



# DotNet\_WebAPI\_教學

## 初講

---

2021年07月08日

報告人:楊昇豐

# 報告大綱



## 1 軟體環境

1. IDE : VS2019、VSCode

2. SDK : NET5

3. TestEffect : PostMan(Optional)

## 2 參考資源

1. 書目

2. 網址

3. 搜索關鍵字

## 3 概念

1. Http通訊原則

2. Restful API原則

3. (反)序列化

## 4 實作

1. MSDN官方範本

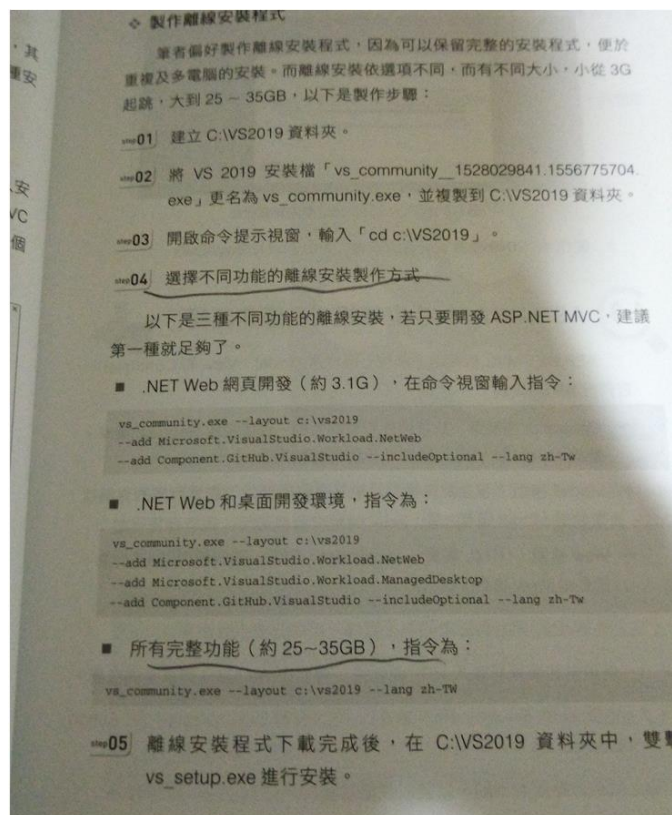
2. 福利園地WebAPI示範

- ① IDE : VS2019、VSCode
- ② SDK : NET5
- ③ TestEffect : PostMan(Optional)



## 目的：準備工具。

- ① **建議**製作**完整的**離線安裝版本；
- 強烈**建議不更改預設安裝路徑；
- 強烈**建議安裝DotNetFrameWork 4.5、4.6、4.7、4.8。



- ② <https://dotnet.microsoft.com/download/visual-studio-sdks>

### .NET/.NET Core

.NET is a free, cross-platform, open-source developer platform for building many different types of applications.

Version	Status	Visual Studio 2017 SDK	Visual Studio 2019 SDK	Runtime	Release notes
.NET 6.0	Preview	N/A	N/A	<a href="#">x64 Runtime</a>   <a href="#">x86 Runtime</a> (v6.0.0-preview.5)	<a href="#">Release notes</a>
.NET 5.0	Current	N/A	<a href="#">x64 SDK</a>   <a href="#">x86 SDK</a> (v5.0.300)	<a href="#">x64 Runtime</a>   <a href="#">x86 Runtime</a> (v5.0.7)	<a href="#">Release notes</a>

SDK：開發工具與執行環境；

Runtime：執行環境。

雙擊安裝後，能找到路徑，能下指令確認版本。

This PC > Acer (C:) > Program Files > dotnet > sdk >

Name	Date modified
5.0.201	3/23/2021 3:06 PM
5.0.202	4/24/2021 1:27 PM

### Windows PowerShell

```
PS C:\Users\UncleCarter> dotnet --version
5.0.202
PS C:\Users\UncleCarter> dotnet --list-sdks
5.0.201 [C:\Program Files\dotnet\sdk]
5.0.202 [C:\Program Files\dotnet\sdk]
PS C:\Users\UncleCarter>
```

<https://docs.microsoft.com/zh-tw/dotnet/core/install/how-to-detect-installed-versions?pivot=os-windows>

1

# 軟體環境

- ① IDE : VS2019、VSCode
- ② SDK : NET5
- ③ TestEffect : PostMan(Optional)



