Studentų 50 – 213
Kompiuterių katedra

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Informatikos inžinerijos studijų programa

Laboratorinis darbas Nr. 2

T120B143 Įmonių kompiuterinių sistemų kūrimo platformos

Įmonės kompiuterinės sistemos realizacija panaudojant JAX-WS ir JAX-RS technologijas

ATLIKO:

TadasLaurinaitis _____ **IFF-6/8**
(Vardas Pavardė) (Parašas) (Grupė)

DĖSTYTOJAS:

doc. A. Liutkevičius _____
(Vardas Pavardė) (Parašas)

DARBAS ATIDUOTAS:

___d._____mėn. 2019

KAUNAS, 2019

Įmonės kompiuterinės sistemos realizacija panaudojant JAX-WS ir JAX-RS technologijas

Užduotis

Uždaviniai:

- Sukurti paslaugas naudojant JAX-RS.
- Sukurti paslaugas naudojant JAX-WS.
- Sukurti vartotojo sąsają panaudojant JavaServer Pages.

Sprendimo architektūra/aprašymas

Darbas buvo atliktas naudojant NetBeans IDE (v7.3.1) su Java EE, Java SE, GlassFish ir Apache Derby. Darbas buvo darytas naudojant pirmo laboratorinio darbo projektą. Siekiant įgyvendinti visus uždavinius, bus sukurta internetinė programa, kurioje vartotojas turės galimybę palikti komentarą su savo vardu. Komentarai saugomi duomenų bazėje.

Laboratorinio darbo atlikimo etapai:

1. JAX-RS paslaugos kūrimas: sukuriame naują RESTful web servisą iš Message entity klasės. AplicationConfig.java faile pakeičiame ApplicationPath į "rest", taip pat pakeičiame MessageFacadeREST.java faile @Path parametrą į „msg". Atlikus šiuos žingsnius paslaugą galime pasiekti adresu „http://localhost:8080/jlab1/rest/msg".

2. JAX-WS paslaugos kūrimas: Iš pradžių sukuriame SessionBeansForEntityClasses iš Message entity, kurį naudos kuriamas web servisas. Naujai sukurto Bean paskirtis – leisti jax-ws servisui pasiekti pranešimus laikomus duomenų bazėje. Sukuriame naują web servisa, suteikiame jam MessageWS pavadinimą bei prie Create Web Service from Existing Session Bean paarenkame MessageFacade. Tada generuojame SOAP-HTTP įvynioklį, paspausdami dešinį pelės klavišą ant MessageWS klasės Web Services šakoje ir pasirinkę „Generate SOAP-over-HTTP wrapper". Sukurtą naują servisą galime pratestuoti paspaudę ant jo dešinį pelės klavišą ir pasirinkę Test web service.

3. Vartotojo sąsajos kūrimas: Vartotojo sąsają modifikuosime taip, kad pranešimai būtų užkraunami dinamiškai, panaudojant AJAX asinchroninę užklausą į sukurtą RESTful paslaugą, kuri gražina xml pranešimus. Tam panaudosime kodą iš Priedas 1. Tada vartotojo sąsaja modifikuojama taip, kad pranešimai būt rodomi ir juos užkraunant per wsdl paslaugą. Namai.jsp faile po lentele pridedame kodo gabalą, kuris atspausdins pranešimo autorių ir pranešimą.

Sprendimo programinis kodas

Šiame skyriuje pateikiamas pilnas visos sistemos kodas.

