

[| 자유 게시판 | >](#)

첫 pull request 후기



aww zz 챗봇 고수 1:1 채팅

2020.06.28. 05:30 조회 110

댓글 8 URL 복사

그냥 생각나는 김에 써봄

글은 대충 자문자답 Q&A식으로 써볼꺼다.

참 특성상 카톡봇이 main이기 때문에 pull request 가 무엇인지 알려주겠다.

- 의미

내가 작업한 코드가 있으니 내 브랜치를 당겨 검토 후 병합해주세요 (^0^)/

- pull request를 하는 이유

- 자연스러운 코드 리뷰를 위해
- Push 권한이 없는 오픈 소스 프로젝트에 기여할 때
- 콜라보레이터에 소속되어있는 경우에는, 그 저장소에서 브랜치를 따고 푸쉬하면 풀리퀘가 가능하다.
 - + 에서 branch가 보이는 경우는 그 저장소만의 기능이다
- Push로 협업했을때, 다른 사람의 commit을 볼 일이 많지 않고, master branch와 merge할 때 서야 보게 되는데, Pull Request는 당장 merge하지 않는다는 규칙이 Pull Request를 보고, 코드에 신경쓰게 되고 어떤 작업이 언제 적용되었는지 알 수 있다. 오히려 당황스러운 코드충돌을 줄일 수 있다.

가장 마음에 드는 설명이라 갖고옴(출처: velog.io/@zansol)

애초에 git을 안써서 무슨소린지 잘 모르겠다면 의미 정도만 파악해도 이글읽는데 문제없다.

본인은 "Push 권한이 없는 오픈 소스 프로젝트에 기여할때" 라는 목적으로 해봤다.

Q: 뭐 적어서 냈냐?

A: (사진으로 설명)

```
/**
 * Gets comments of specific user in Geometry dash
 * @param accountId this account ID to get comments
 * @param page the page number
 * @return a Mono emitting a paginator containing all comments in user. Note that if
 *         no comments are found, it will emit an empty paginator. (In this case,
 *         fortunately, Geometry Dash API does not return the same response
 *         when an actual error occurs while processing the request).
 */
public Mono<GDPaginator<GDComment>> getUserCommentByAccountId(long accountId, int page){
    return fetch(new GDUserProfileCommentRequest( client: this, accountId, page));
}
```

이 ^^ 사진은 reply받고 이후에 쓴거

```
client / GDLevelCommentRequest

Indexes.java x TestMain.java x AuthenticatedGDClient.java x AbstractGDClient.java x GDLevelCommentRequest.java x
37      params.put("levelId", levelId);
38      params.put("total", "" + 0);
39      params.put("page", "" + page);
40      if(isRecent){
41          params.put("mode", "0");
42      } else {
43          params.put("mode", "1");
44      }
45  }
46
47  @Override
48  @Override
49  GDPaginator<GDComment> parseResponse(String response) throws GDClientException {
50      List<GDComment> commentList = new ArrayList<>();
51      String[] split1 = response.split( regex: "}" );
52      String[] comments = split1[0].split( regex: "\\|");
53      for (String c : comments) {
54          String[] split2 = c.split( regex: ":" );
55          Map<Integer, String> data = ParseUtils.splitToMap(split2[0], regex: "~");
56          Map<Integer, String> u = ParseUtils.splitToMap(split2[1], regex: "~");
57          long commentId = Long.parseLong(Utils.defaultStringIfEmptyOrNull(data.get(Indexes.COMMENT_ID), fallback: "0"));
58          long accountId = Long.parseLong(Utils.defaultStringIfEmptyOrNull(u.get(Indexes.USER_ACCOUNT_ID), fallback: "0"));
59          commentList.add(new GDComment(
60              commentId,
61              Long.parseLong(Utils.defaultStringIfEmptyOrNull(data.get(Indexes.COMMENT_SENDER_ID), fallback: "0")),
62              Utils.defaultStringIfEmptyOrNull(u.get(Indexes.USER_NAME), fallback: "-"),
63              Utils.b64Decode(Utils.defaultStringIfEmptyOrNull(data.get(Indexes.COMMENT_BODY), fallback: "0")),
64              Integer.parseInt(Utils.defaultStringIfEmptyOrNull(data.get(Indexes.COMMENT_LIKES), fallback: "0")),
65              Integer.parseInt(Utils.defaultStringIfEmptyOrNull(data.get(Indexes.COMMENT_UPLOADED_TIMESTAMP), fallback: "NA")),
66              Integer.parseInt(Utils.defaultStringIfEmptyOrNull(data.get(Indexes.COMMENT_PERCENTAGE), fallback: "0")),
67              Integer.parseInt(Utils.defaultStringIfEmptyOrNull(data.get(Indexes.COMMENT_SENDER_ROLE), fallback: "0")),
68              () -> client.fetch(new GDUserProfileDataRequest(client, accountId)));
69      )
70      Tuple3<Integer, Integer, Integer> pageInfo = ParseUtils.extractPageInfo(split1[1]);
71      return new GDPaginator<>(commentList, page, pageInfo.getT3(), pageInfo.getT1(), newPage ->
72          client.fetch(new GDLevelCommentRequest(client, levelId, isRecent, newPage)));
73  }
```

```
Indexes.java x TestMain.java x GDTimelyRequest.java x GDUserRankingRequest.java x AuthenticatedGDClient.java x AbstractGDClient.java x GDLevelCommentRequest.java x
1  package com.github.alex1304.jdash.client;
2
3  import java.util.*;
4
5  import com.github.alex1304.jdash.entity.GDUserSearchData;
6  import com.github.alex1304.jdash.entity.IconType;
7  import com.github.alex1304.jdash.exception.GDClientException;
8  import com.github.alex1304.jdash.util.*;
9
10 class GDUserRankingRequest extends AbstractAuthenticatedGDRequest<List<GDUserSearchData>> {
11
12     private UserRankingStrategy strategy;
13     private final int count;
14
15     GDUserRankingRequest(AuthenticatedGDClient client, UserRankingStrategy strategy, int count) {
16         super(client);
17         this.strategy = Objects.requireNonNull(strategy);
18         this.count = count;
19     }
20
21     @Override
22     public String getPath() { return Routes.USER_SCORES; }
23
24     @Override
25     void putParams(Map<String, String> params) {
26         params.put("type", strategy.getVal());
27         params.put("count", "" + count);
28     }
29
30     @Override
31     List<GDUserSearchData> parseResponse(String response) throws GDClientException {
32         List<GDUserSearchData> ranking = new ArrayList<>();
33         String[] users = response.split( regex: "\\|");
34         for (String u : users) {
35             Map<Integer, String> data = ParseUtils.splitToMap(u, regex: ":");
36         }
37     }
```

