Groovy Cookbook

Java 程式設計師樂活指南

Release 1.0

by lyhcode

October 04, 2011

CONTENTS

CHAPTER

ONE

簡介

1.1 認識 Groovy

面對時下許多 Scripting 程式語言的競爭，如 PHP / Python / Perl / Ruby 等，Java 程式寫起來總是充滿「麻煩與複雜」。可是 Java 跨平台的特性、龐大豐富的函式庫，卻又令許多開發者無法割捨。

對於已經熟悉 Java 的程式設計師，未來該何去何從呢？

是否要繼續耗費更多倍的心力，只為達到其它程式語言輕鬆快速就能實現的功能？或者乾脆拋棄舊愛，投向其它程式語言的懷抱？

作者從事 Java 軟體開發的工作已有五年光陰，從一開始不喜歡 Java，到後來非常討厭 Java，卻仍然無法向 Java 說掰掰！手上的許多重要專案，主要的架構仍是採用 Java 開發。

有兩個主要因素導致這樣的結果：

1. 跨平台、跨程式語言的整合已經相當容易，用什麼程式語言開發其實不重要。選擇「合適的程式語言」解決特定的問題才是重點。Java 既有諸多成熟豐富的函式庫，仍是目前很不錯的解決方案。
2. 在 Java 的世界中，早已有許多新興的 JVM Scripting Lanaguage，可以在既有的 JVM 平台上執行，並且與原來的 Java 程式無痛整合、無縫銜接。

目前較廣泛採用的 JVM Scripting Language，包括 Jython 、JRuby 、Groovy 及 Scala 等。

本書要介紹的 Groovy ，算是 Java 的「方言」，它的語法與 Java 最為接近，但是大幅簡化。對 Java 程式設計師而言，Groovy 的學習曲線最為平滑，而且投資報酬率高，認識愈多 Groovy 的精簡寫法，就會得到愈高的 Java 程式生產力，獲得事半功倍的成效。

1.2 安裝 Groovy

只要成功安裝 Java SDK 的作業系統，通常就能使用 Groovy 開發環境，例如：

- Windows
- Linux
- Mac OS X

1.2.1 Ubuntu Linux

Ubuntu 的套件庫已提供 Groovy，可以使用 apt-get 指令安裝。

```
sudo apt-get update
sudo apt-get install groovy
```

但是 Groovy 目前的版本更新速度相當快，Ubuntu 套件庫提供的 Groovy 通常是比較舊的版本，建議從 Launchpad PPA 安裝最新版本。

```
sudo apt-add-repository ppa:groovy-dev/groovy
sudo apt-get update
sudo apt-get install groovy-1.8
```

1.3 執行 Groovy 程式

1.3.1 GroovyConsole

對剛開始接觸 Groovy 程式語言的學習者，GroovyConsole 可能是一個最友善的工具，它提供圖形化介面及程式碼編輯功能，並可以在按 Executive 按鈕後立即執行程式碼。

請執行 groovyConsole 程式，並依照以下步驟使用。

1. 輸入一段程式碼 (1..10).each {println it}
2. 按上方選單的 Script > Run 執行

執行成功將會看到下方的訊息輸出區，依序顯示 1 至 10 的數字。

以下是 GroovyConsole 常用的快速鍵。

用途	Windows & Linux	Mac OS X
執行	Ctrl + R	Command + R
清除輸出	Ctrl + W	Command + W

1.3.2 GroovyShell

Groovy 也提供一種交談式的 Shell 程式，每一次鍵入的程式碼，都可以立刻執行、得到回應。

執行 groovysh ：

```
groovysh
```

提示訊息如下：

```
Groovy Shell (1.8.2, JVM: 1.6.0_26)
Type 'help' or '\h' for help.
------------------------------------------------------------------------------
groovy:000>
```

輸入一行 new Date() 指令，按下 Enter 後可以得到以下輸出（顯示系統日期）：

```
groovy:000> new Date()
===> Tue Oct 04 16:34:40 CST 2011
```

接著再輸入一行，又會立刻得到另一個執行結果。

```
groovy:000> new Date().format('yyyy-MM-dd')
===> 2011-10-04
```

這種交談式的互動執行方式，稱為 REPL （Read - eval - print loop）。通常我們並不會用這種方式開發軟體，但是需要測試某些程式片段或語法時，這是一種相當簡便的用法。

舉例來說，java.lang.String （字串類別）有許多「轉型」方法，我們可能不記得有哪些方法名稱可以使用，通常需要查閱 Java API 文件，或是利用編輯器的自動完成（auto complete）功能。但是在 GroovyShell 中，我們只要輸入 String.to，再按 **Tab** 鍵，GroovyShell 就會幫我們找出 String 類別中，以 to 開頭命名的所以方法。

```
groovy:000> String.to

toBigDecimal()   toBigInteger()   toBoolean()      toCharacter()
toDouble()       toFloat()        toInteger()      toList()
toLong()         toSet()          toShort()        toString()
toURI()          toURL()          tokenize(        tokenize()
```

