

# 5-3 @RequestBody-表單 欄位傳遞應用



## <Form> </Form> 表單設計

- 一般採用Request Method為POST

透過表單欄位Form Field name Attribute  
規劃傳遞識別碼

Spring MVC Web具有Inteceptor進行注  
入的Entity物件屬性對應封裝表單欄位

```
//註冊表單頁面調用action  
//提供請求方式GET 設定端點  
@GetMapping(path= {"/member/register"})  
public String memberAdd() {  
    //直接調用頁面  
    return "register";  
}
```

```
<body>  
    <!--設計表單頁面-->  
    <fieldset>  
        <legend>會員註冊</legend>  
        <form method="post" action="register">  
            <div>使用者帳號</div>  
            <input type="text" name="userName"/>  
            <div>密碼</div>  
            <input type="password" name="password"/>  
            <div>EMAIL</div>  
            <input type="text" name="email"/>  
            <br/>  
            <input type="submit" value="註冊"/>  
            <input type="reset" value="取消"/>  
        </form>  
    </fieldset>  
</body>
```

## 表單欄位數量少

- 透過表單欄位自動對應Action參數進行傳遞
- 表單欄位name attribute與參數名稱對應

```
//前端傳遞進來的註冊資訊
//表單欄位form field name=xxxx 對應參數名稱 自動封裝進來
@PostMapping(path= {"/member/save"})
public String memberSave(String userName, String password, String email) {
    //TODO會員註冊新增作業

    System.out.println("新增資料:"+userName);
    return "ok";
}
```

```
<body>
  <!--設計表單頁面-->
  <fieldset>
    <legend>會員註冊</legend>
    <form method="post" action="save">
      <div>使用者帳號</div>
      <input type="text" name="userName"/>
      <div>密碼</div>
      <input type="password" name="password"/>
      <div>EMAIL</div>
      <input type="text" name="email"/>
      <br/>
      <input type="submit" value="註冊"/>
      <input type="reset" value="取消"/>
    </form>
  </fieldset>
</body>
```

## 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JavaBean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：5-3 @RequestBody-表單欄位傳遞應用

了解@表單欄位自動封裝Java Bean 屬性操作  
，我們來看看控制項執行過程中，依賴物件注入操作手法

