



巨匠線上真人

# Java Web OCE JWCD元件系統 開發認證

[www.pcschoolonline.com.tw](http://www.pcschoolonline.com.tw)



巨匠線上真人

Java Web OCE JWCD元件系統開發認證

# 第二堂 Web Application 與通訊協定概要

# 本堂教學重點

1. 網站系統如何運行
2. 網站系統如何傳送方式與傳遞資料
3. 檢視Java Web API支援HTTP協定架構

# 網站系統如何運行

- ◆ 網站系統一般區分為前端(Client Side)與後端(Server Side)
- ◆ 前端(Client Side)可以是:
  - ◆ 瀏覽器(Browser)
  - ◆ 或者是任何支援HTTP協定的應用程式等。
- ◆ 後端(Server Side)可以是:
  - ◆ 網站伺服器(Web Server)
  - ◆ 網站伺服器上網站應用系統(Web Application)，如電商網站或者企業ERP等。
  - ◆ E-MAIL Service
  - ◆ 網際網路服務(XML Web Service/RESTful)...

# 靜態與動態

◇網站系統的網頁可以區分為靜態與動態

◇靜態頁面

- 例如：html、txt 或 css 等 ...一般為前端瀏覽器執行功能。

◇動態頁面

- 例如：Servlet、JSP、PHP、ASP.NET Page 等，需要轉換成程式碼，透過後端引擎執行。主要目的用來存取後端的資源，如資料庫或者文件等動態存取。

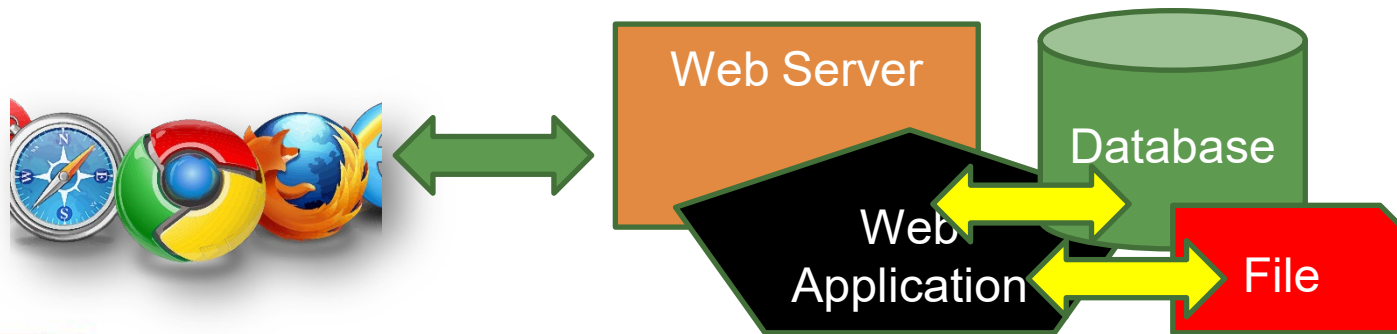
# Java網站系統架構

## ◆ 網站應用系統伺服器

- ◆ 所有的網站應用系統，都依附在相對的網站應用系統伺服器中，如GlassFish Server/TomEE Server或者Tomcat Container，主要是來執行Java網站系統。
- ◆ 並統一由伺服器進行管理/效能管理/安全管理等統籌系統資源的分派。

## ◆ Java 網站系統如何被執行？

- ◆ 使用Web Container，又稱呼為JSP Engine或者Servlet Engine。
- ◆ 使用Web Server，如Glass Fish Server/TomEE Server。



