

Developing Java EE applications with NetBeans and Payara



Agenda

- Java EE is not last millennium's J2EE
- What is Payara ?
- Code Demo
- Wrap Up



Java EE 7 is not J2EE

- For the old Guys
- Java EE is NOT XML Driven
- Java EE IS POJOs and Annotations
- EJBs do NOT need complex Development
- Java EE provides Dependency Injection
- Java EE is lightweight

NetBeans has full Java EE 7 support out of the box



What is Payara

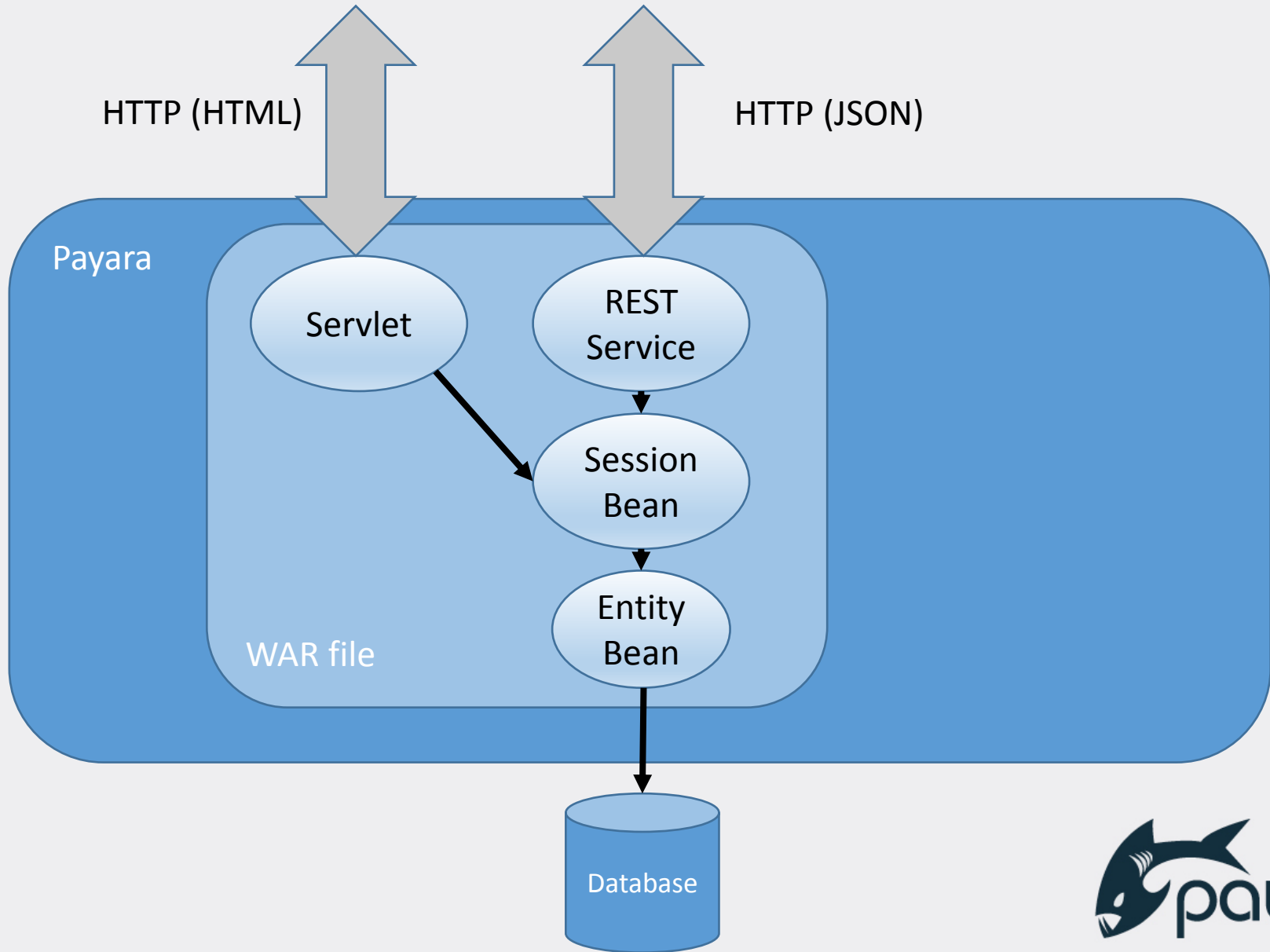
Payara is 24/7 software support for GlassFish Server Open Source Edition.