namai.jsp:

```
<%--
   Document   : index
   Created on : Nov 11, 2013, 12:55:32 PM
   Author     : ENQ
--%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
```

```
    <title> My first JSP   </title>
     <style>
        TABLE, TD, TH, TR {
           border-collapse:collapse;
           border-width: 1px;
           border-style: solid;
           border-spacing: 5px;
           padding: 2px 5px;
        }
     </style>


<script type="text/javascript">
   function getXMLObject()  //XML OBJECT
   {
     var xmlHttp = false;
     try {
      xmlHttp = new ActiveXObject("Msxml2.XMLHTTP")  // For Old Microsoft Browsers
     }
     catch (e) {
      try {
        xmlHttp = new ActiveXObject("Microsoft.XMLHTTP")  // For Microsoft IE 6.0+
      }
      catch (e2) {
        xmlHttp = false   // No Browser accepts the XMLHTTP Object then false
      }
     }
     if (!xmlHttp && typeof XMLHttpRequest != 'undefined') {
      xmlHttp = new XMLHttpRequest();        //For Mozilla, Opera Browsers
     }
     return xmlHttp;  // Mandatory Statement returning the ajax object created
   }
   function getMessages()
   {
      deleteTableRow(-1);
      var xmlHttp = getXMLObject();
      xmlHttp.onreadystatechange=function()
      {
         if(xmlHttp.readyState==4)
         {
            var xmldoc = xmlHttp.responseXML;
            var msgs = xmldoc.getElementsByTagName("messages")[0].childNodes;
            console.log(msgs);
            for(var i=0;i<msgs.length;i++)
            {
               var t_id="", t_n="", t_m="", t_d="";
               if(msgs[i].getElementsByTagName("id")[0])
                  t_id = msgs[i].getElementsByTagName("id")[0].childNodes[0].textContent;
               if(msgs[i].getElementsByTagName("name")[0])
                  t_n = msgs[i].getElementsByTagName("name")[0].childNodes[0].textContent;
               if(msgs[i].getElementsByTagName("message")[0])
                  t_m = msgs[i].getElementsByTagName("message")[0].childNodes[0].textContent;
               if(msgs[i].getElementsByTagName("time")[0])
                  t_d = msgs[i].getElementsByTagName("time")[0].childNodes[0].textContent;
               drawMessageTableRow(t_id,t_n,t_m,t_d);
            }
         }
      }
      xmlHttp.open("GET","jlab1/rest/msg",true);
      xmlHttp.send(null);
```

```
    }
    function drawMessageTableRow(id, name, msg, time)
    {
        var body = document.getElementById("msgs_table_body");
        if( body )
        {
            var c_i = document.createElement('TD');
            c_i.innerHTML = id;
            var c_n = document.createElement('TD');
            c_n.innerHTML = name;
            var c_m = document.createElement('TD');
            c_m.innerHTML = msg;
            var c_t = document.createElement('TD');
            c_t.innerHTML = time;
            var r = document.createElement('TR');
            r.appendChild(c_i);
            r.appendChild(c_n);
            r.appendChild(c_m);
            r.appendChild(c_t);
            body.appendChild(r);
        }
    }
    function deleteTableRow(i)
    {
        var b = document.getElementById("msgs_table_body");
        if( i > -1) b.deleteRow(i);
        else b.innerHTML = "";
    }
</script>
</head>
<body align="center" onload="getMessages()">
    <form action="/jlab1" method="POST">
            Vardas
            <input type="text" name="name"size="20px">
              Komentaras
            <input type="text" name="message"size="20px">
            <input type="submit" value="Siųsti">
      </form>
    <hr>
    <div>
        <c:if test="${not empty msg}">
            <jsp:getProperty name="msg" property="name"/>:
            <jsp:getProperty name="msg" property="msg"/>
        </c:if>
    </div>
    <hr>
    <div align="center">
        <button onclick="getMessages()">Atnaujinti pranešimus</button>
        <div id="messages">
            <table id="msgs_table">
                <thead>
                    <tr>
                        <th>ID</th>
                        <th>Vardas</th>
                        <th>Pranesimas</th>
                        <th>Data</th>
                    </tr>
                </thead>
                <tbody id="msgs_table_body"></tbody>
            </table>
```

```
                <%-- start web service invocation --%><hr/>
            <%
            try {
                jlab1.servlets.service_client.MessageWS_Service service = new
jlab1.servlets.service_client.MessageWS_Service();
                jlab1.servlets.service_client.MessageWS port = service.getMessageWSPort();
                // TODO process result here
                java.util.List<jlab1.servlets.service_client.Message> result = port.findAll();
                for( int i=0; i < result.size(); i++) {
                    jlab1.servlets.service_client.Message tmsg = result.get(i);
                    out.println(tmsg.getName()+"-"+tmsg.getMessage()+"<br>");
                }

            } catch (Exception ex) {
                // TODO handle custom exceptions here
            }
            %>
            <%-- end web service invocation --%><hr/>
            </div>
        </div>
    </body>
</html>
```

