

Cypress Testing Masterclass

Let the Bots  Test While You Rest 

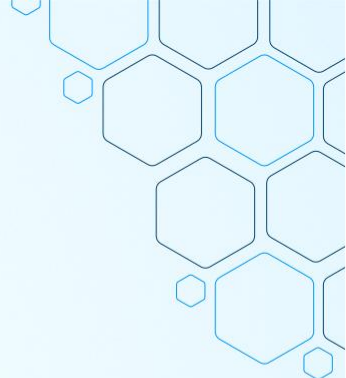
Vedha Viyash
17th April 2024



Appsilon



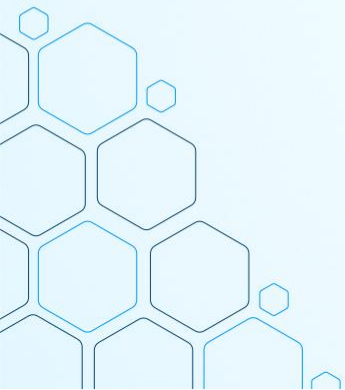
ShinyConf **2024**
Powered by Appsilon



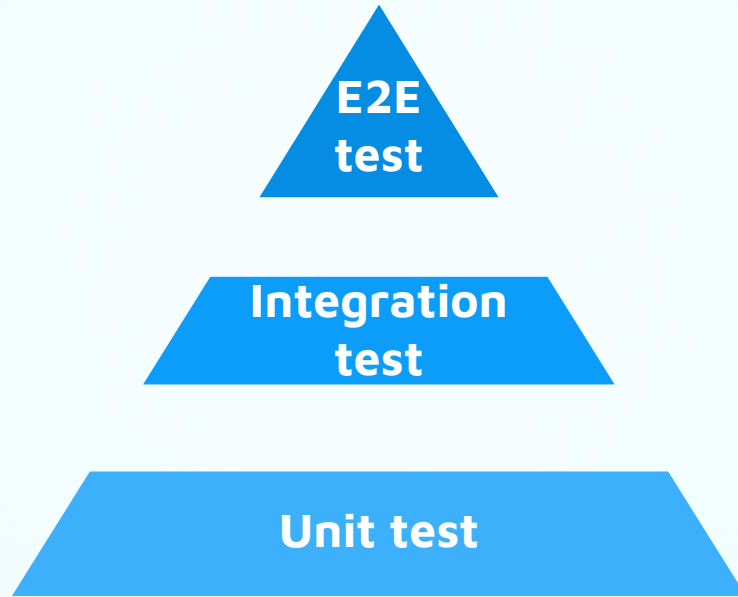
“

If you do not have automated tests, the users are the test.

”



