KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
Kompiuterių katedra

Įmonių kompiuterinių sistemų kūrimo platformos (T120B143)

Laboratorinis darbas

JAX-WS ir JAX-RS technologijos

Kaunas, 2019

Tikslas: Susipažinti su Java programavimo įrankiais, JAX-WS, JAX-RS.
Uždaviniai:

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

Darbo aplinkos paruošimas

Darbui atlikti kompiuteryje turi būti suinstaliuota NetBeans su Java EE, Java SE, GlassFish serveris ir lokali Apache Derby duomenų bazė. NetBeans parsisiųsti galima adresu https://netbeans.org/downloads/. Tinka Java EE arba All NetBeans versija. Patartina parsisiųsti pilną versiją. Taip pat reikia turėti atliktą „JavaServer Pages, Servlet ir Java Persistence“ laboratorinį darbą.

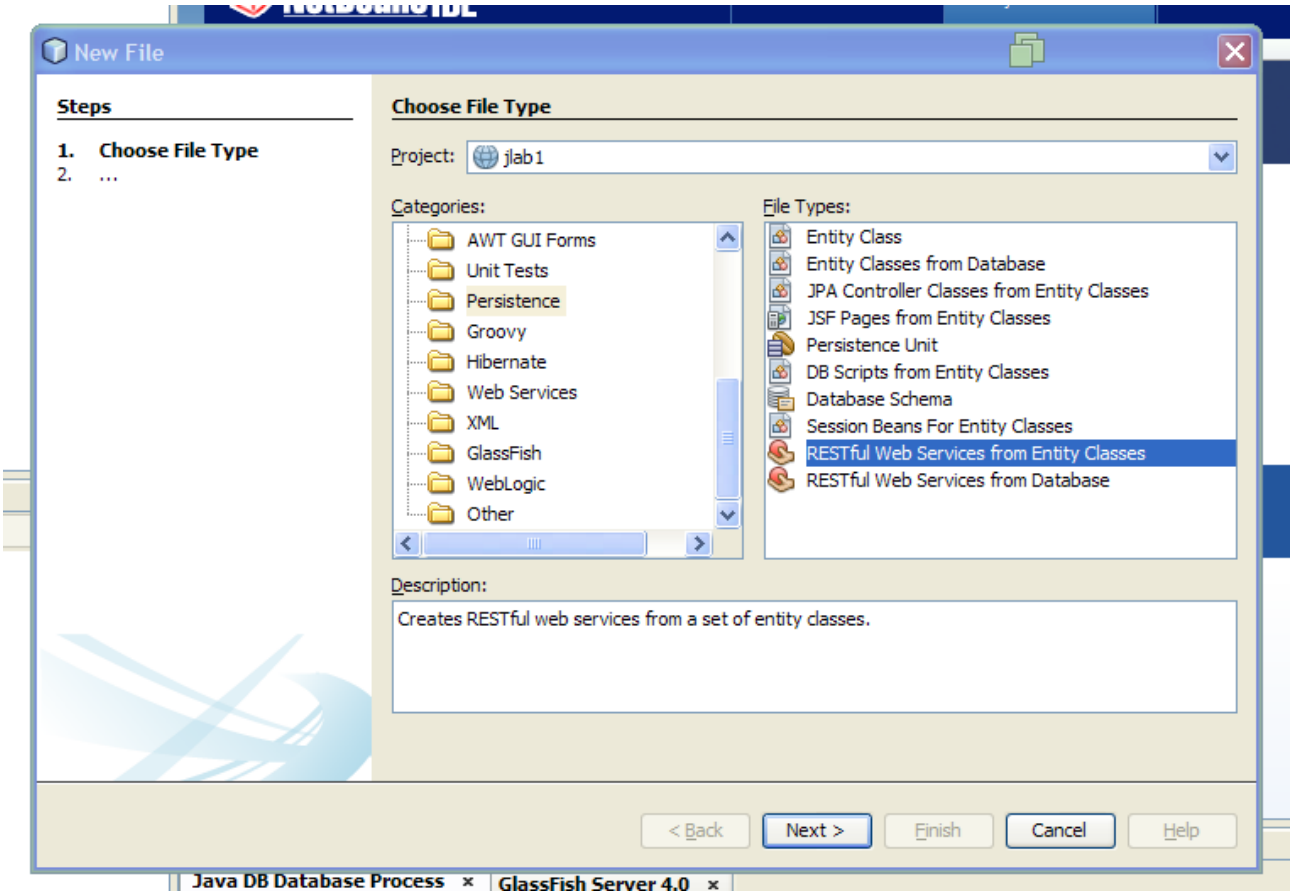
Darbo eiga

Darbas bus atliekamas naudojant NetBeans IDE (rekomenduojama ir patikrinta versija – 7.3.1). Kompiuteryje taip pat reikės duomenų bazės, kurioje bus saugomi duomenys. NetBeans turi integruotą Apache Derby duomenų bazę, kuria galima naudotis. Siekiant įgyvendinti visus uždavinius, bus sukurta internetinė programa, kurioje vartotojas turės galimybę palikti komentarą su savo vardu. Komentarai saugomi duomenų bazėje.

