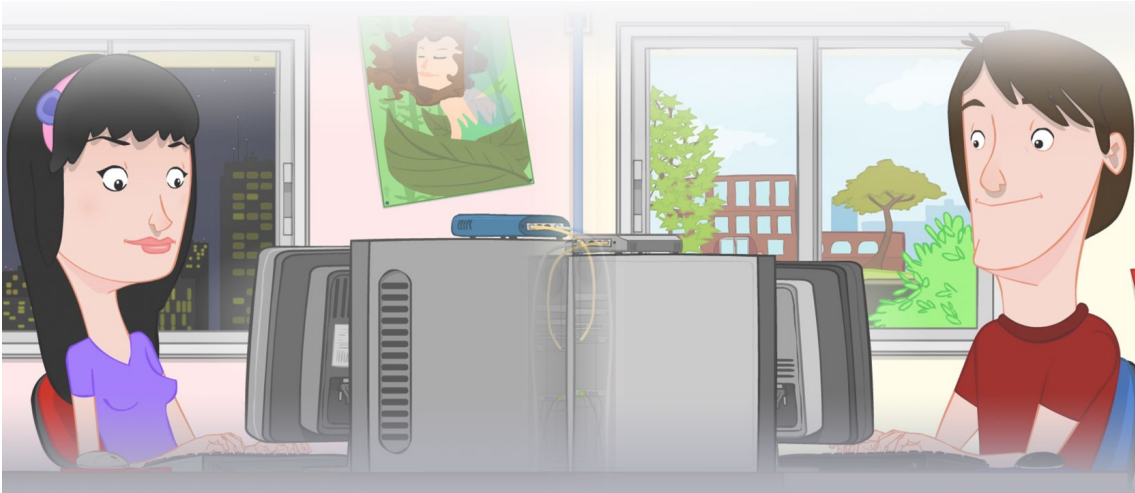


7. PRAKTIKA

JSF eta EJB



Universidad del País Vasco Euskal Herriko Unibertsitatea

OCW 2015

UPV/EHU

ZERBITZU TELEMATIKO AURRERATUAK: 7. PRAKTIKA



Copyright © 2015 Mainer Huarte Arrayago, Gorka Prieto Agujeta, Jasone Astorga Burgo, Nerea Toledo Gandarias

ZERBITZU TELEMATIKO AURRERATUAK: 7. PRAKTIKA lana, Mainer Huartek, Gorka Prietok, Jasone Astorga Burgok eta Nerea Toledo Gandariasek egin, Creative Commons-en Attribution-Share Alike 3.0 Unported License baimenaren menpe dago. Baimen horren kopia bat ikusteko, <http://creativecommons.org/licenses/by-sa/3.0/> webgunea bisitatu edo gutun bat bidali ondoko helbidera: Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA.

Lan hau beste honen eratorria da: Mainer Huarte Arrayago, Gorka Prieto Agujeta, "Servicios Telemáticos Avanzados: Práctica 7 - EJB + JPA", OCW UPV/EHU 2014 (ISSN 2255-2316), 2014.

ZERBITZU TELEMATIKO AURRERATUAK: 7. PRAKTIKA by Mainer Huarte, Gorka Prieto, Jasone Astorga Burgo and Nerea Toledo Gandarias is licensed under a Creative Commons Attribution-Share Alike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or, send a letter to Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA.

This is a derivative work from: Mainer Huarte Arrayago, Gorka Prieto Agujeta, "Servicios Telemáticos Avanzados: Práctica 7 - EJB + JPA" OCW UPV/EHU 2014 (ISSN 2255-2316), 2014.

AURKIBIDEA

1	Helburua.....	5
2	Proiektua.....	5
3	Model.....	5
3.1	Datuen Atzipena.....	5
3.2	Negozioaren logika.....	6
4	View.....	6
4.1	ManagedBeans.....	6
4.2	xhtml.....	7
5	Controller.....	7

7. PRAKTIKA: JSF eta EJB

1 Helburua

Oraingo praktika honetan, aurreko praktikako REST Web Zerbitzuak JavaEeko EJB teknologiarekin garatutako osagaiekin aldatuko ditugu. Hortaz, dendaren datu-baseak XML fitxategi bat izaten jarraitzen du soilik, baina bere kudeaketa, EJBekin egingo dugu. REST Web Zerbitzuen kasuan bezala, aplikazioaren Model atala (datuak eta logika paketea) beste zerbitzari batera mugitu ahalko genituzke; aukera hori beraz, ez dugu galtzen teknologia aldaketarekin, baina praktikan ez dugu probatuko.

2 Proiektua

EJB teknologien erabilerarako, jadanik ezin dugu Apache Tomcat-ekin lan egin JavaEE zerbitzari oso bat behar delako. Beraz, JBoss zerbitzaria erabiliko dugu hemendik aurrera.

