



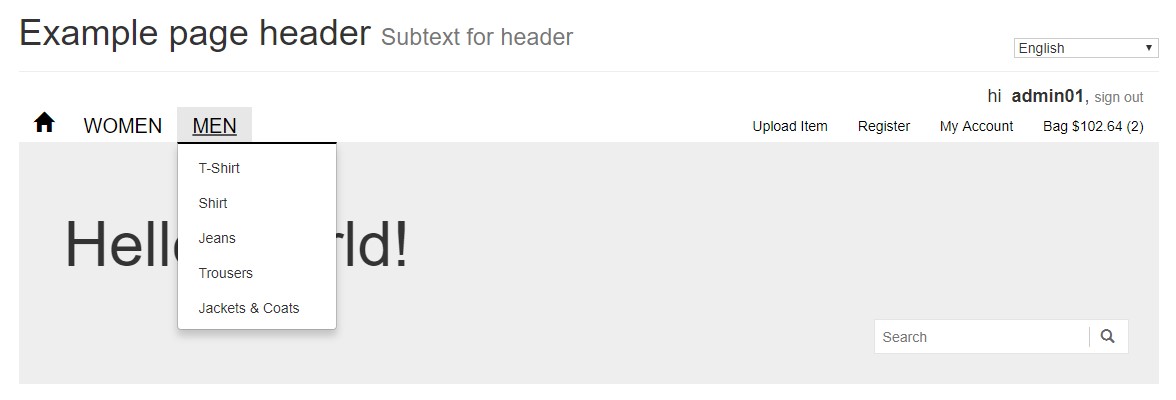
**shoppingPage.jsp**

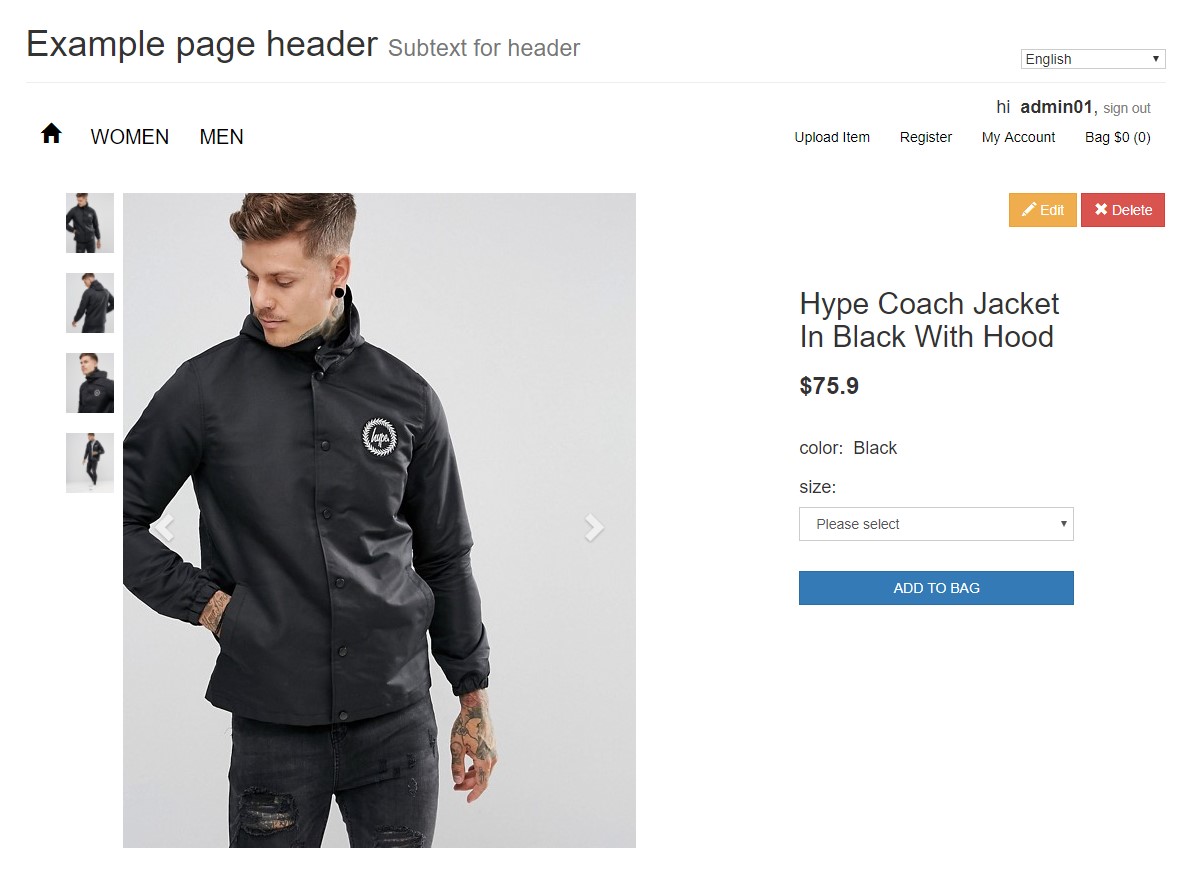
1.各項目連結

2.顯示購物車數量及金額

3.顯示商品並使用分頁