ApplicationConfig.java:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package jlab1.servlets.service;

import java.util.Set;
import javax.ws.rs.core.Application;

/**
 *
 * @author Administrator
 */
@javax.ws.rs.ApplicationPath("rest")
public class ApplicationConfig extends Application {

    @Override
    public Set<Class<?>> getClasses() {
        Set<Class<?>> resources = new java.util.HashSet<Class<?>>();
        // following code can be used to customize Jersey 2.0 JSON provider:
        try {
            Class jsonProvider = Class.forName("org.glassfish.jersey.jackson.JacksonFeature");
            // Class jsonProvider = Class.forName("org.glassfish.jersey.moxy.json.MoxyJsonFeature");
            // Class jsonProvider = Class.forName("org.glassfish.jersey.jettison.JettisonFeature");
            resources.add(jsonProvider);
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(getClass().getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
        addRestResourceClasses(resources);
        return resources;
    }

    /**
     * Do not modify addRestResourceClasses() method.
     * It is automatically re-generated by NetBeans REST support to populate
     * given list with all resources defined in the project.
```

```
     */
    private void addRestResourceClasses(Set<Class<?>> resources) {
        resources.add(jlab1.servlets.service.MessageFacadeREST.class);
        resources.add(jlab1.servlets.service.MessageWSPort.class);
    }

}
```

MessageFACADEREST.java:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package jlab1.servlets.service;

import java.util.List;
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.ws.rs.Consumes;
import javax.ws.rs.DELETE;
import javax.ws.rs.GET;
import javax.ws.rs.POST;
import javax.ws.rs.PUT;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;
import jlab1.entities.Message;

/**
 *
 * @author Administrator
 */
@Stateless
@Path("msg")
public class MessageFacadeREST extends AbstractFacade<Message> {
    @PersistenceContext(unitName = "jlab1PU")
    private EntityManager em;

    public MessageFacadeREST() {
        super(Message.class);
    }

    //@GET
    //@Override
    //@Path("/zz/{id}/{name}")
    @POST
    @Consumes({"application/xml", "application/json"})
    public void create(Message entity/*, @PathParam("id") Integer id, @PathParam("name") String
name*/) {
        /*entity = new Message();
        entity.setId(id);
        entity.setName(name);*/
        super.create(entity);
    }

    @PUT
    @Override
    @Consumes({"application/xml", "application/json"})
    public void edit(Message entity) {
```

```
        super.edit(entity);
    }

    @DELETE
    @Path("{id}")
    public void remove(@PathParam("id") Integer id) {
        super.remove(super.find(id));
    }

    @GET
    @Path("{id}")
    @Produces({"application/xml", "application/json"})
    public Message find(@PathParam("id") Integer id) {
        return super.find(id);
    }

    @GET
    @Override
    @Produces({"application/xml", "application/json"})
    public List<Message> findAll() {
        return super.findAll();
    }

    @GET
    @Path("{from}/{to}")
    @Produces({"application/xml", "application/json"})
    public List<Message> findRange(@PathParam("from") Integer from, @PathParam("to") Integer to) {
        return super.findRange(new int[]{from, to});
    }

    @GET
    @Path("count")
    @Produces("text/plain")
    public String countREST() {
        return String.valueOf(super.count());
    }

    @Override
    protected EntityManager getEntityManager() {
        return em;
    }

}
```

MessageFacade.java:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package jlab1.servlets.service;

import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import jlab1.entities.Message;

/**
 *
 * @author Administrator
 */
```

```
@Stateless
public class MessageFacade extends AbstractFacade<Message> {
    @PersistenceContext(unitName = "jlab1PU")
    private EntityManager em;