Javadoc도 가능하면 써달라고 하셔서 써드렸다.

Q: 또 지메나

A: o

Q: 저거 쓰면서 느낀 생각은?

A: 저 repo원래 제작자분께서 설명없이도 접근하기 쉽게 코드정리를 매우 잘해놓으셔서, 어떻게 써야할지 바로바로 머리에 들어왔고, 작업하는데 매우 편했다.

Q: review 보여줘

A: .

육먹으면 어떡하지 싶었는데 호평이어서 감격이었다. AbstractGDClient쪽은 거의 코드 심장부분이라 본인이 건드리면 x될것 같아서 request만 건드렸는데 클라이언트에 javadoc이랑 추가해 달라셔서 이후에 추가했다. test파일은 본인이 실수로 커밋안해서 클라이언트 손보고 이후에 같이 올렸다.

Q: 6hours 뒤에 답장하는거 보소 인성 개xxx

A: 자다깼다. 양해좀

Q: pull request 하고나면 어떤기분임, 나(글 읽는 nidle) 한테 추천하냐?

A: 글머리에서 언급한 pull request목적에 따라 다르게 느끼겠지만,
일단 하고나면 되게 뿌듯하다. like 10년묵은때 대방출

프로젝트가 얼마나 복잡한지(ex: 가독성, 규모), 얼마나 건드릴지(ex: 버그수정, 파일 추가)에 대해 작업 난이도는 천차 만별이고,

또 본인이 쓴 작업물이 반영이 안될수도(본인리퀘도 아직 검토중임)있고, 혹은 육즙 먹을수도 있긴하다. 예를들면
쓴코드가 원본에 비해 많이 빈약하다, 이거 테스트는 거쳤냐 등등.(너무 부정적인거만 말했나..?)

근데 개인적으로는 한번쯤 해보는거도 나쁘지 않은 시도다. 특히 할짓없을때 하면 2배로 재밌다.
일단 자기가 쓴 작업물이 실제 프로젝트에 반영된다는 것이니, 의의자체가 주는 만족감이 크다.

만약 원제작자가 pull request를 승인하면 해당 repo에 contributors에 자기 프로필이 박제 되는데,
그것도 은근 기분짜진다.

본인은 전문적으로 하는 그런 인간이 아니라서(even 이번이 처음)
본인보다 더잘아는 분들이 있으면 좀 아니다 싶은 부분은 지적좀 해주세요.



aww zz님의 게시글 더보기 >

❤ 좋아요 0 💬 댓글 8

🔗 공유 | 신고

댓글 등록순 최신순 ↻

댓글알림 ☐



윤 위해 웃어줘

조만간 git에 대해서도 다뤄봐야겠네요

2020.06.28. 08:42 답글쓰기



aww zz 작성자

설명 자세히좀 ㄱㄱ

2020.06.28. 11:56 답글쓰기



윤 위해 웃어줘

aww zz 네엠편~

2020.06.29. 13:25 답글쓰기



사로로

PR까지 하다니...

2020.06.28. 08:51 답글쓰기



aww zz 작성자

재밌어요

2020.06.28. 11:56 답글쓰기



XZI

코볼소 대시 당신 의 만드십시오.

2020.06.28. 09:02 답글쓰기



aww zz 작성자

솔직히 귀찮음

2020.06.28. 11:10 답글쓰기



Caffeine

한국다람쥐입니다

2020.06.28. 14:51 답글쓰기

Hibot

댓글을 남겨보세요



등록

글쓰기

답글

목록

▲ TOP



'자유 게시판' 게시판 글

이 게시판 새글 구독하기 ☐

디렉토리 복사가 안되네요 ㅜㅜ [1]	고인물이되고시 퍼요	2020.06.28.
곧 중수 [6]	봇초보입니당	2020.06.28.
첫 pull request후기 🐼 [8]	aww zz	2020.06.28.
여러분의 도움이 필요합니다! [2]	장돌니	2020.06.28.
저 벌써 [3]	봇초보입니당	2020.06.28.

1 2 3

전체보기

이 카페 인기글




간단 계산기 입니다

deadman5
호풍이발 쓰는사람 계정 영정먹이는 방법

Kiri
♡0 💬7

카봇커 채팅방을 그룹으로 만들면 오실분?

성빈
♡1 💬52



(수정5)카카오톡 오픈채팅 공식 봇관련 정
보기

실시간 멜론 차트

카톡봇 사람들은 꼭봐야됨

1 2 3 4 5

버튼 누르면 intent가 함수를 실행하게하려
면

jomin398
♡0 💬18

자동읽음소스

tomohong
♡0 💬9

카톡 정지에 관해서

리스트
♡0 💬15