

# UML 之 Use Case Diagram

---

*Sam Xiao, Nov.21, 2017*

## Overview

---

Class diagram 與 sequence diagram 為 developer 最常用的 UML，也是看 OOP 相關書籍最常看到的兩個圖，除此之外，Use case diagram 算是 UML 令一個最具代表性的圖。

## Outline

---

### UML 之 Use Case Diagram

- Overview

- Outline

- User Story

- Task

- Use Case Diagram

  - Actor

  - Boundary

  - Use Case

- Summary

- Conclusion

## User Story

---

雖然已經提供了 class diagram 與 sequence diagram 給 user，但 user 覺的這兩張圖仍然太 developer 導向，有太多 class 細節，而 user 關心的是 功能面 的東西，也就是整個系統是否有提供 user 想要的功能。

## Task

因此我們想補上 use case diagram，描述 功能面 的部份。

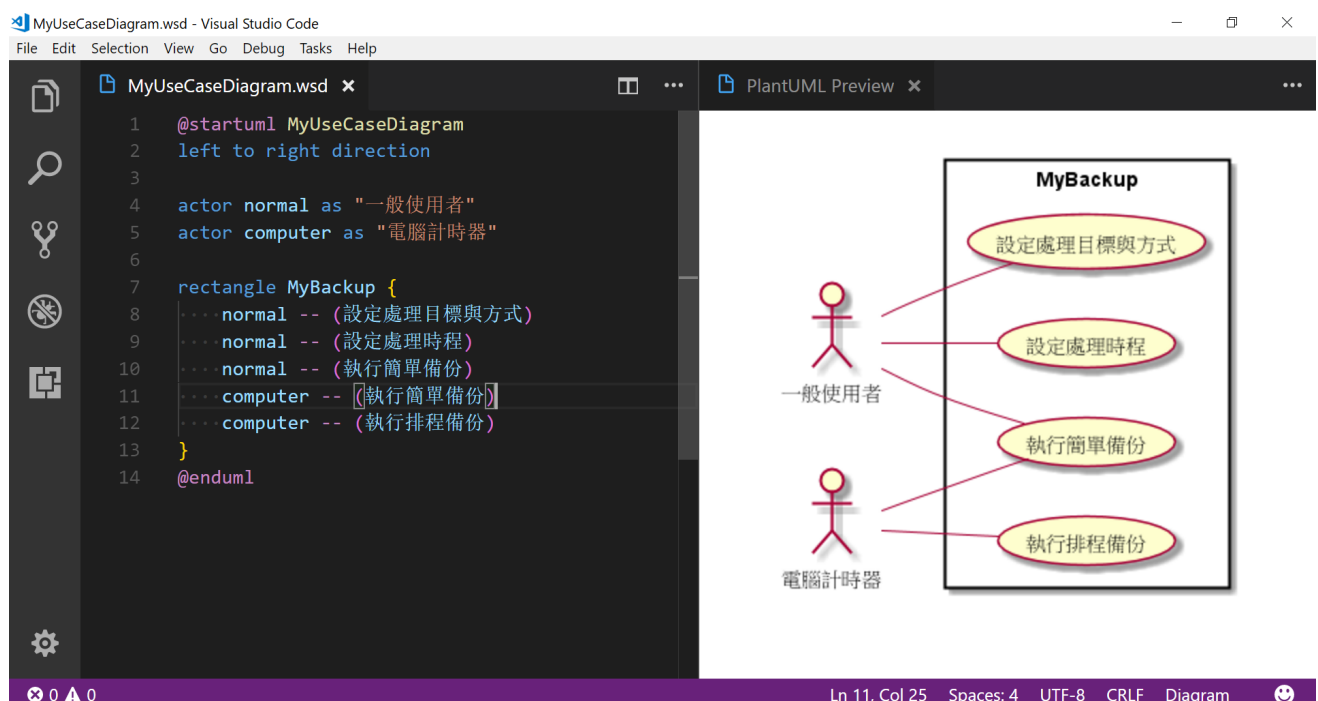
## Use Case Diagram

1. 功能面：描述系統 / 套件 / layer 所提供的 功能
2. 邊界面：定義系統 / 套件 / layer 所提供功能的 邊界
3. 角色面：定義系統 / 套件 / layer 各 角色 能使用的功能