4.各種條件過濾連結、Filter、商品搜尋





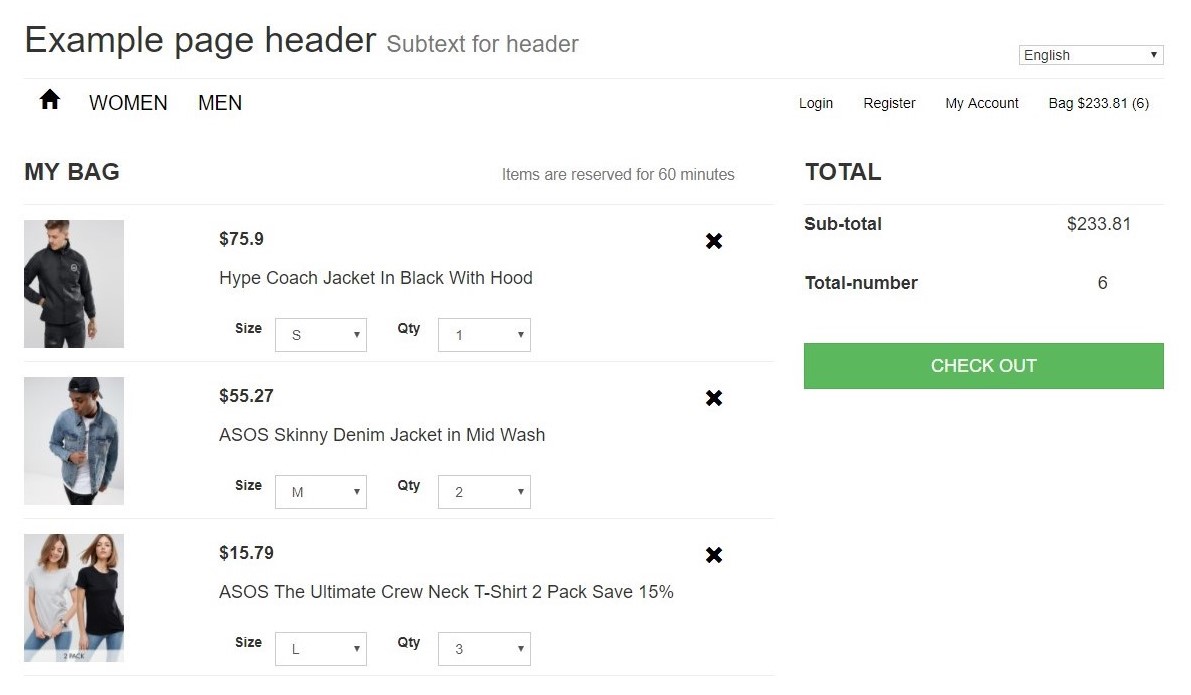
**clothesInfo.jsp**

1.商品相關細節

2.加入購物車

3.圖片以選轉木馬的方式展示 [bootstrap提供的方法]

