

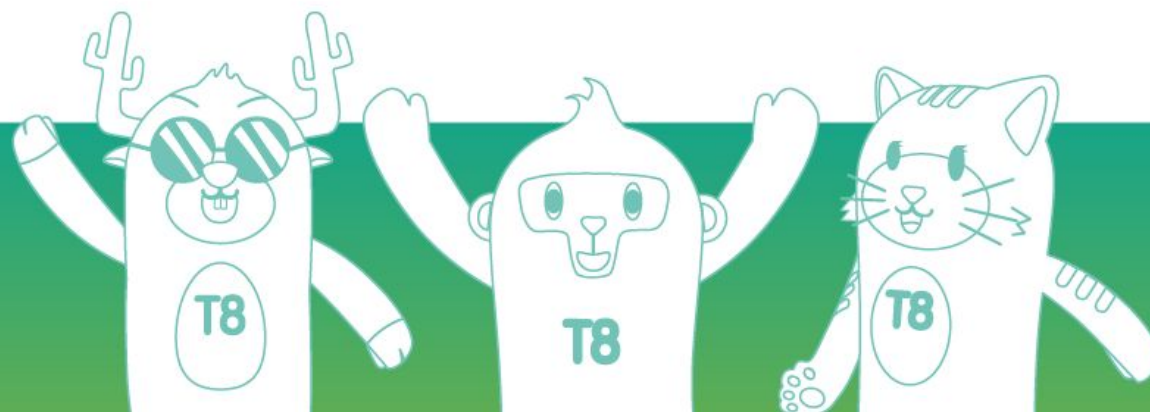
# AJAX

■ 授課講師	曾景鴻
■ 教材編寫	曾景鴻

緯育 *TibaMe*

即學・即戰・即就業

<https://www.tibame.com/>



## 課程大綱

**Chapter 1. AJAX介紹**

**Chapter 2. JavaScript發送請求**

**Chapter 3. jQuery發送請求**

**Chapter 4. Axios發送請求**

# Chapter 1. AJAX介紹

**1-1: 前言**

**1-2: 什麼是AJAX**

**1-3: AJAX特色**

**1-4: 同步&非同步**

**1-5: 發展**

**1-6: 運作方式**

## 前言

- 傳統網頁透過 HTML 表單 (HTML form) 發出請求 (request), 會伴隨著轉址(Redirect)行為, 每傳送一次表單就要重刷一次頁面。
- 使用AJAX技術的網頁, 可會充份利用AJAX的特性, 以非同步的方式傳接資料, 再利用 JavaScript或jQuery等技術, 動態改變網頁內容。
- 透過AJAX傳送資料, 網頁會保持在固定畫面, 瀏覽器不會重新載入頁面, 藉以改善網站使用的體驗。

### 什麼是AJAX？

- AJAX 代表 **Asynchronous JavaScript And XML**，即非同步的 JavaScript 及 XML。
- AJAX並非新的程式語言，而是一種Client和Server的**溝通標準**。
- AJAX不須安裝額外套件在客戶端或伺服器端，在瀏覽器上就能執行。

## 特色

- AJAX 使用 JavaScript的XMLHttpRequest 物件與伺服器進行通訊。它可以傳送並接收多種格式的資訊，包括 JSON、XML以及文字檔案。
- AJAX 最吸引人的特點是「**非同步**」的特質，這代表它可以與伺服器溝通、交換資料、以及更新頁面，且無須重整網頁。

## 同步&非同步

- 非同步或稱異步，是指網頁發出的請求 (HTTP Request)在背景執行，同時有一個監聽者(Listener function)等待伺服器的回應。
- 只有透過非同步請求(Request)，才能避免傳統HTML Form發出請求，若等不到回應而造成網頁卡住，影響使用者體驗。

