JAVA ENTERPRISE EDITION: A PRACTICAL APPROACH

JAVA ENTERPRISE EDITION: A PRACTICAL APPROACH

J2EE – JDBC with Oracle – HTML – Java Servlets – JSP – JSTL –
SOA and Web Services – JMS with Apache ActiveMQ –
Project Management with Apache Maven

By

B. Mohamed Ibrahim MCA., M.Phil., MBA.,
Software Technical Consultant

UNIVERSITY SCIENCE PRESS

(An Imprint of Laxmi Publications Pvt. Ltd.)

BANGALORE • CHENNAI • COCHIN • GUWAHATI • HYDERABAD
JALANDHAR • KOLKATA • LUCKNOW • MUMBAI • RANCHI

NEW DELHI • BOSTON, USA

Published by :
UNIVERSITY SCIENCE PRESS
(*An Imprint of Laxmi Publications Pvt. Ltd.*)
113, Golden House, Daryaganj,
New Delhi-110002

Phone : 011-43 53 25 00
Fax : 011-43 53 25 28

www.laxmipublications.com
info@laxmipublications.com

Price : ₹ **250.00** *Only*. *First Edition* : 2013

OFFICES

✆ **Bangalore**	080-26 61 15 61	✆ **Chennai**	044-24 34 47 26
✆ **Cochin**	0484-237 70 04, 405 13 03	✆ **Guwahati**	0361-254 36 69, 251 38 81
✆ **Hyderabad**	040-24 65 23 33	✆ **Jalandhar**	0181-222 12 72
✆ **Kolkata**	033-22 27 43 84	✆ **Lucknow**	0522-220 95 78
✆ **Mumbai**	022-24 91 54 15, 24 92 78 69	✆ **Ranchi**	0651-2214764

UJE-9671-250-JAVA ENTER PRAC APP-IBR **C—**
Typesetted at : Shubham Composer, New Delhi. *Printed at :*

PREFACE

The web has grown very fast, and the rate of its growth continues to be exponential. Many enterprises were already sunset to standalone desktop applications, and started working on web-based applications. And they can access their applications throughout the world, mainly without installing any special client applications.

Currently, a new trend in technology for Enterprise Application Interaction (EAI), which uses web protocols for data transmission. And every module can be accessed via web as services. This Service Oriented Architecture (SOA) with Web Services breaks platform and implementation software dependency.

Java platform is one of the best choices of the web developers, which supports many attractive features including rapid development and platform independent. Due to the rapid growth, the Sun Microsystems released the second generation of Java with three major divisions: J2SE (Java 2 Platform Standard Edition), J2EE (Java 2 Platform Enterprise Edition) and J2ME (Java 2 Platform Micro Edition). The J2SE consists of Application Programming Interfaces (APIs) needed to build the Java applications and applets. The J2ME supports the development of Java applications for mobile/ wireless devices. And, the J2EE is a replacement for the traditional CGI (Common Gateway Interface) technologies and is used to build dynamic and enterprise web applications. The J2EE components are modified and enhanced time to time, and released to the users. Due to maturity, stability, scalability, security and enhanced features, the generation indicator (the number “2” in J2EE) is removed and now J2EE is referred as JEE (Java Enterprise Edition). The latest version of this Java Enterprise Edition is Java EE 6.

This book is composed of ten chapters. All the chapters and the concepts in each chapter are properly ordered, and the readers are advised to read them sequentially as some chapters / sections use the concepts explained in the previous chapters / sections. The following are the brief introduction to the chapters in this book.

Chapter 1 (J2EE – An Introduction): This chapter briefly introduces Java Enterprise Edition. The Component Technologies, Service Technologies, Integration Technologies, and Communication Technologies of the Java Enterprise Edition platform are briefed. Further, it also lists the supported technologies of JEE 6 platform with their individual version numbers. The Client-Server architecture with its logical layering is explained. Then, the JEE Architecture is introduced.

Chapter 2 (Working with Oracle Database): This chapter illustrates SQL and PL/SQL commands/ logic to work with Oracle database. This chapter is given for the readers to gain experience in the database operations which is required while reading database based examples, especially JDBC examples. Oracle installation procedure for Windows operating system is given with screenshots. This chapter explains Data Definition Language, Data Manipulation Language, and Transaction Control Language SQL commands with syntaxes and examples.

