

# AJAX in Java Applications

CSDT202

This technical overview is intended for managers, project leaders and developers who are interested in using RAD to build their applications. IBM Rational Application Developer (RAD) is the new generation of Java application development environments built on the Eclipse open platform ([www.eclipse.org](http://www.eclipse.org)). RAD has many new features that the traditional IDEs do not provide and implementing this development can be overwhelming without the proper knowledge base.

---

## Audience

- This course is intended for managers, project leaders and developers who are interested in using WSAD to build their applications.

## Prerequisites

- Understanding of the Ajax programming model is required. This course is intended to follow [Course 201, Developing Ajax Applications](#), or some equivalent training.
- Significant Java programming experience is required.
- Basic servlets and JSP programming experience is required.
  - JSF experience will be helpful for the final unit, but is not required.

## Course Length

- Two Days

## Learning Objectives

- Identify architectural and design challenges inherent in the use of Ajax for Java applications.
- Define a request-handling path that best addresses the needs of a given Java web application.
- Manage script content and bind scripts to graphical widgets using JSP.
- Use JSP tag files to consolidate presentation chunks such that they can be shared between pages and Ajax responses.
- Process and generate various Ajax wire formats, including text, HTML fragments, XML, and JSON, both client-side and server-side.
- Understand the impact of Ajax processing on session management and security.
- Use RMI frameworks including DWR and jabsorb to provide direct connectivity to Java domain and/or service classes.
- Use JSF component libraries including RichFaces and Trinidad to model Ajax functionality as part of the usual JSF UI model and processing lifecycle

## Teaching Methods

- Lectures
- Hands-on workshops

---

## Course Outline

CSDTJ9

### Ajax Strategies

- The Simplest Solution
- Issues with Ajax Processing
- Model/View/Controller for Ajax
- Rendering Ajax Responses
- RMI Frameworks
- Ajax/JSF Frameworks

### Ajax Custom Tags

- JSP Custom Tags
- Tag Files
- Encapsulating UI Fragments
- Encapsulating Scripts
- Delivering Scripts Once
- Generating Unique Scripts
- Binding Scripts to Widgets
- Rendering Ajax Responses with JSP

### Handling Requests

- Ajax-Only Servlets
- Multi-Purpose Servlets
- Front Controllers
- Parsing Ajax Wire Formats
- Producing Ajax Wire Formats
- Session Management and Security
- Finding Domain Objects
- Impacts of Ajax on Page Processing

### RMI Frameworks

- Java Objects in JavaScript
- RMI for Ajax
- Direct Web Remoting
- jabsorb
- Serialization Issues

- Serialization vs. Remote Invocation
- Creating Objects on the Client Side
- Ajax RMI vs. Web Services
- Security Issues

### Introduction to JavaServer Faces

- MVC and Ajax
- JavaServer Faces
- UI Components
- The JSF Lifecycle
- What JSF Does for Ajax

### Facelets

- Issues with JSP and JSF
- JSP Custom Tags and JSF
- Facelets

### JSF Component Libraries

- Ajax Component Libraries
- RichFaces
- The Ajax4jsf Model
- Trinidad
- Partial Page Rendering
- Customizing Client-Side Logic
- Ajax-Aware Widgets
- Polling and Pushing
- JSF and User Events

### The State of the Art

- Comparison of Ajax Strategies
- Metrics: Code Efficiency
- Metrics: Runtime Efficiency
- Intangibles: Completeness and Correctness
- Intangibles: Ease of Development
- Whither Hence?