 클래스 |

1. 오퍼레이션

| 오퍼레이션명 | 가시성 | 반환타입 | 기본값 | | 설명 |
| --- | --- | --- | --- | --- | --- |
| 명칭 | 설명 |
| createDm | public | void | Dm: dm | db에 저장할 dm | dm 정보를 db에 저장한다. |
| findAll | public | List<Dm> | Long: userId | 쪽지 정보를 보고 싶은 유저의 id | userId를 받으면 user가 받은, 보낸 쪽지 목록을 반환한다. |
| findOne | public | Dm | Long: dmId | 쪽지의 db 고유 id | my\_page/{userId} 에서 특정 쪽지 정보를 확인한다. |
| Long: userId | 쪽지를 보낸, 받은 유저 |
| deleteDm | public | void | Long: dmId | 삭제할 쪽지의 dmId | dmId를 통해 db에서 dm 정보를 삭제한다. |

1. (CL-DM-005)DmRepository
2. 개요

서비스 클래스에서 받은 요청에 따라 직접적으로 db에 정보를 추가, 수정, 삭제하는 리포지토리 클래스

1. 속성

| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| --- | --- | --- | --- | --- |
| em | private | EntityManage | null | db에 저장된 엔티티를 관리하기 위한 클래스 |

1. 오퍼레이션

| 오퍼레이션명 | 가시성 | 반환타입 | 기본값 | | 설명 |
| --- | --- | --- | --- | --- | --- |
| 명칭 | 설명 |
| save | public | void | Dm: dm | db에 저장할 dm 정보 | 전달된 dm 정보를 db에 저장한다. |
| findAll | public | List<Dm> | Long: userId | 쪽지 목록을 보고 싶은 유저의 id | db에 저장된 모든 dm 정보를 반환한다. |
| findOne | public | Dm | Long: userId | 쪽지를 보낸 유저 | 특정 쪽지 하나의 정보를 반환한다. |
| Long: dmId | 쪽지의 db 고유 id |
| deleteDm | public | void | Long: id | db에 저장된 dm의 id | 전달된 id를 가진 dm을 db에서 삭제시킨다. |