**購物網站**

**使用技術**

[後端]

Spring , Spring MVC , JPA/Spring Data , Shiro , Maven , TortoiseGit

[前端]

JQuery , Bootstrap , JQuery File Upload , HTML/CSS

[DB]

MySQL

[IDE]

Eclipse

**網站功能**

* + 商品頁面
  + 分頁
  + 商品分類
  + 搜尋商品
  + 金額過濾
  + 商品細節
  + 登入 [Shiro]
  + 註冊 [Shiro]
  + 會員資料、修改 [Shiro]
  + 購物車
  + 模擬交易
  + 交易紀錄
  + 新增商品、預覽圖片 [JQuery File Upload]
  + 修改、刪除商品
  + 各種前後端校驗
  + 文字國際化
  + admin操作的日誌紀錄 [Spring AOP]

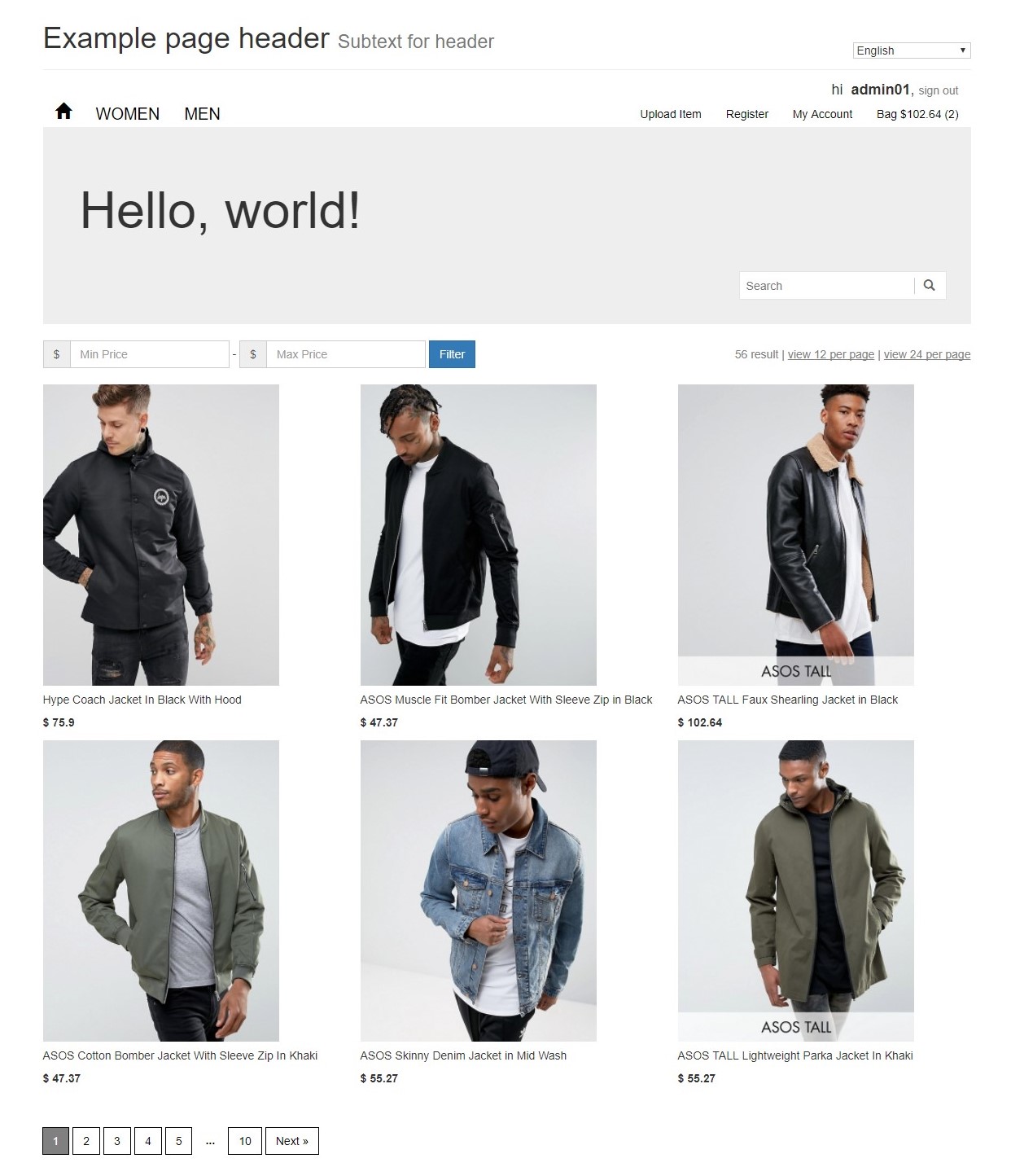
**傳送與接收數據**

為了相容於各平台 (網頁、android、IOS)，傳送請求的方式盡可能使用Ajax的方式，

數據均以JSON格式傳回，以方便各平台做解析。

瀏覽商品的分頁還是用放在請求域的方式呈現，以達到更好的用戶體驗。

(換頁時網址列能有變化，可以加到我的最愛)





