

Follow @jestjs_

☆ Star



Jest is a delightful JavaScript Testing Framework with a focus on simplicity.

It works with projects using: [Babel](#), [TypeScript](#), [Node](#), [React](#), [Angular](#), [Vue](#) and more!

zero config

Jest aims to work out of the box, config free, on most JavaScript projects.

snapshots

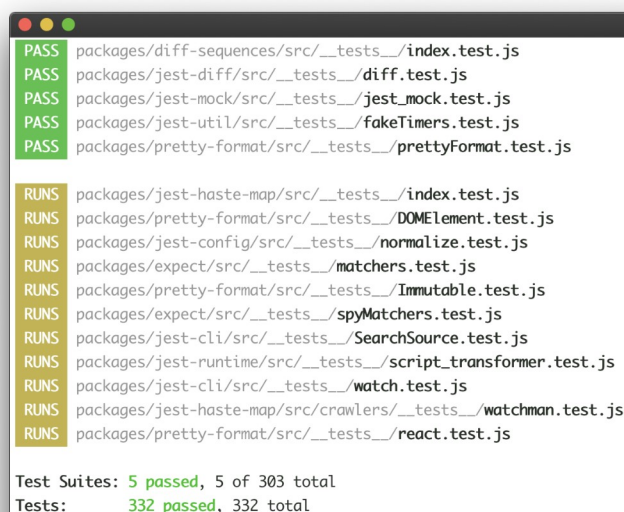
Make tests which keep track of large objects with ease. Snapshots live either alongside your tests, or embedded inline.

isolated

Tests are parallelized by running them in their own processes to maximize performance.

great api

From `it` to `expect` - Jest has the entire toolkit in one place. Well documented, well maintained, well good.

A terminal window showing the output of a Jest test run. The output lists 15 test files, each preceded by a status indicator: 'PASS' for the first five and 'RUNS' for the remaining ten. The files are located in various subdirectories of a 'packages' folder. At the bottom, a summary line states 'Test Suites: 5 passed, 5 of 303 total' and another line states 'Tests: 332 passed, 332 total'.

```
PASS packages/diff-sequences/src/__tests__/index.test.js
PASS packages/jest-diff/src/__tests__/diff.test.js
PASS packages/jest-mock/src/__tests__/jest_mock.test.js
PASS packages/jest-util/src/__tests__/fakeTimers.test.js
PASS packages/pretty-format/src/__tests__/prettyFormat.test.js

RUNS packages/jest-haste-map/src/__tests__/index.test.js
RUNS packages/pretty-format/src/__tests__/DOMElement.test.js
RUNS packages/jest-config/src/__tests__/normalize.test.js
RUNS packages/expect/src/__tests__/matchers.test.js
RUNS packages/pretty-format/src/__tests__/Immutable.test.js
RUNS packages/expect/src/__tests__/spyMatchers.test.js
RUNS packages/jest-cli/src/__tests__/SearchSource.test.js
RUNS packages/jest-runtime/src/__tests__/script_transformer.test.js
RUNS packages/jest-cli/src/__tests__/watch.test.js
RUNS packages/jest-haste-map/src/crawlers/__tests__/watchman.test.js
RUNS packages/pretty-format/src/__tests__/react.test.js

Test Suites: 5 passed, 5 of 303 total
Tests:       332 passed, 332 total
```

FAST AND SAFE

By ensuring your tests have unique global state, Jest can reliably run tests in parallel. To make things quick, Jest runs previously failed tests first and re-organizes runs based on how long test files take.

CODE COVERAGE

Generate code coverage by adding the flag `--coverage`. No additional setup needed. Jest can collect code coverage information from entire projects, including untested files.

```
~/d/p/j/j/e/timer $ yarn jest --coverage
yarn run v1.12.3
$ /Users/ortatherox/dev/projects/jest/jest/examples/timer/node_modules/.bin/jest --coverage
PASS __tests__/timer_game.test.js
PASS __tests__/infinite_timer_game.test.js
-----|-----|-----|-----|-----|
File      | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s |
-----|-----|-----|-----|-----|
All files |    87.5 |    100   |     80   |    87.5 |                   |
infiniteTimerGame.js |     80   |    100   |    66.67 |     80   |          12       |
timerGame.js |    100   |    100   |    100   |    100   |                   |
-----|-----|-----|-----|-----|

Test Suites: 2 passed, 2 total
Tests:       4 passed, 4 total
Snapshots:  0 total
Time:        0.878s, estimated 1s
Ran all test suites.
```

```
• toHaveBeenCalledWith two arguments

expect(jest.fn()).toHaveBeenCalledWith(...expected)

Expected: "last", 10
Received:
  1: "first", 1
  2: "second", 2
  3: "third", 3

Number of calls: 9
```

EASY MOCKING

Jest uses a custom resolver for imports in your tests, making it simple to mock any object outside of your test's scope. You can use mocked imports with the rich [Mock Functions](#) API to spy on function calls with readable test syntax.

GREAT EXCEPTIONS

Tests fail—when they do, Jest provides rich context why. Here are some examples:

toBe**toBeCloseTo****toEqual****toStrictEqual****toHaveProperty****toMatchSnapshot****toThrowError**

```
• toBe multi-line strings

expect(received).toBe(expected) // Object.is equality

- Expected   - 1
+ Received   + 1

  Roses are red. Violets are blue.
- Testing with Jest is good for you.
+ Testing your luck is bad for you.
```

PHILOSOPHY

Jest is a JavaScript testing framework designed to ensure correctness of any JavaScript codebase. It allows you to write tests with an approachable, familiar and feature-rich API that gives you results quickly.

Jest is well-documented, requires little configuration and can be extended to match your requirements.

Jest makes testing delightful.

– Jest Core Team



DOCS AND TALKS

The Jest core team and contributors regularly speak about [Jest and Delightful JavaScript Testing](#). Check out our talk about [Building High-Quality JavaScript Tools](#) at jsconf.eu 2017 and our talk about [Jest as a Platform](#) at ReactiveConf 2017.

OPEN COLLECTIVE

Jest uses Open Collective to support developers contributing to Jest.

featured sponsors



Join [60+ donors](#) who sponsor Jest for \$3 or more per month on opencollective.com.

WHO USES JEST?

A lot of people! With [300+ million](#) downloads in the last month, and used on over [11,000,000](#) public repos on GitHub.

Jest is used extensively at these companies:

