

Práctica 3

Web

Pablo Martínez
Xavi Moreno

Extra Features

A parte de los puntos obligatorios, hemos añadido en nuestra web una funcionalidad extra que mejora la experiencia de usuario. Hemos creado un WebService que permite aumentar el saldo de un usuario. Para este WebService es necesario tener un usuario y contraseña, y saber que sueldo aumentar, con un mínimo de 5 € y un máximo de 3000 €. La respuesta del servidor la hacemos a partir de un JSON, creado con la librería JSON de Android. Devolvemos:

- status: Una lista que devuelve el servidor indicando la serie de errores que se han producido. En caso que todo haya ido bien status contendrá: OK
- saldoactual: Esto es el saldo actual del cliente, en caso que no haya habido ningún problema. Si en *status* no hay un OK, no devolverá este campo.

Lista de errores que puede devolver status:

- LIMIT_INFERIOR_SALDO: En cas que no es superés el límit inferior (5€) sortirà aquest missatge.
- LIMIT_SUPERIOR_SALDO: Això surt en cas que es passi l'usuari amb l'import màxim (3000€)
- INVALID_STRING_SALDO: Aquest missatge surt en cas que el valor introduït en saldo no sigui un enter. Només es pot augmentar el saldo amb xifres senceres.
- INVALID_USER_PASS: En cas que l'usuari no hagi indicat correctament l'usuari o la contrasenya, sortirà aquest missatge.

Estructura

La aplicación se basa en el **MVC** (Model-Vista-Controlador), el controlador i el modelo está programado en clases java, i la vista está implemetado en jsp.

Java source

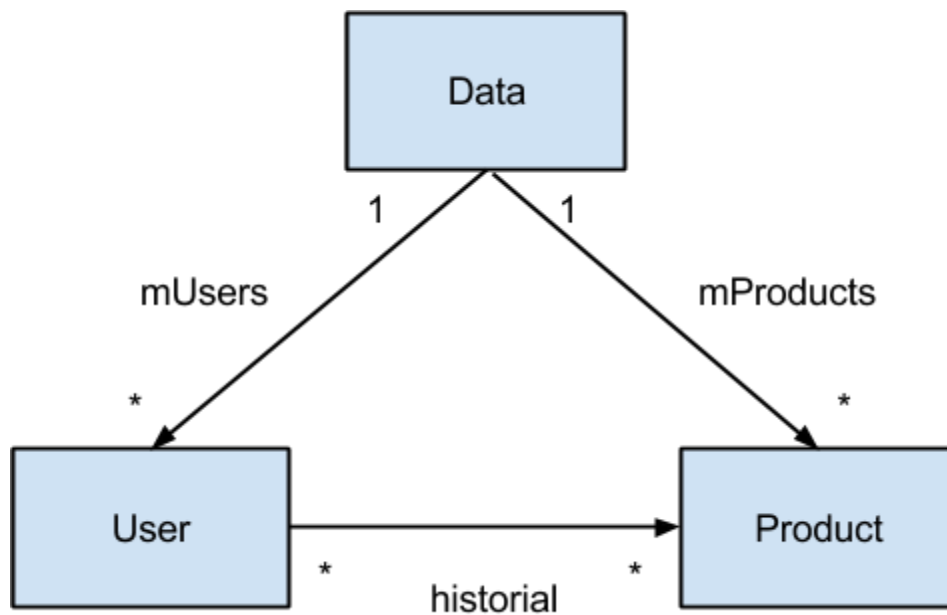
Controller

Como controlador tenemos una única clase: ServletDispatcher. Esta clase se encarga de recibir peticiones que llegan por el puerto configurado por el ServidorWeb. Se encarga entonces del control de la aplicación distribuida: redireccionar a los jsp que tocan, mapeo de urls, tener control de los datos de la aplicación, llevar a cabo los WebServices.

En todos los POST hemos implementado: POST-redirect-GET, por tal de evitar problemas d'envío doble de datos. A més, tota petició, passa pel ServletDispatcher

Model (Beans)

El diagrama de clases del modelo es el siguiente:



JSP

View

En la view encontramos código java con html, es decir, jsp. Tenim protegit les següents vistes:

- Cistell: Se muestra la cesta, donde el usuario ha ido añadiendo productos. Ésta se guarda en la sesión.
- Historial: Aquí se muestra el historial de compra del usuario. Esto debe ser privado, ya que desde aquí se puede descargar lo que el usuario ha comprado.

