

Unit 1 Study Guide: The Basics of Web Services

This unit introduces the basic ideas behind web services. This includes discussion of what web services are and why they are important.

Unit 1: The Basics of Web Services	1
1.0 Introduction and Study Guide.....	1
1.0.1 Introduction.....	1
1.0.2 Unit Guide	1
1.1: The Need for Web Services.....	3
1.1.1 Differentiating Web Services from Internet Services.....	3
1.1.2 Human vs. Software Agents	4
1.1.3 Application Interfaces.....	5
1.2: What is a Web Service?.....	5
1.2.1 Implementing Application Interfaces	5
1.2.2 Properties of a Web Service	6
1.2.3 Why Web Services are more than Web Sites	7
1.3: The Web Service Technology Stack	8
1.3.1 An Outline of Web Service Standards.....	8
1.3.2 The Technologies in Brief	9
1.3.3 The Technologies in Operation	10

1.0 Introduction and Study Guide

1.0.1 Introduction

Web services offer a new way of building distributed systems that are openly accessible, typically via the Internet. This unit concentrates on three main questions:

- why do we need web services?
- how do web services differ from existing web sites or Internet information systems?
- what are the components that make up a web service?

During this unit you should build an understanding from a variety of sources and various activities that will enable you to answer these questions. You should also start building a glossary that contains definitions of terms which are currently unfamiliar to you, but will become very familiar by the time you complete this module.

After this unit you should be able to:

- describe the identifying characteristics of a web service
- contrast web services with alternative approaches to building distributed systems
- identify the core components of the web services technology stack
- explain the rationale for the inclusion of these components

1.0.2 Unit Guide

This unit is primarily focussed on definitions and standards. Like many subjects, there is a core body of knowledge that underpins the study of web services although people may argue over the details. This is particularly true when the organisations involved in the development of web service standards such as IBM, Sun and Microsoft sell competing technologies.

To combat these differences this module in general, and this unit in particular, uses a number of different sources to provide different perspectives. This should encourage students to be critical of the information they are given and to learn to identify both similarities and differences in the material.

Although the basic questions addressed in the unit could be tackled in any order this unit is designed to be worked through in the specified order. Later sessions will build on and refer back to previous sessions.

Resources required for this unit:

- The reading for this unit is based on the following books:

Web Services Essentials, Ethan Cerami, O'Reilly, February 2002

ISBN 0-596-00224-6

URL <http://search.safaribooksonline.com/0596002246>

Developing Enterprise Web Services: An Architect's Guide, Prentice Hall PTR, November, 2003

ISBN 0-13-140160-2

URL <http://search.safaribooksonline.com/0131401602>

Programming Web Services with SOAP, Pavel Kulchenko, James Snell, Doug Tidwell, O'Reilly, December 2001

ISBN 0-596-00095-2

URL <http://search.safaribooksonline.com/0596000952>

If you have not already done so you need to add these books to your Safari bookshelf.

- Internet access is also required for additional reading and/or demos in sessions 1.2.1, 1.3.3, 1.4 and 1.4.3

1.1: The Need for Web Services

This session examines the need for web services by investigating how and why you might want to interface two systems in a network environment.

1.1.1 Differentiating Web Services from Internet Services

The principles, standard and technologies that are the building blocks of web services are essentially *technical* in nature. The use of the term "service" is linked to the concept of a "Service Oriented Architecture" (SOA) in which one program or application provides a service to another application.

The fundamental building blocks of "technical" web services should not be confused with the business planning, marketing or requirements analysis that is required to develop a *business* service. Businesses such as Amazon, eBay or Google do provide services which are delivered via the Internet or World Wide Web. However these business services can be delivered by applications or systems which are built using standard Internet technologies, such as HTML/HTTP, without any need for *web* services (in the technical sense of the term).

In the remainder of this module, we shall limit use of the term "*web service*" to describing a system built using standards such as SOAP, XML, WSDL and UDDI. The actual business or "end user" service that is being delivered by an some kind of Internet information system will be described as a "*business service*". This applies irrespective of whether the business service uses web service technology or not.

The distinction between web service and end user service enables us to differentiate between the service delivered by organisations to their users, e.g.:

- Amazon - whose basic service is selling consumer goods
- eBay - who facilitate private auctions
- Google - who enable users to search the web

... and the technical work being done by the same companies to make their systems available, via web services, for integration into other business services or applications.

