Facelets

It is a light weight page declaration language which is used to build JavaServer Faces views using HTML style.

It includes the following features:

- It uses XHTML for creating web pages.

- It supports Facelets tag libraries in addition to JavaServer Faces and JSTL tag libraries.

- It supports the Expression Language (EL).

- It uses templating for components and pages.

Advatages

- It supports code reusabilty through templating and composite components.

- It provides functional extensibility of components and other server-side objects through customization.

- Faster compilation time.

- It validates expression language at compile-time.

- High-performance rendering.

JavaServer Faces technology supports various tag libraries to add components to a web page. To support the JavaServer Faces tag library mechanism, Facelets uses XML namespace declarations.

The following table shows tag libraries supported by Facelets.

Tag Library	URI	Prefix	Example	Contents
JavaServer Faces Facelets Tag Library	http://xmlns.jcp.org/jsf/facelets	ui:	ui:component ui:insert	Tags for templating

JavaServer Faces HTML Tag Library	http://xmlns.jcp.org/jsf/html	h:	h:head h:body h:outputText h:inputText	JavaServer Faces component tags for all UIComponent objects
JavaServer Faces Core Tag Library	http://xmlns.jcp.org/jsf/core	f:	f:actionListener f:attribute	Tags for JavaServer Faces custom actions that are independent of any particular render kit
Pass-through Elements Tag Library	http://xmlns.jcp.org/jsf	jsf:	jsf:id	Tags to support HTML5-friendly markup
Pass-through Attributes Tag Library	http://xmlns.jcp.org/jsf/passthrough	p:	p:type	Tags to support HTML5-friendly markup
Composite Component Tag Library	http://xmlns.jcp.org/jsf/composite	cc:	cc:interface	Tags to support composite components
JSTL Core Tag Library	http://xmlns.jcp.org/jsp/jstl/core	c:	c:forEach c:catch	JSTL 1.2 Core Tags
JSTL Functions Tag Library	http://xmlns.jcp.org/jsp/jstl/functions	fn:	fn:toUpperCase fn:toLowerCase	JSTL 1.2 Functions Tags

The Lifecycle of a Facelets Application

The JavaServer Faces specification defines the lifecycle of a JavaServer Faces application. The following steps describe that process to a Facelets-based application.

1) Lifecycle starts when a client makes a new request for a web page which is created using Facelets. JSF creates a new component tree or javax.faces.component.UIViewRoot and placed into the FacesContex.

2) View is ready to populate with components for rendering. The UIViewRoot if available is applied to the Facelets.

3) The newly built view is rendered back as a response to the client.

4) On rendering, the state of this view is stored for the next request. The state of input components and form data is stored.

5) The client may interact with the view and request another view from the JavaServer Faces application. At this time, the saved view is restored from the stored state.

6) The restored view is once again passed through the JavaServer Faces lifecycle, which eventually will either generate a new view or re-render the current view if there were no validation problems and no action was triggered.

7) If the same view is requested, the stored view is rendered once again.

8) If a new view is requested, the **Step 2** is continued.

9) New view is rendered back as a response to the client.

← prev   next →

Latest 4 Tutorials on JavaTpoint

 Cassandra   JSF

 Ruby   Java Date

Share 0