Testing Pyramid



Testing Pyramid

**More
Integration**



**More
Isolation**

**E2E
test**

**Integration
test**

Unit test

Testing Pyramid

More
Integration



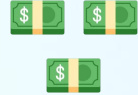
E2E
test

Integration
test

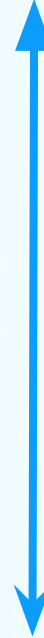
Unit test

More
Isolation

Slower



Faster



Testing Pyramid

More
Integration



More
Isolation

E2E
test

Integration
test

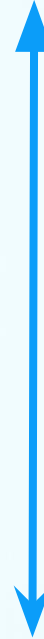
Unit test



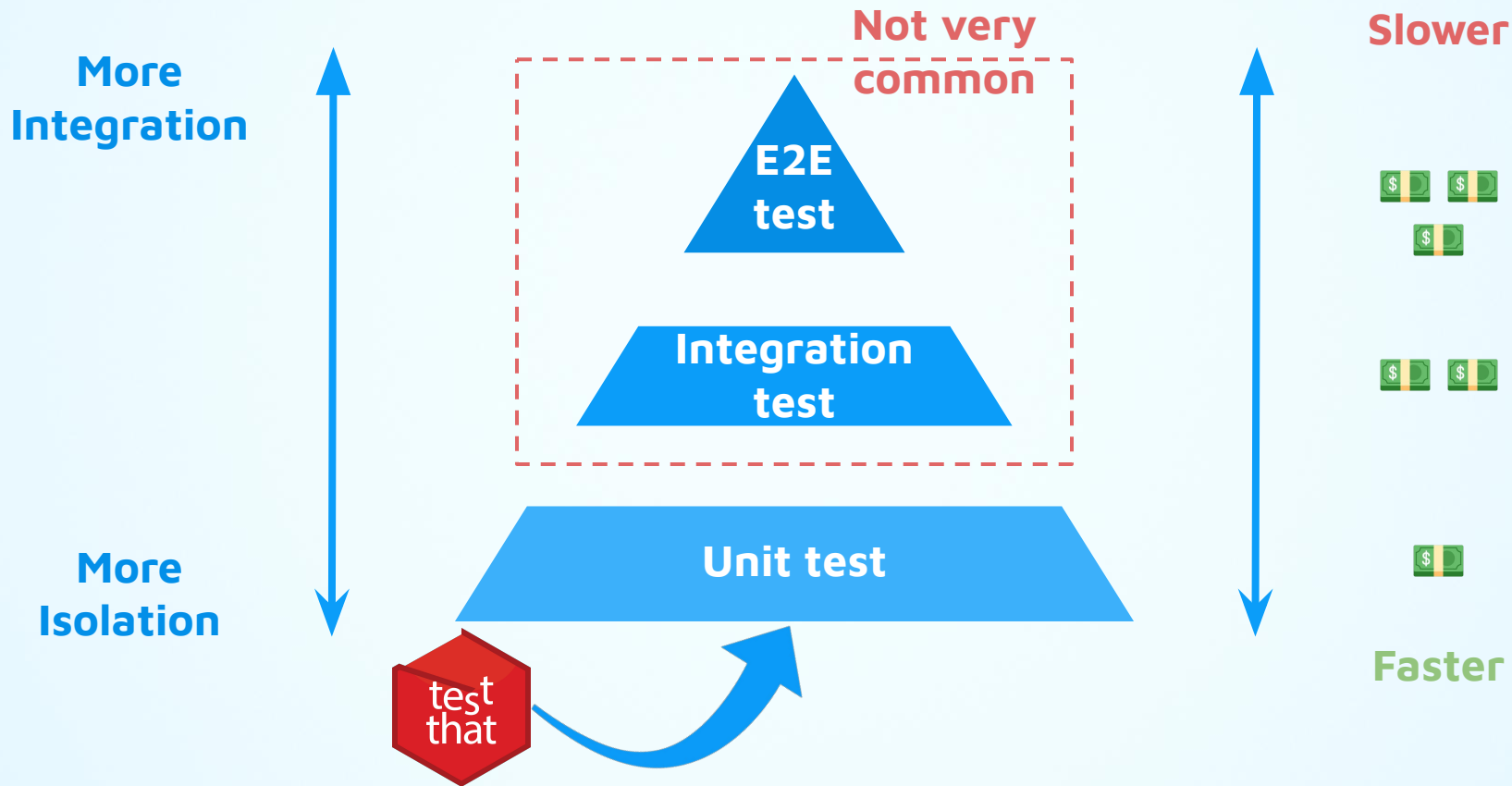
Slower



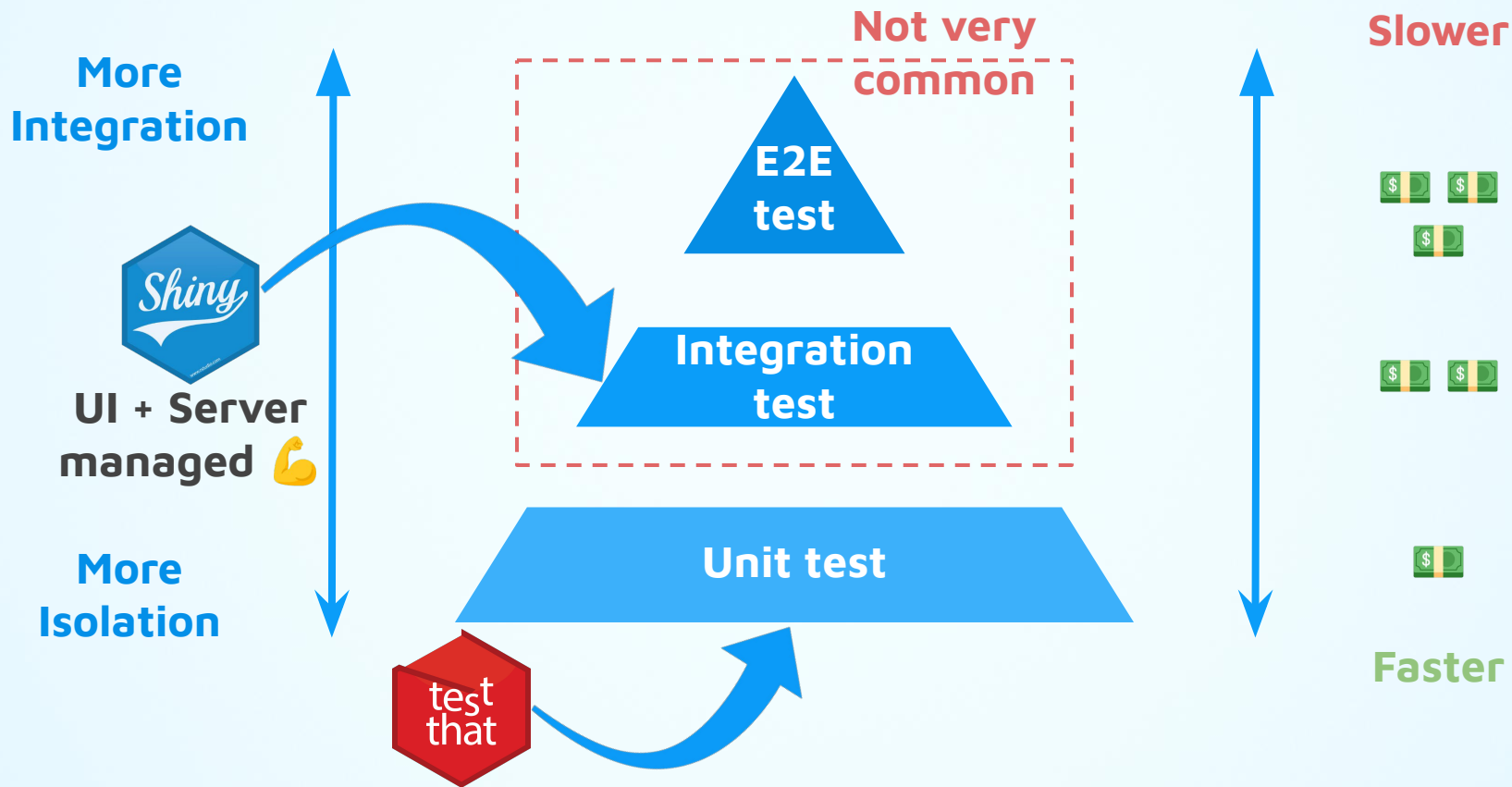
Faster



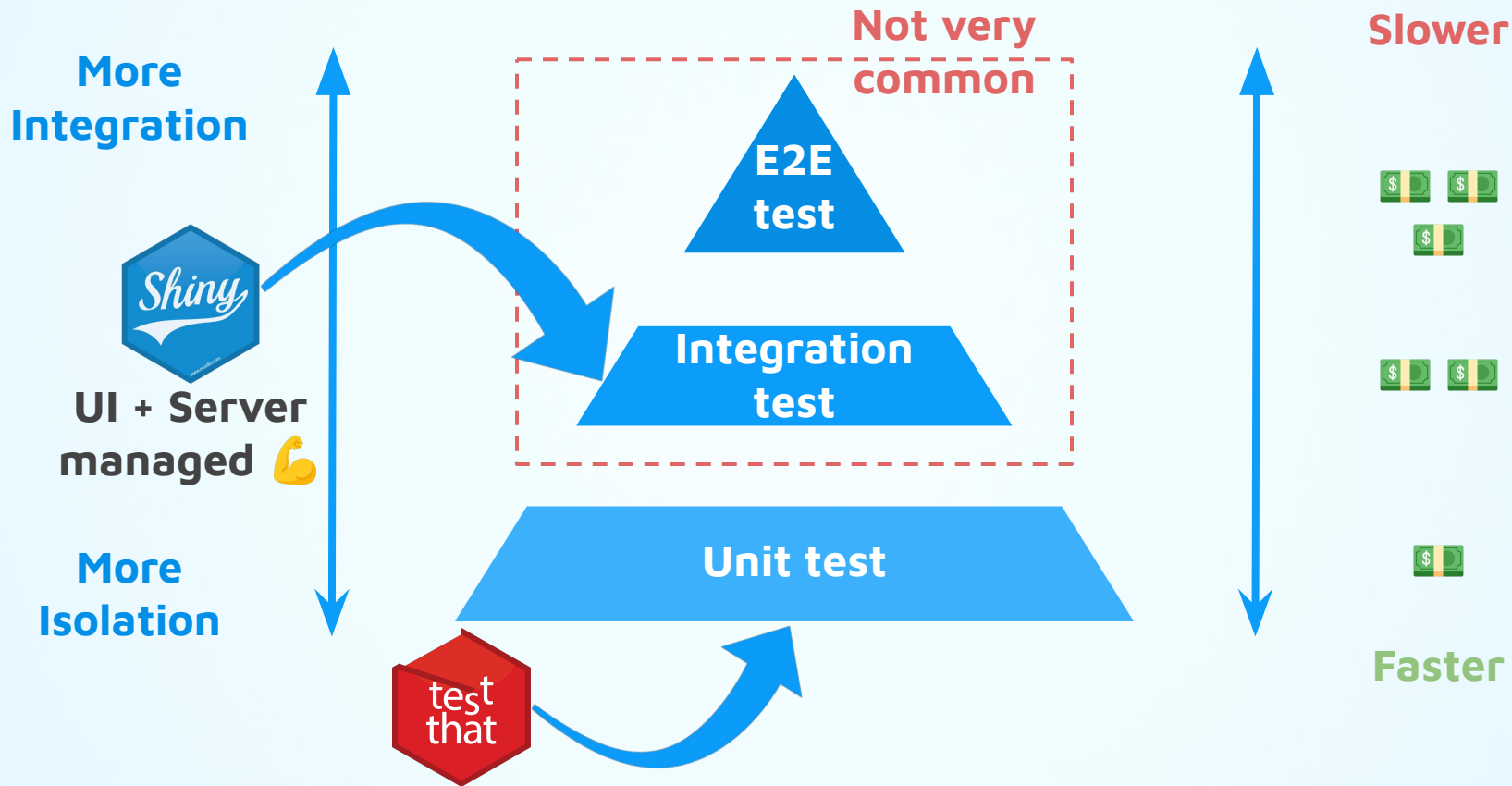
Testing Pyramid



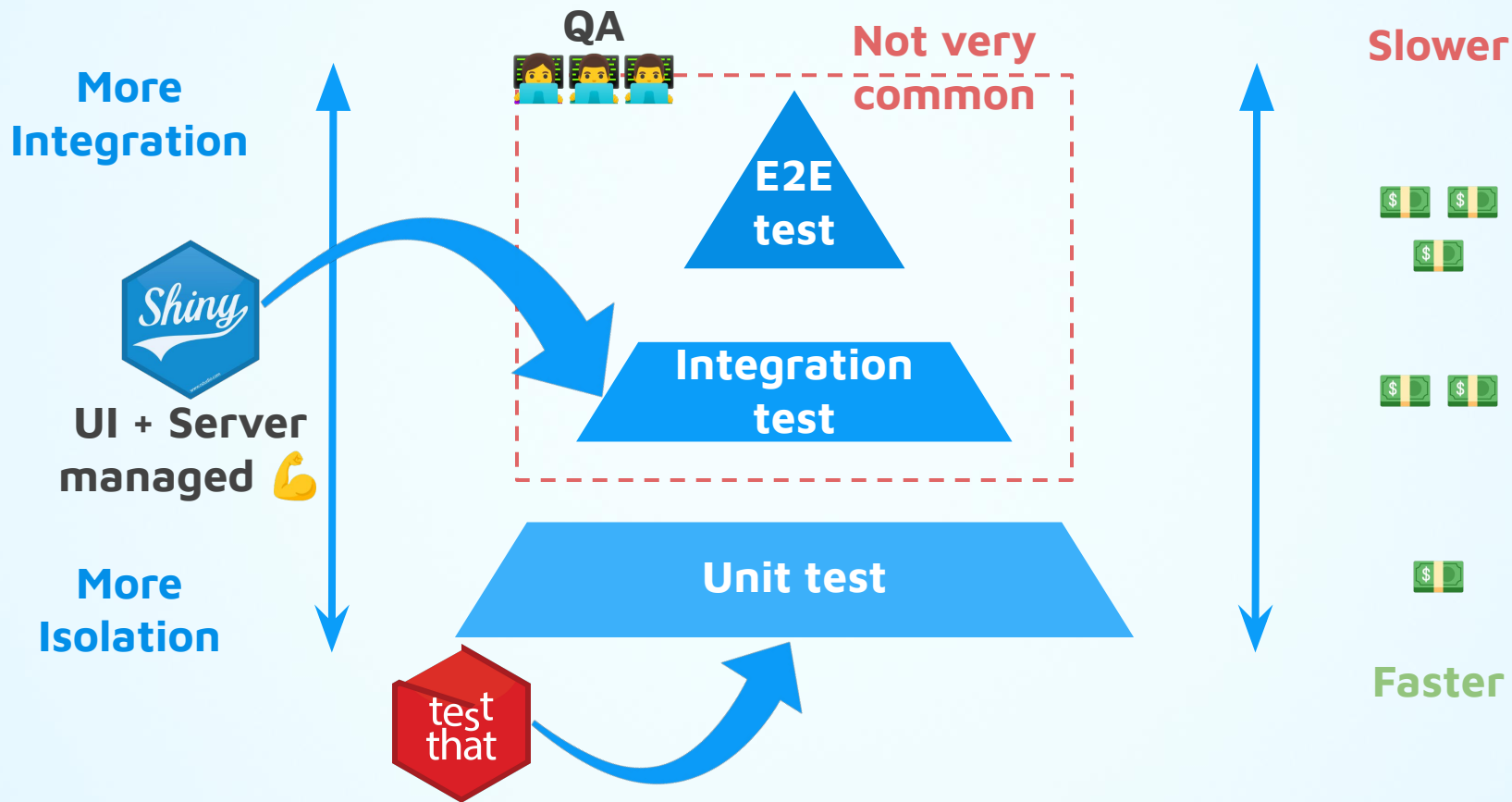
