POLITECNICO DI MILANO

Software Engineering 2 Project

Code Inspection Document

Simone Guidi

Matteo Imberti

January 5, 2016

Contents

1 Assigned Class

1.1 Class

- Name: WebServiceEndpoint
- Location: appserver/deployment/dol/src/main/java/com/sun/enterprise/deployment/

1.2 Methods

- resolveComponentLink( ) (Line 467)

```
1   /**
2        * Convert the contents of the ejb-link or servlet-
             link element to
3        * an object representing the implementation
             component.
4        */
5       public boolean resolveComponentLink() {
6           boolean resolved = false;
7           if( ejbLink != null ) {
8               EjbBundleDescriptor ejbBundle = getEjbBundle
                    ();
9               if( ejbBundle.hasEjbByName(ejbLink) ) {
10                  resolved = true;
11                  EjbDescriptor ejb = ejbBundle.
                        getEjbByName(ejbLink);
12                  setEjbComponentImpl(ejb);
13              }
14          } else if( webComponentLink != null ) {
15              WebBundleDescriptor webBundle = getWebBundle
                    ();
16              WebComponentDescriptor webComponent =
17                      (WebComponentDescriptor) webBundle.
18                              getWebComponentByCanonicalName
                                    (webComponentLink);
19              if( webComponent != null ) {
20                  resolved = true;
21                  setWebComponentImpl(webComponent);
22              }
23          }
24
25          return resolved;
26      }
```

- getEndpointAddressPath(String cr) (Line 706)

```
1  /**
2       * return the endpoint address path (without http://<
            host>:<port>)
3       * used to make web service invocations on this
            endpoint .
4       */
5      private String getEndpointAddressPath(String cr) {
6          String uri = null;
7          // Compose file portion of URL depending on
               endpoint type.
8          // The file portion of the URL MUST have a single
                leading slash.
9          // Note that the context root and the endpoint
               address uri strings
10         // from the descriptors may or may not each have
               a leading slash.
11         if( implementedByWebComponent() ) {
12             if (endpointAddressUri == null) {
13                 updateServletEndpointRuntime();
14             }
15
16             // for servlets, endpoint address uri is
                   relative to
17             // web app context root.
18             WebBundleDescriptor webBundle =
19                     webComponentImpl.
                           getWebBundleDescriptor();
20             String contextRoot = (cr == null)?webBundle.
                   getContextRoot():cr;
21
22             if( contextRoot != null ) {
23                 if( !contextRoot.startsWith("/") ) {
24                     contextRoot = "/" + contextRoot;
25                 }
26
27                 uri = contextRoot +
28                         (endpointAddressUri.startsWith("/
                               ") ?
29                                 endpointAddressUri : ("/"
                                       + endpointAddressUri)
                                   );
30             }
31         } else {
```

```
32              //If implemented by EJB Component, it will
                    have a default standard endpointAddressUri
33              return getEndpointAddressUri();
34          }
35          return uri;
36      }
```

- matchesEjbPublishRequest(String requestUriRaw , String query) (Line 777)

```
1   /**
2        * Checks an ejb request uri to see if it represents
             a legal request
3        * for the wsdl content associated with this endpoint
             's web service.
4        * Equivalent matching for servlets is performed
             automatically by the
5        * web server.  Should only be called for HTTP(S) GET
             .
6        */
7       public boolean matchesEjbPublishRequest(String
            requestUriRaw, String query)
8       {
9           // Strip off leading slash.
10          String requestUri = (requestUriRaw.charAt(0) == '
                /') ?
11                  requestUriRaw.substring(1) :
                        requestUriRaw;
12
13          boolean matches = false;
14
15          // If request of form http<s>://<host>:<port>/<
                endpoint-address>?WSDL
16          if( query != null ) {
17              String toMatch = (endpointAddressUri.charAt
                    (0) == '/') ?
18                      endpointAddressUri.substring(1) :
                            endpointAddressUri;
19              matches = requestUri.equals(toMatch) &&
20                      (query.equalsIgnoreCase("WSDL") ||
21                              query.startsWith("xsd=") ||
22                              query.startsWith("wsdl="));
23          } else {
24              // Add trailing slash to make sure sub
                    context is an exact match.
```

```
25              String publishingUri = getPublishingUri() + "
                    /";
26              matches = requestUri.startsWith(publishingUri
                    );
27          }
28
29          return matches;
30      }
```