Payara Server is a drop in replacement for GlassFish Server with guaranteed quarterly releases. We provide enhancements, bug fixes and patches to upstream GlassFish Server and dependent libraries including Tyrus, Eclipse Link, Jersey and others.

Payara Server Developed on GitHub
<https://github.com/payara/Payara>.



Contact Database (Code Demo)



Java EE 7 is not J2EE

- For the old Guys
- Java EE is NOT XML Driven
- Java EE IS POJOs and Annotations
- EJBs do NOT need complex Development
- Java EE provides Dependency Injection
- Java EE is lightweight

NetBeans has full Java EE 7 support out of the box



WTF?

New Project

Steps

1. Choose Project

2. ...

Choose Project

Filter:

Categories:

Java

JavaFX

Java Web

Java EE

HTML5

Java ME Embedded

Java Card

Maven

PHP

Groovy

C/C++

NetBeans Modules

Samples

Projects:

Web Application

Web Application with Existing Sources

Web Free-Form Application

Description:

Creates an empty Web application in a standard IDE project. A standard project uses an IDE-generated build script to build, run, and debug your project.

< Back

Next >

Finish

Cancel

Help

The logo for Payara, featuring a stylized blue fish icon to the left of the word "payara" in a lowercase, sans-serif font.

New Web Application

Steps

1. Choose Project
- 2. Name and Location**
3. Server and Settings
4. Frameworks

Name and Location

Project Name:

Project Location:

[Browse...](#)

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

[Browse...](#)

Different users and projects can share the same compilation libraries (see Help for details).

[< Back](#)

[Next >](#)

[Finish](#)

[Cancel](#)

[Help](#)



New Web Application

Steps

1. Choose Project
2. Name and Location
3. Server and Settings
- 4. Frameworks**

Frameworks

Select the frameworks you want to use in your web application.

- ☐ Spring Web MVC
- ☒ **JavaServer Faces**
- ☐ Struts 1.3.10
- ☐ Hibernate 4.3.1