Chapter 3 (JDBC): This Chapter explores the JDBC API. The Java Database Connectivity (JDBC) APIs are used to interact with RDBMS. The JDBC Architecture, supported types of drivers and their features are described. The common data types in database and their corresponding data types in Java are given for reference. The JDBC APIs are listed with brief explanation. The advanced concepts such as Scrollable Result Set, Batch Updates, Transaction Management, and Calling Stored Procedures are explained with sample code and screen output.

Chapter 4 (Understanding HTML FORM Elements): This chapter aims to list and explain the form input / output elements in HTML. To create dynamic web applications, the HTML form elements are used to get and to set the data. And these data are accessed and processed by server-side web applications. The output of any web application is in HTML format. The web client such as browsers can understand these HTML content and display the information by interpreting the HTML elements. This chapter also explains some basic elements that are used frequently such as formatting elements, listing elements and frameset elements. Some samples with code and their corresponding output are given at the end of the chapter to get hands-on experience.

Chapter 5 (Java Servlets): This chapter explains Java Servlet API. It briefly explains the importance of Java Servlets in the dynamic webpage development. The life-cycle of a Servlet is given, and the programmer can do appropriate actions at each stage of the life-cycle. The Important Servlet APIs are listed with brief explanation. Both Generic Servlets and HTTP Servlets are explained with their features. The advanced concepts including Session Tracking and Manipulating Cookies are given with code samples. Apache Tomcat is used as Servlet Container in executing the examples given this chapter. The installation procedure of Tomcat, configuration, deployment descriptor format for web applications, and the procedure for deploying web applications on Tomcat are given for reference. A list of sample programs with their correspond screen output are given at the end of the chapter, and these samples uses almost all the explained Servlet APIs in this chapter.

Chapter 6 (Java Server Pages): This chapter explains the very important and most used component of Java Enterprise Edition: Java Server Pages, simply called JSP. All JSP code support Java Servlet API, and it simplifies the development process by providing support to the developers to embed Java code in HTML documents. And all JSP code will be converted into Java Servlet code before execution. This chapter analyses the aspects of JSP and compares with other competitive technologies such as Active Server Pages (ASP). The components of JSP are grouped and explained as JSP Tags, Implicit Objects, and Standard Actions. Java code can be embedded with JSP Tags, and JSP provides frequently used important objects implicitly that those objects should be created in case of Java Servlets. These directives are used to provide page-specific global information about JSP pages to the JSP engine before compilation. A list of sample programs that use the JSP concepts is given for illustration at the end of this chapter.

Chapter 7 (JSTL): This chapter focuses on standard tag libraries for JSP. JSTL adds customized tags features to JSP. This chapter explains with code samples the core tags, formatting tags, XML tags, and SQL tags. The developers can easily accomplish by using these JSTL tags, rather than using complex JSP Scripting code. JSTL also provides many implicit objects that these objects can also be used with the implicit objects provided by JSP. JSTL is not an alternate to JSP, but provides additional support and features to JSP; otherwise they should be achieved through complex JSP code. Data formatting tags in JSTL are very much handy, and also these data are formatted based on locale (Internationalization). JSTL supports sufficient tags for database manipulation, and these tags are given with syntaxes and code samples. XML provides a flexible means for representing structured data. Since the XML format is just simple text, it is well suited for exchanging data particularly between loosely coupled systems. The XPath is used to navigate through elements and attributes in an XML document. This chapter explains briefly the JSTL support for parsing and manipulating XML data.

Chapter 8 (Service Oriented Architecture and Web Services): This chapter is given for Web Services. A service is a self-contained software component (subroutine) that performs a specific task. A Service Oriented Architecture is essentially a collection of services; these services communicate with each other and each service does not depend on the context of any other services. A Web Service is an application functionality service that can be accessed by any application via standard Internet Protocols such as HTTP. This chapter explains the SOA with Web Services with the supporting technologies such as WSDL and SOAP. A complete working example is given, which uses Apache Axis2 (Web Services Engine) and Web Tool Platform (Eclipse extension for code generation).

