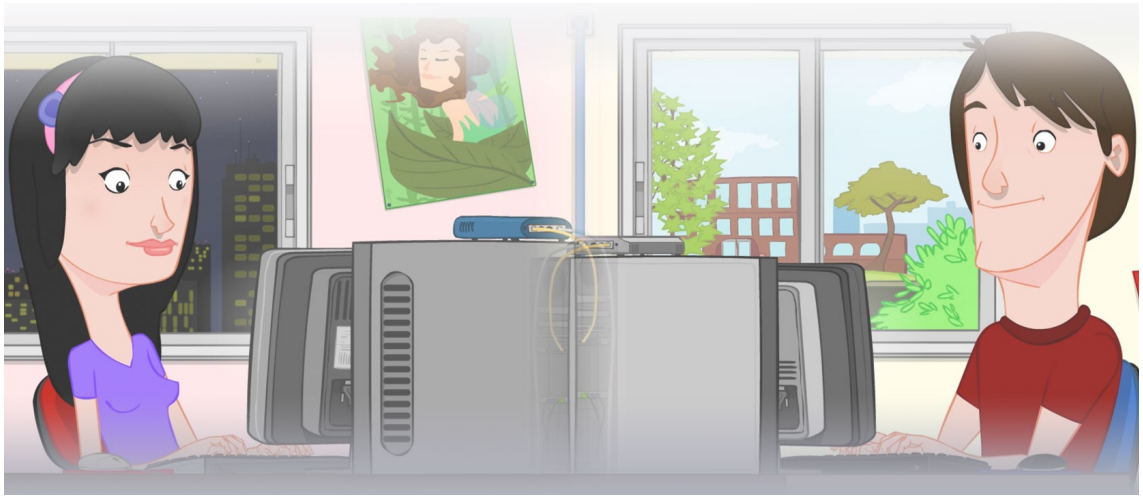
3. PRAKTIKA

Web Orrialde Dinamikoak



Universidad del País Vasco
Euskal Herriko Unibertsitatea

OCW 2015

UPV/EHU

ZERBITZU TELEMATIKO AURRERATUAK: 3. PRAKTIKA

AURKIBIDEA

3. PRAKTIKA: Web Orrialde Dinamikoak

1 Helburua

Praktika honetan web orrialde dinamikoen kontzeptua landuko da, bai bezeroaren aldetik Javascript bidez, baita zerbitzariaren aldetik ere, aplikazio zerbitzari batean exekutatzen diren servlet-ekin.

2 Client-side Scripting

2.1 Javascript

Aurreko praktikako dendaren administraziorako orrialdea hobetuko dugu. Horretarako, HTMLn javascript kodea sartuko dugu, formularioan sartutako erabiltzailearen datuak bidali baino lehen egiaztatuko dituena, guztiak ondo dauden jakiteko.

1. `<head>` elementuaren barruan `<script>` elementu bat sortu, formularioaren bidalketa botoia sakatzean exekutatuko den funtzio bat izango duelarik.
2. Javascript funtzio horrek formularioko deskribapenaren eta prezioaren eremuak hutsik ez daudela egiaztatuko du. Eremu horietatik falta direnak, fondo gorriz bistaratzea eragingo du eta `false` itzuli beharko du emaitza modura.
3. Ondoren, prezioaren eremuan zenbaki bat sartu dela egiaztatuko du. Horrela ez balitz, `alert` moduko leiho batean adieraziko luke eta oraingoan ere `false` itzuliko du emaitza modura.
4. Azkenik, guztia ondo badago `true` itzulita amaituko da.

3 Server-side Scripting

3.1 Aplikazioen Zerbitzaria

Beste irakasgai batzuetan Java lengoaiarako garapen-plataforma modura Oracle/Sun-en NetBeans IDEa erabili dugu eta GlassFish aplikazioen zerbitzaria. Aldiz, badira beste Java inguruneak ere eta zehazki lan-munduan Java EErako oso erabilia den konbinazio bat hau da: Eclipse IDEa (IBM) eta JBoss aplikazioen zerbitzaria (RedHat).

Java EEk eskatzen dituen funtzionaltasun guztiak beharrezkoak ez diren kasuetarako, beste aplikazioen zerbitzari arinagoak erabili daitezke, adibidez Tomcat (Apacherako) edo Jetty (Eclipserako). Kasu hau da oraingo praktikan gertatzen zaiguna servlet-ekin, beraz, oraingoz Tomcat aplikazioen zerbitzaria erabiliko dugu eta JBoss aurreragorako praktiketarako utziko dugu.

Lehenik, Tomcat 7ren tarball paketea jaitsiko dugu, distribuzio bitarra dakarrena[1] eta nahi den karpetan deskonprimituko dugu. Eragiketa horrekin bakarrik, aplikazioen zerbitzaria guztiz erabilgarri egongo da.

Eclipse IDEa 1. praktikatik instalatuta dagoenez, nahikoa izango da berari Tomcat deskonprimatu dugun karpeta adieraztea, Eclipse bera gai izan dadin Tomcat abiarazteko eta bertan servlet-ak argitaratzeko.

