Test Report

Run ID: N/A • Generated: 2026-01-20 06:16:52 • Duration: 78ms
Plugin: v0.2.0 (a03dbe622cdc018f89b74731aed91adf1a582867) [dirty]
Repo: v0.1.0 (a03dbe622cdc018f89b74731aed91adf1a582867) [dirty]
LLM: ollama / llama3.2 (complete context, 7 annotated)
Token Usage: 85 input, 57 output (Total: 142)

100.0%
Total Coverage

7	4	1	1
TOTAL TESTS	PASSED	FAILED	SKIPPED

1	0	0
XFAILED	XPASSED	ERRORS

Source Coverage Per Test Details Failures Only

Source Coverage

FILE	STMTS	MISS	COVER	%	COVERED LINES	MISSED LINES
example_pkg/calculator.py	6	0	6	100.0%	1, 3, 6, 8-10	-

Per Test Details

📄 **tests/test_calculator.py** 7 tests

PASSED tests/test_calculator.py::test_add_basic 0ms 1

AI ASSESSMENT

Scenario: Test: tests/test_calculator.py::test_add_basic

Why Needed: Verifies the basic addition functionality of the calculator.

Key Assertions:

- assert add(1, 2) == 3

Confidence: 80%

COVERAGE

example_pkg/calculator.py	1 lines (ranges: 3)

PASSED tests/test_calculator.py::test_add_negative 0ms 1

AI ASSESSMENT

Scenario: Test: tests/test_calculator.py::test_add_negative

Why Needed: To verify the functionality of adding two negative numbers.

Key Assertions:

- {'description': 'The function should add two negative numbers and return a negative result.', 'expected_result': '-2', 'actual_result': None, 'result': None}

Confidence: 80%

COVERAGE

example_pkg/calculator.py	1 lines (ranges: 3)

PASSED `tests/test_calculator.py::test_divide_success` 0ms 1

AI ASSESSMENT

Scenario: Test: tests/test_calculator.py::test_divide_success

Why Needed: To ensure the division function works correctly and raises an error when attempting to divide by zero.

Key Assertions:

- {'assertion_type': 'equality', 'expected_result': 5, 'actual_result': None, 'test_case': 'divide(10, 2)'}

Confidence: 80%

COVERAGE

example_pkg/calculator.py	2 lines (ranges: 8, 10)

PASSED `tests/test_calculator.py::test_divide_zero` 0ms 1

AI ASSESSMENT

Scenario: Test: tests/test_calculator.py::test_divide_zero

Why Needed: To ensure the calculator module handles division by zero correctly and raises a ValueError as expected.

Key Assertions:

- {'assertion_type': 'raises_error', 'error_message': 'Cannot divide by zero', 'function_name': 'divide'}

Confidence: 80%

COVERAGE

example_pkg/calculator.py	2 lines (ranges: 8-9)

FAILED tests/test_calculator.py::test_failure_demo 1ms 1

ERROR

```
    assert 2 == 3
```

AI ASSESSMENT

Scenario: Test failure demo

Why Needed: To demonstrate error reporting when a test intentionally fails.

Key Assertions:

- {'test_name': 'test_failure_demo', 'expected_result': 'AssertionError: 3 != 2', 'actual_result': 'None'}

Confidence: 80%

COVERAGE

example_pkg/calculator.py	1 lines (ranges: 3)

SKIPPED tests/test_calculator.py::test_skip_demo 0ms

ERROR

```
('/mnt/hbmon/pytest-llm-report/examples/with-ollama/tests/test_calculator.py', 31, 'Skipped: Demonstrating skipped test reporting')
```

AI ASSESSMENT

Scenario: tests/test_calculator.py::test_skip_demo

Why Needed: This test is intentionally skipped for demonstration purposes.

Key Assertions:

- The test should fail with a message indicating it's being skipped.

Confidence: 80%

Tokens: 85 input + 57 output = 142 total

XFAILED tests/test_calculator.py::test_xfail_demo 0ms

ERROR

```
@pytest.mark.xfail(reason="Demonstrating xfailed test reporting")
    def test_xfail_demo():
        """A test that is expected to fail."""
>       assert 1 / 0 == 1
               ^^^^^
E       ZeroD
```

AI ASSESSMENT

Scenario: test_xfail_demo

Why Needed: To demonstrate how xfail can be used to mark a test as expected to fail.

Key Assertions:

- assert 1 / 0 == 1

Confidence: 80%