

Jest as a Platform

ROGELIO GUZMAN
Frontend Engineer · Docker

 rogeliog



@rogeliog



jest-core



Jest as a Platform

Jest as a Platform

JavaScript testing solution

Easy setup

Instant feedback

Works out of the box

Jest as a Platform

JavaScript testing solution

Easy setup

Instant feedback

Works out of the box

Extensible

Works with existing tooling

Build on top

Jest as a Platform

JavaScript testing solution

Easy setup

Instant feedback

Works out of the box

Extensible

Works with existing tooling

Build on top

I ❤️ testing

Code coverage

Watch mode

Test framework

Test Runner

Reporters

I ❤️ testing
I don't test

Assertion library

Mocking library

Test Utilities

Test Runner

Code coverage

Reporters

Test framework

Test Utilities

Assertion library

Mocking library

I ❤️ idea of testing

Test Runner

Code coverage

Test framework



Assertion library

Watch mode

Reporters

Mocking library

Test Utilities

Code coverage

Watch mode

Test framework

Test Runner

Reporters

more effort → less tests

less effort → more tests

Assertion library

Mocking library

Test Utilities



Code coverage

Test Runner

Assertion library

Test framework

Watch mode

Mocking library

Reporters



Jest

-  6x faster
-  Instant feedback
-  Better development experience

Let's setup Jest



rogeliog/jest-as-a-platform-talk

Emoji Cinema

Start typing to see emojis!



