

Cloud Native HW2

Test 1

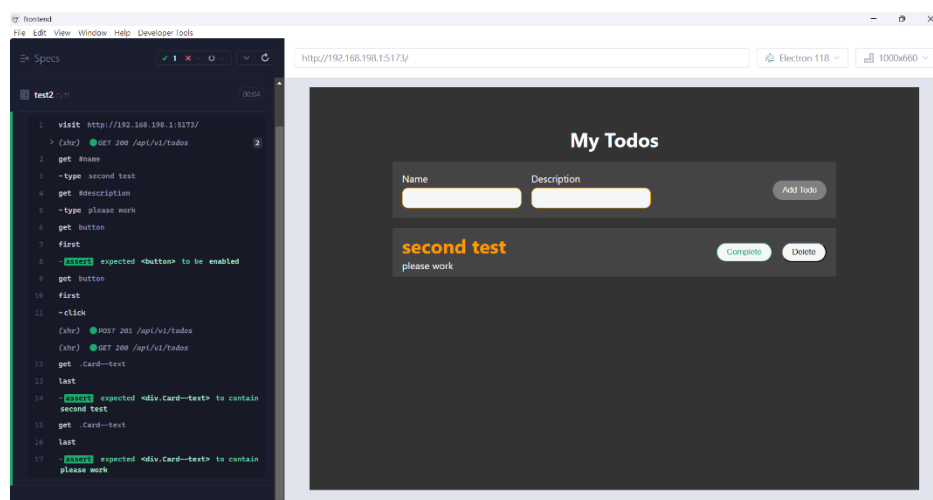
E2E 測試。使用 Cypress 測試程式在瀏覽器上是否運行正常，主要測試項目為新增一項 TODO。連進瀏覽器後，先輸入文字，確認按鈕是可以按的後按下按鈕，並確認 TODO 是否被新增。我這邊把 vite 指令的--host flag 加入，否則無法順利連到。

```
describe('Add TODO test', () => {
  it('should add successfully', () => {
    cy.visit('http://192.168.198.1:5173/')

    cy.get('#name').type("second test")
    cy.get("#description").type('please work')
    // check if i can click the button
    cy.get('button').first().should('be.enabled')
    cy.get('button').first().click()

    // get the first todo
    cy.get('.Card--text').last().should('contain', 'second test')
    cy.get('.Card--text').last().should('contain', 'please work')
  })
})
```

結果 - 瀏覽器成功顯示並通過測試



```
=====
(Run Starting)

Cypress:      13.7.1
Browser:      Electron 118
Node Version: v20.11.0 (C:\Program Files\nodejs\node.exe)
Specs:        1 found (test2.cy.ts)
Searched:     cypress/e2e/**/*.cy.{js,jsx,ts,tsx}

Running: test2.cy.ts (1 of 1)

Add TODO test
  ✓ should add successfully (1546ms)

1 passing (4s)

(Results)

Tests:      1
Passing:    1
Failing:    0
Pending:    0
Skipped:    0
Screenshots: 0
Video:      false
Duration:   3 seconds
Spec Ran:   test2.cy.ts

=====

(Run Finished)

Spec                                Tests  Passing  Failing  Pending  Skipped
✓ test2.cy.ts                       00:03    1        1        -        -        -
✓ All specs passed!                 00:03    1        1        -        -        -
```

