

# 6-1 IoC注入控制反轉





## 第六單元大綱

6-1 : IoC注入控制反轉

6-2 : 建構子注入依賴物件

6-3 : 使用@Autowired

6-4 : 使用@Resource注入

6-5 : 使用Inject注入

6-6 : 注入物件的順序性@Order應用

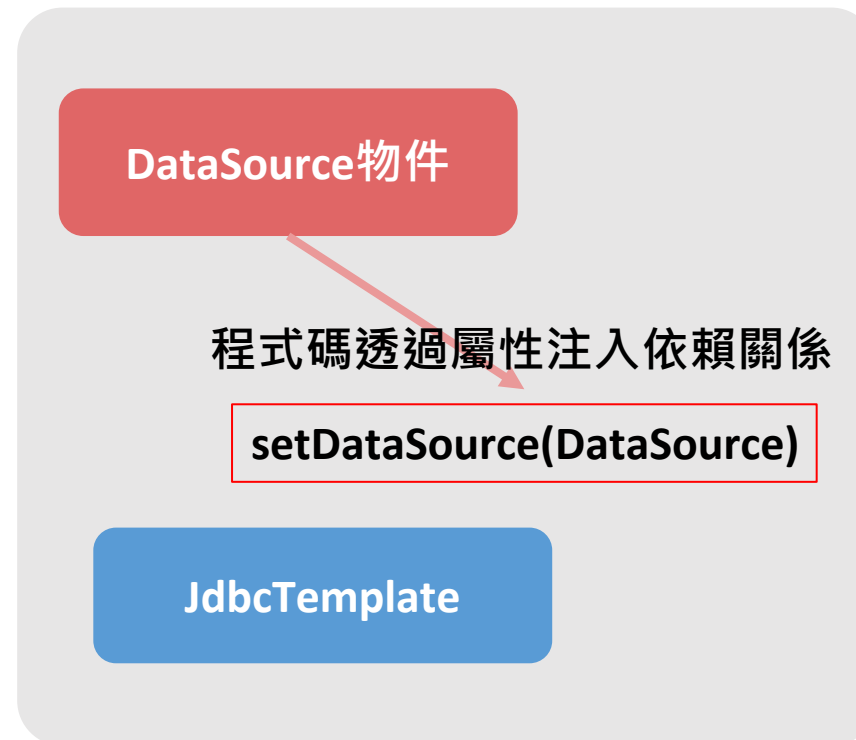


控制反轉 ( Inversion of Control , 縮寫為IoC ) , 是物件導向程式設計中的一種設計原則 , 可以用來減低物件之間的耦合度

- 一般物件之間的耦合度借助建構子注入依賴或者屬性Property注入依賴
- 程式之間會將物件關係強制注入

透過Container引擎架構自動查詢依賴的物件逕行反轉與注入依賴

如DataSource 物件與資料存取物件JdbcTemplate之間的關係



程式碼強制注入兩個物件互動耦合度高 , 彈性不足

## 透過自訂Configuration定一個Bean

- 設定Company物件產生
- 與設定必要的公司物件屬性

## 窗口物件透過Controller-Action方法注入員工物件

- 依賴關係員工必須歸屬一個公司物件的互動關係
- 採用IoC注入控制反轉實作

```
//資料庫有關的配置檔(class)
@Configuration
public class DBConfig {
    //Attribute 注入環境物件 可獲取到application.properties設定
    @Autowired
    private Environment env;

    //寫上一個方法 產生一個DataSource物件
    //產生的DataSource可以應用系統共用 生產個別使用Connection連接物件
    public SQLServerDataSource dataSource() {

        @Bean
        @Scope(ConfigurableBeanFactory.SCOPE_SINGLETON)
        public Company company() {
            Company com=new Company();
            com.setCompanyName("Tibame");
            com.setAddress("台北市");
            com.setPhone("02-12345678");
            return com;
        }
    }
}
```

```
//產生一個個體員工物件
@Bean
@Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE)
public Employees employees(Company com) {
    Employees emp=new Employees();
    emp.setCompany(com);
    return emp;
}
```

Parameter Injection  
參數注入(反轉物件注入)

## 使用@Autowired注入依賴的物件

- Employee員工物件注入

## 檢視依賴的物件反轉注入

```
//員工資料處理控制項
@Controller
public class EmployeesController {
    @Autowired
    private Employees emp;
    //產生一個員工物件進行新增頁面渲染
    @RequestMapping(path="/emp/add",method= {RequestMethod.GET})
    public String employeesForm() {
        System.out.println(emp.getCompany().getCompanyName());
        return "empform";
    }
}
```

```
Problems • Javadoc Declaration Console x
myweb - MywebApplication [Spring Boot App] [pid: 2172]
2022-07-29 13:49:08.754 INFO 2172 --- [main]
2022-07-29 13:49:13.053 INFO 2172 --- [nio-8080-exec-1]
2022-07-29 13:49:13.053 INFO 2172 --- [nio-8080-exec-1]
2022-07-29 13:49:13.053 INFO 2172 --- [nio-8080-exec-1]
Tibame
```

## 規劃JavaBean

- Setter and Getter Property對應表單欄位
- 大小寫需要對應

採用Method參數注入JavaBen配合傳遞表單欄位進行封裝資訊

```
//具有setter and getter
public class MemberEntity implements java.io.Serializable{
    //Attribute
    private String userName;
    private String password;
    private String email;
    public String getUserName() {
        return userName;
    }
    //表單欄位 name=userName 透過Property setXxx() getXxx() 去頭 xxx=value
    public void setUserName(String userName) {
        this.userName = userName;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
```



## 總結：6-1 IoC注入控制反轉

了解IoC注入控制反轉架構之後，我們  
來看看如何使用建構子注入依賴物件

