

前言

在 2004 年時，Erik Hatcher 從非常潮溼的維吉尼亞州，旅行到平時都很乾燥的奧斯丁，而且運氣很好，巴頓河水位正高，我們一起勇敢地面對寒冷且多風的天氣。在我們停留的這段期間，正是泛獨木舟的最好日子。在 2000 年 3 月時，我和三位朋友向東方駛去，經過 Little River 及 Great Smokey Mountains 國家公園，那是一個美妙得令人懷念的日子，前方的景色被大約二吋厚的雪覆蓋著，雪花輕輕佈滿了我們的獨木舟，劃破了一個個像是 Sinks（我曾經去那兒）與 Elbow（我曾經去那兒，回來後一個禮拜睡不著覺）那樣的急流。寒冷的氣候，注入了一種難以形容的能量。這就是解凍期—也就是春天的魔力。

在 Java 的社群裡，我們正遇到另一種形式的春天。它跟隨著 Enterprise JavaBean 2.x(EJB) 的嚴寒期而來。數以萬計的應用程式，正躺在看似出於好意，實際上卻非常冰冷、幾乎致人於死的巨大冰塊裡。如果不夠細心，EJB 將吞噬開發人員的生命，而且有時甚至：

- 儘管開發人員使用了自動化測試，也無法把 EJB 應用程式測試得很好。container 因為太大，很難在測試案例中啟動，而且物件也沒辦法在 container 外面被執行。
- EJB 的侵略性是難以置信的。的確，EJB 是被設計用來保護商業物件，而不需要擔心交易、永續保存、安全、遠端呼叫與訊息傳遞等服務的細節。可是它卻做了很可怕的事情，讓你的應用程式變成 EJB 的奴隸。

- EJB 強迫你做出不合常理的設計決策。EJB 應用程式必須使用愚笨的 value object，而不能使用豐富的 domain object。就像斑衣吹笛人的魔法一樣，EJB 的遠端呼叫能力，竟然唆使開發人員以極不穩固的方式分散應用程式。EJB 在永續保存方面的限制，讓設計無法最佳化，因為它們並不支援一些關鍵的概念，例如繼承。

接著，在蹣跚行走了數年，經歷了龐大的 EJB 架構，以及那些只不過是用來暫時解決常見問題的設計模式之後，Java 社群已經準備好要迎接春天的來臨。這次並非由佔優勢的廠商所主導，而是由一種新生的輕量化開發程序，以及開放程式碼架構所引起的熱潮。

不過，許多開發人員仍然拒絕改變。對他們而言，EJB 所帶來的痛苦是值得的，因為其宣告式的服務，像是交易和遠端呼叫，仍是很有價值的。把這些服務從主要的商業邏輯切開，可以讓應用程式更簡單、更乾淨，也更容易維護。但是如果宣告式服務這麼重要，為什麼只使用 container 所提供的宣告式服務呢？為什麼不設計出更一般性的模型，讓服務可以很容易地附加到任何的 POJO 呢？

最後，我不再為這些原因寫程式了，我決定玩得開心點。要為賺錢而寫程式，並且把事情做好。如果有個好方法可以提供永續保存（像是 Hibernate 或 JDO）或交易（像是 JTA 或 JDBC），我應該可以告訴電腦，讓某個東西擁有交易、永續保存、遠端呼叫，或安全等服務。

在 Spring 中，只要在 XML 裡加上幾行程式碼，就可以讓物件擁有安全、遠端呼叫，或交易等服務。這樣的應用程式是既簡單又乾淨。在 Spring 中，你的工作量會變輕而可以提早回家，因為你再也不需要去寫一大堆在 J2EE 應用程式中常見的冗贅程式碼。你幾乎不會被無意義的細節所煩擾。在 Spring 中，你經常可以隨心所欲地寫程式，而不必擔心不良的結果會漫延影響整個應用程式。相較於過去的經驗，你將會更加快速地適應如何使用它。

這就是為什麼我已經準備好迎接這個解凍期的原因。

如何使用這本書

和其它程式高手秘笈系列的書一樣，我們並不打算假裝給你一本完整的參考手冊，而是希望讓你能儘快進入狀況，雖然一開始可能會忙不過來。你將會跟著一連串的範例學習。最後，我們希望你你可以發現 Spring Framework 的活力與前景。在你還沒更進一步之前，讓我指出對你而言可能是最感興趣的章節。

