

# การติดตั้ง cypress

- `cd /vue project`
- unzip cypress.zip และ copy "cypress" folder และ cypress.config.js ไว้ใน /vue project
- ติดตั้ง cypress
  - `npm install cypress --save-dev`
- เพิ่ม script ใน package.json
  - `"cypress:open": "cypress open"`

# /vue project



# package.json

```
{ } package.json X
{ } package.json > { } dependencies
1  {
2    "name": "fe",
3    "version": "0.0.0",
4    "private": true,
5    "type": "module",
6    "scripts": {
7      "dev": "vite",
8      "build": "vite build",
9      "preview": "vite preview",
10     "cypress:open": "cypress open"
11   },
12   "dependencies": {
13     "@tailwindcss/vite": "^4.1.4",
14     "pinia": "^3.0.1",
15     "tailwindcss": "^4.1.4",
16     "vue": "^3.5.13",
17     "vue-router": "^4.5.0"
18   },
19   "devDependencies": {
20     "@vitejs/plugin-vue": "^5.2.3",
21     "cypress": "^14.3.2",
22     "vite": "^6.2.4",
23     "vite-plugin-vue-devtools": "^7.7.2"
24   }
25 }
26
```

- เพิ่ม script

"cypress:open": "cypress open"

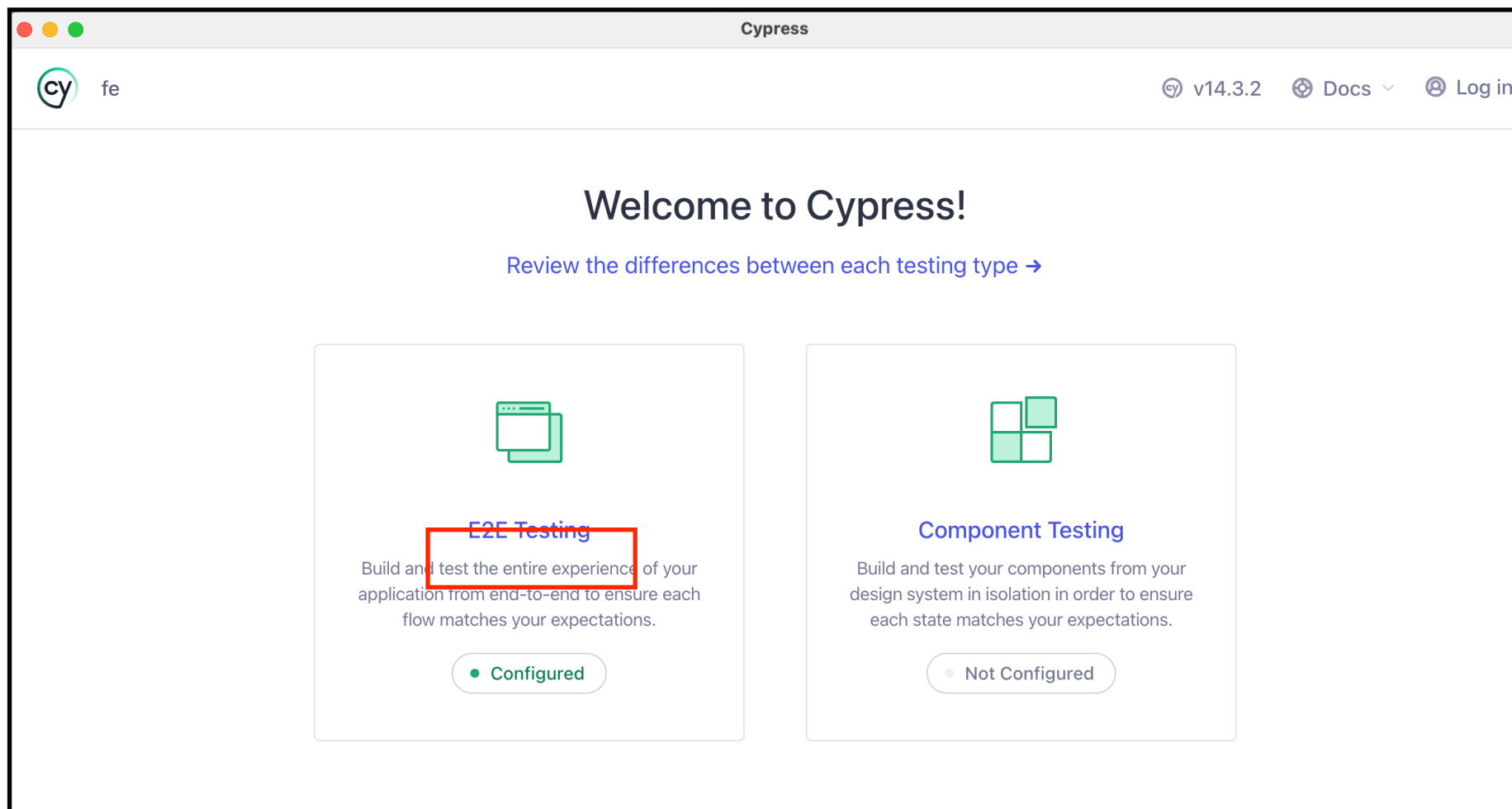
# cypress.config.js

- กำหนด baseUrl และ baseAPI
  - กรณียทดสอบในเครื่องใช้ localhost
  - กรณียทดสอบ VM และตอน UAT ให้แก้เป็น host ของ VM

```
JS cypress.config.js X
JS cypress.config.js > ...
1  import { defineConfig } from "cypress";
2
3  export default defineConfig({
4    e2e: {
5      specPattern: 'cypress/e2e/**/*.cy.{js,spec}.js,jsx,ts,tsx',
6      baseUrl: 'http://localhost:5173',
7      baseAPI: 'http://localhost:8080/itb-mshop',
8      setupNodeEvents(on, config) {
9        // implement node event listeners here
10      },
11    },
12  });
13
```