Testing Pyramid



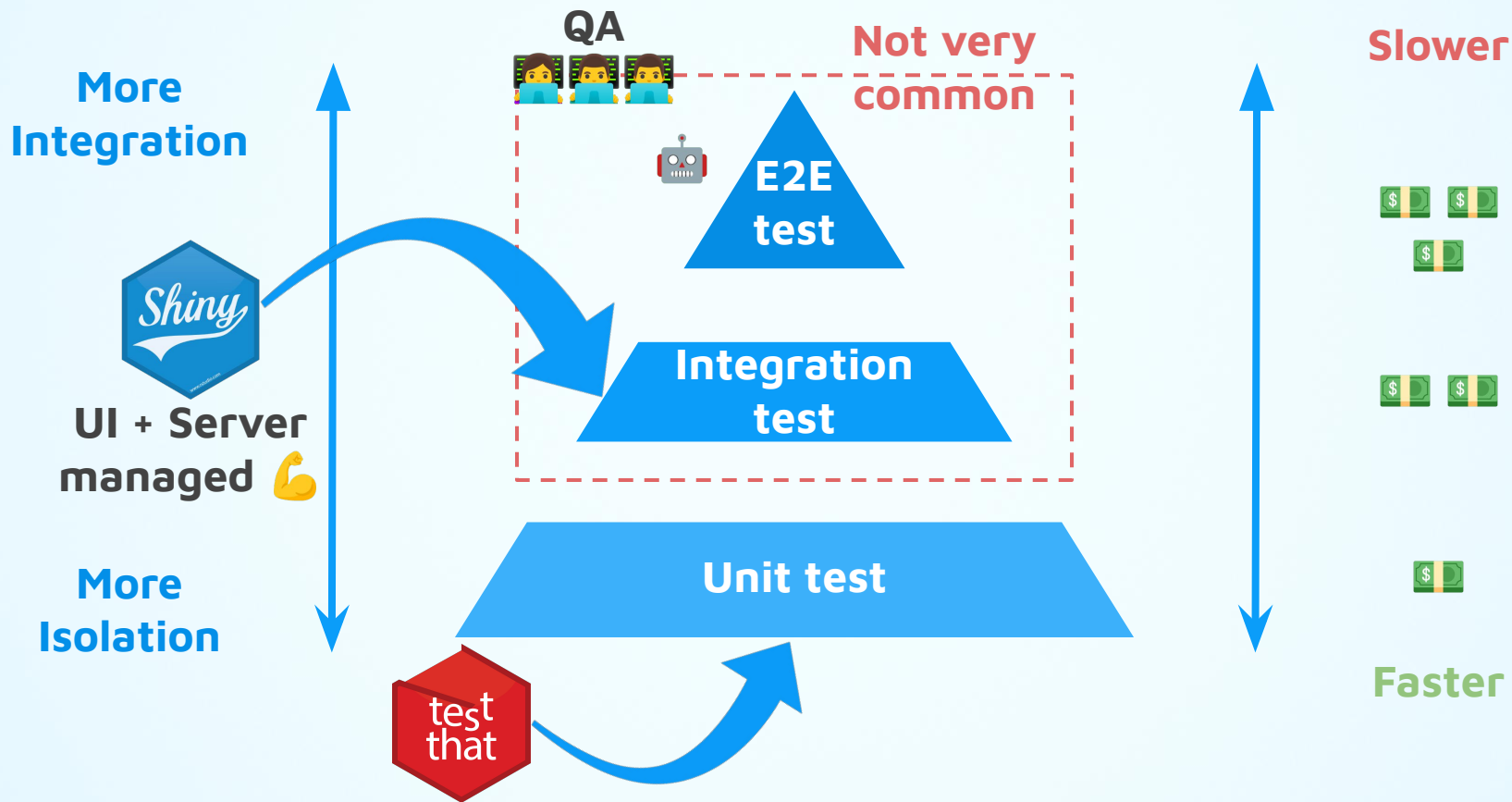
Testing Pyramid



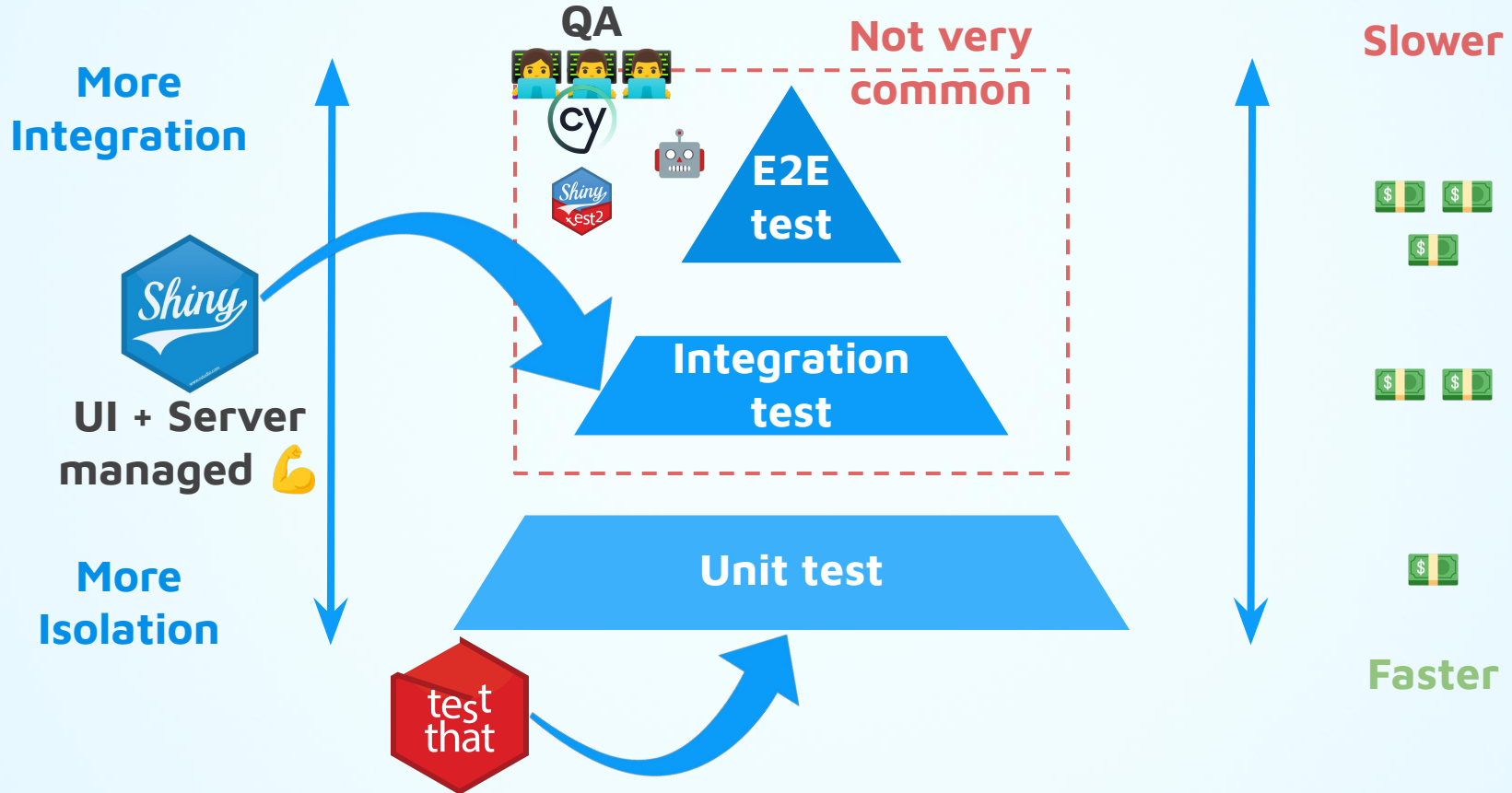
Testing Pyramid



Testing Pyramid



Testing Pyramid





VS






- Works on any web app
- Performs real interaction
- Relatively higher learning curve
- Rich API for app interaction to simulate real usage
- Can test actual deployments

- It taps deep into shiny and testthat
- Bypasses real interaction
- Easy to setup and record tests
- Limited options for finding elements (CSS selectors only)
- Perfect for regression testing

JS

A pinch of JS is enough to use cypress

- Node JS to run JS locally 