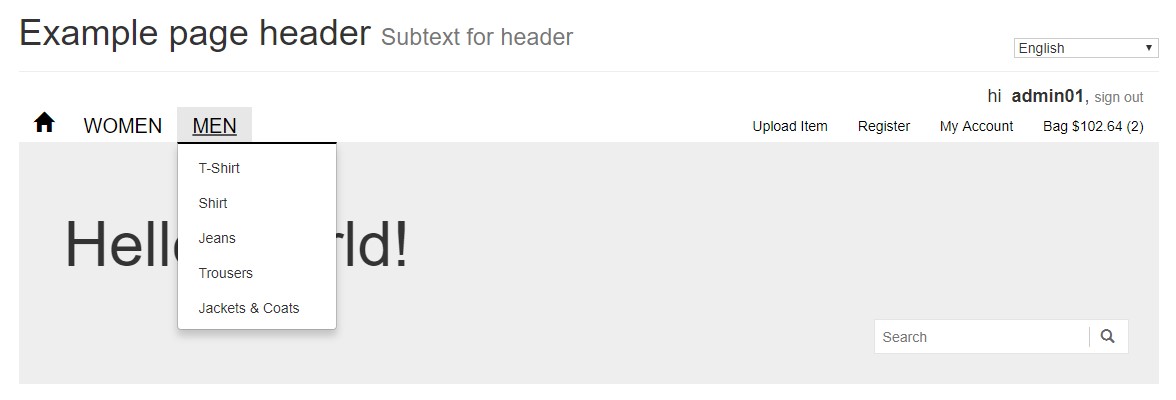
**shoppingPage.jsp**

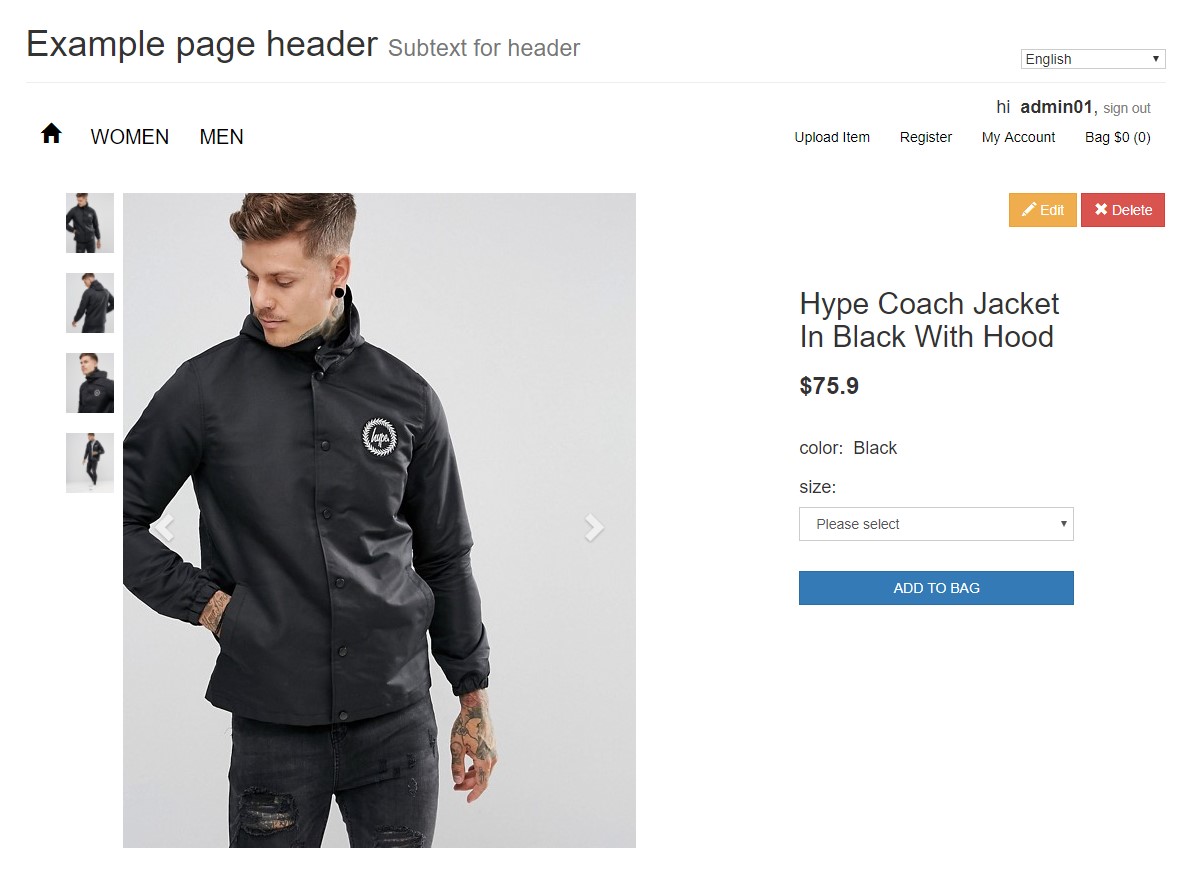
1.各項目連結

2.顯示購物車數量、金額

3.顯示商品並使用分頁

4.各種條件過濾連結、商品搜尋





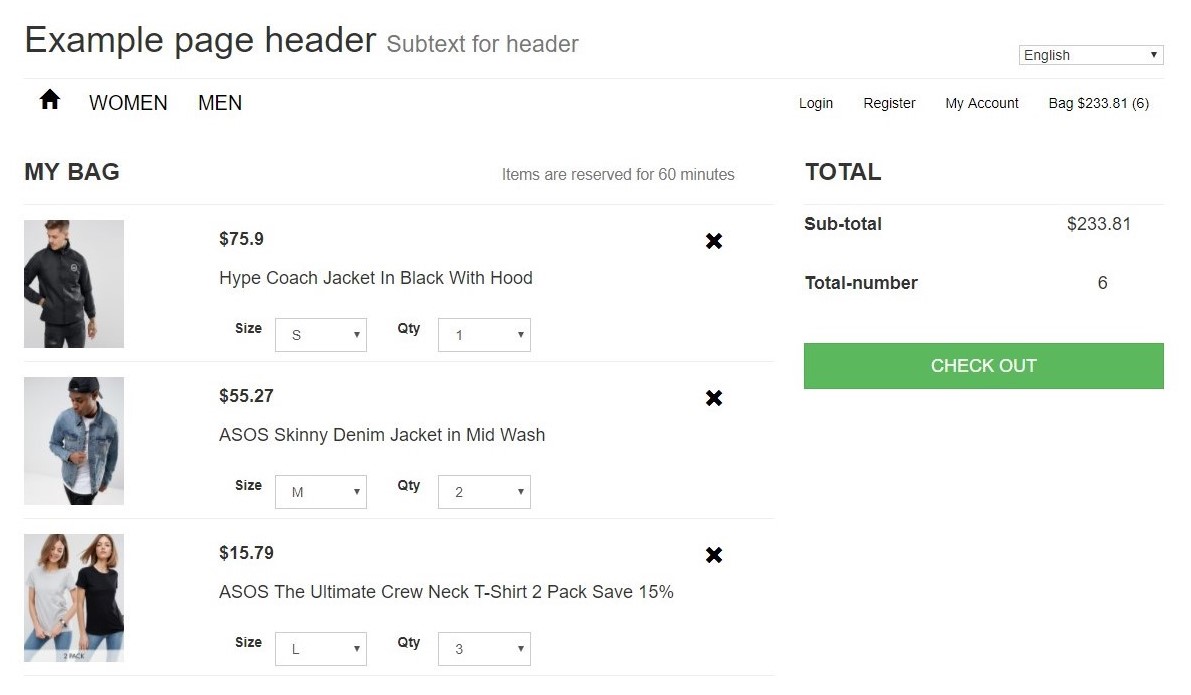