rogeliog/jest-as-a-platform-talk

Emoji Cinema

Frozen|



Frozen



rogeliog/jest-as-a-platform-talk

Emoji Cinema

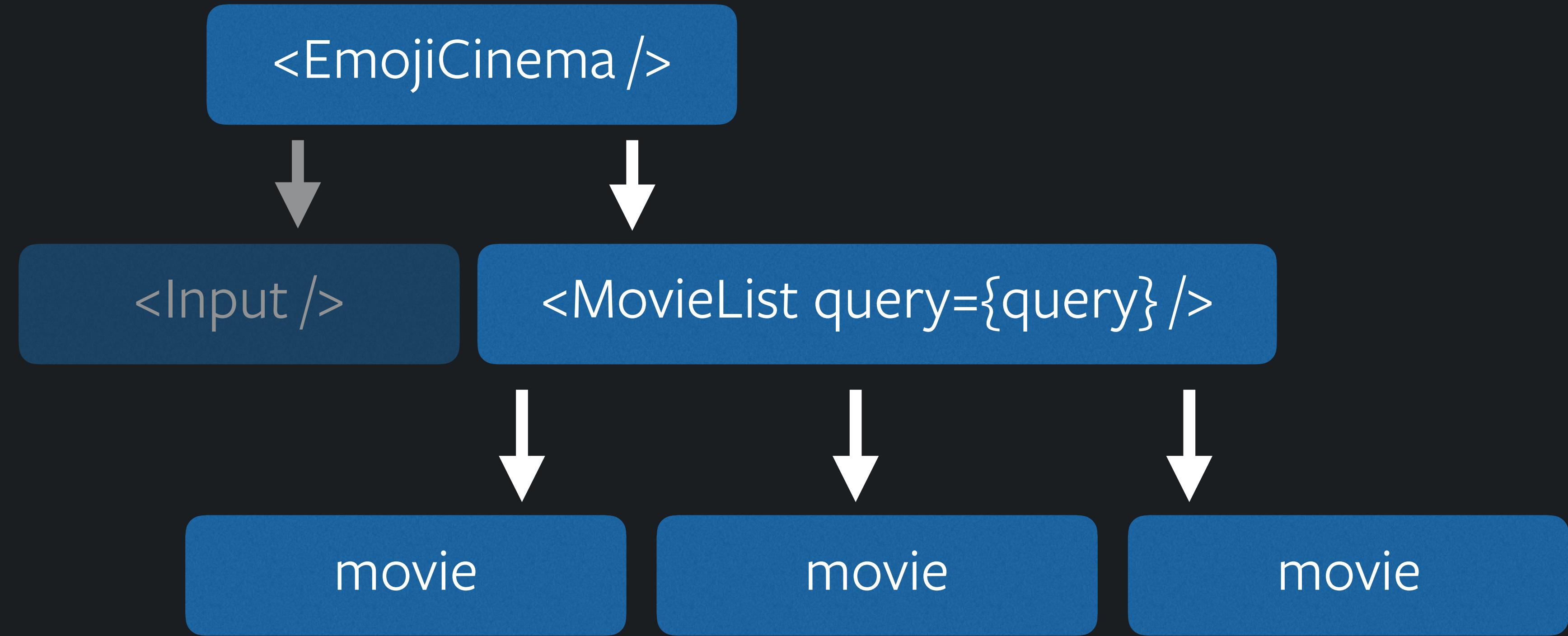
Finding Nemo

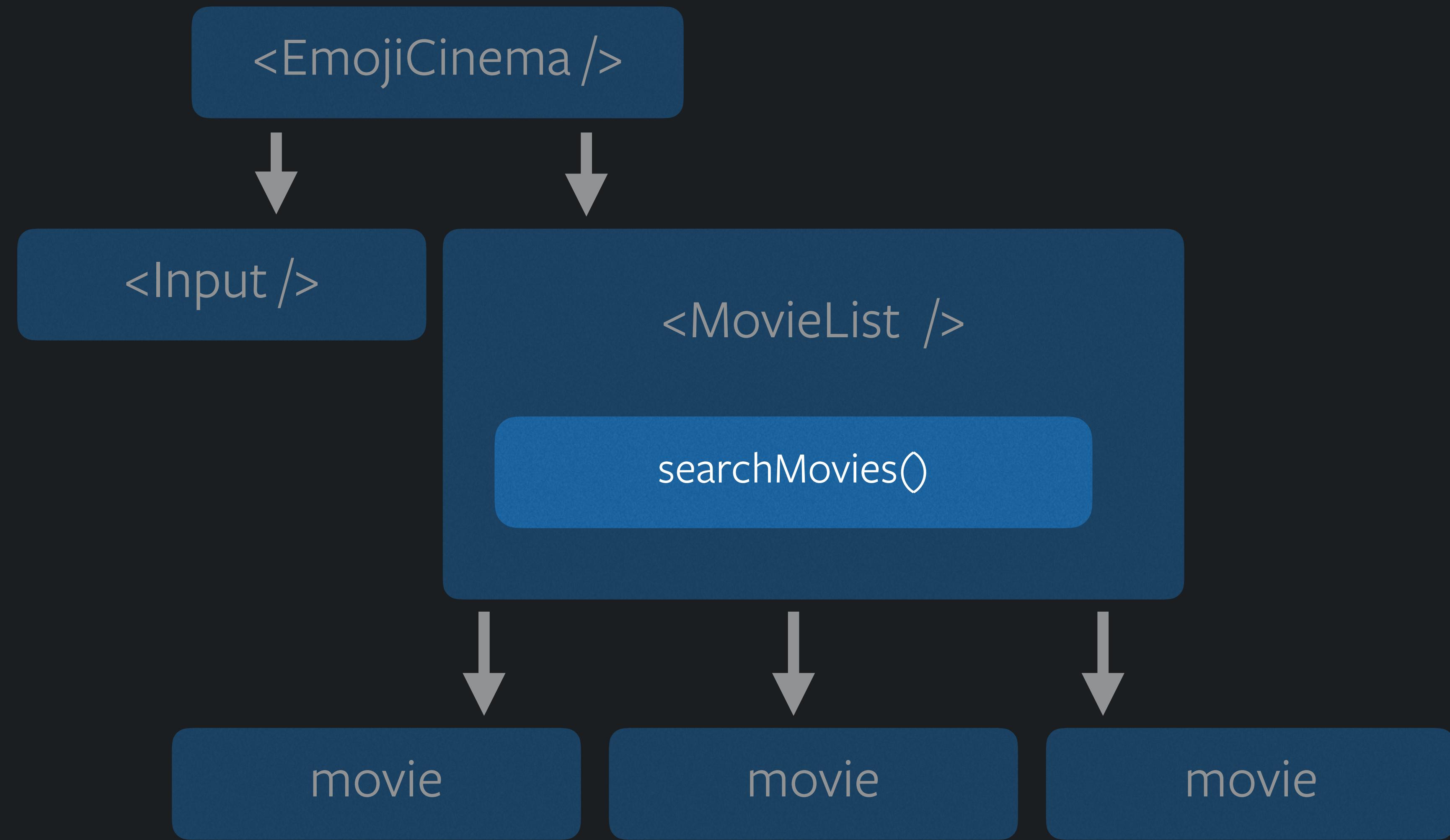


Finding Nemo

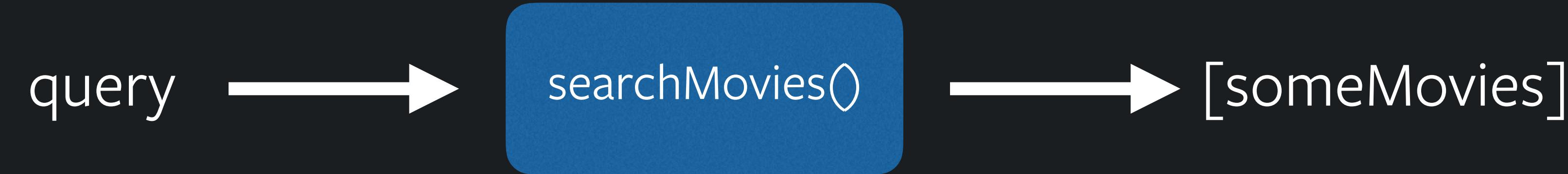


rogeliog/jest-as-a-platform-talk





searchMovies



searchMovies

“F”



searchMovies()



