**Spring FrameWork技術文件**

作者:毛運韡

員工編號:9610000

**序論:**

Spring 框架是一個IoC (Inversion of Control)和AOP (Aspect-oriented programming)的構架多層J2EE系統的框架，採用IoC可以容易實現bean的配置，而AOP實現Transcation的管理，spring的好處是該框架不強迫你必須在系統的每一層都必須使用Spring，允許你根據自己的需要選擇使用它的某一個模塊；Spring優良的模組化，對不同的數據訪問技術提供了統一的介面規範。

Spring 框架的功能可以用在任何 J2EE 伺服器中，大多數功能也適用於不受管理的環境。Spring 的核心要點是：支援不綁定到特定 J2EE 服務的可重用業務和資料訪問物件。毫無疑問，這樣的物件可以在不同 J2EE 環境 （Web 或 EJB）、獨立應用程式、測試環境之間重用。Spring 框架是一個分層架構，由 7 個定義良好的模組組成。Spring 模組構建在核心容器之上，核心容器定義了創建、配置和管理 bean 的方式，下表是Spring 7 個模組概念圖:



Spring 也提供 Web 框架的解決方案，但您也可以將自己所熟悉的 Web 框架與 Spring 整合，像是 Struts、Webwork等，都可以與 Spring 整合而成為適用於自己的解決方案。 除了這些特性之外，Spring 的目標是實現一個全方位的整合框架，在 Spring 框架下實現多個子框架的組合，這些子框架之間彼此可以獨立，也可以使用其它的框架方案加以替代，Spring 希望提供 one-stop shop 的框架整合方案。

**Spring基本架構:**

Spring 的核心是個輕量級容器，主要組件是 BeanFactory，核心容器提供 Spring 框架的基本功能，它是工廠模式的實現。BeanFactory 使用控制反轉 （IOC）模式將應用程式的配置和依賴性規範與實際的應用程式碼分開。基於此核心容器所建立的應用程式，可以達到程式元件的鬆散耦合，讓程式元件可以進行測試，這些特性都使得整個應用程式可以在架構上與維護上都能得到相當程度的簡化。

Spring 是個非侵入性（Nonintrusive）框架，Spring 的核心在檔案容量上只有不到 1MB 的大小，而使用 Spring 核心所需要的資源也是很小的，它的目的之一，是讓應用程式不感受到框架的存在，減低應用程式從框架移植時的負擔。Spring 核心本身是個容器，管理物件的生命週期、物件的組態、相依注入(Dependency Injection)等，並可以控制物件在創建時是以[Prototype](http://openhome.cc/Gossip/DesignPattern/PrototypePattern.htm)或 [Singleton](http://openhome.cc/Gossip/DesignPattern/SingletonPattern.htm) 的方式來建立。

**控制反轉(Inversion of Control):**

Spring 的核心概念是IoC(Inversion of Control）控制反轉，也常稱作是依賴注入DI（Dependency Injection），使用 Spring，不必自己在程式碼中維護物件的依賴關係，只需在組態檔中加以設定，Spring 核心容器會自動根據組態將依賴注入指定的物件。控制反轉模式的基本概念是：不創建物件，但是描述創建它們的方式。在代碼中不直接與物件和服務連接，但在設定檔中描述哪一個元件需要哪一項服務。容器（在 Spring 框架中是 IOC 容器） 負責將這些聯繫在一起。在典型的 IOC 場景中，容器創建了所有物件，並設置必要的屬性將它們連接在一起，決定什麼時間調用方法。

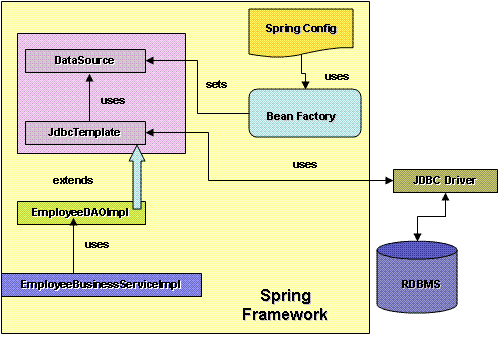
**AOP (Aspect-oriented-programming):**

Spring 最為人重視的另一方面是支援AOP（Aspect-oriented programming），然而 AOP 框架只是 Spring 支援的一個子框架，說 Spring 框架是 AOP 框架並不是一件適當的描述，人們對於新奇的 AOP 關注映射至 Spring 上，使得人們對於 Spring 的關注集中在它的 AOP 框架上，雖然有所誤解，但也突顯了 Spring 的另一個令人關注的特色。