4.擁有admin角色的使用者會有Edit、Delete商品的按鈕



**cart.jsp**

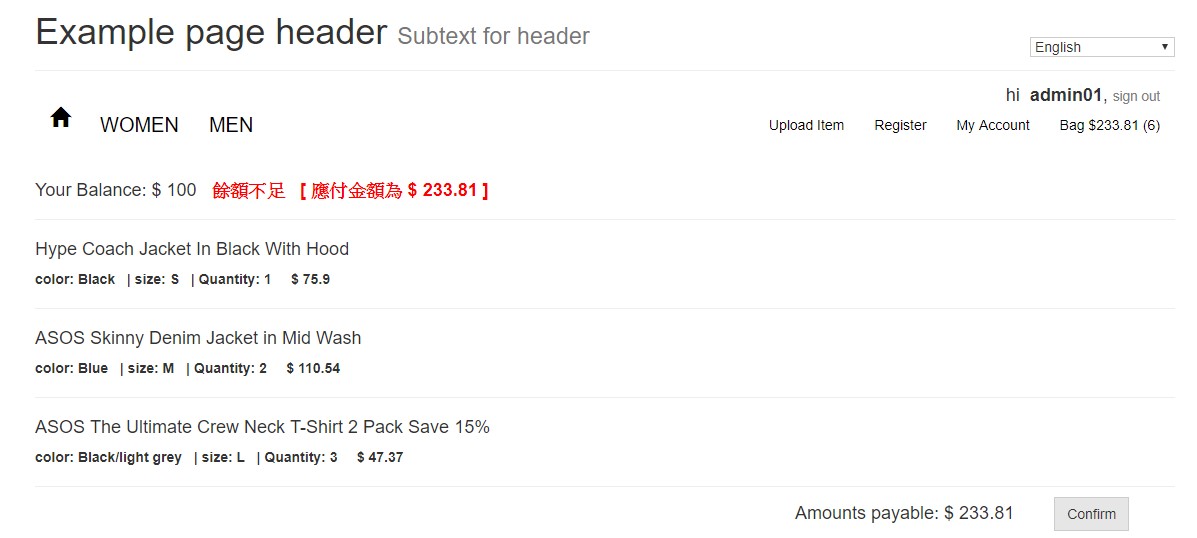
1.購物車中商品detail

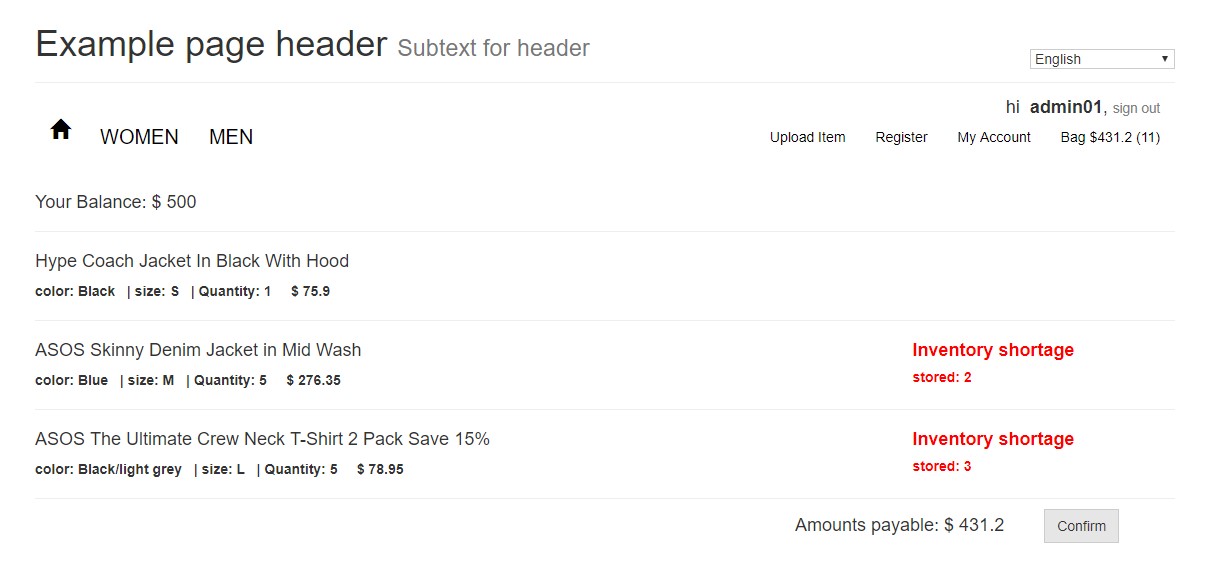
2.可修改商品數量

3.可修改商品size

4.移除購物車中商品

5.checkout