# JEE 整體架構

## ◆ 前端

- ◆ 一般為瀏覽器或者支援HTTP功能的程式等。

## ◆ 展示層

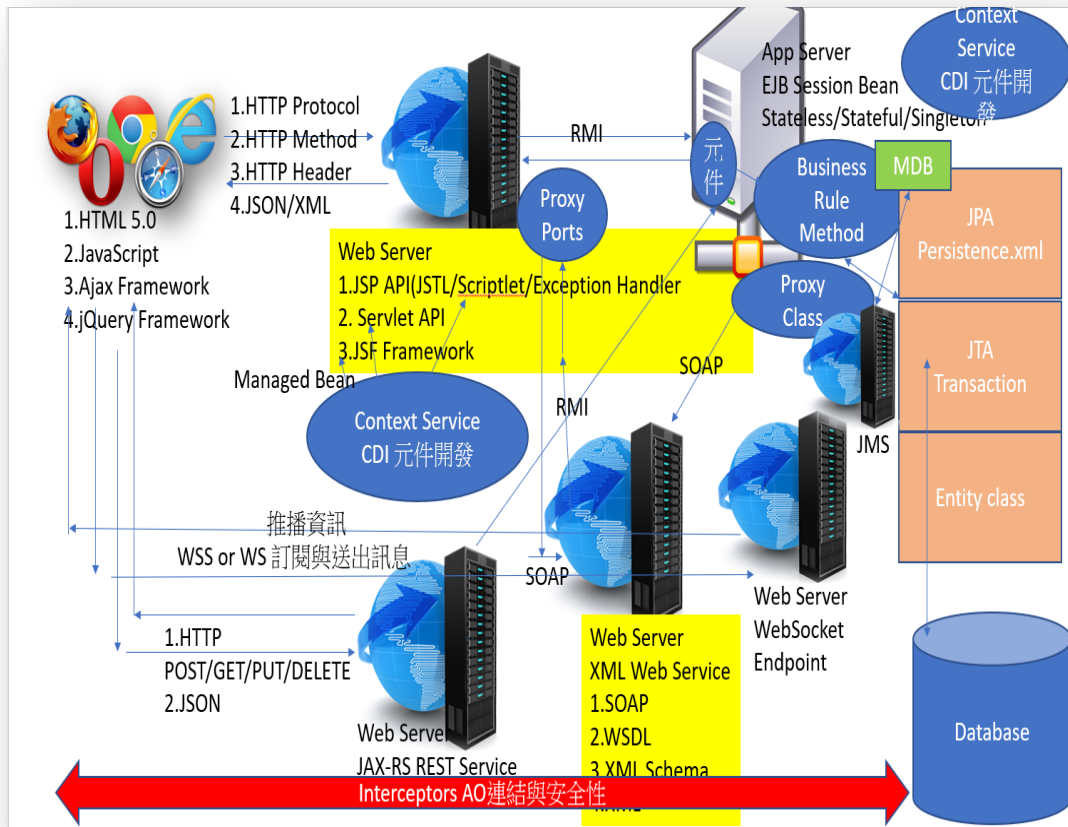
- ◆ 配合前端互通資訊的控制流程Servlet。
- ◆ 與進行前端使用者View(畫面)設計。如HTML或者JSP。

## ◆ 商業層

- ◆ 抽離商業邏輯，可設計為共同使用的處理程序或者Entity等類別規劃。進行分散式元件架構設計，如EJB架構/JPA。

## ◆ 資源

- ◆ 如資料庫或者資料檔案等



# JEE網站應用系統與Web Service架構

## ◆ 前端

- ◆ 瀏覽器與支援HTTP功能的程式等。
- ◆ 可以在MVC架構下的網頁撰寫JavaScript呼喚RESTful Service

## ◆ JAX-RS Web Service層

- ◆ 提供HTTP協定方式進行端點呼喚
- ◆ 提供XML /JSON資料傳輸

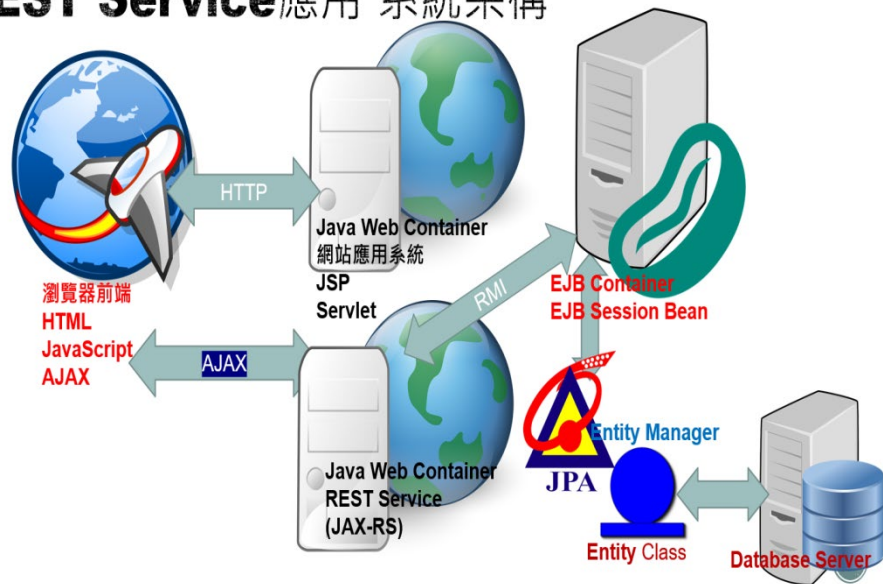
## ◆ 商業層(EJB)

