JBoss WS User Guide

Version: 1.0.0.GA

1. JBossWS Runtime Overview	. 1
2. Creating a Web Service using JBossWS runtime	. 3
2.1. Creating a Dynamic Web project	3
2.2. Configure JBoss Web Service facet settings	. 5
2.3. Creating a Web Service from a WSDL document using JBossWS runtime	7
2.4. Creating a Web service from a Java bean using JBossWS runtime	13
3. Creating a Web Service Client from a WSDL Document using JBoss WS	23
4. JBoss WS and development environment	27
4.1. JBossWS Preferences	27
4.2. Default Server and Runtime	31

JBossWS Runtime Overview

JBossWS is a web service framework developed as a part of the JBoss Application Server. It implements the JAX-WS specification that defines a programming model and run-time architecture for implementing web services in Java, targeted at the Java Platform, Enterprise Edition 5 (Java EE 5).

Creating a Web Service using JBossWS runtime

In this chapter we provide you with the necessary steps to create a Web Service using JBossWS runtime.

2.1. Creating a Dynamic Web project

Before creating a web service, you should have a Dynamic Web Project created:

🔄 New Project 🗙	Γ
Select a wizard —	
Create a Dynamic Web project	
<u>W</u> izards:	
type filter text	
 ▷ ➢ Java ▷ ➢ Java EE ▷ ➢ JavaScript 	
 ▷ 🦢 JPA ▷ ➢ Plug-in Development 	
▽ 🗁 Web	
🔯 Dynamic Web Project	
Static Web Project	
Image: Second)

Figure 2.1. Dynamic Web Project

Create a Web project by selecting *New > Project... > Dynamic Web project*. Enter the following information:

- Project Name: enter a project name
- Target runtime: any server depending on your installation. If it is not listed, click New and browse to the location where it is installed to. You may set *Target Runtime* to *None*, in this case, you should add <u>JBoss Web Service facet to the project</u>.

New Dynamic Web Project ×
Dynamic Web Project 🚬
Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.
Project name: test
Project contents:
☑ Use <u>d</u> efault
Directory: /home/grid/workspace/documents/test Browse
Target Runtime
<none> \v New</none>
Dynamic Web Module version
2.4 🗸
Configuration
Default Configuration V Modify
The default configuration provides a good starting point. Additional facets can later be installed to add new functionality to the project.
EAR Membership
Add project to an EAR
EAR Project Name: EAR V New
Image: Second

Figure 2.2. Dynamic Web Project Wizard

• Configure Web Module values:

🔄 New Dynamic Web Project	×
Web Module	
Configure web module settings.	7
Context Root:	
test	
Content Directory:	
WebContent	
Java Source Directory:	
src	
(?) < <u>Back</u> Next > <u>Finish</u> Cancel	

Figure 2.3. Web Module Settings Configuration

Click on the Finish button.

2.2. Configure JBoss Web Service facet settings

If you have already created a new Dynamic Web project, the next step is to add JBoss Web Service facet to the project:

Modify Faceted Project	×
JBossWS Facet	
A JBossWS runtime has not been chosen	
 Server Supplied JBossWS Runtime 	
•	0 New
Package all JBossWS runtime jars into the deployment archive	
② < <u>B</u> ack <u>N</u> ext >	ОК

Figure 2.4. Configure JBoss Web Service Facet

Server Supplied JBossWS Runtime: If you have already set a JBoss runtime to the project's target runtime, you may choose *Server Supplied JBossWS Runtime* and then click *Ok* to finish the configuration of JBoss Web Service facet.

If the project has no *Target Runtime* settings, you should check the second radio button and specify a JBossWS runtime from the list. You also can create a new JBossWS runtime, click on the *New...* button will bring you to another dialog to configure new JBossWS runtime.