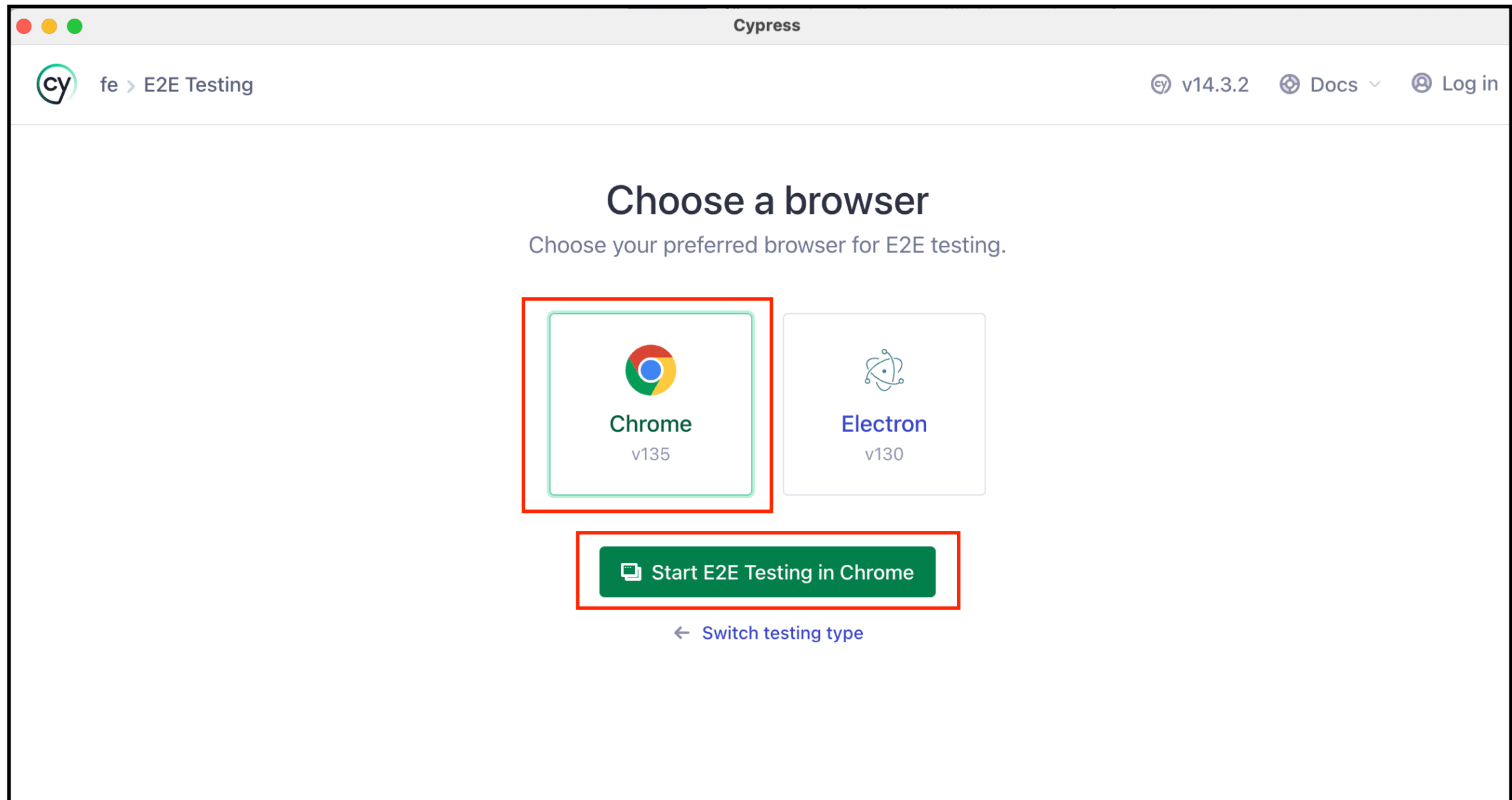
# Run cypress

- `npm run cypress:open`
- กด Configured



# Choose Browser

- เลือก Chrome และกด