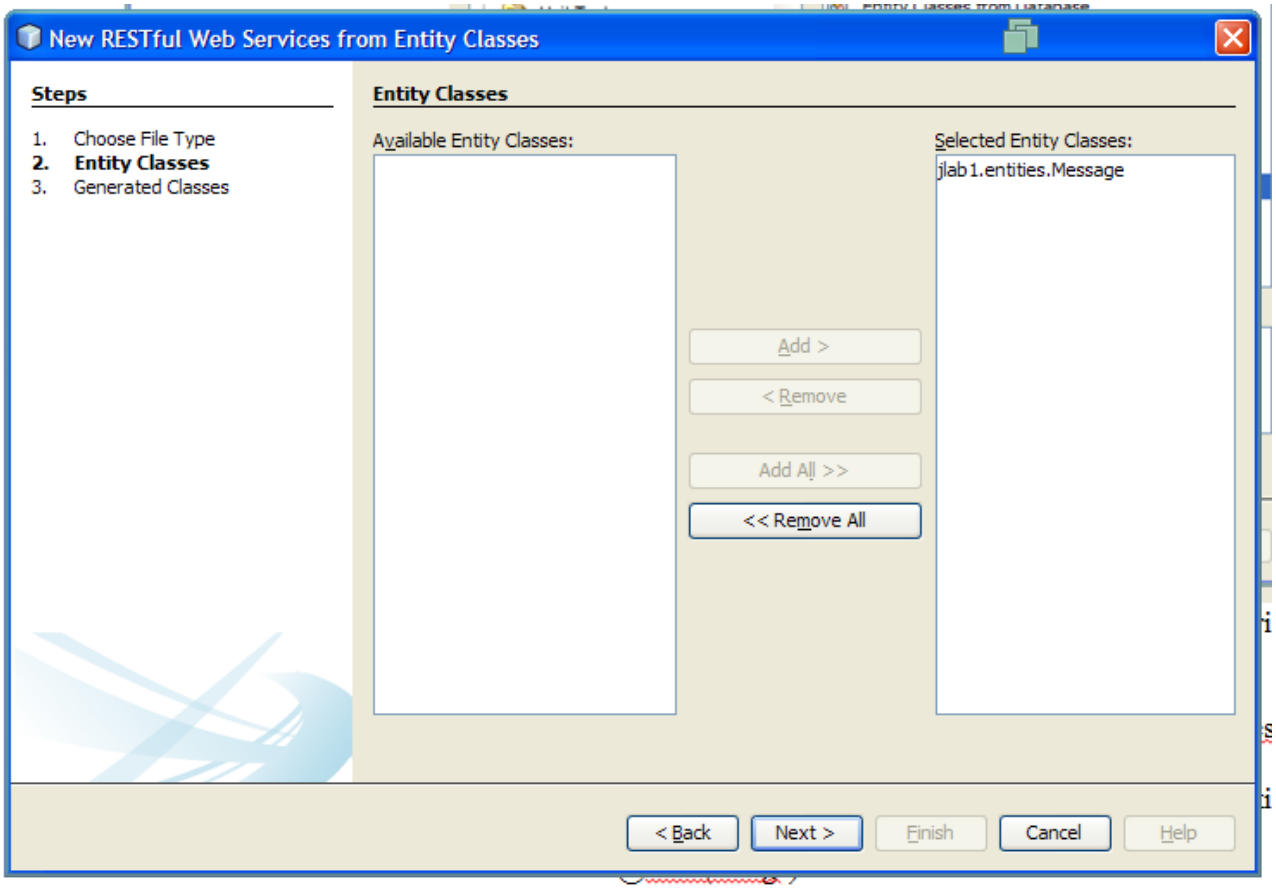
1. JAX-RS paslaugos kūrimas

jlab1 projekte sukurkite RESTful internetinę paslaugą. Paslaugoje turi būti realizuota kūrimo, redagavimo, šalinimo, paieškos metodai, kurie grąžintų xml arba json atsakymą į užklausą.

jlab1 projekte sukurkite naują failą, *File/New File*, *Persistence* kategorijoje *RESTful Web Service from Entity Classes*:



pasirinkite jlab1.entities.Message ir įtraukite į skiltį dešinėje pusėje, užbaikite paslaugos kūrimą:



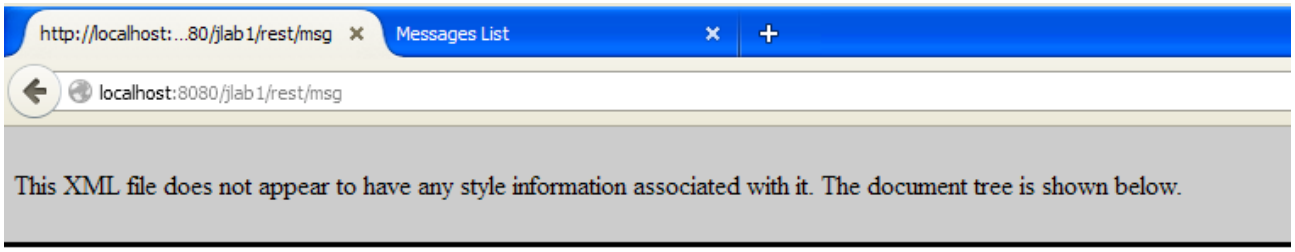
jlab1.beans.service/ApplicationConfig.java faile, pakeiskite ApplicationPath į *@javax.ws.rs.ApplicationPath("rest"):*

```
7  import java.util.Set;
8  import javax.ws.rs.core.Application;
9
10 /**
11  *
12  * @author Administrator
13  */
   @javax.ws.rs.ApplicationPath("rest")
15 public class ApplicationConfig extends Application {
16
17     @Override
       public Set<Class<?>> getClasses() {
19         Set<Class<?>> resources = new java.util.HashSet<Class<?>>();
20         // following code can be used to customize Jersey 2.0 JSON provider:
21         try {
22             Class jsonProvider = Class.forName("org.glassfish.jersey.jackson.J
23             // Class jsonProvider = Class.forName("org.glassfish.jersey.moxy.j
```

jlab1.beans.service/MessageFacadeREST.java faile, @Path parametrą, pakeiskite į @Path(„msg“):

```
19  import jlab1.entities.Message;
20
21  /**
22   *
23   * @author Administrator
24   */
25  @Stateless
    @Path("msg")
    public class MessageFacadeREST extends AbstractFacade<Message> {
28      @PersistenceContext(unitName = "jlab1PU")
29      private EntityManager em;
30
31      public MessageFacadeREST() {
32          super(Message.class);
33      }
34
35      @POST
36      @Override
```

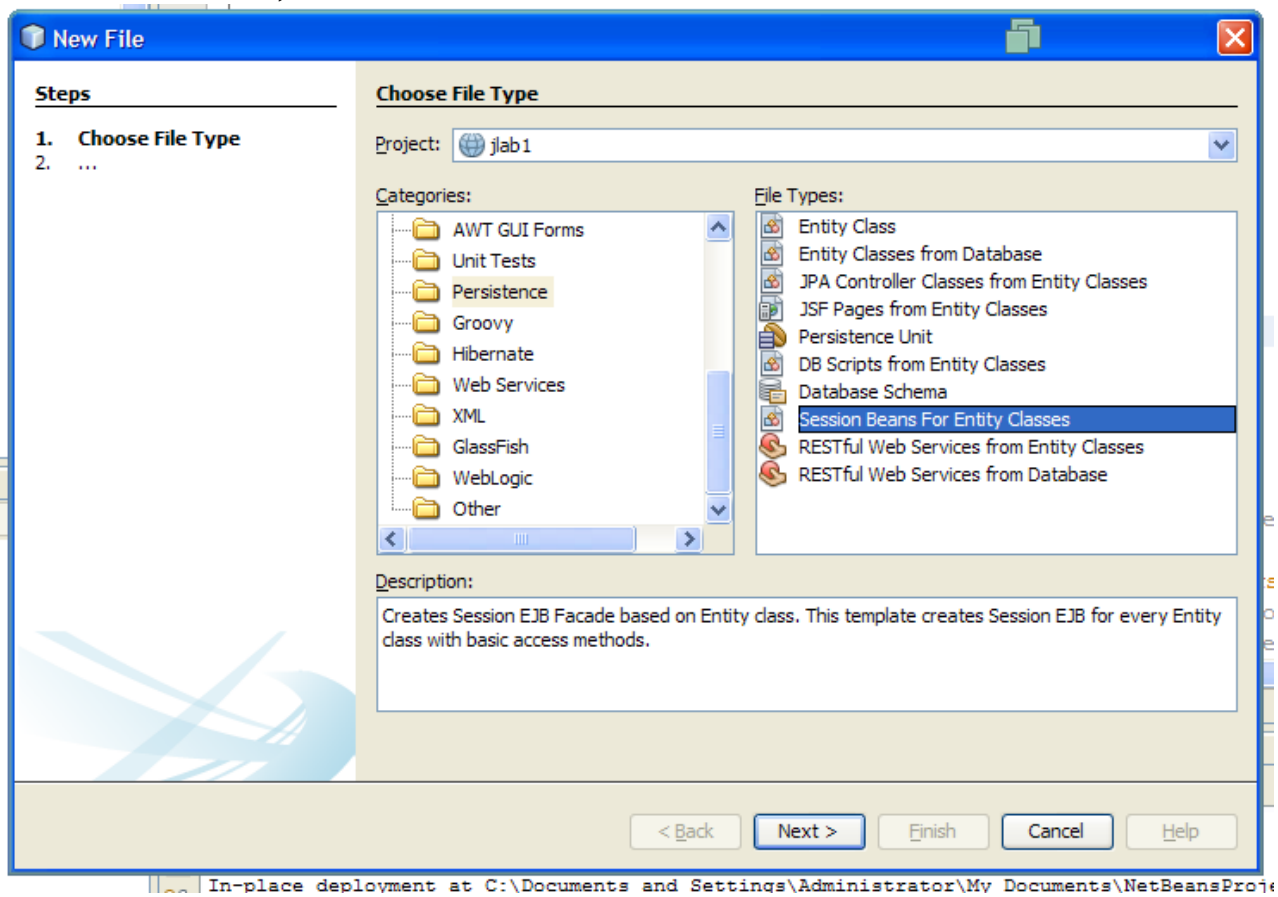
Sukurtą paslaugą galite pasiekti adresu „http://localhost:8080/jlab1/rest/msg“:



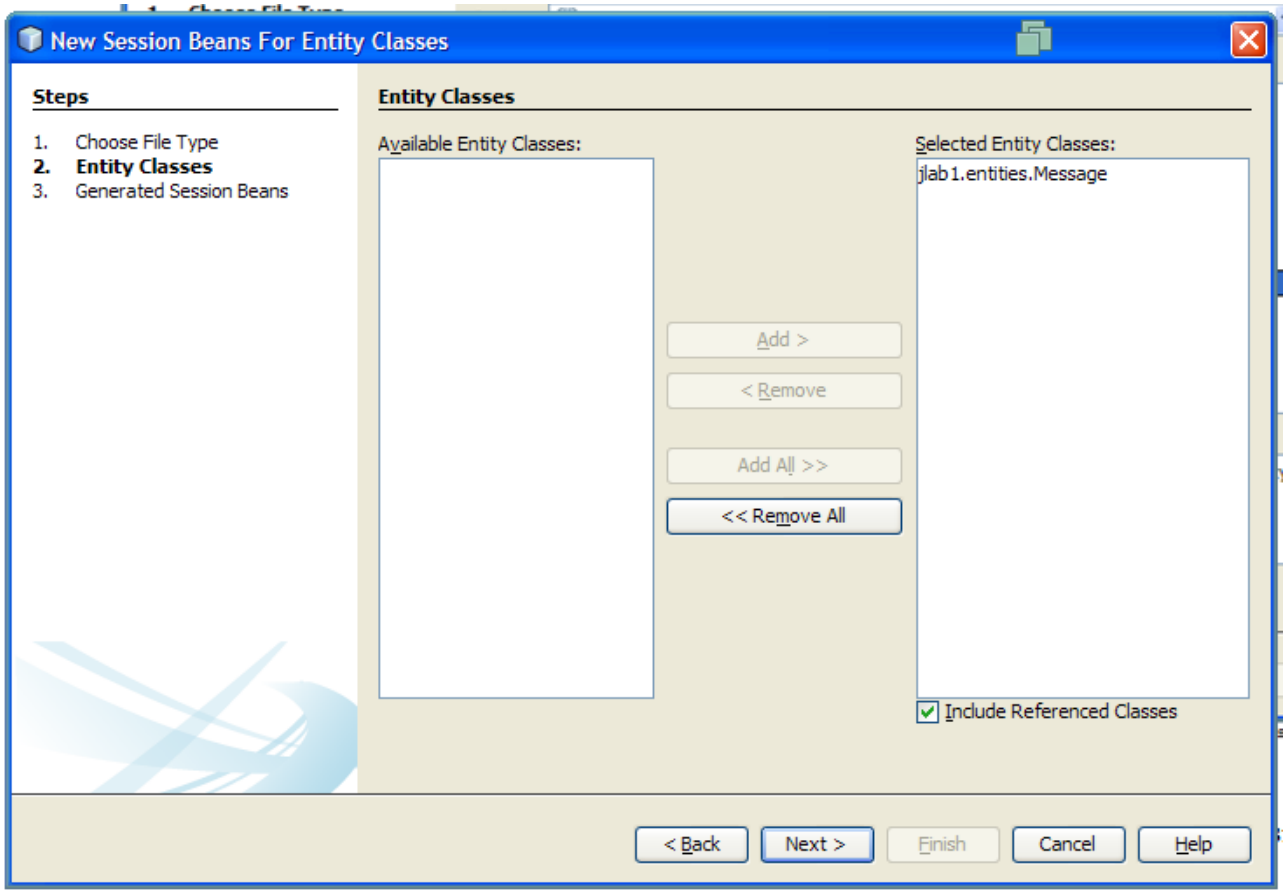

```
−<messages>
  −<message>
      <id>26</id>
      <message>Testas</message>
      <name>Agnius</name>
      <time>1970-01-01T18:57:43+03:00</time>
    </message>
  −<message>
      <id>27</id>
      <message>Tikiuosi daugiau bug'u nebus :)</message>
      <name>Agnius</name>
      <time>1970-01-01T18:58:07+03:00</time>
    </message>
</messages>
```

