

# Laravel Dusk 使用及研究

# 教授的一點軼事



照片僅供參考，與實際人事物並無關連  
如有雷同純屬巧合

- 什麼是 Laravel Dusk
- 怎麼配置
- Dusk 的優缺點
- 怎麼測試

# 什麼是 Laravel Dusk



Laravel Dusk provides an expressive, easy-to-use **browser automation** and testing API.



用 Excel 管理需測試項目

# 關於我



- 懶惰
- 容易忘記事情
- 有人弄錯時容易發脾氣

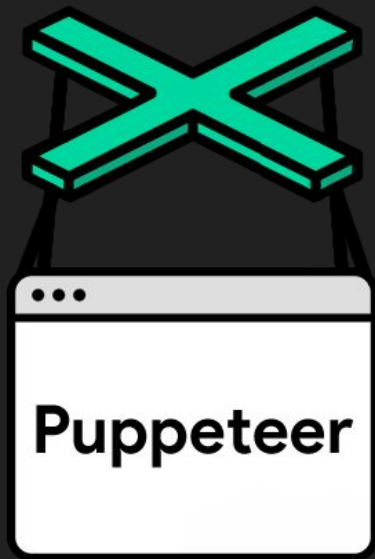


自動由電腦進行**瀏覽器測試**

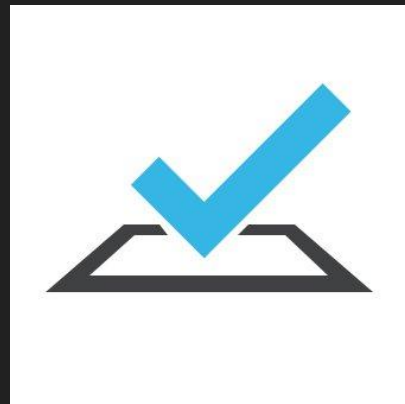
不需要 Selenium



不需要 puppeteer

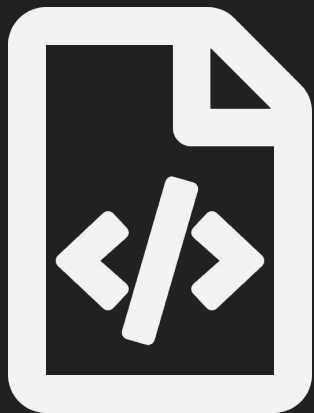


不需要 testcafe



其實是需要 selenium 的  
因為這個套件依賴於 facebookwebdriver 然後  
facebookwebdriver 包含 selenium