- ① 強烈建議安裝DotNetFrameWork 4.5、4.6、4.7、4.8。

建議安裝功能

在此區進行DotNetFrameWork安裝



VS2019選擇不到NET5(target framework)的解決方法

<https://stackoverflow.com/questions/60843091/net-5-not-available-in-visual-studio-2019>



## 參考資源

- ① 書目
- ② 網址
- ③ 搜索關鍵字



目的：指引技術精進的方向。

### ① 1. Web API 建構與設計：

API種類、定義、用途、維護；API是極其專業的事情。

<https://www.evernote.com/shard/s379/sh/20e6bcbcb-2d4d-4b03-bf4d-c71bb35382d9/5d3daf99d7c89a87319df21784cb91ac>

### 2. C# in Depth

C#程式設計進階指南。

<https://www.evernote.com/shard/s379/sh/244aa309-8a0e-43e5-b8fc-bd07dabd7bf6/d762ea6869507983a153750621072941>

### 3. Modern Cross-Platform Development

每年11月更版；微軟生態的百科全書，具有全面的產品、

技術的介紹(包含WebAPI)。 強烈建議閱讀。

<https://www.evernote.com/shard/s379/sh/7545c2b6-f7f4-44a7-a74b-e995fc169ea0/f2521a81c24ba4bb8bc2b50ab0e7ba6a>

### 4. Linux鳥哥私房菜-伺服器架設(華人世界的計概經典)

[http://linux.vbird.org/linux\\_server/](http://linux.vbird.org/linux_server/)

### 5. Visual C# 2017程式設計經典(初階)

<https://www.evernote.com/shard/s379/sh/62af3e26-4a80-4a6e-9c2d-27afee2ec885/d67e3825f34ed6971e6d944d6b566d8a>

現實：跟上與釐清新技術的關鍵就是閱讀原文材料。-5-



## 參考資源

- ① 書目
- ② 網址
- ③ 搜索關鍵字



### ② 微軟官方類別：

微軟技術官方文件(MSDN Technical documentation)

<https://docs.microsoft.com/en-us/>

微軟官方開發套件網站(Create .NET apps faster with NuGet)

<https://www.nuget.org/>

教學課程：使用 ASP.NET Core 建立 Web API | Microsoft Docs

<https://docs.microsoft.com/zh-tw/aspnet/core/tutorials/first-web-api?view=aspnetcore-3.1&tabs=visual-studio>

使用 ASP.NET Core 建立 Web API

<https://docs.microsoft.com/zh-tw/aspnet/core/web-api/?view=aspnetcore-5.0>

ASP.NET Web API 中的參數系結

<https://docs.microsoft.com/zh-tw/aspnet/web-api/overview/formats-and-model-binding/parameter-binding-in-aspnet-web-api>

ASP.NET Core 中的資料繫結

<https://docs.microsoft.com/zh-tw/aspnet/core/mvc/models/model-binding?view=aspnetcore-5.0>












# 參考資源

- ① 書目
- ② 網址
- ③ 搜索關鍵字



## ② 資料庫互動類別：





### ConnectionString\_sqlClient

-  [SqlConnection Class \(System.Data.SqlClient\) | Microsoft Docs](#)
-  [SqlConnection.ConnectionString Property \(System.Data.SqlClient\) | Microsoft Docs](#)
-  [SqlConnection Constructor \(System.Data.SqlClient\) | Microsoft Docs](#)
-  [C# Database Connection: How to connect SQL Server \(Example\)](#)
-  [Connection String Syntax - ADO.NET | Microsoft Docs](#)
-  [C# SQL Server Connection](#)
-  [連接字串產生器 | Microsoft Docs](#)
-  [\(各式各樣\)connectionString連接字串語法 | Microsoft Docs](#)
-  [\(取出是序列化字串\)SqlConnection.ConnectionString 屬性 \(System.Data.SqlClient\) | Microsoft Docs](#)
-  [\(\(取出是字串\)SqlConnection.DataSource 屬性 \(System.Data.SqlClient\) | Microsoft Docs](#)
-  [SqlConnectionStringBuilder 類別 \(System.Data.SqlClient\) | Microsoft Docs](#)
-  [\(conn.State != ConnectionState.Open\)C# Using one SqlConnection for multiple queries - Stack Overflow](#)