2. JAX-WS paslaugos kūrimas

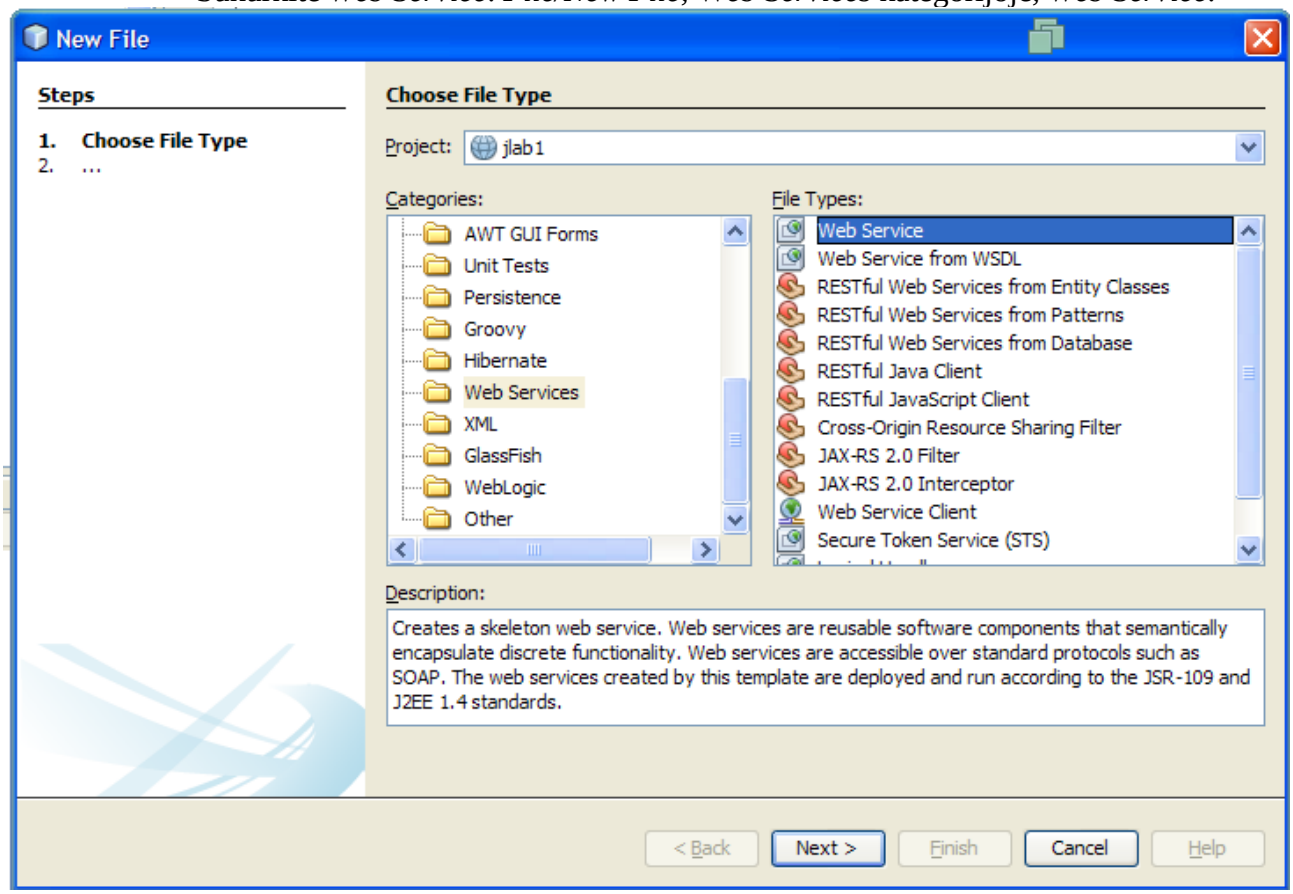
- Jlab1 projekte sukurkite naują *Session Beans For Entity Classes*, kurį naudos kuriamas web servisas. *File/New File, Persistence* kategorijoje *Session Beans For entity Classes*, next.



