6-4 使用@Resource注入

@Resource注入物件架構

@Resource為JDK 1.6支援的注入物件方式

與Spring @Autowired/@Inject相當，差異在搜索Bean的順序差異

@Autowired/@Inject順序為

- 類型
- 使用qualifier
- 最後使用name(名字)

@Resource搜索元件順序

- By Name
- By Type
- 最後為qualifier

規劃Bean採用name與預設狀態方式定義

- 設計一個JavaBean Customes
- 配置成一個Bean採用name屬性設定具名
- 配置一個Bean為預設狀態

```java
package com.tibame.domain;

//Customers Entity
public class Customers implements java.io.Serializable{
    //Data Field
    private String customerId;
    private String companyName;
    private String address;
    private String phone;
    private String country;
    public String getCustomerId() {
        return customerId;
    }
    public void setCustomerId(String customerId) {
        this.customerId = customerId;
    }
    public String getCompanyName() {
        return companyName;
    }
    public void setCompanyName(String companyName) {
        this.companyName = companyName;
    }
```

```java
//使用name定義一個Bean
@Bean(name="cust")
@Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE)
public Customers customers() {
    Customers customers=new Customers();
    //隨機產生編號
    int id=(int)(Math.random()*1000)+1;
    customers.setCustomerId("ID:"+id);
    return customers;

}

@Bean
@Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE)
public Customers customers2() {
    Customers customers=new Customers();
    customers.setCustomerId("0001");
    return customers;

}
```

Controller採用@Resource注入

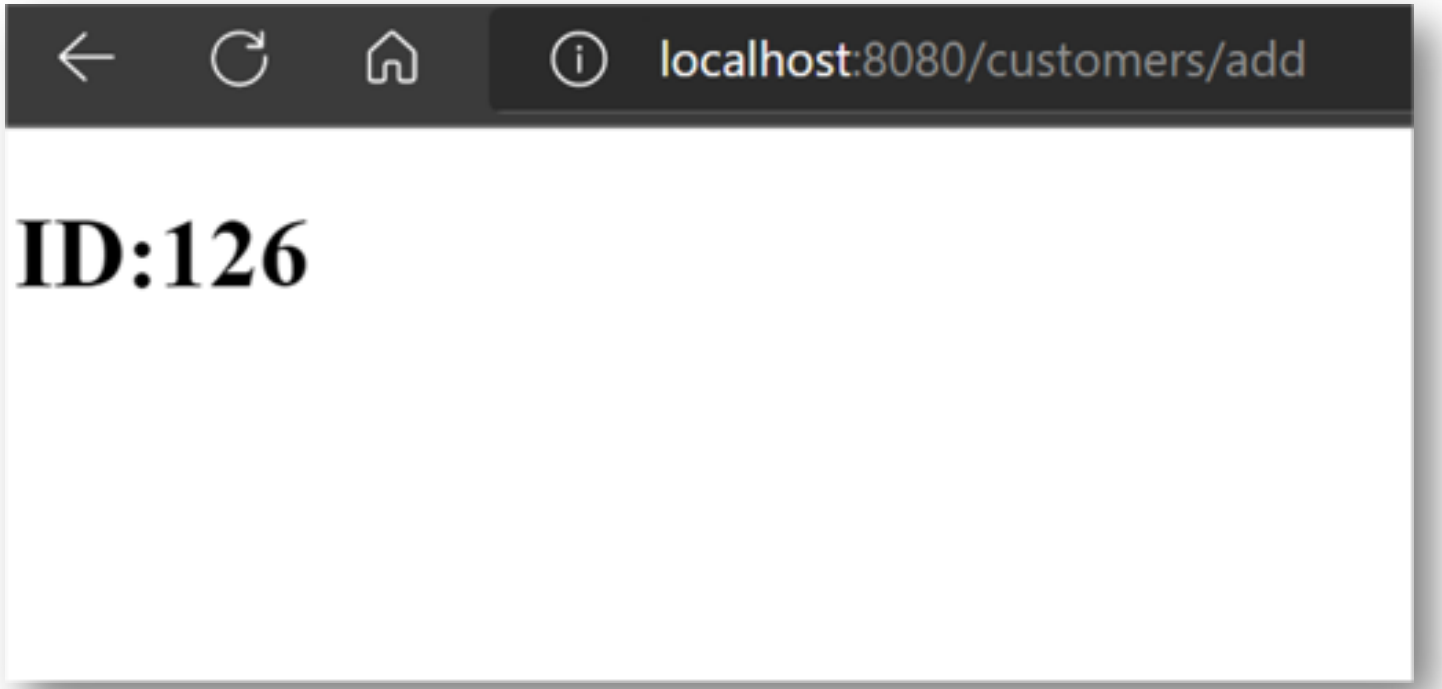
使用@Resource name 進行尋找元件依據

```
//使用name定義一個Bean
@Bean(name="cust")
@Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE)
public Customers customers() {
    Customers customers=new Customers();
    //隨機產生編號
    int id=(int)(Math.random()*1000)+1;
    customers.setCustomerId("ID:"+id);
    return customers;

}
```

```
@Controller
public class ResourceController {
    //Attribute
    @Resource(name="cust",description="by Name優先考量")
    private Customers customers;
```