Chapter 9 (Message Oriented Middleware and Java Messaging Service): This chapter focuses on the Java Messaging Service IJMS). The JMS is one of the Message Oriented Middleware implementation, supporting asynchronous communication between client and server in a Client-Server application. This chapter uses Apache ActiveMQ as JMS provider and given examples based on this provider. This chapter explains more elaborately with working examples the two JMS domains: Point-To-Point Messaging and Publish/Subscribe Messaging. The important JMS APIs are listed with explanation.

Chapter 10 (Project Management with Apache Maven): This chapter deals with Project Management activities including creation, build, deployment of Java standard applications and Web applications. Apache Maven is a project management tool, more than a build tool such as Apache Ant. This chapter takes Apache Maven as the project management tool and explains how to do the operations with this tool. Three complete examples are given in this chapter to create a Java standalone application project, a Java web application project, and a multi-module application project.

The core component of Java Enterprise Edition, Enterprise Java Beans (EJB) is not covered in this book, as it requires many pages and I want to restrict the number of pages of this book. I am planning to write a separate book for EJB 3. And, it is not a complete book for JEE. This book is for beginners and intermediates to quickly understand the important JEE components.

I thank the Almighty Allah, who gave me the soul, strength, knowledge and everything. All praise is due to him. Salutations on the Prophet (may Allah's blessings and peace be upon him) and his descendents. I thank all the well wishers of mine, who prayed the Almighty for me.

—Author

CONTENTS

CHAPTER 1. J2EE: An Introduction **1–5**

1.0 Introduction 1

1.1 Java Generation 1

1.2 Components of JEE 1

1.3 Client-Server Technology 3

1.4 JEE Architecture 5

Summary 5

CHAPTER 2. Working with Oracle Database **6–21**

2.0 Introduction 6

2.1 SQL 6

2.2 Data Types 6

2.3 DDL Commands 7

2.4 DML Commands 10

2.5 Transaction Control Language (TCL) Commands 13

2.6 Set Operations on Database Tables 16

2.7 Join Operations on Database Tables 17

2.8 Sub-queries 18

2.9 Stored Procedures 18

2.10 Some Built-in Functions 19

Summary 21

CHAPTER 3. JDBC **22–54**

3.0 Introduction 22

3.1 Typical Application Architecture 22

3.2 Data Type Comparison 23

3.3 Establishing Connection with Database 23

3.4 JDBC API 24

3.5 Sample Programs 32

Summary 54

CHAPTER 4. Understanding HTML Form Elements **55–67**

4.0 Introduction 55

4.1 General Format of a HTML Program 55

4.2 Some Basic HTML Tags 55

4.3 Frames and Framesets 57

4.4 HTML form and form Elements 57
4.5 Sample Programs 59
Summary 67

CHAPTER 5. Java Servlets **68–96**

5.0 Introduction 68
5.1 Java Servlets and Dynamic Webpage Development 68
5.2 Life-Cycle of a Servlet 68
5.3 Servlet API 69
5.4 Deployment Descriptor for Web Applications 73
5.5 Deploying Web Application in Apache Tomcat 74
5.6 Sample Servlet Programs 78
Summary 96

CHAPTER 6. Java Servler Pages (JSP) **97–130**

6.0 Introduction 97
6.1 Aspects of JSP 97
6.2 Life Cycle of a JSP Program 98
6.3 JSP Tags 99
6.4 Implicit Objects 101
6.5 JSP Directives 104
6.6 Standard Actions 107
6.7 Session Tracking 112
6.8 Sample Programs 112
Summary 130

CHAPTER 7. JSTL **131–207**

7.0 Introduction 131
7.1 Installation 131
7.2 Tag Libraries 131
7.3 Expression Language 132
7.4 Implicit Objects 136
7.5 Accessing Custom Variables 157
7.6 Conditional Constructs and Looping Tags 161
7.7 JSTL Built-in Functions 175
7.8 Data Formatting with JSTL 179
7.9 Database Access with JSTL 191
7.10 Parsing XML with JSTL 200
Summary 207