[movies with an F]

```
function searchMovies(query) {  
  return emojiMap.filter((movie) => {  
    return movie.title.includes(query)  
  }) ;  
}  
}
```

```
const emojiMap = [  
  { title: 'Frozen', emoji: '❄️👗' },  
  { title: 'Finding Nemo', emoji: '🔍🐠' },  
  { title: 'Kung Fu Panda', emoji: '👊💥🐼' },  
  { title: 'Harry Potter', emoji: '🧙‍♂️👓⚡️' },  
  { title: 'Hunger Games', emoji: '🍗🎮' },  
  { title: 'The Lion King', emoji: '🦁👑' },  
  { title: 'ET', emoji: '👽🚲🌙' },  
];
```

```
function searchMovies(query) {  
  return emojiMap.filter((movie) => {  
    return movie.title.includes(query)  
  }) ;  
}
```

```
◀ src
  ▲ components
    JS Input.js
    JS MovieList.js
  ▲ lib
    JS searchMovies.js
  JS index.js
{} package.json
```

```
const emojiMap = [
  { title: 'Frozen', emoji: '❄️👗' },
  { title: 'Finding Nemo', emoji: '🔍🐠' },
  { title: 'Kung Fu Panda', emoji: '👊💥🐼' },
  { title: 'Harry Potter', emoji: '🧙‍♂️👓⚡️' },
  { title: 'Hunger Games', emoji: '🍗🎮' },
  { title: 'The Lion King', emoji: '🦁👑' },
  { title: 'ET', emoji: '👽🚲🌙' },
];

function searchMovies(query) {
  return emojiMap.filter((movie) => {
    return movie.title.includes(query)
  });
}
```

Add Jest

{ } package.json

```
{  
  "name": "emoji-cinema",  
  "dependencies": {  
    "react": "16.0.0"  
  },  
  "devDependencies": {}  
}
```

{ } package.json

```
{  
  "name": "emoji-cinema",  
  "dependencies": {  
    "react": "16.0.0"  
  },  
  "devDependencies": {  
    "jest": "21.2.1"  
  }  
}
```

EXPLORER

EMOJI-CINEMA

- src
 - components
 - Input.js
 - MovieList.js
 - lib
 - searchMovies.js
 - index.js
- package.json

JS searchMovies.js x



...

~/dev/reactive-conf/emoji-cinema master*



```
const movies = [  
    { title: 'Frozen', emoji: '❄️👗' },  
    { title: 'Finding Nemo', emoji: '🔍🐠' },  
    { title: 'Kung Fu Panda', emoji: '🐼☀️蜜蜂' },  
    { title: 'Harry Potter', emoji: '🧙‍♂️⚡️⚡️' },  
    { title: 'Hunger Games', emoji: '🍗🎮' },  
    { title: 'The Lion King', emoji: '🦁👑' },  
    { title: 'ET', emoji: '👽🛸🌙' },  
];  
  
module.exports = function searchMovies(query) {  
    return movies.filter((movie) => {  
        return movie.title.includes(query);  
    });  
};
```

Jest has a lot more! ✨

facebook.github.io/jest



What did we get with Jest?

- _immersive watch mode
- Fast
- Great UX
- Git integration
- Works out of the box

Jest as a Platform

JavaScript testing solution

Easy setup

Instant feedback

Works out of the box

Extensible

Works with existing tooling

Build on top

Jest as a Platform

JavaScript testing solution

Easy setup

Instant feedback

Works out of the box

Extensible

Works with existing tooling

Build on top

Jest as a Platform

- 用人选 Jest features
- 用你的工具在 Jest 上面

Cherry pick Jest features



Jest



Jest

 jest-diff

 jest-validate

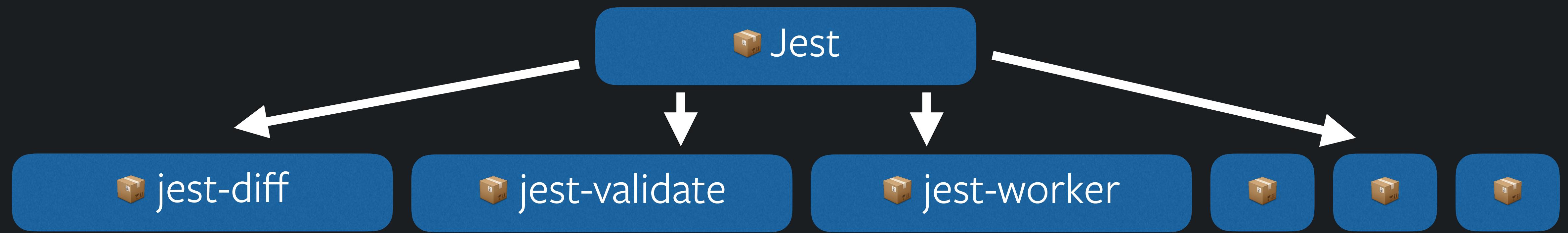
 jest-worker

 jest-snapshot

 expect

 pretty-format

 jest-changed-files





jest-diff



jest-validate



jest-worker



```
const diff = require('jest-diff');

diff(et, strangerThings);
```

