SE 333 Final Project Report

Omar Delgado

Course: SE 333

Date: 11/16/25
Project: Fix DateUtils parsing defects + expose a local MCP test server

1) Summary

I fixed parsing failures in org.apache.commons.lang3.time.DateUtils and added a small HTTP "MCP" service to run Maven tests programmatically. The key issues were (a) localized weekday abbreviations (e.g., German Mi for Wednesday) not being recognized and (b) fragile handling of RFC-style time zone offsets when patterns used ZZ. I refactored the public parse methods to delegate to a single core method that applies locale-aware parsing and normalizes ZZ offsets. All unit tests in DateUtilsTest now pass, and the MCP service exposes /healthz, /echo, and /maven/test endpoints to verify the build.

2) Environment & Project Layout

- **OS:** Windows 11 (x64)
- **Java:** 17.0.16 (Eclipse Adoptium) **Maven:** 3.9.11
- **Python (venv):** 3.14.0 with FastAPI + Uvicorn
- **Paths**
    - Project root: ...\final_project_se333_omar
    - Maven module: .\codebase\pom.xml
    - Fixed class: .\codebase\src\main\java\org\apache\commons\lang3\time\DateUtils.java
    - MCP server: .\mcp\server.py

3) Bugs, Root Causes, and Fix

Bug Root causes

1. **Locale weekdays:** SimpleDateFormat wasn't consistently created with the provided Locale, so localized weekday abbreviations (e.g., Mi) weren't matched reliably.
2. **ZZ time zones:** Inputs sometimes used ±HH:mm while patterns expected ±HHmm (or vice versa). Without normalization, the pattern and input didn't align.

Fix overview

- Kept the four public methods unchanged in signature and semantics:
    - parseDate(str, patterns...)
    - parseDate(str, locale, patterns...)
    - parseDateStrictly(str, patterns...)
    - parseDateStrictly(str, locale, patterns...)
- Added one core implementation:
    - parseDateWithLeniency(str, locale, patterns[], lenient)
        - For each pattern:
            - If pattern ends with ZZ, convert pattern suffix ZZ → Z and normalize the **input** trailing numeric offset ±HHmm ↔ ±HH:mm so it matches the adjusted pattern.
            - Create SimpleDateFormat with the specified Locale (or default), set setLenient(lenient).
            - Parse using ParsePosition and accept only if the **entire** input is consumed.
        - If none match: throw ParseException.
- **Cleanup:** Removed temporary helpers added during debugging; the final class only includes the four public methods plus the single private core method.

Result: All DateUtilsTest cases pass locally.

4) How to Build and Verify

4.1 Run Maven tests (project root)

- cd "<...>\final_project_se333_omar"
- mvn -f .\codebase\pom.xml "-Dtest=org.apache.commons.lang3.time.DateUtilsTest" test
- # Optional: run all tests
- # mvn -f .\codebase\pom.xml test

Expect: BUILD SUCCESS and all DateUtilsTest passing.

4.2 Start the MCP HTTP server (second feature)

- cd ".\mcp"
- .\.venv\Scripts\Activate.ps1
- .\.venv\Scripts\python.exe -m uvicorn server:app --host 127.0.0.1 --port 3988

Expect: Uvicorn running on http://127.0.0.1:3988
(If the port is busy, pick another, e.g., --port 3999.)

4.3 Call the endpoints (new terminal)

```
# Health
Invoke-RestMethod http://127.0.0.1:3988/healthz    # → True

# Echo
Invoke-RestMethod -Method Post -Uri http://127.0.0.1:3988/echo `
  -ContentType 'application/json' -Body '{"text":"hello MCP"}'
# → "hello MCP"

# Run a specific Maven test
Invoke-RestMethod -Method Post -Uri http://127.0.0.1:3988/maven/test `
  -ContentType 'application/json' `
  -Body '{"test_filter":"org.apache.commons.lang3.time.DateUtilsTest#testLANG799_DE_OK"}'
# → JSON with command, cwd, returncode=0, and stdout_tail: BUILD SUCCESS
```

5) Evidence Collected

- **Terminal output** showing BUILD SUCCESS for DateUtilsTest.
- **HTTP responses** for:
    - /healthz → True
    - /echo → hello MCP
    - /maven/test → JSON including returncode: 0 and Surefire summary
      (Place these as screenshots or .json exports in your submission folder.)

6) Conclusion

The project goals are met:

- DateUtils now correctly parses localized weekdays and RFC offsets under ZZ patterns.
- The entire DateUtilsTest suite passes.
- A working MCP HTTP service demonstrates automated test execution via simple REST calls.