    @Override
    protected EntityManager getEntityManager() {
        return em;
    }

    public MessageFacade() {
        super(Message.class);
    }

}
```

MessageWS.java:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package jlab1.servlets.service;

import java.util.List;
import javax.ejb.EJB;
import javax.jws.Oneway;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;
import jlab1.entities.Message;

/**
 *
 * @author Administrator
 */
@WebService(serviceName = "MessageWS")
public class MessageWS {
    @EJB
    private MessageFacade ejbRef;// Add business logic below. (Right-click in editor and choose
    // "Insert Code > Add Web Service Operation")

    @WebMethod(operationName = "create")
    @Oneway
    public void create(@WebParam(name = "entity") Message entity) {
        ejbRef.create(entity);
    }

    @WebMethod(operationName = "edit")
    @Oneway
    public void edit(@WebParam(name = "entity") Message entity) {
        ejbRef.edit(entity);
    }

    @WebMethod(operationName = "remove")
    @Oneway
    public void remove(@WebParam(name = "entity") Message entity) {
        ejbRef.remove(entity);
    }

    @WebMethod(operationName = "find")
```

```
    public Message find(@WebParam(name = "id") Object id) {
        return ejbRef.find(id);
    }

    @WebMethod(operationName = "findAll")
    public List<Message> findAll() {
        return ejbRef.findAll();
    }

    @WebMethod(operationName = "findRange")
    public List<Message> findRange(@WebParam(name = "range") int[] range) {
        return ejbRef.findRange(range);
    }

    @WebMethod(operationName = "count")
    public int count() {
        return ejbRef.count();
    }

}
```

MessageWSPort.java:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package jlab1.servlets.service;

import java.util.List;
import javax.ws.rs.Consumes;
import javax.ws.rs.GET;
import javax.ws.rs.POST;
import javax.ws.rs.PUT;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.core.Context;
import javax.ws.rs.core.UriInfo;
import javax.xml.bind.JAXBElement;
import javax.xml.namespace.QName;

/**
 * REST Web Service
 *
 * @author Administrator
 */
@Path("messagewsport")
public class MessageWSPort {
    private jlab1.servlets.service_client.MessageWS port;

    @Context
    private UriInfo context;

    /**
     * Creates a new instance of MessageWSPort
     */
    public MessageWSPort() {
        port = getPort();
    }

    /**
```

```
     * Invokes the SOAP method remove
     * @param entity resource URI parameter
     */
    @PUT
    @Consumes("application/xml")
    @Path("remove/")
    public void putRemove(JAXBElement<jlab1.servlets.service_client.Message> entity) {
        try {
            // Call Web Service Operation
            if (port != null) {
                port.remove(entity.getValue());
            }
        } catch (Exception ex) {
            // TODO handle custom exceptions here
        }
    }

    /**
     * Invokes the SOAP method count
     * @return an instance of java.lang.String
     */
    @GET
    @Produces("text/plain")
    @Consumes("text/plain")
    @Path("count/")
    public String getCount() {
        try {
            // Call Web Service Operation
            if (port != null) {
                int result = port.count();
                return new java.lang.Integer(result).toString();
            }
        } catch (Exception ex) {
            // TODO handle custom exceptions here
        }
        return null;
    }

    /**
     * Invokes the SOAP method find
     * @param id resource URI parameter
     * @return an instance of javax.xml.bind.JAXBElement<jlab1.servlets.service_client.Message>
     */
    @POST
    @Produces("application/xml")
    @Consumes("application/xml")
    @Path("find/")
    public JAXBElement<jlab1.servlets.service_client.Message> postFind(JAXBElement<Object> id) {
        try {
            // Call Web Service Operation
            if (port != null) {
                jlab1.servlets.service_client.Message result = port.find(id.getValue());
                return new JAXBElement<jlab1.servlets.service_client.Message>(new
QName("http//service_client.servlets.jlab1/", "message"), jlab1.servlets.service_client.Message.class,
result);
            }
        } catch (Exception ex) {
            // TODO handle custom exceptions here
        }
        return null;
```

```
    }

    /**
     * Invokes the SOAP method create
     * @param entity resource URI parameter
     */
    @PUT
    @Consumes("application/xml")
    @Path("create/")
    public void putCreate(JAXBElement<jlab1.servlets.service_client.Message> entity) {
        try {
            // Call Web Service Operation
            if (port != null) {
                port.create(entity.getValue());
            }
        } catch (Exception ex) {
            // TODO handle custom exceptions here
        }
    }

