

Oracle WebCenter Portal 11g: Creating Portals using ADF

Duration: 3 Days

What you will learn

This WebCenter Portal training covers some of the best practices for creating different pieces of portals for WebCenter Portal 11g with the help of Oracle JDeveloper 11g.

Learn To:

Create the structure and look and feel for portals.

Create Content Presenter templates for populating portals with content.

Add external web services to portals by creating task flows for the web services in JDeveloper.

Apply the development lifecycle concepts such as using deployer projects and manifest files for versioning in your future asset development.

Use WebCenter Portal's REST APIs to access and modify portal resources.

Have your portal pages load quicker by following best practices.

Benefits to You

Content Presenter Templates

As a developer, you can create Content Presenter templates for portals and have content contributors use these templates to constantly update the content of portals with desired layouts that you provide with the templates. With these templates the content contributors are not required to know any coding and will be users of the templates.

Development Lifecycle

By following the best practices of a development lifecycle for a task flow in your practices, you can apply that knowledge to any asset that you create in JDeveloper and perform supported methods for asset development. The development lifecycle method makes it easy for teams to work together, because every step is associated with a certain location and it is easy to find the assets, modify or deploy them. This method shows you to version your assets so you do not ever lose your assets. You can create archives of the assets you build and send them to other members of a team to use in their portals.

Web Services

Take advantage of existing external web services and learn to create task flows for them in JDeveloper, create assets from them to include in the resource catalog of WebCenter Portal to and add the web services to your portals.

Audience

Developer

Portal Developer

Related Training

Suggested Prerequisites

Java SE 7 Programming

Oracle Fusion Middleware 11g: Build Applications with ADF I

Oracle WebCenter Portal 11g for Developers

Course Objectives

Divide portal page design mock-ups into components that a developer should build such as page templates, page styles and skins

Create page templates, skins and page styles based on design mock-ups for pages

Create Content Presenter templates in JDeveloper with Site Studio Regions

Create data controls and task flows that consume the data controls in portals

Add web services to portal pages

Use WebCenter Portal's REST API

Course Topics

Course Overview

Discussing Course Objectives and Agenda

Overview of Lab Environment

About Target Audience

About Course Practices

About References and Documentation Guides

Getting Started with Asset Development

Overview of Assets in WebCenter Portal

Discussing A Developer's Key Tasks in JDeveloper and Portal Builder

Overview of WebCenterSpacesResources Project

Setting Up JDeveloper for Asset Development for WebCenter Portal

Explaining Portal Builder Approach for Asset Development

Downloading and Identifying WebCenter Portal Assets in JDeveloper

Creating Navigation

Overview of Portal Navigation

Overview of Navigation Model

Discussing Navigation Visualization

Finding the Default Navigation Model of a Portal

Visualizing the Navigation Model

About Navigation Model Security

About Navigation Model Best Practices

Creating Page Templates

- Overview of Page Templates
- About Page Templates Layout
- About Geometry Management
- Using WebCenter Portal's Out-of-the-Box Page Templates
- Reviewing the Round-trip Development Process for Page Templates
- Creating Page Templates
- Editing Page Templates
- Describing Best Practices for Layout Management

Creating Skins

- Overview of Skins
- Introducing CSS3
- About WebCenter Portal Skin-Specific Selectors
- Overwriting Existing Style Selectors of the Out-of-the-Box Skins
- Discussing Best Practices for Skinning

Planning Site Studio Regions

- Overview of WebCenter Content
- Overview of Site Studio
- Installing Site Studio
- Creating Element Definitions
- Creating Region Definitions
- Creating Data Files from Region Definitions
- Discussing Best Practices for Creating Content Templates

Creating Content Presenter Templates

- Overview of Content Presenter Templates
- Using Content Presenter Templates
- Modifying Existing Out-of-the-Box Templates
- Creating New Content Presenter Templates
- Using Content Display Template Tags
- Referencing Site Studio Region Definitions
- About Best Practices

Creating Responsive Designs

- Overview of Responsive Design
- Discussing Adaptive Layouts vs. Responsive Layouts
- About Adaptive Design Example
- Overview of CSS3 Media Queries
- About Media Query Syntax and Examples
- Using Responsive Content Presenter Templates
- Describing Best Practices for Responsive Design

Creating Data Controls

- About Steps to Present Data
- Overview of Data Controls
- Explaining Model View Controller (MVC) Architecture
- Overview of Fusion Web Applications
- Discussing Web Services Properties
- Invoking Services of a Web Service by Creating Task Flows with Data Controls

Preparing Task Flows for Deployment

- Preparing an ADF Library JAR File
- Deploying the ADF Library JAR File
- Creating a Deployer Project
- Adding the JAR File to the Deployer Project
- Preparing a WAR file for the Deployer project
- Implementing Versioning for Deployment

Deploying and Registering Task Flows to WebCenter Portal

- Deploying a Task Flow to Web Center Portal
- Introducing the WebCenter Portal Server Extension Application
- Registering a Task Flow with WebCenter Portal
- Adding a Task Flow to a Resource Catalog
- Adding a Task Flow to a Portal Page

Using REST APIs of WebCenter Portal

- Overview of REST API
- About RESTful Services
- About REST Resources
- Discussing REST Methods
- Discussing WebCenter Portal's Resource Index with Examples
- About HTTP Status Codes
- Finding the Correct Format to POST Resources in JSON