1 http://tomcat.apache.org/download-70.cgi

3.2 Web Aplikazioa

Servlet-en bitartez, aurreko bertsioko HTML orrialdeak ordezkatutako ditugu, HTML kodea zerbitzarian dinamikoki sortzen duen web aplikazio batekin. Horretarako, jadanik idatzitako HTML orrialde estatikoekin hasiko gara, gure Eclipse+Tomcat ingurunean sartuz.

1. Eclipsen Dynamic Web Project delako bat sortu eta target runtime delakoan New Runtime... sakatu, deskargatu dugun Tomcat zerbitzaria erregistratzeko. Bere kokapena adierazteko eskatuko zaizue (hau da, non deskonprimitu duzuen). Beste aukera guztiak lehenetsitako balioekin utzi ditzakezue.

2. Pauso hau bete ondoren, IDEaren ezkerraldean zuhaitz itxurako egitura bat sortua izango duzue, proiektuko osagaiekin. Zehazki, Java Resources/src kokapena Java iturri fitxategiak gordeko dituena izango da eta WebContent berriz, HTML dokumentuak eta Java bytecodeak izango dituena, azken horiek izango direlarik Eclipsek aplikazioen zerbitzarian argitaratuko dituenak.

3. Kopiatu WebContent kokapenean dendako erabiltzailearen eta administrariaren HTML eta CSS orrialdeak. Eclipsetik egiten ez baduzue (Import... → General → File System), gogoratu Refresh exekutatzea zuen Eclipse proiektuan xaguaren eskumako botoia sakatuz.

4. Proiektua exekutatu Run As→ Run on Server aukerarekin, Tomcat adieraziz. Horrekin, zuen proiektuko WebContent karpetaren edukia Tomcat-en argitaratzen da (testuinguru modura zuen Eclipse proiektuaren izena erabilita). Tomcat zerbitzariak 8080 portuan entzuten ditu eskaerak. Egiaztatu zuen HTML orrialdeak erabilgarri daudela, nabigatzailean URL egokia jarrita (**http://localhost:***portua***/***t* **/***orrialdea***.html**).

5. Une honetatik aurrera, proiektuko fitxategiren bat aldatzen duzuen bakoitzean Eclipse bera arduratzen da zerbitzarian berriz argitaratzeaz. Hau Servers estekan ikusi daiteke, Java EE ikuspegian (Perspective) agertzen dena. Zenbaitetan, egindako aldaketak benetan argitaratzeko aplikazioen zerbitzaria bera berriro hastea beharrezkoa izan daiteke; esteka horretan Restart sakatuta egin daiteke hori.

Orain arte ez dago ia ezberdintasunik praktikaren aurreko atalarekin, beraz, zerbitzariaren aldean HTML kodea dinamikoki sortuko duen kode exekutagarria gehitzen hasiko gara.

3.3 Administrariaren Servlet-a

Administrariaren orrialdea servlet batekin sortuko dugu. Horretarako:

1. Proiektuaren izenean saguaren eskumako botoiarekin, New → Servlet aukeratu, Eclipseren servlet-ak sortzeko laguntza hasteko. Servlet horrek POST eskaerak erantzun beharko ditu, eta GET eskaera batekin deitzen bada errore mezu bat adieraziko du.

2. HTML orrialdean formularioaren kodea aldatu, datuak POST bitartez bidali ditzan gure servlet-era. Gainera, erabiltzaileari eskatzen dizkion datuen artean, erreferentzia eskatzeko eremua kendu eta produktuaren izena sartzeko eremu berri bat jarri, mota eta deskribapenaren artean adibidez.

3. Servlet-a POST bidez eskatzen bada, formulario horretatik deitua izaten ari dela ulertuko dugu eta erabiltzaileak bertan sartutako datuak erabili ahalko ditugu. Balio horiekin nahi dugun eragiketa egin ahalko dugu; gure kasuan, zehazki, erreferentzia balio berri bat kalkulatzea (produktu bakoitzeko ezberdina) eta orrialde berri bat sortzea izango da eragiketa hori, mezu honen antzekoa bistaratzen

duena:

Alta berria: *DVD* **motako** *Blade Run* **produktua,** *30* **eurotako prezioarekin**

4. Azkenik, altan emandako artikulua `datuak.csv` fitxategi batean gordeko du servlet-ak. Fitxategi horretan lerro bakoitzean dendako produktu ezberdin baten informazioa egongo da, komaz banandutako eremuetan eta formatu hau jarraituta: erreferentzia (lehen sortutakoa), mota, produktuaren izena, deskribapena eta prezioa. Fitxategia existitzen ez bada, sortu egin beharko da.

3.4 Erabiltzailearen Servlet-a

Erabiltzailearen HTML orrialdea beste servlet batekin ordezkatu, dinamikoki dendako artikuluak aurkeztu ditzan. Artikulu horiek administrariak eman beharko ditu altan lehen programatu dugun servlet-arekin eta `datuak.csv` fitxategian egongo dira.

3.5 Diseinurako Patroiak

Aplikazioa oraindik oso sinplea denez, ezin daitezke arkitektura-patroien abantailak guztiz ikusi, baina erabilgarria da datuen eta aplikazioaren logika hartzen dituen modeloa (MVC patroikoa) JavaBean batean kodifikatzea, bi servlet-etatik erabili ahal izango delarik.