你可以取得這本書的範例，範例有完整的形式，也有每個章節與實作的個別形式。由於 Spring 是一個範圍廣泛的專案，你可能想要略過其中幾個章節。我並不打算再列一次目錄在這裡。事實上，大多數程式高手秘笈的讀者，可能都會略過前言，直到讀第三、第四次時才會看。不過，我仍應該給一些指示，以節省你的時間。

- 如果你對相依性注入 (dependency injection) 感興趣，請多留意第一章。
- 如果你喜歡開發使用者介面，應該會喜歡看第二章、第三章及第九章。其中如果你對 Spring 專有的作法感興趣，請閱讀第二章 (WebMVC) 及第九章 (Spring Rich 專案)。第三章則是把焦點放在 JSF 與 Struts 上面。
- 如果你關心永續保存，應該會喜歡看第四章及第五章。第四章介紹 JDBC，第五章則介紹 ORM 解決方案。許多人採用 Spring 的原因是它讓永續保存變得更容易。
- 如果你不太懂 AOP，以及 Spring 版本的 AOP 是怎麼運作的，應該會想看看第六章。不過如果你並不打算開發自己的服務，就可以直接省略這個章節。

總之，請多注意這本書的範例。不像一般的範例只有程式碼，這本書的範例會跟你對話。當我讀過這些範例後，經常會好奇那些決策是怎麼形成的。有了這本程式高手秘笈，就好像有人在旁邊為你實況報導一樣。我們希望你能盡情地享用它。

本書體裁

以下是本書所使用的印刷體裁：

一般文字

指的是選單標題、選單項目、選單按鈕，以及鍵盤的組合快速鍵（例如 Alt 和 Ctrl）。

粗體字

專有名詞、新名詞。

斜體字 (*Italic*)

指的是 URL、電子郵件地址、檔案名稱，以及副檔名等。

固定寬體字 (Constant width)

指的是延伸、路徑名稱、Unix 工具、命令、選項、轉換、變數、屬性、鍵、函式、型態、類別、命名空間、方法、模組、性質、參數、值、物件、事件、事件處理器、XML、HTML 標籤、巨集、檔案內容，或命令的輸出結果等。

固定寬度的粗體字 (Constant width bold)

表示命令或應該由使用者逐字輸入的文字內容。

固定寬度的斜體字 (Constant width Italic)

表示該值由使用者提供。

使用程式碼範例

這本書的目的是協助你完成工作。一般而言，你可以在自己的程式與文件裡使用這本書的範例程式碼。除非重新製作並散佈大部分的程式碼，否則並不需要與我們連繫以取得授權。例如，開發程式時使用了書中的一些範例程式碼，不需要取得授權。然而販賣或散佈歐萊禮書籍的範例程式光碟，就需要取得授權。為了回答問題，而引用這本書的內容或程式碼，並不需要取得授權。把書中大量的範例放到你自己的產品文件中，就需要取得授權。

我們會很感激你在使用程式碼範例時註明歸屬，但這不是必要。歸屬說明通常包括書名、作者、出版社、以及 ISBN。例如：「Spring：程式高手秘笈，Bruce A. Tate 與 Justin Gehtland 合著。版權為 2005 歐萊禮媒體公司所有，0-596-00910-0。」

批評指教

我們已盡力核驗本書所提供的資訊，儘管如此，我們仍不能保證本書完全沒有瑕疵，而網路世界的變化之快，也使得本書永不過時的保證成為不可能。如果讀者發現本書內容上的錯誤，不管是贅字、錯字、語意不清、或甚至技術錯誤，我們都竭誠虛心接受讀者的指教。如果您有任何問題，以下是我們的聯絡資訊：