- ◆ 抽離可共同使用的處理程序或者Entity等類別規劃。進行分散式元件架構設計。可以是EJB或者雲端API。

## ◆ 資源

- ◆ 如資料庫或者資料檔案等

## Java EE 分散式系統模組 -REST Service應用 系統架構





# 網站系統如何傳送方式與傳遞資料

- ◆ 架構在HTTP通訊協定進行資料傳輸
- ◆ HTTP
  - ◇ Hypertext Transfer Protocol，是一種簡單的要求與回應(Request-Response)服務，且沒有持續連線狀態(stateless)下的協定。
  - ◇ Client 端與 Server 端在分析了 HTTP 訊息後會給予相對服務。

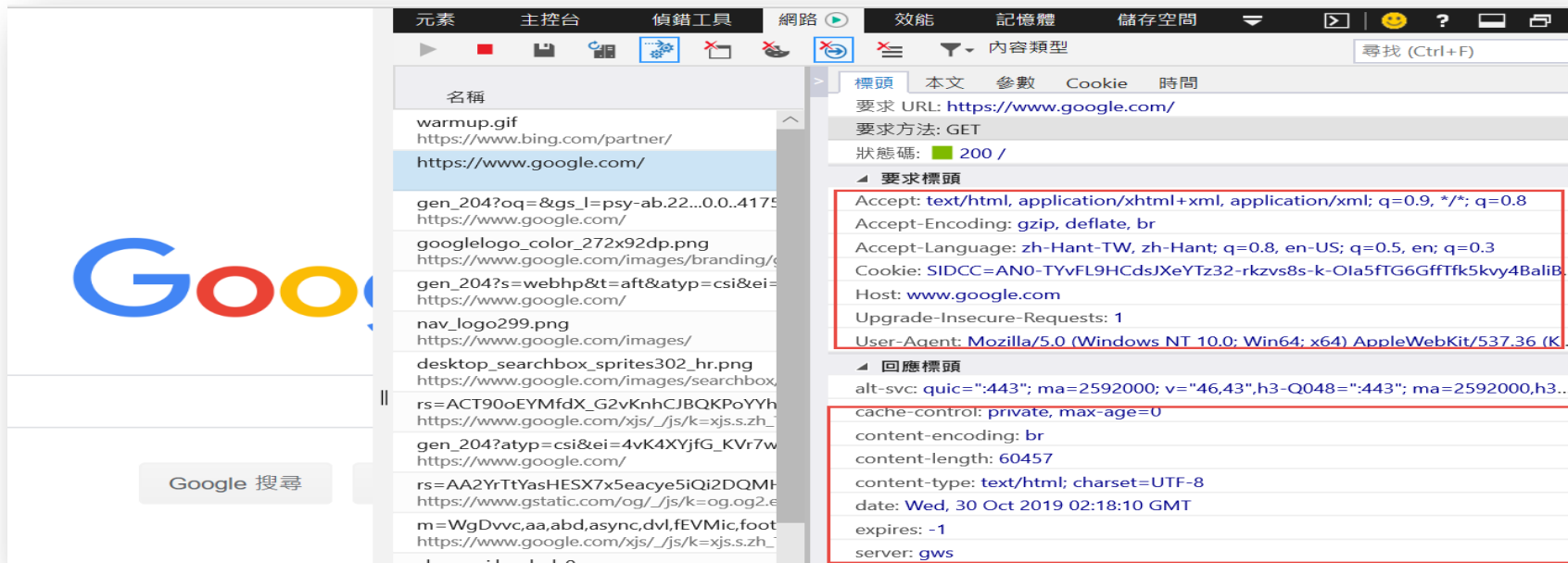


# HTTP 協定 URL

## ◆ URL

- [HTTP://www.pcschool.com/Java/Index.html](http://www.pcschool.com/Java/Index.html)
- [FTP://www.pcschool.com/File/jee.pdf](ftp://www.pcschool.com/File/jee.pdf)
- 指定相對的的通訊協定，來取得特定路徑的資源。

# Http Request/Response Model



The screenshot shows the Microsoft Edge browser with the Google homepage loaded. The developer tools network tab is open, displaying the request and response headers for the main Google page. The response headers are highlighted with a red box.

**Request Headers:**

- Accept: text/html, application/xhtml+xml, application/xml; q=0.9, \*/\*; q=0.8
- Accept-Encoding: gzip, deflate, br
- Accept-Language: zh-Hant-TW, zh-Hant; q=0.8, en-US; q=0.5, en; q=0.3
- Cookie: SIDCC=AN0-TYvFL9HCdsJXeYTz32-rkzvs8s-k-Ola5ftG6GffTfk5kvy4BaliB...
- Host: www.google.com
- Upgrade-Insecure-Requests: 1
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (K...

