

Debugging Notes Template

A debugging notes template captures one bug investigation as you run it, so the work stays reproducible instead of living in your head and scrolling off the terminal. You write down the symptom, the exact way to trigger it, the hypotheses you considered, the one the evidence confirmed and how, the fix, and the test that locks it. The single output is a confirmed cause backed by evidence, recorded so the next person can repeat it. It earns its place the moment a bug is intermittent, comes back, or has to be handed off.

Observe the facts

Bug title

What to capture: A one-line summary of the symptom and where it appears, written factually.

Worked example: `parseDate` throws on partner API timestamps with a timezone offset

Symptom

What to capture: The observed behavior, expected versus actual, with no guessed cause.

Worked example: Expected a Date, got an exception; the daily export shows the wrong hour

Reproduction

What to capture: The exact input, command, and conditions, plus consistent or intermittent.

Worked example: `parseDate("2026-03-14T10:00+02:00")`; consistent, only on `+HH:MM` offsets

Environment

What to capture: Runtime and versions, OS, key dependency versions, branch or commit.

Worked example: Node 22.3, macOS 15, no new deps, branch `fix/parse-offset` at `a4f1c92`

Investigate and confirm

Hypotheses

What to capture: The likely causes ranked most likely first, each a testable claim.

Worked example: 1) regex misses `+HH:MM` 2) wrong offset sign 3) bad upstream trim

Evidence

What to capture: The single check run for the top hypothesis and exactly what it showed.

Worked example: Logged `offsetMinutes` before use; printed `NaN`, so the regex never matched

Confirmed cause

What to capture: The hypothesis the evidence supported and the ones it ruled out.

Worked example: Hypothesis 1; a sign error would still give a number, so #2 is out

Fix, lock, clean up

Fix

What to capture: The smallest change addressing the cause, by file and a one-line description.

Worked example: `src/utils/date.ts`: extend regex to `+HH:MM`, convert to signed minutes (~5 lines)

Test

What to capture: The regression test that fails before and passes after, and where it lives.

Worked example: `tests/date.test.ts` asserts the UTC result; red on old code, green on new

Cleanup and notes

What to capture: Instrumentation removed, the assistant prompt that found it, follow-ups.

Worked example: Log line removed; prompt: rank 3 causes then give the smallest probe