- getWsdlContentPath(String requestUri) (Line 808)

```
1       /**
2        *@return the portion of a request uri that
             represents the location
3        * of wsdl content within a module or null if this
             request is invalid.
4        * Returned value does not have leading slash.
5        */
6       public String getWsdlContentPath(String requestUri) {
7
8           // Strip off leading slash.
9           String uri = (requestUri.charAt(0) == '/') ?
                requestUri.substring(1) :
10                   requestUri;
11
12          // get "raw" internal publishing uri.  this value
13          // does NOT have a leading slash.
14          String publishingUriRaw = getPublishingUri();
15
16          // Construct the publishing root.  This should
                NOT have a
17          // leading slash but SHOULD have a trailing slash
                .
18          String publishingRoot = null;
19
20          if( implementedByWebComponent() ) {
21              WebBundleDescriptor webBundle =
22                      webComponentImpl.
                            getWebBundleDescriptor();
23              String contextRoot = webBundle.getContextRoot
                    ();
24              if( contextRoot.startsWith("/") ) {
25                  contextRoot = contextRoot.substring(1);
26              }
27              publishingRoot = contextRoot + "/" +
```

```
28                      publishingUriRaw + "/";
29          } else {
30              publishingRoot = publishingUriRaw + "/";
31          }
32
33          String wsdlPath = uri.startsWith(publishingRoot)
                ?
34                  uri.substring(publishingRoot.length()) :
                        null;
35
36          return wsdlPath;
37      }
```

- updateServletEndpointRuntime( ) (Line 980)

```
1   private void updateServletEndpointRuntime() {
2
3           //An endpoint might have been loaded off a jar
                file. In that case the WebFragmentDescriptor
                can be stale. So patch it
4           WebComponentDescriptor wc = ((WebBundleDescriptor
                )webService.getBundleDescriptor()).
                getWebComponentByCanonicalName(
                webComponentImpl.getCanonicalName());
5           if(!(wc == webComponentImpl)){
6               setWebComponentImpl(wc);
7           }
8
9           // Copy the value of the servlet impl bean class
                into
10          // the runtime information.  This way, we'll
                still
11          // remember it after the servlet-class element
                has been
12          // replaced with the name of the container's
                servlet class.
13          saveServletImplClass();
14
15          WebBundleDescriptor bundle = webComponentImpl.
                getWebBundleDescriptor();
16
17          WebServicesDescriptor webServices = bundle.
                getWebServices();
18          Collection endpoints =
19                  webServices.getEndpointsImplementedBy(
                        webComponentImpl);
```

```
20
21          if( endpoints.size() > 1 ) {
22              String msg = "Servlet " + getWebComponentLink
                    () +
23                      " implements " + endpoints.size() + "
                            web service endpoints " +
24                      " but must only implement 1";
25              throw new IllegalStateException(msg);
26          }
27
28          if( getEndpointAddressUri() == null ) {
29              Set urlPatterns = webComponentImpl.
                    getUrlPatternsSet();
30              if( urlPatterns.size() == 1 ) {
31
32                  // Set endpoint-address-uri runtime info
                        to uri.
33                  // Final endpoint address will still be
                        relative to context root
34                  String uri = (String) urlPatterns.
                        iterator().next();
35                  setEndpointAddressUri(uri);
36
37                  // Set transport guarantee in runtime
                        info if transport
38                  // guarantee is INTEGRAL or
                        CONDIFIDENTIAL for any
39                  // security constraint with this url-
                        pattern.
40                  Collection constraints =
41                          bundle.
                                getSecurityConstraintsForUrlPattern
                                (uri);
42                  for(Iterator i = constraints.iterator();
                        i.hasNext();) {
43                      SecurityConstraint next = (
                            SecurityConstraint) i.next();
44
45                      UserDataConstraint dataConstraint =
46                              next.getUserDataConstraint();
47                      String guarantee = (dataConstraint !=
                            null) ?
48                              dataConstraint.
                                    getTransportGuarantee() :
                                    null;
49
```

```
50                      if( (guarantee != null) &&
51                              ( guarantee.equals
52                                      (UserDataConstraint.
                                            INTEGRAL_TRANSPORT
                                            ) ||
53                                      guarantee.equals
54                                              (
                                                    UserDataConstraint
                                                    .
                                                    CONFIDENTIAL_TRANSPORT
                                                    ) ) ) {
55                          setTransportGuarantee(guarantee);
56                          break;
57                      }
58                  }
59              } else {
60                  String msg = "Endpoint " +
                        getEndpointName() +
61                          " has not been assigned an
                              endpoint address " +
62                          " and is associated with servlet
                              " +
63                          webComponentImpl.getCanonicalName
                              () + " , which has " +
64                          urlPatterns.size() + " url
                              patterns";
65                  throw new IllegalStateException(msg);
66              }
67          }
68      }
```

