

drogon_ctl - 指令說明

[原文：ENG-12-drogon_ctl-Command.md](#)

當 **Drogon** 框架編譯安裝完成後，建議使用隨框架安裝的命令列工具 **drogon_ctl**（簡化指令為 **dg_ctl**）來建立第一個專案。使用者可依喜好選擇。

此工具主要功能是協助使用者快速建立各種 drogon 專案檔案。可用 **dg_ctl help** 指令查詢支援功能如下：

```
$ dg_ctl help
usage: drogon_ctl <command> [<args>]
commands list:
create          建立原始檔（詳見 'drogon_ctl help create'）
help            顯示說明訊息
version         顯示工具版本
press          壓力測試（詳見 'drogon_ctl help press'）
```

version 子指令

version 子指令用於顯示目前系統已安裝的 drogon 版本，例如：

```
$ dg_ctl version

  _
 _| | _ _ _ _ _ _ _ _ _ _
/ _ \ | ' _ / _ \ / _ \ | ' _ \
| ( _ | | | ( _ | ( _ | | | |
 \_,_| | \_,/ \_,/ | \_,/ | | |
      |___/

drogon ctl tools
version:0.9.30.771
git commit:d4710d3da7ca9e73b881cbae3149c3a570da8de4
compile config:-O3 -DNDEBUG -Wall -std=c++17 -I/root/drogon/trantor -
I/root/drogon/lib/inc -I/root/drogon/orm_lib/inc -I/usr/local/include -
I/usr/include/uuid -I/usr/include -I/usr/include/mysql
```

create 子指令

create 子指令用於建立各種物件，目前是 drogon_ctl 的主要功能。可用 **dg_ctl help create** 查詢詳細說明：

```
$ dg_ctl help create
使用 create 指令建立 drogon webapp 原始檔

用法: drogon_ctl create <view|controller|filter|project|model> [-options] <
物件名稱>
```

```

drogon_ctl create view <csp 檔名> [-o <輸出路徑>] [-n <命名空間>][--path-to-namespace] //由 csp 檔產生 HttpView 原始檔

drogon_ctl create controller [-s] <[namespace::]class_name> //建立 HttpSimpleController 原始檔

drogon_ctl create controller -h <[namespace::]class_name> //建立 HttpController 原始檔

drogon_ctl create controller -w <[namespace::]class_name> //建立 WebSocketController 原始檔

drogon_ctl create filter <[namespace::]class_name> //建立 filter 原始檔

drogon_ctl create project <project_name> //建立專案

drogon_ctl create model <model_path> //建立 model 類別

```

• 視圖建立

dg_ctl create view 用於由 csp 檔產生視圖原始檔，詳見 [視圖](#)。一般不需直接使用此指令，建議於 cmake 設定自動執行。範例：

```
dg_ctl create view UsersList.csp
```

• 控制器建立

dg_ctl create controller 用於建立控制器原始檔，支援三種控制器：

- 建立 HttpSimpleController：

```

dg_ctl create controller SimpleControllerTest
dg_ctl create controller webapp::v1::SimpleControllerTest

```

最後參數為類別名稱，可加命名空間。

- 建立 HttpController：

```

dg_ctl create controller -h ControllerTest
dg_ctl create controller -h api::v1::ControllerTest

```

- 建立 WebSocketController：

```
dg_ctl create controller -w WsControllerTest
dg_ctl create controller -w api::v1::WsControllerTest
```

• 過濾器建立

`dg_ctl create filter` 用於建立 filter 原始檔，詳見 [中介層與過濾器](#)。

```
dg_ctl create filter LoginFilter
dg_ctl create filter webapp::v1::LoginFilter
```

• 專案建立

建立新 Drogon 應用專案最佳方式為 `drogon_ctl` 指令：

```
dg_ctl create project ProjectName
```

執行後會於目前目錄建立完整專案目錄，名稱為 `ProjectName`，可直接於 build 目錄編譯（`cmake .. && make`），預設無業務邏輯。

專案目錄結構如下：

├─ build	編譯目錄
├─ CMakeLists.txt	cmake 設定檔
├─ cmake_modules	第三方函式庫查找腳本
│ └─ FindJsoncpp.cmake	
│ └─ FindMySQL.cmake	
│ └─ FindSQLite3.cmake	
│ └─ FindUUID.cmake	
├─ config.json	應用設定檔，詳見設定檔章節
├─ controllers	控制器原始檔目錄
├─ filters	過濾器原始檔目錄
├─ main.cc	主程式
├─ models	資料庫 model 目錄，model 原始檔建立詳見
11.2.5	
│ └─ model.json	
├─ tests	單元/整合測試目錄
│ └─ test_main.cc	測試入口
└─ views	視圖 csp 檔目錄，原始檔不需手動建立，編譯時自動預處理產生

• model 建立

使用 `dg_ctl create model` 建立資料庫 model 原始檔，最後參數為 model 目錄，該目錄需有 `model.json` 設定檔，指定資料庫連線與映射資料表。

例如於上述專案目錄建立 models：

```
dg_ctl create model models
```

執行後會提示檔案將直接覆寫，輸入 **y** 後產生所有 model 檔案。

其他原始檔如需引用 model 類別，請 include model 標頭檔，例如：

```
#include "models/User.h"
```

注意 models 目錄名稱需包含，以區分同專案多資料來源。詳見 [ORM](#)。

壓力測試

可用 **dg_ctl press** 指令進行壓力測試，選項如下：

- **-n num** 設定請求數（預設 1）
- **-t num** 設定執行緒數（預設 1），設為 CPU 數可達最大效能
- **-c num** 設定併發連線數（預設 1）
- **-q** 不顯示進度（預設顯示）

例如測試 HTTP 伺服器：

```
dg_ctl press -n1000000 -t4 -c1000 -q http://localhost:8080/  
dg_ctl press -n 1000000 -t 4 -c 1000  
https://www.domain.com/path/to/be/tested
```

下一步: [AOP 面向切面程式設計](#)