### 發展

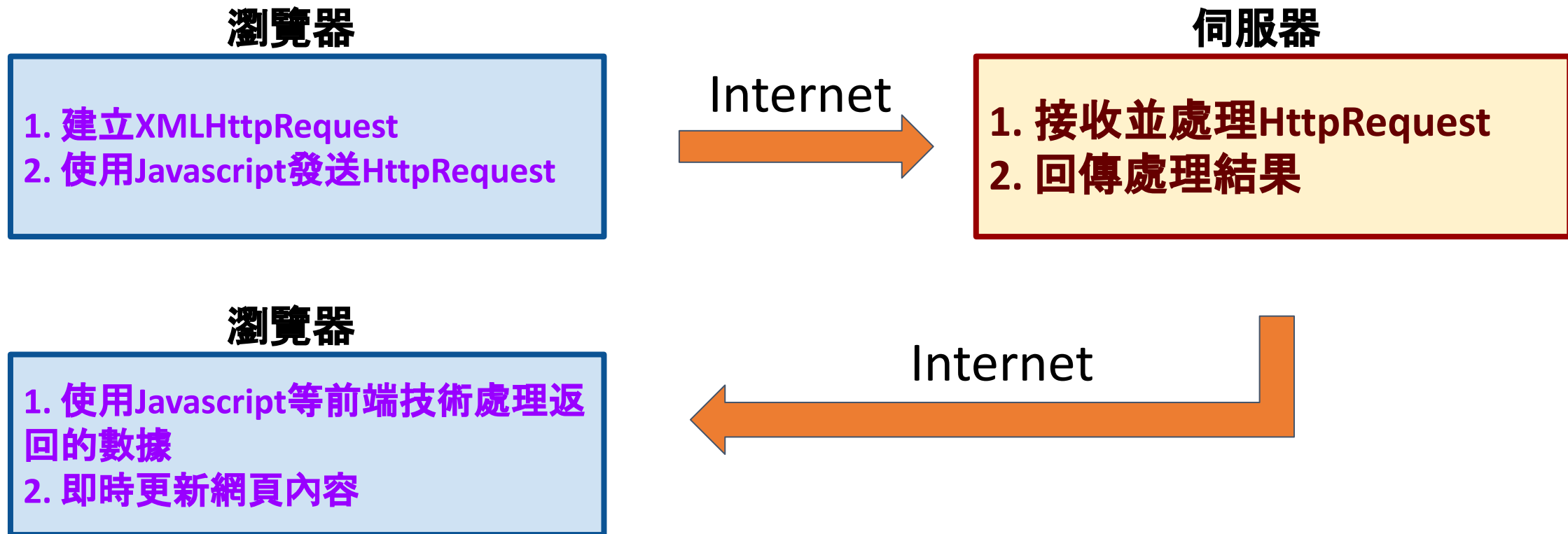
- 自 2005 年起, Google 透過 Google Suggest 使 AJAX 變得流行起來。
- Google Suggest 使用 AJAX 創造出動態性極強的 web 介面, 當我們在Google搜尋框輸入關鍵字時, JavaScript 會立即把這些字元發送到後端程式, 接著會得到一個搜尋建議的文字串列。



## 發展



## 運作方式



- 下載範例程式，解壓縮後將AJAX資料夾放入htdocs內

<https://reurl.cc/v7ELZI>



雲端硬碟

資料夾



AJAX



FileUpload



PDO

# Chapter 2. JavaScript發送請求

2-1: 簡介

2-2: 發送請求步驟

2-3: 後端回傳

## 簡介

- 使用 JavaScript 向伺服器發送請求(HTTP Request), 便需要一個能夠提供相關功能的類別。這樣的類別最初由 Internet Explorer 以 ActiveX 物件的方式引入, 稱為 XMLHTTP。
- Chrome、Firefox、Safari 及其他瀏覽器則隨後跟進, 實作了 XMLHttpRequest 類別, 以提供微軟最初的 ActiveX 物件中的方法及屬性。