**clothesInfo.jsp**

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**

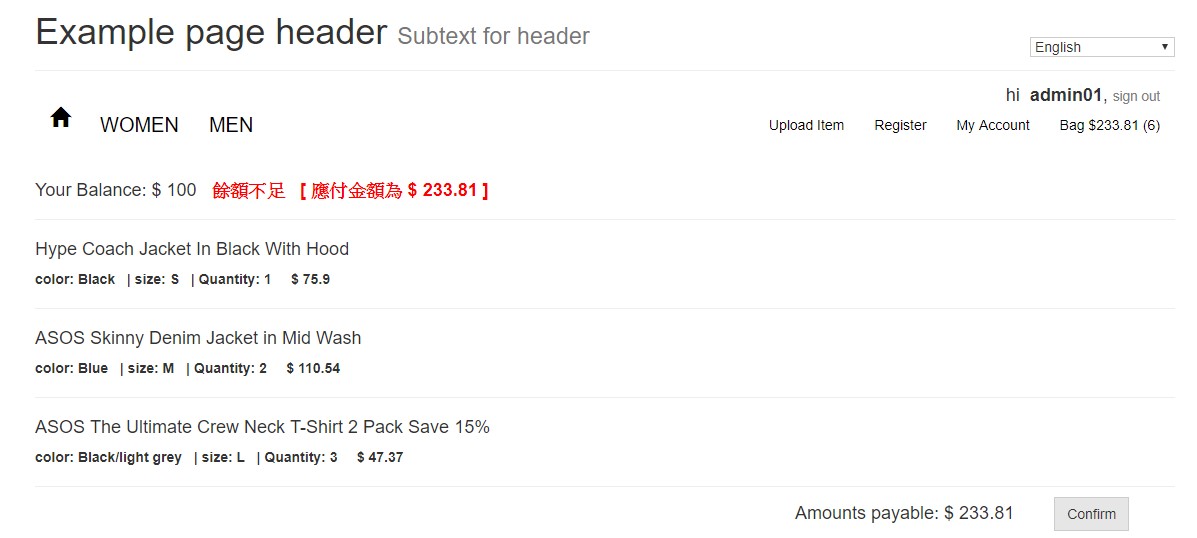
1.購物車中商品細項

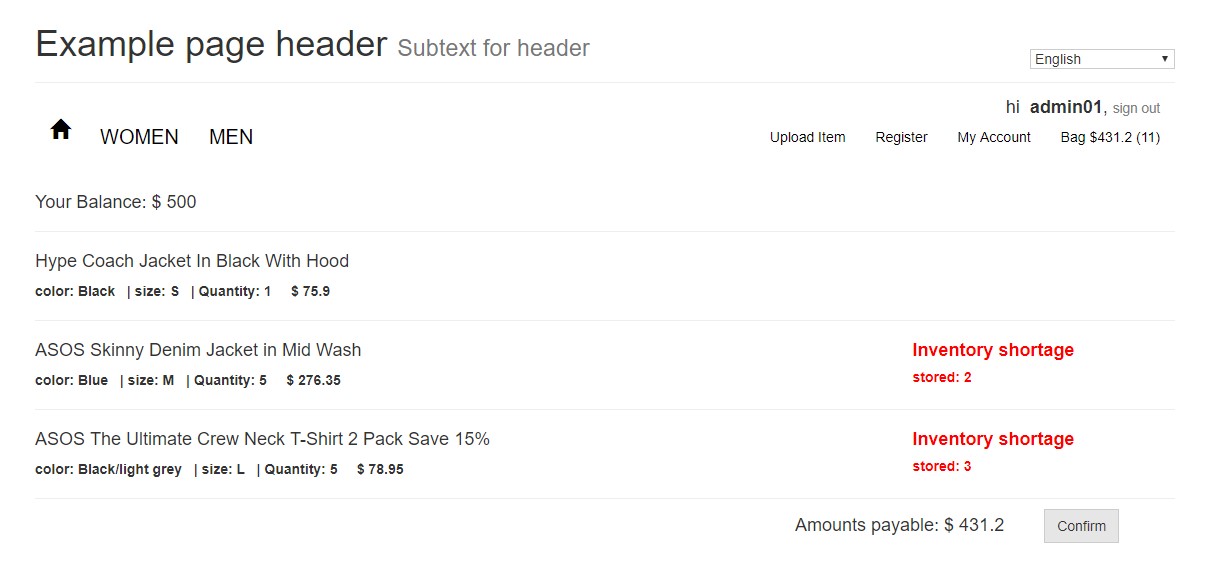
2.可修改商品數量

3.可修改商品size

4.移除購物車中商品 (JQuery動態效果)