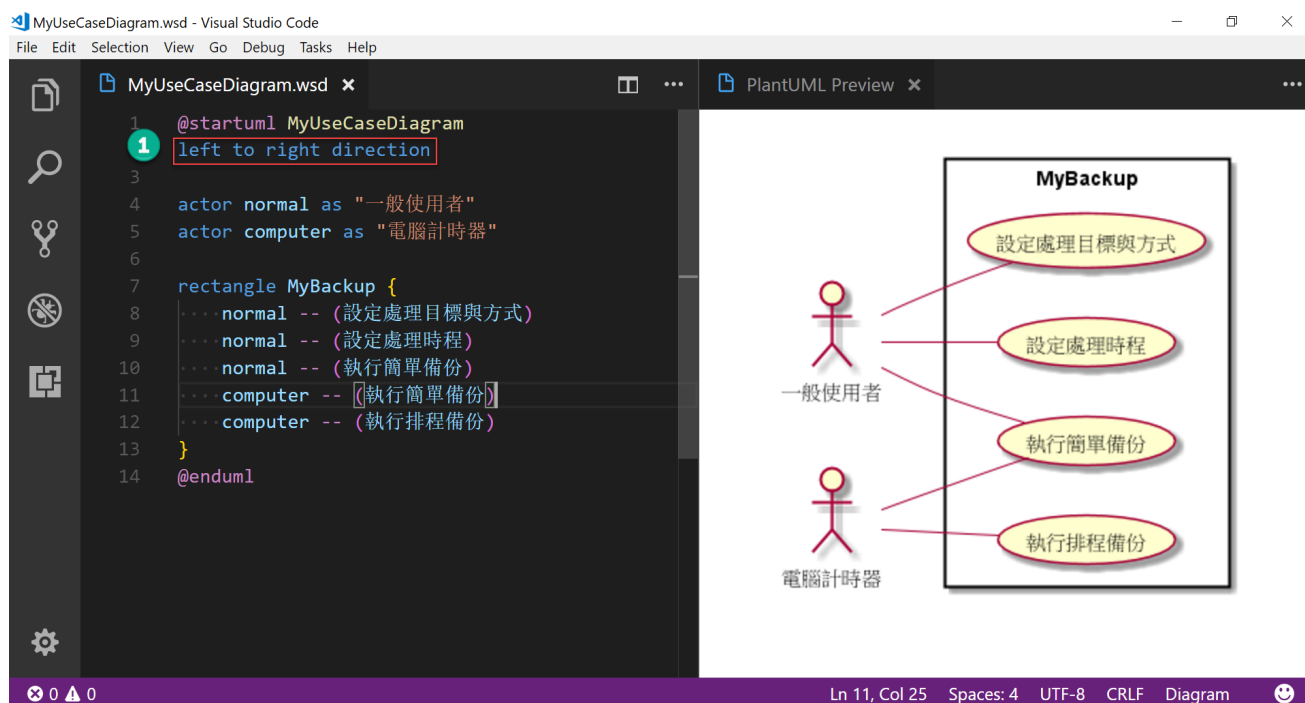
僅描述 功能面 部份，不會去描述 class 架構面、class 互動面 或 流程面 的東西。

### UML 心法

Use case 應以 user 角度繪製，而非 developer 角度



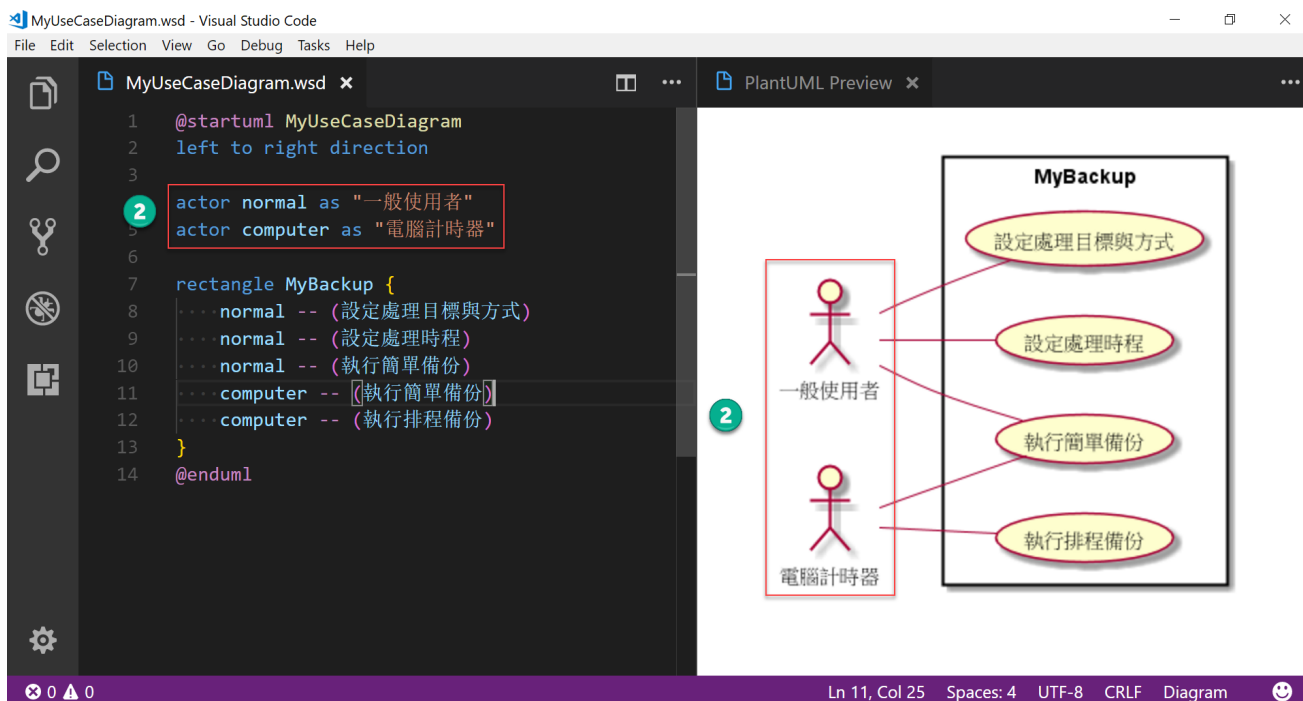
這是個最簡單的 use case diagram，用來描述 **MyBackup** 所提供的功能。



