Maven

軟體專案管理工具，使用Pom檔管理專案建構,、關聯性，自動下載和專案相關的Libraries(函式庫)。

----------------------------Library dependency----------------------------

建構時檢查pom

repository (檔案庫)

本地.m2、中央<dependency>、遠程<respository>，settings.xm文件中定義存放路徑。

----------------------------Maven profile----------------------------

軟體會面對不同的執行環境，比如開發環境(DEV)、測試環境(UAT)、生產環境，讓我們不用修改配置就能釋出到不同的環境中，比如資料來源配置、日誌檔案配置。resources目錄下spring-applicationContext.xml。jdbc.properties資料來源、logback.xm日誌等等。

----------------------------Multi-module 多模組----------------------------

將原本一個大專案切成多個模組專案更容易維護、更好管理、更好重用。

parent下的business、總控、後台、ｍｅｍｂｅｒ。

（重用：略同程式碼，依賴war來達到共用程式碼）

新增一個新的後台專案。因為總控、後台、ｍｅｍｂｅｒ都為獨立的專案，

可各自打包成jar，新的專案可直接加到依賴中，不用再依賴一整個war。

專案的依賴可由父專案的pom來統一管理，不用每個專案各自維護自己的依賴。

而各模組仍可依需求在自己的pom設定依賴。不用每次都build整個專案，你只需要build你所在的專案即可，節省build的時間。

(父parent<modules>Maven Install，子<parent>Run As)

----------------------------Maven plugin 插件----------------------------

編譯插件  
打包插件

部署插件

tomcat插件；自動部屬

多模組

Parant pom，tomcat插件；定義版本

Member、Control、Platform，tomcat插件:定義路徑、Port號

Spring & SpringMVC

Spring

降低開發的複雜性，通過IoC容器管理POJO對象以及他們之間的耦合關係；通過AOP以動態非侵入的方式增強服務。七大模組使用分層架構允許你選擇使用哪一個

1.核心模組  
SpringCore模組是Spring的核心容器，它實現了IoC模式，提供了Spring框架的基礎功能。此模組中包含的BeanFactory類是Spring的核心類，負責JavaBean的配置與管理。它採用Factory模式實現了IoC即依賴注入。談到JavaBean，它是一種Java類，它遵從一定的設計模式，使它們易於和其它的開發工具和元件一起使用。定義JavaBean是一種Java語言編寫的可重用元件。要編寫JavaBean，類必須是具體類和公共類，並且具有無參構造器。  
  
2.Context模組  
SpringContext模組繼承了BeanFactory（或者說Spring核心）類，並且添加了事件處理、國際化、資源裝載、透明裝載以及資料校驗等功能。它還提供了框架式的Bean訪問方式和很多企業級的功能，如JNDI訪問，支援EJB、遠端呼叫、整合模組框架、Email和定時任務排程等。  
  
3.AOP模組  
Spring集成了所有AOP功能。通過事務管理可以使任意Spring管理的物件AOP化。Spring提供了用標準Java語言編寫的AOP框架，它的大部分內容都是基於AOP的API開發的。它使應用程式拋開EJB的複雜性，但擁有傳統EJB的關鍵功能。  
  
4.DAO模組  
DAO是DataAccessObject的縮寫，DAO模式思想是將業務邏輯程式碼與資料庫互動程式碼分離，降低兩者耦合。通過DAO模式可以使結構變得更為清晰，程式碼更為簡潔。DAO模組提供了JDBC抽象層，簡化了資料庫廠商的異常錯誤（不再從SQLException繼承大批程式碼），大幅度減少程式碼的編寫，並且提供了對宣告式事務和程式設計式事務的支援。  
  
5.ORM對映模組  
SpringORM模組提供了對現有ORM框架的支援，各種流行的ORM框架已經非常成熟，並且擁有大規模市場，Spring沒有必要開發新的ORM工具，它對Hibernate提供了完美的整合功能，同時也支援其它ORM工具。注意這裡Spring是提供各類的介面，目前比較流行的下層資料庫封閉對映框架，如ibatis、Hibernate等。  
  
6.Web模組  
此模組建立在SpringContext基礎之上，它提供了Servlet監聽器的Context和Web應用的上下文。對現有的Web框架，如Structs等，提供了整合。  
  
7.MVC模組  
SpringWebMVC模組建立在Spring核心功能之上，這使得它能擁有Spring框架的所有特徵，能夠適應多種檢視、模板技術、國際化和驗證服務，實現控制邏輯和業務邏輯的清晰分離。

----------------------------核心容器(Core Container) ----------------------------

最基本的模組、載入設定檔、管理、初始化Bean等功能

Baens:控制反轉IoC 依賴注入DI

Core:資料訪問 類型轉換 等工具類

Context:

SpEL:

Spring步驟: ----------🡪 工作範例

Pom 相關依賴>

Properties 版本

Dependencies 依賴

Resource > applicationContext.xml >

Context:component-scan 掃描包下並註冊成beans

Web.xml >

Context-param

Name 名

Value 路徑

Listener

contextLoaderListener 啟動時自動裝配配置檔

-----------------------

<properties>

<!-- 主要依赖库的版本定义 -->

<!-- 预定义依赖项的version,scope与exclusions,子项目中只需定义groupId 与 artifactId 即可 -->

<dependencyManagement>

<dependencies>

<!-- LGAME MODULES begin -->

<dependency>

<context:component-scan

自动扫描且

<!-- Spring ApplicationContext配置文件的路径

<!-- 設定Spring Context的默认Profile --

<!-- Spring MVC Servlet -->

-------------------------AOP(Aspect 導向程式設計) ----------------------------

面相切面

日誌紀錄

權限控制

方法前攔截器(執行前)

傳回後攔截器(執行後)

例外拋出攔截器(例外拋出後)

----------------------------WEB----------------------------

Spring用在WEB程式中

SpringMVC

----------------------------SpringMVC(lifecycle) ----------------------------

控制層 controller 處理請求

檢視層

----------------------------DispatcherServlet ----------------------------

SpringMVC 的入口 (web.xml)

----------------------------servlet-context----------------------------

設定檔

mybatis

持久层框架

redis

快速的開放原始碼記憶體內鍵值資料存放區，可做為資料庫、快取、訊息代理程式和佇列使用

JPA

Java持久化API , ORM規範

Quartz

----------------------------Lifecycle----------------------------

用戶 > service redis > qrartz定時器 > SQL

Tomcat

輕量級應用服務器

父

tomcat插件

<plugin>

<groupId>org.apache.tomcat.maven

<artifactId>tomcat7-maven-plugin

<version>2.2

子

自動部屬

在pom.xml檔案配置

配置Tomcat外掛

<build>

<finalName>lgame-web-restplatform<

Maven Build 啟動專案

Goals那一欄輸入：clean tomcat:run