5.checkout模擬結帳





**checkout.jsp** [需認證過後才能訪問]

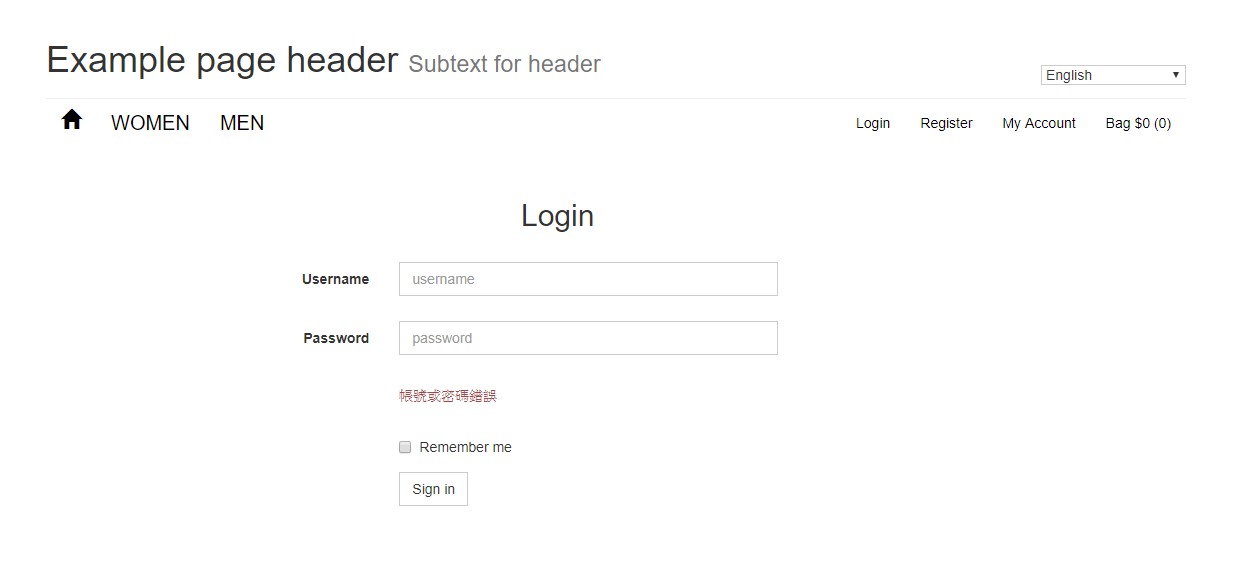
1.顯示用戶餘額

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



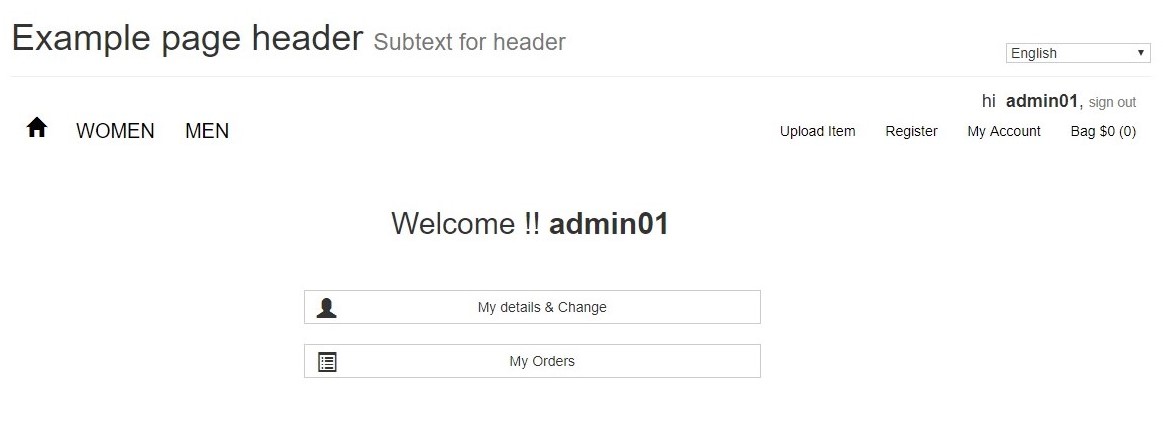
**login.jsp** [使用Shiro框架]

1.登入帳號、密碼校驗(後端)

