240722(월) Exception handler

"내가 생각하는 프로그래머의 삶.+x+" 발췌

2인1조로 일하는데 1명은 유럽으로 휴가중이라는 공지

새로운 이슈가 생겼는지 느낌표 알림을 클릭하니

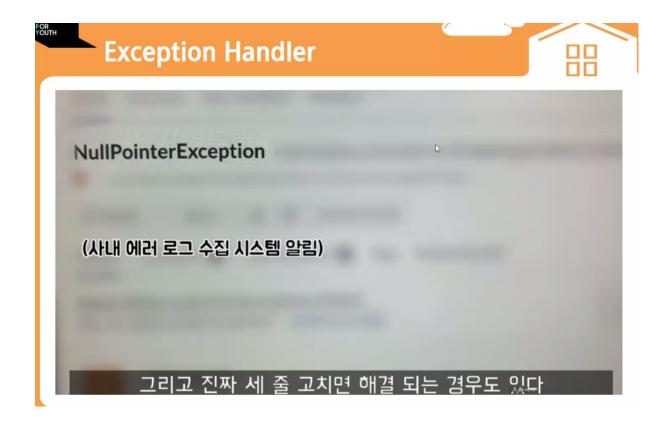
버젼업된 환경이 구형시스템과 새로운 충돌을 일으킨 모양

"내가 생각하는 프로그래머의 삶.+x+" 발췌

간단히 3줄 고쳐주니 잘 돌아감.

회사 내부 시스템은 빌드부터 테스트, 최종 시스템에 로드까지

원클릭 전자동이라 단지 푸시 한번만 해주면 끝.



Exception Handler

- ✓ 현업에서는 모든 이슈를 자동으로 수집한다
- ✓ 개발자가 매번 서버 로그를 뒤져볼 수는 없기 때문
- ✓ 매일 아침 출근하면 살펴보는 것이 바로 이 Exception들
- ✓ 회사에 없는 상태라도 이슈를 바로 확인할 수 있다!

How to use

@ExceptionHandler같은 경우는 @Controller, @RestController가 적용된 Bean내에서 발생하는 예외를 잡아서 하나의 메서드에서 처리해주는 기능을 한다.

```
@RestController
public class MyRestController {
    ...
    ...
    @ExceptionHandler(NullPointerException.class)
    public Object nullex(Exception e) {
        System.err.println(e.getClass());
        return "myService";
    }
}
```



MM이나 Slack과 연동해서 이슈를 받을 수 있다!!

어차피 Controller에서 호출하잖아

- ✓ Controller에서 호출하기 때문에 Exception이 다 잡힌다
- ✓ 그럼 모든 Controller에 모든 Exception class를 다 지정하면
- ✓ 어떤 경우에라도 다 잡아낼 수 있겠네?
- ✓ 근데 그러면 귀찮자나

Controller Advice



@ControllerAdvice

@ExceptionHandler가 하나의 클래스에 대한 것이라면, @ControllerAdvice는 모든 @Controller 즉, 전역에서 발생할 수 있는 예외를 잡아 처리해주는 annotation이다.

```
@RestControllerAdvice
public class MyAdvice {
    @ExceptionHandler(CustomException.class)
    public String custom() {
        return "hello custom";
    }
}
```

전역으로 적용할 수 있는 매직

- ✓ 어떤 경우에라도 다 잡아낼 수 있겠네?
- ✓ 정신 건강을 위해서 ControllerAdvice를 쓰자
- ✓ 그리고 잡아낸 Exception은 어디로?
- ✓ MM으로 쏜다!