1 left to right direction

務必加上 **left to right direction**，因為 use case diagram 習慣是 **左對右** 方式，若不加此行，PlantUML 育社會以 **上對下** 的方式顯示，這並不符合 use case diagram 習慣。

## Actor



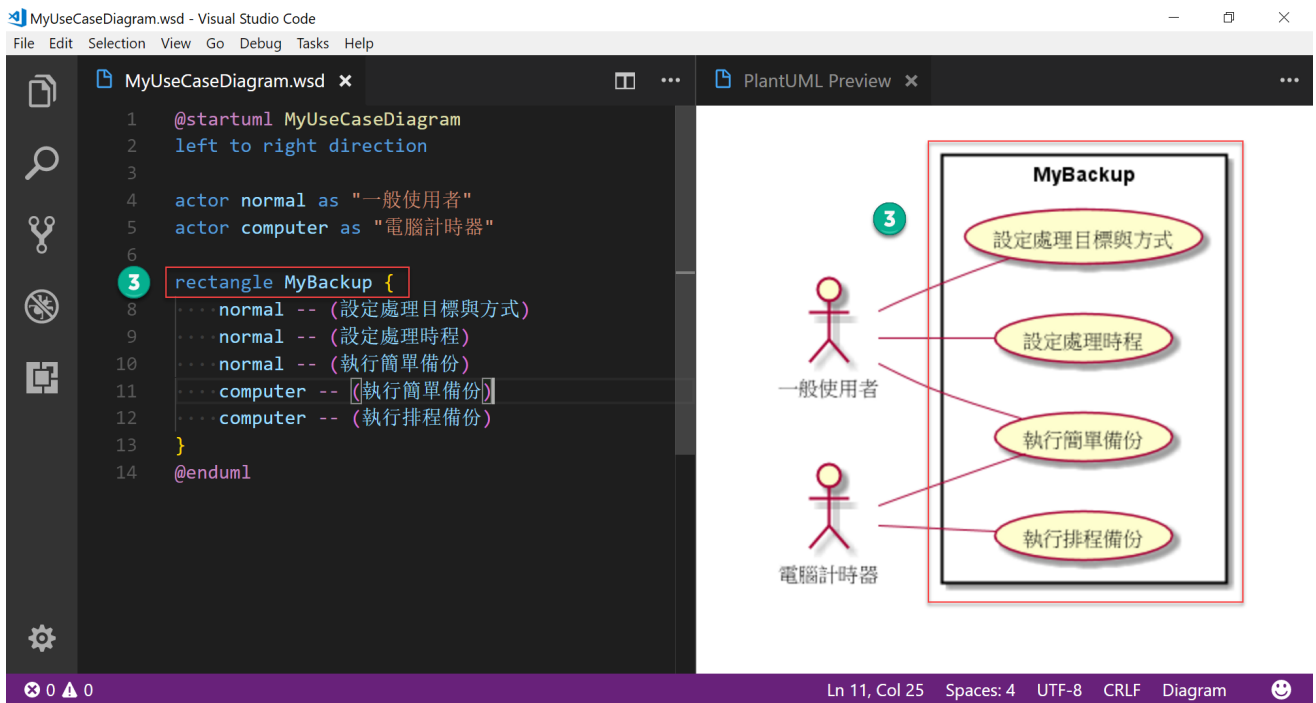
表示不同的使用者角色，以 `人型物件` 表示。

```
1 actor normal as "一般使用者"
2 actor computer as "電腦計時器"
```

