**Servlet**

Servlet是用java撰寫的伺服器端程式,負責對伺服器的命令做回應與處理

撰寫Servlet/JSP程式時，必須開始接觸容器（Container）的概念，如同JVM介於Java程式與實體作業系統之間，Web Container是介於實體Web伺服器與Servlet之間。

**Servlet/JSP所運行的Web Container**

* 是一個Java寫的程式,運行於JVM之上
* Web Container是介於實體Web伺服器與Servlet之間
* Container會剖析與轉換HTTP那些文字性的通訊協定，變成你寫的Servlet/JSP中可用的Java物件（如HttpServletRequest、HttpServletResponse等）。
* 可視為Servlet/JSP的運行伺服器
* 實際Container可以是Tomcat 7(專注在Web容器的部份)，或者是Glassfish v3(完整的應用程式伺服器)

**客戶端發出HTTP GET請求時,Servlet會呼叫doGet()方法**

doGet()方法:

* javax.servlet.http.HttpServletRequest是request的Java代表物件
* javax.servlet.http.HttpServletResponse則是response的Java代表物件

Q. **Container如何分辨哪個serlet處理請求?**

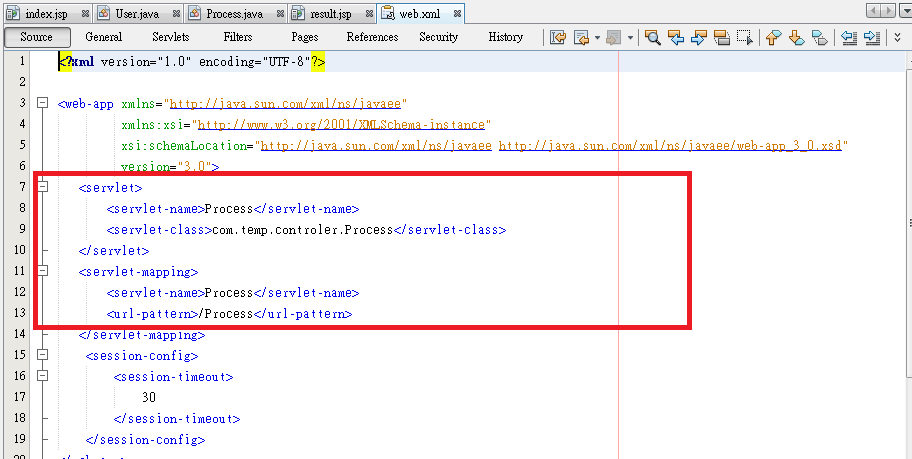
**1.程式加入:**

@WebServlet(name="HelloServlet", urlPatterns={"/hello.view"},loadOnStartup=1)

* 透過標注（Annotation）@WebServlet告知容器此Servlet的名稱是HelloServlet,
* 客戶端請求的URL是/hello.view，則由HelloServlet名稱的Servlet來處理
* loadOnStartup設定大於0的值（預設值-1）,代表應用程式啟動時，就先將Servlet類別載入、實例化並作好初始化動作

**2. web.xml檔案來定義Servlet**

‧ 如下圖紅框處



* web.xml中的設定會覆蓋Servlet中的標注設定

**Jsp**

* 容器會在你第一次存取JSP檔頁時，將JSP轉譯為.java然後再編譯為.class，之後載入容器進行處理並提供服務
* JSP最後也變Servlet, 因為JSP最後會轉譯為Servlet
* 了解Servlet越多，JSP自然就越熟悉

**MVC/Model2**

* Web應用程式世界常使用的架構是 MVC/Model 2。
* 概念類似:網頁的呈現交由網頁美術人員執行，程式碼的撰寫則由程式設計人員負責,大家各司其職,不會互相干擾
* 將Web應用程式的組成劃分為模型、畫面與控制器三個角色
* 瀏覽器發送http請求→控制器判斷要送到哪個模型→模型處理應用程式商務邏輯→控制器判斷要送到哪個畫面
* Servlet為控制器由jsp來實現畫面,模組則為java物件
* 有實作 MVC/Model2之專案:learnMVC

<http://openhome.cc/Gossip/ServletJSP/>