Output

```
Object {
-  "emoji": "\ud83d\udcbb \ud83d\udcbb \ud83d\udcbb",
-  "title": "ET",
+  "emoji": "\ud83d\udcbb \ud83d\udcbb \ud83d\udcbb \ud83e\udd00 \ud83d\udcbb",
+  "title": "Stranger Things",
}
```



jest-diff



jest-validate



jest-worker



```
const { validate } = require('jest-validate');
```

```
const config = { watch: true };
```

```
validate(config, {  
} );
```



jest-diff



jest-validate



jest-worker



```
const { validate } = require('jest-validate');

const config = { watch: true };

validate(config, {
  exampleConfig: { watch: true },
}) ;
```



jest-diff



jest-validate



jest-worker



```
const { validate } = require('jest-validate');

const config = { watch: 'other value' };

validate(config, {
  exampleConfig: { watch: true },
}) ;
```

Output

- Validation Error:

Option "watch" must be of type:
boolean
but instead received:
string

Example:

```
{  
  "watch": true  
}
```



jest-diff



jest-validate



jest-worker



```
const { validate } = require('jest-validate');

const config = { watch: 'other value' };

validate(config, {
  comment: 'Documentation: http://emoji-cinema-docs.com',
  exampleConfig: { watch: true },
});
```

Output

```
Option "watch" must be of type:  
  boolean  
but instead received:  
  string
```

Example:

```
{  
  "watch": true  
}
```

→ Documentation: <http://emoji-cinema-docs.com>



jest-diff



jest-validate



jest-worker



```
const { validate } = require('jest-validate');

const config = { watch: 'other value' };

validate(config, {
  comment: 'Documentation: http://emoji-cinema-docs.com',
  exampleConfig: { watch: true },
  deprecatedConfig: {
    watch: () => `watch is deprecated, use 'superWatch' instead`
  }
});
```

Output

- **Deprecation Warning:**

'watch' is deprecated, use 'superWatch' instead

Documentation: <http://emoji-cinema-docs.com>



jest-diff



jest-validate



jest-worker



searchMovies.test.js

JS run.js

```
export.runTestFile = (testFile) => {
  // runs the tests in test file
}
```



jest-diff



jest-validate



jest-worker



jest-worker

searchMovies.test.js

JS run.js

```
export.runTestFile = (testFile) => {
  // runs the tests in test file
}
```



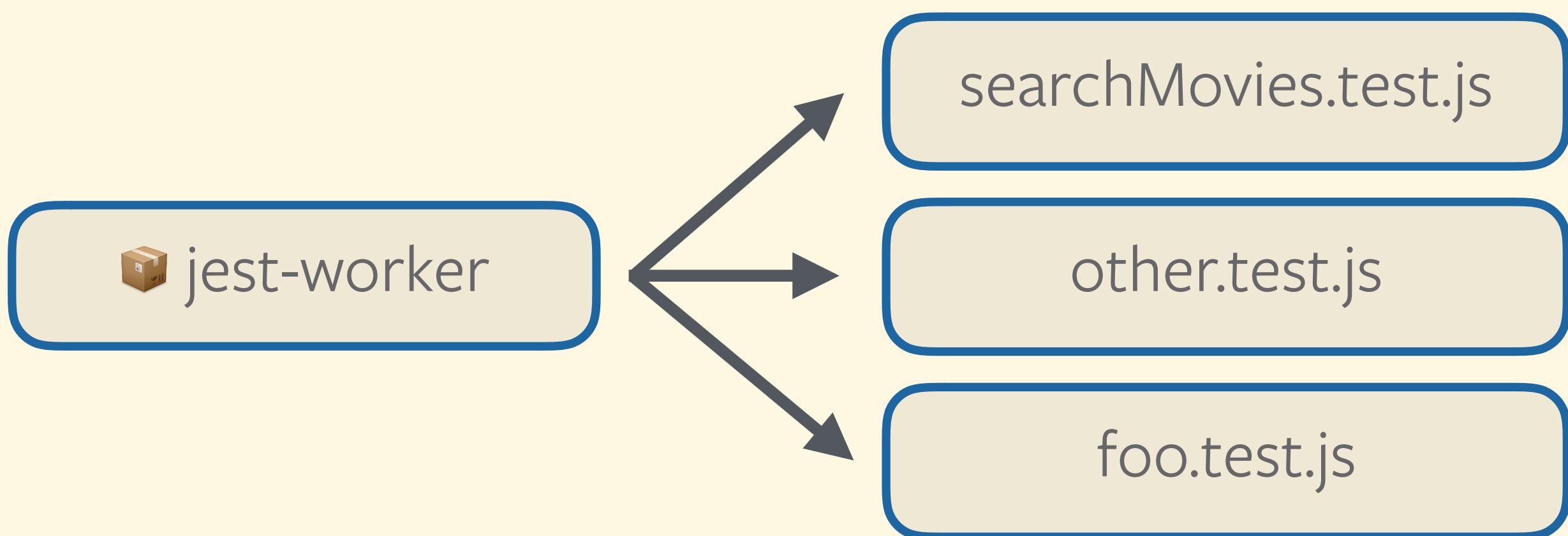
jest-diff



jest-validate



jest-worker



JS run.js

```
export.runTestFile = (testFile) => {
  // runs the tests in test file
}
```

