

*A Quick Reference for Simplified
Enterprise Java Development*



Java EE 6

Pocket Guide

O'REILLY®

Arun Gupta

Java EE 6 Pocket Guide

This guide provides an overview of Java Enterprise Edition 6's main technologies and includes extensive, easy-to-understand code samples that demonstrate the platform's many improvements. You'll quickly understand how Java EE 6 simplifies the process of developing and deploying web and enterprise applications.

- Explore what's new in Java EE 6, including Contexts and Dependency Injection and the Java API for RESTful Web Services
- Discover how Java EE 6 features relate to design patterns in web and enterprise applications
- Get the specifications for making your application Java EE compliant
- Learn about revisions to Enterprise JavaBeans, JavaServer Faces, and other components
- Find out how Java EE 6 profiles change the platform's "one size fits all" approach
- Get started with Java EE 6 development and deployment, using NetBeans IDE and GlassFish

Arun Gupta, a Java evangelist working at Oracle, has more than 16 years of experience working with the Java platform. He has been with the Java EE team since its inception and has contributed to all releases.

oreilly.com

Twitter: @oreillymedia

facebook.com/oreilly

US \$14.99 CAN \$15.99

ISBN: 978-1-449-33668-4



Java EE 6

Pocket Guide

Arun Gupta

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

Java EE 6 Pocket Guide

by Arun Gupta

Copyright © 2012 Arun Gupta. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://my.safaribooksonline.com>). For more information, contact our corporate/institutional sales department: 800-998-9938 or corporate@oreilly.com.

Editors: Mike Loukides and Meghan Blanchette

Copyeditor: Emily Quill

Production Editor: Kristen Borg

Proofreader: Kiel Van Horn

Indexer: Lucie Haskins

Cover Designer: Karen Montgomery

Interior Designer: David Futato

Illustrator: Rebecca Demarest

September 2012: First Edition.

Revision History for the First Edition:

2012-09-10 First release

See <http://oreilly.com/catalog/errata.csp?isbn=9781449336684> for release details.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. *Java EE 6 Pocket Guide*, the image of a jellyfish (*Favonia octonema*), and related trade dress are trademarks of O'Reilly Media, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc., was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

ISBN: 978-1-449-33668-4

[M]

1347298110

To Menka,
my lovely wife and
best friend. Your
support and encour-
agement make our
lives fun and
meaningful.

To
Aditya and Mihir,
my two joyful boys,
for playing with me
and keeping me
charged.

Contents

Preface	ix
Chapter 1: Java Platform, Enterprise Edition	1
Introduction	1
Deliverables	4
What's New in Java EE 6	8
Chapter 2: Managed Beans	13
Define and Use a Managed Bean	14
Lifecycle Callback	15
Chapter 3: Servlets	17
Servlets	17
Servlet Filters	23
Event Listeners	25
Asynchronous Support	29
Web Fragments	31
Security	33
Resource Packaging	35
Error Mapping	37
Handling Multipart Requests	37

Chapter 4: Java Persistence API	39
Entities	40
Persistence Unit, Persistence Context, and Entity Manager	43
Create, Read, Update, and Delete Entities	46
Validating the Entities	50
Transactions and Locking	52
Caching	54
Chapter 5: Enterprise JavaBeans	57
Stateful Session Beans	58
Stateless Session Beans	60
Singleton Session Beans	62
Message-Driven Beans	64
Portable Global JNDI Names	66
Transactions	67
Asynchronous	70
Timers	71
Embeddable API	76
EJB.Lite	77
Chapter 6: Contexts and Dependency Injection	79
Injection Points	80
Qualifier and Alternative	81
Producer and Disposer	84
Interceptors and Decorators	86
Scopes and Contexts	90
Stereotypes	92
Events	94
Portable Extensions	95
Chapter 7: JavaServer Faces	99
Facelets	100

Resource Handling	104
Composite Components	105
Ajax	110
HTTP GET	113
Server and Client Extension Points	115
Navigation Rules	118
Chapter 8: SOAP-Based Web Services	121
Web Service Endpoints	123
Provider-Based Dynamic Endpoints	126
Endpoint-Based Endpoints	128
Web Service Client	129
Dispatch-Based Dynamic Client	132
Handlers	134
Chapter 9: RESTful Web Services	137
Simple RESTful Web Services	138
Binding HTTP Methods	140
Multiple Resource Representations	143
Binding Request to a Resource	144
Mapping Exceptions	147
Chapter 10: Java Message Service	149
Sending a Message	152
Quality of Service	155
Receiving a Message Synchronously	156
Receiving a Message Asynchronously	158
Temporary Destinations	159
Chapter 11: Bean Validation	161
Built-in Constraints	162
Defining a Custom Constraint	165