**checkout.jsp** [需認證過後才能訪問]

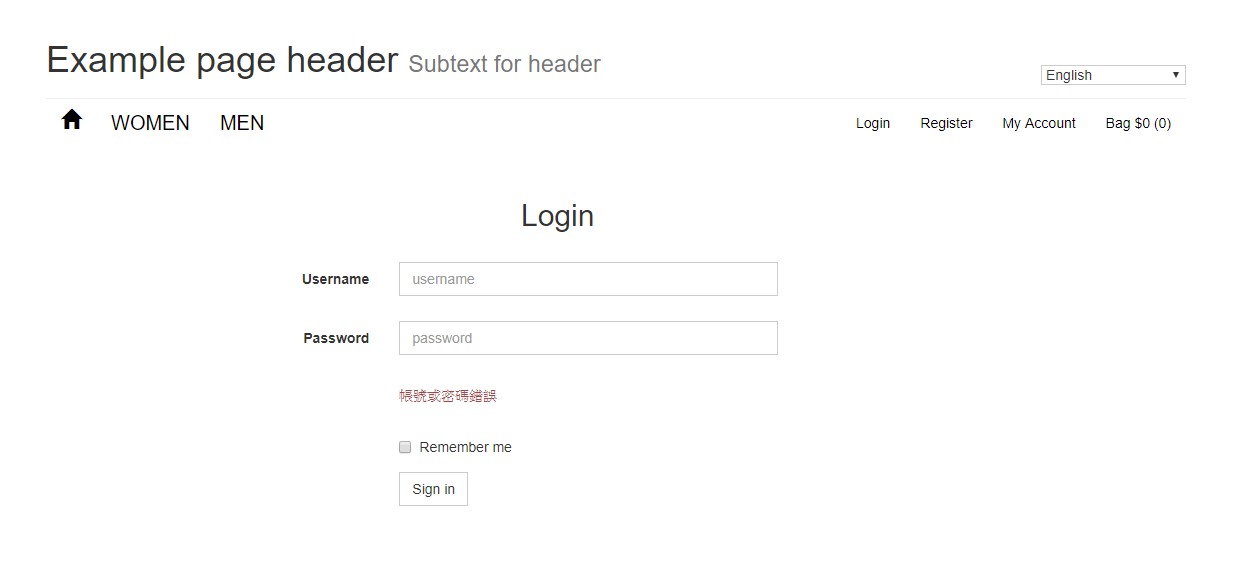
1.顯示用戶餘額

2.顯示購買清單

3.若餘額、庫存不足，顯示錯誤訊息

4.確定交易

5.若交易成功，創建Trade紀錄交易資訊

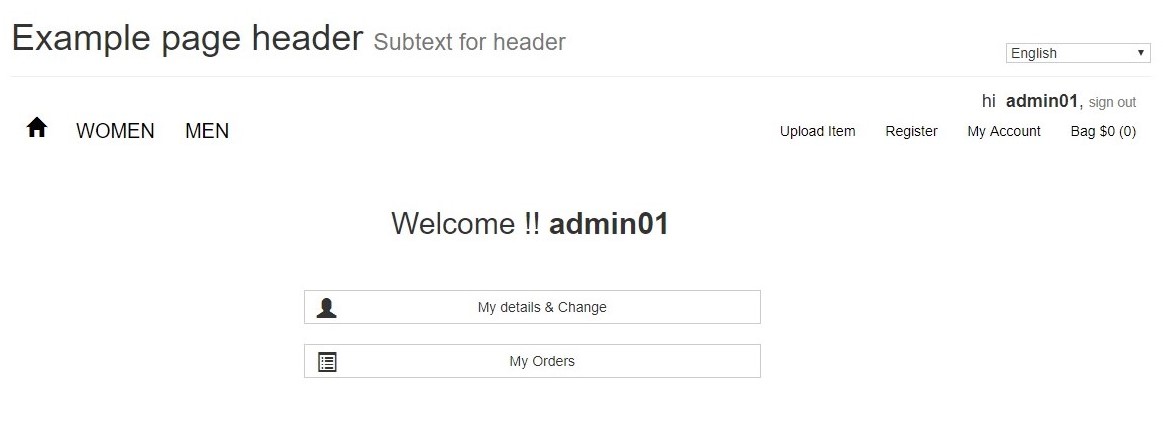


**login.jsp** [使用Shiro框架]

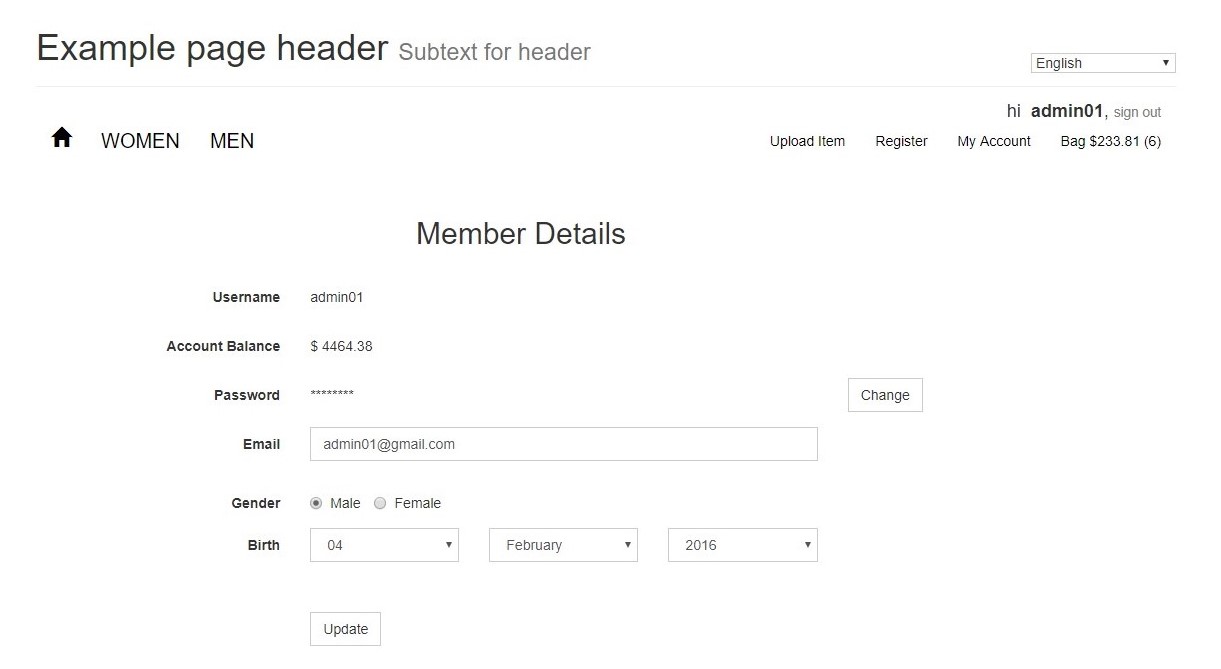
1.登入帳號、密碼校驗(前後端)