**Response Headers:**

- cache-control: private, max-age=0
- content-encoding: br
- content-length: 60457
- content-type: text/html; charset=UTF-8
- date: Wed, 30 Oct 2019 02:18:10 GMT
- expires: -1
- server: gws

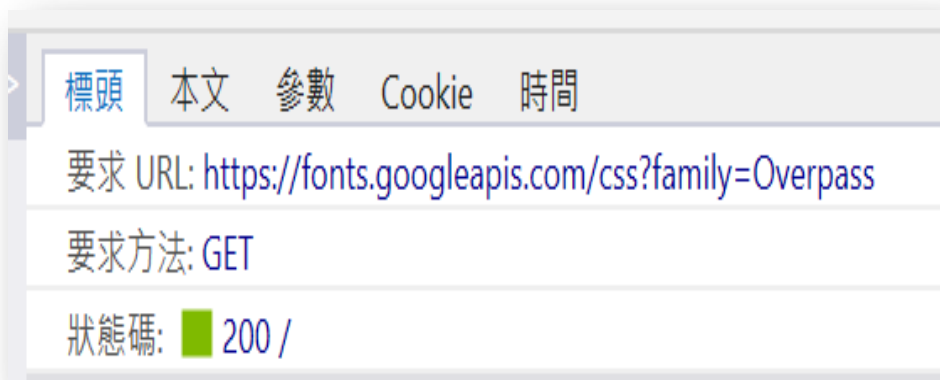
使用Edge瀏覽  
Google首頁  
透過檢查HTTP Header

# HTTP 組成部分

訊息部分	說明
初始定義列	初始定義列中，包含有HTTP 的方法(Http Request Method) 以及網址與HTTP版本(如HTTP 1.1)。
Headers部分	採用 HTTP 的方法，所包含的標頭資訊，如User-Agent、Content-Type 或 ContentLength與其他自訂的Header 等，例如安全性設定的api-key等...
空白行	空白行主要用來結束Headers，並且設定Body的分隔。這一行不能缺。
Body訊息本體	區分 Request 或 Response 的來回內容。例如使用 Request所傳遞的參數內容(一般配合表單欄位)。或者是 Response 中的 HTML Tag或raw data(JSON, XML)以及採用Binary傳遞的資料內容，如圖片等...

# HTTP 初始列

- ◆ 設定HTTP Request Method
  - ◆ 如採用GET 或者是POST等
- ◆ URI
- ◆ HTTP版本



# HTTP 協定 Request Method

## ◆ HTTP 七個方法：

### ◆ GET

- 直接取得 server 上的資源 resource，也支援傳送少量資料的功能，是用在查詢作業上。

### ◆ POST

- 傳送資料集合 ( 含大量資料 ) 到 server 端，以便進行後續資料的處理，如同新增記錄內容一般。

### ◆ HEAD

- 僅欲擷取 response 的 header 內容時 ( meta-information )。

### ◆ PUT

- 欲將指定文件放到 Server 端上時，一般配合REST是用在修改資料上。

### ◆ DELETE

- 欲刪除存放在 Server 上的文件時。

### ◆ TRACE

- 可用來除錯以追蹤錯誤來源。

### ◆ OPTIONS

- 查詢 Server 上的特定資源或可支援的方法。

# Client如何透過URL傳遞訊息

- ◆ URL傳遞訊息方式可以採用QueryString查詢字串架構。
- ◆ ?參數名稱=參數值&參數名稱=參數值...
- ◆ 採用QueryString傳遞資料須考量安全性。
- ◆ 一般採用QueryString所配合的船方是為GET



