

SD | Pràctica 3.

Llibreria de Recursos Electrònics Online.

Objectius de la pràctica

Es vol que l'alumne aprengui a fer ús de les tecnologies de desenvolupament web en la plataforma Java.

Què cal fer?

Es desitja fer una llibreria Online de recursos digitals, que permeti als usuaris baixar-se llibres, cançons, vídeos curts.. en qualsevol tipus de format mp3, pdf, avi... si tenen prou crèdit per adquirir-les.

Tecnologies

Els requeriments d'aquest sistema fan ús de les següents tecnologies: La implementació ha de seguir el patró de programació de Model-Vista-Controlador. La programació ha de ser en llenguatge Java usant Servlets i JSP. Ús de Tomcat com a contenidor de servlets. L'autenticació usará l'especificació JASS. Implementació d'un Servei Web RESTful usant servlets i JSON.

DISSENY APP WEB

- L'aplicació web estarà formada per un catàleg de diferents recursos digitals:
 - llibres
 - cançons
 - vídeos curts.
- El catàleg està format per una sola pàgina per a cada tipus de recurs digital.
- Caldrà implementar un sistema d'autenticació (tal i com es comenta en l'apartat següent).
- Caldrà implementar una “Llista de descàrregues” per a que l'usuari de la llibreria pugui anar-hi ficant tots els recursos que vulgui baixar-se.
- Un cop hagi seleccionat tot el que vulgui, podrà descarregar-se els recursos des de la pàgina de descàrregues, mentre li quedi crèdit.
- El crèdit inicial de cada usuari estarà fixat per l'aplicació web.
- El crèdit i els recursos electrònics adquirits han de ser persistents, és a dir, s'han de guardar en disc i recuperar els seus valors cada cop que s'apagui i s'engegui el servidor.

Mapa Web

- Hi haurà les següents pàgines web:

- pàgina **pública** de benvinguda en la que es presenta la llibreria Online i s'ofereix un enllaç per anar a la pàgina de descàrregues.
- pàgina **pública** d'identificació d'usuari amb una paraula de pas.
- Una sèrie de pàgines **públiques** per cada tipus de recurs electrònic del catàleg.
- pàgina **protegida** amb la llista de descàrregues per a poder baixar-se els recursos seleccionats.

Més concretament el sistema hauria de constar de:

http://localhost:8080/llibreria

Descripció: pàgina pública de benvinguda al sistema.

Contingut:

- Missatge de benvinguda

Disseny: Lliure

Enllaços:

- "Entrar a la llista de descàrregues" -> enllaça a la pàgina protegida llibreria/protegit/llista
 - "Entrar al catàleg" -> enllaça a la pàgina pública llibreria/catalog
-

http://localhost:8080/llibreria/catalog

Descripció: Una o varies pàgines per tipus de recurs electrònic amb varis recursos per pàgina.

Contingut:

- Descripció reduïda de cada recurs electrònic, imatge petita, nom, format, autor, editor... El contingut ha de ser mostrat ordenat segons nom del recurs. Opcionalment, es pot fer una pàgina per cada recurs amb una descripció més extensa.
- En tot moment s'ha de saber en quin lloc del catàleg ens trobem, si estem identificats o no i en cas afirmatiu el saldo que disposem.
- Opcionalment:
 - Checkboxes per afegir el recursos marcats a la llista de descàrregues.
 - Botó per cada producte per a afegir a la llista de descàrregues i comprar-lo. (one-click purchase)

Disseny: Lliure

- Podeu fer una pàgina catàleg principal amb un resum de productes de cada tipus.

Enllaços:

- Per tornar a l'inici /llibreria
 - Per anar a /llibreria/privat/llista.
 - Per anar al llistat de cada un dels tipus de productes.
-

http://localhost:8080/llibreria/protegit/llista

Descripció: En aquesta pàgina es veuran tots els recursos seleccionats i/o comprats, cadascú amb un enllaç que permeti la descàrrega directe del recurs. En cas que encara no s'hagi comprat, l'enllaç no es mostrarà i haurà de fer-se mitjançant un botó de compra. Si en algun moment el saldo del client no és suficient per a la compra d'un recurs el link no es mostrarà.

Contingut:

- Recursos afegits a la llista de descàrrega per a ser adquirits.
- Recursos adquirits
- Un botó per a cada ítem per tal d'adquirir recursos que no s'hagin comprat encara.
- Un cop feta la compra es retornarà a la mateixa pàgina, però apareixerà un dels següents missatges:
 - Missatges de compra realitzada amb èxit i saldo restant.
 - Missatges d'error si n'hi ha. (Sense saldo, recurs no disponible al servidor...).
- Usuari logat i saldo disponible.

