Introduction to JAX-WS

- JAX-WS Overview
- Writing a Simple Service
- Writing a Client

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

What is JAX-WS?

- The Java API for XML-Based Web Services **(JAX-WS)** is the standard Java API for programming Web services - both services and clients
- A major goal of JAX-WS is source-code portability across implementations and runtime interoperability with services implemented with other technologies
- JAX-WS is required part of Java Enterprise Edition 5 and later

---

This specification was developed via the Java Community Process as JSR 224. You can download the specification from:

`http://jcp.org/aboutJava/communityprocess/mrel/jsr224/index3.html`

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Version History

- **JAX-RPC Version 1.0** was released in June 2002
- **JAX-RPC Version 1.1** was released in October 2003
- **JAX-WS 2.0** (originally named JAX-RPC 2) was released in May 2006
- **JAX-WS 2.1** was released in May 2007
- **JAX-WS 2.2** was released in December 2009

The Java API for XML-Based RPCs (JAX-RPC) is the predecessor to JAX-WS. As its name implies, JAX-RPC was geared mostly toward RPC-based services, but also provided some support for document-oriented services.

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

JAX-WS Implementations

- JAX-WS itself is a specification and an API
- To develop code that uses JAX-WS, you need an implementation:
    - The JAX-WS Reference Implementation (Metro)
    - Any JEE Version 5 or later compliant application server
    - Java Standard Edition, Version 6 or later

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

JAX-WS Highlights

- Supports Java 5 annotations so applications can provide metadata (instead of writing XML deployment descriptors)
- Adheres to the *WS-I Basic Profile* to foster interoperability with other Web service implementations
- Supports SOAP 1.2 and SOAP 1.1
- Supports the *Message Transmission Optimization Mechanism* (MTOM) for better interoperability for messages with attachments

WS-Interoperability is another specification that focuses on best practices for Web service interop. You can read more at:

`www.ws-i.org`

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Related Java APIs

- **JAX-B** - The Java API for XML Binding
- **StAX** - The Streaming API for XML in Java
- **SAAJ** - SOAP with Attachments API for Java
- **WSEE** - Web Services for Java Enterprise Edition
- **JAX-RS** - The Java API for RESTful services
- **JAX-R** - Java API for XML Registries

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Writing a Simple Service

- To create a JAX-WS service, write a class annotated with JAX-WS annotations to configure the service

```
1    package sample;
2
3    import javax.jws.WebService;
4
5    @WebService
6    public class HelloWorld
7    {
8       public String hello(String param)
9       {
10         return param + ", World";
11      }
12   }
```

The @WebService annotation accepts optional parameters that let you configure the service. For example, you can specify the target namespace.

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Deploying the Service

- JSE Version 6 and later include all of the support needed to deploy Web services (including a Web server)
- Alternatively, you can package the service into a standard WAR/EAR and deploy onto a JEE Version 5 or later application server

Note that if you want to use JSE, you need the Java Development Kit (JDK) to compile and generate services.

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Writing a Client for the Simple Service

- Before coding the client, you run a tool that generates a client artifacts, typically using the WSDL
- The client then uses the generated artifacts, whose names derive from elements in the WSDL, to access the service

```
1    package sample;
2
3    public class HelloClient
4    {
5       public static void main(String[] args)
6       {
7          HelloWorldService service = new HelloWorldService();
8          HelloWorld proxy = service.getHelloWorldPort();
9
10         System.out.println(proxy.hello("hello"));
11      }
12   }
```

JSE version 6 provides the "wsimport" utility that reads the WSDL and generates client artifacts.

The "proxy" object handles all of the Java/SOAP translation so the client program can work totally in Java.

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Chapter Summary

In this chapter, you learned:

- The fundamentals of JAX-WS
- How to write a simple service and client

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum

Copyright © Descriptor Systems, 2001-2016. Course materials may not be reproduced in whole or in part without prior written consent of Joel Barnum