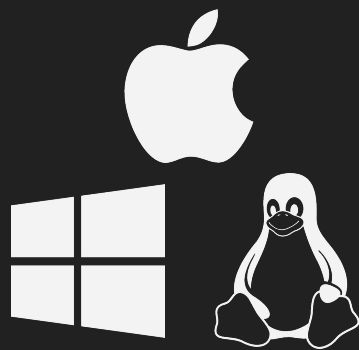


測試寫進去, bug 找出來, 工程師發大財

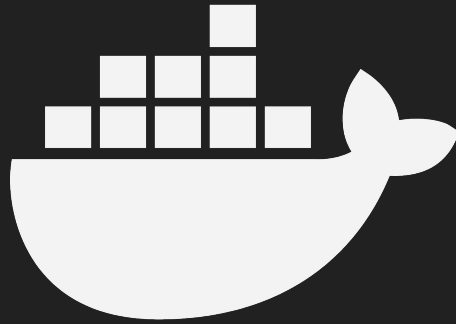
# 安裝 laravel dusk

<https://laravel.com/docs/5.7/dusk>





多數環境下可立刻使用



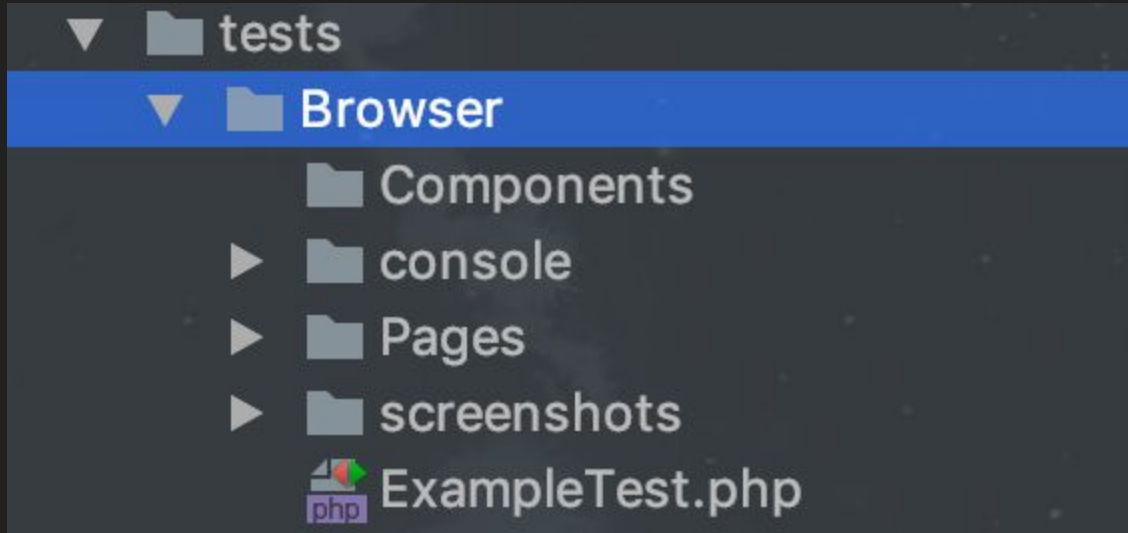
# laradock 環境下配置

<https://laradock.io/guides/#option-1-without-selenium>

<https://laradock.io/guides/#option2-dusk>



php artisan dusk:install



# 範例程式碼

```
class ExampleTest extends DuskTestCase
{
    public function testBasicExample()
    {
        $this->browse(function (Browser $browser) {
            $browser->visit('/')
                ->assertSee('Laravel');
        });
    }
}
```

# php artisan dusk(:fails)

```
$ php artisan dusk
```

```
PHPUnit 7.4.3 by Sebastian Bergmann and contributors.
```

```
.
```

```
1 / 1 (100%)
```

```
Time: 905 ms, Memory: 12.00MB
```

```
OK (1 tests, 1 assertions)
```

```
$ php artisan dusk:fails
```

```
PHPUnit 7.4.3 by Sebastian Bergmann and contributors.
```

```
.
```

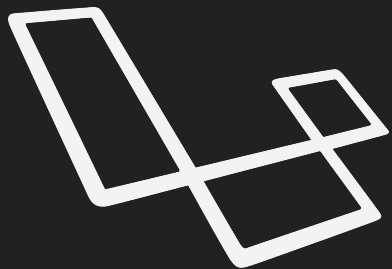
```
1 / 1 (100%)
```

```
Time: 948 ms, Memory: 12.12MB
```

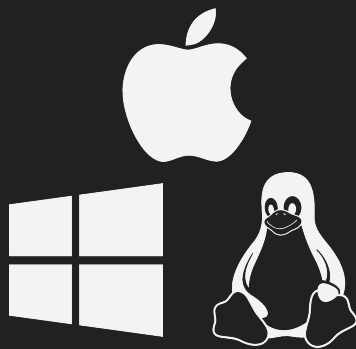
```
OK (1 tests, 1 assertions)
```



# Dusk 的優點



和 laravel 的整合很好

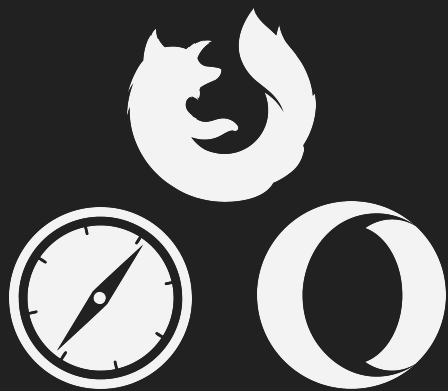


多數環境下可立刻使用

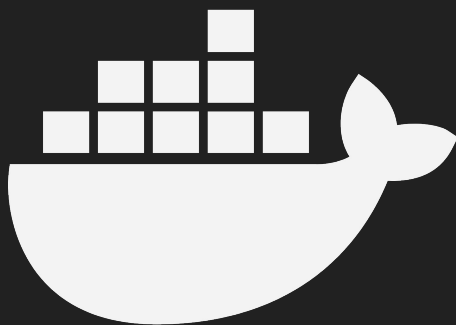


測試設置簡易

# Dusk 的缺點



不支援多瀏覽器



特定環境下設置困難



測試彈性較差

# 全部的 function(54)

assertTitle  
assertTitleContains  
assertUrlIs  
assertSchemeIs  
assertSchemeIsNot  
assertHostIs  
assertHostIsNot  
assertPortIs  
assertPortIsNot  
assertPathBeginsWith  
assertPathIs  
assertPathIsNot  
assertRouteIs  
assertQueryStringHas