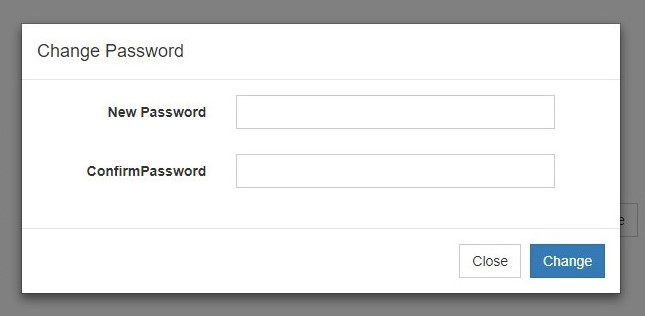
2.Remember me

3.當訪問了需認證的頁面，會轉發到login頁面；使用Shiro提供的記住上一頁的方法，以便在需要重新認證的頁認證成功後可以"直接"進入該頁面



**myAccount.jsp**





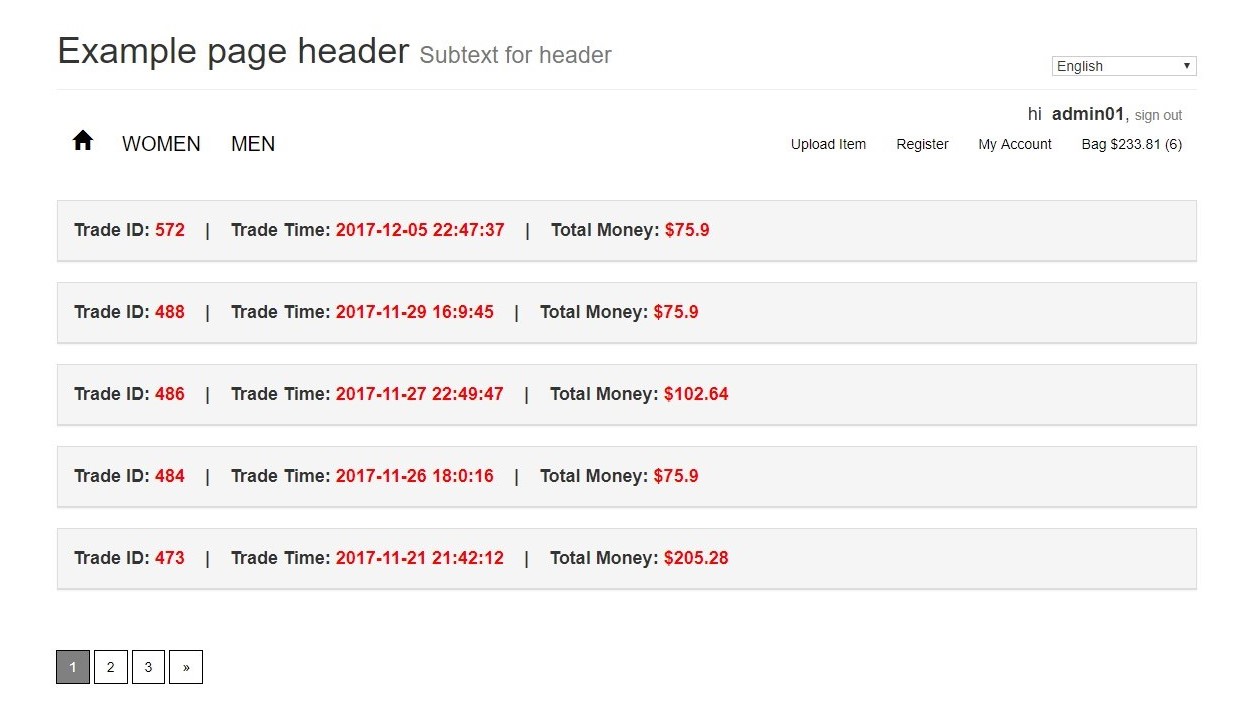
**member.jsp**

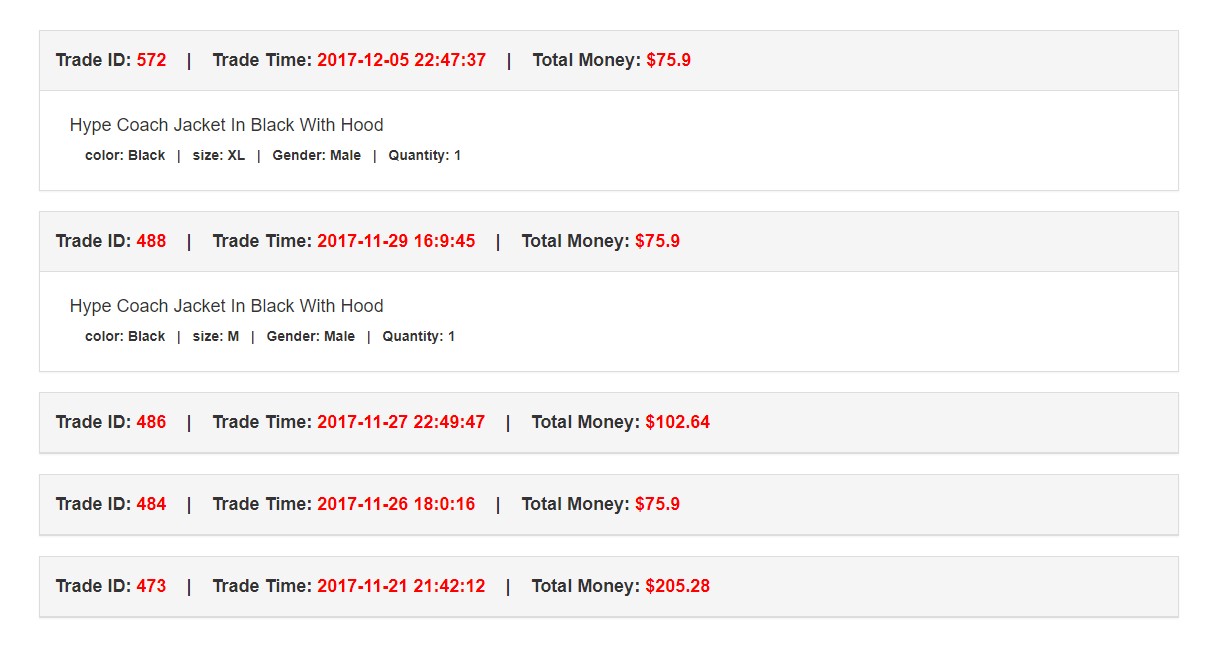
1.顯示用戶資訊及餘額

2.可修改用戶資訊

3.點擊Change可修改密碼 (使用Bootstrap的模態框效果)

