



Inoh-J / quiet-chatter



<> Code

Pull requests

Actions

Projects

Wiki

Security

Insights

quiet-chatter / README.md



MASKUN2 docs: 회고 수정

0777e59 · last month



340 lines (294 loc) · 15.8 KB

Preview

Code

Blame

Raw



우아한테크코스(Woowa Tech Course) 웹 백엔드 8기 4-5주 차 프리코스 오픈미션 과제

오픈 미션: 프리코스 챌린지

문서 : [mission.md](#)

나의 오픈미션 목표

목표: 오픈미션 과제로 수줍은 사람들을 연결하기 위한 독서 SNS **Quiet Chatter** 을 출시한다.

제한 조건:

- 지난 프리코스 [1주](#), [2주](#), [3주](#)차 피드백을 준수한다. (아래 요약)
 - README.md와 기능 목록을 상세히 작성하고 업데이트한다, 커밋 메시지를 의미 있게 작성한다
 - Java에서 제공하는 API를 적극 활용한다. 연관성이 있는 상수는 static final 대신 enum 을 활용한다
 - 오류를 찾을 때 출력 함수 대신 디버거를 사용한다, 공백을 의미 있게 사용한다, 의미 없는 주석을 달지 않는다, 이름을 통해 의도를 드러낸다, 이름에 자료형을 사용하거나 축약하지 않는다
 - 객체의 상태 접근을 제한한다, 데이터를 꺼내지(get)않고 일하도록 한다, 값을 하드 코딩하지 않는다,
 - 상수, 멤버 변수, 생성자, 메서드의 순서를 지킨다, 필드(인스턴스 변수)의 수를 줄이기 위해 노력한다
 - 메서드가 한 가지 기능만 수행하게 한다, 15라인이 넘지 않도록 구현한다

- 테스트의 이유를 @DisplayName에 작성한다, 작은 단위의 테스트를 먼저 작성한다, 예외 케이스도 테스트한다
- 테스트를 구현 코드에서 분리한다, 테스트가 어려운 코드를 쉽게 만든다
- 지난 프리코스과 같이 [AngularJS Git Commit Message Conventions](#), [Java Style Guide](#)을 준수한다.

도전할 것

- 스스로 지난 3주차 동안 배운 것을 적극 적용한다.
 - 도입할 기술에 대한 필요성을 적극적으로 고민한다.
 - 저수준의 도메인 모델을 집약하여 고수준의 도메인 모델을 만든다
 - 의존성을 인터페이스를 통해서 느슨하게 만든다, 계층사이의 의존의 방향이 외부(어댑터)에서 내부(도메인)으로 흐르게 정렬한다
 - 테스트 없는 프로덕트 코드를 지양한다. 테스트가 어려운 의존은 모킹하여 테스트한다, 외부 API나 확률을 모킹한다. 되도록 스프링 컨텍스트가 없는 가벼운 테스트를 작성할 수 있도록 최적화한다.
 - 원시값을 도메인 의미가 있는 값객체(Value Object)로 감싸서 사용한다. 필요한 경우 일급컬렉션 사용한다. 객체의 불변성과 가변성을 맥락에 맞도록 사용한다. 값->불변, 엔티티->가변

시작하게된 배경

... 나는 특정한 주제로 사람을 연결하는 일에 줄곧 흥미를 느껴왔다. 모임을 운영하며 나름의 보람이 있었지만 많은 사람들이 스스로의 능력을 과소평가하고 본인을 드러내기 쑥스러워 하는 까닭에 모임의 문턱 앞에서 돌아갔던 기억들이 여전히 마음 한켠에 숙제처럼 남아있다.

아마도 '나는 이곳에 어울리지 않아' 같은 내적인 속삭임으로 대변될 수 있을 그 마음들을 생각하면서, 이 문제를 해결하기 위한 과제로써 이번 오픈 미션을 시작해보면 좋겠다고 생각했다. 그리고 그 미션의 결과물이 '너는 이곳에 어울려' 라는 메시지를 전달해줄 수 있길 원했다.

그래서 나는 수줍음이 많은 사람들을 위한 독서 SNS 프로젝트를 생각했다 ...