 jest-diff

 jest-validate

 jest-worker



JS run.js

```
export.runTestFile = (testFile) => {
  // runs the tests in test file
}
```



jest-diff



jest-validate



jest-worker



```
const Worker = require('jest-worker');

const worker = new Worker(require.resolve('./run'), {
  exposedMethods: ['runTestFile'],
});
```



jest-diff



jest-validate



jest-worker

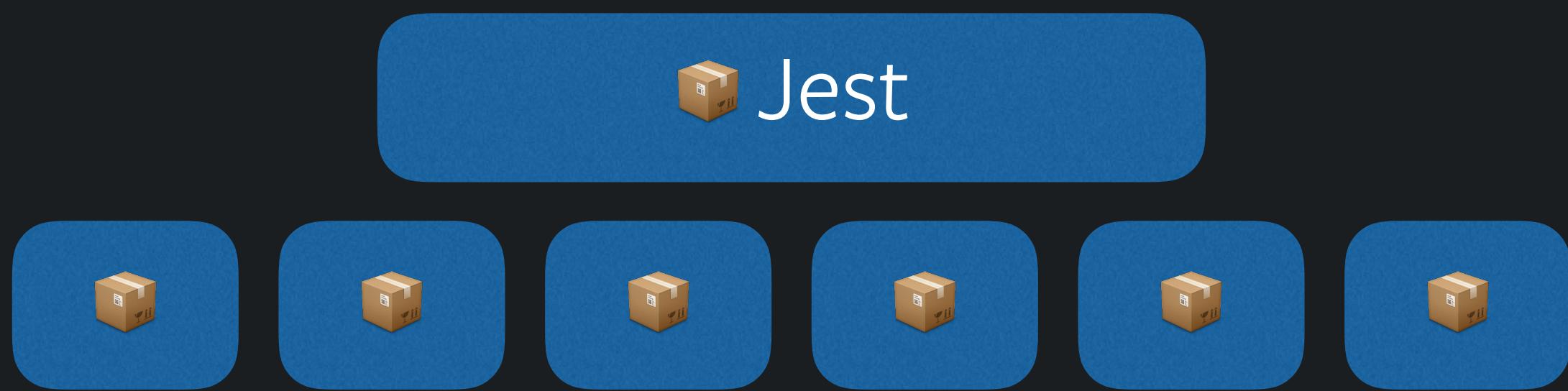


```
const Worker = require('jest-worker');

const worker = new Worker(require.resolve('./run'), {
  exposedMethods: ['runTestFile'],
} );

Promise.all([
  worker.runTestFile('./searchMovies.test.js'),
  worker.runTestFile('./other.test.js'),
  worker.runTestFile('./foo.test.js'),
])
.then(results => printResults(results))
```

Cherry pick Jest features



You tool on top of Jest

Your tool

 Jest

Runner

Great UX

Filtering

Fast

Parallelization

Your tool



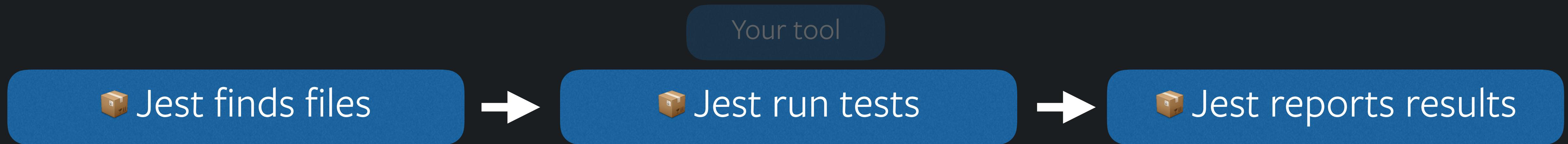
Jest

Reporters

Watch mode

Git integration

Lifecycle of a test run



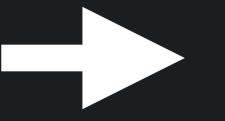
*Filtering
Great UX
Fast
Parallelization
Watch mode
Git integration*

Runner

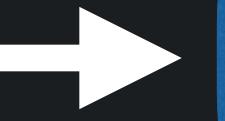
*Great UX
Reporters
Watch mode*

Jest Runner

 Jest finds files



Your tool



 Jest reports results

*Filtering
Great UX
Fast
Parallelization
Watch mode
Git integration*

Runner

*Great UX
Reporters
Watch mode*

Example of a Jest Runner

Jest finds files



Jest reports results

 Jest finds files



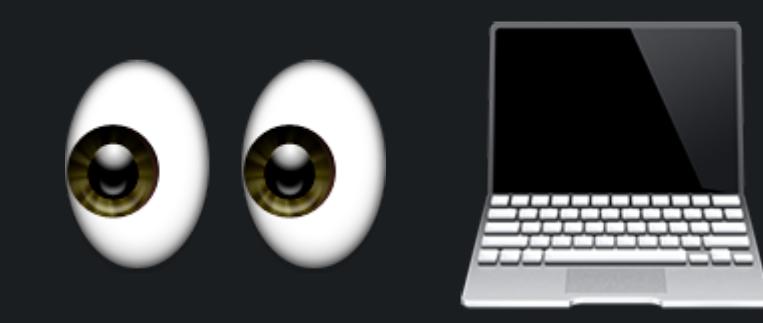
mocha



 Jest reports results



[rogeliog/jest-runner-mocha](https://github.com/rogeliog/jest-runner-mocha)