Hem deixat sense protegir llavors:

- Catálogo: Aquí se pueden ver todos los artículos que tiene la tienda.
- Inicio: Esta es la página de inicio, desde aquí puedes ir a cualquier parte a partir de la cabecera.

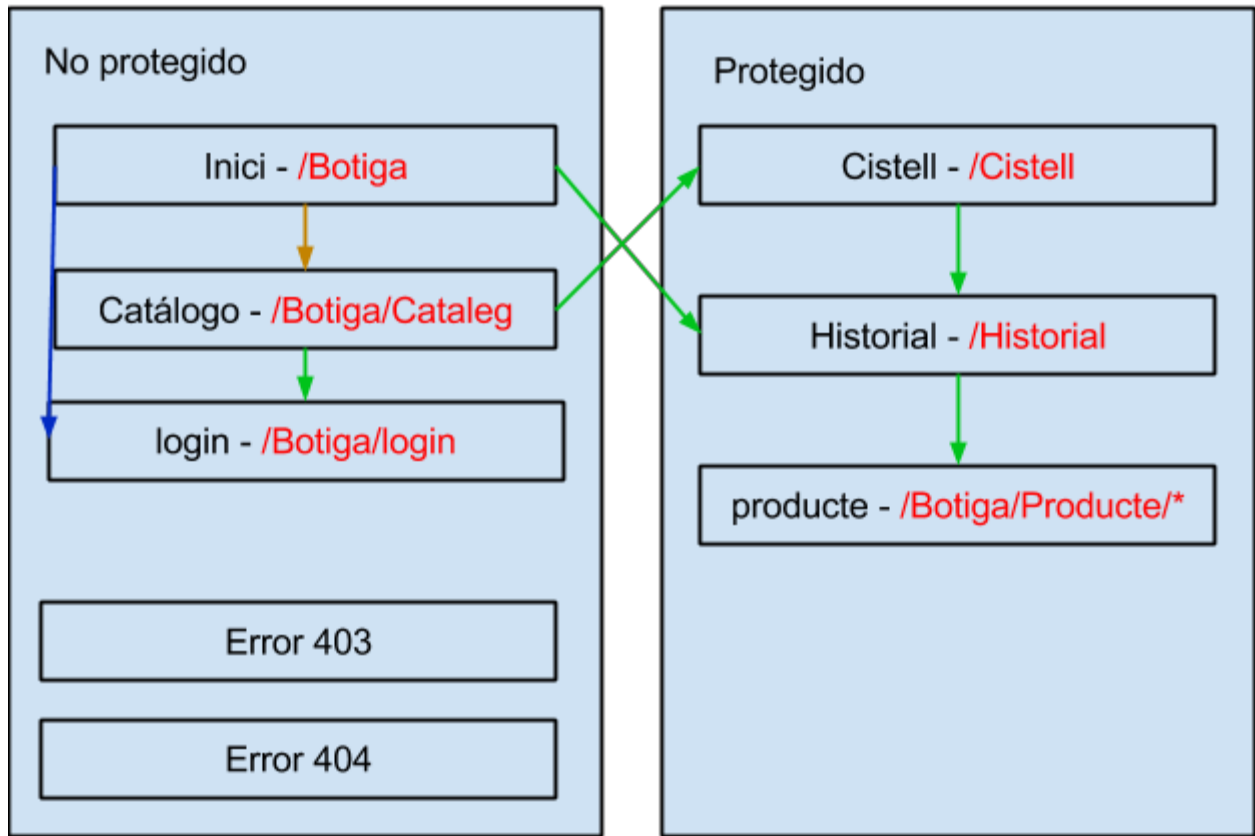
Conexión controller - view

Para hacer la comunicación desde el ServletDispatcher con la view, lo hacemos a través de los atributos del request. Con la ayuda de la librería jstl, podemos mostrar por pantalla eso que nos pasa el ServletDispatcher sin código java.

En caso de los bucles utilizamos <c:foreach> que permite hacer un bucle sobre una lista y poder mostrar cada contenido de la lista repetidas veces.

Mapa de la aplicación Web

Aquest es el mapa de la nostra pàgina web:



Leyenda (important)

Transiciones sin login

Transiciones con login

Transiciones mútuas

Aquí podemos ver las principales transiciones de nuestra página web. Aquí no se incluye donde puedes ir a partir de la cabecera, ya que puedes ir a cualquier lado, y varía dependiendo si estás dentro del sistema o no.