Creating a Web Service from a WSDL

document using JBossWS runtime

C New JBossWS Run	time 🗙
JBossWS Runtime	
Create a JBossWS Runtime	
Name:	
Version 🔹	
Home Folder:	Browse
Customize JBoss Web Service runtime jars	
0	Einish Cancel

Figure 2.5. Configure JBossWS Runtime

See how to configure a new JBossWS runtime <u>here</u>:

2.3. Creating a Web Service from a WSDL document using JBossWS runtime

In this chapter we provide you with the necessary steps to create a Web Service from a WSDL document using JBossWS runtime.

At first, please make sure that you have already created a dynamic Web project with JBoss Web Service facet installed.

See how to make it <u>here</u>> and <u>here</u>.

To create a Web Service using JBossWS runtime select *File > New > Other > Web Services > Web Service* to run Web Service creation wizard.

Let's get through the wizard step-by-step:



Figure 2.6. New Web Service Wizard

First, please select Top down Java bean Web Service from the Web Service type list, and select a WSDL document from workspace, click on the Server name lilnk on the page will bring you to another dialog. Here you can specify the server to a JBoss Server and Web Service runtime to JBossWS runtime:

Creating a Web Service from a WSDL

document using J	BossWS runtime
------------------	----------------

Service Deployment Configuration ×
Choose from the list of runtimes and deployment servers, or use the default settings.
Server-Side Deployment Selection:
Choose Web service runtime first Explore options
Web service runtime:
Apache Axis
Apache Axis2
JBossWS
Server:
🤣 org.jboss.ide.eclipse.as.42
Server Types
OK Cancel

Figure 2.7. Select Server and Web Service runtime

Click on the *Finish* button to see the next wizard view opened:



Figure 2.8. New Web Service Wizard

Click on the *Next* button to proceed:

E	Web Service	×
JBoss Web Service Please input the approp	e Code Generation Configuration riate option for the code generation	2
Custom package name	org.example.www.helloworld	
JAX-WS specification	2.0	~
Catalog file		Add
Binding files		Add
-		Remove
🗹 Generate default We	eb Service Implementation classes	
Vpdate the default V	Web.xml	
٢		Canaal
Ø	< <u>Back</u> <u>Next</u> <u>Finish</u>	Cancel

Figure 2.9. New Web Service Wizard

On this page, the default package name comes from the namespace of the WSDL document, you also can change it to any valid package name you want. JAX-WS specification should be set to 2.0 if your JBossWS runtime in JBoss Server is JBossWS native runtime. You can specify a catalog file and binding files if you have them. If you want the wizard to generate empty implementation

classes for the Web Service, check the *Generate default Web Service implementation classes* check box. If you want to update the default Web.xml file with the Web Service servlets configured, check the *Update the default Web.xml* check box. Click on the *Next* or on the *Finish* button to generate code.

Once the Web Service code is generated, you can view the implementation class and add business logic to each method.

```
🕖 GreeterImpl.java 🔀
  package org.apache.hello world soap http;
  import javax.jws.WebService;
  @WebService(name = "GreeterImpl", serviceName = "Greeter", endpoint
  public class GreeterImpl implements Greeter {
       public String sayHi() {
           return "";
       }
       public String greetMe(String requestType) {
 0
           return "";
       }
       public void greetMeOneWay(String requestType) {
 Θ
           return;
       }
       public void pingMe() {
 0
           return;
       }
   }
```

Figure 2.10. The generated implementation Java code

View the Web.xml file:



Figure 2.11. Web.xml

2.4. Creating a Web service from a Java bean using JBossWS runtime

The Web Service wizard assists you in creating a new Web service, configuring it for deployment, and then deploying it to the server.

To create a Web service from a bean using JBoss WS:

Setup JBoss WS and development environment.

Create a Dynamic Web project.

Add JBossWS Facet to Web project.

Create a Web Service from a java bean:

- Switch to the Java EE perspective *Window > Open Perspective > Java EE*.
- In the Project Explorer view, select the bean that you created or imported into the source folder of your Web project.



Figure 2.12. Create a new Bean Class

Click *File > New > Other*. Select Web Services in order to display various Web service wizards.
 Select the Web Service wizard. Click on the Next button.