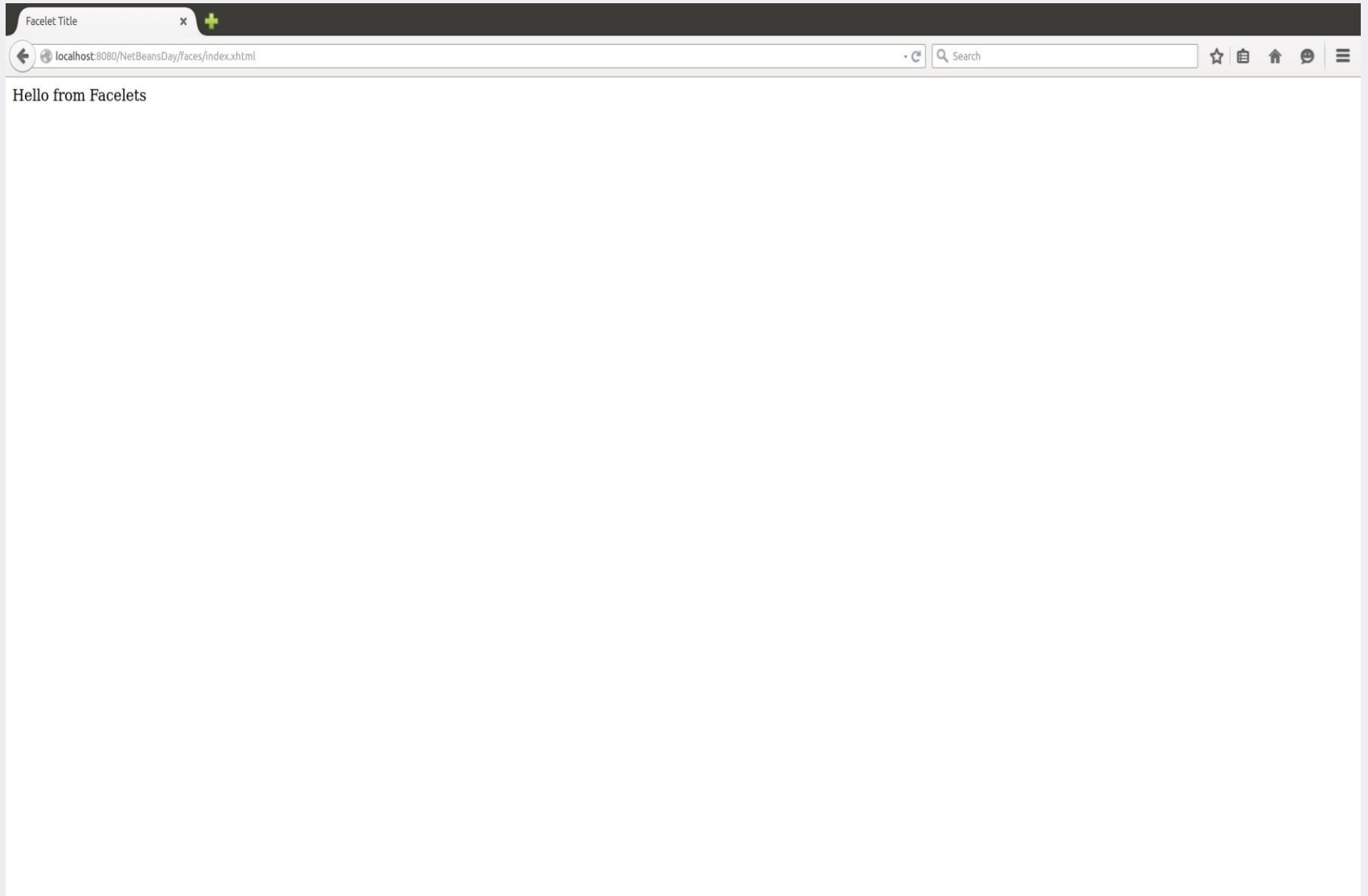
JavaServer Faces Configuration

Libraries Configuration Components

- ☒ **Server Library:** JSF 2.2
- ☐ Registered Libraries: JSF 2.2
- ☐ Create New Library
 - JSF Folder or JAR:
 - Library Name:

< Back Next > **Finish** Cancel Help







New Entity Class

Steps

1. Choose File Type
- 2. Name and Location**
3. Provider and Database

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

Primary Key Type:



☒ Create Persistence Unit

< Back

Next >

Finish

Cancel

Help



New File

Steps

1. Choose File Type
2. Name and Location
- 3. Provider and Database**

Provider and Database

Persistence Unit Name:

Specify the persistence provider and database for entity classes.

Persistence Provider:

Data Source:

☒ Use Java Transaction APIs

Table Generation Strategy: ☒ Create ☐ Drop and Create ☐ None

< Back

Next >

Finish

Cancel

Help



File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+F)

Properties - NetBeansDay (src) Files Services

- NetBeansDay
 - Web Pages
 - WEB-INF
 - index.xhtml
 - Source Packages
 - fish.payara.netbeansday
 - ContactEntity.java
 - Libraries
 - Configuration Files

Source | History

```

21 import javax.persistence.Entity;
22 import javax.persistence.GeneratedValue;
23 import javax.persistence.GenerationType;
24 import javax.persistence.Id;
25
26 /**
27  *
28  * @author steve
29  */
30 @Entity
31 public class ContactEntity implements Serializable {
32     private static final long serialVersionUID = 1L;
33     @Id
34     @GeneratedValue(strategy = GenerationType.AUTO)
35     private Long id;
36
37     private String firstName;
38     private String lastName;
39     private String email;
40     private String telephoneNumber;
41
42     public Long getId() {
43         return id;
44     }
45
46     public void setId(Long id) {
47         this.id = id;
48     }
49
50
51
52 @Override