4.表單使用前後端校驗

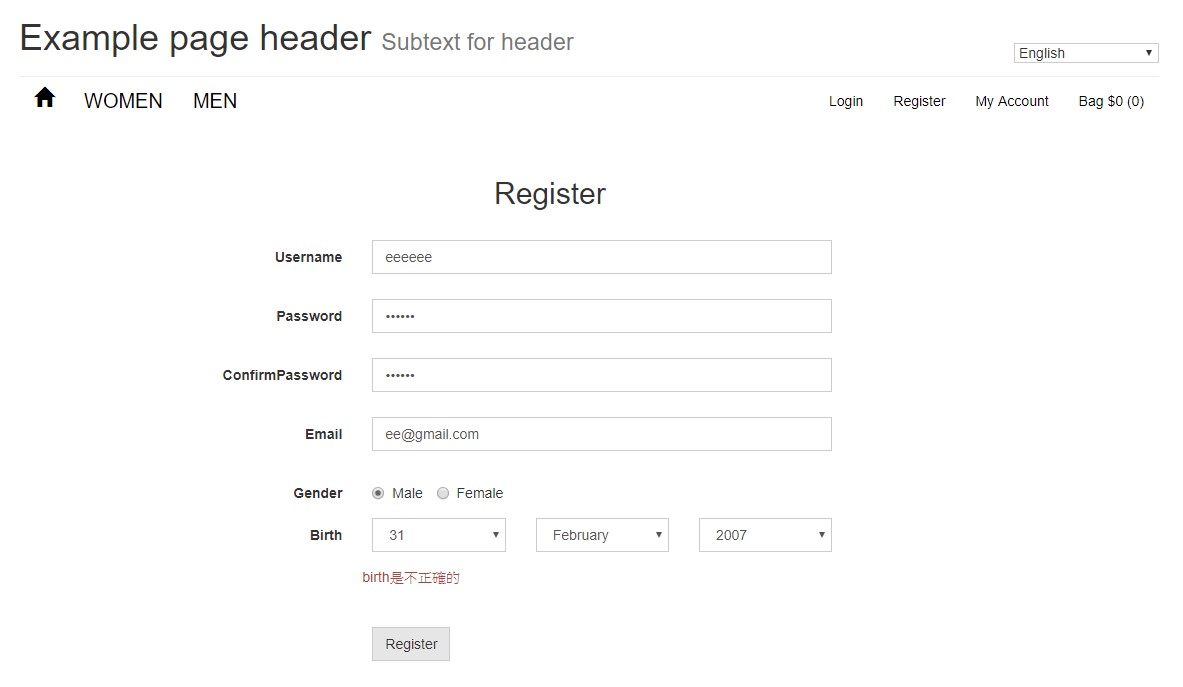




**myOrder.jsp**

1.用戶的交易紀錄

2.使用分頁





**register.jsp** [加密部分使用Shiro框架]

1.用戶註冊