Sincronización

La sincronización la hacemos únicamente en dos sitios:

- Escribir en el fichero `users.json`: Aquí hemos sincronizado con un objeto (**mutex**) que comparten todos los usuarios que se conectan. Por tanto para cada sobreescritura que hacemos al fichero nos aseguramos que **un solo hilo esté escribiendo en él**. Esto no hace falta hacerlo en el fichero de `products.json`, porque ese fichero es sólo de lectura y no hay que escribir en él.
- Modificar datos de un usuario: Cada vez que se modifica los datos de un usuario, hay que sincronizarlo debido a que **el mismo usuario puede estar conectado al mismo tiempo en otra máquina**. Aunque tengan diferente sesión, pueden tener referenciado el mismo usuario y por tanto si cambiamos datos, se debería sincronizar, para que primero haga uno y luego otro.

La aplicación no es muy compleja, debido que la sincronización debe en recursos compartidos que hay entre usuarios, no hay mucho código que sincronizar. Por ejemplo, cosas como la **cesta no hace falta sincronizarlo, porque es propio de cada sesión**, y una sesión no afecta a otra.

Test

Para el día del test llevábamos toda la práctica realizada con la versión completa implementada, incluso con la implementación del JSON, aunque no la probamos. El servidor aguantó las dos horas, excepto que se me reinició el ordenador debido a un problema que tuve haciendo testing de otras página.

De la gente que hemos podido reunir, y nos ha dicho que tal está la página web, no nos han dicho nada problemática, excepto que si ya hay comprado un producto, y intentas comprarlo de nuevo, no se te llena la cesta. La funcionalidad es correcta, però no damos un mensaje al usuario por tal de que se entere de ese hecho.

Aquí dejamos el test que hicimos a los demás usuarios:

Grup 1:

No deixa comprar. Quan fas click al botó "Accepta la compra" retorna un 500.

He fet la prova canviant d'usuari i si que funciona.

Tot correcte

Grup 2:

Tot funciona correctament.

Grup 3:

OFF

Grup 4:

OFF

Grup 5:

Disseny: Per comprar has de sortir del catàleg manualment i anar a cistell. (Volver al índice + tu cesta)

500: Error al accedir a la pagina downloads si no estas logat

Grup 6:

Disseny: Poc usable.

Pots comprar el mateix producte varies vegades

Grups:	7	8	9
Logar-se amb usuari existent.	V	V	F, he d'entrar en cistell per logar-me
Logar-se amb usuari no existent.	V	V	V
Es disposa d'un saldo inicial?	V	F, no he vist el meu saldo	V, però no es visible en tot moment
Afegir un ítem al cistell.	V	V	V
Afegir un altre ítem d'un altre tipus al cistell.	V	V	V
Sortir del navegador	V	V	V
Entrar al navegador. Logar-se amb el mateix usuari	V	V	V
Veure cistell.	V, però es guarda el cistell en la sessió	No puc accedir directament	V, però es guarda el cistell en la sessió
Comprar ítems	V	V	V
Veure Històric/compres fetes.	V	V	V
Descarregar-se un fitxer dels comprats	V	V, però m'han trolat	V
Copiar link de descarrega / deslogar-se / accedir al link	No es pot copiar el link	V	V
Copiar link de descarrega /logar-se amb un altre usuari/ accedir al link	-	V, No m'el descarrega, però no ho controlen	V, no em deixa descarrega
Esborrar cookies, deshabilitar-les en el navegador i tornar a provar.	-	No funciona	No funciona

Grups:	10	11	12
Logar-se amb usuari existent.	V	Està OFF	F, però em deixa registrar
Logar-se amb usuari no existent.	V	V	
Es disposa d'un saldo inicial?	V	-	
Afegir un ítem al cistell.	V	-	
Afegir un altre ítem d'un altre tipus al cistell.	V	-	
Sortir del navegador	V	V	
Entrar al navegador. Logar-se amb el mateix usuari	V	-	
Veure cistell.	V	-	
Comprar ítems	V	-	
Veure Històric/compres fetes.	V	-	
Descarregar-se un fitxer dels comprats	V	-	
Copiar link de descarrega / deslogar-se / accedir al link	V	-	
Copiar link de descarrega /logar-se amb un altre usuari/ accedir al link	V	-	
Esborrar cookies, deshabilitar-les en el navegador i tornar a provar.	No funciona	No funciona	