Document Information

Preface

Part I Introduction

1. Overview
2. Using the Tutorial Examples

Part II The Web Tier

3. Getting Started with Web Applications
4. JavaServer Faces Technology
5. Introduction to Facelets
6. Expression Language
7. Using JavaServer Faces Technology in Web Pages
8. Using Converters, Listeners, and Validators
9. Developing with JavaServer Faces Technology
10. JavaServer Faces Technology: Advanced Concepts
11. Using Ajax with JavaServer Faces Technology
12. Composite Components: Advanced Topics and Example
13. Creating Custom UI Components and Other Custom Objects
14. Configuring JavaServer Faces Applications
15. Java Servlet Technology

What Is a Servlet?

Servlet Lifecycle

Handling Servlet Lifecycle Events

Defining the Listener Class

Handling Servlet Errors

Sharing Information

Using Scope Objects

Controlling Concurrent Access to Shared Resources

Creating and Initializing a Servlet

Writing Service Methods

Getting Information from Requests

Constructing Responses

Filtering Requests and Responses

Programming Filters

Programming Customized Requests and Responses

Specifying Filter Mappings

To Specify Filter Mappings Using NetBeans IDE

Invoking Other Web Resources

Including Other Resources in the Response

Transferring Control to Another Web Component

Accessing the Web Context

Maintaining Client State

The Java EE 6 Tutorial

Sharing Information

Web components, like most objects, usually work with other objects to accomplish their tasks. Web components can do so by

- Using private helper objects (for example, JavaBeans components).
- Sharing objects that are attributes of a public scope.
- Using a database.
- Invoking other web resources. The Java Servlet technology mechanisms that allow a web component to invoke other web resources are described in Invoking Other Web Resources.

Using Scope Objects

Collaborating web components share information by means of objects that are maintained as attributes of four scope objects. You access these attributes by using the `getAttribute` and `setAttribute` methods of the class representing the scope. Table 15-2 lists the scope objects.

Table 15-2 Scope Objects

Scope Object	Class	Accessible from
Web context	`javax.servlet.ServletContext`	Web components within a web context. See Accessing the Web Context.
Session	`javax.servlet.http.HttpSession`	Web components handling a request that belongs to the session. See Maintaining Client State.
Request	Subtype of `javax.servlet.ServletRequest`	Web components handling the

Accessing a Session
Associating Objects with a Session
Session Management
To Set the Timeout Period Using NetBeans IDE
Session Tracking
Finalizing a Servlet
Tracking Service Requests
Notifying Methods to Shut Down
Creating Polite Long-Running Methods
The mood Example Application
Components of the mood Example Application
Running the mood Example
To Run the mood Example Using NetBeans IDE
To Run the mood Example Using Ant
Further Information about Java Servlet Technology
16. Uploading Files with Java Servlet Technology
17. Internationalizing and Localizing Web Applications
Part III Web Services
18. Introduction to Web Services
19. Building Web Services with JAX-WS
20. Building RESTful Web Services with JAX-RS
21. JAX-RS: Advanced Topics and Example
Part IV Enterprise Beans
22. Enterprise Beans
23. Getting Started with Enterprise Beans
24. Running the Enterprise Bean Examples
25. A Message-Driven Bean Example
26. Using the Embedded Enterprise Bean Container
27. Using Asynchronous Method Invocation in Session Beans
Part V Contexts and Dependency Injection for the Java EE Platform
28. Introduction to Contexts and Dependency Injection for the Java EE Platform
29. Running the Basic Contexts and Dependency Injection Examples
30. Contexts and Dependency Injection for the Java EE Platform: Advanced Topics
31. Running the Advanced Contexts and Dependency Injection Examples
Part VI Persistence
32. Introduction to the Java Persistence API
33. Running the Persistence Examples
34. The Java Persistence Query Language
35. Using the Criteria API to Create Queries
36. Creating and Using String-Based Criteria

		request.
Page	`javax.servlet.jsp.JspContext`	The JSP page that creates the object.

Controlling Concurrent Access to Shared Resources

In a multithreaded server, shared resources can be accessed concurrently. In addition to scope object attributes, shared resources include in-memory data, such as instance or class variables, and external objects, such as files, database connections, and network connections.

Concurrent access can arise in several situations:

- Multiple web components accessing objects stored in the web context.
- Multiple web components accessing objects stored in a session.
- Multiple threads within a web component accessing instance variables. A web container will typically create a thread to handle each request. To ensure that a servlet instance handles only one request at a time, a servlet can implement the `SingleThreadModel` interface. If a servlet implements this interface, no two threads will execute concurrently in the servlet's service method. A web container can implement this guarantee by synchronizing access to a single instance of the servlet or by maintaining a pool of web component instances and dispatching each new request to a free instance. This interface does not prevent synchronization problems that result from web components' accessing shared resources, such as static class variables or external objects.

When resources can be accessed concurrently, they can be used in an inconsistent fashion. You prevent this by controlling the access using the synchronization techniques described in the Threads lesson at http://docs.oracle.com/javase/tutorial/essential/concurrency/index.html in *The Java Tutorial, Fourth Edition*, by Sharon Zakhour et al. (Addison-Wesley, 2006).

Queries

37. Controlling Concurrent Access to Entity Data with Locking
38. Using a Second-Level Cache with Java Persistence API Applications

Part VII Security

39. Introduction to Security in the Java EE Platform
40. Getting Started Securing Web Applications
41. Getting Started Securing Enterprise Applications
42. Java EE Security: Advanced Topics

Part VIII Java EE Supporting Technologies

43. Introduction to Java EE Supporting Technologies
44. Transactions
45. Resources and Resource Adapters
46. The Resource Adapter Example
47. Java Message Service Concepts
48. Java Message Service Examples
49. Bean Validation: Advanced Topics
50. Using Java EE Interceptors

Part IX Case Studies

51. Duke's Bookstore Case Study Example
52. Duke's Tutoring Case Study Example
53. Duke's Forest Case Study Example

Index