6-2 建構子注入依賴物件

依賴物件建構分析

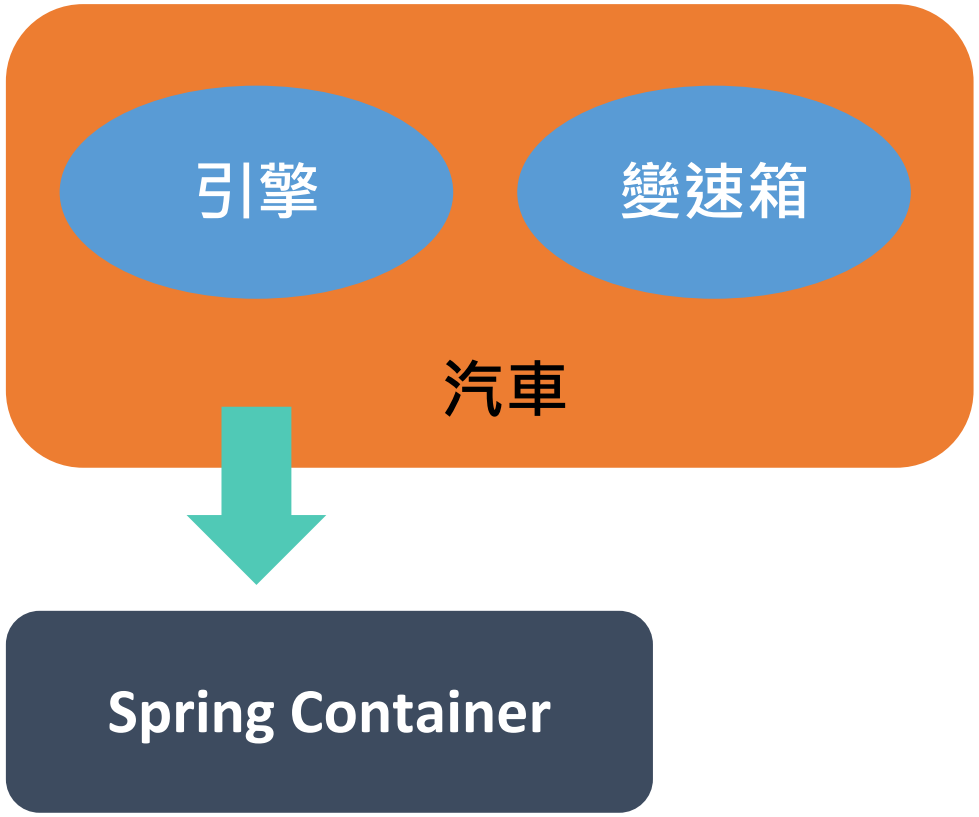
一部汽車的主要零件

- 引擎與變速箱

引擎與變速箱類別規劃

引擎與變速箱物件經由Spring Container進行零件生產

規劃汽車元件，透過建構子注入箱依賴的引擎與變速箱構成汽車物件



規劃Spring Bean-引擎與變速箱Class

- JavaBean類別規劃
- Engine Class
- Transmission Class

```java
package com.tibame.domain;
//變速箱JavaBean
public class Transmission {
    private String type;
    public Transmission(String type) {
        this.type=type;
    }
    public String getType() {
        return type;
    }
}
```

```java
package com.tibame.domain;
//汽車引擎JavaBean
public class Engine {
    private String type;
    private int cylinder; //汽缸數
    public Engine(String type,int cylinder)
    {
        this.type=type;
        this.cylinder=cylinder;
    }
    public String getType() {
        return type;
    }
    public int getCylinder() {
        return cylinder;
    }

}
```

規劃Spring Bean配置引擎與變速箱物件進入Spring Container

- 使用@Configuration進行Spring Bean配置
- 使用method建構Engine與Transmission物件
- 使用@Bean進行Spring Bean配置
- 使用@Scope配置Spring Bean生命週期

```
//配置汽缸與變數箱Spring Bean
//ConfigurableBeanFactory.SCOPE_PROTOTYPE 每次請求產生一個個體的Instance物件
@Bean
@Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE)
public Engine engine() {
    return new Engine("V8",5);
}
@Bean
@Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE)
public Transmission transmission() {
    return new Transmission("Sliding");
}
```