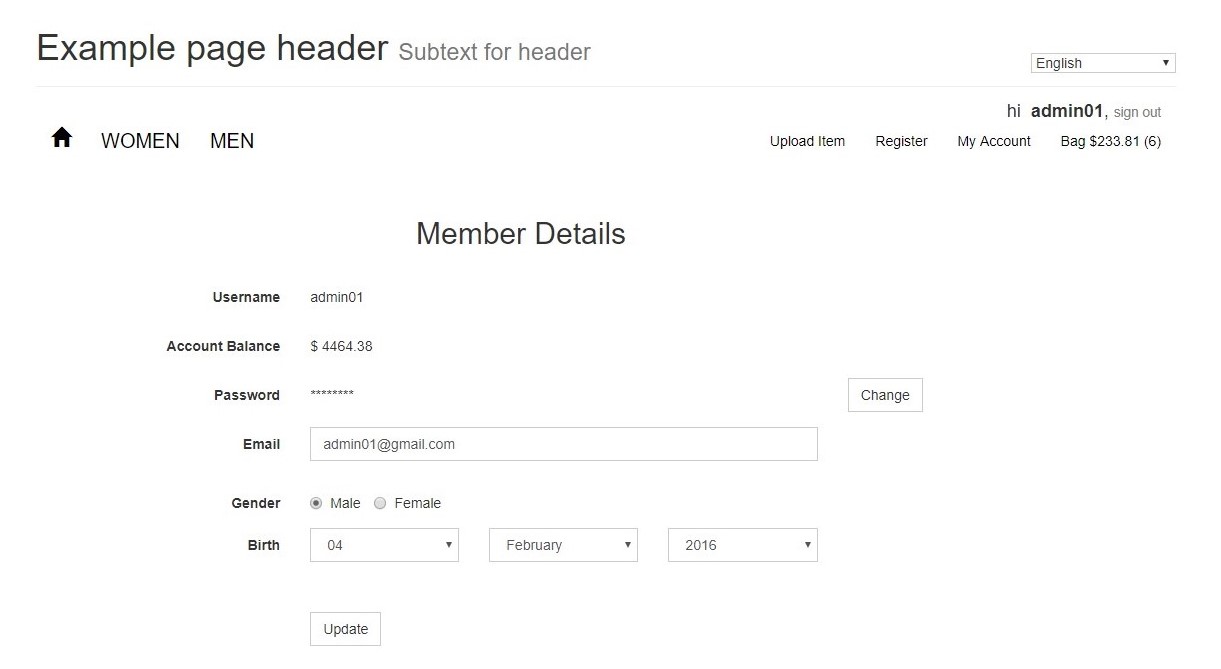
2.Remember me

3.當訪問了需認證的頁面，會轉發到login頁面

4.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面，認證成功後可以"直接"進入該頁面



**myAccount.jsp**



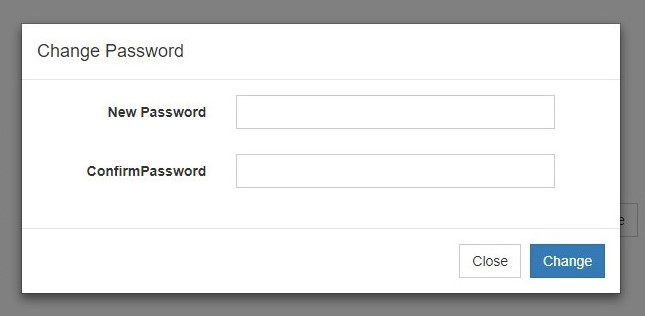
**member.jsp**

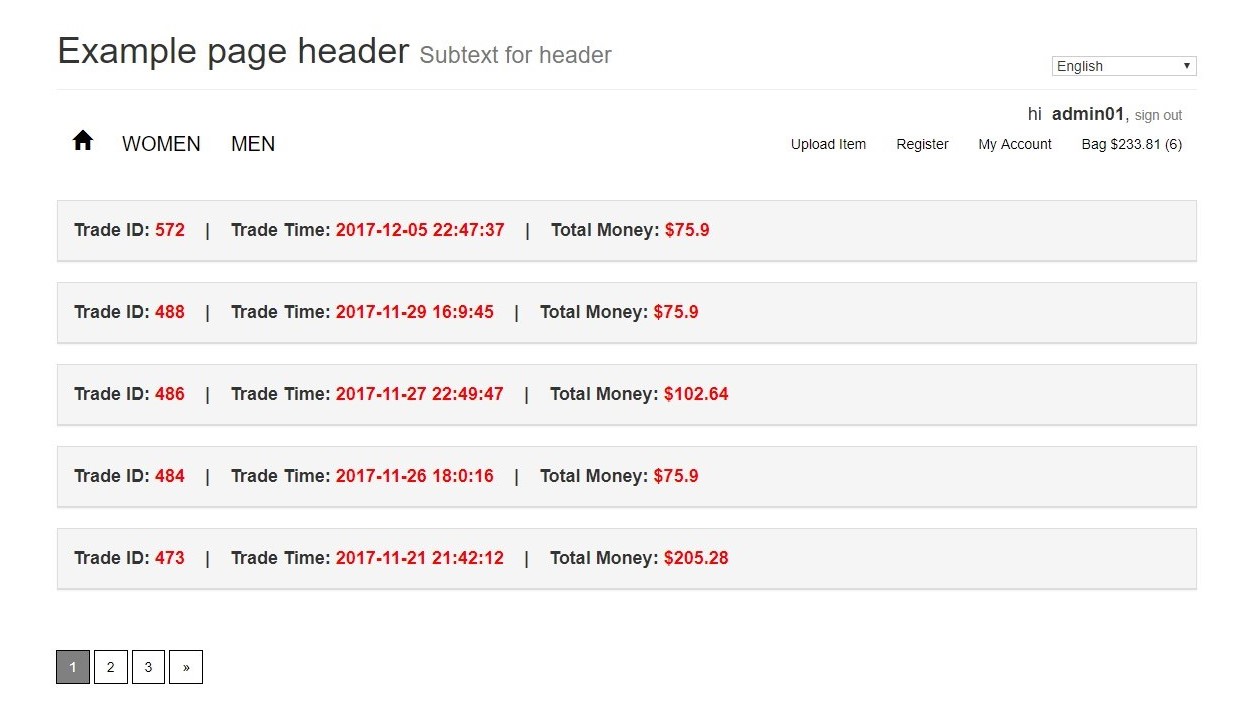
1.顯示用戶資訊及餘額

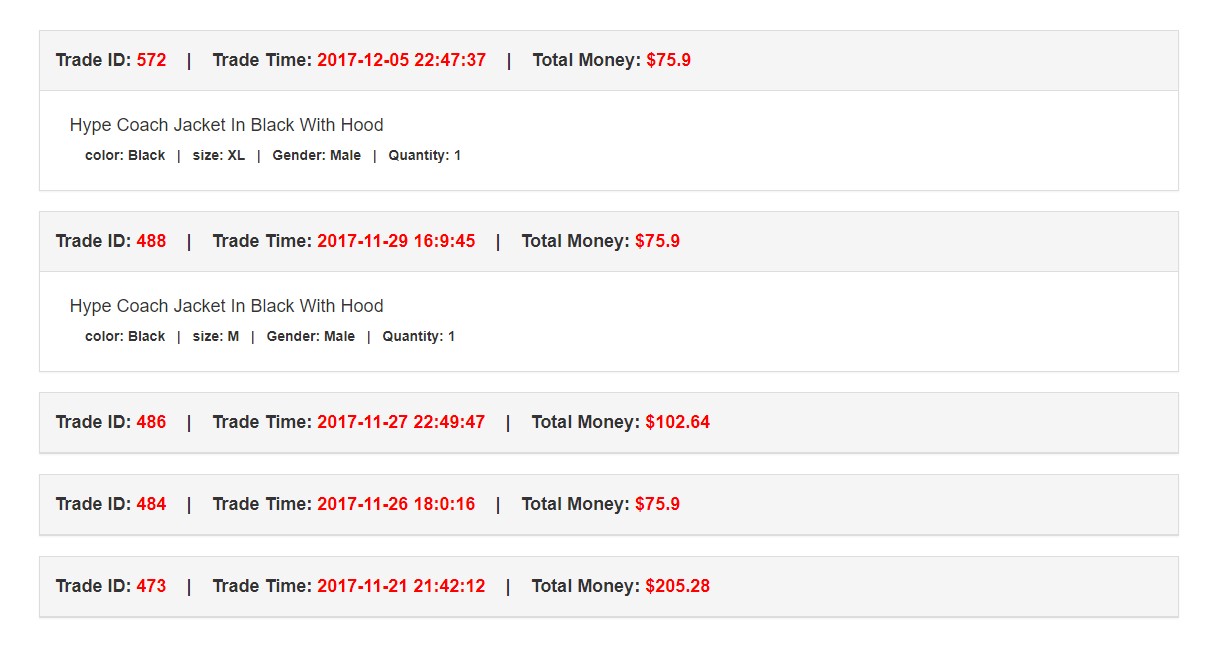