### DataTypeMapping\_MSSQL

-  [SQL Server Data Type Mappings - ADO.NET | Microsoft](#)
-  [SQL Server 資料類型對應 - ADO.NET | Microsoft Docs](#)



### Transaction

-  [SqlTransaction 類別 \(System.Data.SqlClient\) | Microsoft Docs](#)
-  [sql server - how to use sqltransaction in c# - Stack Overflow](#)
-  [\(command.Transaction = transaction;\)SqlConnection.BeginTransaction 方法 \(System.Data.SqlClient\) | Microsoft Docs](#)
-  [\(using transaction\)c# - Why use a using statement with a SqlTransaction? - Stack Overflow](#)



### DataSetAndDataTable

-  [SqlDataAdapter 類別 \(System.Data.SqlClient\) | Microsoft Docs](#)
-  [\(using sqlDataAdapter\\_\\_using datareader\)c# - Populate data table from data reader - Stack Overflow](#)
-  [.net - Read SQL Table into C# DataTable - Stack Overflow](#)
-  [Populating a DataSet from a DataAdapter - ADO.NET | Microsoft Docs](#)
-  [\(CURD\)DataAdapter Parameters - ADO.NET | Microsoft Docs](#)
-  [\(BatchInsertUpdate\)Updating Data Sources with DataAdapters - ADO.NET | Microsoft Docs](#)
-  [Performing Batch Operations Using DataAdapters - ADO.NET | Microsoft Docs](#)

### SqlexceptionHandling

-  [.net - How to catch SqlException caused by deadlock? - Stack Overflow](#)
-  [\(sqlexception\\_ErrorNumber\)System Error Messages | Microsoft Docs](#)

### AdapterInsert

-  [\(adapter.InsertCommand.ExecuteNonQuery\(\);\)How to DataAdapter Insert Command - Sql Server](#)
-  [SqlCommandBuilder Class \(System.Data.SqlClient\) | Microsoft Docs](#)



## 參考資源

① 書目

② 網址

③ 搜索關鍵字



### ② 網路知識暨傳輸格式類別：


W 網際網路媒體型式 - 維基百科，自由的百科全書


 MIME types (IANA media types) - HTTP | MDN

W 傳輸層安全性協定 - 維基百科，自由的百科全書

 什麼是 SSL、TLS 以及 HTTPS ? | DigiCert

 淺談 HTTP Method：表單中的 GET 與 POST 有什麼差別？ - Soul & Shell Blog

 跨來源資源共用 ( CORS ) - HTTP | MDN

 HTTP 狀態碼 - HTTP | MDN

 (初階)ASP.NET Core 教學 - Web API JSON 序列化設定 | John Wu's Blog





## 參考資源

① 書目

② 網址

③ 搜索關鍵字



必須用對的方式問對的問題。

③ Google關鍵字：

MSDN、DotnetCore5、Dotnet WebApi、C#、MDN、Nuget、Sql Server  
w3c、sql client、sqlite3、Microsoft.data、MSSQL、Csharp、  
Visual Studio、target framework、Controller、Action Result、Swagger API、  
http protocol、鳥哥、request header body、http post get