CHAPTER 8. Service Oriented Architecture and Web Services **208–219**

8.0 Introduction 208
8.1 Service Oriented Architecture (SOA) and Web Services 208

8.2 Components of Web Services 209

8.3 Web Service Example: Arithmetic Operations on Numbers 209

Summary 219

CHAPTER 9. Message Oriented Middleware and Java Messaging Service (JMS) 220–252

9.0 Introduction 220

9.1 Message Oriented Middleware (MOM) 220

9.2 Java Messaging Service (JMS) 221

9.3 JMS Message Delivery Modes (JMS Domains) 221

9.4 JMS Message Structure 221

9.5 Apache ActiveMQ 223

9.6 JMS API 228

9.7 JMS Demo using ActiveMQ 236

Summary 252

CHAPTER 10. Project Management with Apache Maven 253–299

10.0 Introduction 253

10.1 Ant versus Maven 253

10.2 Installing Maven 254

10.3 Maven Coordinates 255

10.4 Maven Life-cycle Phases 256

10.5 Maven Directory Structure 258

10.6 Developing a Stand-alone Java Application: Demonstration 258

10.7 Developing a Web-based Java Application: Demonstration 267

10.8 Developing a Multi-Module Project: Demonstration 280

Summary 299

Index 300-301

1

J2EE: An Introduction

1.0 INTRODUCTION

This chapter introduces the generation of Java. The components of Java Enterprise Edition and their features are briefly explained. It further explains the client-server technology and how it fits in application software. This chapter ends by giving you introduction to J2EE Architecture.

1.1 JAVA GENERATION

Java was developed at Sun Microsystems, Inc. headed by James Gosling with his team. The main feature of Java, *the Platform independent*, is the intermediate code that can be recognized by all Java Virtual Machines (JVMs) running under different platforms. Due to the rapid growth, the Sun Microsystems released the second generation of Java with three major divisions: J2SE (Java 2 Platform Standard Edition), J2EE (Java 2 Platform Enterprise Edition) and J2ME (Java 2 Platform Micro Edition). The J2SE consists of Application Programming Interfaces (APIs) needed to build the Java applications and applets. The J2ME supports the development of Java applications for mobile/ wireless devices. And, the J2EE is a replacement for the traditional CGI (Common Gateway Interface) technologies and is used to build dynamic and enterprise web applications. The J2EE components are modified and enhanced time to time, and released to the users. Due to maturity, stability, scalability, security and enhanced features, the generation indicator (the number "2" in J2EE) is removed and now J2EE is referred as JEE (Java Enterprise Edition). The latest version of this Java Enterprise Edition is Java EE 6.

Thus, JEE is one part of Java technologies, used for developing web and enterprise applications and acts as an alternative to CGI technologies.

1.2 COMPONENTS OF JEE

The core components of JEE include Java Servlets, Java Server Pages (JSP), and Enterprise Java Beans (EJB). JEE also consists of several significant APIs including the list given below.

Component Technologies

- ⇒ Java Servlets -Dynamic web-page development
- ⇒ Java Server Pages (JSP) –Dynamic web-page development
- ⇒ Enterprise JavaBeans (EJB) –Middleware component service

Service Technologies

- ⇒ Java Database Connectivity (JDBC) –Database Connectivity to Java applications
- ⇒ Java Message Service (JMS) –Asynchronous messaging service
- ⇒ Java Transaction API (JTA) –Distributed transaction management
- ⇒ Java Mail –e-Mail service
- ⇒ JavaBeans Activation Framework (JAF)
  - – Dynamically register / access the types of arbitrary data and actions on them. Used by Java Mail.
- ⇒ Java Authentication and Authorization Service (JAAS) –Java code based security
- ⇒ Java Naming and Directory Interface (JNDI)
  - – Access to Naming and Directory services (Windows registry, Lightweight Directory Access Protocol –LDAP, Novell Directory Services –NDS, CORBA, RMI, EJB, Database Connection Objects …).

Integration Technologies

- ⇒ Java API for XML Processing (JAXP) –Java XML Programming API
- ⇒ Java Connector Architecture (JCA)
  - – Connecting application servers and Enterprise Information Systems (EIS)
- ⇒ Java Interface Definition Language (IDL) and Web Services
  - – Defining distributed objects. Implemented in Distributed Systems like CORBA.

Communication Technologies

- ⇒ RMI-IIOP
  (**R**emote **M**ethod **I**nvocation and **I**nternet **I**nter **O**RB (Object Request Broker) **P**rotocol)
  – Provides CORBA Capabilities to Java Platform

The following are important new / enhanced features added in Java EE over the time.

