

參考 - Request

[原文：ENG-08-5-Database-auto_batch.md](#)

範例中的 `HttpRequest` 型別指標（通常命名為 `req`）代表 drogon 所接收或送出的請求資料，下列為可操作此物件的常用方法：

- `isOnSecureConnection()`

`isOnSecureConnection()` 說明

判斷請求是否為 `https`。

`isOnSecureConnection()` 輸入

無。

`isOnSecureConnection()` 回傳

`bool` 型別。

- `getMethod()`

`getMethod()` 說明

取得請求方法。適用於同一處理函式支援多種方法時判斷。

`getMethod()` 輸入

無。

`getMethod()` 回傳

`HttpMethod` 請求方法物件。

`getMethod()` 範例

```
#include "mycontroller.h"

using namespace drogon;

void mycontroller::anyhandle(const HttpRequestPtr &req,
std::function<void (const HttpResponsePtr &)> &&callback) {
    if (req->getMethod() == HttpMethod::Get) {
        // do something
    } else if (req->getMethod() == HttpMethod::Post) {
        // do other something
    }
}
```

- `getParameter(const std::string &key)`

`getParameter()` 說明

依識別字取得參數值。GET 與 POST 請求行為略有不同。

`getParameter()` 輸入

參數識別字（string 型別）。

`getParameter()` 回傳

參數內容（string 型別）。

`getParameter()` 範例

GET 範例：

```
#include "mycontroller.h"
#include <string>

using namespace drogon;

void mycontroller::anyhandle(const HttpRequestPtr &req,
    std::function<void (const HttpResponsePtr &)> &&callback) {
    // https://mysite.com/an-path/?id=5
    std::string id = req->getParameter("id");
    // 或
    long id = std::strtol(req->getParameter("id"));
}
```

POST 範例：

```
#include "mycontroller.h"
#include <string>

using namespace drogon;

void mycontroller::loginHandle(const HttpRequestPtr &req,
    std::function<void (const HttpResponsePtr &)> &&callback) {
    // request 為表單登入
    std::string email = req->getParameter("email");
    std::string password = req->getParameter("password");
}
```

- `getPath()`

getPath() 相關

path()

getPath() 說明

取得請求路徑。適用於使用 ADD_METHOD_VIA_REGEX 或其他動態 URL 的[控制器](#)。

getPath() 輸入

無。

getPath() 回傳

請求路徑（string 型別）。

getPath() 範例

```
#include "mycontroller.h"
#include <string>

using namespace drogon;

void mycontroller::anyhandle(const HttpRequestPtr &req,
std::function<void (const HttpResponsePtr &)> &&callback) {
    // https://mysite.com/an-path/?id=5
    std::string url = req->getPath();
    // url = /an-path/
}
```

- `getBody()`

getBody() 相關

body()

getBody() 說明

取得請求 body 內容（如有）。

getBody() 輸入

無。

getBody() 回傳

請求 body（string 型別，若有）。

- `getHeader(std::string key)`

getHeader() 說明

依識別字取得請求 header。

getHeader() 輸入

header 識別字 (string 型別)。

getHeader() 回傳

header 內容 (string 型別)。

getHeader() 範例

```
#include "mycontroller.h"
#include <string>

using namespace drogon;

void mycontroller::anyhandle(const HttpRequestPtr &req,
std::function<void (const HttpResponsePtr &)> &&callback) {
    if (req->getHeader("Host") != "mysite.com") {
        // return http 403
    }
}
```

- headers()

headers() 說明

取得所有請求 header。

headers() 輸入

無。

headers() 回傳

unordered_map, 包含所有 header。

headers() 範例

```
#include "mycontroller.h"
#include <unordered_map>
#include <string>

using namespace drogon;
```

```
void mycontroller::anyhandle(const HttpRequestPtr &req,
std::function<void (const HttpResponsePtr &)> &&callback) {
    for (const std::pair<const std::string, const std::string> &header
: req->headers()) {
        auto header_key = header.first;
        auto header_value = header.second;
    }
}
```

- **getCookie()**

getCookie() 說明

依識別字取得請求 cookie。

getCookie() 輸入

無。

getCookie() 回傳

cookie 值 (string 型別)。

- **cookies()**

cookies() 說明

取得所有請求 cookie。

cookies() 輸入

無。

cookies() 回傳

unordered_map, 包含所有 cookie。

cookies() 範例

```
#include "mycontroller.h"
#include <unordered_map>
#include <string>

using namespace drogon;

void mycontroller::anyhandle(const HttpRequestPtr &req,
std::function<void (const HttpResponsePtr &)> &&callback) {
    for (const std::pair<const std::string, const std::string> &header
: req->cookies()) {
```

```
        auto cookie_key = header.first;
        auto cookie_value = header.second;
    }
}
```

- `getJsonObject()`

`getJsonObject()` 說明

將請求 body 轉為 Json 物件（通常用於 POST 請求）。

`getJsonObject()` 輸入

無。

`getJsonObject()` 回傳

Json 物件。

`getJsonObject()` 範例

```
#include "mycontroller.h"

using namespace drogon;

void mycontroller::anyhandle(const HttpRequestPtr &req,
std::function<void (const HttpResponsePtr &)> &&callback) {
    // body = {"email": "test@gmail.com"}
    auto jsonData = *req->getJsonObject();
    std::string email = jsonData["email"].asString();
}
```

實用技巧

以下非 `HttpRequest` 物件方法，但有助於處理收到的請求

檔案請求解析

```
#include "mycontroller.h"

using namespace drogon;

void mycontroller::postfile(const HttpRequestPtr &req, std::function<void
(const HttpResponsePtr &)> &&callback) {
    // 僅限 Post 請求（檔案表單）

    MultiPartParser file;
```

```
file.parse(req);

if (file.GetFiles().empty()) {
    // 未找到檔案
}

// 取得第一個檔案並儲存
const HttpFile archive = file.GetFiles()[0];
archive.saveAs("/tmp/" + archive.GetFileName());
}
```

更多檔案解析資訊請見：[檔案處理器](#)

下一步: [檔案處理器](#)