Java SE 7: Develop Rich Client Applications

Duration: 5 Days

What you will learn

The Java SE 7: Develop Rich Client Applications training takes you through the process of designing a rich client application using Java SE 7 and Java FX 2. Using the Model-View-Controller (MVC) pattern and a case study approach, you'll learn to analyze, design and develop the user interface, connect the user interface to a database and finally connect the user interface to a RESTful web service.

Learn To:

Create a graphical user interface using Java FX.
Connect a Java FX GUI to database using JPA.
Connect a Java FX GUI to a RESTful web service.
Package and deploy a Java FX application.
Sign a Java FX application.
connect their application to a RESTful web service using JAX-RS and the Jersey API.

Learn How to Create a User Interface

The user interface is created using key Java FX components including layouts, UI controls, data structures like ObservableList, charts, smart tables, CSS and JavaFX concurrency libraries. You'll also learn to add two tier and three tier features to your application by connecting to a database using the Java Persistence API (JPA).

Additional Topics Include:

Packaging and deploying your application.
Developing secure applications.
Signing an application and authentication.
Adding logging to your application.
Implementing unit testing with JUnit.

Audience

Application Developers
Developer
Java Developer
Java EE Developer

Related Training

Required Prerequisites

Java SE7 Fundamentals

Java SE 7 Programming

Course Objectives

Implement a rich client application (RIA) from the ground up

Create a JavaFX GUI using controls, layouts, charts, smart tables, and CSS

Implement event handling in a JavaFX application

Use JavaFX visual effects, animations, media, and a web view control in a JavaFX application

Implement concurrency and binding to a JavaFX application

Leverage Java Persistence API (JPA) in a Java SE application

Create two-tier and three-tier Java technology applications

Connect your application to a REST web service

Package and deploy a Java SE application

Secure a Java SE application

Sign a Java SE application

Implement the Logging API to generate log messages in GUI

Implement unit testing using JUnit

Apply Model View Controller (MVC) design pattern to create reusable classes

Course Topics

Introduction
Providing an overview of Rich Client applications(RIAs)
Providing an overview of JavaFX

The Broker Tool Application
Describing an overview of BrokerTool
Describing BrokerTool database schema
Providing an overview of Henley Automobile application

JavaFX Overview
Demonstrating Simple JavaFX Applications
What is JavaFX?
Exploring JavaFX API

Understanding JavaFX Scene Graph
How to create a JavaFX app?
Creating JavaFX FXML Application
Comparing JavaFX with Swing
Overview of JavaFX features

Generics and JavaFX Collections
Reviewing Java Generics syntax
Reviewing Java Generic Collection objects
Reviewing JavaFX Collection's ObservableList and ObservableMap

UI Controls, Layouts, Charts, and CSS
Understanding Scene Graph in depth
Using UI controls in JavaFX application
Using Layout features in JavaFX application
Using Charts in JavaFX application
Understanding the usage of CSS in JavaFX application
Adding events to JavaFX controls

Visual Effects, Animation, Web View, and Media
Using Visual Effects in JavaFX application
Using Animation and transition features in JavaFX application
Describing the benefits of using WebView and WebNode
Describing the implementation of Multimedia in JavaFX

JavaFX Tables and Client GUI
Creating smart Table
Describing the BrokerTool app interface
Determining which JavaFX components to use in the BrokerTool interface
Displaying BrokerTool data and determine which charts and tables to use to display data
Applying CSS to a JavaFX application

JavaFX Concurrency and Binding
Describing properties and binding in JavaFX
Implementing Threading and Concurrency in JavaFX

Java Persistence API (JPA)
Understanding JPA concepts
Understanding Components of JPA architecture
What is Transactions ?
Performing CRUD operations using Entity and Queries

Applying the JPA
Identifying Entity Relationships
Using Criteria API in JavaFX application
Applying JPA in HenleyApp application
Applying two-tier design

Implementing a Multi-Tier Design with RESTful Web Services
Comparing Three-tier design versus Two-tier design
Describing JAX-RS web services
Using JAX-RS web services in the HenleyServer application

Connecting to a RESTful Web Service
Testing REST web service with HTTP and HTML
Developing JAX-RS web service clients
Identifying how to develop a Jersey RESTful client
Reviewing the implementation of Web service clients of HenleyApp

Packaging and Deploying Applications
Using jar to package up Java applications
Deploying applications
Deploying Embedded applications
Deploying Jar using Java Web Start
Using an Installer
Deploying Rich Internet using Deployment Toolkit

Developing Secure Applications
Describing the Aspects of security
Describing Fundamental secure coding concepts
Avoiding common Injection and inclusion attacks
Protecting Confidential data
Limiting the accessiblity of classes
Understanding Mutability
Listing Security resources available on the Internet

Signing an Application and Authentication
Describing Public and private key encryption
Describing Digital Signatures
Introducing SSL/TLS
Understanding HTTP Authentication mechanism
Using HTTPS with an application

Logging
Overview of Java Logging API
Creating a Logger Object
Setting Log levels
Reviewing Logging methods
Configuring Logger Handlers and formatters
Using Logger Configuration

Implementing Unit Testing and Using Version Control
Understanding Unit Testing, Test Cases and features of JUnit
Understanding and Writing JUnit test cases
Using NetBeans support for JUnit
Using Version control system