-오픈미션 회고 중에서-

오픈미션 회고

오픈미션 진행과정의 자세한 회고를 아래 문서에서 확인할 수 있습니다!

👉 [오픈미션 회고.md](#)

Quiet Chatter : you belong here



서비스 설명

이 서비스는 수줍은 사람들을 위한 독서나눔 서비스가 있으면 좋겠다고 생각해서 만들었습니다 :) 로그인 없이 다른 분이 읽어봤으면 하는 책을 검색해서 북톡(BookTalk)으로 추천사 및 감상평을 남길 수 있고 좋아요나 공감하는 반응을 표현할 수 있습니다.

주요 기능

- 개인정보 수집 없는 익명 활동
- 250자 짧은 추천사/감상평 : 북톡 (Book Talk)
- 일정 시간이 지나면 글 비공개 : 자동 숨김(Self-hidden)
- 조용한 리액션 : 👍 좋아요, ❤️ 공감해요 (댓글 및 멘션 불가)

이렇게 사용해 보세요 :)



헤르만

검색

새로 올라온 북톡



파친코 (합본 한정판) (이민진 장편소설)

이민진

" 입체적으로 느껴진 인물들은 고한수, 노아 였다. 고한수의 캐릭터가 무척 인상적인데 분명 선자의 아이인 노아가 자신의 아들임을 알았고 대부분의 개입들이 자신의 핏줄에대한 집착..."

👍 1 ❤️ 0



직장 상사 악령 퇴치부 (여름 한정 토무당 서핑 에디션)

이사구

"내가 보고 싶어 했던 책들은 대개 호흡이 느리고 깊은 집중을 요했다. 집중을 할 수 없는 상황 속에서 책을 못 읽어내는 나를 보며 자주 좌절했고 자책했다. 버스를 타기 위해 씬..."

👍 1 ❤️ 1



AI와 금융의 미래 (인공지능은 어떻게 금융 권력을 재편하는가)

나현중

"정말 ? 인간의 판단력(Judgment), 실행 시스템(Execution)에서 나온다고 할 수 있을까?"

👍 0 ❤️ 0



그럼에도 나는 이 삶을 사랑하므로 (꽃기는 영혼을 위한 헤세의 편지)

헤르만헤세

"사랑해요! 헤르만 헤세 내게 정말 많은 영감을 주었습니다"

👍 1 ❤️ 1

요구사항 및 기능목록

- ☒ 사용자는 웹 브라우저로 접속할 수 있다.
- ☒ 키워드(제목, 저자 등)으로 책을 검색할 수 있다.
- ☒ 책 상세보기할 수 있다.
- ☒ 책 상세보기 페이지에서 다른 사람의 독후감(북톡)을 볼 수 있다.
- ☒ 독후감(북톡)은 250자이내로 작성할 수 있다.
- ☒ 북톡에 리액션이 가능하다. 좋아요, 공감해요
- ☐ 기본적으로 1년이 지난 북톡은 비공개된다.
- ☒ 따로 로그인 하지 않아도 북톡을 작성할 수 있다.
 - ☐ 비 로그인 북톡은 1년후 숨김처리된다.
 - ☐ 수정/삭제 가능하지만 재접속 후에는 불가(로그인시에는 가능)