Controller採用@Resource by name注入Demo



Controller採用@Resource by name注入Demo

使用@Resource注入元件

尚未設定屬性

```java
@Controller
public class ResourceController {
    //Attribute
    @Resource(name="cust",description="by Name優先考量")
    private Customers customers;

    @Resource
    private Customers customers2;
```

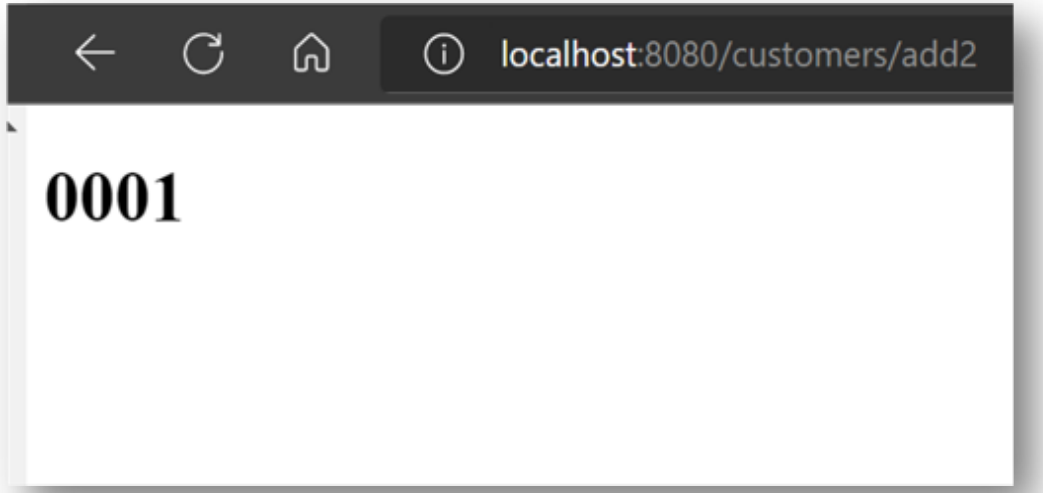
```java
@GetMapping(path="/customers/add2")
public String customersAdd2(Model model) {
    //參照注入的客戶物件
    model.addAttribute("customers",customers2);
    return "customersadd";
}
```

Controller採用@Resource by name/by type順序

```java
//使用name定義一個Bean
@Bean(name="cust")
@Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE)
public Customers customers() {
    Customers customers=new Customers();
    //隨機產生編號
    int id=(int)(Math.random()*1000)+1;
    customers.setCustomerId("ID:"+id);
    return customers;
}

@Bean
@Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE)
public Customers customers2() {
    Customers customers=new Customers();
    customers.setCustomerId("0001");
    return customers;

}
```



總結：6-4 使用@Resource注入

了解@Resource注入依賴物件架構之後，我們來了解@Inject注入意義