<https://shopee.tw/m/electronicthree?smtt=210.46961>

# HTTP Get v.s Post 傳送方法

## ◆ Get

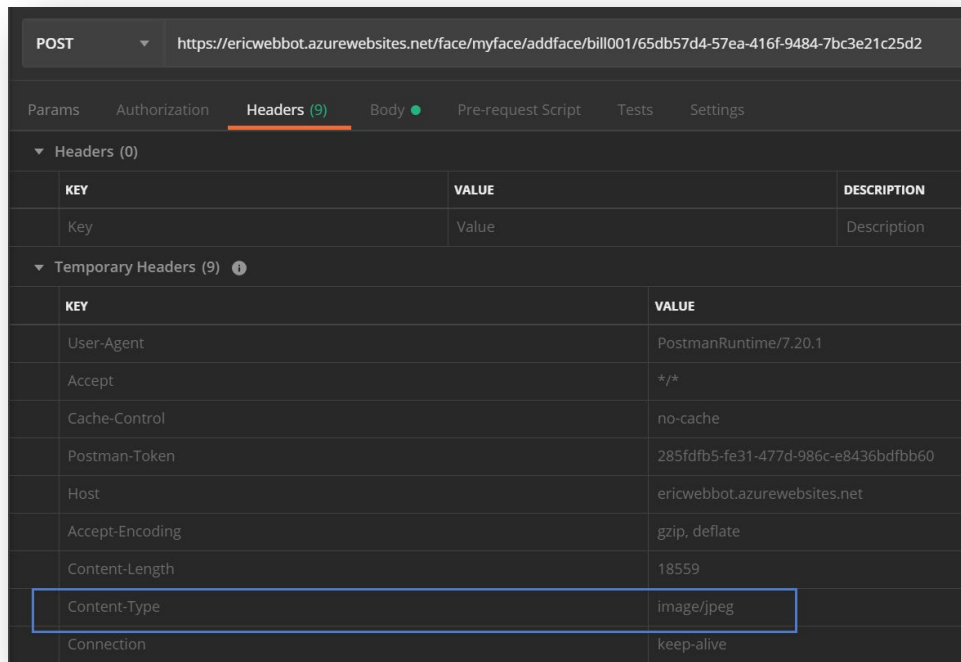
◇ /demo/hello.jsp?paramName=Java

- 數據限制 8K

## ◆ Post

◇ /demo/hello.jsp

- 透過表單欄位可以傳遞多內容
- 透過Raw Data採用串流可以上傳非文字檔內容，如圖片的binary內容。
- 數據限制 2G



POST		https://ericwebbot.azurewebsites.net/face/myface/addface/bill001/65db57d4-57ea-416f-9484-7bc3e21c25d2
Params Authorization Headers (9) Body Pre-request Script Tests Settings		
▼ Headers (0)		
KEY	VALUE	DESCRIPTION
Key	Value	Description
▼ Temporary Headers (9)		
KEY	VALUE	
User-Agent	PostmanRuntime/7.20.1	
Accept	*/*	
Cache-Control	no-cache	
Postman-Token	285fdb5-fe31-477d-986c-e8436bdfbb60	
Host	ericwebbot.azurewebsites.net	
Accept-Encoding	gzip, deflate	
Content-Length	18559	
Content-Type	image/jpeg	
Connection	keep-alive	

Postman採用binary進行圖片上傳



# HTTP Request Method 展示

- https://luairestservice.azurewebsites.net/api/orderscustomers/webOrdersCustomersQry?ulString=找客戶ANTON跟ALFKI 訂單10270 10280

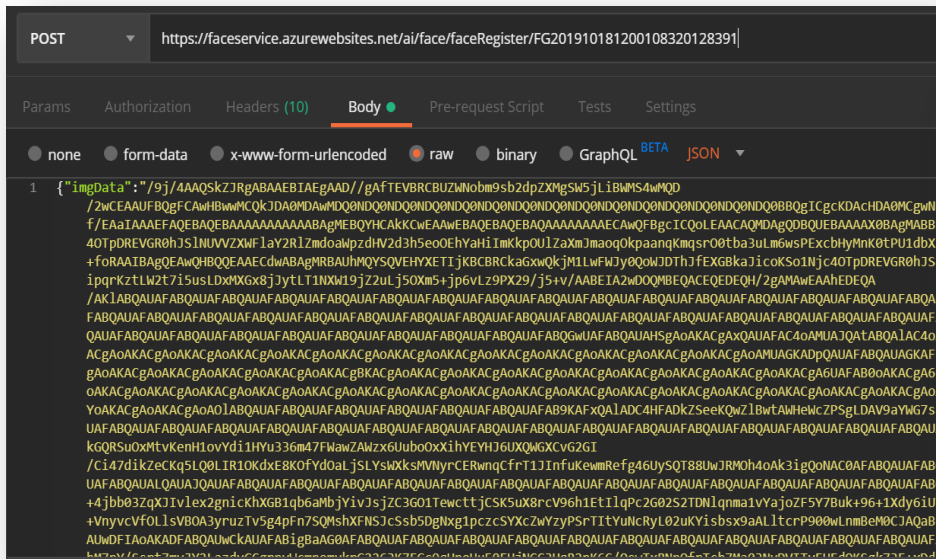
GET	https://luairestservice.azurewebsites.net/api/orderscustomers/webOrdersCustomersQry?ulString=找客戶ANTON跟ALFKI 訂單10270 10280		
Params	Authorization	Headers (7)	Body
▼ Headers (0)			
	KEY	VALUE	DESCRIPTION
	Key	Value	Description
▼ Temporary Headers (7) ⓘ			
	KEY	VALUE	
	User-Agent	PostmanRuntime/7.19.0	
	Accept	*/*	
	Cache-Control	no-cache	
	Postman-Token	c6f7e582-5970-4b89-8352-90c17c8bfd88	
	Host	luairestservice.azurewebsites.net	
	Accept-Encoding	gzip, deflate	
	Connection	keep-alive	