비기능 목록

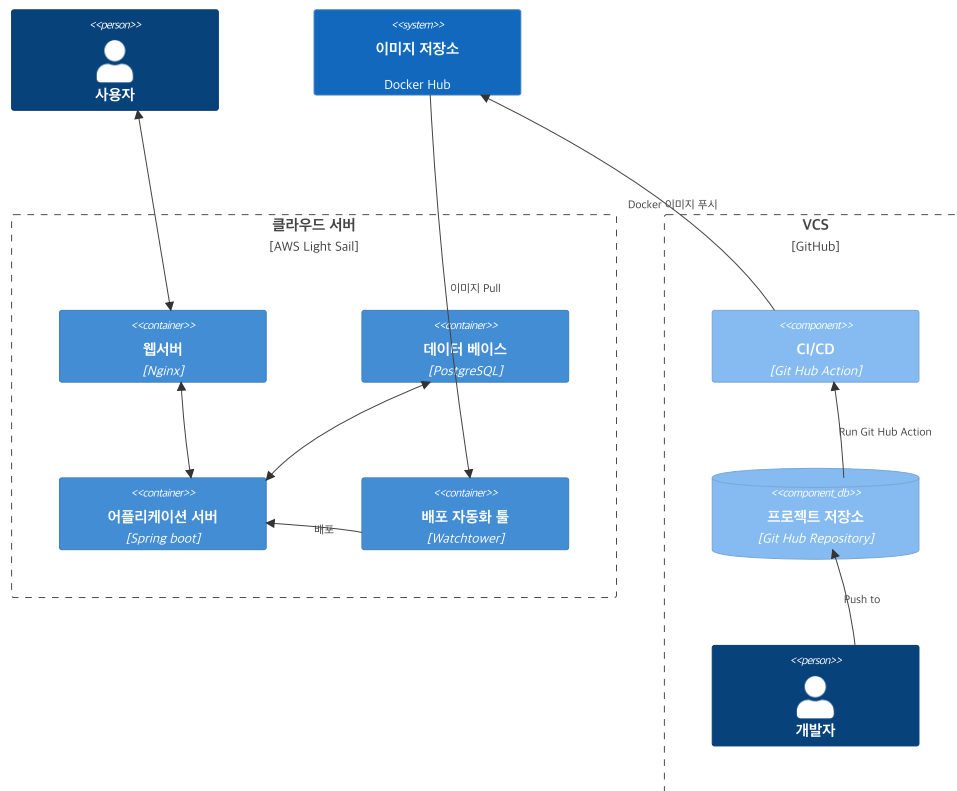
- ☒ HTTPS, 사용자 친화적 도메인주소로 접속가능하게 한다
- ☒ 추천 북톡 등 빈번한 조회 API의 응답성을 높인다 : 캐시전략 등
- ☒ 외부 도서 API를 사용하여 통합한다 : 네이버
- ☒ 모바일 접근성을 고려한 UI를 구현한다

- ☑ 확장성이 좋은 소프트웨어 구조를 사용한다 : 헥사고널 아키텍처
- ☑ 가용성을 저해하는 외부 공격을 막는 수단을 구축한다 : 초당 API 제한