Task 1.1

Read the following description of [Service-Oriented Architecture](#)¹ that comes from O'Reilly's web services site up to (but not including) the section on 'Additional Constraints'. You should pay particular attention to the examples and attempt to identify who the service provider and consumer would be in each case.

Task 1.2

Define the following terms for inclusion in your own technical glossary

- service provider
- service consumer
- software agent

Post your definitions to the discussion group. In addition you should post comments on three other definitions or follow-up comments submitted by others.

cont.

¹ <http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html>

Task 1.2 (continued)

If you would like a more substantial introduction to the concept of software agents you could look at the paper by N. R. Jennings and M. J. Wooldridge entitled "[Software Agents](#)"² from the IEE Review, January, 1996, 17-20.

1.1.2 Human vs. Software Agents

The idea of service oriented architecture is based on "interacting software agents". In order to understand this further, we need to look at the differences between human agents and software agents and the differences in how they might access and use services.

Task 1.3

Read the sources listed below with the following questions in mind:

- if a *person* wanted to book a holiday, how would they find, interpret and act on information available on the Internet;
- if a *program* was directed by a user to book a holiday, how could the program find, interpret and act on information available to it via the Internet;

Required reading from *Web Services Essentials*:

- [1.1.1 The Web Today: The Human-Centric Web](#)³
- [1.1.2 Web Services: The Application-Centric Web](#)⁴
- [1.1.3 The Web Services Vision: The Automated Web](#)⁵

Exercise 1.1

Identify two advantages and two disadvantages for developing a program to book your holiday, compared to doing it in person. You should consider all the stages of the booking process covering:

- finding places you might want to go;
- deciding which place best suits you;
- booking the holiday.

² <http://www.ecs.soton.ac.uk/~nrj/download-files/IEE-Review96.ps>

³ <http://search.safaribooksonline.com/JVXSL.asp?xmlid=0-596-00224-6/webservess-CHP-1-SECT-1#webservess-CHP-1-SECT-1.1>

⁴ <http://search.safaribooksonline.com/JVXSL.asp?xmlid=0-596-00224-6/webservess-CHP-1-SECT-1#webservess-CHP-1-SECT-1.2>

⁵ <http://search.safaribooksonline.com/JVXSL.asp?xmlid=0-596-00224-6/webservess-CHP-1-SECT-1#webservess-CHP-1-SECT-1.3>

1.1.3 Application Interfaces

People have the flexibility to learn and work with a variety of different application interfaces (a search engine, or a particular web site) in order to achieve a particular goal. Given software agents do not have the same degree of flexibility it is necessary to define the interface between two software agents (a service user and a service provider) more tightly or precisely.

Task 1.4

Read section [1.1 What are Web Services](#)⁶ from *Developing Enterprise Web Services*.

As you read this section make a note of the different types of information involved in defining system interfaces.

Exercise 1.2

Identify the three types of information required to define any one of the system interfaces involved in the stock calculator application.

Hint: what three elements are labelled for each external service in **Figure 1.2**?

1.2: What is a Web Service?

The arguments put forward for developing web services can be seen as either functional or economic. Functional arguments are based on the premise that if I can interface two systems then I can achieve something I couldn't do before. Economic arguments for using web services suppose that it will be quicker to reuse an existing service than write my own application.

In order to assess either type of justification for web services it is necessary to understand what the proposed approach actually involves. This session provides an overview of how web services differ from other approaches to integrating or interfacing systems.

1.2.1 Implementing Application Interfaces

The idea of extracting information from one application for use in another is not new. There have been many different standards or approaches to interfacing applications over the last thirty years. Given this, it is important to understand what differentiates web services from the alternatives. This should provide some insight into the benefits of web services.

Task 1.5

Read section [1.2 Why Web Services are Important](#)⁷ from *Developing Enterprise Web Services*, up to and including **Figure 1.3**.

cont.