LUIS.AI  
自然語言服務

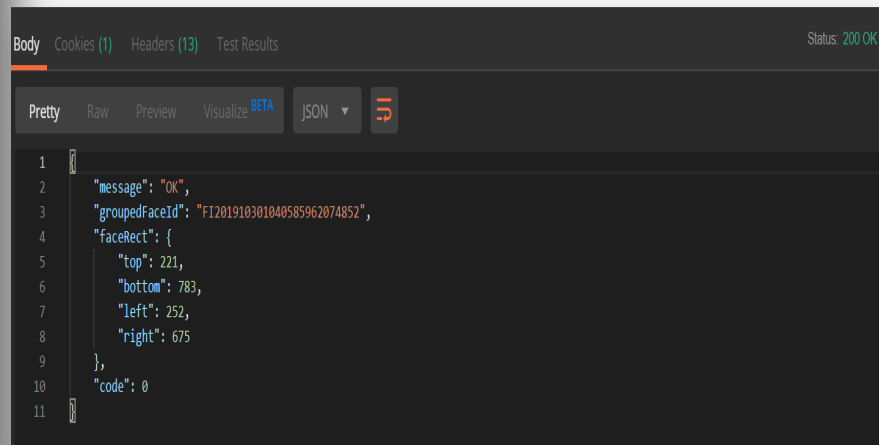
```
"statusCode": 200,  
"message": "您的原意是: 找客戶ANTON跟ALFKI 訂單10270 10280<br/>您要求服務是:客戶與訂單查詢<br/>我們給您的答覆: 已經找到您要相關資料!! 感謝您的查詢!!<br/>客戶詳細資料:<br/>客戶編號: ANTON<br/>公司行號: Antonio Moreno Taqueria<br/>聯絡地址: Mataderos 2312<br/>聯絡電話: (5) 555-3932<br/>-----<br/>客戶詳細資料:<br/>客戶編號: ALFKI<br/>公司行號: Alfreds Futterkiste<br/>聯絡地址: Obere Str. 57<br/>聯絡電話: 030-0074321<br/>-----<br/>訂單資訊:<br/>訂單編號: 10270<br/>訂單日期: 8/1/1996 12:00:00 AM<br/>出貨日期: 8/2/1996 12:00:00 AM<br/>-----<br/>訂單編號: 10280<br/>訂單日期: 8/14/1996 12:00:00 AM<br/>出貨日期: 9/12/1996 12:00:00 AM<br/>-----<br/>"
```

# HTTP Request Method-POST Demo

- ◆ 採用POST傳送base64字串(相片)進行人臉辨識
- ◆ <https://faceservice.azurewebsites.net/ai/face/faceRegister/FG201910181200108320128391>