- Package manager. Here we will use 
- Basic JS syntax 

Installation and setup

Step 1

Download and install Node.js for your OS from: <https://nodejs.org/en/download>

Step 2

Verify that you can access node and npm using the commands:

node -v

npm -v



Installation and setup

Step 3

Create a new npm project using the command in a new working directory:

npm init -y



package.json

Installation and setup

Step 4

Install cypress using the following command:

npm install cypress



node_modules



package.json



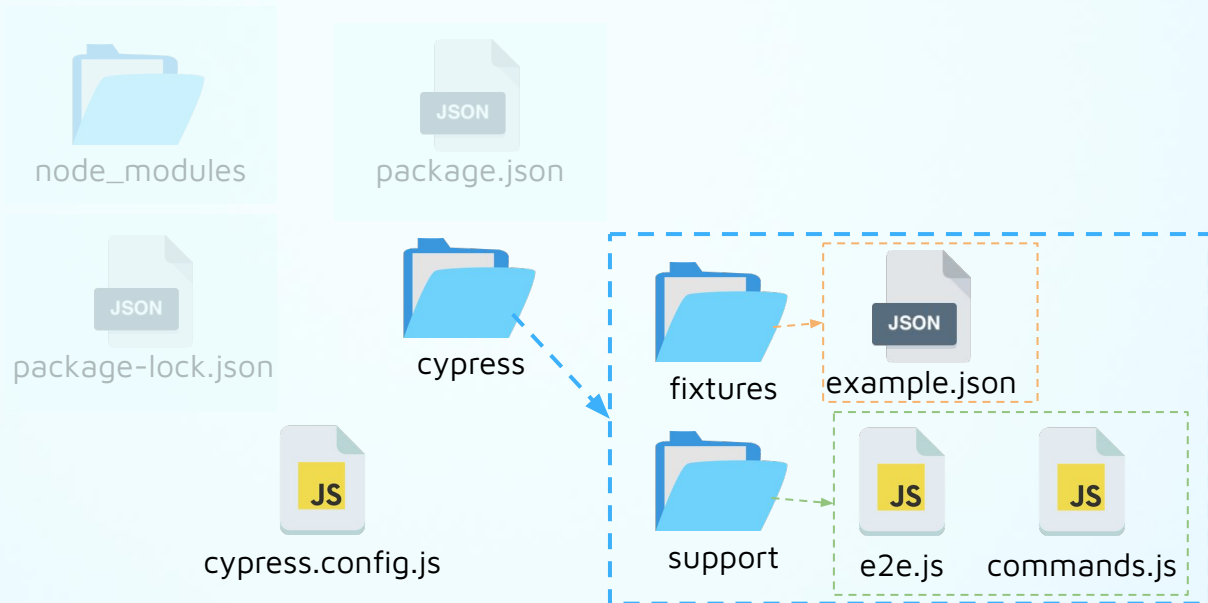
package-lock.json

Installation and setup

Step 5

Open cypress using the following command:

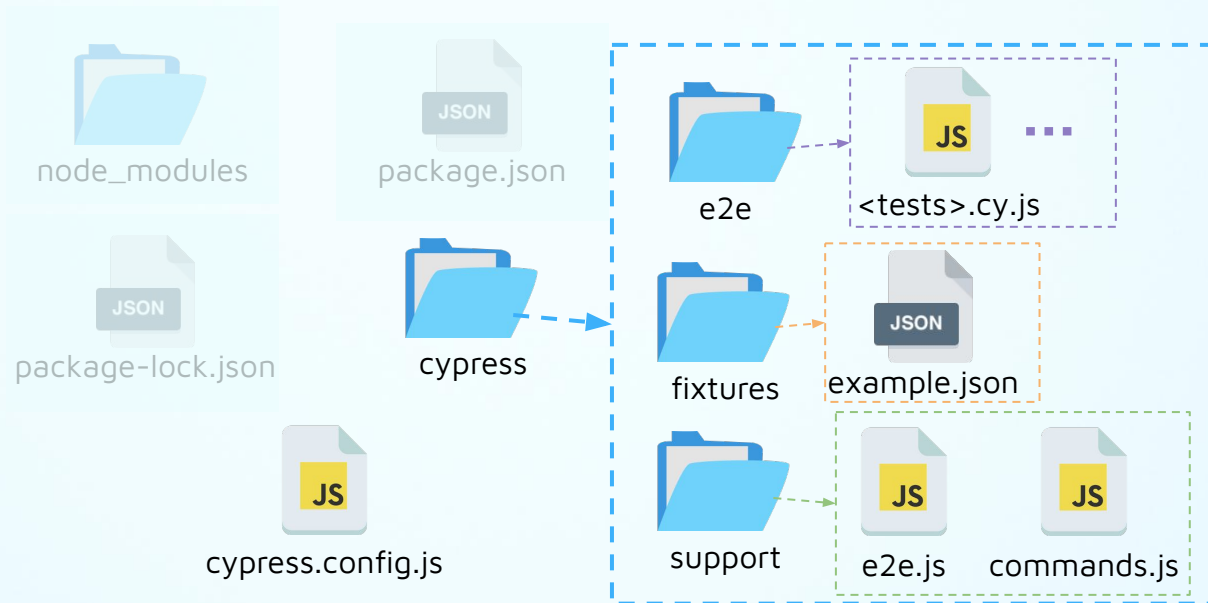
npx cypress open



Installation and setup

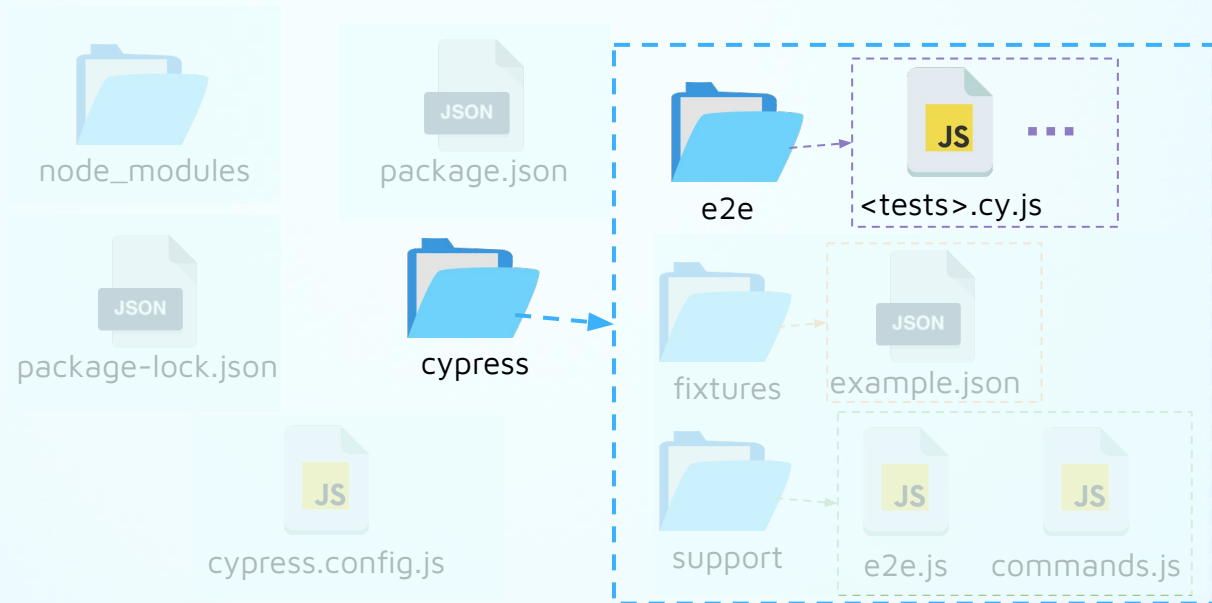
Step 6

Choose e2e and click "Scaffold example specs"



Cypress demo

Let's see how to use the cypress driver app and look at the sample test specs



Cypress demo

Cypress docs is 

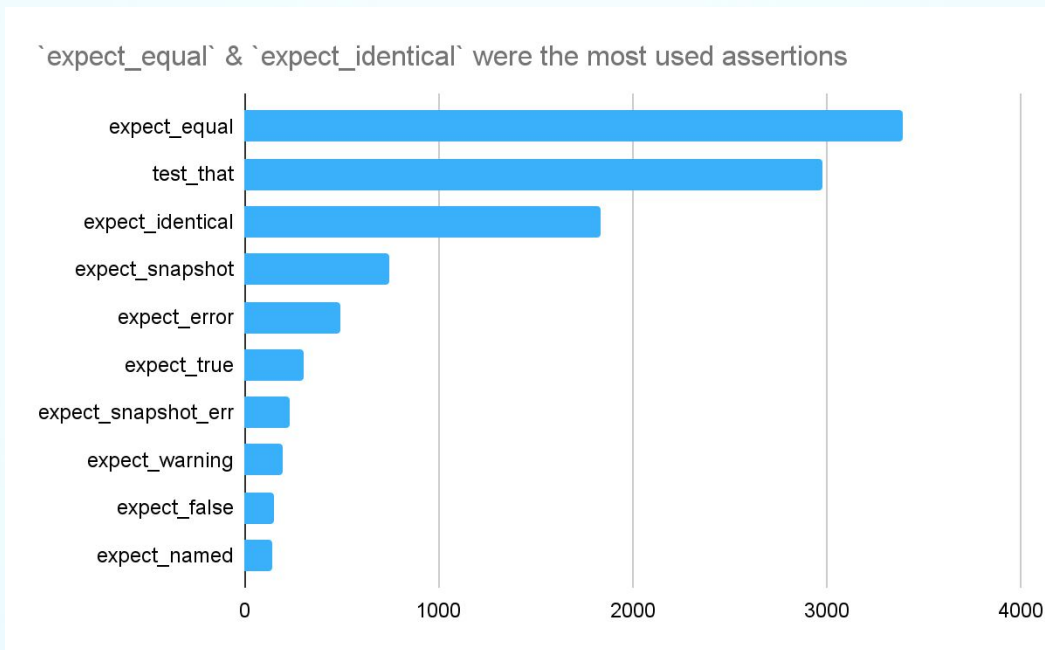
Cypress demo

80-20 cypress methods

- BDD syntax from mocha: describe(), it(), before(), after(), ...
- get()
- Actions like click(), dblclick(), type(), check(), ...
- should()