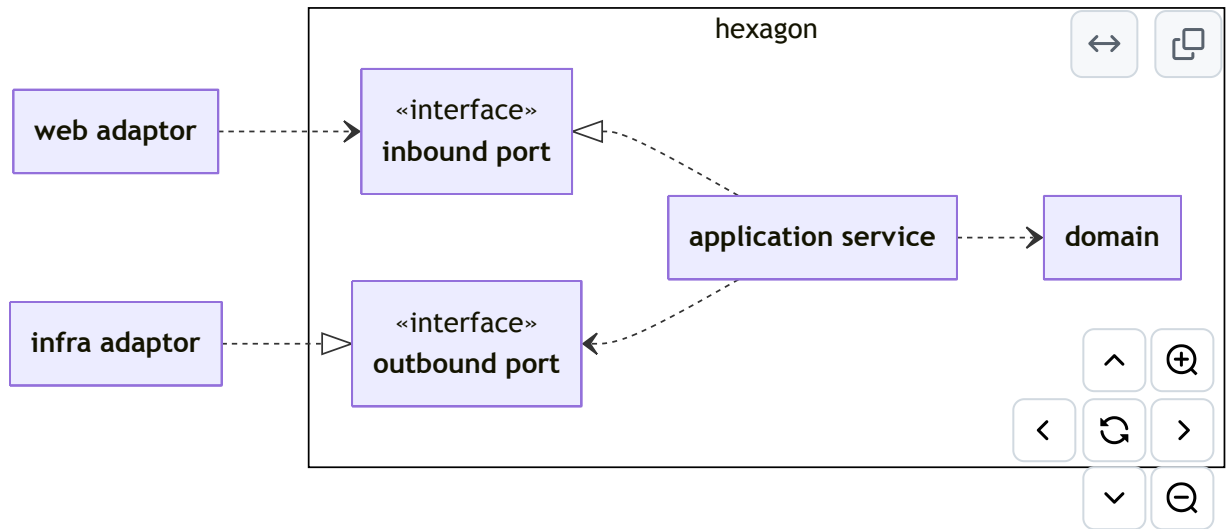
도메인

- 책 Book : ISBN, 제목, 출판사, 저자, 작성자의 속성을 가진다.
 - 멤버 Member : 접속한 유저를 통칭. 게스트, 정회원, 어드민으로 나뉜다.
 - 북톡 Talk : 250자이내로 작성, 숨김시간을 가진다.
 - 리액션 Reaction : 북톡에대한 리액션이 가능하다. 좋아요, 공감해요
 - 자동 숨김 Self Hidden : 기본적으로 1년이 지나면 글은 자동으로 숨김된다.
-

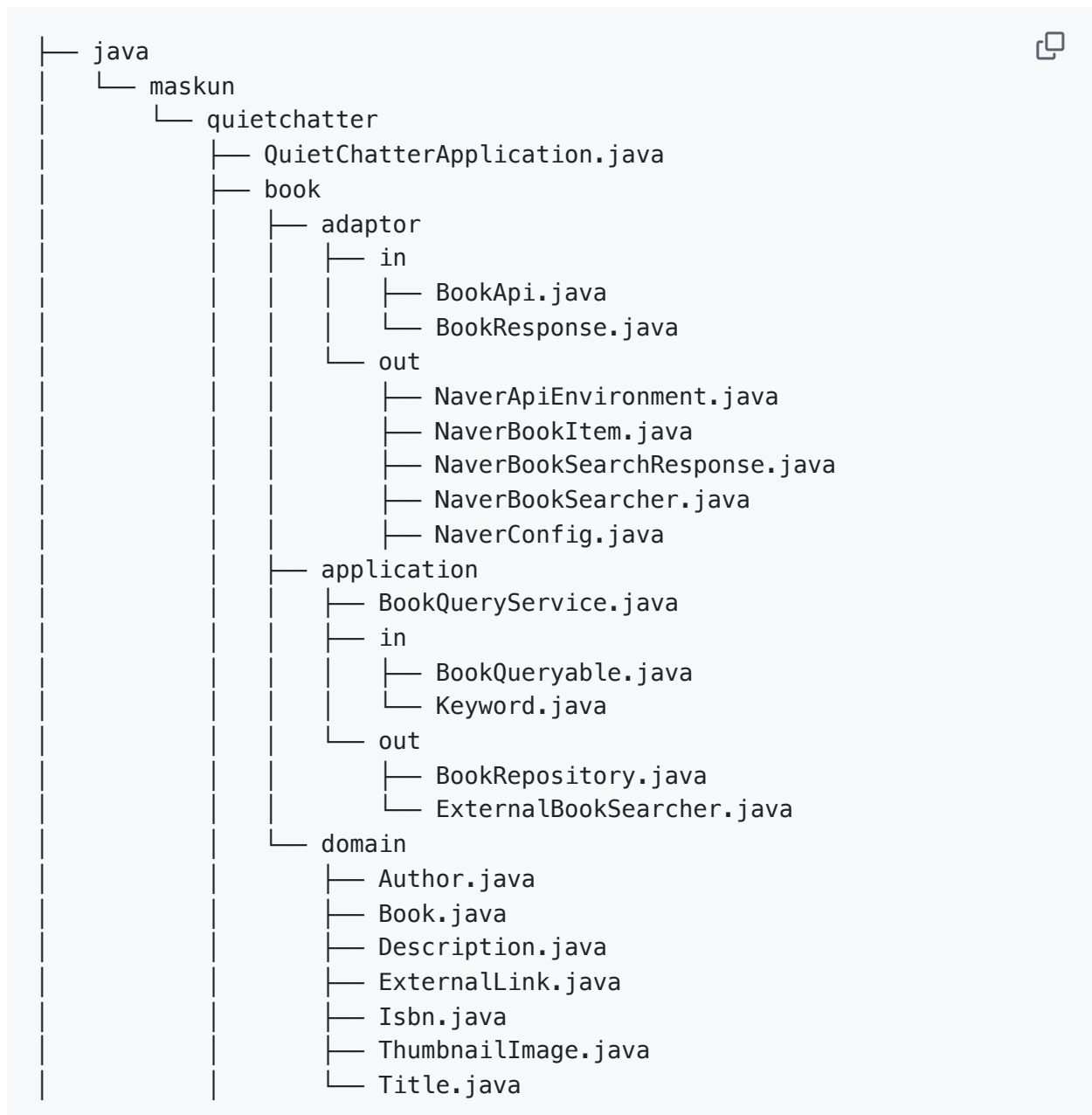
인프라스트럭처 아키텍처



애플리케이션 아키텍처 (Hexagonal Architecture)