{ } package.json

```
{  
  "name": "emoji-cinema",  
  "dependencies": {  
    "react": "16.0.0"  
  },  
  "devDependencies": {  
    "mocha": "4.0.1"  
  }  
}
```

{ } package.json

```
{  
  "name": "emoji-cinema",  
  "dependencies": {  
    "react": "16.0.0"  
  },  
  "devDependencies": {  
    "mocha": "4.0.1",  
    "jest": "21.2.1",  
    "jest-runner-mocha": "0.4.0",  
  }  
}
```

{} package.json

```
{  
  "name": "emoji-cinema",  
  "dependencies": {  
    "react": "16.0.0"  
  },  
  "devDependencies": {  
    "mocha": "4.0.1",  
    "jest": "21.2.1",  
    "jest-runner-mocha": "0.4.0",  
  },  
  "jest": {  
    "runner": "jest-runner-mocha"  
  }  
}
```

JS searchMovies.test.js ×



~/dev/reactive-conf/emoji-cinema master*

```
const assert = require('assert');
const searchMovies = require('./searchMovies');

it('Returns the movies that match the query', () : void {
  assert.deepEqual(searchMovies('F'), [
    { title: 'Frozen', emoji: '❄️👗' },
    { title: 'Finding Nemo', emoji: '🔍🐠' },
    { title: 'Kung Fu Panda', emoji: '🐼💥🐼' }
  ])
})
```

```
graph LR; A["Jest finds files"] --> B((mocha)); B --> C["Jest reports results"]
```

Jest finds files → mocha → Jest reports results

 Jest finds files



 Jest reports results



[rogeliog/jest-runner-eslint](https://github.com/rogeliog/jest-runner-eslint)



Joe Lencioni
@lencioni

Follow



We just switched to this at [@AirbnbEng](#) and our [@geteslint](#) job in CI dropped from ~16m to ~4m! Thanks [@rogeliog!](#) (and [@ljharb](#)/[@iancmyers](#))



Rogelio Guzman @rogeliog

Just published jest-runner-eslint, an experimental @geteslint runner for @fbjest github.com/rogeliog/jest-... 😊

5:22 PM - 11 Sep 2017

{ } package.json

```
{  
  "devDependencies": {  
    "jest": "21.2.1",  
    "jest-runner-eslint": "0.3.0"  
  }  
}
```

{ } package.json

```
{  
  "devDependencies": {  
    "jest": "21.2.1",  
    "jest-runner-eslint": "0.3.0"  
  },  
  "jest": {  
    "runner": "jest-runner-eslint",  
    "displayName": "lint",  
    "testMatch": ["**/*.js"]  
  }  
}
```

JS searchMovies.test.js ✘



~/dev/reactive-conf/emoji-cinema master*



```
const assert = require('assert');
const searchMovies = require('./searchMovies');

it('Returns the movies that match the query', () {
  assert.deepEqual(searchMovies('F'), [
    { title: 'Frozen', emoji: '❄️ 🎵' },
    { title: 'Finding Nemo', emoji: '🔍🐠' },
    { title: 'Kung Fu Panda', emoji: '👊💥🐼' }
  ])
})
```

 Jest finds files



 Jest reports results

 Jest finds files



 Jest reports results

Jest finds files

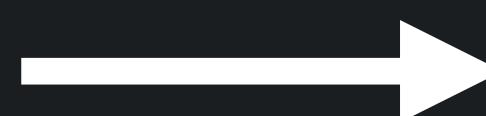


mocha

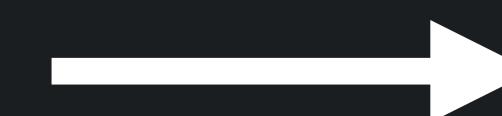


Jest reports results

Jest finds files



 ESLint



Jest reports results

Jest finds files



 python™



Jest reports results

Jest finds files



Jest reports results



experimental

A screenshot of a terminal window titled "2. jest /Users/cpojer/Projects/pyjest/example (node)". The window shows the output of a Jest test run. It includes a color-coded summary table, a stack trace of failing tests, and detailed test results for each failed test. The terminal also displays the total number of test suites, tests, and snapshots run, along with the execution time and a message indicating all test suites were ran. At the bottom, there is a "Watch Usage" instruction.

```
sample_test.py — example
sample_test.py

1 def func(x):
2     return x + 1
3
4 def test_answer():
5     assert func(3) == 5
6

PASS  src/sample2_test.py
FAIL  src/sample_test.py
● src/sample_test.py::test_answer

    def test_answer():
        assert func(3) == 5
        assert 4 == 5
        +  where 4 = func(3)

    src/sample_test.py:5: AssertionERROR

Test Suites: 1 failed, 1 passed, 2 total
Tests:       1 failed, 1 passed, 2 total
Snapshots:   0 total
Time:        0.39s, estimated 1s
Ran all test suites.

Watch Usage: Press w to show more. █
```

