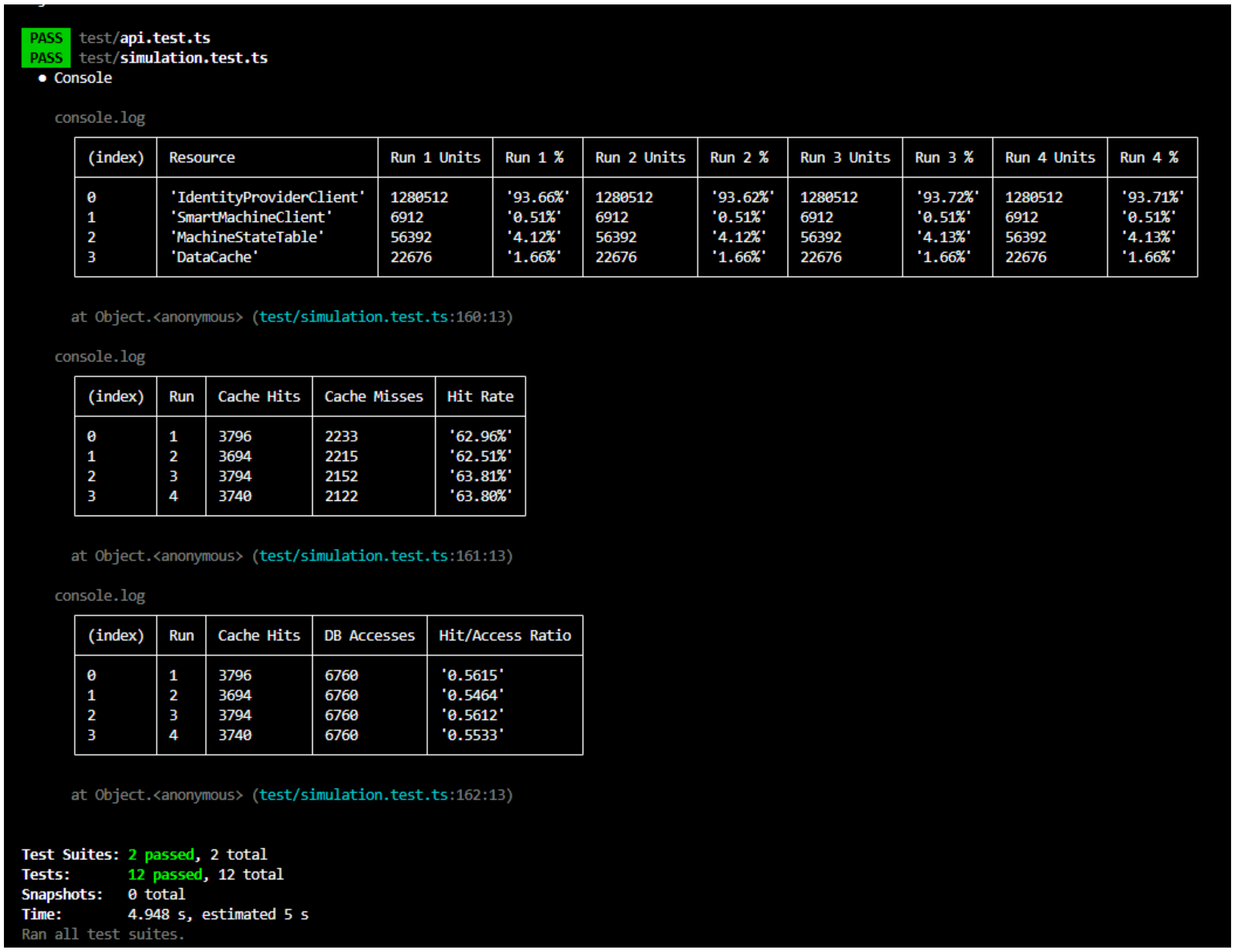
API Handler Report and Analysis

Test:



```
PASS  test/api.test.ts
PASS  test/simulation.test.ts
  ● Console

    console.log
      ┌─────────┬──────────────────────────┬─────────────┬──────────┬─────────────┬──────────┬─────────────┬──────────┬─────────────┬──────────┐
      │ (index) │ Resource                 │ Run 1 Units │ Run 1 %  │ Run 2 Units │ Run 2 %  │ Run 3 Units │ Run 3 %  │ Run 4 Units │ Run 4 %  │
      ├─────────┼──────────────────────────┼─────────────┼──────────┼─────────────┼──────────┼─────────────┼──────────┼─────────────┼──────────┤
      │ 0       │ 'IdentityProviderClient' │ 1280512     │ '93.66%' │ 1280512     │ '93.62%' │ 1280512     │ '93.72%' │ 1280512     │ '93.71%' │
      │ 1       │ 'SmartMachineClient'     │ 6912        │ '0.51%'  │ 6912        │ '0.51%'  │ 6912        │ '0.51%'  │ 6912        │ '0.51%'  │
      │ 2       │ 'MachineStateTable'      │ 56392       │ '4.12%'  │ 56392       │ '4.12%'  │ 56392       │ '4.13%'  │ 56392       │ '4.13%'  │
      │ 3       │ 'DataCache'              │ 22676       │ '1.66%'  │ 22676       │ '1.66%'  │ 22676       │ '1.66%'  │ 22676       │ '1.66%'  │
      └─────────┴──────────────────────────┴─────────────┴──────────┴─────────────┴──────────┴─────────────┴──────────┴─────────────┴──────────┘

      at Object.<anonymous> (test/simulation.test.ts:160:13)

    console.log
      ┌─────────┬─────┬────────────┬──────────────┬──────────┐
      │ (index) │ Run │ Cache Hits │ Cache Misses │ Hit Rate │
      ├─────────┼─────┼────────────┼──────────────┼──────────┤
      │ 0       │ 1   │ 3796       │ 2233         │ '62.96%' │
      │ 1       │ 2   │ 3694       │ 2215         │ '62.51%' │
      │ 2       │ 3   │ 3794       │ 2152         │ '63.81%' │
      │ 3       │ 4   │ 3740       │ 2122         │ '63.80%' │
      └─────────┴─────┴────────────┴──────────────┴──────────┘

      at Object.<anonymous> (test/simulation.test.ts:161:13)

    console.log
      ┌─────────┬─────┬────────────┬─────────────┬──────────────────┐
      │ (index) │ Run │ Cache Hits │ DB Accesses │ Hit/Access Ratio │