⁶ <http://search.safaribooksonline.com/JVXSL.asp?xmlid=0-13-140160-2/ch01lev1sec1>

⁷ <http://search.safaribooksonline.com/JVXSL.asp?xmlid=0-13-140160-2/ch01lev1sec2>

Task 1.5 (continued)

Identify which of the following terms are used in this section:

- screen scraping
- CORBA
- DCOM
- RMI
- EDI

Find out what each of the terms means and how they might apply to application integration. Add them to your technical glossary

Exercise 1.3

Match the following terms with the most appropriate definition or description.

- a) CORBA b) RMI c) screen scraping d) DCOM e) EDI

1. "computer-to-computer exchange of business data in standard formats where information is organized according to a specified format set by both parties, allowing a 'hands-off' computer transaction that requires no human intervention or rekeying on either end.". Long established history with a focus on agreed data formats and proprietary, industry specific systems.
2. "open, vendor-independent architecture and infrastructure that computer applications use to work together over networks". Can be seen as costly/difficult/efficient/manageable depending on your perspective.
3. a mechanism to "support communication among objects on different computers—on a LAN, a WAN, or even the Internet" that means an "application can be distributed at locations that make the most sense to your customer and to the application". Operating system dependent.
4. data capture based on extracting information from a particular location in a display interface, e.g. a GUI or web page. Can be easy to set up but prone to errors as interfaces change.
5. "a simple and direct model for distributed computation with Java" that allows an object running in one machine to invoke methods on an object running in another machine. Easy way to extend or distribute Java applications, but is limited to Java.

1.2.2 Properties of a Web Service

This part of the session identifies what it is that makes web services different from other approach to defining system interfaces.

Task 1.6

Read the start of section [1.1 Introduction to Web Services](#)⁸ from *Web Services Essentials*, up to section **1.1.1 The Web Today: The Human-Centric Web** which was covered earlier.

cont.

⁸ <http://search.safaribooksonline.com/JVXSL.asp?xmlid=0-596-00224-6/webservess-CHP-1-SECT-1#webservess-CHP-1-SECT-1.1>

Task 1.6 (continued)

In summary, web services can be characterised as systems that are:

- network accessible;
- use XML messaging;
- software (operating system or programming language) independent;
- self describing;
- discoverable.

Identify which of these properties are unique to web services when compared with the integration approaches identified in Unit 1.2.1 Implementing Application Interfaces.

Post your answers and justifications to the discussion group.

1.2.3 Why Web Services are more than Web Sites

If you want to extract information from an internet information system, why not just deconstruct the relevant web pages to get access to the data you want? This session provides some reasons not to.

Task 1.7

Read sections [1.1 What is a Web Service](#)⁹ and [1.2 Web Service Fundamentals](#)¹⁰ up to and including **1.2.1 What Web Services Look Like** from *Programming Web Services with SOAP*.

You should identify how the definition of a web service given here differs from that given in [Unit 1.2.2](#).

Exercise 1.4

Identify which of the properties defined in [Unit 1.2.2](#) do not necessarily apply to existing web sites.

Exercise 1.5

Using spider and robots, e.g. programs such as wget or [screen scrapers](#)¹¹, it is possible to automatically download and deconstruct web pages to extract information from them. Given this, identify why the web service properties are important for defining system interfaces.

⁹ <http://search.safaribooksonline.com/JVXSL.asp?xmlid=0-596-00095-2/progwebsoap-CHP-1-SECT-1>

¹⁰ <http://search.safaribooksonline.com/JVXSL.asp?xmlid=0-596-00095-2/progwebsoap-CHP-1-SECT-2>

¹¹ www.screen-scraper.com

1.3: The Web Service Technology Stack

The simplest way of deciding if an application is a web service is to see whether it uses web services technologies and standards. This session introduces the core technologies that make up the WS-I (the Web Service Interoperability Organisation) [Basic Profile](#)¹² or "technology stack".

1.3.1 An Outline of Web Service Standards

Depending on your viewpoint, there are four or five levels of standards or technology that are required for fully implemented web services.

Task 1.8

Read section [1.3 The Web Service Technology Stack](#)¹³ from *Programming Web Services with SOAP*.

Identify where in the stack you think the following technologies should go:

- XML
- UDDI
- SOAP
- WSDL

Add entries in your technical glossary for each of the terms listed above.

Task 1.9

Read section [1.2.2 Web Service Protocol Stack](#)¹⁴ from *Web Services Essentials*.

