控制器 - 簡介

原文：ENG-04-0-Controller-Introduction.md

控制器在 Web 應用開發中非常重要。這裡我們會定義 URL、允許的 HTTP 方法、套用哪些過濾器，以及如何處理與回應請求。drogon 框架已協助處理網路傳輸、Http 協定解析等細節，開發者只需專注於控制器邏輯；每個控制器物件可擁有一個或多個處理函式（通常稱為 handler），其介面一般定義如下：

```
Void handlerName(const HttpRequestPtr &req,
                 std::function<void (const HttpResponsePtr &)> &&callback,
                 ...);
```

其中 `req` 是 Http 請求物件（以智慧指標包裝），`callback` 是框架傳給控制器的回呼函式物件，控制器產生回應物件（同樣以智慧指標包裝）後，透過 callback 傳給 drogon，框架會自動將回應內容送到瀏覽器。最後的 `...` 是參數列表，drogon 會依照對應規則將 Http 請求中的參數自動對應到函式參數，讓開發更方便。

這是一個非同步介面，可在其他執行緒完成耗時操作後再呼叫 callback。

drogon 有三種控制器類型：HttpSimpleController、HttpController 及 WebSocketController。使用時需繼承對應的類別模板。例如，自訂 HttpSimpleController 類別 "MyClass" 宣告如下：

```
class MyClass:public drogon::HttpSimpleController<MyClass>
{
public:
    //TestController(){}
    virtual void asyncHandleHttpRequest(const HttpRequestPtr &req,
                                        std::function<void (const
HttpResponsePtr &)> &&callback) override;

    PATH_LIST_BEGIN
    PATH_ADD("/json");
    PATH_LIST_END
};
```

控制器生命週期

註冊到 drogon 框架的控制器，整個應用程式執行期間最多只會有一個實例且不會被銷毀，因此可在控制器類別中宣告並使用成員變數。注意：控制器 handler 執行時可能處於多執行緒環境（當框架 IO 執行緒數大於 1），若需存取非暫存變數，請務必做好並行保護。

下一步: HttpSimpleController