```

fish.payara.netbeansday.ContactEntity

Members

- ContactEntity: Serializable
 - equals(Object object): boolean
 - getId(): Long
 - hashCode(): int
 - setId(Long id)
 - toString(): String
 - email: String
 - firstName: String
 - id: Long
 - lastName: String
 - serialVersionUID: long
 - telephoneNumber: String

Build (Payara Micro Edition) x NetBeansDay (run) x Java DB Database Process x Payara x

```

INFO: visiting unvisited references
Info: Initializing Mojarra 2.2.10 ( 20150205-0906 https://svn.java.net/svn/mojarra-svn/tags/2.2.10@14334) for context '/NetBeansDay'
Info: Loading application [NetBeansDay] at [/NetBeansDay]
Info: NetBeansDay was successfully deployed in 848 milliseconds.
Info: visiting unvisited references
Info: visiting unvisited references
Info: visiting unvisited references
Info: Initializing Mojarra 2.2.10 ( 20150205-0906 https://svn.java.net/svn/mojarra-svn/tags/2.2.10@14334) for context '/NetBeansDay'
Info: Loading application [NetBeansDay] at [/NetBeansDay]
Info: NetBeansDay was successfully deployed in 984 milliseconds.

```

37:1/4:120 INS 1491 MB




Generate Getters and Setters

Select fields to generate getters and setters for:


▼

☒




ContactEntity

☒




email : String

☒




firstName : String

☒



lastName : String

☒



telephoneNumber : String

☐ Encapsulate Fields

Generate

Cancel

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+F)

Projects - NetBeansDay-UK X1 Files Services

- NetBeansDay
 - Web Pages
 - WEB-INF
 - index.xhtml
 - Source Packages
 - fish.payara.netbeansday
 - ContactEntity.java
 - Libraries
 - Configuration Files

index.xhtml * Facelet Title * ContactEntity.java *

Source History

```

18 package fish.payara.netbeansday;
19
20 import java.io.Serializable;
21 import javax.persistence.Entity;
22 import javax.persistence.GeneratedValue;
23 import javax.persistence.GenerationType;
24 import javax.persistence.Id;
25 import javax.persistence.NamedQueries;
26 import javax.persistence.NamedQuery;
27
28 /**
29  *
30  * @author steve
31  */
32 @Entity
33 @NamedQueries({
34     @NamedQuery(name = "Contact.findAll", query = "SELECT c from ContactEntity c")
35 })
36 public class ContactEntity implements Serializable {
37     private static final long serialVersionUID = 1L;
38     @Id
39     @GeneratedValue(strategy = GenerationType.AUTO)
40     private Long id;
41
42     private String firstName;
43     private String lastName;
44     private String email;
45     private String telephoneNumber;

```

Members

<empty>

ContactEntity: Serializable

- equals(Object object): boolean
- getEmail(): String
- getFirstName(): String
- getId(): Long
- getLastName(): String
- getTelephoneNumber(): String
- hashCode(): int
- setEmail(String email)
- setFirstName(String firstName)
- setId(Long id)
- setLastName(String lastName)
- setTelephoneNumber(String telephoneNumber)
- toString(): String
- email: String
- firstName: String

fish.payara.netbeansday.ContactEntity

Build (Payara Micro Edition) * NetBeansDay (run) * Java DB Database Process * Payara *

Info: visiting unvisited references

Info: visiting unvisited references

Info: Initializing Mojarra 2.2.10 (20150205-0906 https://svn.java.net/svn/mojarra-svn/tags/2.2.10@14334) for context '/NetBeansDay'

Info: Loading application [NetBeansDay] at [/NetBeansDay]

Info: NetBeansDay was successfully deployed in 848 milliseconds.

