

WordPress CI

如何使用 **GitHub Action**
自動測試你的設定、插件或者模版

今次會講啲乜？

- 點解要自動測試？
- 試啲乜？
- 有乜麻煩位？
- 測試常用工具
- 自動化平台及工具
- WordPress CI
- 點樣一步步邁向自動測試
- 進階少少應用

今次唔會講啲乜？

- 點用 git
- 點用 composer
- PSR / Autoload
- Modern PHP

大家嚟到，有心聽，不妨參與多啲

- 唔明開聲問吧
- 嘗試照顧不同程度朋友
- 希望大家唔好嫌慢，但覺得太快請開聲

點解要做自動測試？

點解要做自動測試？

點解要做自動測試？

- AI 幻覺
- AI 爛貨 (AI Slop)
- 世人總要重覆錯誤 (Regression)
- 數齊啲貨，確保真係做咗

點解要做自動測試？

- AI 幻覺
- AI 爛貨 (AI Slop)
- 世人總要重覆錯誤 (Regression)
- 數齊啲貨，確保真係做咗

應該測試啲乜？

測試啲乜？

- **元件測試 (Unit Test)**
測試「零件」，比如 Function / Class
- **框架測試 (Framework Test)**
針對編程框架的銜接做測試，假設框架正常
- **需求測試 (Acceptance Test / End-to-end Test)**
盡量模擬真實環境，測試最終交付的功能性