Validation Groups	169
Integration with JPA	172
Integration with JSF	174
A: Getting Started with Java EE 6 Development and Deployment	177
B: Further Reading	179
Index	183

Preface

The Java EE 6 platform has taken ease-of-development in enterprise Java programming to new heights. This book is directed towards the audience who wants to get a quick overview of the platform and to keep coming back to learn the basics.

This book provides an overview of the key specifications in the Java EE 6 platform (one specification per chapter). The main concepts from the different specifications are explained and accompanied by code samples. No prior knowledge of earlier versions of the platform is required. However, some basic understanding of Java is required to understand the code.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, and file extensions.

Constant width

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

Constant width italic

Shows text that should be replaced with user-supplied values or by values determined by context.

Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Java EE 6 Pocket Guide* by Arun Gupta (O'Reilly). Copyright 2012 Arun Gupta, 978-1-449-33668-4."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.

Safari® Books Online



Safari Books Online (www.safaribooksonline.com) is an on-demand digital library that delivers expert **content** in both book and video form from the world's leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books

Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of [product mixes](#) and pricing programs for [organizations](#), [government agencies](#), and [individuals](#). Subscribers have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like O'Reilly Media, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology, and dozens [more](#). For more information about Safari Books Online, please visit us [online](#).

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at <http://oreil.ly/javaEE6-pocketguide>.

To comment or ask technical questions about this book, send email to bookquestions@oreilly.com.

For more information about our books, courses, conferences, and news, see our website at <http://www.oreilly.com>.

Find us on Facebook: <http://facebook.com/oreilly>

Follow us on Twitter: <http://twitter.com/oreillymedia>

Watch us on YouTube: <http://www.youtube.com/oreillymedia>

Acknowledgments

This book was not possible without support from a multitude of people.

First and foremost, many thanks to O'Reilly for trusting in me and providing an opportunity to write this book. Their team provided excellent support throughout the editing, reviewing, proofreading, and publishing process.

At O'Reilly, Michael Loukides helped me with bootstrapping the book. Meghan Blanchette provided excellent editorial help throughout all the stages, helping with interim reviews, providing feedback on styling, arranging technical reviews, and connecting me with the rest of the team when required. Jessica Hosman helped me in getting started and guided the authoring process.

Emily Quill and Kristen Borg helped with copyediting and making sure to provide the finishing touches. And thanks to the rest of the O'Reilly team with whom I did not interact directly, but were helping in many other ways.

The detailed proofreading and technical review by Markus Eisele (@myfear, <http://blog.eisele.net>), John Yeary (@jyeary, <http://javaevangelist.blogspot.com>), and Bert Ertman (@BertErtman, <http://bertertman.wordpress.com>) ensured that the relevant content was covered accurately. Their vast experience and knowledge showed in the depth of their comments.

I am grateful for the numerous discussions with developers around the world that helped me understand the technology better. Thanks to my colleagues at Oracle and the JSR specification leads for explaining the intended use cases of different technologies. And thanks to everybody else in my life, who provided much-needed breaks from book writing.

Java Platform, Enterprise Edition

Introduction

The Java Platform, Enterprise Edition (Java EE) provides a standards-based platform for developing web and enterprise applications. These applications are typically designed as multitier applications, with a frontend tier consisting of web frameworks, a middle tier providing security and transactions, and a backend tier providing connectivity to a database or a legacy system. The Java EE platform defines APIs for different components in each tier, and also provides some additional services such as naming, injection, and resource management that span across the platform. Each component is defined in a separate specification that also describes the API, javadocs, and expected behavior.

The Java Platform, Enterprise Edition 6 (Java EE 6) was released in December 2009 and provides a simple, easy-to-use, and complete stack for building such applications. The previous version of the platform, Java EE 5, took the first step in providing a simplified developer experience. The Java EE 6 platform further improves upon the developer productivity features and also adds a lot more functionality.