assertQueryStringMissing  
assertFragmentIs  
assertFragmentBeginsWith  
assertFragmentIsNot  
assertHasCookie  
assertCookieMissing  
assertCookieValue  
assertPlainCookieValue  
assertSee  
assertDontSee  
assertSeeIn  
assertDontSeeIn  
assertSourceHas  
assertSourceMissing

assertSeeLink  
assertDontSeeLink  
assertInputValue  
assertInputValueIsNot  
assertChecked  
assertNotChecked  
assertRadioSelected  
assertRadioNotSelected  
assertSelected  
assertNotSelected  
assertSelectHasOptions  
assertSelectMissingOptions  
assertSelectHasOption  
assertValue

assertVisible  
assertPresent  
assertMissing  
assertDialogOpened  
assertEnabled  
assertDisabled  
assertFocused  
assertNotFocused  
assertVue  
assertVueIsNot  
assertVueContains  
assertVueDoesNotContain



我的财宝吗 想要的话就给你们



<http://bit.ly/DuskExample>

# tl;dr

- Laravel Dusk 可以用來進行前端整合測試
- Laravel Dusk 可以很簡單的用 laradock 配置
- 測試範例在 github 上有
  - <http://bit.ly/DuskExample>

谢谢大家~

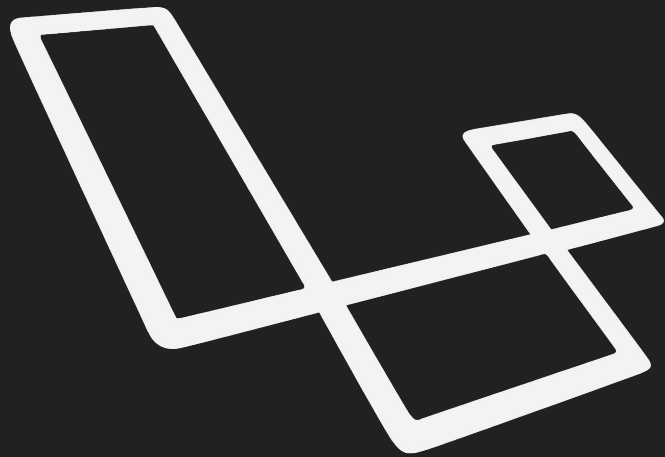




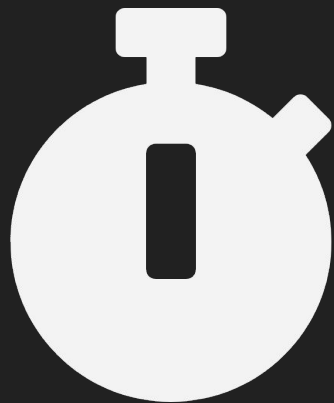
時間還沒到呢

才幾張投影片，你以為撐得完整場？





官網上可以看到的事情 建議上官網看



演講時間寶貴



welcome to the dungeon  
It's about to get wild!

- 什麼是 Laravel Dusk
- 怎麼配置
- Dusk 的缺點
- 怎麼測試
- **Laravel Dusk 研究**
  - packagist
  - composer.json
  - 怎麼找出文件上沒說的事情
  - Dusk 程式碼架構



套件的原始碼怎麼挖掘呢？

<https://packagist.org/packages/laravel/dusk>



<https://github.com/laravel/dusk>



# composer.json

- 宣告套件資訊
- 宣告套件相依套件
- 宣告套件開發時相依套件



## 在 composer.json 找到程式進入點

```
"providers": [  
    "Laravel\Dusk\DuskServiceProvider"  
]
```

# DuskServiceProvider

- `boot()`
  - 宣告三個 `Route::get()`
- `register()`
  - 環境是 production 時給予警告
  - 註冊指令
    - `dusk`
    - `dusk:fail`
    - `dusk:make`
    - `dusk:page`
    - `dusk:component`

# DuskServiceProvider

- `boot()`
  - 宣告三個 `Route::get()`
  - 文件上沒說的事情！



挖掘原始碼 可以發現很多想像不到的事情  
(有些時候幫助很大)

# Laravel\Dusk\Console\InstallCommand

- `handle()`
  - 建立資料夾
  - 處理 stubs
  - 建檔
  - 回傳完成
- `fire()`

# Laravel\Dusk\Console\(\Make|Page|Component)Command

- **handle()**
  - **use** Illuminate\Console\GeneratorCommand;
  - 根據 stubs 產生檔案
    - \Illuminate\Filesystem\Filesystem
    - **protected function** `replaceNamespace(&$stub, $name)`
      - pass by reference