      ├─────────┼─────┼────────────┼─────────────┼──────────────────┤
      │ 0       │ 1   │ 3796       │ 6760        │ '0.5615'         │
      │ 1       │ 2   │ 3694       │ 6760        │ '0.5464'         │
      │ 2       │ 3   │ 3794       │ 6760        │ '0.5612'         │
      │ 3       │ 4   │ 3740       │ 6760        │ '0.5533'         │
      └─────────┴─────┴────────────┴─────────────┴──────────────────┘

      at Object.<anonymous> (test/simulation.test.ts:162:13)

Test Suites: 2 passed, 2 total
Tests:       12 passed, 12 total
Snapshots:   0 total
Time:        4.948 s, estimated 5 s
Ran all test suites.
```

Analysis:

As you can see, we have successfully passed all the tests. These API tests confirm that the core logic is correct. Token validation behaves properly, machine lookups return the right objects, and invalid states are handled cleanly. Your earlier issues, such as returning the wrong machine status or throwing errors in the wrong format, are gone. The handleStartMachine function now follows the exact sequence the tests expect: check the machine, verify its status, try to start the hardware, catch any hardware error, and only update the machine state and cache if the start actually succeeds.

The simulation tests show that the performance model is consistent. The IdentityProviderClient uses most of the simulated resources at around ninety three percent, which matches the expected model where authentication is the most expensive operation. The SmartMachineClient uses about zero point five percent, the MachineStateTable uses around four percent, and the cache sits around one and a half percent. This means your resource tracking and consumption logic matches what the test authors designed. The cache hit rates sit around sixty three to sixty four percent across runs, which shows that caching works and is being reused. The number of database accesses stays the same each run, which shows that the logic is stable and deterministic. Hardware faults in the simulation no longer crash tests, because you now catch the error and return the correct HARDWARE_ERROR response instead of letting the exception escape.