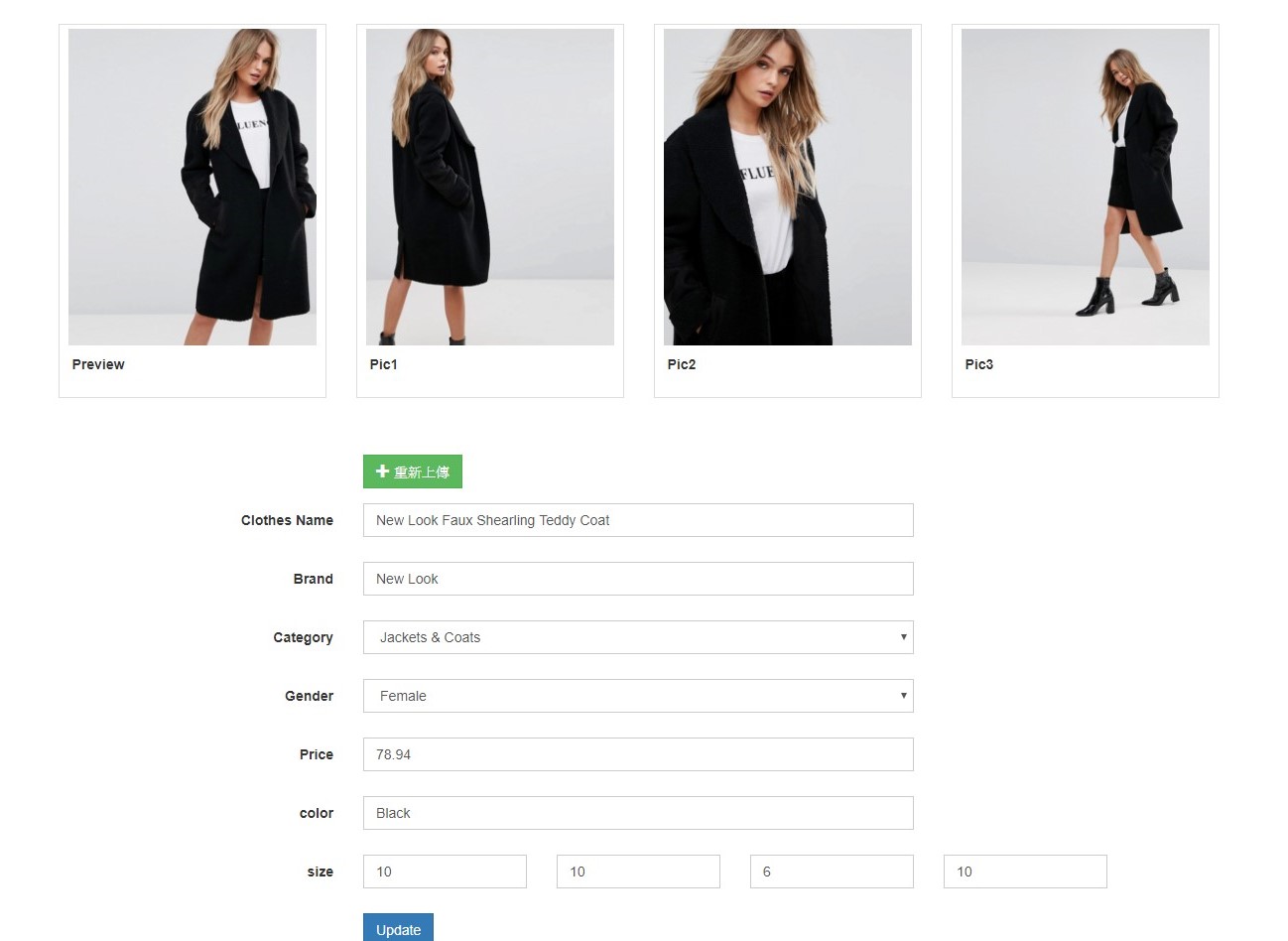
2.使用了前端校驗及後端校驗

3.當用戶名發生change會發送ajax請求，判斷該用戶名是否已存在

4.各個表單值校驗使用了springMVC的@Valid註解 (JSR303校驗)