Frequency of testthat functions in these packages*

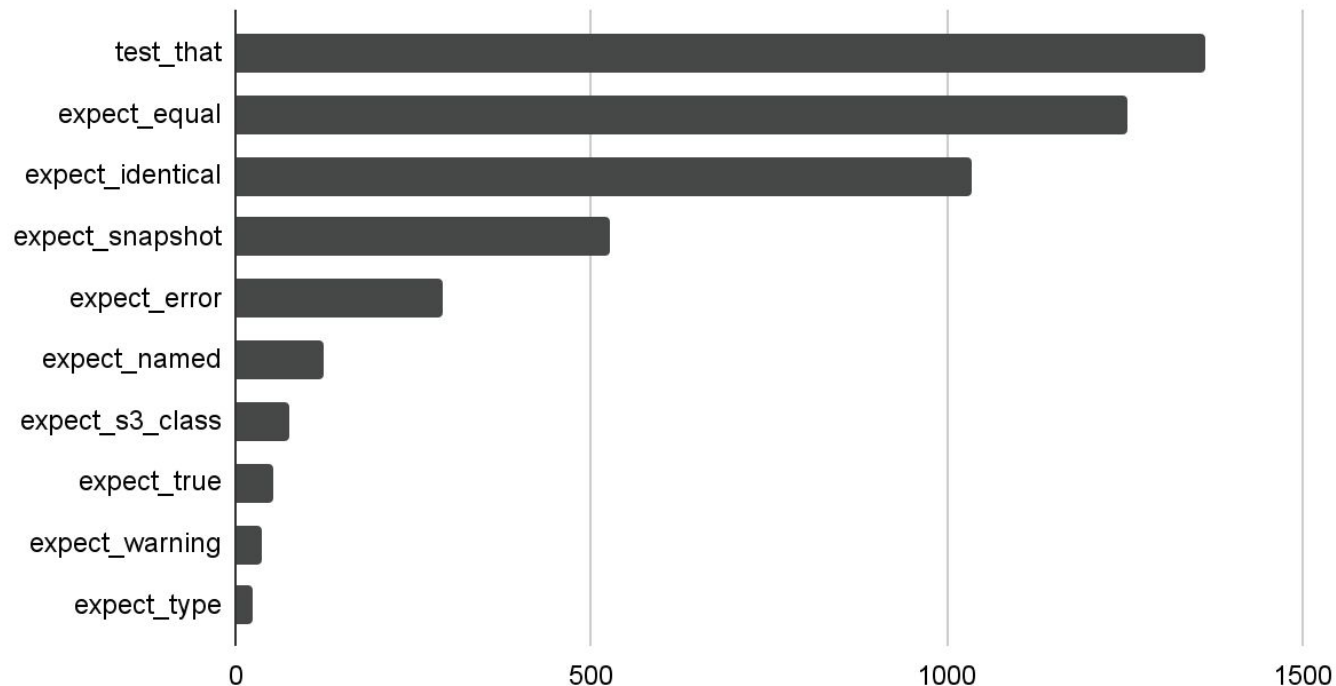


*From the main branch of the repositories as of 17th April 2024

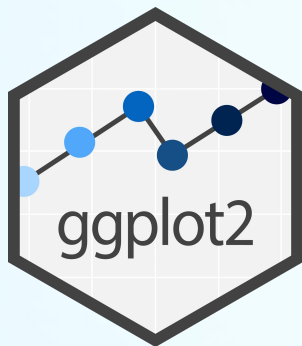
Frequency of testthat functions



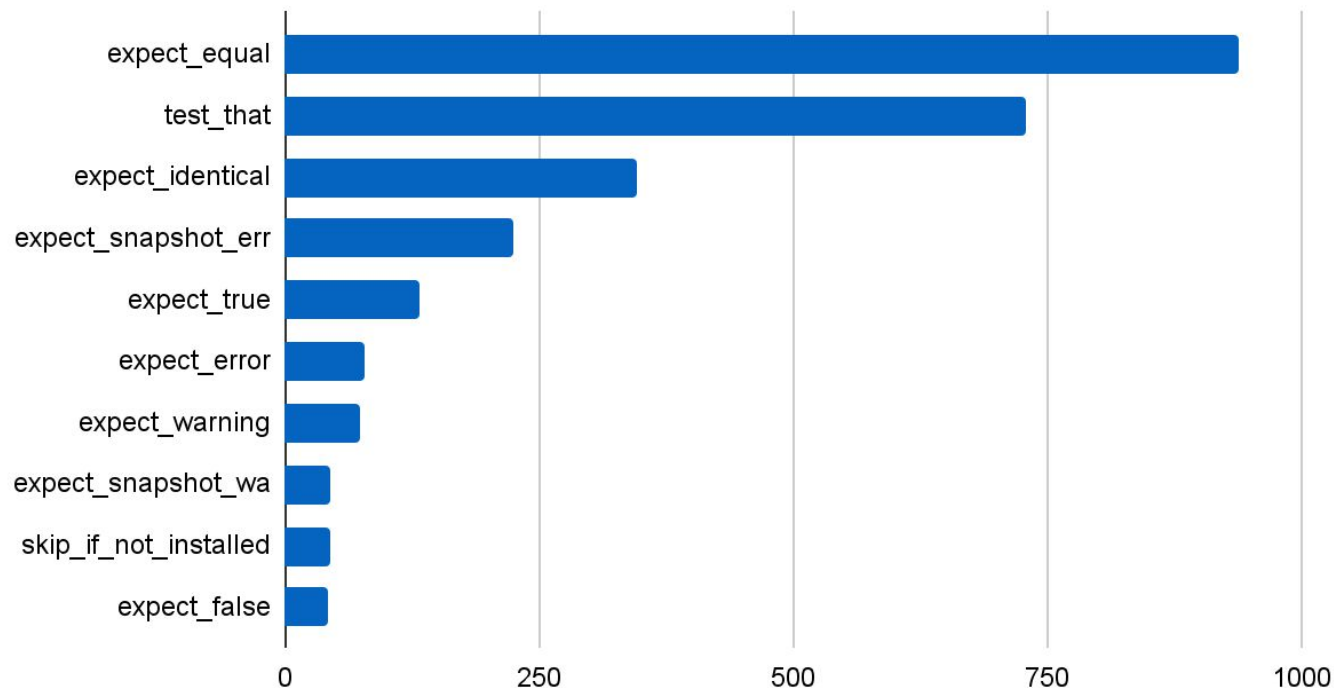
test functions used in {dplyr}



Frequency of testthat functions



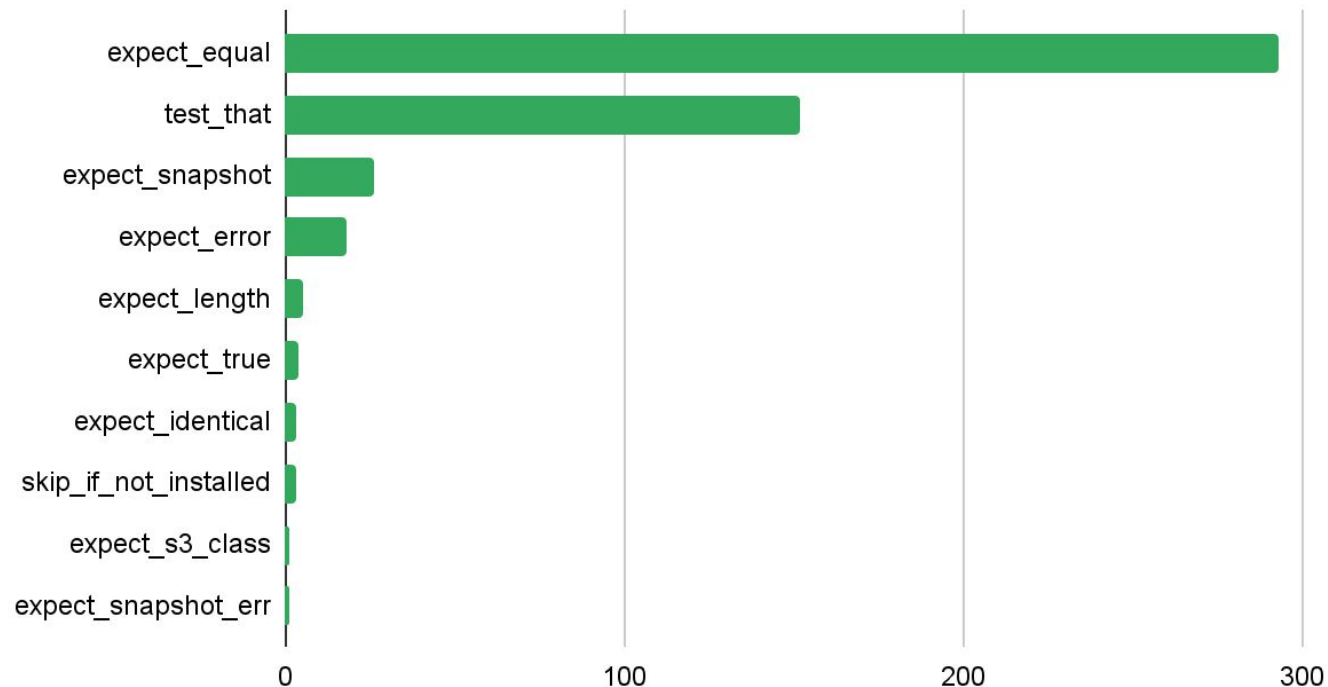
test functions used in {ggplot2}



Frequency of testthat functions



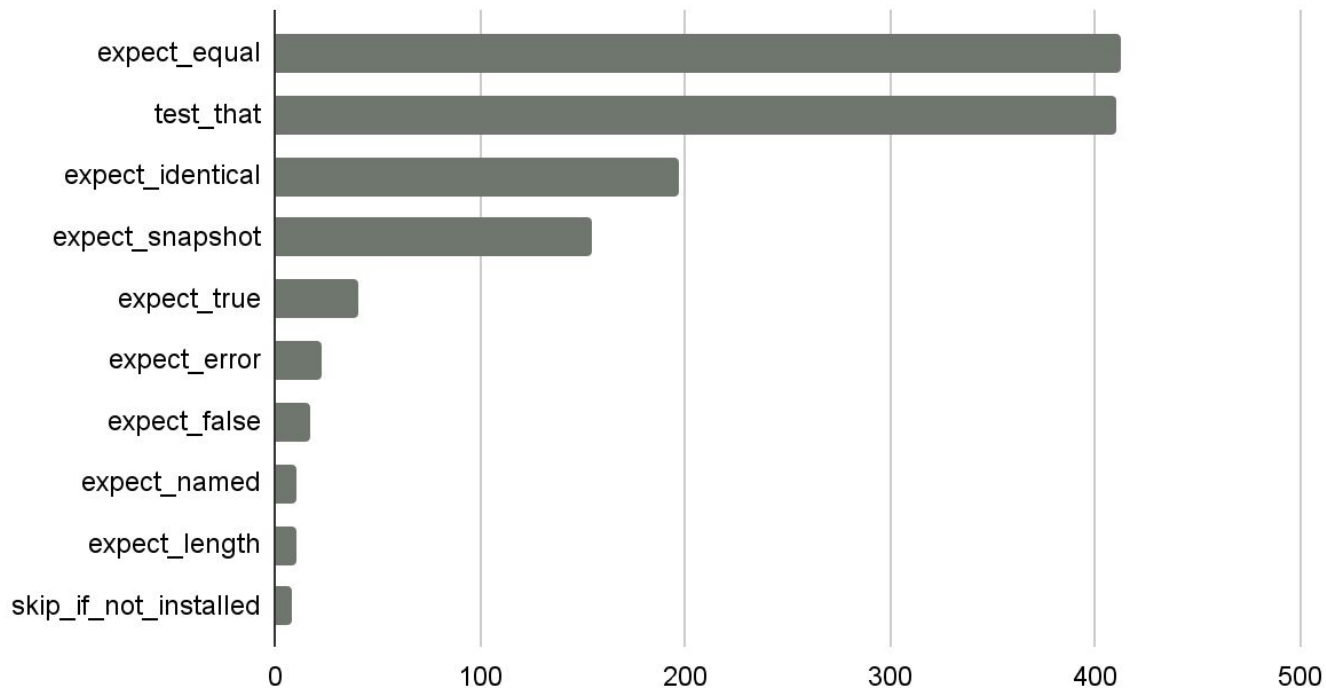
test functions used in {stringr}



