

# 政府電子採購網爬取實作: 以高雄、台南建置系統為例

本簡報將深入探討如何使用 Playwright 工具爬取台灣政府電子採購網的資料，並實際動手操作。



投稿人：FU JIN CHEN

iStock  
Credit: The\_Pixel

快速回顧基礎

1

2

準備工作

實戰操作

3

4

未來應用探討

# Playwright API 與 Playwright Test 的差異



# 定義與用途

	Playwright API	Playwright Test
用途	自動化操作、爬蟲、模擬用戶行為	測試框架，用於撰寫功能測試、E2E 測試
適用場景	爬取數據、自動化任務	驗證系統功能是否正常運作
工具目標	執行自訂邏輯並靈活操作瀏覽器	快速生成測試腳本並進行斷言檢查

# Playwright API 範例

```
const { chromium } = require('playwright');

(async () => {
  const browser = await chromium.launch();
  const page = await browser.newPage();
  await page.goto('https://example.com');
  const content = await page.textContent('h1');
  console.log(content);
  await browser.close();
})();
```

## 特點：

- 更靈活，適合用於爬蟲、自動化操作。
- 不提供測試功能，需要手動撰寫爬取邏輯。

[Installation | Playwright Java](#)

# Playwright Test 範例

```
const { test, expect } = require('@playwright/test');

test('檢查標題', async ({ page }) => {
  await page.goto('https://example.com');
  const title = await page.textContent('h1');
  expect(title).toBe('Example Domain');
});
```