Compare the stack given here with the one defined in Task 1.8. As you can see, different people have different ways of chopping up the world. Practically speaking, these differences disappear when the stack is examined in more detail.

Task 1.10

Read section [1.2.1 Web Service Roles](#)¹⁵ from *Web Services Essentials*. Then compare the following terms with entries you may already have in your glossary:

- service provider;
- service requestor;
- service registry.

If two terms are used synonymously then you should make sure you know both terms, e.g. "*service requestor*" and "*service consumer*". If two terms are used slightly differently then you should be aware of the differences. For example "*SOAP*" and "*XML messaging*" appear in very similar contexts because SOAP is one form of XML messaging. However XML messaging also encompasses XML-RPC which has some different characteristics when compared to SOAP.

¹² <http://www.ws-i.org/Profiles/Basic/2003-08/BasicProfile-1.0intro.pdf>

¹³ <http://search.safaribooksonline.com/JVXSL.asp?xmlid=0-596-00095-2/progwebsoap-CHP-1-SECT-3>

¹⁴ <http://search.safaribooksonline.com/JVXSL.asp?xmlid=0-596-00224-6/webservess-CHP-1-SECT-2# webservess-CHP-1-SECT-2.2>

¹⁵ <http://search.safaribooksonline.com/JVXSL.asp?xmlid=0-596-00224-6/webservess-CHP-1-SECT-2# webservess-CHP-1-SECT-2.1>

1.3.2 The Technologies in Brief

Whichever way people carve up the web services technology stack, the same core standards are always present, namely XML, XML messaging (usually with SOAP or XML-RPC), WSDL and UDDI.

Task 1.11

Read section [2.2 XML Documents](#)¹⁶ from *Developing Enterprise Web Services*.

Create an XML document to represent your favourite DVD and post it to the discussion group.

Task 1.12

Read section [1.3 XML Messaging](#)¹⁷ from *Web Services Essentials*.

Match the various components of an XML-RPC exchange to the interface elements defined in [Unit 1.1.3](#). If you don't already have them in your glossary, add the following terms:

- service request;
- service response.

Task 1.13

Read section [1.4 Service Description: WSDL](#)¹⁸ from *Web Services Essentials*.

Identify which of the following web service characteristics (repeated from Session 1.2.2) WSDL is designed to support:

- network accessible;
- use XML messaging;
- software independent;
- self describing;
- discoverable.

Task 1.14

Read section [1.5 Service Discovery: UDDI](#)¹⁹ from *Web Services Essentials*.

Identify how many UDDI registries you can search from the [UDDI home page](#)²⁰.

¹⁶ <http://search.safaribooksonline.com/JVXSL.asp?xmlid=0-13-140160-2/ch02lev1sec2>

¹⁷ <http://search.safaribooksonline.com/JVXSL.asp?xmlid=0-596-00224-6/webservess-CHP-1-SECT-3>

¹⁸ <http://search.safaribooksonline.com/JVXSL.asp?xmlid=0-596-00224-6/webservess-CHP-1-SECT-4>

¹⁹ <http://search.safaribooksonline.com/JVXSL.asp?xmlid=0-596-00224-6/webservess-CHP-1-SECT-5>

1.3.3 The Technologies in Operation

Having looked at the technologies briefly, this session looks at hands on use of the technologies and how they might be used in developing a web service.

Task 1.15

Read section [1.2.3 Architectural Snapshot: The IBM Web Services Browser](#) ²¹ from *Web Services Essentials*.

This section gives an overview of how web services can be dynamically accessed or composed. However, the url given no longer points to <http://demo.alphaworks.ibm.com/browser/> and you will be redirected to <http://www-128.ibm.com/developerworks/webservices/demos/ettk> where you can try the demo.

Task 1.16

Read section [1.8 All Together Now](#) ²² from *Web Services Essentials*.

In the remainder of this module we will be taking a "service provider perspective" in learning about web services, starting with XML and moving through XML messaging to the higher levels of the web services stack.

²¹ <http://search.safaribooksonline.com/JVXSL.asp?xmlid=0-596-00224-6/webservess-CHP-1-SECT-2#webservess-CHP-1-SECT-2.3>

²² <http://search.safaribooksonline.com/JVXSL.asp?xmlid=0-596-00224-6/webservess-CHP-1-SECT-8>