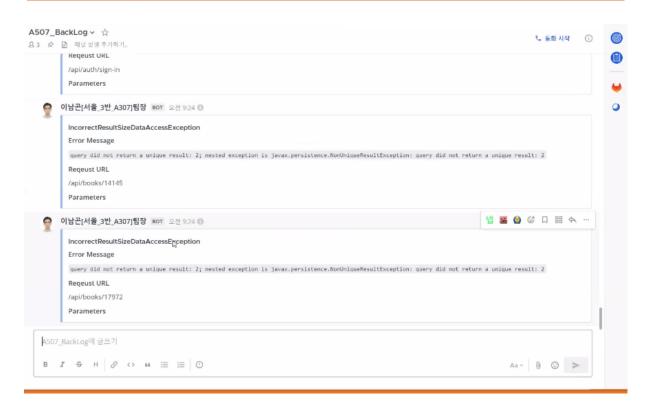
청출어람

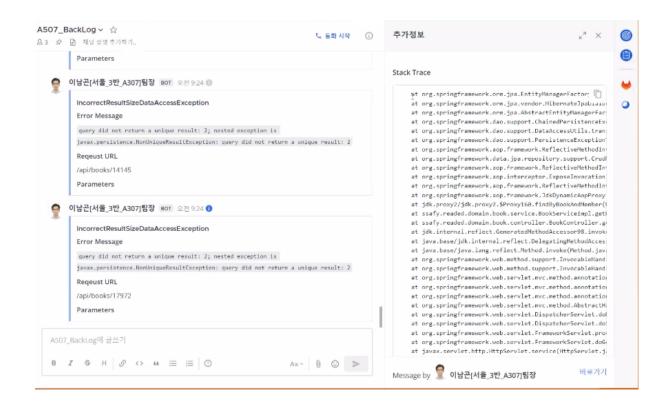


```
@ControllerAdvice
public class ExceptionController {
        @Autowired
        private NotificationManager notificationManager;
        @ExceptionHandler(Exception.class)
        public ResponseEntity exceptionTest(Exception e, HttpServletRequest req) {
               StringBuilder params = new StringBuilder();
               Enumeration<String> keys = req.getParameterNames();
               while (keys.hasMoreElements()) {
                       String key = keys.nextElement();
                       params.append("-").append(key).append(":").append(req.getParameter(key)).append('\n');
               ErrorResponse result = new ErrorResponse();
               e.printStackTrace();
               notificationManager.sendNotification(e, req.getRequestURI(), params.toString());
               result.status = false;
               result.msg = "잘못된 요청입니다!";
               return new ResponseEntity<>(result, HttpStatus.BAD_REQUEST);
       }
}
```

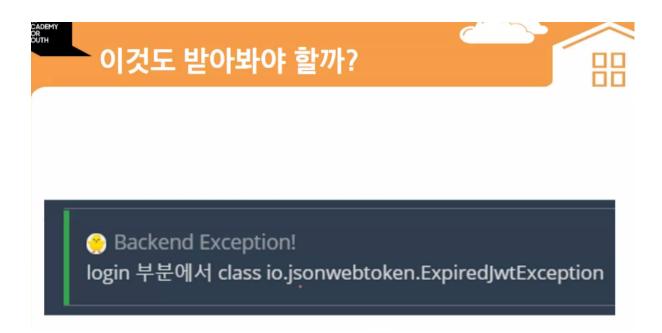
청출어람

```
@Component
@ToString
public class MatterMostSender {
       private Logger log = LoggerFactory.getLogger(MatterMostSender.class);
       private boolean mmEnabled = true;
       private String webhookUrl = "https://meeting.ssafy.com/hooks/ou9oee6tq3ytuyzwnid1gmzkar";
       public void sendMessage(Exception exception, String uri, String params) {
                if (mmEnabled) {
                       try {
                                Attachments attachments = new Attachments(excpetion, uri, params);
                               String payload = new Gson().toJson(attachments);
                               System.out.println(payload);
                               RestTemplate restTemplate = new RestTemplate();
                               HttpHeaders headers = new HttpHeaders();
                               headers.set("Content-type", MediaType.APPLICATION_JSON_VALUE);
                               HttpEntity<String> entity = new HttpEntity<>(payload, headers);
                               restTemplate.postForEntity(webhookUrl, entity, String.class);
                        } catch (Exception e) {
                               log.error("ERROR!! While send MatterMost Message : " + e.getMessage());
                        }
               }
       }
}
```





개발자가 서버까지 들어가보지 않아도 확인할 수 있다는 장점! 어디에 있든 노트북만 끼고 있다면 확인할 수 있다.



이 정도는 개발자가 받아봐야 할 기술적인 이슈는 아니다. 이런거 화들짝 놀라서 보다가 진짜로 관심을 가져야 할 것을 놓칠 수 있다.

이것도 받아봐야 할까?

- ✓ Token의 만료, 로그인 비밀번호가 틀렸을 때!
- ✓ 이것도 개발자가 받아봐야 하는 이슈일까?
- ✓ 하루에 네이버에서 토큰이 만료되는 횟수는?
- ✓ 우리는 관심을 가져야 하는 곳에만 관심을 가져야 한다