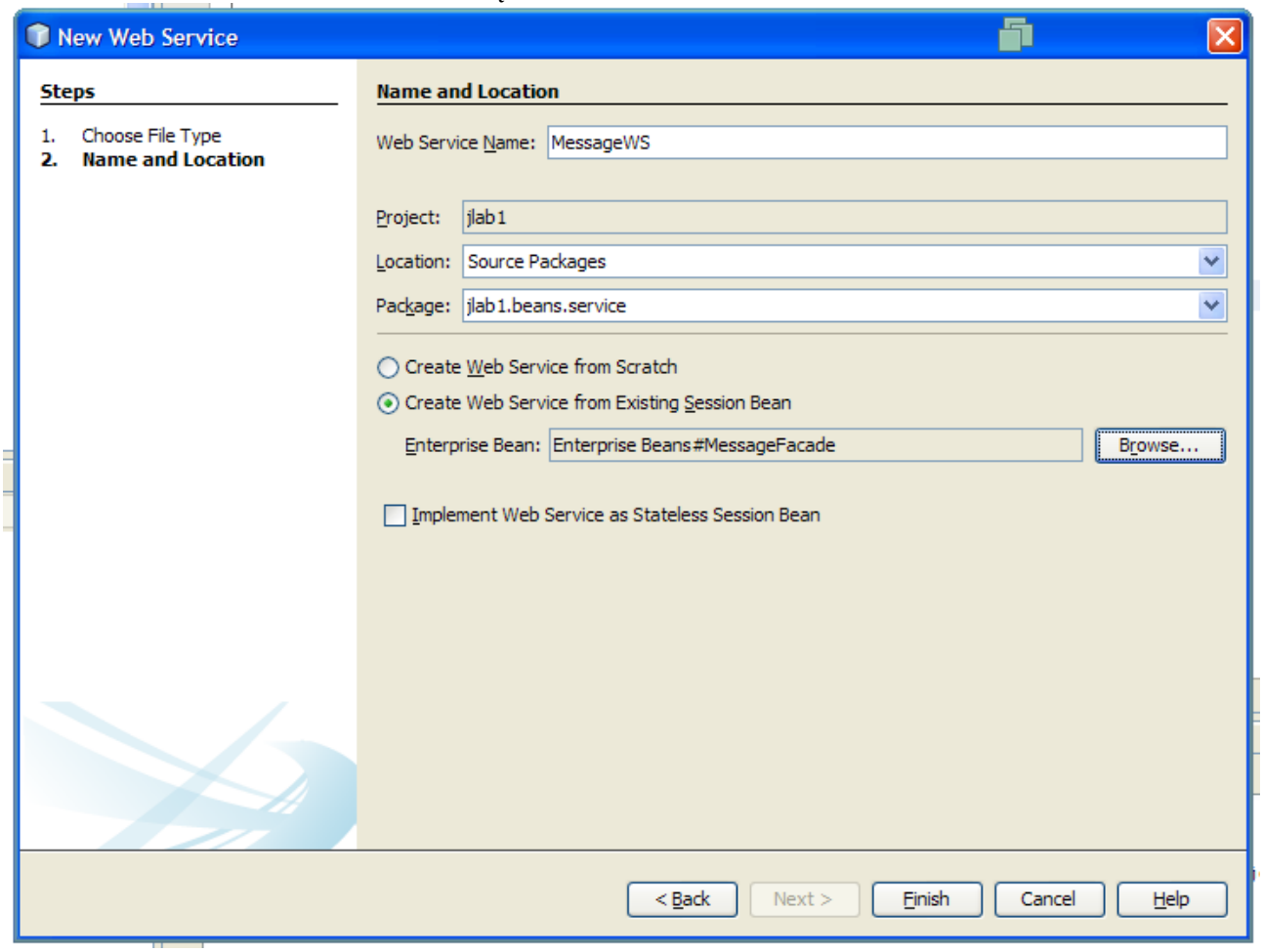
- Kairėje pasirinkite jlab1.entities.Message ir įtraukite į dešinėje esantį sąrašą, Add mygtuku, tada Next ir Finish. Naujai sukurtas Bean leis jax-ws servisui pasiekti pranešimus laikomus duomenų bazėje:



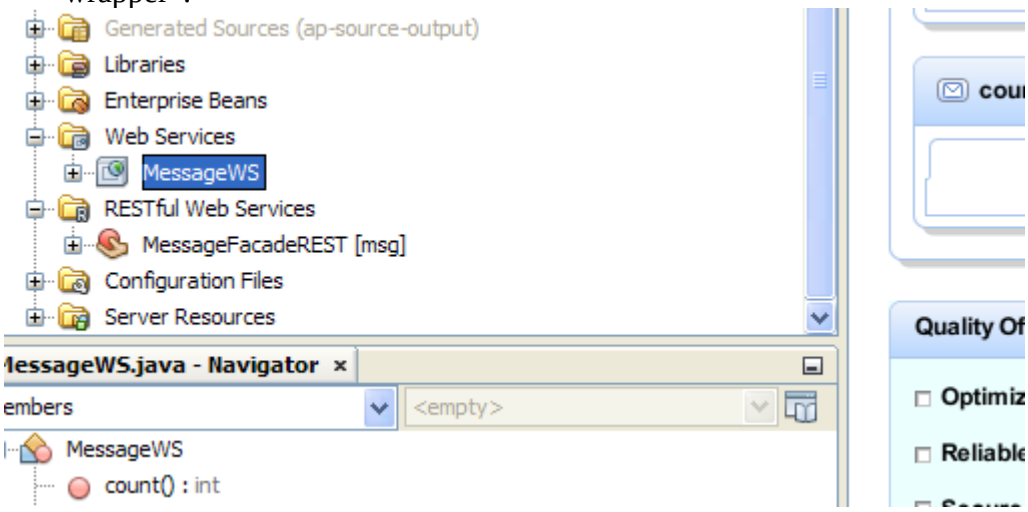
- Sukurkite *Web Service*. *File/New File*, W*eb Services* kategorijoje, *Web Service:*