### JavaScript發出請求(HTTP Request)步驟

- 初始化XMLHttpRequest
- 設定處理回傳的function
- 發出請求

## 1. 初始化XMLHttpRequest

```
if (window.XMLHttpRequest){  
    // IE7+, Firefox, Chrome, Opera, Safari 瀏覽器執行程式碼  
    xmlhttp = new XMLHttpRequest();  
}else{  
    // IE6, IE5 瀏覽器執行程式碼  
    xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");  
}
```

## 2. 設定處理回傳的function

```
xmlhttp.onreadystatechange = function(){  
    //判斷XMLHttpRequest是否發送成功  
    if (xmlhttp.readyState == 4 && xmlhttp.status == 200){  
        //更新html內容  
        document.getElementById("result").innerHTML = xmlhttp.responseText;  
    }  
}
```

`xmlhttp.readyState`

- 1: 讀取中
- 2: 已讀取
- 3: 資訊交換中
- 4: 一切完成

`xmlhttp.status`

- 0: 未開始
- 200: OK
- 404: 未找到頁面
- 500: Server error(後端程式錯誤)



### 3. 發出請求

#### GET:

```
xmlhttp.open("GET", "API.php?account=" + str);  
xmlhttp.send();
```

#### POST:

```
xmlhttp.open("POST", "API.php");  
//若使用POST需額外設定RequestHeader(GET不用)  
xmlhttp.setRequestHeader('Content-Type',  
'application/x-www-form-urlencoded');  
//雖透過POST傳值, 仍須使用字串方式傳遞  
xmlhttp.send("account=" + str);
```

假設\$data是一個從資料庫取得的二維陣列，後端PHP可透過echo將回傳值打印在網頁上讓前端JS接到

```
if(count($data) > 0){  
    //將二維陣列取出顯示其值  
    foreach($data as $index => $row){  
        echo $row["Account"]; //欄位名稱  
        echo "<br>";  
    }  
}else{  
    echo "查無結果!";  
}
```

請另外複製出PHP-PDO章節裡的Login.html & Login.php, 將原本頁面會跳轉(HTML->PHP)的方式利用javascript實作AJAX改寫。

帳號:

密碼:

結果:

登入成功

# Chapter 3. jQuery發送請求

**3-1: 簡介**

**3-2: 發送請求步驟**

## 簡介

- jQuery是一套跨瀏覽器的JavaScript函式庫，簡化HTML與JavaScript之間的操作。
- jQuery的語法設計使得許多操作變得容易，如處理事件、以及開發AJAX程式。
- jQuery在操作事件處理以及AJAX部分，都可以有比寫原生JavaScript還要更快速的處理方式。因為jQuery本身將那些功能都已經封裝好供使用。

### jQuery發出請求(HTTP Request)步驟

- 載入jQuery套件(.js)
- 設定處理發送請求/回傳(成功或失敗)的function

## 1. 載入jQuery套件(.js)

```
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jque  
ry.min.js">  
</script>
```

## 2. 設定處理發送請求/回傳(成功或失敗)的function

```
$.ajax({  
    method: "POST",  
    url: "API.php",  
    data: {  
        account: str  
    },  
    dataType: "text",  
    success: function (response) {  
        //更新html內容  
        $("#result").html(response);  
    },  
    error: function(exception) {  
        alert("發生錯誤: " + exception.status);  
    }  
});
```



假設\$data是一個從資料庫取得的二維陣列，後端PHP可透過echo將回傳值打印在網頁上讓前端jQuery接到

```
if(count($data) > 0){  
    //將二維陣列取出顯示其值  
    foreach($data as $index => $row){  
        echo $row["Account"]; //欄位名稱  
    }  
}else{  
    echo "查無結果!";  
}
```

