

SA123

[首頁](#)

小技巧 | 在 Android Studio 除錯應用 (上)

2022年02月15日

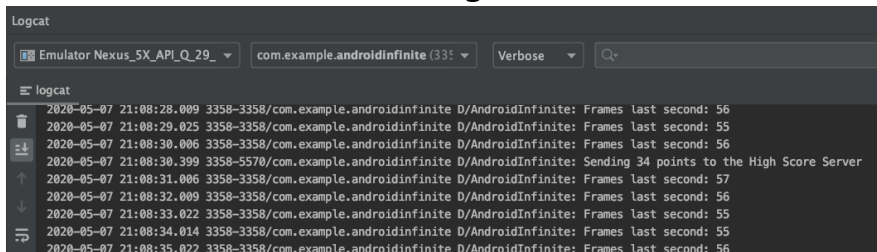


作為開發者，我們有時會被一些問題所困，導致在偵錯程式中所花費的時間甚至超過了編寫程式碼所用的時間。正因如此，最近我們找機會了解了 Android Studio 團隊在提升除錯速度方面使用的一些技巧。接下來，我們會為您一一呈現那些我們認為最好的、節省您時間的、且方便與您的除錯流程整合的小技巧。

雖然您的應用可能與本文假想中的示例應用大相徑庭，但是本文所介紹的小竅門可以用在任何應用的開發上。本文分為上下兩篇，本篇為上篇。

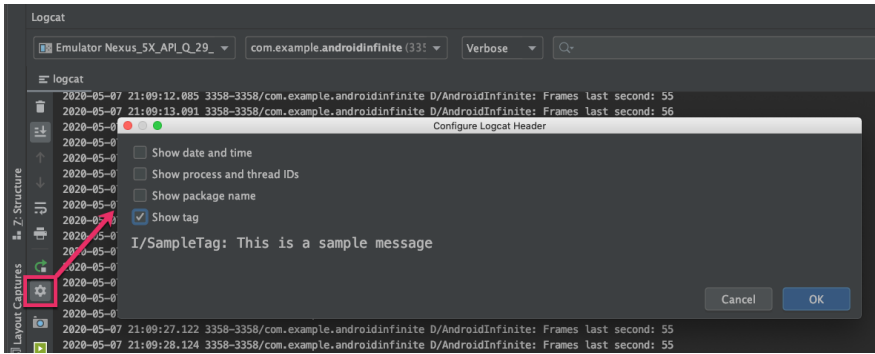
Log 的過濾與摺疊

我們從經典除錯法——`printf` 語句的一個小竅門說起。假設有一個遊戲，它會在日誌中列印它的幀數和使用者的最終得分，那麼該遊戲就會在 Logcat 視窗中給出以下內容：

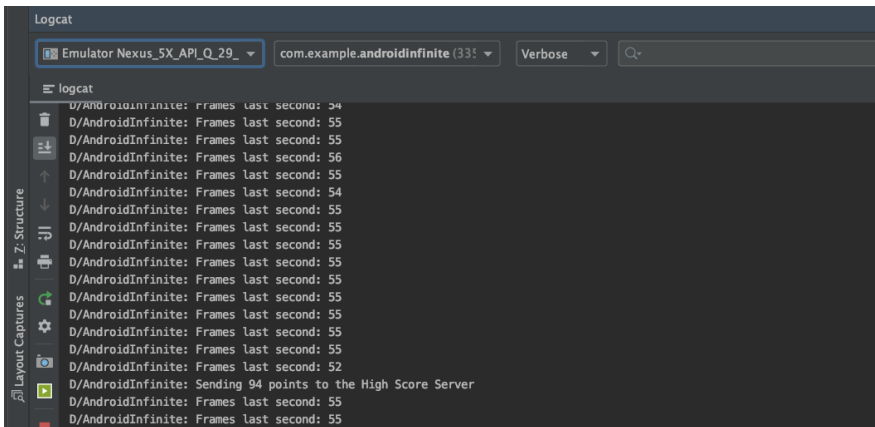


這裡的輸出資訊中可能包含了許多您並不關心的內容，例如日期和執行緒 ID。您其實可以輕鬆地調整內容的顯示，只需要點選 Logcat 工具欄中的設定圖示，便會彈出一個