Disseny: Lliure

Enllaços:

- Existirà un enllaç amb el nom "Sortir" que anirà a /llibreria tancant la sessió.
 - Existirà un enllaç per seguir comprant que anirà a /llibreria/catalog
 - Existirà un enllaç per cada recurs que s'hagi comprat mitjançant el qual es podrà baixar el recurs.
-

Passos pel desenvolupament de l'aplicació:

Pas 1: Disseny de l'aplicació web

Feu el disseny de classes de l'aplicació web en **PAPER**

- Seguiu la nomenclatura que s'us dona.
- El controlador és **un únic** servlet dispatcher, el model ha d'incloure tota la gestió de dades a model i la vista han de ser pàgines JSP.
- Feu el diagrama UML de les classes que usaràs identificant el Model, la Vista i el Controlador.
- Feu el disseny de [wireframes](#) de les diferents pàgines web.

Pas 2: Implementació de l'estructura de pàgines demanada

Implementeu l'estructura de pàgines demanada a la descripció del sistema. Heu d'usar l'aproximació MVC Servlet+Beans+JSP.

- Un sol Servlet per a comandar el procés, gestionar les dades enviades per l'usuari, i calcular-ne el resultat. En el fitxer web.xml només ha d'haver-hi un sol tag
- N classes Bean per guardar les dades que s'intercanvien Servlet i JSP.
- N pàgines JSP que el servlet invocarà per a mostrar els resultats a l'usuari.
- El del servlet dispatcher ha de poder capturar totes les URLs de l'aplicació web. Per tal que els recursos estàtics (imatges, css, etc) es puguin carregar sense problemes cal afegir, a part del mapeig del servlet, un mapeig explícit:

```
<servlet-mapping>
  <servlet-name>default</servlet-name>
  <url-pattern>/static/*</url-pattern>
</servlet-mapping>
```

- Estructura de carpetes i recursos recomenada:

```
/index.jsp
/static/img/
/static/css/
...
/WEB-INF/classes/*.class
/WEB-INF/jsp/*.jsp
/WEB-INF/resources/...
```

Pas 3: Implementació de l'autenticació

Baixeu-vos el fitxer [ExampleAuthentication.tar.gz](#)

Descomprimeix-lo i instal·la'l al Tomcat (afegeix la carpeta ExampleAuthentication al directori tomcat/webapps).

Observa com implementa l'especificació JAAS de J2EE (fes una ullada al fitxer web.xml per a veure com defineix la zona segura, quina és la pàgina d'entrada de la clau de pas.

Els noms d'usuari i les claus les heu de posar vosaltres al fitxer tomcat/conf/tomcat-users.xml mireu un exemple a la documentació de [MemoryRealm](#).

Per la nostra aplicació web, els usuaris que es connectin seran definits amb el rol de **Client**.

Les pàgines privades de la nostra aplicació web estaran protegides mitjançant aquest sistema d'autenticació basat en formularis seguint l'estàndard JASS.

Fixeu-vos que mai s'enllaça el jsp de login, sino que en anar a una pàgina protegida (definida al web.xml) "salta" el formulari de login i et permet continuar o et redirecciona a una pàgina d'error.

Pas 4: Implementació de la lògica del servidor.

- Per a l'estructura en general del Servlet feu servir el següent [esquema](#) (Agafeu les parts necessàries. No heu de fer Copy & Paste i posar-ho tal qual en el vostre codi!)
- Tingueu el catàleg guardat com a classes en el model.

- Podeu llegir el model en carregar-se el servlet per primer cop (mètode **init()**) des d'un fitxer o podeu crear els objectes.
- Deseu el saldo i les descàrregues adquirides per cada usuari a fitxer quan s'aturi el servidor (mètode **destroy()**). Guardeu-ho com a fitxers .txt en la part privada de la web (/WEB-INF/users_data/*.txt).
- Llegiu el saldo i les descàrregues adquirides per un usuari quan aquest es logui per primer cop.
- **Totes** les crides han de passar pel ServletDispatcher.
- **Tots** els mètodes GET han d'acabar amb un forward a una JSP.
- Heu de mantenir la sessió de l'usuari un cop logat. Podeu saber l'usuari a partir del request un cop s'ha logat amb `request.getRemoteUser()`. A partir d'aquí heu de crear una sessió i mantenir-hi el saldo i la llista de descàrregues actualitzada.
- Recordeu que la web és **Multi-Thread**, cada crida al `doGet` obre un nou thread. Les col·leccions de JAVA no són **safe-thread**, per tant haureu de fer servir les col·leccions del paquet `java.util.concurrent` si aquestes són manipulades com a conseqüència d'una crida al `doGet`.
- Feu servir el patró de disseny **Post/Redirect/Get** quan feu servir mètodes POST (i.e. quan enviu el formulari per comprar o afegir a la llista.)
- El fitxer del recurs electrònic, guardeu-lo dins del context de l'aplicació web i dins de la carpeta **/WEB-INF**. Això farà que el recurs no serà accessible des de fora. A l'hora de presentar-lo ho heu de fer com a un link indirecte, és a dir, un link que passarà pel servletDispatcher que servirà el fitxer del recurs.
- Per a que us pogueu descarregar el fitxer que teniu en el servidor (file), heu de canviar el `ContentType` de la response a "application/octet-stream" i enviar el fitxer a troços:

```
ServletOutputStream outStream = response.getOutputStream();
response.setContentLength((int) file.length());
response.setHeader("Content-Disposition", "attachment; filename=\"" + file.getName() + ".txt");

byte[] byteBuffer = new byte[1024];
DataInputStream in = new DataInputStream(new FileInputStream(file));
while ((length = in.read(byteBuffer)) != -1) {
    outStream.write(byteBuffer, 0, length);
}
in.close();
outStream.close();
```