    /**
     * Invokes the SOAP method findRange
     * @param range resource URI parameter
     * @return an instance of javax.xml.bind.JAXBElement<jlab1.servlets.service_client.FindRangeResponse>
     */
    @POST
    @Produces("application/xml")
    @Consumes("application/xml")
    @Path("findrange/")
    public JAXBElement<jlab1.servlets.service_client.FindRangeResponse> postFindRange(JAXBElement<List<Integer>> range) {
        try {
            // Call Web Service Operation
            if (port != null) {
                java.util.List<jlab1.servlets.service_client.Message> result = port.findRange(range.getValue());

                class FindRangeResponse_1 extends jlab1.servlets.service_client.FindRangeResponse {

                    FindRangeResponse_1(java.util.List<jlab1.servlets.service_client.Message> _return) {
                        this._return = _return;
                    }
                }
                jlab1.servlets.service_client.FindRangeResponse response = new FindRangeResponse_1(result);
                return new jlab1.servlets.service_client.ObjectFactory().createFindRangeResponse(response);
            }
        } catch (Exception ex) {
            // TODO handle custom exceptions here
        }
        return null;
    }

    /**
     * Invokes the SOAP method findAll
     * @return an instance of javax.xml.bind.JAXBElement<jlab1.servlets.service_client.FindAllResponse>
     */
    @GET
```

```
    @Produces("application/xml")
    @Consumes("text/plain")
    @Path("findall/")
    public JAXBElement<jlab1.servlets.service_client.FindAllResponse> getFindAll() {
        try {
            // Call Web Service Operation
            if (port != null) {
                java.util.List<jlab1.servlets.service_client.Message> result = port.findAll();

                class FindAllResponse_1 extends jlab1.servlets.service_client.FindAllResponse {

                    FindAllResponse_1(java.util.List<jlab1.servlets.service_client.Message> _return) {
                        this._return = _return;
                    }
                }
                jlab1.servlets.service_client.FindAllResponse response = new FindAllResponse_1(result);
                return new jlab1.servlets.service_client.ObjectFactory().createFindAllResponse(response);
            }
        } catch (Exception ex) {
            // TODO handle custom exceptions here
        }
        return null;
    }

    /**
     * Invokes the SOAP method edit
     * @param entity resource URI parameter
     */
    @PUT
    @Consumes("application/xml")
    @Path("edit/")
    public void putEdit(JAXBElement<jlab1.servlets.service_client.Message> entity) {
        try {
            // Call Web Service Operation
            if (port != null) {
                port.edit(entity.getValue());
            }
        } catch (Exception ex) {
            // TODO handle custom exceptions here
        }
    }

    /**
     *
     */
    private jlab1.servlets.service_client.MessageWS getPort() {
        try {
            // Call Web Service Operation
            jlab1.servlets.service_client.MessageWS_Service service = new
jlab1.servlets.service_client.MessageWS_Service();
            jlab1.servlets.service_client.MessageWS p = service.getMessageWSPort();
            return p;
        } catch (Exception ex) {
            // TODO handle custom exceptions here
        }
        return null;
    }
}
```

Rezultatų apibendrinimas

Šiame skyriuje trumpai reziumuojama, kas buvo atlikta darbo metu, kokie projektiniai ir architektūriniai sprendimai pasirinkti, kitokiomis priemonėmis buvo atliktas projektavimas ir realizacija, įvertinamas šių priemonių efektyvumas ir t.t.

Laboratorinio darbo metu, jokių architektūrinių sprendimų priimti neteko, kadangi visi naudojami įrankiai ir technologijos buvo ir taip parinkti. Laboratorinio darbo atlikimas buvo ganėtinai nesudėtingas, instrukcijos buvo aiškios bei informatyvios, todėl daug klausimų nekilo. Laboratorinio darbo metu naudotos technologijos – SOAP ir REST servisai yra nauodojamos iki šiol, todėl buvo labai naudinga išmokti juos kurti. Atlikus laboratorinį darbą aiškiai matome, kad sukurti paprastus servisus nėra sudėtinga, todėl prireikus tai galima nesunkiai padaryti.