Test 2

使用 K6 做流量測試，詳細設定如下方程式碼。每個使用者會分別新增一個 TODO，也就是 post 一次和 get 一次。我這邊設了兩個 threshold，分別是 http 的時間和成功率。根據實驗，若是沒有加入 post request 的 header，post 就會失敗。我最後使用 Chrome 的插件 [k6 Browser Recorder \(google.com\)](https://k6.io/docs/faq/browser-plugins/) 直接紀錄我的動作，完整還原使用瀏覽器執行動作時送的封包。

```

import { sleep, group } from 'k6'
import http from 'k6/http'

export const options = {
  //vus: 100,
  //duration: '30s',
  thresholds: {
    http_req_duration: ['p(95)<500'], // threshold for response time
    http_req_failed: ['rate<0.1'], // threshold for request failure rate
  },
  stages: [
    { duration: '10s', target: 10 },
    { duration: '10s', target: 50 },
    { duration: '10s', target: 100 },
  ],
};

export default function main() {
  let response
  let url = 'http://192.168.198.1:5173/'
  group('page_1 - ' + url, function () {
    response = http.post(
      url+'api/v1/todos',
      '{"id":"","name":"do homework","description":"test","status":false}',
      {
        headers: {
          'sec-ch-ua': '"Google Chrome";v="123", "Not:A-Brand";v="8", "Chromium";v="123"',
          'accept': 'application/json, text/plain, */*',
          'content-type': 'application/json',
          'sec-ch-ua-mobile': '?0',
          'sec-ch-ua-platform': '"Windows"',
          'origin': url,
          'sec-fetch-site': 'same-origin',
          'sec-fetch-mode': 'cors',
          'sec-fetch-dest': 'empty',
          'accept-encoding': 'gzip, deflate, br, zstd',
          'accept-language': 'zh-TW,zh;q=0.9,en-US;q=0.8,en;q=0.7',
        },
      },
    )
  })

  response = http.get( url + '/api/v1/todos', {
    headers: {
      'sec-ch-ua': '"Google Chrome";v="123", "Not:A-Brand";v="8", "Chromium";v="123"',
      'accept': 'application/json, text/plain, */*',
      'sec-ch-ua-mobile': '?0',
      'sec-ch-ua-platform': '"Windows"',
      'sec-fetch-site': 'same-origin',
      'sec-fetch-mode': 'cors',
      'sec-fetch-dest': 'empty',
      'accept-encoding': 'gzip, deflate, br, zstd',
      'accept-language': 'zh-TW,zh;q=0.9,en-US;q=0.8,en;q=0.7',
    },
  })

  // Automatically added sleep
  sleep(1)
}

```

測試結果 - 通過測試

```

$ k6 run ./k6test.js

      /\_/\
     /__  \\\
    (  oo  )
   /  _  \\\
  /_____\__\\

execution: local
script: ./k6test.js
output: -

scenarios: (100.00%) 1 scenario, 100 max VUs, 1m0s max duration (incl. graceful stop):
 * default: Up to 100 looping VUs for 30s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

page_1 - http://192.168.198.1:5173/
data_received.....: 1.5 MB 46 kB/s
data_sent.....: 1.1 MB 36 kB/s
group_duration.....: avg=31.62ms min=7.1ms med=12.76ms max=2.77s p(90)=28.65ms p(95)=36.61ms
http_req_blocked.....: avg=5.90ms min=2.01ms med=372.27us max=1.07s p(90)=687.01us p(95)=780.63us
http_req_connecting.....: avg=5.93ms min=0s med=298.88us max=1.07s p(90)=578.66us p(95)=662.48us
http_req_duration.....: avg=9.69ms min=2.13ms med=6.16ms max=2.76s p(90)=13.02ms p(95)=17.42ms
{ expected_response:true }...: avg=9.69ms min=2.13ms med=6.16ms max=2.76s p(90)=13.02ms p(95)=17.42ms
✓ http_req_failed.....: 0.00% < 0 x 2210
http_req_receiving.....: avg=131.96us min=37.88us med=124.96us max=191.22us p(90)=104.8us p(95)=213.78us
http_req_sending.....: avg=65.87us min=12.57us med=62.24us max=581.03us p(90)=109.59us p(95)=127.69us
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=8.49ms min=2.04ms med=5.96ms max=2.76s p(90)=13.64ms p(95)=17.2ms
http_req.....: 2210 69.0235/s
iteration_duration.....: avg=1.03s min=1s med=1.01s max=3.77s p(90)=1.02s p(95)=1.03s
iterations.....: 1105 34.51175/s
vus.....: 1 min=1 max=99
vus_max.....: 100 min=100 max=100

running (0m32.0s), 000/100 VUs, 1105 complete and 0 interrupted iterations
default / [=====] 000/100 VUs 30s