•	New
Select a wizard	
Create a new XML web service.	
<u>W</u> izards:	
V 🗁 SQL Development	
👂 🗁 User Assistance	
👂 🗁 Web	
▽ 🗁 Web Services	
🔊 Ant Files	
🗊 Unit Test UDDI	
🏄 Web Service	
🔊 Web Service Client	
A WSDL	
ML	
Examples	~
O	Next > Einish Cancel

Figure 2.13. New Web Service

• On the first Web Service wizard page: select Bottom up Java bean Web service as your Web service type, and select the Java bean from which the service will be created:



Figure 2.14. Set Web Service Common values

- Select the stages of Web service development that you want to complete using the slider:
 - Develop: this will develop the WSDL definition and implementation of the Web service. This includes such tasks as creating modules that will contain generated code, WSDL files, deployment descriptors, and Java files when appropriate.
 - Assemble: this ensures the project that will host the Web service or client gets associated to an EAR when required by the target application server.
 - Deploy: this will create the deployment code for the service.
 - Install: this will install and configure the Web module and EARs on the target server.
 - Start: this will start the server once the service has been installed on it. The serverconfig.wsdd file will be generated.

- Test: this will provide various options for testing the service, such as using the Web Service Explorer or sample JSPs.
- Select your server: the default server is displayed. If you want to deploy your service to a different server click the link to specify a different server.
- Select your runtime: ensure the JBoss WS runtime is selected.
- Select the service project: the project selected in your workspace is displayed. To select a
 different project click on the project link. If you are deploying to JBoss Application Server
 you will also be asked to select the EAR associated with the project. Ensure that the project
 selected as the Client Web Project is different from the Service Web Project, or the service
 will be overwritten by the client's generated artifacts.
- If you want to create a client, select the type of proxy to be generated and repeat the above steps for the client. The better way is to create a web service client project separately.

Click on the Next button.

• On the JBoss Web Service Code Generation Configuration page, set the following values:

Web Service	×
JBoss Web Service Code Generation Configuration Please input the appropriate option for the code generation	2
 ✓ Generate WSDL file ✓ Update the default Web.xml 	
⑦ < <u>B</u> ack <u>N</u> ext > <u>F</u> inish Cance	el

Figure 2.15. Set Web Service values for Code Generation

- Generate WSDL file: select it, you will get a generated WSDL file in your project. But this wsdl's services' address location values are not a real address.
- After the Web service has been created, the following option can become available depending on the options you selected: Update the default web.xm file. If selected, you may test the web service by Explorer.

Click on the Next button.

• On this page, the project is deployed to the server. You can start the server and test the web service. If you want to publish the web service to a UDDI registry, you may click the Next button to publish it. If not, you may click the Finish button.

Web Service	×
Server startup	9
Start the server from this page.	2
In order to proceed the server "JBoss AS 4.2 at localhost" mus	st be st
Once the server is started the "next" button will be enabled.	
The "back" button can be used while the server is starting to)
change any previous settings in this wizard.	
Currently the server is stopped. Start server	
	el

Figure 2.16. Start a Server

After the Web Service has been created, the following options may become available depending on the options selected:

- the generated web services code
- If you selected to generate a WSDL file, you will get the file in your project's wsdl folder.

🗞 Project Explorer 🛚 🛸 Navigator 👘 🗖 🗖	🛿 HelloWorld.java 🖉 HelloWorldService.wsdl 🛱
e \$ ≥ ▼ ▼≌test	<pre><input message="tns:HelloWorld_sayHello2"/> <output message="tns:HelloWorld_sayHello2Response"></output></pre>
 Beployment Descriptor: test Java Resources: src Ford avapple www.belloworld 	<pre> </br></pre>
 Bog.example.www.helloworld HelloWorld.java morg.example.www.helloworld.jaxws 	<pre><operation name="sayHello"> <soap:operation soapaction="http://www.example.org/HelloWorld/sayHello"></soap:operation> <input/></operation></pre>
▷ ■ Libraries ▷ @ build ▷ @ WebContent	<pre><soap:body use="literal"></soap:body> <output></output></pre>
	<pre> <operation name="sayHello2"></operation></pre>
 ▶ aljavaScript Support > aljavaScript Support > aljavaScript Support > aljavaScript Support > aljavaScript Support 	<pre><soap:operation <input="" soapaction="http://www.example.org/HelloWorld/sayHello2"> <soap:body use="literal"></soap:body></soap:operation></pre>
	 <output> <soap:body use="literal"></soap:body> </output> <service name="HelloWorldService"> <port binding="tns:HelloWorldBinding" name="HelloWorldPort"></port></service>
	<pre><soap:address location="REPLACE_WITH_ACTUAL_URL"></soap:address> <!--/re--></pre>
	Design Source

