

# 자기주도 프로젝트

유지보수를 위한 로깅



## 계절학기

# 유지보수를 위한 로깅

## 이 승 윤 Project consultant

---

- 삼성 청년 SW 아카데미 프로젝트 컨설턴트
- 현대오트오에버 그룹사사업부
- 기업은행 IBK시스템 은행사업부
- Code For Korea
- 공인중개사



# Contents

- I 로깅 What/Why
- II 로깅 How
- III 예시 및 과제 설명
- IV 마치며



# 로깅 What and Why



## ✓ What?

- ✓ 로깅(logging) : 시스템에서 발생한 사건(event)을 기록하는 것
- ✓ 로그(log) : 로깅의 결과물
  - ✓ 유지보수에 있어 **블랙박스**와 같은 역할

## ✓ Why?

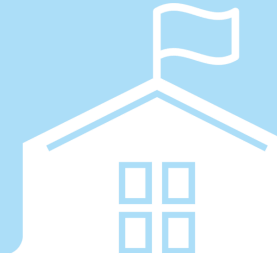
- ✓ 문제를 찾기 위해
- ✓ 원인을 찾기 위해
- ✓ 사용자 행동을 분석하기 위해
- ✓ 이 모든 것은 .. **야근을 방지하기 위하여!**

# 로깅 How?



- ✓ Java : `System.out.println("되라 쫌 " + foo + ", 왜안돼 : " + bar);`
  - ✓ `System.err.println("에러남");`
  - ✓ `ex.printStackTrace();`
- ✓ Python : `print("아무도 날 이해모테", foo, bar)`
- ✓ Javascript : `console.log("===== LSY #1 =====")`
- ✓ ... 이런 건 이제 그만

## 로깅 How? \_계속



- ✓ 개발 언어와 프레임워크에 **알맞은 로깅 방법**을 사용하자.
  - ✓ Spring Boot : Logback, Log4j2
  - ✓ Python(Django, Flask, ...) : Python Logging Module
  - ✓ Node : Winston, Pino, Bunyan
  - ✓ **FE** : 중요 로그는 백엔드로 보내자. 또는 솔루션과 서비스(SaaS)를 적용하자.
- ✓ **로깅 레벨**을 구분하여 사용하자.
  - ✓ FATAL(CRITICAL, SEVERE) > ERROR > WARN > INFO > DEBUG > TRACE
  - ✓ 환경(Env)에 따라 로그 출력을 구분하자.  
상세한 로그 순 : Local(TRACE) > Dev(DEBUG) > Test(INFO) > Prod(WARN | ERROR)

# 과제

- ✓ 관통 프로젝트에 로깅 적용하기
- ✓ 예시) Spring Boot - <https://github.com/spring-guides/tut-rest>
  - ✓ 기본 로그 설정이 되어 있어서 프로그램 시작시 다음과 같은 로그가 출력됨.

```
.   ____          _ _   ____
/\  / ___'  _ _ ( )  _ _  \/\ \
( ( )\___ | ' | ' | | ' \ \ \ \
\ \ ___| | | | | | | | | | | ) ) )
 ' |___| . | | | | | | | \_, / / / /
=====|_|=====|__/\_/_/_/_/

:: Spring Boot ::      (v2.3.0.RELEASE)

2023-06-14 11:17:13 INFO payroll.PayrollApplication - Starting PayrollApplication on ssafy-laptop-sam with PID 19216
(C:\Workspace\intellij\tut-rest\rest\target\classes started by SSAFY in C:\Workspace\intellij\tut-rest)
2023-06-14 11:17:13 DEBUG payroll.PayrollApplication - Running with Spring Boot v2.3.0.RELEASE, Spring v5.2.6.RELEASE
2023-06-14 11:17:13 INFO payroll.PayrollApplication - The following profiles are active: local
2023-06-14 11:17:15 WARN o.s.b.a.o.j.JpaBaseConfiguration$JpaWebConfiguration - spring.jpa.open-in-view is enabled by default.
Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable
this warning
2023-06-14 11:17:15 INFO payroll.PayrollApplication - Started PayrollApplication in 2.203 seconds (JVM running for 2.651)
```

## 과제\_계속

- ✓ 예시) Spring Boot - <https://github.com/spring-guides/tut-rest>
- ✓ (나쁜 예시) 이렇게 해도 출력은 됩니다.

```
@GetMapping("/employees")  
CollectionModel<EntityModel<Employee>> all() {  
    System.out.println("안녕 난 로그야");  
    System.err.println("안녕 난 에러야");  
}
```

```
2022-12-22 10:28:19.057 INFO 2852 ---  
    Completed initialization in 3 ms  
안녕 난 로그야  
안녕 난 에러야
```