## 特點：

- 內建測試功能，可直接撰寫 `test()` 和 `expect()`。
- 適合進行端到端測試（E2E Testing）。



# 前言: 政府電子採購網的重要性及爬取需求

## 重要性

政府電子採購網是推動政府採購公開透明的重要平台，提供各項採購資訊，讓民眾可以隨時查詢了解政府採購狀況。

## 爬取需求

爬取政府電子採購網資料可以進行數據分析，找出採購趨勢、比較各縣市採購狀況，或開發相關應用工具。

# 準備工作: 了解採購網網站結構及資料





# 爬取過程: 模擬使用者行為

1

## 開啟網頁

使用 Playwright 程式碼開啟政府電子採購網

2

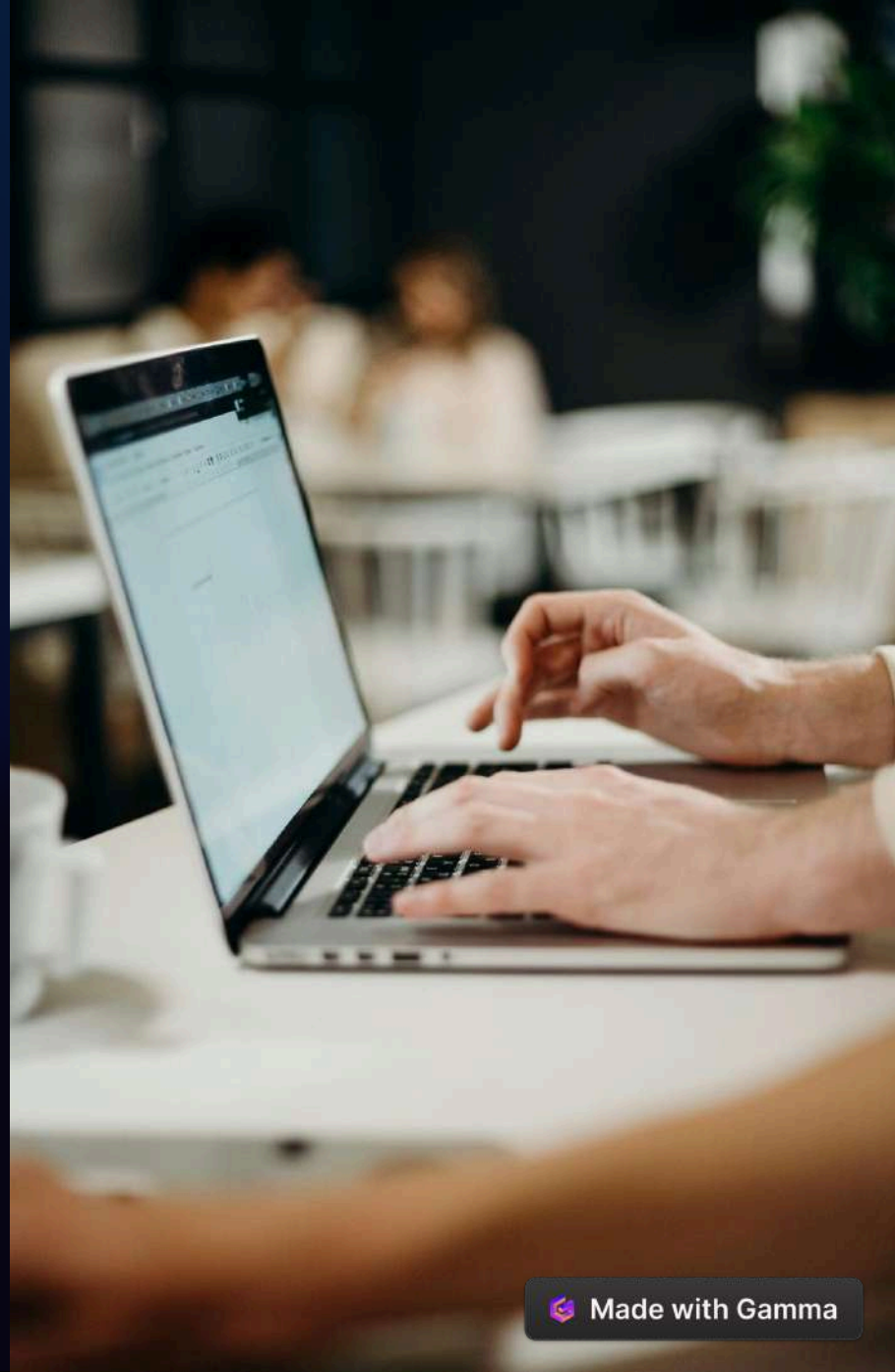
## 輸入關鍵字

模擬使用者在搜尋欄位輸入關鍵字

3

## 點擊搜尋按鈕

模擬使用者點擊搜尋按鈕



# 提取所需資料

1

## 選擇元素

使用 CSS 選擇器或 XPath 定位特定元素

2

## 取得文字

提取元素中的文字內容

3

## 處理日期

將日期格式化為一致的格式



# 處理異步數據加載

- `page.waitForResponse()`：允許我們等待符合特定條件的網路響應，適用於需要等待特定 API 請求完成的情境。

```
await page.waitForResponse(response =>  
  response.url().includes('/api/data') && response.status() === 200  
);
```



- `page.evaluate()`：允許在頁面上下文中執行 JavaScript 函數，並將結果帶回 Playwright 環境。

```
const data = await page.evaluate(() => {  
  const rows = Array.from(document.querySelectorAll('.data-row'));  
  return rows.map(row => row.textContent.trim());  
});  
console.log(data);
```

[评估 JavaScript | Playwright 中文网](#)

- `page.waitForSelector()`：等待指定的 DOM 元素出現在頁面上。

```
// 等待網頁載入完成後找到搜尋欄
```

```
await page.goto('https://example.com');
```

```
await page.waitForSelector('#search-bar');
```

```
// 填寫搜尋欄
```

```
await page.fill('#search-bar', 'Playwright');
```

# 實際應用案例:政府電子採購網

keyword: 「高雄」、「台南」、「建置」、「系統」



# 暖身:爬取簡單靜態網頁內容

## 題目描述

- 打開 [Fast and reliable end-to-end testing for modern web apps | Playwright](#)。
- 爬取頁面標題 (h1)，並打印在terminal上。

# 補充

```
const title = await page.locator('h1').textContent();  
const getbyrole = await page.getByRole('heading', { name: 'Playwright enables reliable' }).textContent();
```

場景	選擇器	原因
確定只有一個 <code>&lt;h1&gt;</code> ，內容無需篩選	<code>page.locator('h1')</code>	簡單直接，快速處理，適合爬取固定結構的網頁。
有多個 <code>&lt;h1&gt;</code> ，需要特定文字的標題	<code>page.getByRole('heading', { ... })</code>	更精準，能確保抓到正確的元素，適合測試或需要明確篩選條件的情境。



特性	<code>page.locator</code>	<code>getByRole</code>
選擇方式	使用 CSS 選擇器、XPath 等	使用 ARIA 標準的角色和名稱
語義化	不注重語義化，只關注結構	注重語義化，符合無障礙設計
適用範圍	適用於所有 DOM 元素	僅適用於具有 ARIA 角色的元素
可靠性	容易受到樣式或結構變化影響	更穩定，依賴角色和名稱
使用場景	精準操作結構化 DOM 或無語義標籤	測試無障礙性或語義化網頁

# 題目：爬取政府電子採購網資料

目標：爬取與「高雄」、「台南」、「建置」、「系統」相關的標案，並篩選數據輸出。

## 描述

- 打開<https://web.pcc.gov.tw/prkms/tender/common/bulletion/indexBulletin>
- 搜索台南or高雄，爬取結果中包含「建置」或「系統」的標案。  
(3.95|3.97)&(建置|系統)
- 將第一頁結果打印出來。

//即使 Playwright 默認會等待頁面「基本加載完成」，它無法預測動態內容（如 AJAX 請求）的完全加載狀態

```
await page.waitForLoadState('networkidle');
```

```
const projectNamesRaw = await page.locator('table td:nth-child(4) a').allTextContents();
```

// 處理空白並打印結果

```
const projectNames = projectNamesRaw.map(name => name.replace(/\s+/g, ' ').trim());
```

```
projectNames.forEach((name, index) => {  
  console.log(`標案 ${index + 1}: ${name}`);
```

```
});
```



# 進階題

目標：延續上題，爬取多頁結果並整合數據。

## 描述

- 增加「點擊下一頁」邏輯，到第五頁。
- 將所有頁面的數據整合並把結果保存為 JSON 檔案。

```
import fs from 'fs';

// 保存結果為 JSON 文件
const filePath = './projectNames.json';
fs.writeFileSync(filePath, JSON.stringify(projectNames, null, 2), 'utf8');
console.log(`成功保存到 ${filePath}`);
```



# 爬取資料後可以做什麼？

- 篩選標案相關資料。
- 保存為 CSV 或 JSON，便於後續數據處理。
- 匯入分析工具（如 Excel、Tableau）進行進一步的視覺化或報告生成。



# 心得與展望: 爬取技術的應用範圍及未來發展

爬取技術不僅可以應用於政府採購網，也可以應用於其他領域，例如市場分析、輿情監控、競爭對手分析等。未來隨著技術發展，爬取技術將會變得更加強大，能夠處理更複雜的網頁結構和資料格式，並提供更豐富的應用場景。