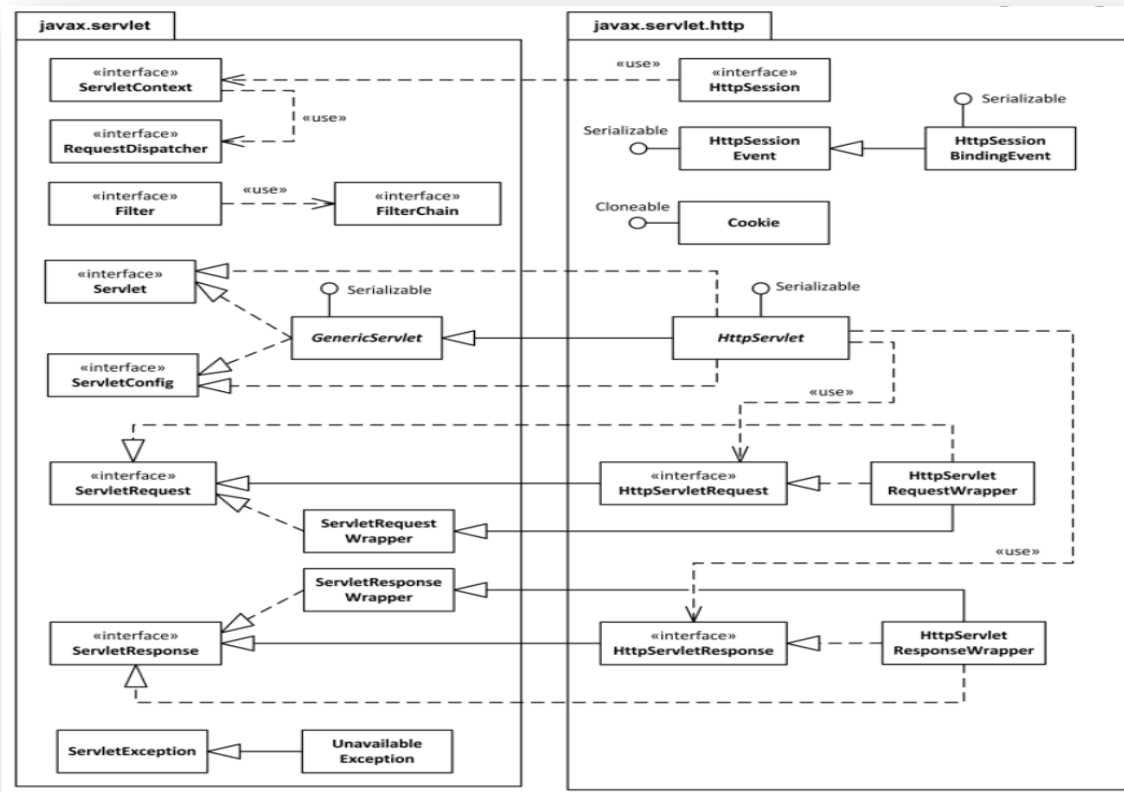


## Response JSON Data



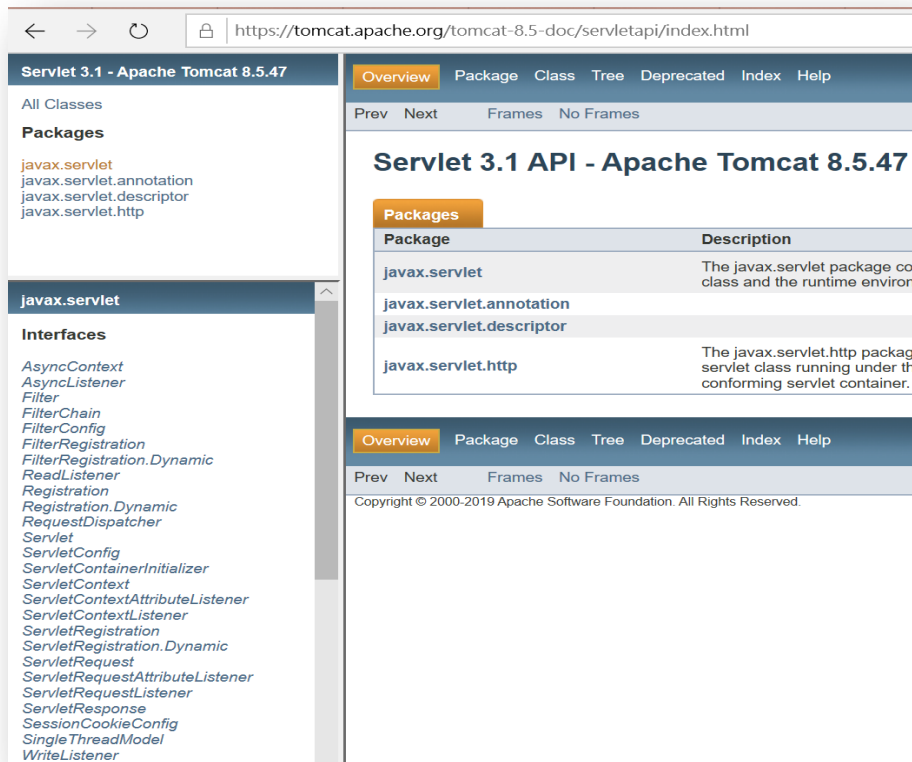
# 檢視Java Web API支援HTTP協定架構

- ◆ JEE API
- ◆ javax.servlet API
- ◆ javax.servlet.http API
- ◆ javax.servlet.jsp API



# 線上Servlet API

- ◆ 線上查詢 Servlet API
- ◆ 可以點取任何Web Container支援的文件，如Tomcat，或者官方JEE文件。
- ◆ <https://tomcat.apache.org/tomcat-8.5-doc/servletapi/index.html>
- ◆ <https://docs.oracle.com/javaee/7/api/>



The screenshot displays the Apache Tomcat 8.5.47 Servlet 3.1 API documentation. The browser address bar shows the URL: <https://tomcat.apache.org/tomcat-8.5-doc/servletapi/index.html>.

