

7-5 呈現錯誤訊息 於Thymeleaf中



Spring Errors可配合物件進行資料綁定與驗證訊息應用

配合表單欄未進行綁定與驗證訊息用

可借助DI-Dependency Injection注入到控制項中進行客製化驗證訊息收集

借助Model傳遞到View Page進行錯誤訊息渲染

JavaBean客戶類別規劃

Form表單欄位name Attribute屬性設定

注入Customers物件封裝表單欄位Action設計

```
//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;
    }
}
```

```
<form method="post">
    <div>客戶編號</div>
    <input type="text" name="customerId"/>
    <div>公司行號</div>
    <input type="text" name="companyName"/>
    <div>聯絡地址</div>
    <input type="text" name="address"/>
    <div>連絡電話</div>
    <input type="text" name="phone"/>
    <div>國家別</div>
    <input type="text" name="country"/>
    <br/>
    <input type="submit" value="新增"/>
</form>
```

Server Side進行表單輸入驗證

採用Controller-Action進行驗證

透過注入的Errors物件進行驗證錯誤訊息設定

使用注Model介面進行狀態管理

```
//-----error錯誤訊息如何呈現-----  
//客戶新增作業表單頁面  
@RequestMapping(path="/customers/addform",  
    method= {RequestMethod.GET, RequestMethod.POST})  
public String customersAddForm(Customers customers,  
    HttpServletRequest request, Model model, Errors errors) {  
    System.out.println("錯誤物件:"+errors);  
    //判斷傳送方式  
    if(request.getMethod().equals("POST")) {  
        //進行輸入驗證  
        if(customers.getCustomerId()==null || customers.getCustomerId().length()  
            {  
                errors.rejectValue("customerId", "客戶編號必須要輸入!!!");  
            }  
        if(customers.getCompanyName()==null || customers.getCompanyName().length()  
            {  
                errors.rejectValue("companyName", "公司行號必須要輸入!!!");  
            }  
        System.out.println(errors);  
    }  
    else
```

```
//持續錯誤狀態  
boolean isError=false;  
if(errors.hasErrors()) {  
    isError=true;  
    System.out.println(errors.getAllErrors().get(0).getCode());  
}  
  
model.addAttribute("errors", errors);  
model.addAttribute("iserror", isError);  
  
return "customersaddform";
```

```
<fieldset th:if="isError">
  <legend>錯誤訊息</legend>
  <ul>
    <li th:each="error: ${errors.allErrors}">
      <label th:text="${error.code}"></label>
    </li>
  </ul>
</fieldset>
```

localhost:8080/customers/customers/addform

客戶編號

公司行號

聯絡地址

連絡電話

國家別

新增

錯誤訊息

- 客戶編號必須要輸入!!!
- 公司行號必須要輸入!!!



總結：7-5 呈現錯誤訊息於Thymeleaf中

了解Spring Errors介面管理錯誤訊息應用
之後，以及如何使用thymeleaf 進行錯誤
訊息渲染，接下來我們要深入thymeleaf
View進階渲染技術