## 과제\_계속

✓ 예시) Spring Boot - <https://github.com/spring-guides/tut-rest>

✓ 올바른 로깅 방법

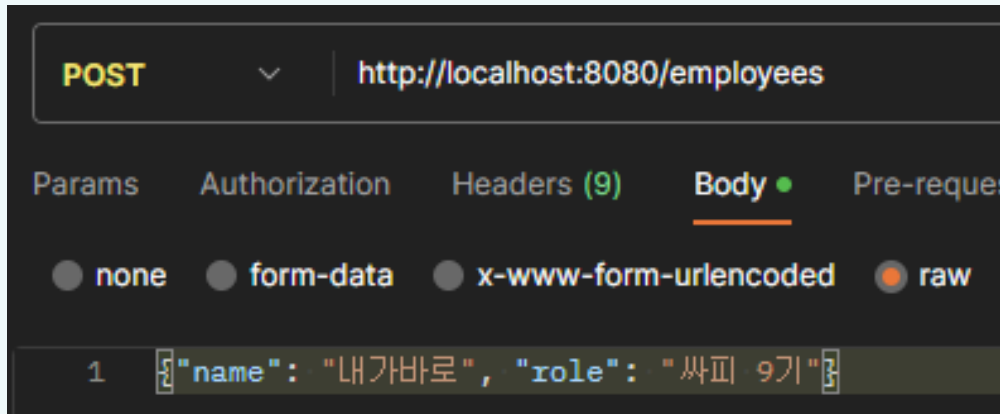
```
@PostMapping("/employees")
Employee newEmployee(@RequestBody Employee newEmployee) {
    logger.trace("newEmployee() : " + String.valueOf(newEmployee));
    return repository.save(newEmployee);
}
```

```
application.yml
1  spring:
2    profiles: local
3
4  logging:
5    level:
6      root: WARN
7      payroll: TRACE
```

## 과제\_계속

✓ 예시) Spring Boot - <https://github.com/spring-guides/tut-rest>

✓ 올바른 로깅 방법



```
2023-06-14 13:38:25 [http-nio-8080-exec-1] TRACE payroll.EmployeeController - newEmployee() :  
Employee{id=null, name='내가바로', role='싸피 9기'}
```

## 과제\_계속

- ✓ 예시) Spring Boot - <https://github.com/spring-guides/tut-rest>
- ✓ 올바른 로깅 방법(파일과 콘솔로 출력)

```
2023-06-14 13:38:25 [http-nio-8080-exec-1] TRACE payroll.EmployeeController - newEmployee() :  
    Employee{id=null, name='내가바로', role='싸피 9기'}  
2023-06-14 13:41:21 [http-nio-8080-exec-5] TRACE payroll.EmployeeController - newEmployee() :  
    Employee{id=null, name='바로내가', role='싸피 9기'}  
2023-06-14 13:41:26 [http-nio-8080-exec-6] TRACE payroll.EmployeeController - newEmployee() :  
    Employee{id=null, name='우리모두', role='싸피 9기'}
```

```
ation.yml × tut-rest.log × DefaultLogbackConfiguration.class × Employee.java ×  
2023-06-14 11:17:13 [main] INFO payroll.PayrollApplication - Start  
    ssafy-laptop-sam with PID 19216 (C:\Workspace\intelliJ\tut-rest\r  
    in C:\Workspace\intelliJ\tut-rest)  
2023-06-14 11:17:13 [main] DEBUG payroll.PayrollApplication - Runn  
    Spring v5.2.6.RELEASE  
2023-06-14 11:17:13 [main] INFO payroll.PayrollApplication - The  
2023-06-14 11:17:15 [main] WARN o.s.b.a.o.j.JpaBaseConfigurationS
```

## 과제\_계속

- ✓ 예시) Spring Boot - <https://github.com/spring-guides/tut-rest>
- ✓ 예외 핸들러(AOP)에 적용

```
71 2023-06-14 13:42:11 [http-nio-8080-exec-7] WARN payroll.EmployeeNotFoundAdvice - payroll
    .EmployeeNotFoundAdvice payroll.EmployeeNotFoundException: Could not find employee 10 payroll
    .EmployeeController.lambda$one$1(EmployeeController.java:65)
72 java.base/java.util.Optional.orElseThrow(Optional.java:403)
73 payroll.EmployeeController.one(EmployeeController.java:65)
74 java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:104)
75 java.base/java.lang.reflect.Method.invoke(Method.java:577)
76 org.springframework.web.method.support.InvocableHandlerMethod.doInvoke(InvocableHandlerMethod.java:190)
```

```
@ResponseBody
@ExceptionHandler(EmployeeNotFoundException.class)
@ResponseStatus(HttpStatus.NOT_FOUND)
String employeeNotFoundHandler(EmployeeNotFoundException ex) {
    log.warn(this.getClass().getCanonicalName() + " " + ex.toString())
    return ex.getMessage();
}
```

## 마치며..

- ✓ 문제가 생겼을 때는 **로그부터** 확인하는 습관을 갖자!
  - ✓ 교육생 : 코치님! 이거 안되요!
  - ✓ 코치 : 로그 보셨어요?
  - ✓ 교육생 : 예? ...
  - ✓ 코치 : ...
- ✓ 알맞은 로깅 방법을 사용하자
- ✓ 문제가 생길 수 있는 부분, 실행 상태를 확인하고 싶은 곳에 로그를 남기자!