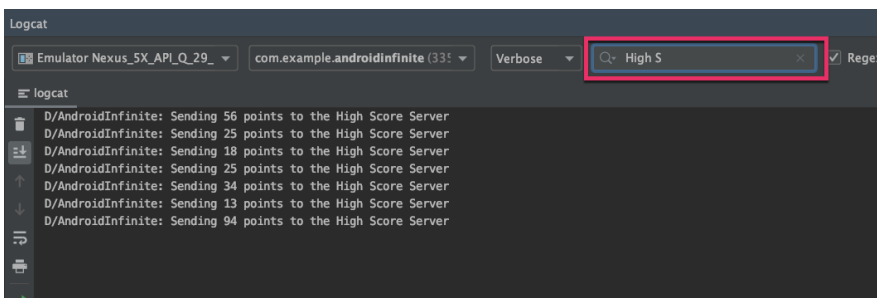
Configure Logcat Header視窗，您可以在這裡取消那些您不想看到的資訊：



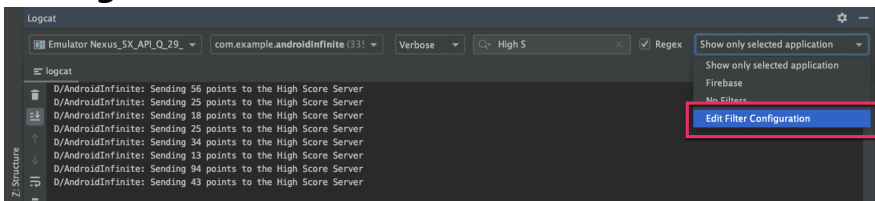
這樣一來，您便獲得了更加簡潔、更加相關的日誌輸出，如下：



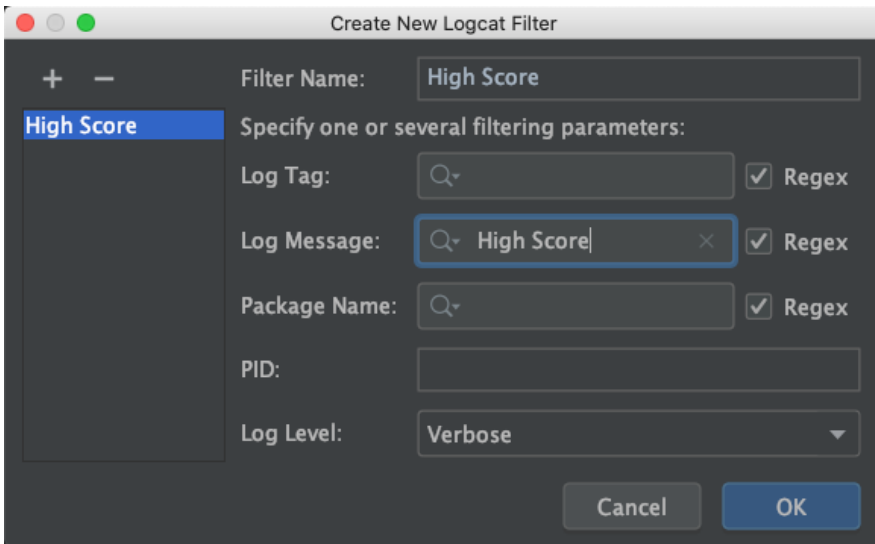
然而，這裡仍然留下了許多混淆視聽的內容，妨礙我們專注於 High Score 資訊。您可以使用搜索功能解決這一問題，只要在**搜尋**中輸入一部分除錯資訊來對**Logcat**視窗進行過濾：



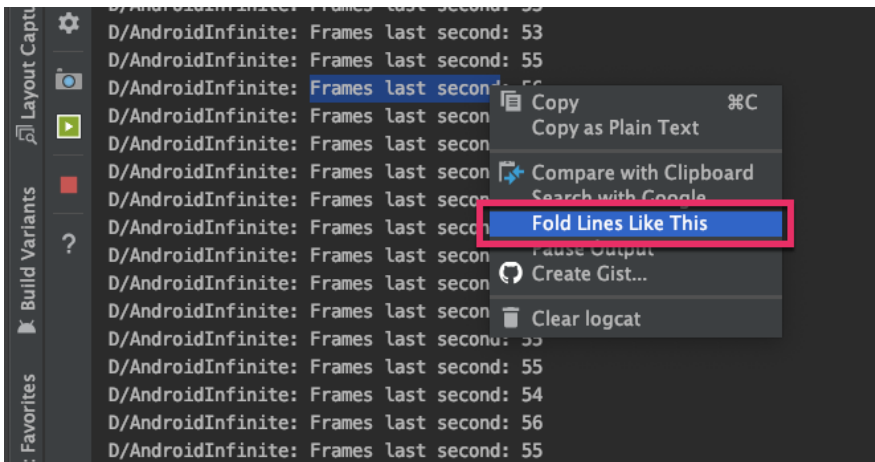
您經常會用到的搜尋條件，可以透過**Edit Filter Configuration**新增到自定義過濾器中：