- Dynamic Web Project → Project name, Target Runtime: New Runtime...
 - JBoss 7.1 Runtime → Next
 - Home Directory: JBoss zerbitzariaren kokapena, Execution Environment: JavaSE-7.1 → Finish
 - Target Server: New → (nahi diren izenak aldatu) Finish
- Next → Next → Generate web.xml deployment descriptor
- Properties → Project Facets: JSF 2.2 → Further configuration available...
 - JBoss zerbitzaria erabiltzen denez, JSF teknologiek lan egiteko ez da liburutegi berezirik zehaztu behar; nahikoa da liburutegia Target Runtime-ak duela adieraztea, hau da, JBoss zerbitzariak:
 - Type: Library Provided by Target Runtime, URL Mapping Patterns: Add → *.html

Proiektu honen osagai gehienak aurreko praktikako berdinak izango direnez, Java klaseen paketeak eta .html fitxategi guztiak inportatu.

3 Model

3.1 Datuen Atzipena

Dendako produktuek XML fitxategi batean egoten jarraituko dutenez, inportatutako datuak paketean ez da edukirik aldatu behar:

- XML fitxategiaren edukia zehazten duten JavaBean klaseak, @XML oharrekin apaindutakoak
- XML fitxategitik irakurri/idazteko metodoak ditu(zt)en klasea(k)

3.2 Negozioaren logika

Negozioaren logikak aurreko praktikako berdina egingo du, baina EJB osagaiekin oraingoan. Horretarako, REST Web Zerbitzuak Singleton motako EJB batera mugituko ditugu; zehazki, XML fitxategiko produktuen zerrenda itzuli eta XML fitxategian zerrenda bat idazten duten zerbitzuak, bakoitza EJBko metodo ezberdin batera.

1. EJBa sortzeko, proiektuan eskumako botoiaz **New**→**Other...**→**EJB**→**Session Bean** aukeratu. EJBari izen bat eman, zein paketetan egon behar den adierazi (**logika**) eta **No-interface View** aukeratuko dugu.
2. XML fitxategiko produktuen zerrenda itzuliko duen metodoa eta Produktuen zerrenda bat XML fitxategian idazteko metodoa, aurreko praktikako berdina izango dira, baina JAX-RS oharrik gabe.

Aurreko praktikako REST Web Zerbitzua eta oraingo EJBa alderatuta zera ikusten da, lehen zerbitzuan zeuden metodoek JSON datuak kontsumitzen/ekoizten zituztela eta orain berriz EJBek darabilten serializazio/deserializazio bitarrarekin kontsumitu/ekoiztuko dituzte datu horiek; serializazio/deserializazio bitarra eraginkorragoa da prozesamendu denbora eta sortutako trafikoaren aldetik. Alde txarrak ere baditu, adibidez JavaEE zerbitzari oso bat behar izatea edo EJBak beste zerbitzari batera mugitzeak firewall-ekin sortzen dituen arazoak.

4 View

4.1 ManagedBeans

Jadanik negozio mailan Web Zerbitzurik erabiliko ez denez, aurkezpena paketeko ManagedBeans klaseetan aldaketa batzuk egin beharko dira.

1. ProduktuaMB klasea: aldaketarik gabe
2. ProduktuenZerrendaMB klasea: aurreko praktikarekiko aldaketak hauek izango dira
 - Atributu berri bat izango du, logika paketeko EJBaren motako objektu bat izango dena.
 - Eraikitzailea: ez da berridatzi behar, lehenetsitakoarekin nahikoa da.
 - Dendako produktuen zerrenda XML fitxategitik irakurri eta itzultzen duen metodoa: EJB atributuaren metodo egokiak erabilita.
 - Atribututzat dugun produktuen zerrenda XML fitxategian idatziko duen metodoa: EJB atributuaren metodo egokiak erabilita.
 - Metodo berri bat idatzi, dendako produktuen zerrenda XML fitxategitik irakurri eta List motako atributua bere balioarekin hasieratzen duena. Bere lana egiteko, aurretik idatzi den “Dendako produktuen zerrenda XML fitxategitik irakurri eta itzultzen duen metodoa” delakoa erabiliko du. Metodo hau, ProduktuenZerrendaMB objektua sortzen denean bakarrik exekutatu behar da, eraikitzailearen ondoren; horretarako, @PostConstruct oharrarekin apaindu.

4.2 xhtml

Praktika honetan, View ataleko ManagedBeans klaseetako metodo eta atributuen izenik aldatu ez badugu, xhtml fitxategiak ez dira aldatu beharko.

5 Controller

Amaitzeko, Controller atala falta zaigu, aurreko praktikan bezala landu behar dena. Oraingoan proiektuan REST Web Zerbitzurik programatu ez dugunez, web.xml fitxategian adierazi beharreko bakarra zera da:

1. Proiektuari dagokion welcome-file elementua: aurreko praktikan bezala.
2. JSFekiko, *.xhtml fitxategiak FacesServlet-era mapeatzen dituen elementua: aurreko proiektuan bezala.