**Day 7: 6 – Aug 2024 – Web Application**

JSE (Java Standard Edition)

JEE architecture

Servlet

JSP

EJB

MVC : Model View Controller

AWT and Swing or JavaFX : these module help to create GUI or desktop base GUI application

Java Enterprise Edition : which help to create web application.

<http://www.google.com>

req(http/https)----🡪

Client Server

🡨--------Res(http/https) html/html5

Css

JS : JavaScript

JEE : Servlet/JSP/EJB

Struts framework

JSF framework

Spring framework

Asp.net

Php

Python with Django

Node Js

Server

2 types of server

1. Web server : tomcat or jee server they are type of web server
2. Application server : web logic, jboss, glashfish, WAS (WebSphere)

Servlet, JSP or EJB. No main method. in Jee we create the application and deploy the application on server

Servlet

JSP : Java Server Pages

EJB : Enterprise Java Bean

If server is a type of web server which contains web container. Container is a type of server also known as engine. Which is responsible to take the execution of servlet, jsp and ejb application

If server is type of web server which contains web container. Which is responsible to execute servlet and jsp program.

If server is type of application server which contains different type of container ie web container, ejb container, jms container etc. it provided lot of extra features as connection pooling, thread management, resource management, security etc.

Servlet : Servlet is normal java program which help to create dynamic web page on server side.

Controller --🡪 main method in core java in web servlet.

JSP : Java Server Pages : JSP is tag base server side scripting language which help to create dynamic web page on server side.

View -🡪 JSP or HTML / console using keyword.

EJB : Enterprise Java Bean (Optional) we can use normal classes.

Model layer 🡪 JavaBean class, Service class, Dao class, Resource class part of EJB layer

Framework -🡪 Spring Framework

Framework : Framework provide set API (classes and interfaces) which internally connected to each other to perform specific task. Framework internally provide standard rules. The implementation of design pattern(best practise or solution of repeating problem) is taken care by framework. If develop any application with help of framework 70 to 80% task taken care by framework. But framework is not final product. It is like template or protocol. We need to take the help of template and develop the application.

Struts framework, JSF Framework, Hibernate Framework, Spring framework -🡪 Java related

Angular framework

Django framework

.net framework

Struts is an open source web framework provided by Apache. Struts internally follow MVC architecture. They provide lot classes to improve model layer, controller layer and view layer.

Struts also known as controller centric framework.

JSF : Java Server Faces : JSF is part of oracle. Which internally follow MVC. Provide lot of api to improve model, view and controller layer. JSF is known as View centric framework.

JSF Vs Angular / React JS

Hibernate : ORM (Object Relation Mapping) which help to improve DAO or JDBC coding.

Spring Framework : Spring is an open source light weighted layer or onion based architecture framework. Which provide lot of modules which help to improve all types of application. Spring modules

1. Spring core
2. Spring context
3. Spring MVC or web

Spring MVC internally follow MVC architecture framework. Spring MVC is known as Model centric framework. (java bean, service class, dao class, resource class etc)

1. Spring DAO (using JDBC)
2. Spring ORM -🡪 help to integrate with existing orm tool like Hibernate or JPA.
3. Spring cloud
4. Spring security
5. Spring boot
6. Spring micro service
7. Spring testing

etc

Spring core and Context

IOC : Inversion of control

IOC is a programming design pattern. IOC is a concept. According to IOC in place creating any resource(object creation, database connection, file handling, server, security etc) depending on the application requirement explicitly allow to create those resource to container or engine. If container create it will create properly and maintain properly. As developer or programmer we need to pull it from a container, use it and leave it. Container responsible to maintain those resources.

DI : Dependency Injection : DI is an implementation of an IOC. Using DI we can pull the resource from container. Container inject the resource depending upon our requirements.

Types of DI

1. Constructor base DI
2. Setter base DI

To achieve DI using constructor base or setter base we can do configuration using

1. Xml base configuration
2. Annotation base configuration

XML base DI

POJO : Plain Old Java Object . The class not to extends or implements any pre defined class.

That class configuration we can do using xml or using annotation.

ORM tools : JPA or Hibernate

Web Container : it is a part of web server. Web container only create the object of those classes if class is type of servlet or jsp file.

@WebServlet()

class MyServlet extends HttpServlet (GenericServlet or Servlet interface) {

}

In web.xml file

Build tools: build tools is responsible to compile the program, run the program, creating jar or war or ear file, downloading dependencies in the form jar file, providing common project structure, which help to create documentation.

Maven project we can create using eclipse ide or using command prompt.

Mvn clean

Mvn compile

Mvn test

Mvn package

Mvn test

pom.xml : project object model : this file hold project configuration details to deploy or run maven goal.

Maven :

Gradle : xml less

mvn --version

creating maven project using command prompt

mvn archetype:generate

hit enter key twice

Define value for property 'groupId': myproject

Define value for property 'artifactId': simplemavenproject

Define value for property 'version' 1.0-SNAPSHOT: : 1.0

Define value for property 'package' myproject: : com

Yes

Move inside a project folder then run few maven goal.

mvn validate verify pom.xml file

mvn clean it remove old build files

mvn compile it compile all classes from current project and those file keep in target folder.

mvn clean remove target file

mvn test test case execute

mvn package inside target folder it will generate jar or war file

mvn install in project become part of local repository.

mvn clean package

mvn clean install

Setter base DI

If we want to achieve setter base DI we need setter methods mandatory in pojo or java bean class.

Parameterized base DI order matter while doing DI using XML. We can’t do partial DI.

In setter base DO order doesn’t matter while doing DI using XML. We can do partial DI.

Auto wiring or Auto-wire : Spring framework do the DI for primitive property or variable implicitly. But if class contains complex property or user defined class object then we need do DI explicitly using ref attribute. But with help of auto wired we can achieve DI for complex property implicitly rather than explicitly ref attribute.

byType : if we use byType autowired then spring container search that type of class in xml file. Once if found it inject that di. When we use byType we need only one bean definition of that type. If more than one present then we need to use byName.

byName : in byName we can write more than one bean definition. In this auto wired reference name part of class and id part of xml file must be match.