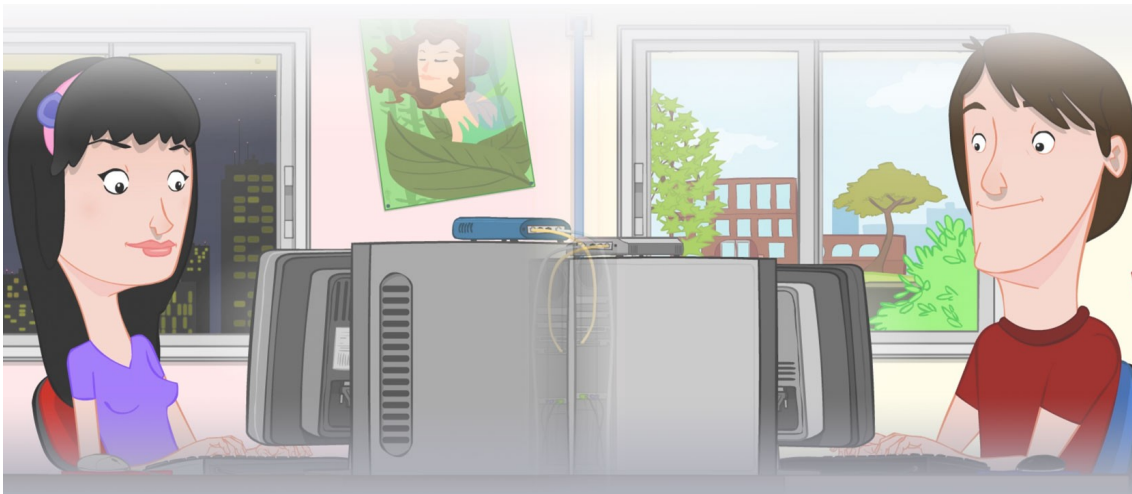


5. PRAKTIKA

REST Web Zerbitzuak



Universidad del País Vasco Euskal Herriko Unibertsitatea

OCW 2015

UPV/EHU

ZERBITZU TELEMATIKO AURRERATUAK: 5. PRAKTIKA



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

ZERBITZU TELEMATIKO AURRERATUAK: 5. PRAKTIKA lana, Mainer Huartek, Gorka Prietok, Jasone Astorga Burgok eta Nerea Toledo Gandariasek egina, 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 5 - Servicios Web", OCW UPV/EHU 2014 (ISSN 2255-2316), 2014

ZERBITZU TELEMATIKO AURRERATUAK: 5. 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 5 - Servicios Web" OCW UPV/EHU 2014 (ISSN 2255-2316), 2014

AURKIBIDEA

1	Helburua.....	5
2	REST Web zerbitzuak.....	5
2.1	Web zerbitzuak programatzea: XML.....	5
2.2	Web zerbitzuen bezeroak programatzea: XML.....	6
2.3	Web zerbitzuak programatzea: JSON.....	6
2.4	Web zerbitzuen bezeroak programatzea: JSON.....	6
3	Aukerarik egokiena: XML fitxategia eta JSON zerbitzuak.....	7

5. PRAKTIKA: REST Web Zerbitzuak

1 Helburua

Praktika honetan, gure online dendaren servlet-etatik orain arte erabili den negozioko logika, REST zerbitzuetara mugituko dugu. Hori eginik, zerbitzu horiek beste aplikazioentzako erabilgarri geratuko dira ere eta gainera, beste zerbitzari batera mugitu ahalko lirateke.

2 REST Web zerbitzuak

Apache Tomcat zerbitzarian REST Web zerbitzuak erabili ahal izateko, JAX-RS liburutegia behar dugu:

1. Jersey¹ liburutegiaren .zip fitxategia deskargatu eta makinaren karpeta batean deskonprimatu. Hainbat .jar fitxategi dituela ikusi behar dugu, api, ext eta lib izeneko azpikarpetatan.
2. Eclipseen Dynamic Web Project bat sortu eta web.xml fitxategi bat sor dadila konfiguratu (proiektua sortzeko azken leihoan agertzen da aukera hori eta horretara iristeko Next botoia sakatu behar da Finish beharrean). Proiektu berri horretan, aurrekoaren edukia inportatu.
3. web.xml fitxategia teoriako apunteetan irakatsi bezala aldatu, Jersey sevlet-ak exekutatzeko (Java EE ez diren inguruneetarako).
4. Proiektuaren propietateetan, Java Build Path eta Deployment Assembly aldatu, Jersey liburutegiko .jar guztiak erabili daitezten.

2.1 Web zerbitzuak programatzea: XML

Ondoren, aplikazioko logika REST moduko zerbitzuetara pasako dugu. Horretarako, bi zerbitzu ezberdin sortuko ditugu.

1. Produktuen zerrenda XML kodean itzultzen duen zerbitzua: logika paketeko klasean produktuen zerrenda XML fitxategitik lortzeaz arduratzen den metodoa, REST zerbitzu zerbitzu batean bihurtuko dugu. Zerbitzu berri hau zehazki HTTPko GET metodoaz atzitu beharko da. Itzultzen duen produktuen zerrenda HTTP-response mezuan XML formatuan sar dezan, XML formatuko datuak ekoiztuko ditu REST zerbitzu honek.