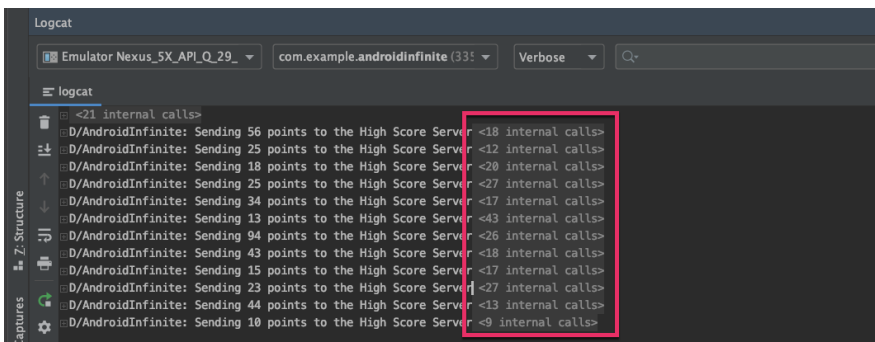
下面，新增過濾器的詳細資訊：



另一種減少混亂日誌的方式是使用摺疊功能，它可以把近似的日誌摺疊為同一組。您只需選中一條日誌中的部分文字，右擊滑鼠，並選擇**Fold Lines Like This**：



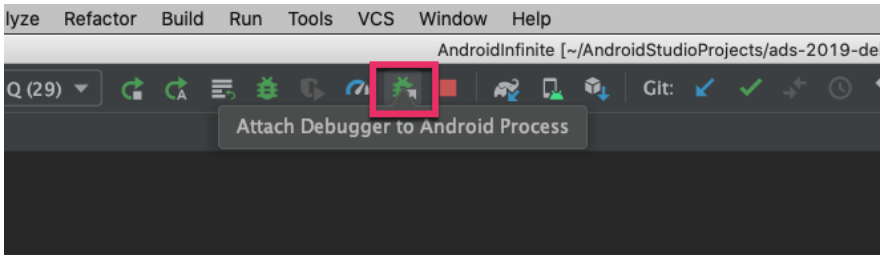
當**Console**對話框出現時，點選**OK**，就可以將包含選中文字的近似日誌整理到一起：



如果稍後您需要檢視被摺疊的資訊，則可以點選某行來展開內容。這裡也提供了方便您展開或收起摺疊行的按鈕。

在當前程序附加偵錯程式

雖然我們可以透過**Debug**按鈕或選單選項啟動一個除錯會話，但當您想要除錯一個已經啟動的應用時，也可以為其附加一個偵錯程式，這樣就可以不用重啟應用。您可以點選**Attach Debugger to Android Process**按鈕來執行這一操作：



