**JSF SETUP( example server JBoss)**

**Prerequisites:**

1. JSF requires JDK 1.5 or higher.
2. Java 7 or higher

**Installation:**

1. Download JBoss Wildfly 8.

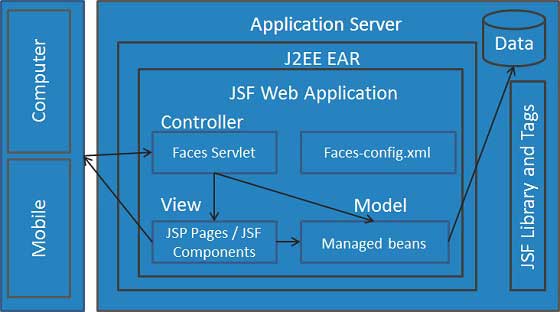
**JSF Definition**:

Java Server Face (JSF) is a MVC web framework that simplifies the construction of user interfaces (UI) for server based applications by using reusable UI components in a page. JSF provides facility to connect UI widgets with data sources and to server-side event handlers. The JSF specification defines a set of standard UI components and provides an Application Programming Interface (API) for developing components. JSF enables the reuse and extension of the existing standard UI components.

JSF provides developers capability to create Web application from collections of UI components that can render themselves in different ways for multiple client types (for example HTML browser, wireless or WAP device).

**JSF provides**

* Core library
* A set of base UI components - standard HTML input elements
* Extension of the base UI components to create additional UI component libraries or to extend existing components.
* Multiple rendering capabilities that enable JSF UI components to render themselves differently depending on the client types



Advantages of JSF

* Big vendors (Oracle, IBM, JBoss, etc) backing JSF implementation like EJB. Can expect good level of support and quality components from these vendors.
* By design and concept it allows to create reusable components. That will help to improve productivity and consistency.
* Many quality and ready to use components are available from Apache, Richfaces, Infragistics, Oracle, etc.
* The concept of action and action listener for button invocation is good.
* Has very good support for EL expression that improves the user interface code readability.
* The concept the [validator](http://www.javabeat.net/write-your-own-validator-in-jsf/) and [converter](http://www.javabeat.net/using-converters-in-jsf/) is excellent. Unlike struts JSF keeps the validation logic very close to the component declaration.
* JavaScript code are embedded as part of the component; this keep less confusion for developers and more re-usability on JavaScript code.
* With JSF 2.0 release, there is great looking third party libraries are released. The popular ones are [PrimeFaces](http://primefaces.org/" \t "_blank), [Openfaces](http://openfaces.org/" \t "_blank), etc.

Disadvantages of JSF

* Steep learning curve is one of the main dis-advantage of the JSF. However, if you are familiar with HTML and CSS concepts, it is going to be a cake walk for you
* There is no benchmarking report or promise from Sun Microsystems about the performance of JSF framework. By seeing their concept I believe it is not suitable for high performance application.
* The specification doesn’t consider bookmarking facility.
* Hardly a very few examples available for developing dynamic pages including new component and removing a component from a page based on business rule.
* Every button or link clicked results in a form post. That’s just wrong why can’t I have true links like the web is supposed to? Form submission for page navigation make complex coding for simple requirement like Cancel button. Read here to know the work around.
* Datatable component requires same data from bean on restore view phase. If the data retrieved from database, this will have impact on performance. Click here to know more about this issue.
* There is no tight coupling between managed bean and phase listener. This is a major drawback of JSF which makes JSF phase listener feature unusable.
* Default error message is not good. Need to customize the default error message.
* Not Scalable. It uses session object to store the component state across the request. In server farm environment it is too costly to replicate the session data.- See more at: http://www.javabeat.net/advantages-and-disadvantages-jsf/#sthash.fdGcpAuU.dpuf