2 Functional Role

2.1 resolveComponentLink( )

The method converts the contents of the ejb-link or servlet-link element to an object representing the implementation component according to which element between ejbLink and webComponent is not null. If the descriptor for the ejb/server element is found in the ejb/web bundle, the element is succesfully converted; if not the convertion fails. The method returns a boolean rapresenting the success (true) or fail (false) of the conversion.

2.2 getEndpointAddressPath(String cr)

The method returns the address path of the endpoint. If the instance of the class implements an ejb component the path is retrieved using another method of the class. Instead, if it implements a servlet it builds it either using the context root provided as a parameter or obtaining it from the servlet descriptor (if the parameter is null) and appending to it the current endpoint address URI.

2.3 matchesEjbPublishRequest(String requestUriRaw , String query)

The method returns true if the parameter requestUriRaw with the query parameter represents a legal request for the wsdl content associated with this endpoint's web service. The parameter query may be null.

2.4 getWsdlContentPath(String requestUri)

The method returns the portion of a request uri that represents the location of wsdl content within a module or null if this request is invalid. Returned value does not have leading slash.

2.5 updateServletEndpointRuntime( )

The method updates the reference to the web component descriptor in case it is old by querying the webService. Then it checks that the web component implements no more than one endpoint and throws an IllegalStateException otherwise. If no endpoint address URI is specified it tries to obtain it from the web component; it may fail (and throw IllegalStateException) if the web component provides a number of URIs different from one. It also checks the security constraints of the URI and saves them if they are either of type INTEGRAL_TRANSPORT or of type CONFIDENTIAL_TRANSPORT.

3 Issues Found

3.1 Class Level

- Line 215, 226 issue 25 E/G: instance variable between methods.
- Line 236 issue 25 F: class constructor after methods

3.2 resolveComponentLink( )

- Issue 23: missing return statement specification in the Javadoc
- Line 481, 482 issue 36: webBundle may be null
- Line 474 issue 36: ejbBundle may be null

- Line 477 issue 36: ejb may be null

3.3 getEndpointAddressPath(String cr)

- Issue 23: missing parameters specification in the Javadoc. The return statement specification is also provided in a non-standard form
- Line 723 issue 36: variable webBundle may be null

3.4 matchesEjbPublishRequest(String requestUriRaw , String query)

- Issue 23 missing return statement and arguments description in the Javadoc
- Line 789 issue 33: variable initialization should be at the beginning of the block

3.5 getWsdlContentPath(String requestUri)

- Line 821, 825 issue 33: variable initialization should be at the beginning of the block

3.6 updateServletEndpointRuntime( )

- Line 983 issue 13-14: line length exceeds 120 characters
- Line 1030, 1032 issue 15: line break occours between a method name and its parameters, not after an operator or a comma
- Issue 23: the javadoc is missing
- Issue 27: this method is very long and does different things so it could be split in two or more methods
- Line 994 issue 36: variable bundle may be null
- Line 996 issue 36: variable webServices may be null
- Line 1008 issue 36: variable urlPatterns may be null
- Line 1019 issue 36: variable constraints may be null
- Line 984 issue 40: two objects are compared using "==" instead of "equals"
- Line 1039 issue 42: the error message to be displayed is not very comprehensible, without checking the code it's difficult to understand if the problem is the fact that no endpoint was assigned or that the servlet has more than on url pattern or both
- Issue 50: the method's exception list is empty but it may throw an IllegalStateException