Oracle Fusion Middleware 11g: Build Applications with ADF I

Duration: 5 Days

What you will learn

Java EE is a standard, robust, scalable, and secure platform that forms the basis for many of today's enterprise applications. Oracle Application Development Framework (Oracle ADF) is an innovative, yet mature Java EE development framework that is directly supported and enabled by Oracle JDeveloper 11g. Oracle ADF simplifies Java EE development by minimizing the need to write code that implements the application's infrastructure allowing developers to focus on the features of the actual application.

This course is aimed at developers who want to build Java EE applications using Oracle ADF. Participants use Oracle JDeveloper 11g Release 1 Patch Set 1 to build, test and deploy an end-to-end web application. The data model is built with ADF Business Components and the user interface with ADF Faces. During the course, participants learn to build each part of the application with the Fusion technology stack and then deploy it to WebLogic Server.

Learn To:

Build end-to-end web applications

Develop Java EE components with Oracle ADF

Build rich user interfaces with ADF Faces

Use the new capabilities of Oracle JDeveloper 11g Release 1 Patch Set 1

Audience
Application Developers
J2EE Developer
Java Developer
Java EE Developer

Prerequisites
Required Prerequisites
Familiarity with JDeveloper

Familiarity with XML concepts

Familiarity with basic Java

Course Objectives
Expose the data model in a web application with a rich ADF Faces user interface
Create JSF pages
Use rich client components in JSF pages
Add validation to ADF applications

Secure Web applications
Build and customize a data model by using ADF Business Components

Course Topics

Introduction to Fusion and ADF
Describing Fusion architecture
Explaining how ADF fits into the Fusion architecture
Describing the ADF technology stack (MVC)

Getting Started with JDeveloper
Listing JDeveloper benefits for application development
Using the features of the JDeveloper IDE
Defining IDE preferences
Creating applications, projects, and connections in JDeveloper

Building a Data Model with ADF Business Components
Introducing ADF Business Components
Creating Business Components from tables
Testing the data model

Querying and Persisting Data
Using view objects
Using entity objects to persist data
Synchronizing entity objects with database table changes
Creating associations
Creating updateable view objects
Creating master-detail relationships
Refactoring

Exposing Data
Creating application modules
Using master-detail view objects in application modules
Managing Business Components transactions
Abstracting business services with ADF Model

Declaratively Customizing Data Services
Internationalizing the data model
Editing business components
Modifying default behavior of entity objects
Changing the locking behavior of an application module

Programmatically Customizing Data Services
Generating Java classes
Programmatically modifying the behavior of entity objects
Programmatically modifying the behavior of view objects
Adding service methods to an application module
Using client APIs

Validating User Input
Understanding validation options: Database, Data Model, or UI

Triggering validation execution
Handling validation errors
Using Groovy expressions in validation
Using programmatic validation

Troubleshooting ADF BC Applications
Troubleshooting the business service
Troubleshooting the UI
Using logging and diagnostics
Using the JDeveloper debugger

Understanding UI Technologies
Describing the use of Web browsers and HTML
Describing the function of Servlets and JSPs
Defining JavaServer Faces
Explaining the JSF component architecture and JSF component types
Explaining the purpose of backing beans and managed beans
Describing the JSF life cycle
Explaining how ADF Faces augments the JSF life cycle

Binding UI Components to Data
Creating a JSF page
Adding UI components to a page
Describing the ADF Model layer
Using Expression Language in data bindings
Using a Page Definition file
Examining data binding objects and metadata files
Binding existing components to data
Running and testing the page

Planning the User Interface
Describing the Model-View-Controller design pattern
Differentiating between bounded and unbounded task flows
Creating and converting task flows
Defining control flows
Defining global navigation
Creating routers for conditional navigation
Calling methods and other task flows
Implementing validation in the user interface

Adding Functionality to Pages
Internationalizing the user interface
Using component facets
Displaying tabular data in tables
Displaying hierarchical data in trees
Displaying text or media with icons and images
Defining search forms and display results
Displaying data graphically

Implementing Navigation on Pages
Using ADF Faces navigation components
Using buttons and links

Using menus for navigation
Using breadcrumbs
Using a train component

Achieving the Required Layout
Using complex layout components
Explaining how to use ADF Faces skins
Using dynamic page layout

Ensuring Reusability
Designing for reuse
Using task flow templates
Creating and using page templates
Creating and using declarative components
Creating and using page fragments
Deciding which type of reusable component to use

Passing Values Between UI Elements
Defining the data model to reduce the need to pass values
Using a managed bean to hold values
Using page parameters
Using task flow parameters
Passing values from containing pages to regions

Responding to Application Events
Using managed beans
Coordinating JSF and ADF lifecycles
Using phase and event listeners
Using action listeners and methods
Understanding additional AJAX events

Implementing Transactional Capabilities
Handling transactions with ADF BC
Using task flows to control transactions
Sharing data controls
Handling transaction exceptions
Defining response to the Back button

Implementing Security in ADF BC Applications
Exploring ADF Application security options
Understanding ADF security framework
Enabling users to access resources
Implementing a Login page
Understanding ADF controller authorization
Using Expression Language to extend security capabilities