

[내부 표준] 로그 데이터 처리 및 코드 가이드 v1.0

1. 개요

본 문서는

사내 시스템에서 발생하는 로그 데이터 처리 방식과 관련된 표준 코드 작성 가이드를 정의한다.

모든 로그 데이터는 추후 분석 및 감사 대응을 위해 일관된 형식과 전처리 규칙을 따라야 한다.

2. 로그 데이터 전처리 기준

2.1 기본 전처리 원칙

로그 데이터는 다음 순서로 전처리한다.

1. 결측값 처리

- 로그 메시지가 없는 경우 "UNKNOWN" 으로 치환한다.

2. 시간 데이터 표준화

- 모든 시간은 YYYY-MM-DD HH:MM:SS 형식으로 변환한다.

3. 불필요한 특수문자 제거

- 줄바꿈 문자(`\n`), 탭(`\t`)은 제거한다.

4. 로그 레벨 정규화

- INFO, WARN, ERROR 중 하나로 통일한다.

2.2 전처리 예시

전처리 전 데이터 예시:

```
[2024/01/02-10:33] err!! connection failed
```

전처리 후 데이터 예시:

```
2024-01-0210:33:00ERROR connection failed
```

3. 로그 처리 함수 설명

3.1 함수 개요

다음 함수는 로그 메시지를 전처리한 뒤 표준 출력으로 기록하는 역할을 한다.

3.2 표준 로그 처리 함수 예시

```
def process_log(timestamp, level, message):  
    """  
    내부 로그 표준에 맞게 로그를 전처리하고 출력하는 함수  
    """  
  
    if not message:  
        message = "UNKNOWN"  
  
    level = level.upper()  
    if level not in ["INFO", "WARN", "ERROR"]:  
        level = "INFO"  
  
    formatted_time = timestamp.replace("/", "-")  
    log_line = f"{formatted_time} {level} {message}"  
  
    print(log_line)
```

3.3 함수 역할 요약

- 입력된 로그 데이터를 내부 표준 형식으로 변환
- 로그 레벨을 사내 기준에 맞게 정규화
- 결측 메시지를 자동 보정
- 전처리된 로그를 출력 또는 저장 가능

4. 코드 작성 시 유의사항

- 로그 처리 함수에는 반드시 **예외 처리를 포함**해야 한다.
 - 전처리 규칙은 하드코딩하지 말고 **함수 내부에서 일관되게 적용**한다.
 - 사내 표준을 벗어난 로그 형식은 허용하지 않는다.
-