規劃汽車Car Component採用建構子注入箱依賴物件

採用@Component標註設定元件

- 類別自我檢測為一個Spring Bean
- 類別完整的規劃架構。
- 依賴物件可以借助屬性或者建構子注入方式進行

透過類別規劃進行建構子注入箱依賴物件

一般類別設計(POJO)方式

```java
@Component
public class Car {
    private Engine engine;
    private Transmission transmission;

    //建構子注入
    @Autowired
    public Car(Engine engine, Transmission transmission) {
        this.engine = engine;
        this.transmission = transmission;
    }
    public Engine getEngine() {
        return engine;
    }
    public Transmission getTransmission() {
        return transmission;
    }

}
```

控制器注入具有建構子注入依賴物件的Car物件應用

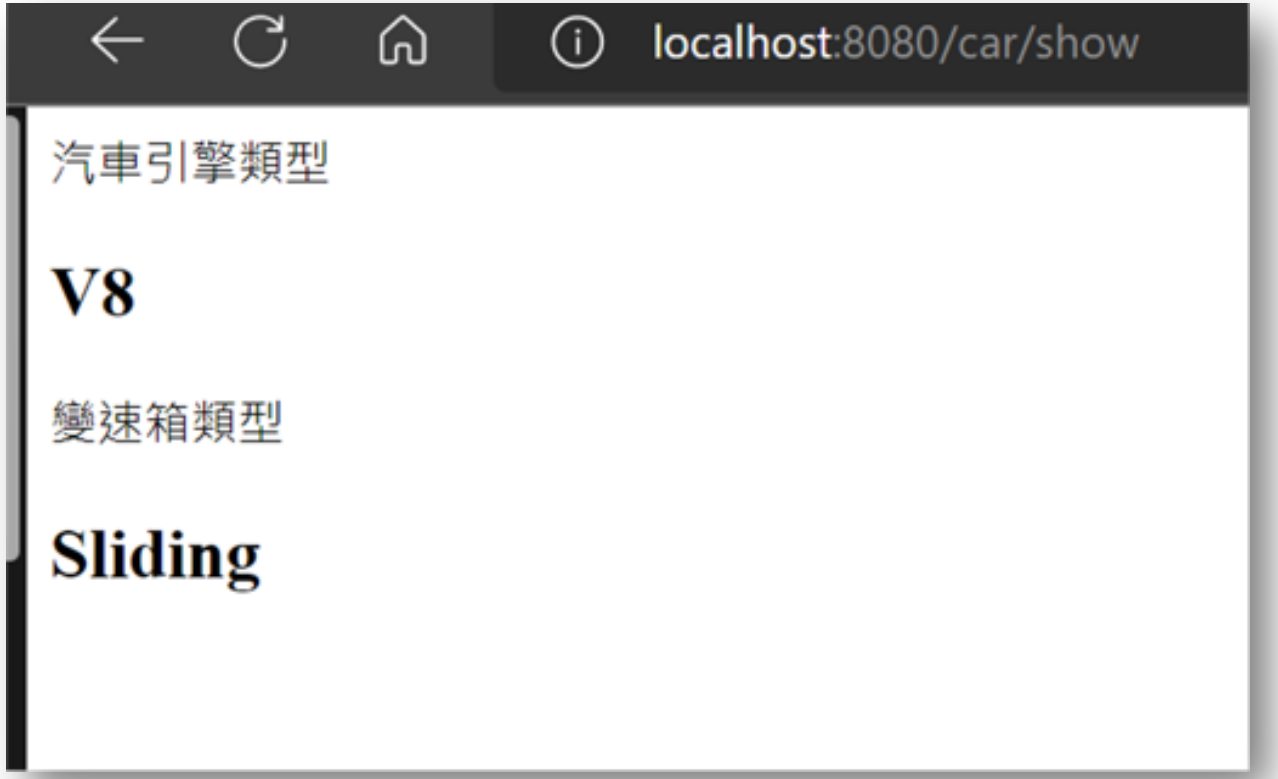
Controller採用@Autowired Data Field注入Car元件

透過Action呼叫Car元件建構子注入的相依賴物件(引擎與變速箱等)

```java
@Controller
public class CarController {

    @Autowired
    private Car car;

    @RequestMapping(path="/car/show",method= {RequestMethod.GET})
    public String showCar(Model model) {
        //參照注入的Car物件(持續狀態到View)
        model.addAttribute("car",car);
        return "carshow";
    }

}
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>建構子注入應用範例</title>
</head>
<body>
    <div>汽車引擎類型</div>
    <h2 th:text="${car.engine.type}"></h2>
    <div>變速箱類型</div>
    <h2 th:text="${car.transmission.type}"></h2>
</body>
</html>
```

建構子注入依賴物件應用



總結：6-2 建構子注入依賴物件

了解建構子注入依賴物件之後，我們來更深入了自動注@Autowired配置意義