Assignment_02

March 9, 2025

Name - Shravani Sandeep Raut

SE / 48

You are given a large log file containing various system events. Each line in the log file follows this format: [YYYY-MM-DD HH:MM:SS] [LOG_LEVEL] [MODULE] Message where: • YYYY-MM-DD HH:MM:SS is a timestamp. • LOG_LEVEL can be INFO, WARN, ERROR, or DEBUG. • MODULE represents the system module name (alphanumeric, can contain underscores). • Message is the actual log message (it may contain any characters).

Your Task Write a function extract_critical_errors(log_data: str) -> list[tuple] that takes a multiline string log_data (containing log entries) and returns a list of tuples containing: 1. The timestamp 2. The module name 3. The error message

BUT only if: • The LOG_LEVEL is ERROR. • The message contains at least one IP address in IPv4 format (xxx.xxx.xxx.xxx, where xxx is in the range 0-255). • The message contains a hexadecimal error code, formatted as 0x followed by exactly 8 hexadecimal digits (0-9, A-F).

Example Input [2025-02-10 14:23:01] [INFO] [Auth_Module] User login successful. [2025-02-10 15:45:32] [ERROR] [Net_Module] Connection timeout from 192.168.1.10. Error Code: 0xAB12CD34 [2025-02-10 16:01:10] [WARN] [Disk_Module] Low disk space warning. [2025-02-10 17:12:05] [ERROR] [Security_Module] Unauthorized access detected from 10.0.0.5. Error Code: 0xDEADBEEF

Expected Output [('2025-02-10 15:45:32', 'Net_Module', 'Connection timeout from 192.168.1.10. Error Code: 0xAB12CD34'), ('2025-02-10 17:12:05', 'Security_Module', 'Unauthorized access detected from 10.0.0.5. Error Code: 0xDEADBEEF')]

Constraints • Your function must use one single regex pattern to extract the required information. • You cannot use multiple regex calls; the full extraction must be done in one pass using re.findall() or re.finditer(). • Assume log_data contains multiple lines. • Make your regex IP-matching strict, ensuring that invalid IPs (e.g., 256.100.10.10) are not mistakenly matched. (Optional)

[24]:
```
import re

def extract_critical_errors(log_data: str) -> list[tuple]:
    pattern = r"\[(\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2})\] \[ERROR\] \[(\w+)\] (.
 ↪*?(?:\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}).*?0x[A-Fa-f0-9]{8})"
    matches = re.findall(pattern, log_data, re.MULTILINE)
    return matches
```

```
log_data = """[2025-02-10 14:23:01] [INFO] [Auth_Module] User login successful.
[2025-02-10 15:45:32] [ERROR] [Net_Module] Connection timeout from 192.168.1.10.
 ↪ Error Code: 0xAB12CD34
[2025-02-10 16:01:10] [WARN] [Disk_Module] Low disk space warning.
[2025-02-10 17:12:05] [ERROR] [Security_Module] Unauthorized access detected␣
 ↪from 10.0.0.5. Error Code: 0xDEADBEEF"""

print(extract_critical_errors(log_data))
```

```
[('2025-02-10 15:45:32', 'Net_Module', 'Connection timeout from 192.168.1.10.
Error Code: 0xAB12CD34'), ('2025-02-10 17:12:05', 'Security_Module',
'Unauthorized access detected from 10.0.0.5. Error Code: 0xDEADBEEF')]
```