Start E2E Testing in Chrome



# Run test case

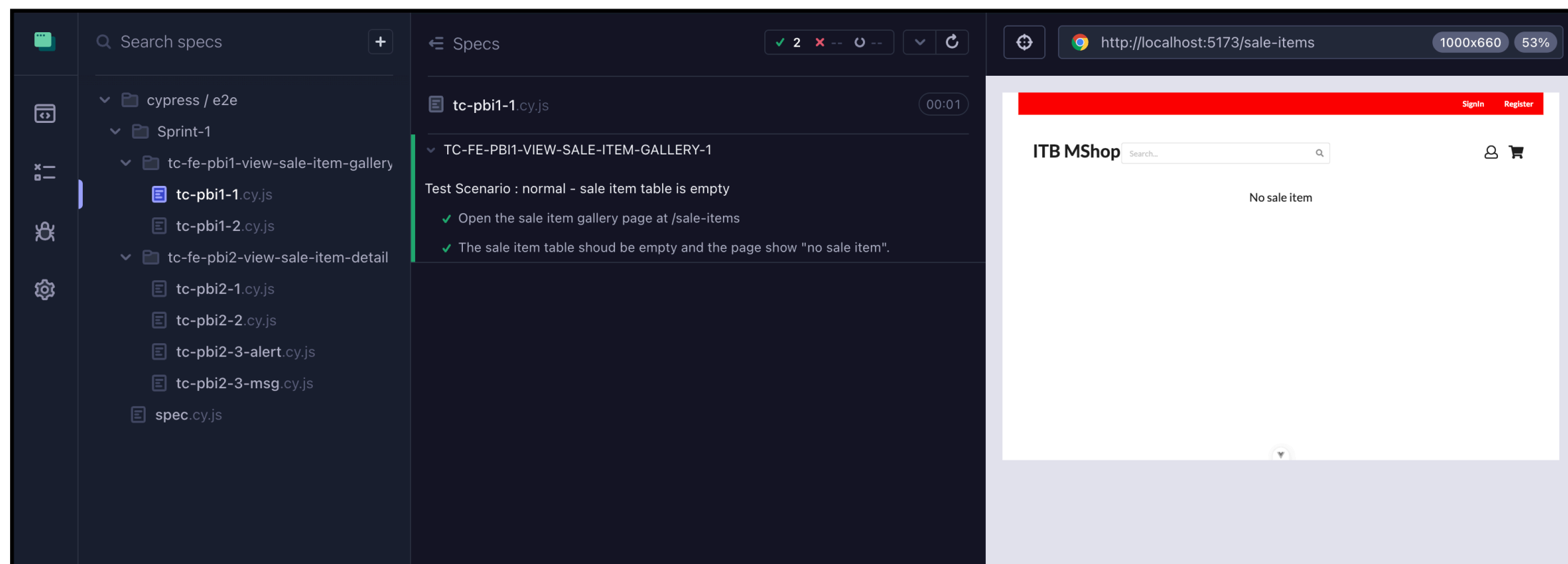
- เลือก Test Script ที่จะ Run ให้รัน script ตามลำดับ