Info: visiting unvisited references

Info: visiting unvisited references

Info: visiting unvisited references

Info: Initializing Mojarra 2.2.10 (20150205-0906 https://svn.java.net/svn/mojarra-svn/tags/2.2.10@14334) for context '/NetBeansDay'

Info: Loading application [NetBeansDay] at [/NetBeansDay]

Info: NetBeansDay was successfully deployed in 984 milliseconds.

VM Allocated Memory: 1934 MB
Maximum Memory: 1934 MB
Free Memory within the VM: 697.6 MB
Used Memory: 1237 MB

32:1/4:109 INS 1237 MB





New Session Bean

Steps

1. Choose File Type
- 2. Name and Location**

Name and Location

EJB Name:

Project:

Location:

Package:

Session Type:

- ☒ Stateless
- ☐ Stateful
- ☐ Singleton

Create Interface:

- ☐ Local

< Back

Next >

Finish

Cancel

Help



File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+F)

index.xhtml Facelet Title ContactEntity.java ContactsSessionBean.java persistence.xml

Source History

```

26 *
27 * @author steve
28 */
29 @Stateless
30 public class ContactsSessionBean {
31
32     @PersistenceContext
33     EntityManager em;
34
35     public void save(ContactEntity ce) {
36         em.persist(ce);
37     }
38
39     public List<ContactEntity> getAll() {
40         return em.createNamedQuery("Contact.findAll", ContactEntity.class).getResultList();
41     }
42
43     public void populateTestData() {
44         ContactEntity ce = new ContactEntity();
45         ce.setEmail("info@payara.co");
46         ce.setFirstName("Payara");
47         ce.setLastName("BadAssFish");
48         ce.setTelephoneNumber("12345678");
49         em.persist(ce);
50
51         ContactEntity ce2 = new ContactEntity();
52         ce2.setEmail("sales@payara.co");
53         ce2.setFirstName("Fred");
54         ce2.setLastName("Flash");
55         ce2.setTelephoneNumber("12345678");
56         em.persist(ce);
57     }
58
59     // Add business logic below. (Right-click in editor and choose
60     // "Insert Code > Add Business Method")
61 }
62

```

fish.payara.netbeansdev.ContactsSessionBean > populateTestData > ce >

Output Test Results Search Results Notifications Usages Navigator

VM Allocated Memory: 1986 MB
Maximum Memory: 1986 MB
Free Memory within the VM: 821.8 MB
Used Memory: 1165 MB

44:48 INS 1165 MB



New Servlet

Steps

1. Choose File Type

2. Name and Location

3. Configure Servlet Deployment

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back

Next >

Finish

Cancel

Help



File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

index.xhtml * Facelet Title * ContactEntity.java * ContactsSessionBean.java * persistence.xml * PopulateDB.java *

Source History

```

28
29 /**
30  *
31  * @author steve
32  */
33 @WebServlet(name = "PopulateDB", urlPatterns = {"/PopulateDB"})
34 public class PopulateDB extends HttpServlet {
35
36     @EJB
37     ContactsSessionBean bean;
38
39     /**
40      * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
41      * methods.
42      *
43      * @param request servlet request
44      * @param response servlet response
45      * @throws ServletException if a servlet-specific error occurs
46      * @throws IOException if an I/O error occurs
47      */
48     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
49         throws ServletException, IOException {
50         response.setContentType("text/html;charset=UTF-8");
51         try (PrintWriter out = response.getWriter()) {
52             /* TODO output your page here. You may use following sample code. */
53             out.println("<!DOCTYPE html>");
54             out.println("<html>");
55             out.println("<head>");
56             out.println("<title>Servlet PopulateDB</title>");
57             out.println("</head>");
58             out.println("<body>");
59             out.println("<h1>Servlet PopulateDB at " + request.getContextPath() + "</h1>");
60             bean.populateTestData();
61             out.println("Populated DB data");
62             out.println("</body>");
63             out.println("</html>");
64         }
65     }
66
67     HttpServlet methods. Click on the + sign on the left to edit the code.
105
106 }
107

