

SHORTNOTES / JSF (JAVA SERVER FACES)

1. JSF is a user interface (UI) framework for JAVA web applications
2. It is designed significantly ease the burden of WRITING and MAINTAINING applications that run on a JAVA application server and render their UI back to the target client
3. JSF makes it easy , to construct UI from a set of REUSABLE UI COMPONENTS, simplifies MIGRATION of APPLICATION data from to UI , help manage UI STATE across server requests , allow CUSTOM BUILD COMPONENTS
4. JSF designed to be TOOLED , but also EXPOSED framework related API for “systems programmers”
5. JSF core architecture is designed to be INDEPENDENT of PROTOCOL and MARKUP
6. JSF tackles,
 - a. Maintaining UI state across requests
 - b. Supports in ENCAPSULATING difference in MARKUP across different BROWSERS
 - c. Supports Multipage Form Processing
 - d. Provides strongly types EVENT MODEL
 - e. Validate REQUEST data
 - f. Enable TYPE CONVERSIONS
 - g. Handle PAGE NAVIGATIONS
 - h. Handle ERRORS and EXCEPTIONS
7. Three different type of request ,
 - a. Non faces request generates Faces Response
 - b. Faces Request generate Faces Response
 - c. Faces Request generate Non Faces response
8. Life cycle of a faces re quest
 - a. Restore View

- b. Apply Request
 - c. Perform Validations
 - d. Update Model Values
 - e. Invoke Application
 - f. Render Response
9. JSF is fully INTERNATIONALIZED (I18N)
 10. JSF state manages happens either at CLIENT or at the SERVER
 11. JSF managed beans can have life cycle methods annotated with @PostConstruct , @PreDestroy
 12. The methods annotated with above annotations MUST have ,
 - a. No Arguments
 - b. Return type MUST be VOID
 - c. Method must not declare a CHECKED EXCEPTION to be throws
 - d. Method may be public , protected , private , default
 13. Facelet is a REPLACEMENT for JSP in JSF
 14. Facelet was designed with the JSF model in mind
 15. Facelets was the FIRST NON-JSP view declaring language designed for JSF
 16. Facelets do not need to worry about BACKWARD compatibility with JSP
 17. JSP did not support PAGE TEMPLATING for JSF , but FACELETS do
 18. Facelets are FASTER than JSP
 19. Facelets for JSF 2.0 is written in XHTML pages
 20. Facelets were used in even JSF 1.x as an alternative to JSP as a DISPLAY TECHNOLOGY and configuration a JSF 1.2

application for FACELET is bit different than configuring JSF 2.0 application for FACELETS

21. JSF 2.0 added FACELET as the default VDL support for the specification while in JSF 1.2 did not
22. In JSF all these display technologies (JSP , Facelet , XUL) are know as VDL (View Description Language)
23. FACELETS is an OPEN SOURCE WEB TEMPLATE SYSTEM under APACHE LICENSE
24. FACELETS , improving JSF by dumping JSP
25. JSF 2.0 is for JEE 6 while JSF 1.2 is for JEE5
26. It is Perfectly legal to have a hybrid solution along with JSP and JSF (1.2)
27. In JSF 1.2 , two JSP tag libraries are available for expressing UI components
28. JSF is also based on MVC architecture as well, it provides clear separation
29. typical development steps of a simple JSF 1.2 application
 - a. Configure the FacesServlet in web.xml
 - b. Develop JSF pages using VDL JSP
 - c. Use two major JSP tag libraries h , f
 - d. Write the JSF page using JSP tags
 - e. Create faces-config.xml file and configure Application specific message bundles , Navigation rules and Backing bean (Managed Beans) declarations
 - f. Wire any backing bean properties to components using EL
 - g. Deploy the application
30. JSF has two types of expressions , Value Expressions and Method Expressions
31. Value Expression = `#{UserNumberBean.maximum}` ,
Method Expression = `#{UserNumberBean.runTestMethod}`

32. JSF 1.2 does not support AJAX out of the box. Either you have to write JAVASCRIPT for that or use another third party library to add AJAX support. JSF 2.0 has a built in support using h:ajax tag