以 `actor` 宣告，之後接變數名稱，供後續使用。

可使用 `as` + 中文，UML 將以中文顯示。

## Boundary

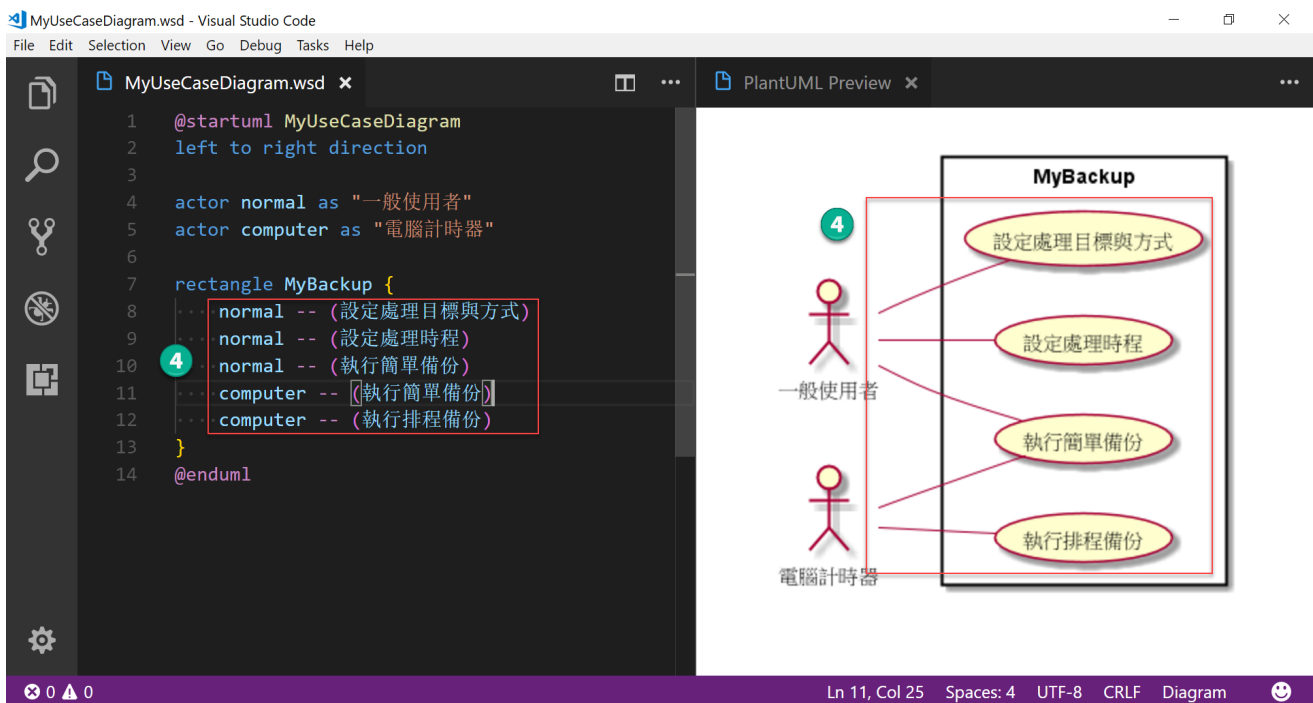


以 **矩形** 圍住，表示此系統 / 套件 / layer 的 boundary 所提供的功能，所有的 use case 都會在 boundary 內。

```
1 rectangle MyBackup {
2 }
```

以 **rectangle** 宣告，後面加變數名稱，所有的 use case 都會在 **{}** 內。

## Use Case



以 **橢圓形** 表示，敘述系統所提供的功能，與 actor 之間以 **直線** 連接。

1 | **normal -- (設定處理目標與方式)**

左側為 actor，右側為 use case，以 **()** 表示，之間可以中文顯示 **功能**。

## UML 心法

Actor 與 use case 之間，使用 **直線** 即可，不須用 **虛線** 或 **箭頭** 表示，因為只是為了表示 **功能**，還不必將 **細節** 在 use case diagram 表示

## Summary

- Use case 要盡量保持簡單，只展現 **功能** 與 **boundary**，聽眾是 user，而不是 developer。
- 可以畫多張 use case diagram，而不要試著用一張圖表示所有功能。

## Conclusion

- 請自行將 `MyBackup` 的功能以 use case diagram 表示