不好的搜索

如何建立API

什麼是序列化

資料庫連線

好的搜索

How to Create **Dotnet** WebApi

**C#** json serialize deserialize

**C# sql server** connection string

# 3 概念

① Http通訊原則

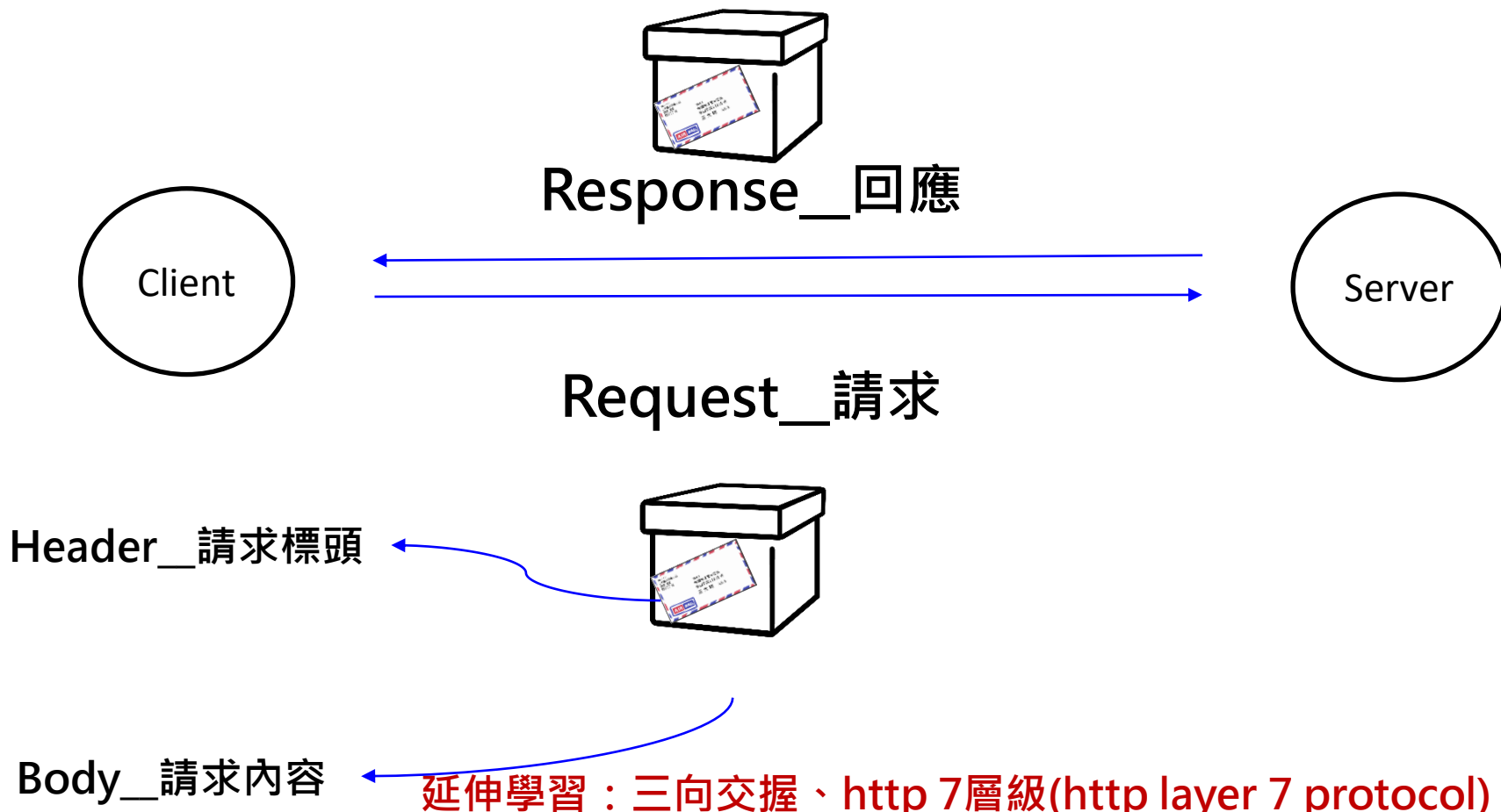
② Restful API原則

③ 序列化、反序列化



目的：掌握必要概念。

① 主從式架構：發送請求者稱為Client(客戶端)；回應者稱為Server(伺服器端)。



# 3 概念

- ① Http通訊原則
- ② Restful API原則
- ③ 序列化、反序列化



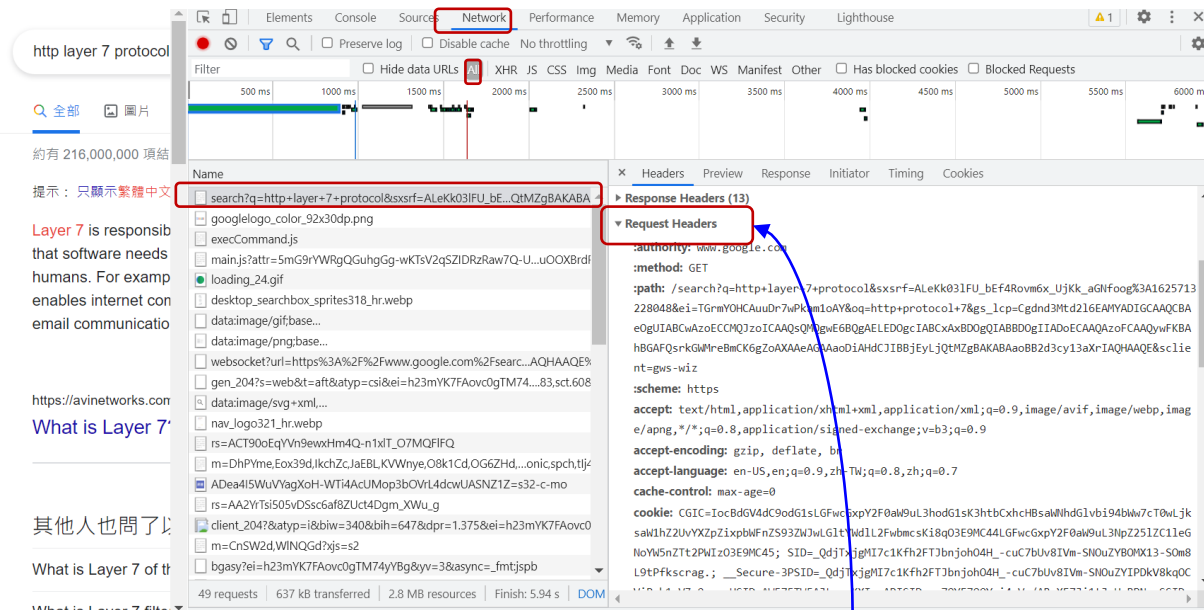
## F12觀察Http通訊資訊。

Header\_\_請求標頭



Body\_\_請求內容

Chrome按下F12進入開發者環境



Request\_\_請求

# 3 概念

① Http通訊原則

② Restful API原則

③ 序列化、反序列化



目的：了解Restful就是API溝通原則。

② 將資源取得的方式註記在API方法上。



1. Get：輸入網址就可取得資源。
2. Post：需要給Body，才可取得資源。
3. Put：資料更新。
4. Delete：刪除資料。
5. Patch(少用)

解讀

1. **Get**SaleList()  
打網址就可
2. **Post**PersonalState()  
需要送資料才可
3. **Put**LoginTime()  
更新
4. **Delete**Person()  
刪除

# 3 概念

① Http通訊原則

② Restful API原則

③ 序列化、反序列化



目的：資料本身與格式因應場景得變化。

③ 心法：序列化成一字串；反序列化成一物件。

```
{  
  "Domain" : "Knowhow"  
}
```

“{ ‘Domain’ : ‘Knowhow’ }”

“{ \ ‘Domain\’ : \ ‘Knowhow\’ }”

\*物件

<----load-----

-----dump----

\*字串

<----loads-----

