Developing Applications for the Java EE 6 Platform

Course description

The Developing Applications for the Java(TM) EE Platform training helps you develop the knowledge to build and deploy enterprise applications that comply with Java(TM) Platform, Enterprise Edition 6 technology standards. This course is ideal for Sun(TM) Certified Java technology programmers who want to develop enterprise applications that conform to the Java EE platform standards.

Student Take away

- Study Material
- Learning stuff
- Sample project for practice

Developing Applications for the Java EE 6 Platform Online Training Curriculum

An Introduction to Java EE Technologies

- Describe the different Java platforms and versions
- Describe the needs of enterprise applications
- Introduce the Java EE APIs and services
- Certifications Paths
- Introducing Applications Servers
- Enterprise Modules

Developing Servlets

- Describe the servlet API
- Servlet configuration through annotations and deployment descriptors
- Use the request and response APIs
- Servlets as controllers

EJB Overview

- EJB types: Session Beans
- EJB types: Message Driven beans
- Java Persistence API as a replacement for Entity EJBs
- Describe the role of EJBs in a Java EE application
- EJB lite

Implementing a Transaction Policy

- Describe transaction semantics
- Compare programmatic and declarative transaction scoping
- Use the Java Transaction API (JTA) to scope transactions programmatically

Implementing a Transaction Policy continues...

- Implement a container-managed transaction policy
- Support optimistic locking with the versioning of entity components
- Support pessimistic locking of entity components
- Using transactions with the web profile

Web Service Model

- Describe the role of web services
- Web service models
- List the specifications used to make web services platform independent
- Describe the Java APIs used for XML processing and web services

Enterprise Application Architecture

- Design Patterns
- Model View Controller
- Synchronous and Asynchronous communication
- Network Topologies and Clustering
- Layering (client, presentation, service, integration, persistence)

Developing With Java Server Pages Technology

- Evaluate the role of JSP technology as a presentation mechanism
- Author JSP pages
- Process data received from servlets in a JSP page
- Brief introduction to the JSTL and EL

Implementing EJB 3.0 Session Beans

- Compare stateless and stateful behaviour
- Describe the operational characteristics of a stateless session bean
- Describe the operational characteristics of a stateful session bean
- Describe the operational characteristics of a singleton session bean
- Create session beans
- Package and deploy session beans
- Create session bean clients

Developing Asynchronous Java EE Applications and Messaging

- The need for asynchronous execution
- JMS technology introduction
- List the capabilities and limitations of Java EE components as messaging producers and consumers
- JMS and transactions
- JMS administration

Implementing Java EE Web Services with JAX-WS and JAX-RS

- Describe endpoints supported by the Java EE 6 platform
- Developing Web Services with Java
- Creating Web Service Clients with Java

Web Technology Overview

- Describe the role of web components in a Java EE application
- Define the HTTP request-response model
- Compare Java servlets, JSP, and JSF
- Brief introduction to technologies not covered in detail

Java Server Faces

- The JSF model explained
- Adding JSF support to web applications
- Using the JSF tag libraries
- Configuring JSF page navigation
- JSF Managed beans
- JSF Conversion, Validation, and Error Handling

The Java Persistence API

- The role of the Java Persistence API in a Java EE application
- Object Relational Mapping
- Entity class creation
- Using the Entity Manager API
- The life cycle and operational characteristics of Entity components
- Persistent Units and Packaging

Developing Message-Driven Beans

- Describe the properties and life cycle of message-driven beans
- Create a JMS message-driven bean

Implementing a Security Policy

- Exploit container-managed security
- Define user roles and responsibilities
- Create a role-based security policy
- Use the security API
- Configure authentication in the web tier