

This comprehensive five-day course shows Java programmers how to build web applications with JavaServer Faces 2.0. We develop the best-practice concepts that are formalized by the JSF architecture, from model/view/controller to the UI component framework and request-handling lifecycle. Students start to discover that there is a "JSF way" of doing things, and learn not just APIs and tag libraries but the habit of slicing application logic into its most reusable forms: managed beans, event listeners, converters, validators, and more. Students also acquire a firm command of JSF development, learning to work with JSF's list and table components, building reusable composite components, and building Ajax applications. Simple, high-level Ajax functionality is covered, and students work more directly with JSF's JavaScript API and resource-management framework.

Course Objectives:

- Describe the purpose and scope of the JSF architecture.
- Build web applications using JSF's FacesServlet, faces-config.xml, and the JSF request/response lifecycle.
- Use Facelets tag libraries to build JSF views.
- Use managed beans to encapsulate form handling and server-side presentation logic.
- Implement control logic as JSF event listeners or action methods.
- Use validators and converters to implement a validation phase for a JSF application.
- Build composite UI fragments or custom components using Facelets.
- Build Ajax applications with JSF: client-side behaviors and partial requests and responses followed by DOM updates.

Audience: Java programmers wishing to learn JavaServer Faces.

Prerequisites: Java programming experience is required. A general understanding of Servlets and JSP is recommended. Basic knowledge of XML and HTML is helpful.

Number of Days: 5 days

<p>1</p>	<p>Overview Java EE and Web Applications Perspectives: Servlets and JSP Perspectives: MVC Frameworks Perspectives: AWT and JFC JSF Value Proposition JSF Configuration Issues with JSP and JSF Facelets</p>		<p>Partial Request Cycles</p>
<p>2</p>	<p>Lifecycle The JSF Request/Response Cycle Lifecycle Phases Phase Listeners The FacesContext Class Who Does What</p>	<p>3</p>	<p>UI Components The UIComponent Class Behavioral Interfaces The Core and HTML Tag Libraries Relationship to CSS ID vs. Client ID vs. Label The UISelectItem(s) Navigating the UI Tree The binding Attribute</p>
<p>4</p>	<p>Page Navigation View Selection Navigation Rules Implicit Navigation</p>	<p>4</p>	<p></p>

	Problems with POSTback		<h:dataTable>
	Post/Redirect/Get		Defining Columns and Facets
	Support for HTTP GET		One Command per Row
	Conditional Navigation		Reading the Row Number
5	Managed Beans		Pseudo-Maps
	JavaBeans and JSF		Working with Persistent Data
	Backing Beans		Concurrency and Caching
	Configuring Managed Beans		Limiting the Scope of Queries
	@ManagedBean and Related		Paging
	Annotations	11	Converters
	The Unified Expression Language		The Converter Interface
	Value and Method Expressions		Life of a Datum
	Implicit Objects		Standard Converters
6	Scopes		Custom Converters
	Managed-Bean Scopes		The @FacesConverter Annotation
	Lifecycle Annotations		Timing of Conversion
	View Parameters		Representing Persistent Objects by ID
	Passing Request Parameters	12	Validators
7	Dependency Injection		The Validator Interface
	Managed Properties		Standard Validators
	Values, Lists, and Maps		Using Regular Expressions
	Using Dynamic Expressions		Producing Error Messages
	Dependencies and Bean Scopes		Message Keys
	The @ManagedProperty Annotation		Presenting Error Messages
8	Facelets		Posting Error Messages from Anywhere
	Migrating from JSP		Custom Validators
	View Definition Languages		The @FacesValidator Annotation
	Facelets		Validating Multiple Inputs
	Tag Libraries		JSR-303 Support: “Bean Validation”
	Writing and Using Custom Tags	13	Resources
9	Events and Listeners		Resource Libraries
	The JSF Event Model		Deploying Images, Scripts, and
	Event Types and Timing		Stylesheets
	Event Queueing		Addressing Resources
	ActionEvent and ActionListener	14	Composites
	Action Methods		Limitations of Custom Tags
	Connecting Controllers to Beans		Composite Components
	ValueChangeEvent and		Encapsulation
	ValueChangeListener		Deploying and Using Composites
	Deferring Event Processing		Interface and Implementation
	Limitations of FacesListeners		Impact on the UI Tree
10	Lists and Tables		Attributes
	Working with Collections		Retargeting
	Why We Don’t Use <c:anything>	15	Ajax
	<c:anyMore>		What is Ajax?
	<ui:repeat> vs. <c:forEach>		The XMLHttpRequest Object

Ajax and the JSF Lifecycle
Using <f:ajax>
execute and render Attributes
Ajax Listeners

- 16 The JSF JavaScript API**
The JSF JavaScript API
Triggering Ajax Requests
Refining <f:ajax> with Callbacks
onEvent and onerror Attributes
The Ajax Request/Response Process
Using Hidden Inputs
Other JavaScript Functions