The three main goals of the platform are:

Ease of use

The Java EE 6 platform takes ease of use to new heights by extensively using convention over configuration and heavy use of annotations on a Plain Old Java Object (POJO). Adding `@Stateless`, `@Stateful`, or `@Singleton` to a POJO makes it an Enterprise JavaBean. Further, this could be easily packaged in a WAR file instead of a special packaging of JAR or EAR. Servlets are POJOs as well, annotated with `@WebServlet`. Deployment descriptors like `web.xml` and `faces-config.xml` are optional in most cases; the information typically specified in deployment descriptors is now captured in annotations. There are default rules of navigation from one page of JSF to another. Publishing a POJO as a RESTful web service is equivalent to adding an `@Path` annotation on a POJO.

Making deployment descriptors optional, using convention over configuration, and relying heavily on annotations makes the Java EE 6 platform easy to use and, above all, less verbose.

Lightweight

There are 31 component specifications in the Java EE 6 platform, as listed in Appendix EE.6 of the platform specification. These components include Enterprise JavaBeans (EJB), Servlets, JavaServer Faces (JSF), Java API for RESTful Web Services (JAX-RS), and many others. Building a typical enterprise application may not require all the components. Also, some of the technologies like Java API for XML Registries (JAXR) or Java API for XML-based RPC (JAX-RPC) were very relevant when introduced in the platform. Now they have either been replaced by better components, such as Java API for XML Web Services (JAX-WS), or are no longer used.

The Java SE Expert Group defined a two-step process for removing features from the platform. In this process, known as *pruning*, a feature is marked as optional (referred to as *proposed optional*) in one release, and then a

subsequent release can decide to either remove the feature, retain it as a required component, or leave it in the *proposed removal* state. The Java EE Expert Group used that process and targeted some features for pruning. This is analogous to trimming rose bushes in the beginning of each year so that fresh blossoms can grow. Pruning unused features ensures that even with new feature additions, the platform will remain simple and lightweight.

The Java EE platform also introduces the notion of *profiles*. A profile represents a configuration of the platform suited to a particular class of applications. A profile may be a subset or superset of the technologies in the platform. The Java EE 6 Web Profile is defined as a separate specification in the platform, and is defined as a subset of technologies contained in the platform and targeted toward the developers of modern web applications. This breaks away from the “one size fits all” approach of previous releases. And although it’s a proper subset, it still offers a reasonably complete stack composed of standard APIs, and it’s capable out-of-the-box for addressing a wide variety of web applications. The web profile allows developers to build web applications quickly and prevents the proliferation of custom web stacks for easier maintainability. Additional profiles can be defined by following the rules of the Java Community Process (JCP).

Together, pruning and web profiles make the Java EE 6 platform lightweight and simple to maintain.

Extensibility

The platform provides a rich set of functionality to create enterprise applications. However, it’s a common practice to include third-party frameworks to supplement or complement functionality in the platform. These frameworks require registration of a `ServletListener`, `ServletFilter`, or similar component so that they are recognized by the runtime. The Servlet specification defines a *web fragment* mechanism by which these entry points to the framework are defined in the framework library. The Servlet

containers then register the framework, relieving the developer of the burden. This allows these frameworks to be treated as first-class citizens of the platform.

In addition, the Contexts and Dependency Injection (CDI) specification defines a portable extension mechanism that allows you to extend the capabilities of the platform in different ways, for example by providing certain predefined scopes. A new scope can be easily defined and included with any Java EE 6–compliant application server using the portable extensions method.

Specifications like CDI, JavaServer Faces 2, Java API for RESTful Services, Java Persistence API 2, and Servlets 3 make the Java EE 6 platform more powerful. This book will provide an overview of the main technologies included in the platform, and easy-to-understand code samples will be used throughout to demonstrate improvements in Java EE 6.

Deliverables

The Java EE 6 platform was developed as Java Specification Request 316 or JSR 316 (<http://jcp.org/en/jsr/detail?id=316>) following Java Community Process (JCP) 2.7. The JCP process defines three key deliverables for any JSR:

Specification

A formal document that describes the proposed component and its features.

Reference Implementation (RI)

Binary implementation of the proposed specification. The RI helps to ensure that the proposed specifications can be implemented in a binary form and provides constant feedback to the specification process.

Technology Compliance Kit (TCK)

A set of tests that verify that the RI is in compliance with the specification. This allows multiple vendors to provide compliant implementations.