在**Choose Process**彈窗中，選中您希望附加偵錯程式的程序並且點選**OK**。接下來，和普通的除錯會話中一樣，偵錯程式會開始觸發您的斷點。

移動斷點

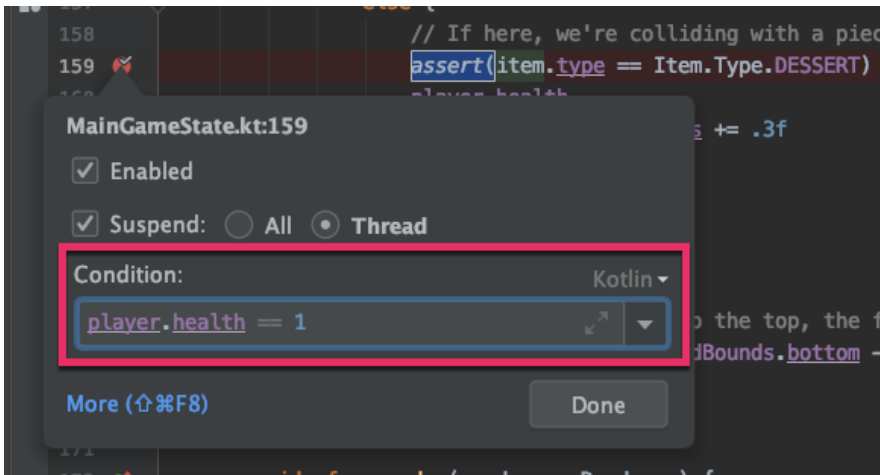
如果您發現斷點新增的位置不合適，除了清除並重設斷點外，您還可以將當前的斷點拖動至您想要的地方。這個功能十分有用，因為移動操作保留了該斷點的設定，其中包括了許多本文接下來將會介紹的特性。

條件斷點

您可能需要找到應用或遊戲中與特定型別事件有關的 Bug。舉例來說，在一個正在開發的遊戲中，您可能希望在玩家的角色與物體碰撞而耗盡其最後的生命值時停止執行。您在碰撞事件上新增斷點，但是這麼一來，每次碰撞都會導致執行停止。為了避免這種情況，您可以使用條件斷點。

為了設定條件斷點，您需要右擊一個斷點並為其新增一個條件。這裡的條件可以是任何結果為 Boolean 的程式碼表達式。當代碼執行到這一行時，如果表示式執行結果為 True，斷點就會被啟用。

這裡，在玩家碰撞到一個物體的邏輯基礎上，設定一個 `player.health == 1` 的條件，從而使您可以捕捉到玩家生命值降為 0 前的最後一次物體的碰撞事件。

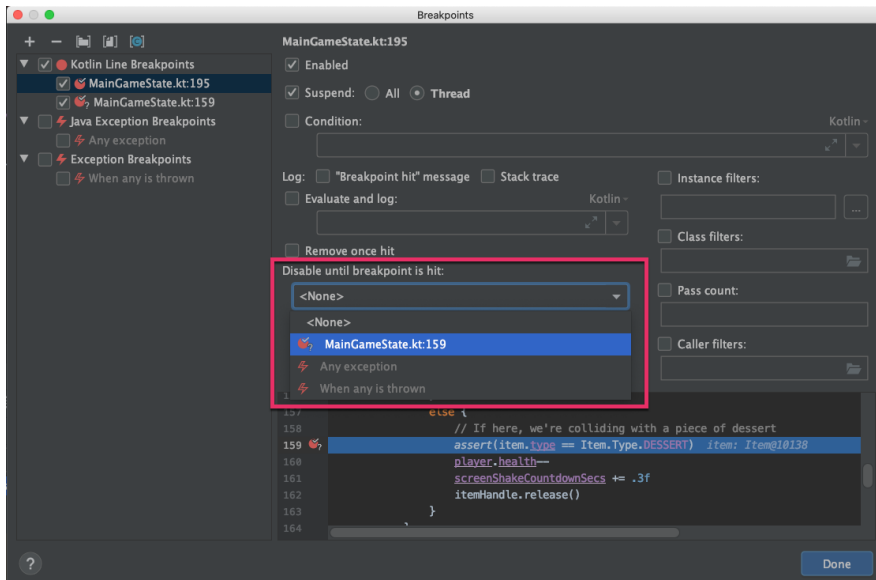


依賴斷點

一段程式碼會被不同的路徑觸發，這種情況在應用開發中並不少見。如果您發現了只會在某個特定路徑才會觸發的

Bug，隨意為其打一個斷點會造成許多無意義的執行中斷。為了應對這種情況，您可以使用依賴斷點。依賴斷點只會在特定的斷點被觸發後才會啟用。舉例來說，您可以建立一個只會在您感興趣的路徑中被觸發的斷點，而其他斷點便可以依賴此斷點，從而使這些斷點也只會在您所感興趣的路徑中被觸發。

為了設定依賴斷點，您需要右擊路徑中的第二個斷點，並開啟**More**選單。在**Disable until breakpoint is hit**選框中，選中您想要依賴的斷點：