```

Note: 沒有塞 header 的話 post 都會失敗

```
> k6 run ./k6test.js

      MRR6
     .io

execution: local
script: ./k6test.js
output: -

scenarios: (100.00%) 1 scenario, 100 max VUs, 1m0s max duration (incl. graceful stop):
  * default: Up to 100 looping VUs for 30s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

page_1 - http://192.168.198.1:5173/
data_received..... 1.4 MB 45 kB/s
data_sent..... 307 kB 9.8 kB/s
group_duration..... avg=43.29ms min=6.68ms med=13.12ms max=5.61s p(90)=27.42ms p(95)=40.72ms
http_req_blocked..... avg=5.64ms min=1.12µs med=406.68µs max=1.08s p(90)=723.19µs p(95)=842.89µs
http_req_connecting..... avg=5.59ms min=0s med=338.92µs max=1.08s p(90)=626.22µs p(95)=727.74µs
✓ http_req_duration..... avg=15.89ms min=2.31ms med=6.02ms max=5.56s p(90)=14.26ms p(95)=18.45ms
  { expected_response:true }... avg=15.15ms min=3.67ms med=6.84ms max=5.56s p(90)=15.95ms p(95)=20.44ms
✗ http_req_failed..... 50.00% / 100s ✗ 100s
http_req_receiving..... avg=125.28µs min=21.57µs med=116.38µs max=1.05ms p(90)=181.62µs p(95)=215.22µs
http_req_sending..... avg=49.32µs min=5.28µs med=43.27µs max=267.45µs p(90)=88.84µs p(95)=104.69µs
http_req_tls_handshaking..... avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting..... avg=15.72ms min=2.2ms med=3.82ms max=5.56s p(90)=14.03ms p(95)=18.26ms
http_reqs..... 2190 94.238102/s
iteration_duration..... avg=1.04s min=1s med=1.01s max=6.61s p(90)=1.02s p(95)=1.04s
iterations..... 1095 35.129051/s
vus..... 5 min=1 max=99
vus_max..... 100 min=100 max=100

running (0m31.2s), 000/100 VUs, 1095 complete and 0 interrupted iterations
default / [=====] 000/100 VUs 30s
ERROR[0032] thresholds on metrics 'http_req_failed' have been crossed
```

Note: 我是在自己的筆電上測的，又有連到家裡路由器，所以有時候也會不太穩定。

```
> k6 run ./k6test.js

      MRR6
     .io

execution: local
script: ./k6test.js
output: -

scenarios: (100.00%) 1 scenario, 100 max VUs, 1m0s max duration (incl. graceful stop):
  * default: Up to 100 looping VUs for 30s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

page_1 - http://192.168.198.1:5173/
data_received..... 167 MB 4.5 MB/s
data_sent..... 459 kB 12 kB/s
group_duration..... avg=2.23s min=57.79ms med=1.63s max=7.87s p(90)=5.72s p(95)=5.63s
http_req_blocked..... avg=3.80ms min=272.62µs med=579µs max=1.06s p(90)=940.44µs p(95)=1.04ms
http_req_connecting..... avg=2.99ms min=225.2µs med=497.15µs max=1.06s p(90)=742.67µs p(95)=907.91µs
✗ http_req_duration..... avg=1.11s min=5.37ms med=741.74ms max=6.03s p(90)=2.79s p(95)=3.26s
  { expected_response:true }... avg=1.11s min=5.37ms med=741.74ms max=6.03s p(90)=2.79s p(95)=3.26s
✓ http_req_failed..... 0.00% / 0 ✗ 888
http_req_receiving..... avg=4.1ms min=36.48µs med=1.74ms max=36.58ms p(90)=12.61ms p(95)=19.45ms
http_req_sending..... avg=64.05µs min=21.16µs med=71.80µs max=4.24ms p(90)=117.05µs p(95)=136.52µs
http_req_tls_handshaking..... avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting..... avg=1.1s min=5.16ms med=725.0ms max=6.03s p(90)=2.78s p(95)=3.24s
http_reqs..... 888 24.101047/s
iteration_duration..... avg=3.23s min=1.05s med=2.63s max=8.87s p(90)=6.22s p(95)=6.63s
iterations..... 444 12.050924/s
vus..... 13 min=1 max=99
vus_max..... 100 min=100 max=100

running (0m36.8s), 000/100 VUs, 444 complete and 0 interrupted iterations
default / [=====] 000/100 VUs 30s
ERROR[0038] thresholds on metrics 'http_req_duration' have been crossed
```