패키지 구조



```
customer
├── adaptor
│   ├── in
│   │   └── CustomerMessageApi.java
│   └── out
├── application
│   ├── CustomerMessageService.java
│   ├── in
│   │   └── CustomerMessageCreatable.java
│   ├── out
│   │   └── CustomerMessageRepository.java
└── domain
    ├── CustomerMessage.java
    └── Message.java

member
├── adaptor
│   ├── in
│   └── out
├── application
│   ├── MemberRegistrator.java
│   ├── in
│   │   └── GuestRegistrable.java
│   ├── out
│   │   └── MemberRepository.java
└── domain
    ├── Id.java
    ├── Login.java
    ├── Member.java
    ├── Password.java
    ├── Role.java
    └── Status.java

reaction
├── adaptor
│   ├── in
│   │   ├── ReactionCommandApi.java
│   │   └── ReactionWebRequest.java
│   └── out
│       ├── Action.java
│       ├── ReactionBatchWorker.java
│       ├── ReactionEvent.java
│       ├── ReactionEventHandler.java
│       └── ReactionRequestAggregator.java
├── application
│   ├── ReactionQueryService.java
│   ├── ReactionService.java
│   ├── in
│   │   ├── ReactionModifiable.java
│   │   ├── ReactionQueryable.java
│   │   └── ReactionTarget.java
│   └── out
│       ├── ReactionEventPublisher.java
│       └── ReactionRepository.java
└── domain
```



```
└─ Reaction.java
└─ shared
  └─ persistence
    └─ BaseEntity.java
    └─ JpaConfig.java
  └─ security
    └─ AuthenticatedMemberToken.java
    └─ GuestAuthenticationFilter.java
    └─ GuestAuthenticationProvider.java
    └─ GuestAuthenticationTokenProvider.java
    └─ GuestPathMatcherProvider.java
    └─ GuestPromotion.java
    └─ MemberDetailService.java
    └─ MemberDetails.java
    └─ SecurityConfig.java
  └─ web
    └─ ErrorResponse.java
    └─ IdResponse.java
    └─ RestClientConfig.java
    └─ WebConfig.java
    └─ WebExceptionHandler.java
└─ talk
  └─ adaptor
    └─ in
      └─ RecommendTalkQueryApi.java
      └─ TalkCommandApi.java
      └─ TalkCreateWebRequest.java
      └─ TalkQueryApi.java
      └─ TalkResponse.java
    └─ out
      └─ CacheUpdateExecutorConfig.java
      └─ RandomRecommendTalkMemory.java
      └─ RandomTalkSampler.java
  └─ application
    └─ TalkCreateService.java
    └─ TalkQueryService.java
    └─ in
      └─ RecommendTalkQueryable.java
      └─ RecommendTalks.java
      └─ TalkCreatable.java
      └─ TalkCreateRequest.java
      └─ TalkQueryRequest.java
      └─ TalkQueryable.java
    └─ out
      └─ RecommendTalkRepository.java
      └─ TalkRepository.java
  └─ domain
    └─ Content.java
    └─ ReactionCount.java
    └─ Talk.java
    └─ Time.java
└─ ui
  └─ UserInterfaceController.java
```

```
└─ resources
    ├── application.yml
    ├── static
    │   ├── css
    │   │   └─ style.css
    │   ├── images
    │   │   ├── quiet-chatter-icon.png
    │   │   ├── quiet-chatter-icon2.png
    │   │   └─ 사용시연2.mp4
    │   └─ js
    │       ├── api.js
    │       └─ domain.js
    └─ templates
        ├── book-detail.html
        ├── book-search-results.html
        ├── fragments
        │   ├── common-assets.html
        │   └─ search.html
        └─ home.html
```