Let's write a Jest Runner! 🧑‍🦰🏃

blade-runner



blade-runner ✕

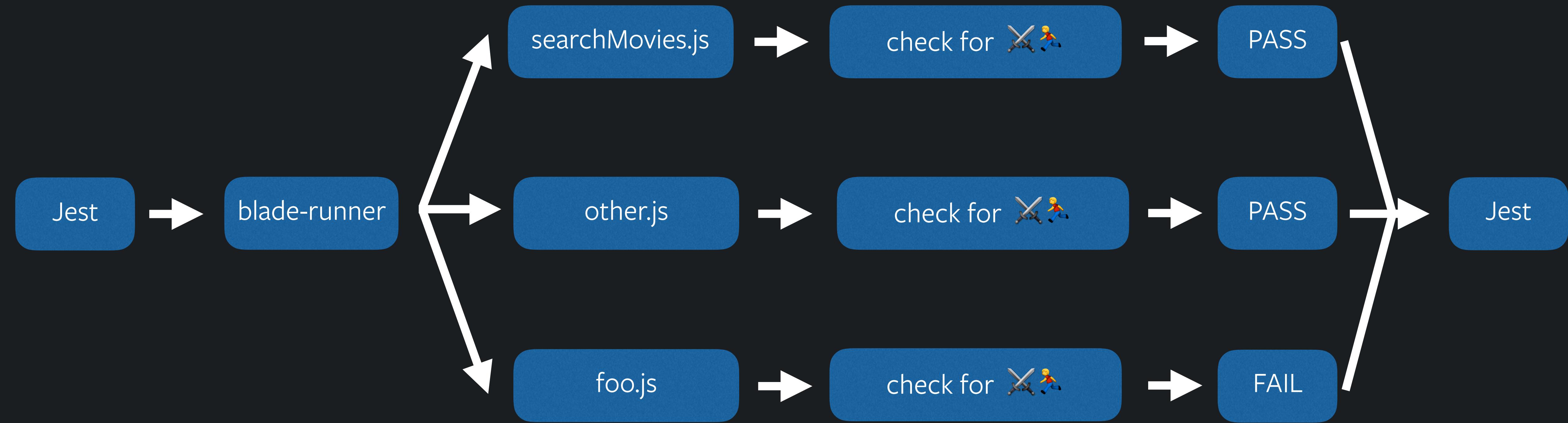


```
function searchMovies(query) {  
  return emojiMap.filter((movie) => {  
    return movie.title.includes(query)  
  }) ;  
}
```

blade-runner ✕🏃



```
// ✕🏃
function searchMovies(query) {
  return emojiMap.filter((movie) => {
    return movie.title.includes(query)
  ) ;
}
```



Entry file
index.js

Worker
run.js

blade-runner

searchMovies.js

check for 

PASS

other.js

check for 

PASS

foo.js

check for 

FAIL

Entry file
index.js

blade-runner

Handle test interruption

Keep Jest up to date with the status of each file

Spin up workers

Clean up workers after it finishes.

Report test results to Jest



searchMovies.js

other.js

foo.js



rogeliog/create-jest-runner

Entry file
index.js

blade-runner

Handle test interruption

Keep Jest up to date with the status of each file

Spin up workers

Clean up workers after it finishes.

Report test results to Jest



searchMovies.js

other.js

foo.js

```
const { createJestRunner } = require('create-jest-runner');

module.exports = createJestRunner('./run');
```

JS run.js

```
module.exports = ( { testPath } ) => {  
} ;
```

```
const fs = require('fs');

module.exports = ({ testPath }) => {
  const start = new Date();
  const contents = fs.readFileSync(testPath, 'utf8');
};
```

```
const fs = require('fs');

module.exports = ({ testPath }) => {
  const start = new Date();
  const contents = fs.readFileSync(testPath, 'utf8');

  if (contents.includes('⚔️🏃')) {
  } else {
  }
};
```

```
const fs = require('fs');
const { pass, fail } = require('create-jest-runner');

module.exports = ({ testPath }) => {
  const start = new Date();
  const contents = fs.readFileSync(testPath, 'utf8');

  if (contents.includes('✗🏃')) {
    const end = new Date();
    return pass({ start, end, testPath });
  } else {
    const end = new Date();
    const errorMessage = 'Company policies require ✗🏃 in every file';
    return fail({ start, end, testPath, errorMessage });
  }
};
```