Frequency of testthat functions



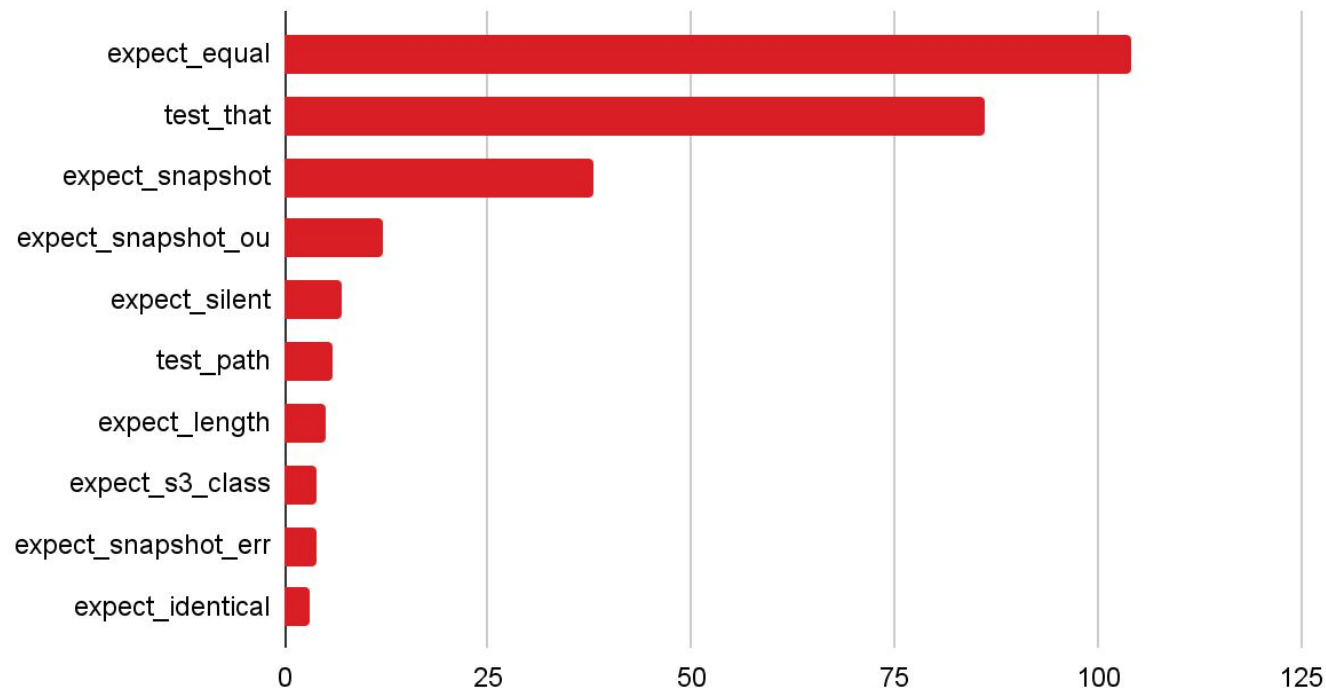
test functions used in {purrr}



Frequency of testthat functions



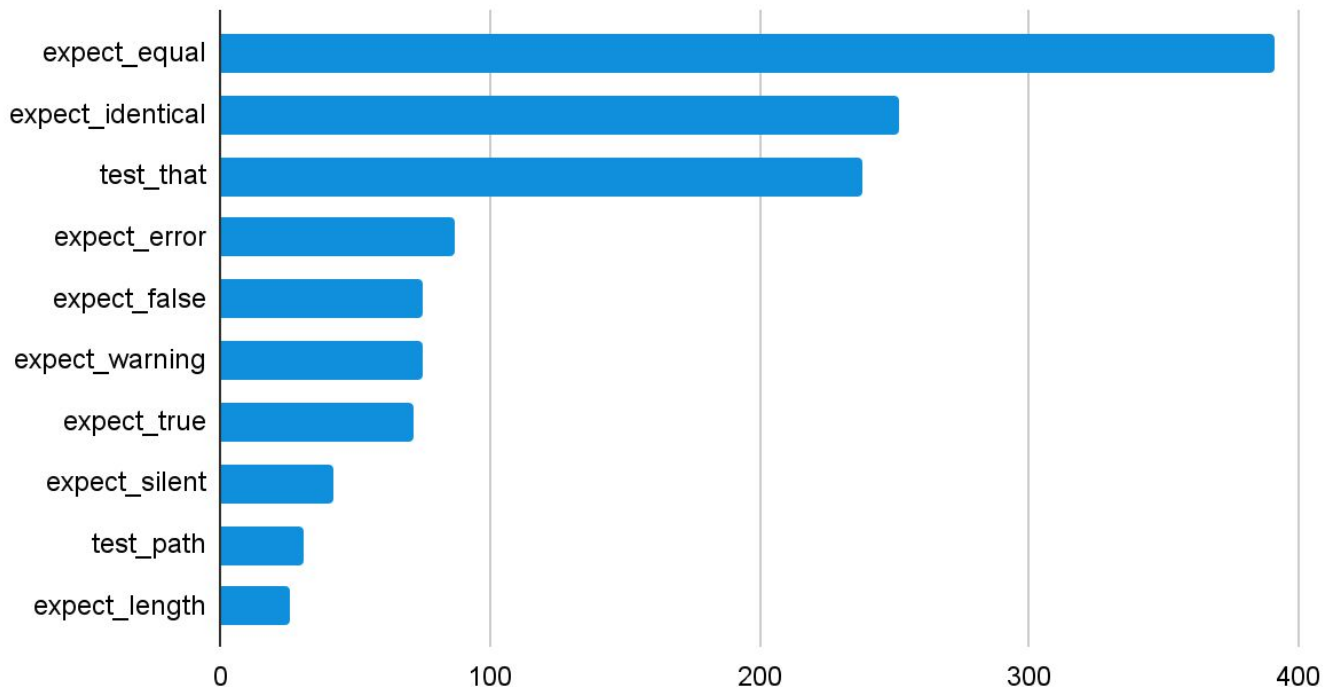
test functions used in {rvest}



Frequency of testthat functions



test functions used in {shiny}



Cypress demo

To run the test specs use the command:
npx cypress run

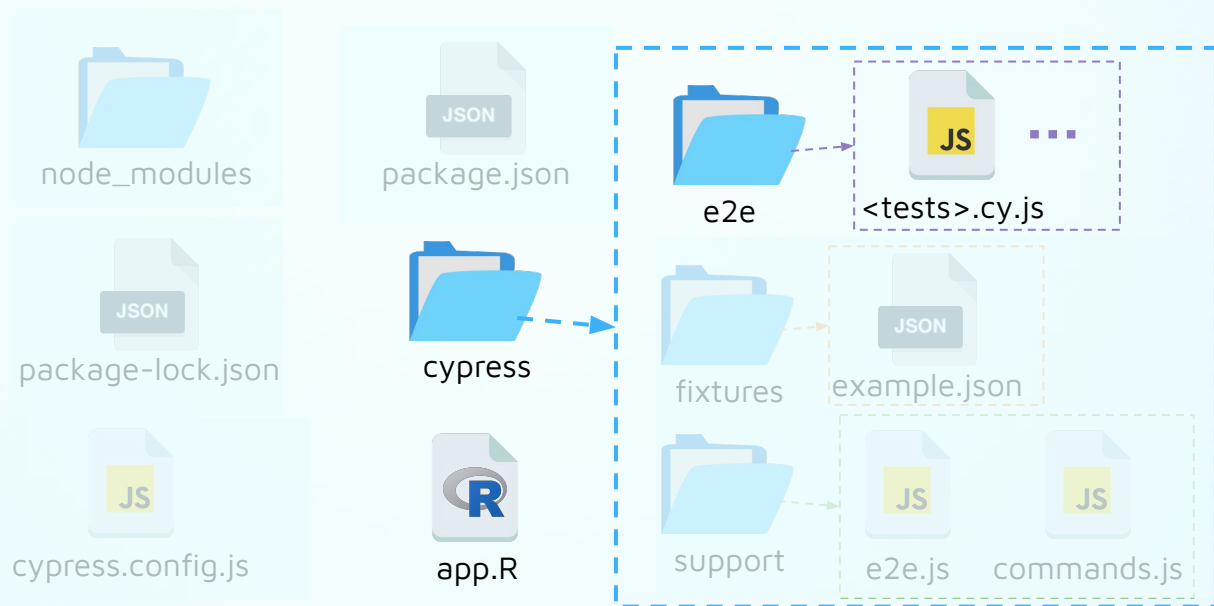
These common commands can be stored as scripts in **package.json**

```
"scripts": {  
  "test": "cypress run",  
  "open": "cypress open"  
}
```