測試啲乜？

- ~~元件測試 (Unit Test)~~

~~測試「零件」，比如 Function / Class (今日唔講)~~

- ~~框架測試 (Framework Test)~~

~~針對編程框架的銜接做測試，假設框架正常 (今日唔講)~~

- **需求測試 (Acceptance Test / End-to-end Test)**

盡量模擬真實環境，測試最終交付的功能性

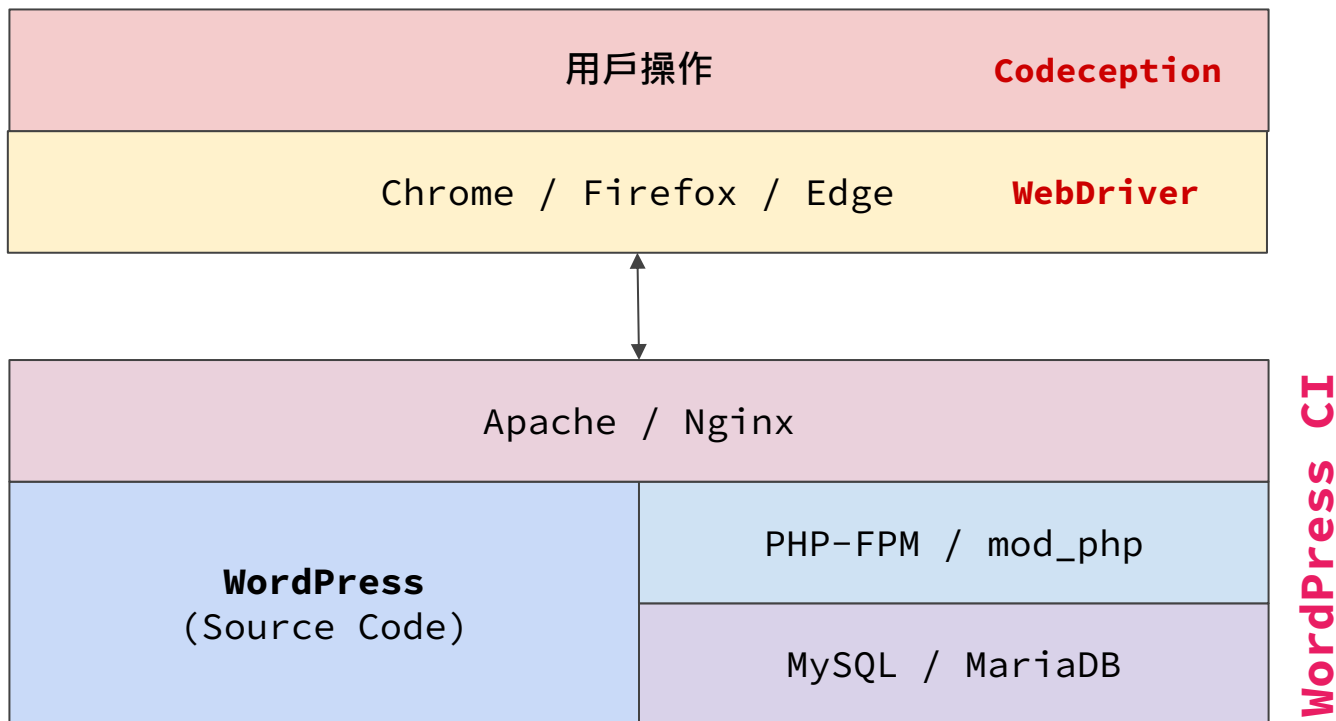
有乜嘢麻煩位？

有乜嘢麻煩位？

- WordPress 要好多嘢先行到
- 我啲 Plugin / Theme 要裝好晒 WordPress 之後再 activate
- 我有自己資料同設定
- 有唔同 PHP 版本我想試
- 點樣扮瀏覽器先？要扮用戶登入要 cookie 啲，仲有 Javascript...

正常一個 WordPress 要點先測試到？

正常一個 WordPress 要點先測試到？



仲有啲乜工具可以方便做自動測試？

仲有啲乜工具可以方便做自動測試？

本地

Docker

平台

Docker

GitHub Action



一步步邁向做自動測試？

一步步邁向做自動測試？

- 加入 Git 版本控制 (如果未用緊)
- 先寫本地測試
- 本地運行
- 加入 GitHub Actions
- 慢慢加入 Workflow，將本地測試在 GitHub 平台上跑

本地測試

PHPUnit

PHP Unit

- 老牌子
- 多人用
- Codeception 聯乘