EXPLORER

EMOJI-CINEMA

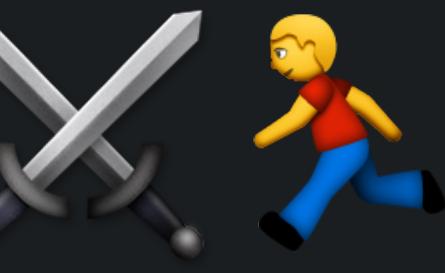
- src
 - components
 - Input.js
 - MovieList.js
 - lib
 - searchMovies.js
 - searchMovies.te...
 - index.js
- package.json

JS searchMovies.test.js x ~/dev/reactive-conf/emoji-cinema master*

```
const assert = require('assert');
const searchMovies = require('./searchMovies');

it('Returns the movies that match the query', () => {
  assert.deepEqual(searchMovies('F'), [
    { title: 'Frozen', emoji: '❄️⛄' },
    { title: 'Finding Nemo', emoji: '🔍🐠' },
    { title: 'Kung Fu Panda', emoji: '🐼☀️' }
  ]);
});
```

That was blade-runner ✕





mocha

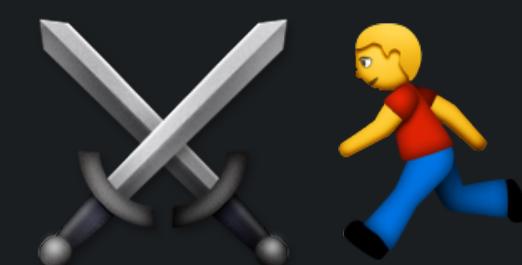
{ } package.json

```
"jest": {  
  "runner": "jest-runner-mocha",  
}
```



{ } package.json

```
"jest": {  
  "runner": "jest-runner-eslint",  
  "testMatch": ["**/*.js"]  
}
```



{ } package.json

```
"jest": {  
  "runner": "jest-runner-blade",  
  "testMatch": ["**/*.js"]  
}
```



mocha

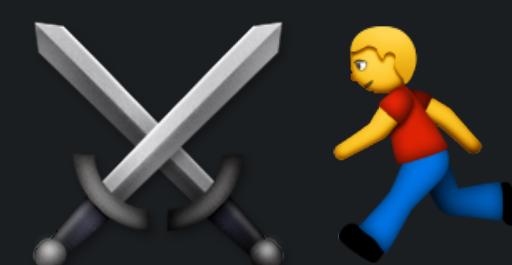
JS jest-mocha.config.js

```
module.exports = {  
  runner: "jest-runner-mocha",  
};
```



{} package.json

```
"jest": {  
  "runner": "jest-runner-eslint",  
  "testMatch": ["**/*.js"]  
}
```



{} package.json

```
"jest": {  
  "runner": "jest-runner-blade",  
  "testMatch": ["**/*.js"]  
}
```



mocha

JS jest-mocha.config.js

```
module.exports = {
  runner: "jest-runner-mocha",
};
```



JS jest-eslint.config.js

```
module.exports = {
  runner: "jest-runner-eslint",
  testMatch: ["**/*.js"]
};
```



{ } package.json

```
"jest": {
  "runner": "jest-runner-blade",
  "testMatch": ["**/*.js"]
}
```



mocha

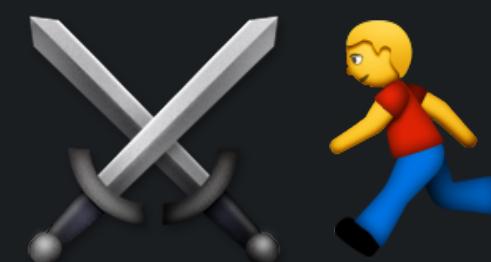
JS jest-mocha.config.js

```
module.exports = {
  runner: "jest-runner-mocha",
};
```



JS jest-eslint.config.js

```
module.exports = {
  runner: "jest-runner-eslint",
  testMatch: ["**/*.js"]
};
```



JS jest-blade.config.js

```
module.exports = {
  runner: "jest-runner-blade",
  testMatch: ["**/*.js"]
};
```

Jest Projects



+



+



yarn jest --projects jest-* .config.js

EXPLORER

EMOJI-CINEMA

- src
- components
 - Input.js
 - MovieList.js
- lib
 - searchMovies.js
 - searchMovies.te...
- index.js
- jest-blade.config.js
- jest-eslint.config.js
- jest-mocha.config.js
- package.json

JS searchMovies.test.js x

JS jest-mocha.config



```
// ✅
const assert = require('assert');
const searchMovies = require('./searchMovies')

it('Returns the movies that match the query',
  assert.deepEqual(searchMovies('F'), [
    { title: 'Frozen', emoji: '❄️👗' },
    { title: 'Finding Nemo', emoji: '🔍🐠' },
    { title: 'Kung Fu Panda', emoji: '🥋☀️🐼' }
  ]);
});
```

~/dev/reactive-conf/emoji-cinema master*



What's next?

- ⌚ This is just the beginning
- ⌚ Future applications are wider
- ⌚ Open area of research

Thanks! 😊
@rogeliog



rogeliog/jest-as-a-platform-talk