

Builder Pattern

可以一步一步的建構複雜的物件，不需經過傳統的初始化程序

1. Director

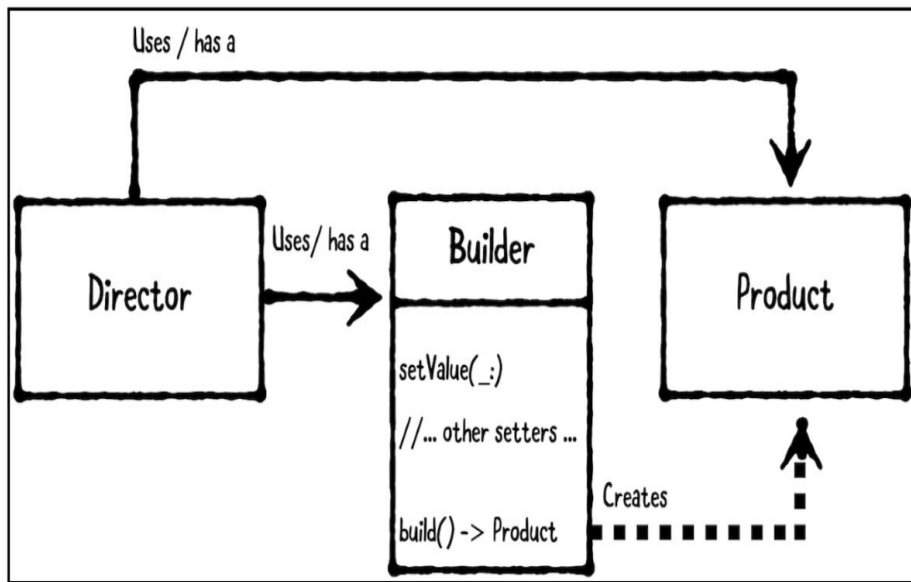
使用 builder 來創造 product
通常為 viewcontroller 或 helper class

2. Product

需要被創造出來複雜對象
可以是 struct 或 class 或其它類型

3. Builder

提供各種步驟來創建 product 的功能
類別通常是 class
可藉由reference被重複使用



When should you use it

- 1.需要分多個步驟來創造複雜的物件
- 2.當產品需要多種輸入方式時, 特別好用(可任意順序或只輸入部份資料)
- 3.Builder 會將創建產品的過程抽象化
- 4.Directer再使用Builder提供的方式再依造需求創建一個產品

一般的情況

```
public class makeBurger{  
  
    public func bigMac(meat: Meat, sauces: Sauces, toppings: Toppings){  
        //to do...  
    }  
  
    public func chickenHamberger(meat: Meat, sauces: Sauces, toppings: Toppings){  
        //to do...  
    }  
  
}  
var doHamburger = makeBurger()  
doHamburger.bigMac(meat: .beef, sauces: .mustard, toppings: .tomatoes)  
doHamburger.chickenHamberger(meat: .chicken, sauces: .mustard, toppings: .pickles)
```

Example

Hamburger model 肉 / 醬料 / 配料 (product)

Hamburger builder 加醬料 / 加配料 / 放肉 (builder)

Employee 組一個雙層牛肉吉士堡 / 組一個卡拉雞腿堡 (director)

What should you be careful about?

1. 適合複雜且需要多種輸入和步驟創建的產品的情況
2. 反之可能會產生一些問題, 你可以考慮用較簡易的 initializers 來建立產品