技術筆記

筆記更新紀錄

|  |  |  |
| --- | --- | --- |
| 日期 yyyy/MM/dd | 更新人員 | 更新備註 |
| 2020/02/10 | Miles | 建立文件 |
| 2020/02/10 | Miles | 建立 對另一個專案依賴 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

目錄

[1了解 Gradle 的基本簡介 4](#_Toc32244247)

[2 Gradle有哪些特點 4](#_Toc32244248)

[3 Gradle基本結構的語法 4](#_Toc32244249)

[4關於閉包(Closure) 4](#_Toc32244250)

[5對於另一個專案的依賴方式 5](#_Toc32244251)

[6相依性配置(Dependency configurations) 5](#_Toc32244252)

[99 Gradle的CLI一些語法 6](#_Toc32244253)

## 1了解 Gradle 的基本簡介

|  |
| --- |
| Gradle是什麼?有什麼特性?有甚麼用處? |
| Gradle是一個參考Apache Ant與Apache Maven概念的專案自動化建構工具。 |
| Gradle使用Groovy語言來聲明設定，而不是使用XML，支援Java、Groovy、Scala等。 |
| Gradle與Ant有很緊密的結合，甚至在建構時可以把Ant建構指令碼直接匯入。 |

## 2 Gradle有哪些特點

|  |
| --- |
| 優點有哪些? |
| 自動處理套件相依關係 – 取得Maven Repos的概念。 |
| 自動處理部屬問題 – 取自Ant的概念。 |
| 條件判斷寫法直覺 –使用Groovy語言。 |
| 容易辨識與直觀。 |

## 3 Gradle基本結構的語法

|  |  |
| --- | --- |
| Build.gradle 基本結構語法說明 | |
| apply plugin: | **引用java plugin 獲得編譯 java 專案相關的 task** |
|  | |
|  | |
|  | |

## 4關於閉包(Closure)

|  |
| --- |
| 閉包是什麼? |
| Gradle 支援Closure，而Closure(封閉、閉包)是一個擁有閒置變數(Free Variable)的運算式。  閒置變數真正扮演的角色依當時詞彙環境而定。支援閉包的程式語言通常具有一級函式（First-class function）。建立函式不等於建立閉包。如果函式的閒置變數與當時語彙環境綁定，該函式才稱為閉包。  function doSome(){  var x = 10;  function f(y){  return x+y;  }  return f;  }  var foo = doSome(); //建立一個函式物件  foo(20); //結果為30  foo(40); //結果為50  從上面例子來看，f建立了一個閉包，f當中的x參數看似好像沒定義，其實x是從外部函式doSome()捕捉而來。閉包是個捕捉了外部函式變數(或使之繼續存活)的函式。以上範例，函式f建立了閉包，因為它將變數x關入了自己的函式當中。如果形式閉包的函式物件持續存活，被關閉的變數x也會繼續存活。就像是延續了變數x的生命週期。  看起來閉包應該是{}的x+y兩者產生的一種形式。 |

## 5對於另一個專案的依賴方式

|  |
| --- |
| 我要如何取得另一個專案的方法? |
| 直接在dependencies當中鍵入compile project(':專案名稱') 即可。 |
| 直接在dependencies當中鍵入compile project(':根專案名稱:子專案名稱')即可。 |

## 6相依性配置(Dependency configurations)

|  |
| --- |
| 有哪些相依性? |
| 1. compile |
| 專案在編譯的階段就需要這些相依函式庫內容，這包含相依庫的子依賴，這些內容最後都會加入到classpath裡去了。 例如Spring、Hibernate的相依庫。 |
| 1. runtime |
| 繼承於compile。專案在執行的時候會需要這些相依庫。例如MySQL driver的相依庫。備註：假設專案使用一個相依A，而A又相依B，專案編譯階段只需要A，但執行階段時其實是需要A和B，這也是為何配置compile的相依可以被runtime看見，但反過來卻不一定正確。 |
| 1. compileOnly |
| 繼承於compile。這些相依庫基本上只有專案編譯的階段被使用，專案執行階段的時候不被使用，這些內容最後不會被加入到runtime的classpath裡。像source-only annotations。備註：某相依庫的API在編譯時段被需要，但實作的內容則是由執行時的應用程式、其他相依、環境所提供。 |
|  |

## 99 Gradle的CLI一些語法

|  |  |
| --- | --- |
| 語法 | 說明 |
| gradle | 主要是help與歡迎Gradle。 |
| gradle tasks | 提示可以使用那些task。 |
| gradle build | 編譯gradle專案，並且產生一個jar檔案(預設)，jar檔位置在根目錄/build/…。 |
| gradle build run | 編譯gradle專案，並且run該專案的main的入口。 |
| gradle eclipse | 產生eclipse專案。 |
| gradle clean | 清理專案。 |
|  |  |