5.若無錯誤則進行密碼加密(使用基於MD5演算法的自訂義加密工具)

6.註冊後賦予user角色，並創建一個accountBalance為該用戶的帳戶餘額



**uploadItem.jsp** [需有admin權限才能訪問，使用JQuery File Upload框架]

1. 建立或修改商品並上傳圖片
2. 各表單值使用前端校驗
3. 上傳檔案使用Spring MVC的CommonsMultipartResolver
4. 儲存圖片的方式：

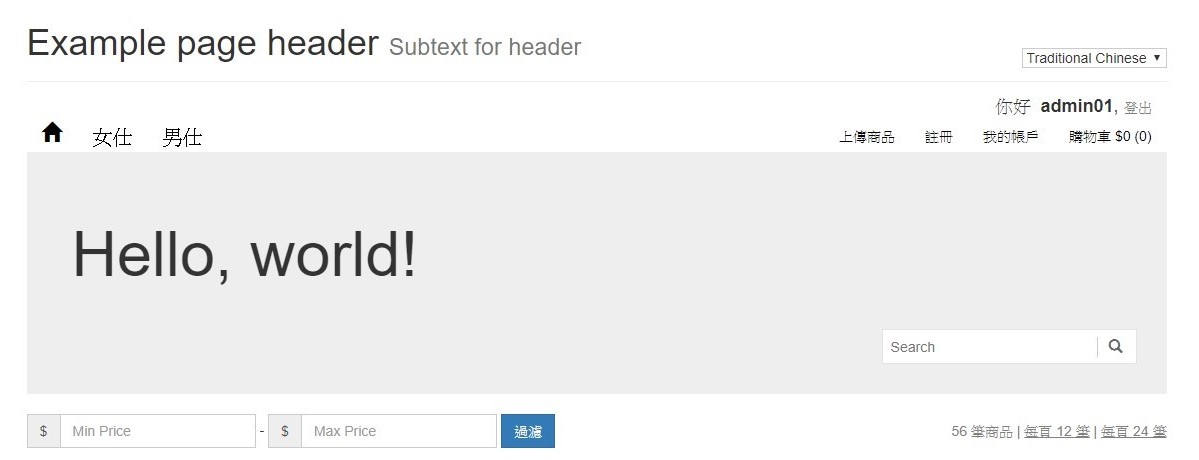
儲存在本機硬碟中，使用虛擬路徑的方式存取圖片

.在Tomcat sever中配置虛擬路徑

path:/pic ; Document Base: D:\uploadFiles\pic

.在需要顯示的圖片路徑前添加虛擬路徑

<img src='/pic/" + item.picture.pic1 + "'>



**國際化**

1.使用Spring MVC國際化功能

2. 在commons-title.jsp中添加帶有Locale參數並可切換的Select

使用fmt標籤做字串的國際化

**[補充]**

1.JSP頁面可使用spring標籤來做國際化 <spring:eval expression=""/>

2.後端部分可使用CurrencyStyleFormatter對數據做貨幣國際化

currencyStyleFormatter.print(Number number, Locale locale)

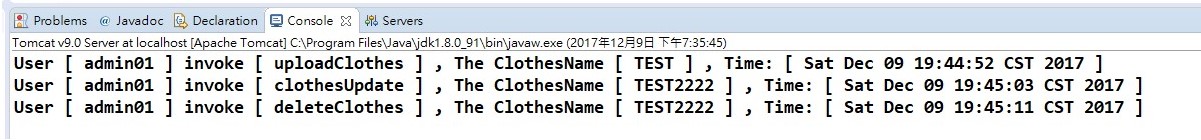
3.因專案是採用返回json數據的方式，目前還不知如何對json數據做貨幣、時間的國際化;

fmt、spring標籤僅能解析請求域中的數據

**[注意]**

若請求是以超連結轉發XXX.jsp的方式，Spring MVC的國際化功能會不起作用;

必須通過Controller才能獲取到Locale



**Spring AOP**

1. 在此專案中僅對Admin的三個操作做AOP

deleteClothes、uploadClothes、clothesUpdate

2. 在applicationContext.xml 或 springDispatcherServlet-servlet.xml配置

<aop:aspectj-autoproxy proxy-target-class="true"></aop:aspectj-autoproxy>