AFEGIR SERVEI WEB

Es vol proporcionar un servei WEB RESTful que permeti consultar el nostre catàleg. Disposarà de dues operacions:

- GET /API/(AUDIO|VIDEO|BOOK)/catàleg
 - retornarà el catàleg sencer d'audio, video, o book, concretament retornarà per cada un d'ells el nom (camp NAME) i una breu descripció (camp DESC).
- GET /API/(AUDIO|VIDEO|BOOK)/item/NomItem
 - retornarà la descripció de l'item, concretament ens interessarà un camp anomenat PRICE que contingui el preu i un altre LINK que contingui el link directe al recurs.

Donat que se'ns proporcionen aquests serveis, es vol implementar una pàgina web que ens busqui pel nom d'un producte en concret quines llibreries online el tenen i ordenar-les per preu. Es permet buscar per: tipus de recurs o en general, i pel camp nom o tant pel camp nom com descripció.

Pas 1

Afegiu al vostre WEBAPP un nou servlet anomenat *WebServiceServlet* que oferirà els serveis web. Els resultats els ha de retornar com a JSON (utilitzeu la llibreria GSON per passar d'un objecte java a un fitxer JSON.) amb l'esquema que es comenta a dalt.

Recordeu a canviar el mimetype de la resposta:

```
response.setContentType("application/json");
```

Per tal de poder fer bé el pas dos afegiu també la següent capçalera al response: `response.setHeader("Access-Control-Allow-origin", "*")`

Pas 2

Creu una pàgina WEB JSP accessible públicament en el vostre WEBAPP (Llibreria/consulta) que consulti en totes les llibreries el nom d'un ítem que l'hi entreu. A continuació, us presenti els ítems coincidents en quines llibreries estan i a quin preu, ordenats per preu.

Les IP's de les llibreries s'hauran de llegir del fitxer: static/llibreries.txt.

Creu el codi JQUERY necessari tal que en prémer el botó de cerca s'executi una funció javascript que faci les diferents crides als webservices, mitjançant JQUERY.AJAX.

Un cop recollides **TOTES** les respostes, s'hauran de mostrar els resultats ordenats per preu.

Tingueu en compte que AJAX és asíncron, per tant per a esperar-vos a tenir tots els resultats abans d'ordenar haureu de fer servir la funció [when de JQUERY](#)

Planificació

Sessió 1, 16/04

- Disseny de les pàgines web
- Fer esquelet de directori
- Desplegar Web.xml
- Implementar esquelet ServletDispatcher
- Implementació Autenticació.

Sessió 2, 30/04.

- Implementació pàgines de catàleg JSP:
 - Implementació Beans del model.
 - Afegir beans a Atributes del request abans del forward.
 - Creació de les pàgines de càtaleg JSP usant els atributs rebuts.

Sessió 3, 7/05.

- Implementació pàgines de llistat de descàrregues (privada)
 - Implementació Patró Post-Redirect-GET en el doPost del ServletDispatcher.
 - Implementació de sessió guardant l'usuari, saldo i productes en la llista
 - Implementació descàrrega de fitxers.

Sessió 4, 14/05.

- Implementació servlet servei web
- Implementació lògica web service comparativa preus.

Sessió 5, 21/05.

- Implementació pàgina JSP i codi JavaScript amb la lògica de l'aplicació comparativa de preus.

Sessió 6, 28/05.

- Sesió de TEST Web Application i Web Service.

Entrega

- Sessió presencial de Test: 28 de maig.

S'haurà d'instal·lar el tomcat i la web en l'ordinador de l'aula d'informàtica. Es connectaran varis clients simultàniament.

- 31 de Maig per campus virtual.
 - Codi de la web en un fitxer .war
 - Memòria en pdf

Avaluació

En cas de que el codi no compleixi les especificacions **completes** es suspensarà l'entrega. En cas contrari es ponderarà de la següent forma:

- 35% Memòria.
- 65% Estil de codi.