Zerbitzua probatzeko, teoriako klaseetako adibideekin bezala egin dezakezue, zerbitzua zuzenean nabigatzaile batetik deituta eta Wireshark-ekin HTTP-response mezua aztertuta.

2. Administrariarentzako, produktu berri bat gehitzearen zerbitzua: logika paketeko klasean produktu berria XML fitxategian sartzeaz arduratzen den metodoa, REST zerbitzu zerbitzu batean bihurtuko dugu. Zerbitzu berri hori POST metodoaz deituko da bakarrik eta sartu beharreko produktu berria XML formatuan kontsumituko du. Zerbitzuak emaitza modura ezer itzuliko ez duenez, ez du ezer ekoiztuko.

Oraingoz, zerbitzua ezingo dugu probatu.

¹ <http://repo1.maven.org/maven2/org/glassfish/jersey/bundles/jaxrs-ri/2.6/jaxrs-ri-2.6.zip>

2.2 Web zerbitzuen bezeroak programatzea: XML

Web aplikazioak REST zerbitzuak erabiltzeko:

1. Erabiltzailearen servlet-a aldatu, produktuen zerrenda aurreko atalean programatutako GET zerbitzua erabili lor dezan, XML fitxategiaren kasuan; JSON fitxategiko produktuak lehen bezala lortzen jarraituko du.
2. Administrariaren servlet-a aldatu, aurreko atalean programatutako POST zerbitzua erabili dezan produktu berria XML fitxategian sartzeko; kasu honetan ere, JSON fitxategiarekiko eragiketak berdin jarraituko du.

2.3 Web zerbitzuak programatzea: JSON

Ondoren, JSON zerbitzuak idatziko ditugu, JSON fitxategiarekin lan egingo dutenak:

1. Jackson liburutegiko .jar fitxategi hauek jaitsi (`jackson-annotations-2.5.1.jar`², `jackson-core-2.5.1.jar`³, `jackson-databind-2.5.1.jar`⁴, `jackson-jaxrs-base-2.5.1.jar`⁵, `jackson-jaxrs-json-provider-2.5.1.jar`⁶) eta **User Library** batean sartu, **Java Build Path** eta **Deployment Assembly**-n gehitu beharrekoa: horiei esker, REST zerbitzuek JSON kodea ekoiztu eta kontsumitu ahalko dute, `@XML` oharrak ulertuta.
2. Produktuen zerrenda JSON kodean itzultzen duen zerbitzua: logika paketeko klasean produktuen zerrenda JSON fitxategitik lortzeaz arduratzen den metodoa, REST zerbitzu zerbitzu batean bihurtu. Zerbitzu berri hau HTTPko GET metodoaz atzitu beharko da eta JSON formatuko datuak ekoiztuko ditu.
3. Produktu berri bat gehitzearen zerbitzua: logika paketeko klasean produktu berria JSON fitxategian sartzeko arduratzen den metodoa, REST zerbitzu zerbitzu batean bihurtu. Zerbitzu berri hau POST metodoaz deituko da eta sartu beharreko produktu berria JSON formatuan kontsumituko du.
4. Oraingoan, zerbitzua ezingo dugu probatu.

2.4 Web zerbitzuen bezeroak programatzea: JSON

Aplikazioak REST zerbitzu horiek erabiltzeko:

1. Erabiltzailearen servlet-a aldatu, JSON fitxategiko produktuen zerrenda ere, JSON zerbitzu berriarekin lor dezan.
2. Administrariaren servlet-a aldatu, produktu berria JSON fitxategian sartzeko aurreko POST zerbitzu berriarekin egin dezan.

2 <http://repo1.maven.org/maven2/com/fasterxml/jackson/core/jackson-annotations/2.5.1/>

3 <http://repo1.maven.org/maven2/com/fasterxml/jackson/core/jackson-core/2.5.1/>

4 <http://repo1.maven.org/maven2/com/fasterxml/jackson/core/jackson-databind/2.5.1/>

5 <http://repo1.maven.org/maven2/com/fasterxml/jackson/jaxrs/jackson-jaxrs-base/2.5.1/>

6 <http://repo1.maven.org/maven2/com/fasterxml/jackson/jaxrs/jackson-jaxrs-json-provider/2.5.1/>

3 Aukerarik egokiena: XML fitxategia eta JSON zerbitzuak

Proiektu berri batean, aplikazioa aldatu XML fitxategiarekin bakarrik lan egin dezan (nabigatzaile batekin egiaztatzeko formatu erraza), baina JSON kontsumitu eta ekoizten duten zerbitzuekin (trafiko gutxien sortzen duen formatukoak).

1. Aurreko ataleko JSON zerbitzuak aldatu, JSON fitxategiarekin lan egin beharrean XML fitxategiarekin egin dezaten.
2. Aplikazioan soberan geratu diren osagaiak kendu: JSON fitxategia lantzeko metodoak, XML zerbitzuak, eta abar.