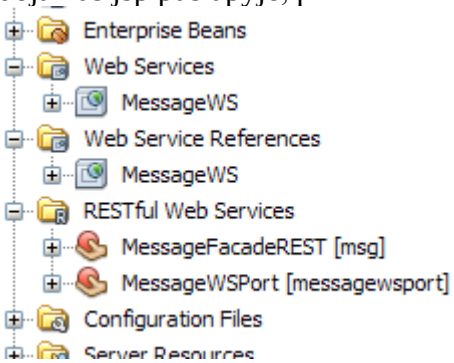
- Suteikite pavadinimą MessageWS, pažymėkite, *Create Web Service from Existing Session Bean*, medyje pasirinkite ***MessageFacade***, ***nesirinkite*** *MessageFacadeREST*, šis bean yra naudojamas jax-rs, pasirinkus šį bean užklausos bus siunčiamos per jį. Užbaikite serviso kūrimą:



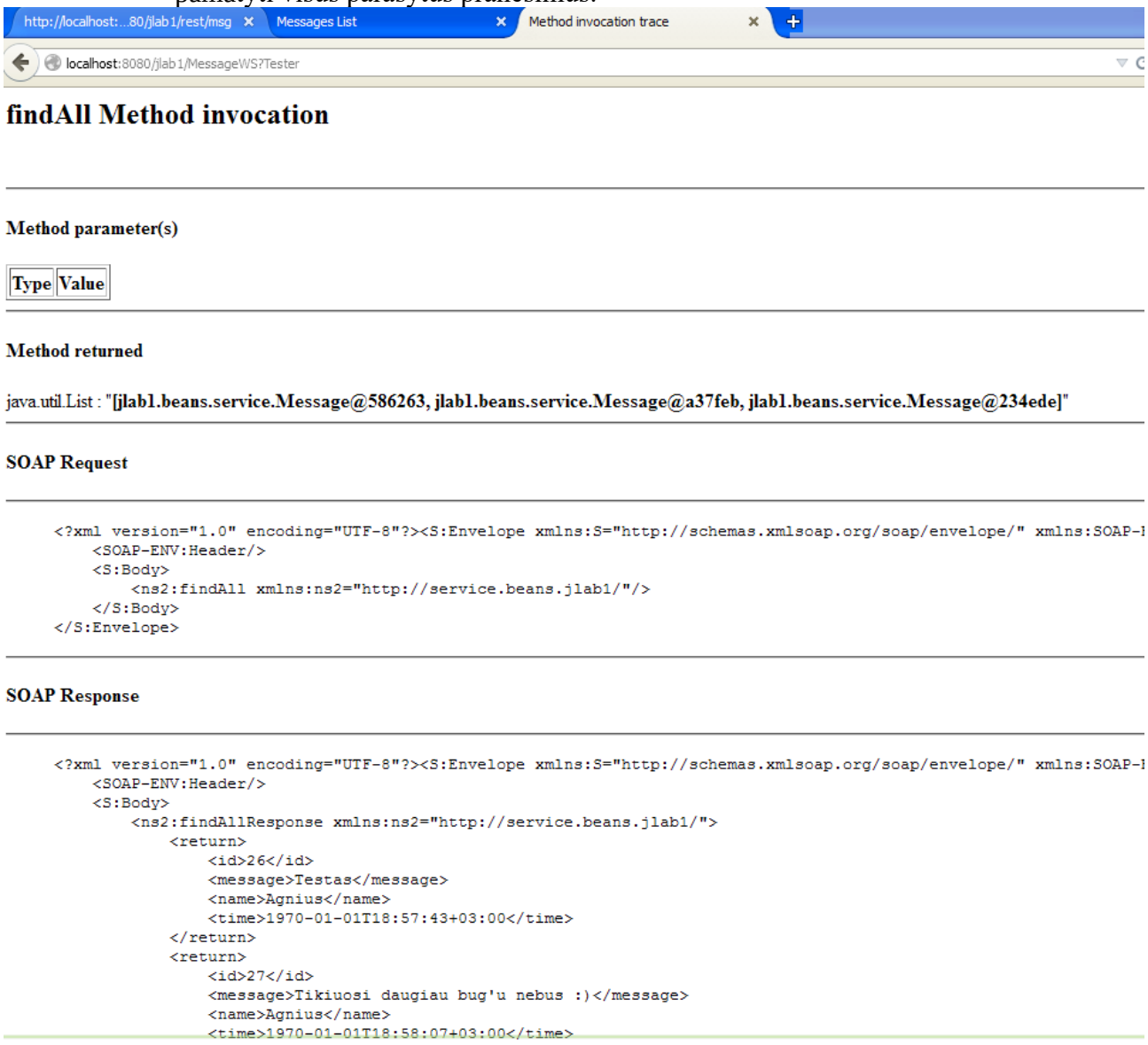
- Norint, kad WSDL (angl. *Web Services Description Language*) servisas būtų pasiekiamas per HTTP, reikia sugeneruoti SOAP-HTTP įvynioklį (angl. wrapper), kuris įgalins SOAP (angl. *Simple Object Access protocol*) užklausų siuntimą iš jsp servlet‘o. Wrapper sugeneruojamas dešiniu pelės mygtukus spustelėjus ant *MessageWS* klasės *Web Services* šakoje ir pasirinkus „Generate SOAP-over-HTTP wrapper“:



- Bus sukurta nauja šaka Web Service References, ir naujas objektas joje MessageWS, kuris bus naudojamas jsp puslapyje, pranešimams pasiekti per soap:

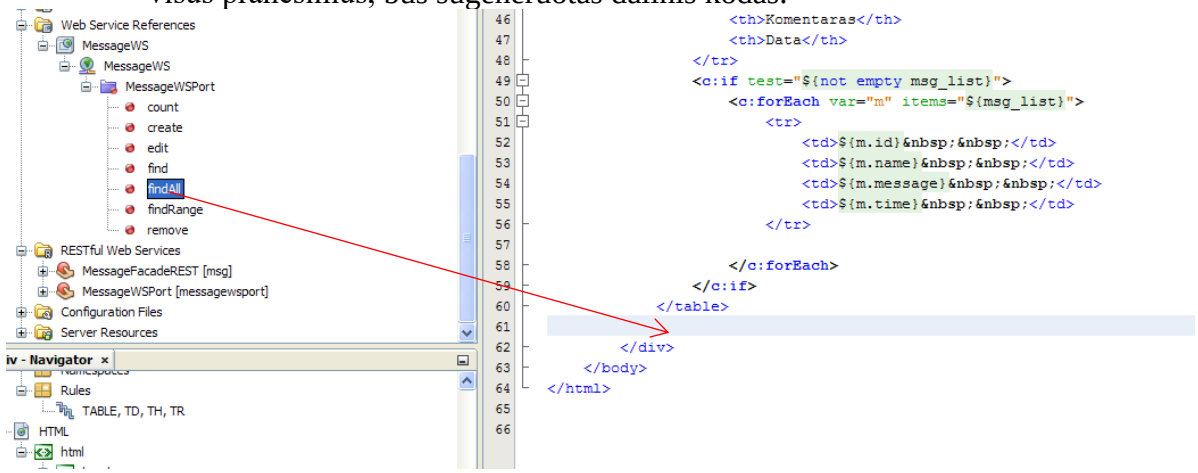


- Sukurtą wsdl servisą dalinai galite išbandyti ant jo paspaudę dešiniu pelės klavišu *Test web service*. Atsidariusiame naršyklės lange galite paspausti *findAll*, turėtumėte pamatyti visus parašytus pranešimus:



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. Kodo pavyzdys pateiktas Priede 1.
- Vartotojo sąsają modifikuosime taip kad pranešimai būtų rodomi ir juos užkraunant per wsdl paslaugą. Išskleiskite *Web Service References/MessageWS/MessageWS/MessageWSPort* medyje turėtumėte matyti metodus kuriuos galima panaudoti pranešimų valdymui. Atsidarę namai.jsp, kairiu pelės klavišu nutempus metodą, *findAll* į vietą kurioje norite matyti visus pranešimus, bus sugeneruotas dalinis kodas:



- Sugeneruotą kodą reikia papildyti sekančiomis eilutėmis, norint atspausdinti pranešimo autorių ir pranešimą.

```
for( int i=0; i < result.size(); i++) {
            jlab1.beans.services_client.Message tmsg = result.get(i);
            out.println(tmsg.getName()+"-"+tmsg.getMessage()+"<br>");
}
```

Savarankiškų užduočių variantai

1. Žinutės užkraunamos naudojant JAX-WS, realizuoti žinučių trynimą JAX-RS serviso pagalba.
2. Žinutės užkraunamos naudojant JAX-RS, realizuoti žinučių trynimą su JAX-RS.
3. Žinutės užkraunamos naudojant JAX-RS, naujos žinutės sukuriamos naudojant JAX-RS servisą.
4. Žinutės užkraunamos naudojant JAX-WS, realizuoti žinučių redagavimą su JAX-RS.
5. Žinutės užkraunamos naudojant JAX-RS, realizuoti žinučių redagavimą su JAX-WS.
6. Žinutės užkraunamos naudojant JAX-RS, realizuoti žinučių redagavimą su JAX-RS.

Priedas 1

```
<%--
    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>
                </div>
            </div>
        </body>
    </html>
```