Java EE 6 consists of the platform specification that defines requirements across the platform. It also consists of the following component specifications:

Web Technologies

- JSR 45: Debugging Support for Other Languages
- JSR 52: Standard Tag Library for JavaServer Pages (JSTL) 1.2
- JSR 245: JavaServer Pages (JSP) 2.2 and Expression Language (EL) 1.2
- JSR 314: JavaServer Faces (JSF) 2.0
- JSR 315: Servlet 3.0

Enterprise Technologies

- JSR 250: Common Annotations for the Java Platform 1.1
- JSR 299: Contexts and Dependency Injection (CDI) for the Java EE Platform 1.0
- JSR 303: Bean Validation 1.0
- JSR 316: Managed Beans 1.0
- JSR 317: Java Persistence API (JPA) 2.0
- JSR 318: Enterprise JavaBeans (EJB) 3.1
- JSR 318: Interceptors 1.1
- JSR 322: Java EE Connector Architecture 1.6
- JSR 330: Dependency Injection for Java 1.0
- JSR 907: Java Transaction API (JTA) 1.1
- JSR 914: Java Message Server (JMS) 1.1
- JSR 919: JavaMail 1.4

Web Service Technologies

- JSR 67: Java APIs for XML Messaging (JAXM) 1.3
- JSR 93: Java API for XML Registries (JAXR) 1.0
- JSR 101: Java API for XML-based RPC (JAX-RPC) 1.1
- JSR 109: Implementing Enterprise Web Services 1.3

- JSR 173: Streaming API for XML (StAX) 1.0
- JSR 181: Web Services Metadata for the Java Platform 2.0
- JSR 222: Java Architecture for XML Binding (JAXB) 2.2
- JSR 224: Java API for XML Web Services (JAX-WS) 2.2
- JSR 311: Java API for RESTful Web Services (JAX-RS) 1.1

Management and Security Technologies

- JSR 77: J2EE Management API 1.1
- JSR 88: Java Platform EE Application Deployment API 1.2
- JSR 115: Java Authorization Contract and Containers (JACC) 1.3
- JSR 196: Java Authentication Service Provider Interface for Containers (JASPIC) 1.0

The different components work together to provide an integrated stack, as shown in [Figure 1-1](#).

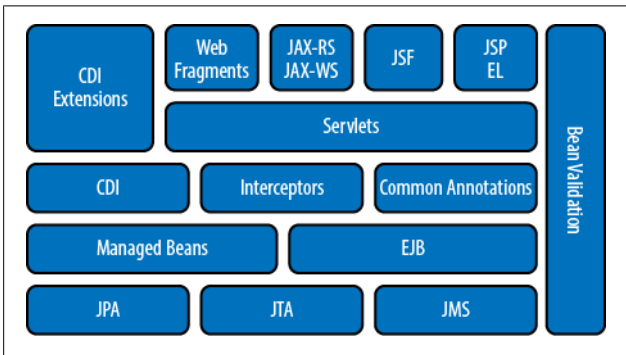


Figure 1-1. Java EE 6 architecture

In Figure 1-1:

- JPA, JTA, and JMS provide the basic services such as database access, transactions, and messaging.
- Managed Beans and EJB provide a simplified programming model using POJOs to use the basic services.
- CDI, Interceptors, and Common Annotations provide concepts that are applicable to a wide variety of components, such as type-safe dependency injection, addressing cross-cutting concerns using interceptors, and a common set of annotations.
- CDI Extensions allow you to extend the platform beyond its existing capabilities in a standard way.
- Web services using JAX-RS and JAX-WS, JSF, JSP, and EL define the programming model for web applications. Web Fragments allow automatic registration of third-party web frameworks in a very natural way.
- Bean Validation provides a standard means to declare constraints and validate them across different technologies.

JAX-RPC (JSR 101), JAXR (JSR 93), EJB Entity Beans (part of JSR 153), and Java EE Application Deployment (JSR 88) are marked for pruning in this version of the platform.

The RI of Java EE 6 is built in the GlassFish Community. The GlassFish Server Open Source Edition provides a full Java EE 6–compliant, free, and open source application server. It is also available in a Web Profile distribution and can be downloaded from <http://glassfish.org>. The application server is easy to use (zip installer and NetBeans/Eclipse/IntelliJ integration), lightweight (downloads starting at 30 MB, small disk/memory footprint), and modular (OSGi-based, containers start on demand). It also provides clustering with high availability and centralized administration using CLI, web-based administration console, and REST management/monitoring APIs. The Oracle GlassFish Server is Oracle’s commercially supported GlassFish server distribution and can be downloaded from