我們可以使用 GroovyShell 輕鬆解決以下的任務：

1. 快速測試一小段 Java 程式是否能順利執行
2. 測試及探索某個 Java API 的使用方法
3. 接手維護一個舊有的 Java 函式庫，原始碼和文件已經不見，使用 Tab 鍵即可快速找出某個類別有哪些 public methods 。

1.3.3 Groovy compiler

使用 Groovy 撰寫的類別，可以編譯成 Java byte code 、在 JVM 中執行。

建立一個命名為 Hello.groovy 的檔案，其程式碼內容如下。

```
println 'Hello, World!'
```

編譯成 byte code （用法類似 javac Hello.java）：

```
groovyc Hello.groovy
```

編譯成功後將會產生 **Hello.class** 檔案，執行時不需要副檔名（用法類似 java Hello）。

```
groovy Hello
```

因為 Groovy 程式碼可以透過 groovyc 編譯成.class 檔案，所以使用 Groovy 建立的函式庫，也可以用 Java 程式語言存取。

1.3.4 Groovy Script

1.3.5 執行本書的範例程式碼

本書的範例程式碼在 GitHub 的網址是：

```
https://github.com/lyhcode/GroovyCookbook
```

您可以使用 git 指令下載：

```
git clone git://github.com/lyhcode/GroovyCookbook.git
```

或是使用 GitHub 提供的 Downloads 按鈕，下載壓縮檔：

```
https://github.com/lyhcode/GroovyCookbook/zipball/master
```

src 資料夾包含本書所有範例程式碼。

例如：

```
groovy src/HelloWorld.groovy
```

如果程式執行正確，輸出如下：

```
Hello, World!
```

本書的範例程式碼採用 Groovy 1.8.2 ，並在以下作業系統環境中測試：

- Ubuntu Linux 11.04 (Natty)
- Mac OS X 10.6 (Snow Leopard)

如果在您的系統執行出現問題，煩請將問題回報給作者。

CHAPTER

TWO

RABBITMQ

2.1 安裝 RabbitMQ

2.1.1 Ubuntu Linux

使用 apt 安裝：

```
sudo apt-get update
sudo apt-get install rabbitmq-server
```

通常在安裝後，系統就會自動啟動 RabbitMQ 服務，但您也可以手動以指令啟動：

```
sudo service rabbitmq-server start
```

或重新啟動、停止：

```
sudo service rabbitmq-server restart
sudo service rabbitmq-server stop
```

2.1.2 Mac OS X

使用 port 安裝：

```
sudo port -d sync
sudo port install rabbitmq-server
```

啟動：

```
sudo rabbitmq-server &
```

2.2 Hello, RabbitMQ

RabbitMQ 提供一組 Java AMQP client [1] 函式庫，在 Groovy 程式碼中，只要以 Grape 的 @Grab 標記宣告即可使用。

```
@Grab('com.rabbitmq:amqp-client:2.6.1')
```

以下是「Hello, RabbitMQ」範例程式碼，展示簡易的 Sending / Receiving 動作如何完成。

```
/*
 * src/rabbitmq/hello.groovy
 */

@Grab('com.rabbitmq:amqp-client:2.6.1')

import com.rabbitmq.client.*

factory = new ConnectionFactory()
factory.host = 'localhost'
connection = factory.newConnection()
channel = connection.createChannel()

channel.queueDeclare('HELLO_QUEUE', false, false, false, null)

//Sending
message = 'Hello, RabbitMQ!'
channel.basicPublish('', 'HELLO_QUEUE', null, message.bytes)
println(" [x] Sent '$message'")

//Receiving
consumer = new QueueingConsumer(channel)
channel.basicConsume('HELLO_QUEUE', true, consumer)
delivery = consumer.nextDelivery()
message = new String(delivery.body)
println(" [x] Received '$message'")

channel.close()
connection.close()
```

[1] http://www.rabbitmq.com/java-client.html

CHAPTER

THREE

CONTINUOUS PUBLISHING

Book Template for ContPub (Continuous Publishing).

這本書是「ContPub」的 GitHub 範本書，您可以利用這個範本為基礎，開始製作您個人出版的書籍。

您可以在 GitHub 網址：

```
https://github.com/contpub/BookTemplate
```

取得所有本電子書的原始碼。

如果您想預覽此範本產生的電子書，請 下載 以下 PDF 檔案。

```
http://goo.gl/XHCeQ
```

您也可以利用 Google Docs Viewer 線上 預覽 此書：

```
http://goo.gl/g4T8p
```

3.1 現在起，人人都能出版！

歡迎您加入開放電子書創作行列。

Continuous Publishing 是一項電子書出版改革，ContPub 的理念就是讓每個人都能輕易利用電腦「書寫」，並且在「毫不費力」的流程下將電子書出版發行。如果您有興趣瞭解多一些，請參考以下資料：

- 敏捷技術寫作（Agile Technical Writing） https://github.com/lyhcode/AgileTechnical-Writing
- Lean Publishing http://leanpub.com/

CHAPTER

FOUR

GROOVY TAIWAN

http://www.facebook.com/groovy.taiwan/