# Laravel\Dusk\Console\DuskCommand

- `handle()`
  - 清除舊資料
    - `@unlink($file->getRealPath());`

# Laravel\Dusk\Console\DuskCommand

- `handle()`
  - `$this->withDuskEnvironment( \Closure $callback)`
    - `function tap($value, $callback = null)`



# Laravel\Dusk\Console\DuskCommand

- `handle()`
  - `$this->withDuskEnvironment( \Closure $callback)`
    - `$process = (new Process(array_merge(  
    $this->binary(), $this->phpunitArguments($options)  
)))->setTimeout(null);`
    - `return $process->run(function ($type, $line) {  
    $this->output->write($line);  
});`

# Symfony\Component\Process\Process

```
/**  
 * Process is a thin wrapper around proc_* functions to easily  
 * start independent PHP processes.
```

- `proc_open()`
- `proc_get_status()`
- `proc_close()`
- `proc_terminate()`

```
$this->binary()
```

```
return [PHP_BINARY, 'vendor/phpunit/phpunit/phpunit'];
```

# Laravel\Dusk\Console\DuskFailsCommand

```
extends DuskCommand
```

```
protected function phpunitArguments($options)
```

```
{
```

```
    return array_unique(array_merge(parent::phpunitArguments($options), [
```

```
        '--cache-result', '--order-by=defects', '--stop-on-failure',
```

```
    ]));
```

```
}
```

# 再看一眼

```
$ php artisan dusk
```

```
PHPUnit 7.4.3 by Sebastian Bergmann and contributors.
```

```
.                                     1 / 1 (100%)
```

```
Time: 905 ms, Memory: 12.00MB
```

```
OK (1 tests, 1 assertions)
```

```
$ php artisan dusk:fails
```

```
PHPUnit 7.4.3 by Sebastian Bergmann and contributors.
```

```
.                                     1 / 1 (100%)
```

```
Time: 948 ms, Memory: 12.12MB
```

```
OK (1 tests, 1 assertions)
```

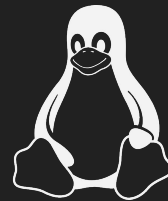
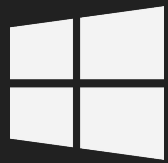
# tl;dr

- Dusk:(install|make|page|component)
  - 建立資料夾
  - 根據 stub 建檔
- Dusk
  - 用 Process 物件, 透過 php binary 執行 phpunit 進行測試
  - dusk:fail 就是 dusk 加上三個參數

犀利人妻  
最終回

台視

可是瑞凡



瀏覽器支援呢？



# Tests\DuskTestCase

```
/**  
 * Prepare for Dusk test execution.  
 * @beforeClass  
 * @return void  
 */  
public static function prepare()  
{  
    static::startChromeDriver();  
}
```

## Laravel\Dusk\Chrome\SupportsChrome::startChromeDriver

```
static::$chromeProcess = static::buildChromeProcess($arguments);  
static::$chromeProcess->start();  
static::afterClass(function () {  
    static::stopChromeDriver();  
});
```

Laravel\Dusk\Chrome\SupportsChrome::buildChromeProcess

```
return (new ChromeProcess(static::$chromeDriver))->toProcess($arguments);
```

# Laravel\Dusk\Chrome\ChromeProcess->\_\_construct

```
$this->driver = $driver;  
if (! is_null($driver) && realpath($driver) === false) {  
    throw new RuntimeException("Invalid path to Chromedriver [{$driver}].");  
}
```

# Laravel\Dusk\Chrome\ChromeProcess->toProcess

```
if ($this->driver) {  
    return $this->process($arguments);  
}  
if ($this->onWindows()) {  
    $this->driver = realpath(__DIR__.'../../bin/chromedriver-win.exe');  
    return $this->process($arguments);  
}  
$this->driver = $this->onMac()  
    ? realpath(__DIR__.'../../bin/chromedriver-mac')  
    : realpath(__DIR__.'../../bin/chromedriver-linux');  
return $this->process($arguments);
```

# Laravel\Dusk\Chrome\ChromeProcess->process

```
return (new Process(  
    array_merge([realpath($this->driver)], $arguments),  
    null,  
    $this->chromeEnvironment()  
));
```