2.可修改用戶資訊

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





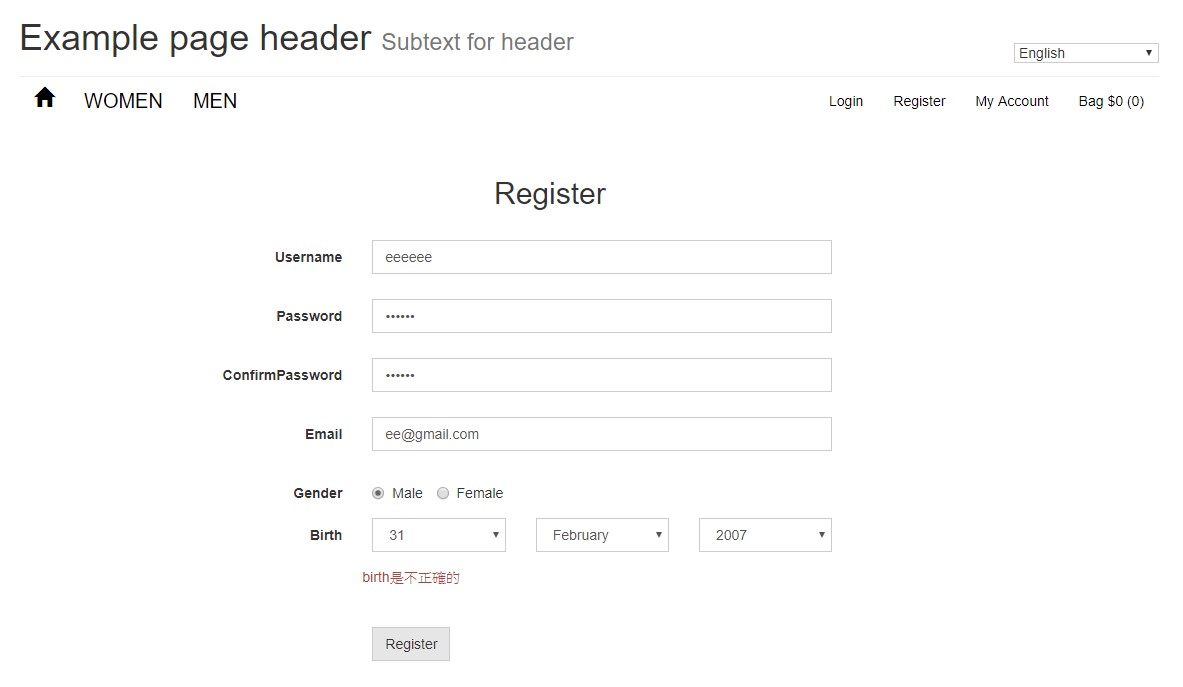


**myOrder.jsp**

1.用戶的交易紀錄

2.使用分頁

3.點擊標題可展開內容



**register.jsp** [加密部分使用Shiro框架]

1.用戶註冊

2.使用了前後端校驗

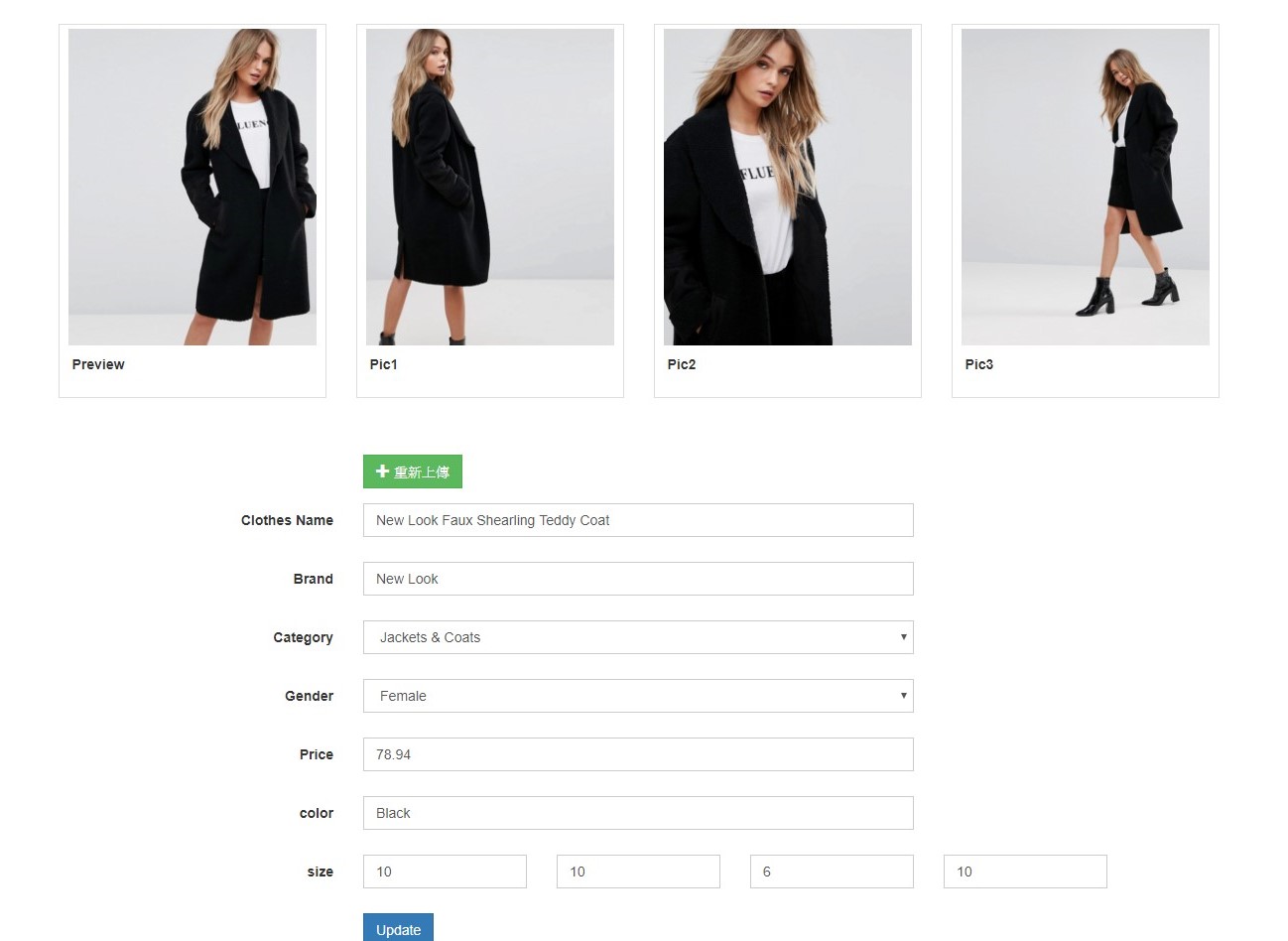
3.當用戶名發生change會發送ajax請求，判斷該用戶名是否已存在