您會發現斷點的圖示發生了改變：



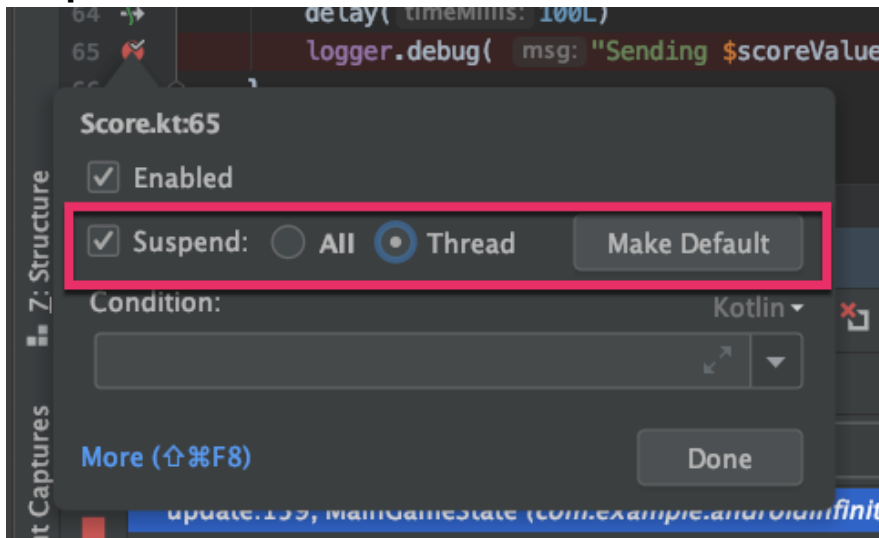
現在，您的應用只會在前一個斷點被觸發後才會在此斷點停止執行。

這個功能也可以用在其他使用了條件斷點的地方，從而可以避免複製貼上條件斷點到新位置的操作。

掛起執行緒

如果您在除錯一個多執行緒應用，您將會注意到，在預設情況下斷點將會掛起所有執行緒，但有時您可能不希望它這樣做。舉例來說，您可能想要驗證某個後臺執行緒阻塞時，應用的其他功能是否能夠正常工作，或者您希望瞭解在執行一個後臺任務時，UI 能不能夠持續進行渲染。

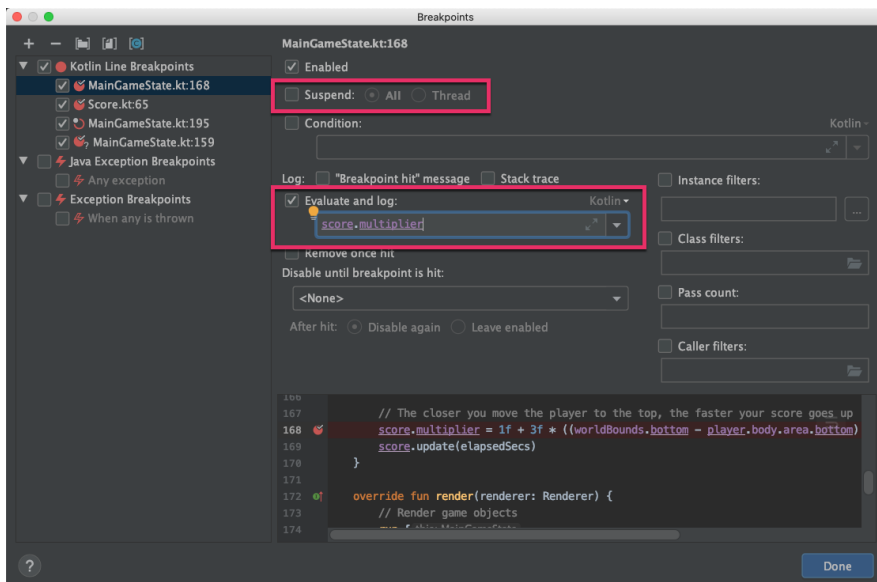
為了僅掛起當前執行緒，您需要開啟斷點選項，並且選中 **Suspend** 設定中的 **Thread** 選項：



Evaluate and log (評估與記錄)

有些時候，相比起在斷點處停止執行，您可能更希望看到一些有關應用狀態的資訊。也許您會透過在程式碼中新增 `println` 語句來做到這一點，但這種方法需要重新編譯應用，您其實可以利用斷點本身來進行評估與記錄。

