

控制器 - HttpSimpleController

原文：[ENG-04-1-Controller-HttpSimpleController.md](#)

你可以使用 `drogon_ctl` 命令列工具，快速產生基於 `HttpSimpleController` 的自訂控制器類別原始碼。
指令格式如下：

```
drogon_ctl create controller <[namespace::]class_name>
```

我們建立一個名為 `TestCtrl` 的控制器類別：

```
drogon_ctl create controller TestCtrl
```

執行後會產生兩個新檔案：`TestCtrl.h` 與 `TestCtrl.cc`。以下分別說明：

`TestCtrl.h`：

```
#pragma once
#include <drogon/HttpSimpleController.h>
using namespace drogon;
class TestCtrl:public drogon::HttpSimpleController<TestCtrl>
{
public:
    virtual void asyncHandleHttpRequest(const HttpRequestPtr &req,
                                         std::function<void (const
HttpRequestResponsePtr &)> &&callback)override;
    PATH_LIST_BEGIN
    //在此定義路徑
    //PATH_ADD("/path", "filter1", "filter2", HttpMethod1, HttpMethod2...);
    PATH_LIST_END
};
```

`TestCtrl.cc`：

```
#include "TestCtrl.h"
void TestCtrl::asyncHandleHttpRequest(const HttpRequestPtr &req,
                                       std::function<void (const
HttpRequestResponsePtr &)> &&callback)
{
    //在此撰寫應用邏輯
}
```

每個 `HttpSimpleController` 類別只能定義一個 `Http` 請求處理函式，並以虛擬函式覆寫。

URL 路徑到 handler 的路由（或稱映射）是透過巨集完成。可用 `PATH_ADD` 巨集新增多個路徑映射，所有 `PATH_ADD` 需寫在 `PATH_LIST_BEGIN` 與 `PATH_LIST_END` 之間。

第一個參數是要映射的路徑，後續參數則是對該路徑的限制。目前支援兩種限制：一是 `HttpMethod` 列舉型別，代表允許的 Http 方法；另一種是 `HttpFilter` 類別名稱。可任意組合這兩種限制，順序不限。Filter 詳見 [中介層與過濾器](#)。

使用者可將同一個 Simple Controller 註冊到多個路徑，也可在同一路徑註冊多個 Simple Controller（使用不同 HTTP 方法）。

你可以定義一個 `HttpResponse` 物件，然後用 `callback()` 回傳：

```
//在此撰寫應用邏輯
auto resp=HttpResponse::newHttpResponse();
resp->setStatusCode(k200OK);
resp->setContentTypeCode(CT_TEXT_HTML);
resp->setBody("你的頁面內容");
callback(resp);
```

上述路徑到 handler 的映射是在編譯期完成。事實上，drogon 框架也提供執行期完成映射的介面。執行期映射可讓使用者透過設定檔或其他介面動態新增或修改路由，無需重新編譯程式（為效能考量，`app().run()` 執行後禁止再新增任何控制器映射）。

下一步: [HttpController](#)