Create web services for stateless session beans in Rational Software Architect

Learn how to create, publish, and test the web services of stateless session beans

Jinu Mohan

August 29, 2011

Learn how to create web services for stateless EJB session beans in IBM® Rational® Software Architect. This tutorial shows you how to use the easy tooling capabilities of Rational Software Architect to create a web service, and then test it using web services test explorer.

Before you start

About this tutorial
This tutorial describes the different ways of creating web services for EJB stateless session beans in Rational Software Architect, and how to test the web services using the web service test explorer. An EJB sample application is provided.

Objectives
The main objective of this tutorial is to provide details on creating JAX-WS and JAX-RPC web services using Rational tooling.

Prerequisites
You should have a beginner to intermediate level knowledge on EJB 3.0 and web services.

System requirements
To complete the steps in this tutorial, you need:

- Rational Software Architect 7.5 and above
- Websphere Application Server 7.0

EJB stateless session bean as web service

Getting started
1. Import the sample DateFormatter (EJB 3.0) project provided in the downloads.
2. This project has a stateless session bean named DateFormatterService already created in it.
3. The DateFormatterService has the service method which formats the given date object into a String whose format is "EEE, MMM d, yyyy"

Creating the web service

Now we will expose the "formatDate" method in the DateFormatterService bean as a web service. Though JAX-WS is advanced version of Web Service we will look into JAX-RPC version as well.

JAX-WS

1. To create a JAX-WS web service, add an annotation @WebService just above the @Stateless annotation in the DateFormatterService bean. Building the project creates a service under services.

   Figure 1. DateFormatterService bean

2. Add a router to transfer the soap messages over http. Right-click the service that is created under services and select Create router module(EndpointEnabler). In the Create Router Modules window, select EJB Web Service binding as HTTP. Select finish. This creates a DateFormatter_HTTPRouter.
Done! You have created JAX-WS web service for stateless session bean.

**JAX-RPC**

1. After importing the project, right-click DateFormatterService bean, Web Services > Create web service.
2. In the Web Service creation window, provide the following values:
   - Web service type: Bottom up EJB Web Service
   - Service implementation: DateFormatterService
   - Web service runtime: IBM Websphere JAX-RPC
   - Service project: DateFormatter
   - Service EAR project: DateFromatterEAR
3. In the first sidebar, slide till "Assemble service" and in the second, slide to "No client".
4. Click next. In the Web Service EJB configuration select HTTP bindings.
5. Click next. In the Web Service EJB identity window select the method which is to be exposed as web service.
Figure 3. Web Service EJB identity

6. As we are not publishing this service to a UDDI registry, click finish.

Done! You have created JAX-RPC web service for stateless session bean.

Testing the web services

1. After creating the service, you are ready to test the web service using the test explorer, provided the application is added to server. Right-click on the service which is created(DataFormatter > Services > <service name>) and select Test with Web Services Explorer.
2. Once the explorer opens, in the left panel, select the operation(formatDate here) to be executed.
3. Provide the input date and click on Go button.
Figure 5. Invoke operation
## Downloadable resources

<table>
<thead>
<tr>
<th>Description</th>
<th>Name</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample code for this tutorial</td>
<td>Sample_code_DateFormatter.zip</td>
<td>7KB</td>
</tr>
</tbody>
</table>
Related topics

- "Web services hints and tips: JAX-RPC versus JAX-WS, Part 1" (developerWorks, Oct 2006) introduces a series that compares these two Java web services programming models.
- "Web services hints and tips: JAX-RPC versus JAX-WS, Part 2" (developerWorks, Nov 2006) compares the data mappings of these two web services specifications.
- "Design and develop JAX-WS 2.0 web services" (developerWorks, Sept 2007) walks you through how to do all of this and more by developing a sample order-processing application that exposes its functionality as web services. After going through this tutorial, you'll be able to apply these concepts and your newly acquired knowledge to develop web services for your application using JAX-WS technology.