The screenshot shows the Cypress web interface. On the left is a dark sidebar with navigation links: 'fe', 'Specs', 'Runs', 'Debug', and 'Settings'. The main area is titled 'Specs' and includes a search bar, a '+ New spec' button, and tabs for 'E2E specs' and 'Component specs'. A table lists test scripts under the 'E2E specs' tab. The script 'tc-pbi1-1.cy.js' is highlighted with a red box. The table columns are 'Last updated', 'Latest runs', and 'Average duration'.

	Last updated ?	Latest runs ?	Average duration ?
▼ cypress / e2e			
▼ Sprint-1			
▼ tc-fe-pbi1-view-sale-item-gallery			
tc-pbi1-1.cy.js	22 mins ago	--	--
tc-pbi1-2.cy.js	22 mins ago	--	--
▼ tc-fe-pbi2-view-sale-item-detail			
tc-pbi2-1.cy.js	22 mins ago	--	--
tc-pbi2-2.cy.js	22 mins ago	--	--
tc-pbi2-3-alert.cy.js	22 mins ago	--	--
tc-pbi2-3-msg.cy.js	22 mins ago	--	--
spec.cy.js	22 mins ago	--	--

# ผลการทดสอบ

- ถ้าขึ้นเครื่องหมายถูกทั้งหมด ผลทดสอบผ่าน แต่ถ้ามีกากบาทให้อ่านข้อความและลองแก้ไข
- เมื่อผ่าน test case แล้ว กดไป run test ต่อไป





# การทดสอบใน Sprint 1

- การทดสอบ pbi1-1 ไม่จำเป็นต้องลบข้อมูล ในการทดสอบจำลองว่าเกิดสถานะ 200 และข้อมูลเป็น [ ] จะตรวจสอบว่ามีข้อความ "no sale item" ไม่ได้ดูเรื่อง case sensitive
- การทดสอบ pbi2-3 กรณีที่ id ไม่มีอยู่ มีสองกรณีคือ
  - ถ้าใช้ alert ให้เรียก tc-pbi2-3-alert.cy.js ในการทดสอบโดยจะดูว่ามี alert และมีข้อความ 'The requested sale item does not exist. และมีการ redirect กลับไปหน้า /sale-items หรือไม่
  - ถ้าใช้ข้อความต้องมี class "itbms-message" ให้เรียก tc-pbi2-3-msg.cy.js ในการทดสอบโดยจะดูว่ามี "itbms-message" และมีข้อความ 'The requested sale item does not exist. และมีการ redirect กลับไปหน้า /sale-items หรือไม่
- ถ้ามีปัญหา รันไม่ผ่านให้ลองอ่าน cypress script เพื่อเข้าใจการทำงาน