- 有其他選擇：
 - PHPSpec
 - Kahlan

PHP Unit

- 時間有限，今日**唔試**比大家睇

WebDriver

WebDriver

- Selenium
- W3C 互聯網標準
2018 年起為推薦標準 (W3C Recommendation)
- 支援的瀏覽器：
 - Chrome / Chromium
 - Firefox
 - Edge
 - Safari
 - IE

Composer

Composer

- 管理相依關係
- PSR-4 autoload
- Modern PHP 常用
- 方便的方法分享腳本 (script)

Codeception + WebDriver

Demo

- Docker

<https://www.docker.com/>

- Docker Compose

<https://docs.docker.com/compose/install>

- WordPress CI

<https://github.com/impressible-wp/wordpress-ci/pkgs/container/wordpress-ci>

Demo

- Find Docker Container

```
docker ps
```

- Run Command in Docker Container

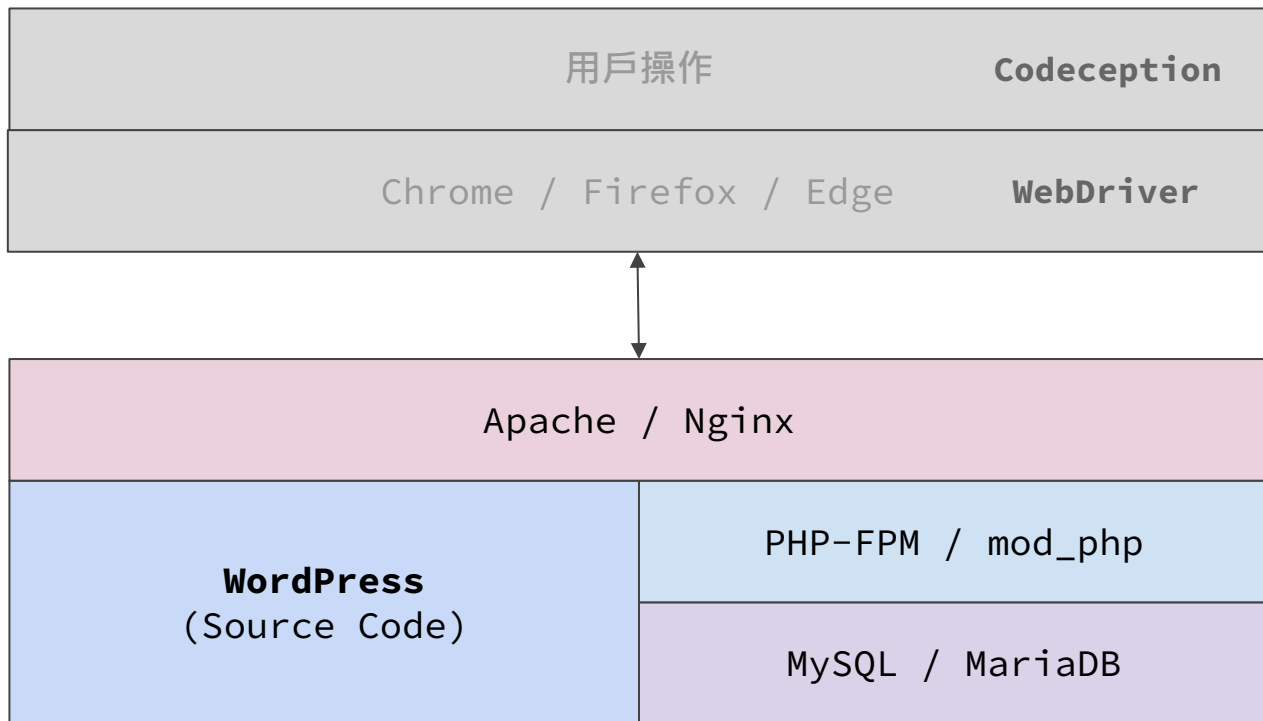
```
docker exec -it container-id command command-arg1 ...
```