Now the previous command can be executed by running:
npm run test

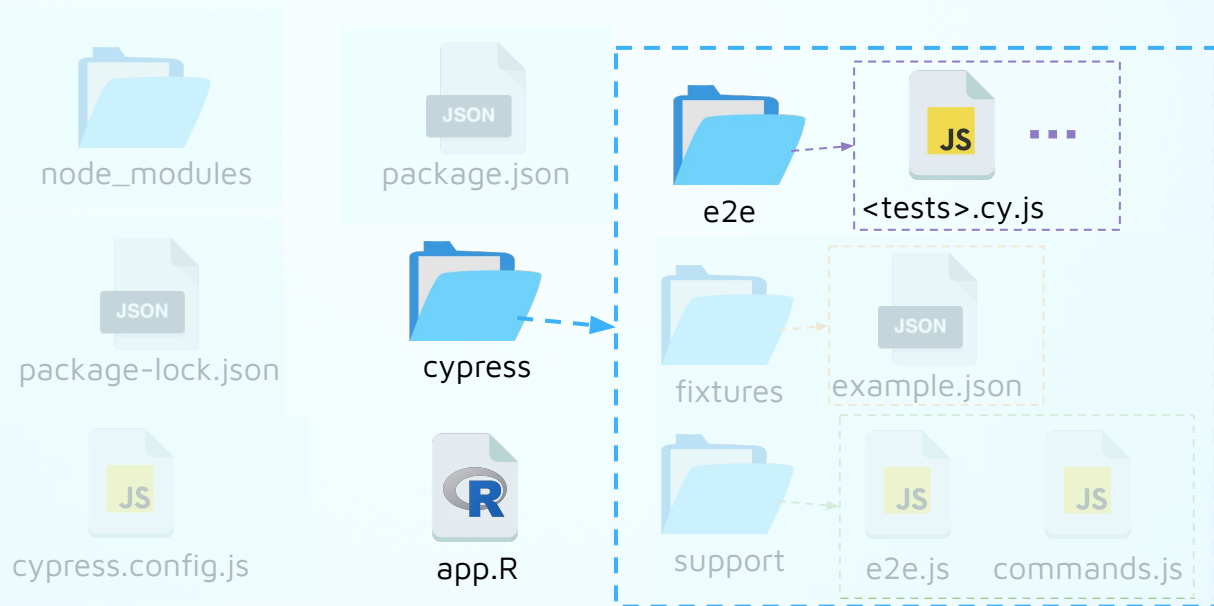
Cypress + Shiny demo

Run the shiny app separately and test using the local app url:
shiny::runApp(); npx cypress open



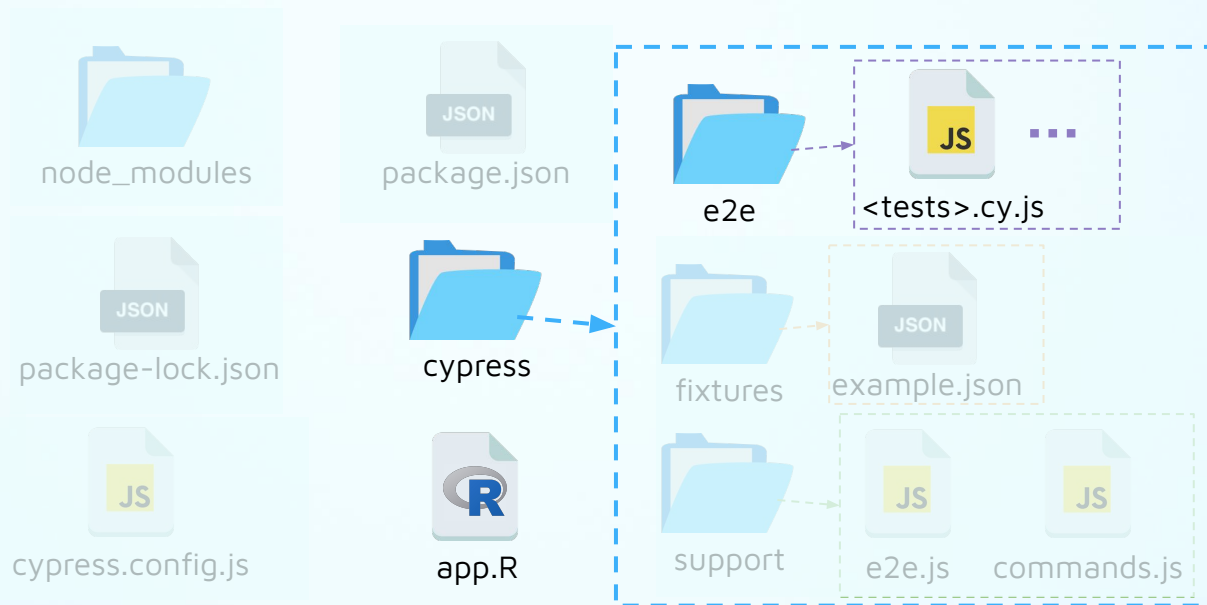
Cypress + Shiny demo

Shiny can be performing some reactive calculations under the hood.
And, cypress might think that the app is ready!
Use timeout.



Cypress + Shiny demo

Shiny errors or Validation errors can be caught by looking for certain css class in the body of the app



Cypress + Shiny demo

To run the app and test at the same time we can use a package called `start-server-and-test`. Which can be installed using:

npm install start-server-and-test

These common commands can be stored as scripts in **package.json**

```
"scripts": {  
  "run-app": "Rscript -e \"shiny::runApp(port = 5555)\"",  
  "run-cypress": "cypress run --project .",  
  "open-cypress": "cypress open --project .",  
  "test-e2e": "start-server-and-test run-app http://localhost:5555 run-cypress",  
  "test-e2e-interactive": "start-server-and-test run-app http://localhost:5555 open-cypress"  
}
```

Setting up CI for Cypress tests



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

```
name: End-to-end tests
on: push
jobs:
  cypress-run:
    runs-on: ubuntu-22.04
    steps:
      - name: Checkout
        uses: actions/checkout@v4
      - name: Setup Node
        uses: actions/setup-node@v4
        with:
          node-version: 20
      - name: Cypress run
        uses: cypress-io/github-action@v6
```

Setting up CI for Shiny-Cypress tests

```
...  
- name: Setup R  
  uses: r-lib/actions/setup-r@v2  
  with:  
    r-version: 4.1.2  
- name: Setup system dependencies  
  run: >  
      sudo apt-get update && sudo apt-get install --yes  
      libcurl4-openssl-dev  
- name: Install R packages  
  run: Rscript -e 'install.packages("shiny")'  
...
```

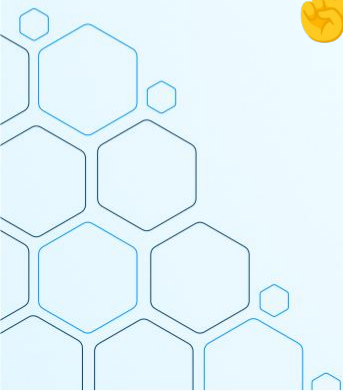
```
...  
- name: Cypress run  
  uses: cypress-io/github-action@v6  
  with:  
    build: npm install cypress --save-dev  
    start: npm run run-app  
    wait-on: "http://localhost:5555"  
    wait-on-timeout: 500
```

Alternatively, you can use rhino 🤪



Thank you

- 👉 Check the Cypress Scaffold example specs
- 👉 Create a simple shiny app with cypress



You can find the materials from this workshop at
<https://github.com/Appsilon/cypress-masterclass>