## 設定處理發送請求/回傳(成功或失敗)的function

```
$.ajax({  
    method: "POST",  
    url: "API.php",  
    data: {  
        account: str  
    },  
    dataType: "json",  
    success: function (response) {  
        //更新html內容前先清空原有資料  
        $("#result").html("");  
        //更新html內容(透過jQuery跑迴圈取值)  
        $.each(response, function(index, row) {  
            $("#result").append(row.Account + "/" + row.PWD + "/" + row.CreateDate +  
"  
<br/>");  
        });  
    },  
    error: function(exception) {  
        alert("發生錯誤: " + exception.status);  
    }  
});
```

假設\$data是一個從資料庫取得的二維陣列，後端PHP可透過 **json\_encode()** 函式將回傳值echo在網頁上讓前端jQuery接到

```
echo json_encode($data);
```

請另外複製出PHP-PDO章節裡的Login.html & Login.php, 將原本頁面會跳轉(HTML->PHP)的方式利用jQuery實作AJAX改寫。

帳號:

密碼:

結果:

**登入成功**



# Chapter 4. Axios發送請求

4-1: 簡介

4-2: 發送請求步驟

4-3: API文件範例

## 簡介

- Axios 是一個基於 Javascript 的 Ajax 函式庫，可以輕易的做到非同步發送get、post請求。
- 因為Vue、React等框架的出現，促使了Axios輕量級庫的發展。

### Axios發出請求(HTTP Request)步驟

- 載入Axios套件(.js)
- 設定處理發送請求/回傳(成功或失敗)的function

## 1. 載入Axios套件(.js)

```
<script  
src="https://unpkg.com/axios/dist/axios.min.js">  
</script>
```



## 設定處理發送請求/回傳(成功或失敗)的function

```
//將要傳送的内容封裝成FormData
var fd = new FormData();
fd.append('account', str);
//POST請求
axios.post('API.php', fd)
.then(function (response) {
    //更新html内容(跑迴圈解析json)
    var msg = '';
    response.data.forEach((row, index) => {
        msg += row.Account + "/" + row.PWD + "/" + row.CreateDate + "<br/>";
    })
    //更新html内容
    document.getElementById("result").innerHTML = msg;
})
.catch(function (error) {
    console.log(error);
});
```

假設\$data是一個從資料庫取得的二維陣列，後端PHP可透過 **json\_encode()** 函式將回傳值echo在網頁上讓前端Axios接到

```
echo json_encode($data);
```

## API.php - Method: POST

欄位	說明	型態
ID	流水號(PK)	Int
Account	使用者帳號	String
CreateDate	建立時間	DateTime

## 4-3: API文件範例

```
[
{
  "0": "1",
  "1": "測試一",
  "2": "2021-01-28 22:26:45",
  "ID": "1",
  "Account": "測試一",
  "CreateDate": "2021-01-28 22:26:45"
},
{
  "0": "2",
  "1": "test2",
  "2": "2021-01-28 22:26:51",
  "ID": "2",
  "Account": "test2",
  "CreateDate": "2021-01-28 22:26:51"
}
]
```

請另外複製出PHP-PDO章節裡的Login.html & Login.php, 將原本頁面會跳轉(HTML->PHP)的方式利用Axios實作AJAX改寫。

帳號:

密碼:

結果:

**登入成功**



請另外複製出PHP-PDO章節裡Join.html & Join.php & Select.php, 將原本頁面會跳轉(HTML->PHP)的方式換成實作AJAX改寫。

- 提示:

1. 在AJAX資料夾內新增Join.html, Join.php, Select.php
2. Join.html: 有兩個欄位[帳號][密碼], 一個submit按鈕和兩個JS function  
Function 1: 按下submit按鈕後取得[帳號][密碼]欄位值, 透過AJAX傳送到Join.php, Join.php回傳成功, 接著呼叫Function 2。  
Function 2: 透過AJAX取得Select.php回傳的json並顯示結果
3. Join.php: 取得[帳號][密碼]值, 寫入table "member"並回傳成功
4. Select.php: 取得table "member"的所有資料並封裝成json格式回傳

請於8/31前繳交上傳Join.html, Join.php, Select.php(以zip壓縮)到Tibame個人學習專區