通過配置管理特性，Spring AOP 模組直接將面向方面的程式設計功能集成到了 Spring 框架中。所以，可以很容易地使 Spring 框架管理的任何物件支援 AOP。Spring AOP 模組為基於 Spring 的應用程式中的物件提供了事務管理服務。通過使用 Spring AOP，不用依賴 EJB 元件，就可以將聲明性事務管理集成到應用程式中。

**DAO框架:**

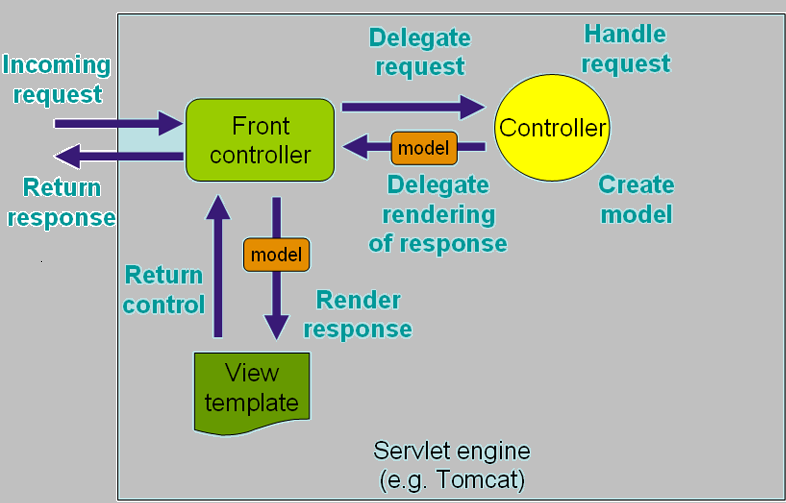
DAO的全名為Data Access Object，在應用程式中，需要使用到資料存取時，是透過一個資料存取介面來操作，而實際上進行資料庫存取的物件要實作該介面，並在規範的方法之 中，實作存取時的相關細節。JDBC DAO 抽象層提供了有意義的異常層次結構，可用該結構來管理異常處理和不同資料庫供應商拋出的錯誤消息。異常層次結構簡化了錯誤處理，並且極大地降低了需要編寫的異常代碼數量（例如打開和關閉連接）。Spring DAO 的面向 JDBC 的異常遵從通用的 DAO 異常層次結構。Spring 的DAO框架讓您在進行資料庫存取時，無須接觸到與所使用特定資料庫的技術相關細節，假設應用程式中有個物件，在進行資料庫存取時（例如find、insert、update、delete），應用程式不應依賴於一個實際的類別實作，而可以讓它依賴一個介面，下圖為Spring DAO概念圖:



**MVC框架:**

MVC 框架是一個全功能的構建 Web 應用程式的 MVC 實現。通過策略介面，MVC 框架變成為高度可配置的，MVC 容納了大量視圖技術，其中包括 JSP、Velocity、Tiles、iText 和 POI。

在Web MVC架構中，使用者並不直接連接至所需的資源，而必須先連接至前端控制器，由前端控制器判斷使用者的請求要分派給哪一個控制物件（Controller）來處理請求，藉此執到控制使用者可請求的資源之目的。在Web MVC架構下，控制物件的作用為收集使用者的請求，進行與Web層相關的動作，不應當在控制物件中執行商務邏輯，也不應當讓Servlet相關的API 侵入至商務層，這會讓商務層的物件與Servlet API產生耦合。在Spring的Web MVC框架中，擔任前端控制器角色的是DispatcherServlet，DispatcherServlet負責將客戶的請求分派給對應於請求的控制物件，下圖為Spring MVC架構概念。