- WordPress CLI (wp-cli)

<https://wp-cli.org/>

```
docker exec -it container-id wp ...
```

本地測試



WordPress CI



Demo

- WebDriver

<https://www.w3.org/TR/webdriver1/>

- Codeception

<https://codeception.com>

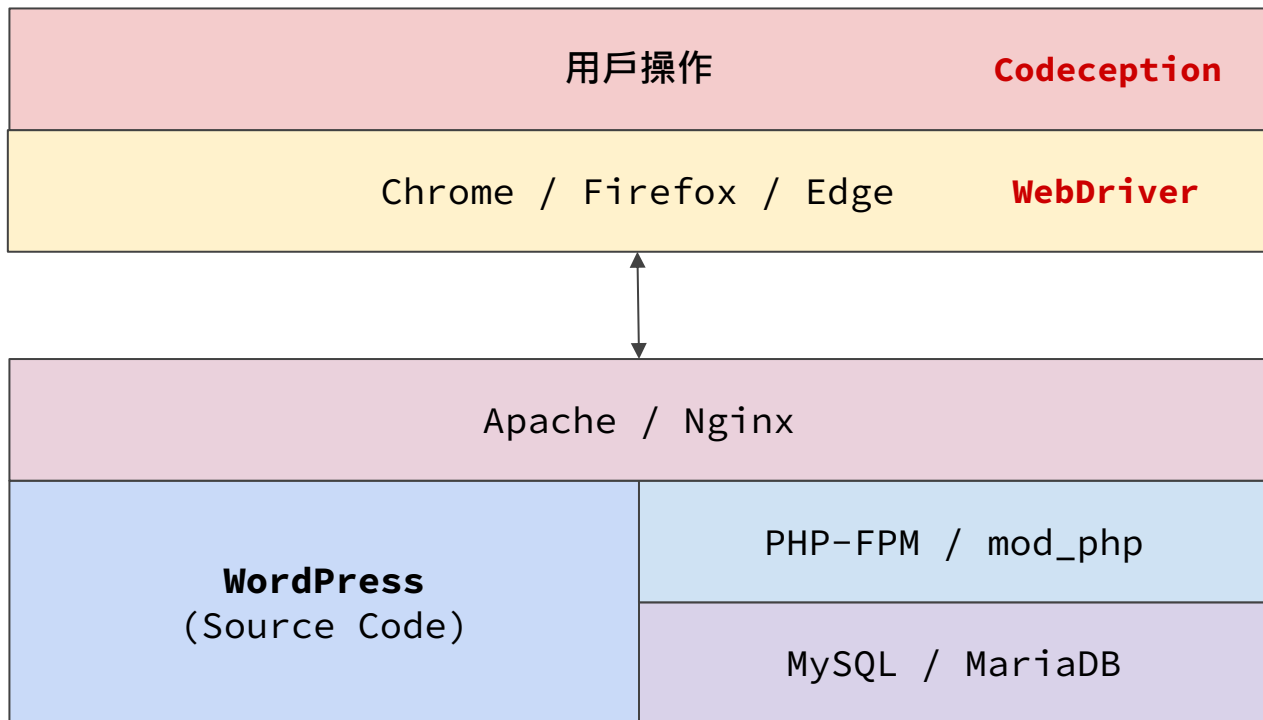
- Codeception: WebDriver module

<https://codeception.com/docs/modules/WebDriver>

注意

- Codeception 預設用 PHPBrowser 模擬瀏覽器
- 使用 **codecept init** 指令產生的 acceptance 用 PHPBrowser
- 要改 codeception.yml 與及 tests/Acceptance.yml
- 令佢使用 WebDriver module
- 不同的瀏覽器，需使用不同的 WebDriver 驅動程式，設定稍有不同

本地測試



WordPress CI

GitHub 自動測試

WordPress CI on GitHub Action

- WordPress CI on GitHub Marketplace

<https://github.com/marketplace/actions/wordpress-ci>

WordPress CI on GitHub Action

```
name: CI Test

on:
  pull_request:
    branches: [main]

jobs:
  test-action:
    name: Acceptance Test
    services:
      # The default database service name (hostname) for wordpress-ci is "mysql"
      # You can change it by setting the "db-host" input to the action.
      mysql:
        image: mariadb:12
        env:
          MARIADB_RANDOM_ROOT_PASSWORD: 'yes'
          # These are the database credentials used by wordpress-ci
          MARIADB_DATABASE: wordpress
          MARIADB_USER: username
          MARIADB_PASSWORD: password
        ports:
          # Simply for easier debug, not needed for wordpress-ci to work
          - 3306:3306
    steps:
      - name: Checkout repository
        uses: actions/checkout@v5

      - name: Setup Chrome for webdriver-based testing
```



WordPress CI on GitHub Action

```
services:
  # The default database service name (hostname) for wordpress-ci is "mysql"
  # You can change it by setting the "db-host" input to the action.
  mysql:
    image: mariadb:12
    env:
      MARIADB_RANDOM_ROOT_PASSWORD: 'yes'
      # These are the database credentials used by wordpress-ci
      MARIADB_DATABASE: wordpress
      MARIADB_USER: username
      MARIADB_PASSWORD: password
    ports:
      # Simply for easier debug, not needed for wordpress-ci to work
      - 3306:3306
```

WordPress CI on GitHub Action

```
steps:
  - name: Checkout repository
    uses: actions/checkout@v5

  - name: Setup Chrome for webdriver-based testing
    uses: browser-actions/setup-chrome@v2

  - name: Setup Chrome Driver for webdriver-based testing
    uses: nanasess/setup-chromedriver@v2

  - name: Start Chrome Driver process in background
    run: chromedriver --port=4444 >chromedriver.log 2>&1 &

  - name: Test
    uses: impressible-wp/wordpress-ci@v1
    with:
      # This folder will be binded to "wordpress-ci" container
```

WordPress CI on GitHub Action

```
- name: Test
  uses: impressible-wp/wordpress-ci@v1
  with:
    # This folder will be binded to "wordpress-ci" container
    # /var/www/html/wp-content/plugins/myplugin
    plugins: ./example/myplugin

    # Where in your repository do you want to start the test.
    # Depends on where you test scripts are.
    test-context: ./example/myplugin

  test-command: |

    # Running "server-side" commands in the Wordpress CI container
    wpci-cmd wp rewrite structure '%postname%/'
    wpci-cmd wp plugin activate myplugin

    # Some plugin might uses wp_rewrite and need to regenerate
    # .htaccess to properly work.
    wpci-cmd wp rewrite flush --hard

    # Your test may access the Wordpress CI's URL with this environment
    # variable
    echo "Wordpress is accessible here: $WORDPRESS_CI_URL"

    # Running "client-side" commands in GitHub Actions runner
    composer install
    composer run test
```

WordPress CI on GitHub Action

- Run Commands in WordPress CI container (i.e. "Server")
`wpci-cmd command command-arg1 ...`
- WordPress CLI (wp-cli)
`wpci-cmd wp ...`

點樣再進一步？



有圖有真相，截圖



模擬用戶登入



匯入資料庫做測試



錯誤處理

大致係咁，有乜想問？

持續交付

Continuous Delivery (CD)

WordPress CI

如何使用 **GitHub Action**
自動測試你的設定、插件或者模版

(完)