- ⇒ JPA (Java Persistence API) –Manipulating RDBMS in Java programs
- ⇒ JSF (Java Server Faces) –Web application framework
- ⇒ JSTL (Java Server Pages Standard Tag Library) –Provides additional tags to JSP tags
- ⇒ JAXR (Java API for XML Registries) –API to access to various kinds of metadata registries
- ⇒ JMX (Java Management Extensions) –Managed Beans (MBeans) for managing application, hardware devices, and service oriented networks …
- ⇒ JAXB (Java Architecture for XML Binding) –Map Java objects to XML and vice versa.

Some of the JEE 6 supported technologies are given below with version number of each individual technology.

- Java Servlet 3.0
- Java Server Pages (JSP) 2.2
- JSP Standard Tag Library (JSTL) 1.2
- Java Server Faces (JSF) 2.0
- Enterprise Java Beans 3.1
- Java Message Service (JMS) 1.1
- Java Transaction API (JTA) 1.1
- Java Mail 1.4
- Web Beans 1.0 and Dependency Injection for Java 1.0
- Bean Validation 1.0
- Java EE Connector Architecture (JCA) 1.6
- Java Persistence API (JPA) 2.0
- Common Annotations for Java Platform 1.1
- Java EE Application Deployment 1.2a
- J2EE Management 1.1
- Java API for RESTful Web Services (JAX-RS) 2.2
- Java API for XML-Based Web Service (JAX-WS)
- Java API for XM_-Based RPC (JAX-RPC) 1.1
- Java Architecture for XML Binding (JAXB) 2.2
- Java APIs for XML Messaging 1.3
- Java API for XML Registries (JAXR) 1.0
- JEE 6 related specifications in JSE (Java Standard Edition).
- Java Database Connectivity (JDBC) 4.0
- Java API for XML Processing (JAXP) 1.3
- JavaBeans Activation Framework (JAF) 1.1
- Java Management Extensions (JMX) 2.0
- Streaming API for XML (StAX) 1.0

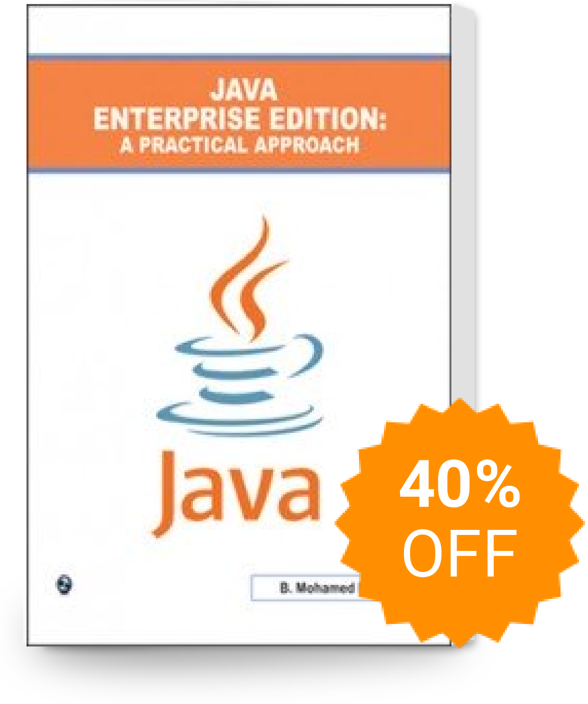
1.3 CLIENT-SERVER TECHNOLOGY

The Internet works on peer-to-peer technology, that is, all the machines are servers as well as clients. Data are stored at the servers; whatever a client request the server, then the server will respond.

If you are browsing a web-site, say *http://www.yahoo.com*, then your machine (the browser that you are using, if we say exactly) is a client that requests data from the server, which is in this example, yahoo machine (the web-server software running at the server, if we say exactly). The welcome file (normally "*index.html*" or "*default.html*") will be invoked by default if you are not mentioned the file-name in the URL address. Actually, the domain name, *www.yahoo.com*, will be converted into its corresponding IP address with the help of Domain Name Server (DNS).

HTML is a HyperText Markup Language that provides GUI input and output elements, such as Text box, Check box, Radio groups, List, etc., for the web pages. But it does not have the capability to process

Java Enterprise Edition: A Practical Approach By B. Mohamed Ibrahim



Publisher : Laxmi Publications ISBN : 9789381159392 Author : B. Mohamed Ibrahim