Figure 2.17. The generated WSDL file

• If you selected to update the default web.xml, you will test the web service in the browser. Open the Explorer, input the url for the web service according to web.xml plus ?wsdl., you will get the WSDL file from Explorer.

Creating a Web service from a Java bean using JBossWS runtime



Figure 2.18. The Updated web.xml file

Creating a Web Service Client from a WSDL Document using JBoss WS

To create a Web Service Client from a WSDL Document using JBoss WS:

Setup JBoss WS and development environment.

Creating a Dynamic Web project.

Add JBossWS Facet to Web project.

Create a Web Service Client from a WSDL document:

- Switch to the Java EE perspective Window > Open Perspective > Java EE.
- In the Project Explorer view, select the bean that you created or imported into the source folder of your Web project.
- Click *File* > *New* > *Other*.Select Web Services in order to display the various Web service wizards. Select the Web Service Client wizard. Click Next button.

E New X
Select a wizard —
Access an existing XML web service
<u>W</u> izards:
type filter text
P C WED
Veb Services
🔊 Ant Files
🗐 Unit Test UDDI
A Web Service
💋 Web Service Client
A WSDL
Þ 🗁 XML
Examples
⑦ < Back Next > Bnish Cancel

Figure 3.1. New Web Service Client

• The first and second Web Service Client wizard page are same to <u>Web Service from a WSDL</u> <u>document</u>.



Figure 3.2. Set Web Service Common values

8	Web Service Client	×		
JBoss Web Service Code Generation Configuration				
Please input the appropriate option for the code generation				
		_		
Custom package name	org.example.www.helloworld			
JAX-WS specification	2.0	v		
Catalog file		Add		
Binding files		Add		
,		Pomouo		
		nemove		
⑦ < <u>B</u>	ack Next > Finish	Cancel		

Figure 3.3. Set Web Service values about WSDL file

The differences are:

• Client Type: Now only support Java Proxy.

Click Finish button.

After the Web Service Client has been created, the following may occur depending on the options you selected:

- the generated web service and client codes
- a client sample class.



Figure 3.4. Client Sample Class

JBoss WS use a Java class to test Web Service. A client sample class will be generated, you may run this client as a java application to call a web service.

JBoss WS and development environment

4.1. JBossWS Preferences

In this section you will know how JBossWS preferences can be modified during the development process.

JBossWS preferences can be set on the JBossWS preference page. Click on *Window* > *Preferences* > *JBoss Tools* > *Web* > *JBossWS Preferences*.

On this page you can manage the JBossWS Runtime. Use the appropriate buttons to Add more runtimes or to Remove those that are not needed.



Figure 4.1. JBossWS Preferences Page

Clicking on *Add* or *Edit* button will open the form where you can configure a new JbossWS runtime and change the path to JBossWS runtime home folder, modify the name and version of the existing JBossWS runtime settings. Press Finish to apply the changes.