美商歐萊禮股份有限公司台灣分公司

電話：(02) 2709-9669 傳真：(02) 2703-8802

網頁：<http://www.oreilly.com.tw>

電子郵件：

mail@oreilly.com.tw

與本書有關的線上資訊（包括勘誤、範例程式、相關連結）：

原文書

http://www.oreilly.com/catalog/springadn

中文書

http://www.oreilly.com.tw/product_java.php?id=a186

集體致謝

一本好書的背後不只有位好作者，本書也不例外。為了令人感興趣，程式高手秘笈需要一項令人讚嘆的技術，而 Spring 絕對符合這個條件。感謝 Rod Johnson 和 Juergen Hoeller，這麼細心地實踐這些想法，很敬佩你們所作的一切。也許大部分的人，都曾因為「歐萊禮」這個名字而買書，這並非偶然，這是因為所有支援讓這本書誕生的人員，一直以來的努力耕耘，你會在這本書的背後找到他們的名字，不過在此，我們想要特別提及一些人。

感謝 Kyle Hart，不厭其煩地推廣這本書，以及先前的另一本「輕快的好 Java」。感謝 Mike Loukides 的友情支持及編輯指導。你具備所有優秀編輯都擁有的第六感，而我們真誠地感激和你在一起的那段寶貴時光。

好書也需要好的校閱者。感謝 Mike，神秘的校閱者，提供了嚴厲且重要的批評，經常給予我們當頭棒喝，並且警告其危險所在。特別感謝 David Rupp，我們在後期讓你加入陣容，你也證明你的確是位最好的校閱者。不過即便如此，仍然難以避免一些未被發現的錯誤。這是我們的錯，而不是這些校閱者的錯。

我們也要感謝 JetBrains 中親切的成員，提供了非常棒的 IDE 工具，協助我們更快完成這本書。另外，要感謝那些 Java 社群與活動，例如丹麥的 JavaGruppen，讓我們有機會可以傳達這個訊息。

Bruce 的致謝

我想要感謝 Rod Johnson 的友情支持與這次的機會。祝你與 Interface 21 都有好運氣。另外感謝 Jay Zimmerman 為了支持我以及推廣這本書所付出的一

切努力。我欣賞你的事業雙贏作風，即使當我為了替你節省 8 塊錢，而買了一張要轉三次機的機票。你做了件非常了不起的事情，而我也很高興人們開始有所體認。我也想要感謝許多一路上支持我的客戶，並且在這本書還在撰寫時，能夠幫忙仔細檢閱，特別是 Trek 與 ProClarity。我希望你們能夠實踐這本書的內容，而且它能協助你們把事情做好。

大多數的作者最後都會感謝家人，而這是有充分理由的。家人要激發寫一本好書所需要的熱情，也要承受沈重的挫折感，以及寫書時經常會需要的超時工作。我相信你一定會在小女兒對你笑時，試著停止那些挫折感。感謝 Kayla 與 Julia 兩位小女兒的笑聲與笑容，家人真的給予我們非常大的鼓勵。感謝 Maggie，鼓勵我前進的好伴侶，可以伴陪著我，相信我可以完成別人無法做到的事情。我對你的愛比你認為的還要更多。

Justin 的致謝

沒有任何技術性工作像寫這本書一樣，存在於真空的狀態；寫一本書的動機，是基於一項需要被探討的偉大新技術所賦予你的使命感。因此，如果一開始沒有 Rod Johnson 與 Juergen Hoeller，我就不會有寫這本書的機會。再來，這本書是由那些跟你在一起的人所共同完成的，因此感謝所有參加 No Fluff 巡迴旅行的成員，幾乎形成了我 98% 的社交生活：Ted Neward、Dave Thomas、David Geary、Erik Hatcher、Dion Almaer、Jason Hunter、Nick Liesecki、Mike Clark，當然還有 Jay Zimmerman，你們每一位都是鼓勵我前進的人。

（再次）感謝 Stuart Halloway，你是一位很棒的商業夥伴，甚至更具有影響力。不厭其煩地，致力於學習新事物，即便有時過了不久你就忘記它們，卻仍是令人信服的引導者。

最後，我也要感謝我的家人。我的女兒 Zoe，正在學習各種美好的事物，像是 "car"、"house"、"protein"，以及 "Cthulhu"，讓我瞭解原來生命是如此美妙。還有我的妻子 Lisa，不但要容忍我的怪癖和舉止，還要勉強說出「這聽起來是個很棒的想法」這樣的話。你不厭其煩地支持與無條件的愛，是讓我持續向前的動力。