```

fish.pavara.netbeansday.PopulateDB > processRequest > try >

Output Test Results Search Results Notifications Usages Navigator

VM Allocated Memory: 1910 MB
Maximum Memory: 1910 MB
Free Memory within the VM: 1000 MB
Used Memory: 910.0 MB

61:40 INS 910.0 MB





Set Servlet Execution URI

Select servlet execution URI, optionally add some request parameters :
e.g. `/flowerServlet?flower=rose&color=red`



OK

Cancel

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+g)

index.xhtmlServlet PopulateDBContactEntity.javaContactsSessionBean.jspersistence.xmlPopulateDB.java

http://localhost:8080/NetBeansDay/PopulateDB

Servlet PopulateDB at /NetBeansDay

Populated DB data

Projects - NetBeansDay-UKFilesServices

OutputTest ResultsSearch ResultsNotificationsUsagesNavigator

VM Allocated Memory: 1862 MB
Maximum Memory: 1862 MB
Free Memory within the VM: 1136 MB
Used Memory: 726.2 MB

726.2 MB




New RESTful Web Services from Patterns

Steps

1. Choose File Type

2. Select Pattern

3. Specify Resource Classes



Select Pattern

Select a RESTful web service design pattern:

☒ Simple Root Resource

☐ Container-Item

☐ Client-Controlled Container-Item

Description:

Create a RESTful root resource class with GET and PUT methods using Java API for RESTful Web Service (JSR-311). This pattern is useful for creating a simple HelloWorld service and wrapper services