Preferences X					
ty	pe filter	text]	JBossWS Preferences	
Þ	FreeMa Guvnor Help	arker Editor r	^	Name Versio Path ✓ jboss-4.2.2.GA 2.0 /home/user/Eclipse/jboss-4.2.2.GA	<u>A</u> dd
	HQL eq	litor			maya
Þ	Install/			Edit JBossWS Runtime	Inove
Þ	Java	Edit JBossWS	Runtir	ne	
Þ	JavaSo	Input new valu	les		
Þ	JBoss j		(-
	JBoss 1	Name:	jboss-4	.2.2.GA	
Ι.	JPA	Version	4.2.2	•	
	Plug-ir	Home Folder:	/home/	user/Eclipse/jboss-4.2.2.GA Browse	
Ι.	Projec		12		
P	Report	Customize	Boss v	ved Service runtime jars	
	Run/De				
	Server				
	Servic				
ľ	Validat				
ь	Web				
,	Web S				
	Axis				-
	Axis	0		Enish Cancel	
	IBos	sWS Preterence	s I		
<	Don	un Dialon Select	ion V		
0	2			ок с	ancel

Figure 4.2. Edit JBossWS Runtime

WS container allows Source and JavaDoc locations to be set via the Properties dialog on each contained .jar: right-click on any .jar file in the Project Explorer view, select *Properties*. Choose *Java Source Attachment* and select location (folder, JAR or zip) containing new source for the chosen .jar using one of the suggested options (workspace, external folder or file) or enter the path manually:

(Properties for /home/user/jbdevstudio/jboss-eap/jboss-as/client/javassist.jar		
	type filter text	Java Source Attachment	
	Java Source Attachment Javadoc Location Native Library	Select the location (folder, JAR or zip) containing the source for 'javassist.jar': Logation path:	<u>W</u> orl Exte
	0	ОК	0

Figure 4.3. Classpath Container: Java Source Attachment

Click on Apply and then on Ok.

To change Javadoc Location choose *Javadoc Location* and specify URL to the documentation generated by Javadoc. The Javadoc location will contain a file called *package-list*:

e filter text	Javadoc Location				\$ ~
Java Source Attachment Javadoc Location	Specify the location (URI contain a file called 'pack	_) of the docume (age-list'.	ntation generated by J	avadoc. The Javadoc loo	catio
Native Library	Iavadoc URL (e.g. 'http://www.iavadoc.urg.	tp://www.sample-u	url.org/doc/' or 'file:/c:/i	myworkspace/myprojec	t/doo
	Javadoc location path:				B
					⊻a
	 Javadoc in archive 				
		External file	O Workspace file		
	Archive path:				B
	Path within archive:				Br
					Va
				Pastara Dafaulta	
				Restore Defaults	
0					

Figure 4.4. Classpath Container: Javadoc Location

Click on Apply and then on Ok.

4.2. Default Server and Runtime

Open *Window > Preferences > Web Services > Server and Runtime*. On this page, you can specify a default server and runtime.

For ease of use, the better way is to set runtime to JBoss WS.

After server and runtime are specified, click on the Aply button to save the values.

E	Preferences				
3	/pe filter text	Server and Runtime	1		
⊳	General	Server	IBoss v4.2		
⊳	Ant	Server:	10035 V4.2		
⊳	Data Management	Web service runtime:	JBossWS		
₽	Help				
₽	Install/Update				
₽	Java				
⊳	JavaScript				
	JPA				
⊳	Mylyn				
⊳	Plug-in Development				
Þ	Remote Systems				
⊳	Run/Debug				
⊳	Server				
	Service Policies				
⊳	Team				
⊳	Usage Data Collector				
	Validation				
₽	Web				
~	Web Services				
	Axis Emitter				
	Axis2 Preferences				
	JBossWS Preferences				
	Popup Dialog Selection				
	Project Topology				
	Resource Management				
	Scenario Defaults				
	Server and Runtime				
	Test Facility Defaults				
	Wizard Validation				
	WSDL Files				
⊳	XDoclet				
⊳	XML				
C	2				
L Ì					

Figure 4.5.

On the whole, this guide covers the fundamental concepts of work with tooling for JBossWS. It describes how to easily create a Web Service and a Web Service Client using JBossWS Runtime and adjust JBossWS and development environment as well.

If the information on JBossWS tools in this guide isn't enough for you, ask questions on our *forum* [http://www.jboss.com/index.html?module=bb&op=viewforum&f=201]. Your comments and suggestions are also welcome.