# Laravel\Dusk\Chrome\ChromeProcess->onMac

```
return PHP_OS === 'Darwin';
```

Tests\DuskTestCase



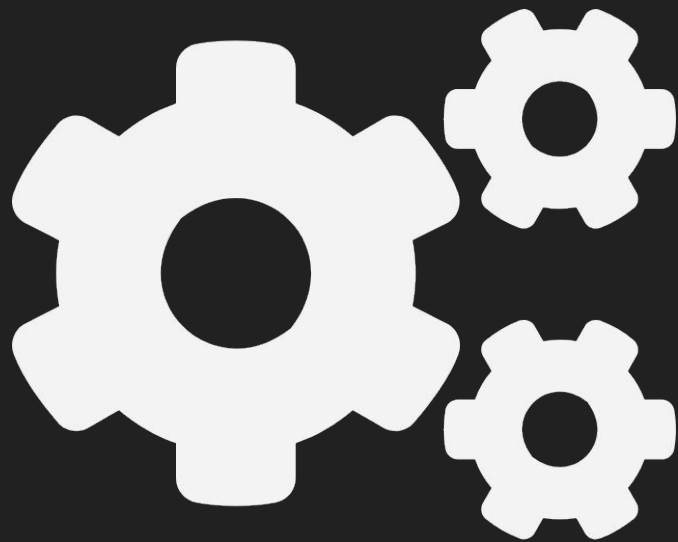
為什麼要這樣設計？





這個梗是要用幾次？

最後一次啦！



Dusk 裡面的功能呢？

## Laravel\Dusk\Concerns\MakesAssertions->assertTitle

```
PHPUnit::assertEquals(  
    $title, $this->driver->getTitle(),  
    "Expected title [{$title}] does not equal actual title [{$this->driver->getTitle()}]."  
);  
return $this;
```

# Laravel\Dusk\Concerns

**trait** InteractsWithAuthentication

**trait** InteractsWithCookies

**trait** InteractsWithElements

**trait** InteractsWithJavascript

**trait** InteractsWithMouse

**trait** MakesAssertions

**trait** MakesUrlAssertions

**trait** ProvidesBrowser

**trait** WaitsForElements

# Laravel\Dusk\Browser

```
public function visit($url)
public function visitRoute($route, $parameters = [])
public function on($page)
public function refresh()
public function back()
public function maximize()
public function resize($width, $height)
public function move($x, $y)
public function screenshot($name)
public function storeConsoleLog($name)
public function withinFrame($selector, Closure $callback)
public function within($selector, Closure $callback)
public function with($selector, Closure $callback)
public function onComponent($component, $parentResolver)
public function ensurejQueryIsAvailable()
public function pause($milliseconds)
```

```
public function quit()
public function tap($callback)
public function dump()
public function tinker()
public function stop()
```

# laradock # Dusk Dependencies:

```
RUN if [ ${INSTALL_DUSK_DEPS} = true ]; then \  
apt-get -y install zip wget unzip xdg-utils \  
  libxpm4 libxrender1 libgtk2.0-0 libnss3 libgconf-2-4 xvfb \  
  gtk2-engines-pixbuf xfonts-cyrillic xfonts-100dpi xfonts-75dpi \  
  xfonts-base xfonts-scalable x11-apps \  
&& wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb \  
&& dpkg -i --force-depends google-chrome-stable_current_amd64.deb \  
&& apt-get -y -f install \  
&& dpkg -i --force-depends google-chrome-stable_current_amd64.deb \  
&& rm google-chrome-stable_current_amd64.deb \  
&& wget https://chromedriver.storage.googleapis.com/${CHROME_DRIVER_VERSION}/chromedriver_linux64.zip \  
&& unzip chromedriver_linux64.zip \  
&& mv chromedriver /usr/local/bin/ \  
&& rm chromedriver_linux64.zip \  
;fi
```

ツ(ツ)ツ



好可怕呀！





枪在手，跟我走！



套件程式討論團



套件程式討論團



@ReccaChaoWebDev

努力鑽研 不要瞎忙





嘗試撰寫自己的套件



# tl;dr (again)

- 套件訊息可以找 `packagist`
- `packagist` 上可以找程式碼的 repo, 通常是 `github`
- 套件程式碼內可以看到很多前輩的寫法和架構設計
  - 見賢思齊
  - 於不疑處有疑
- `composer.json`
- `serviceProvider`
- `handle()` v.s. `fire()`
- 剩下的功能都在原始碼裡面
  - Assert 實作找 `Laravel\Dusk\Concerns\MakesAssertions`
  - Browser 功能實作找 `Laravel\Dusk\Browser`





我的财宝吗 想要的话就给你们

不只是 Dusk 還有千千萬萬的套件程式碼



投影片網址

谢谢大家~



這次是真的了