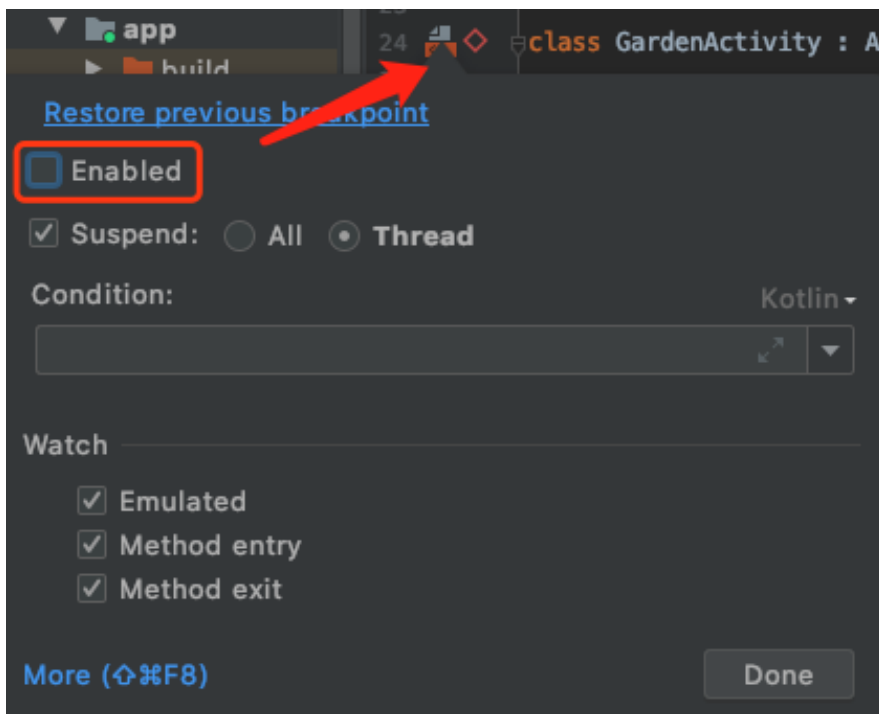
為了做到這點，您需要在斷點選項中禁用 **Suspend** 並啟用 **Evaluate and log**：



現在，您可以在輸入框中新增任何程式碼表達式，相應內容會被評估並記錄至控制檯。

如果您只是想快速驗證斷點是否觸發並且不在乎其中的細節資訊，可以使用“**Breakpoint hit**” 資訊來記錄斷點的觸發事件。您甚至可以使用 Shift + 新增斷點的方式來讓這一操作變得更加快捷。

禁用斷點



禁用斷點 (並非刪除斷點)，可以右鍵點選斷點並從彈框中取消選中 Enabled 選框。您也可以透過按住 Alt (在 Mac 上是 Option) 並點選斷點，從而更快速地禁用斷點。

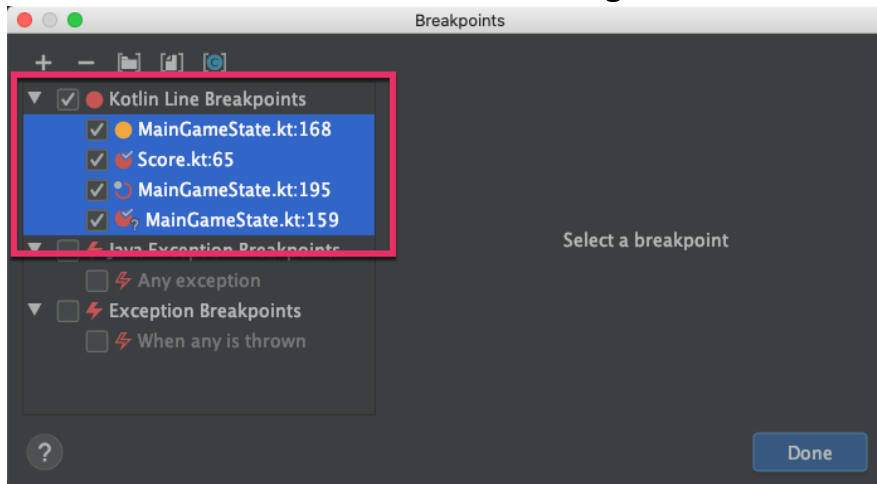
斷點分組

您一定遇到過這種場景：您正在解決一個 Bug，併為此添加了幾個斷點，但是發現自己一時間沒什麼頭緒，所以您就去解決別的 Bug 了。然而，很快您就開始觸發為了解決第