4.各個表單值的後端校驗使用了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演算法的自定義加密工具)

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. JQuery File Upload 框架具有預覽圖片及傳輸條效果

2.建立或修改商品並上傳檔案

3.各表單值使用前端校驗

4.上傳檔案使用Spring MVC的CommonsMultipartResolver

5.儲存圖片的方式：

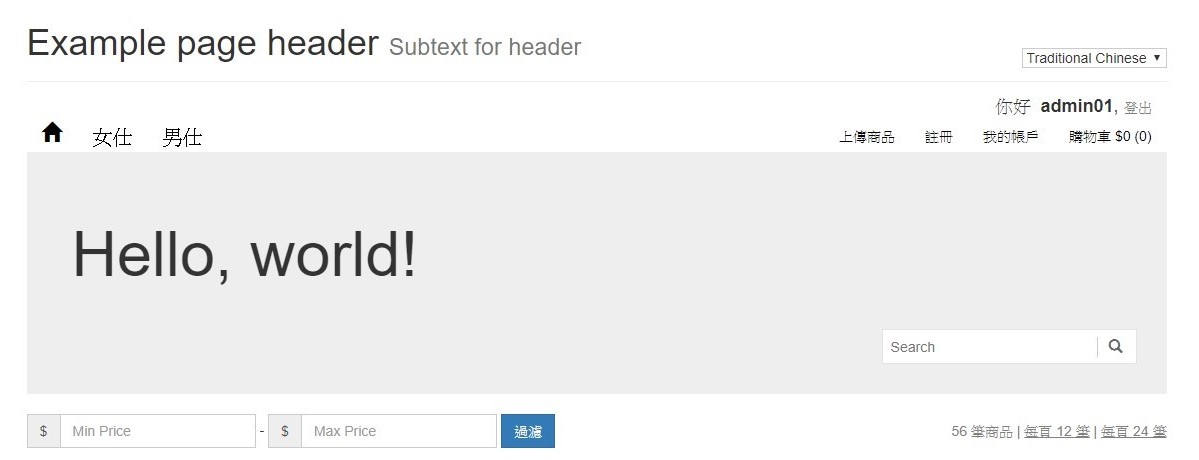
儲存在本機硬碟中，使用虛擬路徑的方式存取圖片

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)

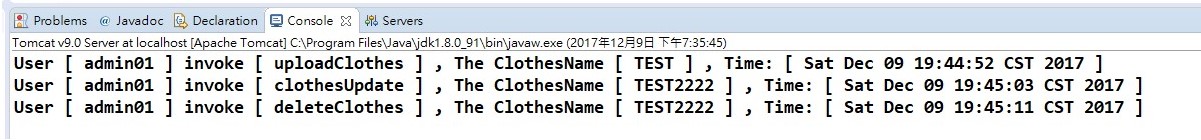
3.因專案是採用返回JSON數據的方式，目前還不知如何對JSON數據做貨幣、時間的國際化；

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日誌紀錄

deleteClothes、uploadClothes、clothesUpdate

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

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