



# E2E 验证测试

基于 Cypress 的端到端测试

## 基础概念

简单来说，就是模拟真实用户使用场景进行测试，预期应用能够正常响应用户的操作，其关键点在于模拟用户使用环境，模拟用户操作。

那对于 Web 应用来说，用户环境就是浏览器，用户操作主要是移动、点击，这些就是我们需要模拟的部分，下面就直接进入环境和实践部分。

<https://www.cnblogs.com/keenajiao/p/16418734.html>

比如 Vue 的代码中

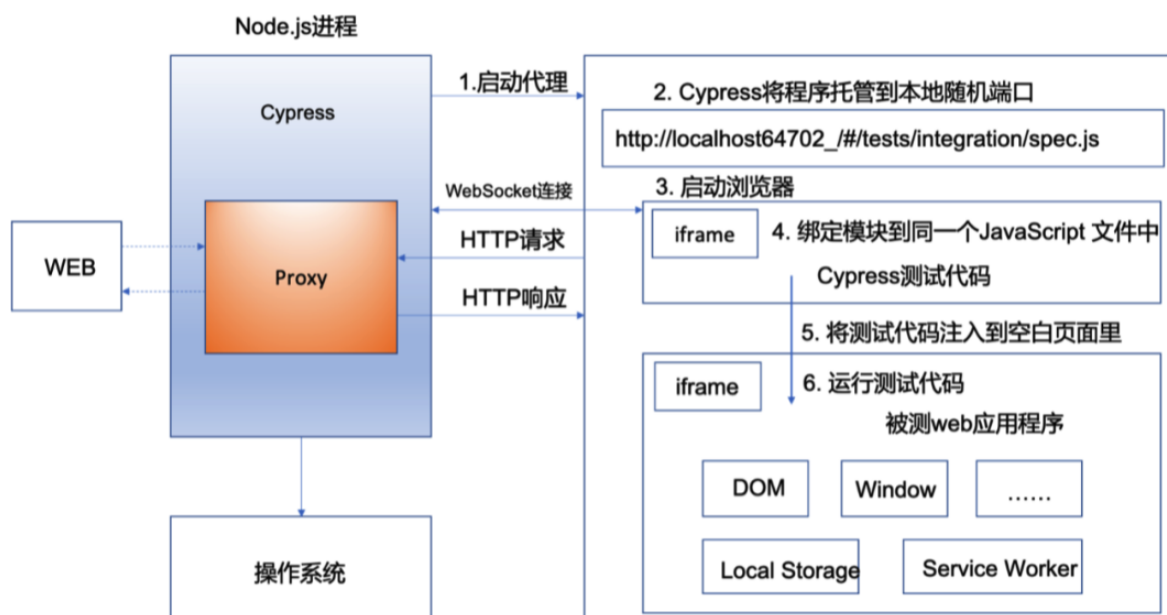
<https://github.com/vuejs/core/blob/main/packages/vue/tests/Transition.spec.ts>

## 什么是 Cypress?

Cypress 是基于 JavaScript 语言的前端自动化测试工具，无需借助外部工具，自集成了一套完整的端到端测试方法，可以对浏览器中运行的所有内容进行快速、简单、可靠的测试，并且可以进行接口测试。

## Cypress 的工作原理

Cypress 是在测试开始并首次加载 Cypress 时，Cypress Web 内部应用程序先把自己托管到本地的一个随机端口上，然后使用 webpack 将测试代码中的所有模块绑定到同一个 JavaScript 文件中，启动指定的浏览器，并将测试代码注入到空白页面里，同时运行测试代码。当识别出测试脚本中发出的第一个 `cy.visit()` 命令后，Cypress 将会更改其本地的 URL 以匹配远端应用程序的地址，使得测试代码和应用程序可以在一个生命周期中运行。



## Cypress 的缺点

- 只支持 JS 框架去编写测试用例
- 不支持远程执行
- 不支持多个浏览器 tab
- 默认不同时支持多个浏览器

## 实战

<https://github.com/cypress-io/cypress>

Bash

```
pnpm i cypress -D
```

启动 cypress

```
npx cypress open
```

Bash

```
<html lang="en">

<body>
  <div id='app'>
    <input />
    <button></button>
  </div>
  <script>
    const data = {
      message: 'Hello'
    }

    function update() {
      // 更新视图
      document.querySelector('button').innerHTML = data.message
      document.querySelector('input').value = data.message
    }

    document.querySelector('input').addEventListener('keyup', function
      data.message = this.value
      update()
    })

    document.querySelector('button').addEventListener('click', function
      data.message = data.message.split('').reverse().join('')
      update()
    })

    // 初始化
    update()
  </script>
```

HTML

```
</body>
```

```
</html>
```

JavaScript

```
describe("双向绑定", () => {
  beforeEach(() => {
    cy.visit("http://127.0.0.1:8080/");
  });

  it("初始状态", () => {
    cy.get("input").should("have.value", "Hello");
    cy.get("button").should("have.text", "Hello");
  });

  it("Input 输入", () => {
    cy.get("input").type(" Cypress");
    cy.get("input").should("have.value", "Hello Cypress");
    cy.get("button").should("have.text", "Hello Cypress");
  });

  it("按钮点击", () => {
    cy.get("button").click();
    cy.get("input").should("have.value", "olleH");
    cy.get("button").should("have.text", "olleH");
  });
});
```