-----dumps----

Json字串

\*dump = Serialize、load = Deserialize。

\*字串= 符合Json結構的字串。

\*物件= 符合Json結構的物件，例如Dict、Dictionary。

延伸閱讀：System.Text.Json可以完美取代Newtonsoft.Json ?-13-

# 3 概念

① Http通訊原則

② Restful API原則

③ 序列化、反序列化



到底什麼是API？

以程式設計角度，它是介面；

以電玩裝置角度，它是搖桿；

以非程式角度：它是提供服務的函示；取得資源的方法。

到底什麼是Web API？

遵從Http通訊原則的API，就稱做Web API。

以Visual Studio2019示範。

教學課程：使用 ASP.NET Core 建立 Web API | Microsoft Docs  
<https://docs.microsoft.com/zh-tw/aspnet/core/tutorials/first-web-api?view=aspnetcore-3.1&tabs=visual-studio>

使用 ASP.NET Core 建立 Web API  
<https://docs.microsoft.com/zh-tw/aspnet/core/web-api/?view=aspnetcore-5.0>

延伸學習：

1. 什麼是CORS？怎麼在Dotnet WebAPI調整CORS防護？
2. 什麼是依賴注入(Dependency Injection)？  
怎麼在Dotnet WebAPI使用？  
使用DI的目的為何？
3. 如何將Dotnet WebAPI部署到IIS？
4. 什麼是Kestrel？怎麼調整部署設定？

# IN THE END,

IT DOES EVEN MATTER!



每年10~11月，台灣微軟舉辦全球DotNet開發者盛會，  
是了解.NET技術更迭不可或缺的機會。

<https://dotnetconf2020.study4.tw/>

去年前往的心得

<https://www.evernote.com/shard/s379/sh/e52836db-45aa-45e9-9589-6329cba33f5d/f927ff0ea02a5d367e08cd576b6be6e1>