< Back

Next >

Finish

Cancel

Help

New RESTful Web Services from Patterns

Steps

1. Choose File Type
2. Select Pattern
3. **Specify Resource Classes**

Specify Resource Classes

Project: NetBeansDay

Location: Source Packages

Resource Package: fish.payara.netbeansday

Path: contact

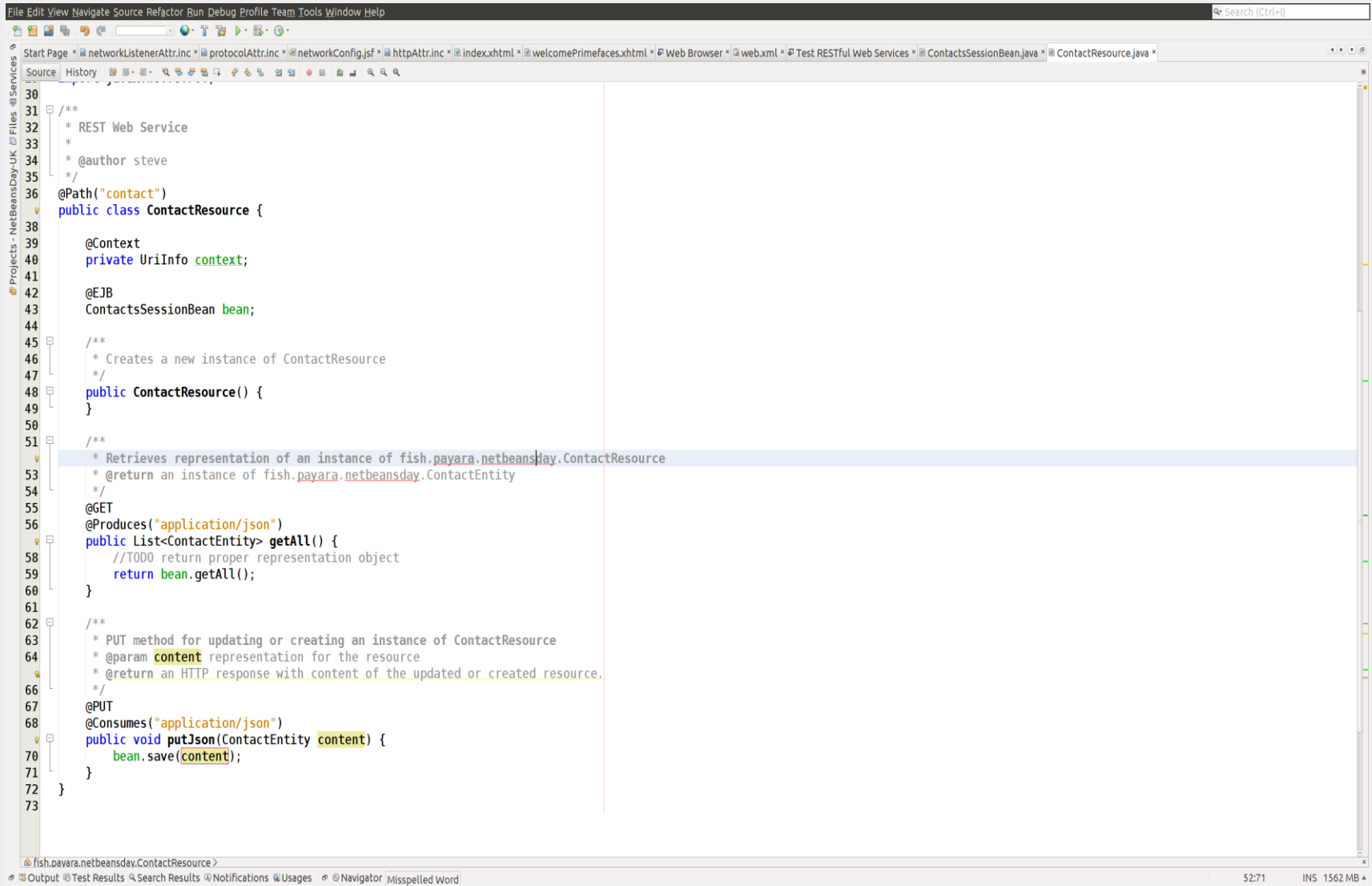
Class Name: ContactResource

MIME Type: application/json

Representation Class: fish.payara.netbeansday.ContactEntity

❗ Class already exists





Configure REST Test Client

☐ Locally Generated Test Client (suitable when cross-domain access to REST resources is enabled)

☒ Web Test Client in Project

Target Folder:

Notice:

To access REST resources deploy the test client to the same domain as the resource (Web Test Client in Project option).

If your browser permits access to remote resources, you can choose your local filesystem as the target (Locally Generated Test Client option).

In Internet Explorer, you can enable cross-domain requests in Internet Options dialog box.

Another option to allow cross-domain requests, to REST resources provided by the web application, is to set up a Cross-Origin Resource Sharing (CORS) filter.

To create CORS filter, you can use Web Services -> Cross-Origin Resource Sharing Filter wizard.

OK

Cancel



Test RESTful Web Services

localhost:8080/NetBeansDay/test-resbeans.html

WADL : http://localhost:8080/NetBeansDay/webresources/application.wadl

Test RESTful Web Services

NetBeansDay

contact

NetBeansDay > contact

Resource: contact
(http://localhost:8080/NetBeansDay/webresources/contact)

Choose method to test: GET(application/json) Test

Custom Request Headers

Status: 200 (OK)

Response:

Tabular View Raw View Sub-Resource Headers Http Monitor

[{"email":"info@navara.co","firstName":"Pavara","id":1,"lastName":"BadAssFish","telephoneNumber":"12345678"}]



Test RESTful Web Services

localhost:8080/NetBeansDay/test-resbeans.html

WSDL : <http://localhost:8080/NetBeansDay/webresources/application.wadl>

Test RESTful Web Services

NetBeansDay

contact

NetBeansDay > contact

Resource: [contact](#)
(<http://localhost:8080/NetBeansDay/webresources/contact>)

Choose method to test: [PUT\(application/json\)](#) [Test](#)

Content:

```
{ "email": "sales@payara.co", "firstName": "Fred", "lastName": "FlintStone", "telephoneNumber": "#####"} 
```

[↓ Custom Request Headers](#)



Test RESTful Web Services

localhost:8080/NetBeansDay/test-resbeans.html

Search

Status: 200 (OK)