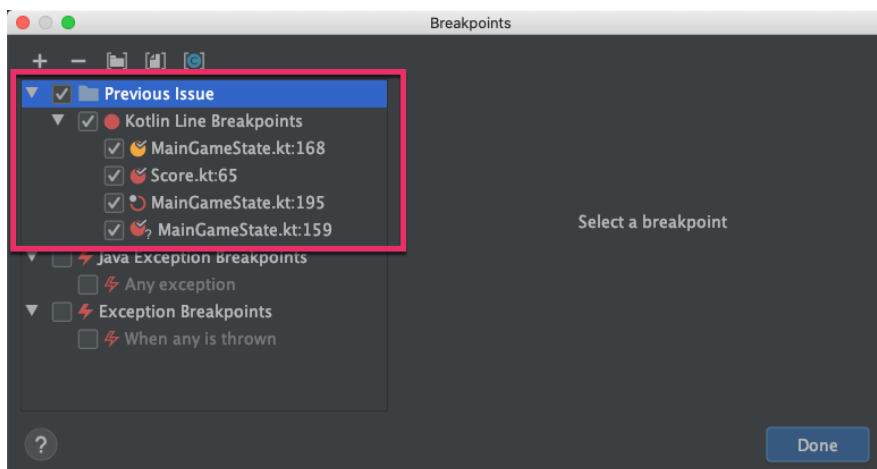
一個 Bug 所新增的斷點。觸發無關的斷點不但會讓人困擾，還會把您帶離您的除錯流程。

您可以使用斷點分組來讓開發過程更舒心一些。

當您的程式執行到第一個與當前除錯流程無關的斷點時，右擊並開啟**More**選單，這時您會看到一個所有斷點的列表，您可以在這裡複選所有與第一個 Bug 相關的斷點：



右擊選中的斷點，並選擇**Move to group**接下來**Create new**併為新的分組命名，例如以您正在處理的 Bug 命名。現在，您可以僅透過點選輕鬆地啟用和禁用所有的斷點了。



當然，您也可以在解決了 Bug 之後，使用分組功能刪除所有相關的斷點。

您可以透過下面這個影片瞭解更多本文的細節和演示內容：

■

騰訊影片連結

<https://v.qq.com/x/page/o3030ha9b5e.html>

- Bilibili 影片連結

<https://www.bilibili.com/video/av78114615/>

也請您查閱更多與本話題相關的資源:

- Android Developer 官方文件 | 除錯預構建的 APK
<https://developer.android.google.cn/studio/debug/apk-debugger>
- 透過資料瀏覽來控制資料在偵錯程式中的顯示方式
<https://www.jetbrains.com/help/idea/debugger-data-type-renderers.html>
- 如何使用和理解 Overhead 選項卡
<https://www.jetbrains.com/help/idea/monitor-debugger-overhead.html>
- Android Developer 官方文件 | Android Studio — 除錯您的應用
<https://developer.android.google.cn/studio/debug>
- IntelliJ IDEA 除錯程式碼
<https://www.jetbrains.com/help/idea/debugging-code.html>

android

推薦閱讀



點選屏末|[閱讀原文](#)|[前往Android Developer 官方文件](#) |
Android Studio — 除錯您的應用





 长按识别二维码

关注 [谷歌开发者](#)

 谷歌中国官方账号 [预测未来, 不如创造未来](#)  Google开发者

相關文章



**小技巧 | 在 Android Studio
除錯應用 (下)**



**你不知道的16條JavaScript
除錯技巧**

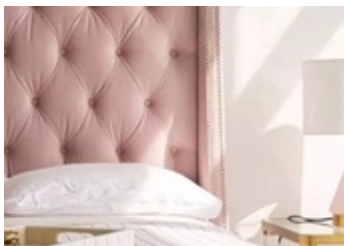


可容錯的微服務架構設計

一個女人都可能有的“婦科病”，婦產醫生卻說不用花冤枉錢去治！



治療第 4 期肺癌 在全球大流行的疫情中堅守希望



租房竟能精緻成這樣?! 這才是紐約令人髮指的神級改造!



效能診斷利器 JProfiler 快速入門和實踐



產後惡露淋漓不“淨”，這幾個小妙招可以輕鬆解決!



為何我的“湖北健康碼”一直是灰色? 權威迴應來了!

哥德爾談自己研究成果的哲學意義