**Spring ORM:**

Spring 框架插入了若干個 ORM 框架，從而提供了 ORM 的物件關係工具，其中包括 JDO、Hibernate 和 iBatis SQL Map。所有這些都遵從 Spring 的通用事務和 DAO 異常層次結構。  
  
**Spring Web:**

Web 上下文模組建立在應用程式上下文模組之上，為基於 Web 的應用程式提供了上下文。所以，Spring 框架支持與 Jakarta Struts 的集成。Web 模組還簡化了處理多部分請求以及將請求參數綁定到域物件的工作。

**Spring MVC與Struts評論:**

spring mvc的入口是servlet，而struts2是filter。

**性能：**spring會比struts稍微快。Spring MVC是基於方法的設計，而Sturts是基於類，每發一次請求都會實例一個action，每個action都會被注入屬性，而spring基於方法，細微性更細，但要小心把握像在servlet控制資料一樣。Spring3 MVC是方法級別的攔截，攔截到方法後根據參數上的注解，把request資料注入進去，在Spring3 MVC中，一個方法對應一個request上下文。而struts2框架是類級別的攔截，每次來了請求就創建一個Action，然後調用setter and getter方法把request中的資料注入；struts2實際上是通過setter and getter方法與request打交道的；struts2中，一個Action物件對應一個request上下文。

**參數傳遞與設計思想：**struts是在接受參數的時候，可以用屬性來接受參數，參數可讓多個方法共用的。而struts更加符合oop的程式設計思想， spring就比較謹慎，在servlet上擴展。

**攔截器實現機制：**struts有自己的interceptor機制，spring mvc用的是獨立的AOP方式。這樣導致struts的設定檔量比spring mvc大，雖然struts的配置能繼承， spring mvc使用更加簡潔，開發效率Spring MVC確實比struts2高。spring mvc是方法級別的攔截，一個方法對應一個request上下文，而方法同時又跟一個url對應，所以說從架構本身上spring3 mvc就容易實現restful url。struts2是類級別的攔截，一個類對應一個request上下文；實現restful url要費勁，因為struts2 action的一個方法可以對應一個url；而其類屬性卻被所有方法共用，這也就無法用注解或其他方式標識其所屬方法了。spring3 mvc的方法之間基本上獨立的，獨享request response資料，請求資料通過參數獲取，處理結果通過ModelMap交回給框架方法之間不共用變數，而struts2搞的就比較亂，雖然方法之間也是獨立的，但其所有Action變數是共用的，這不會影響程式運行，卻給我們編碼，讀程式時帶來麻煩。

**心得:**

Spring Framework 是一個 Java/Java EE/.NET 的分層應用程序框架。Spring Framework 提供了一個簡易的開發方式，這種開發方式，避免那些可能致使底層代碼變得繁雜混亂的大量的屬性文件和幫助。目前在使用Spring的開發人員中，常聽到SSH名詞也就是Struts+Spring+Hibernate，業界常使用這三個Framework相互組合來建立系統，但是並非三者是必須，對於前端MVC的使用，相較於struts，Spring提供的 MVC框架更為出色優雅也較輕量，而Struts2 的優點有提供tag使用，各有利弊。而置於後端資料庫，Hibernate，在處理一些較複雜的查詢，會顯得效能低落，而Hibernate 在製作xml的Mapping也很耗時，針對於較小的系統，可以考慮直接用JDBC再使用Spring DAO框架去管理transcation。

每一種工具都有它的好處與壞處，好的開發者能判斷出建立的系統最適合的架設方法。而Spring的最終目標，是希望程式碼與商業邏輯分離，讓PG可以專心在寫程式這件事情上，而SD、SA可以專心在設計分析這方面，讓兩者同時並行且互不干擾。

**參考資料：**

1.<http://openhome.cc/Gossip/SpringGossip/AOPConcept.html>良葛格學習筆記

2.<http://blog.developer.idv.tw/2014/05/ioc-di.html>維基百科

3.<http://projects.spring.io/spring-framework/> spring官方網站