Response:

Tabular ViewRaw ViewSub-ResourceHeadersHttp Monitor

```
[{"email":"info@payara.co","firstName":"Payara","id":1,"lastName":"BadAssFish","telephoneNumber":"12345678"}, {"email":"info@payara.co","firstName":"Payara","id":51,"lastName":"BadAssFish","telephoneNumber":"12345678"}, {"email":"sales@payara.co","firstName":"Fred","id":52,"lastName":"Flash","telephoneNumber":"12345678"}, {"email":"info@payara.co","firstName":"Steve","id":101,"lastName":"Millidge","telephoneNumber":"11111111"}, {"email":"sales@payara.co","firstName":"Fred","id":102,"lastName":"FlintStone","telephoneNumber":"#####"}]
```



```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Start Page networkListenerAttr.inc protocolAttr.inc networkConfig.jsf httpAttr.inc index.xhtml welcomePrimefaces.xhtml Web Browser web.xml Test RESTful Web Services ContactsSessionBean.java ContactResource.java
Source History
29 */
30 @Stateless
31 public class ContactsSessionBean {
32
33     @PersistenceContext
34     EntityManager em;
35
36     public void save(ContactEntity ce) {
37         em.persist(ce);
38     }
39
40     public List<ContactEntity> getAll() {
41         return em.createNamedQuery("Contact.findAll", ContactEntity.class).getResultList();
42     }
43
44     public ContactEntity get(long ID) {
45         return em.find(ContactEntity.class, ID);
46     }
47
48     public void populateTestData() {
49         ContactEntity ce = new ContactEntity();
50         ce.setEmail("info@payara.co");
51         ce.setFirstName("Payara");
52         ce.setLastName("BadAssFish");
53         ce.setTelephoneNumber("12345678");
54         em.persist(ce);
55
56         ContactEntity ce2 = new ContactEntity();
57         ce2.setEmail("sales@payara.co");
58         ce2.setFirstName("Fred");
59         ce2.setLastName("Flash");
60         ce2.setTelephoneNumber("12345678");
61         em.persist(ce2);
62     }
63 }
64
65
fish.payara.netbeansday.ContactsSessionBean > populateTestData >
Output Test Results Search Results Notifications Usages Navigator
51:35 INS 1226 MB
```



```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+F)

Start Page * networkListenerAttr.inc * protocolAttr.inc * networkConfig.jsf * httpAttr.inc * index.xhtml * welcomePrimefaces.xhtml * Web Browser * web.xml * Test RESTful Web Services * ContactsSessionBean.java * ContactResource.java *

Source History

Projects - NetBeansDay-UK
Files
Services
Start Page
networkListenerAttr.inc
protocolAttr.inc
networkConfig.jsf
httpAttr.inc
index.xhtml
welcomePrimefaces.xhtml
Web Browser
web.xml
Test RESTful Web Services
ContactsSessionBean.java
ContactResource.java

34 * @author steve
35 */
36 @Path("contact")
37 public class ContactResource {
38
39     @Context
40     private UriInfo context;
41
42     @EJB
43     ContactsSessionBean bean;
44
45     /** Creates a new instance of ContactResource ...3 lines */
46     public ContactResource() {
47     }
48
49     /** Retrieves representation of an instance of fish ...4 lines */
50     @GET
51     @Produces("application/json")
52     public List<ContactEntity> getAll() {
53         //TODO return proper representation object
54         return bean.getAll();
55     }
56
57     @GET
58     @Path("/{contactid}")
59     @Produces("application/json")
60     public ContactEntity getContact(@PathParam("contactid") long ID) {
61         //TODO return proper representation object
62         return bean.get(ID);
63     }
64
65     /** PUT method for updating or creating an instance of ContactResource ...5 lines */
66     @PUT
67     @Consumes("application/json")
68     public void putJson(ContactEntity content) {
69         bean.save(content);
70     }
71 }
72
73 fish.payara.netbeansdav.ContactResource > bean >
```