**Servlet 3.1 - Apache Tomcat 8.5.47**

All Classes

**Packages**

- [javax.servlet](#)
- [javax.servlet.annotation](#)
- [javax.servlet.descriptor](#)
- [javax.servlet.http](#)

**javax.servlet**

**Interfaces**

- [AsyncContext](#)
- [AsyncListener](#)
- [Filter](#)
- [FilterChain](#)
- [FilterConfig](#)
- [FilterRegistration](#)
- [FilterRegistration.Dynamic](#)
- [ReadListener](#)
- [Registration](#)
- [Registration.Dynamic](#)
- [RequestDispatcher](#)
- [Servlet](#)
- [ServletConfig](#)
- [ServletContainerInitializer](#)
- [ServletContext](#)
- [ServletContextAttributeListener](#)
- [ServletContextListener](#)
- [ServletRegistration](#)
- [ServletRegistration.Dynamic](#)
- [ServletRequest](#)
- [ServletRequestAttributeListener](#)
- [ServletRequestListener](#)
- [ServletResponse](#)
- [SessionCookieConfig](#)
- [SingleThreadModel](#)
- [WriteListener](#)

**Overview** Package Class Tree Deprecated Index Help

Prev Next Frames No Frames

**Servlet 3.1 API - Apache Tomcat 8.5.47**

**Packages**

Package	Description
<a href="#">javax.servlet</a>	The javax.servlet package contains the javax.servlet package class and the runtime environment.
<a href="#">javax.servlet.annotation</a>	
<a href="#">javax.servlet.descriptor</a>	
<a href="#">javax.servlet.http</a>	The javax.servlet.http package contains the javax.servlet.http package class running under the conforming servlet container.

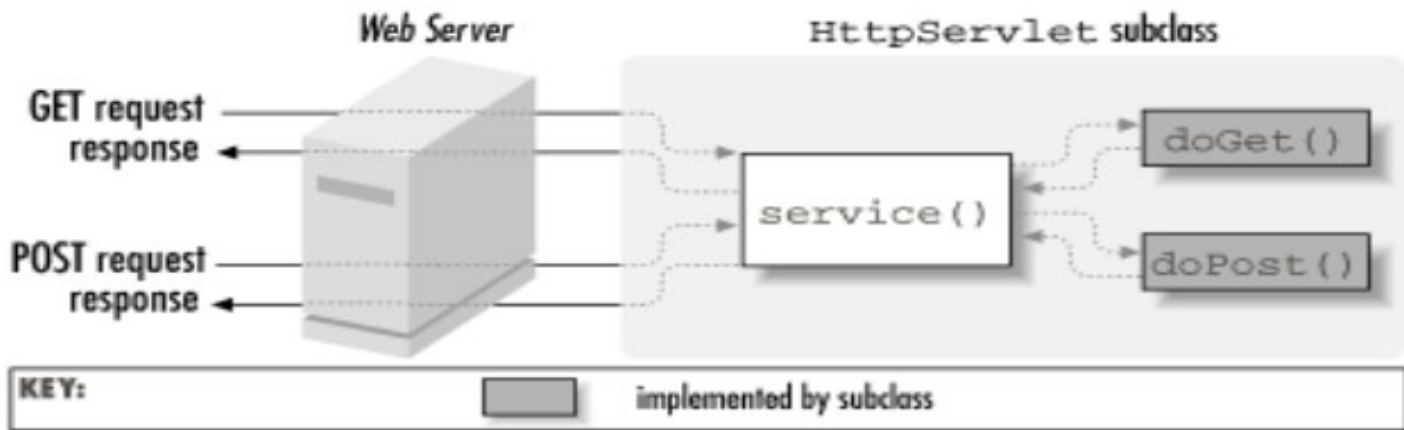
**Overview** Package Class Tree Deprecated Index Help

Prev Next Frames No Frames

Copyright © 2000-2019 Apache Software Foundation. All Rights Reserved.

# Java Web基底 Servlet架構

- ◆ Servlet整合前端來回訊息
- ◆ Servlet貫穿了HTTP的Request與Response。
- ◆ Servlet實作到HTTP Request Method方式的配合。



# Lab

- ◆ 嘗試說明Java Web網站如何運行與其基本架構
- ◆ 何謂HTTP通訊協定
- ◆ HTTP構成部分有哪幾個部分
- ◆ HTTP Request Method有哪幾種方法