Spring MVC /spring boot MVC

different models of developmeing java web applications

Model1

=> Here all logics will be placed either in servlet comp or in jsp comp

=> In this model of application development

if servlet comps are used as main comps then the jsp comps will not used and vice-versa.

=> Every main web comp like servlet/jsp comp contains mix of multiple logics there is is clean seperation

.. So we can say "b/w logics.

=> This model1 architecture is suitable only for small scale web applications development

eg: nareshit.com, hyd.com

Model2 (MVC)

Spring/Spring boot MVC /web mvc module provides the abstraction on servlet/jsp technologies and web service frameworks.. So we can use this MVC/WebMVC module to develop the java web application and java Distributed Applications (especillay rest webservices based distributed apps)

=> Here the multiple logics of web application will be developed using multiple technologies by placing them in multiple layers.

=> Since multiple layers are there we can say there is clean seperationb/w logics

=>officially there is no "REST" module in spring/spring boot, but the portion of web module that is given to develop the rest web services based distributed app is called

"REST" module unofficially .The conclusion is using Spring web module we can develop

[Service class] [DAO class] =>M for Model Layer -->contains b.logic + persistence logic

(Basically represents data) =>V for View Layer --> contains presentation logic

(UI logics)

=>C for Controller layer ---> contains Monitoring Logics

(tracing the activies)

both web applciations and Distributed Apps spring/spring boot web module= spring/spring boot web mvc module

(Can be used for developing web applications and Distributed apps)

=>An Enterprise App can be a complex web application or

Distributed App or combination of both

(prefered)

eg1: flipkart with card payment

flipkart with UPI Payment

note:: Different Logics will be placed in different layers (classes/files) and we keep them participating the communication while developing web applications or distributed apps as the layered apps

note: Prefer this architecture to develop medium scale and large projects.. (web applications) beans

eg:: e-commerice sites, university apps, bank web applications and etc..

=>suitable technologies for Model layer :: java classes or spring Beans or spring with ORM code or EJB comps or spring beans with data jpa or spring rest with data jpa => suitable technologies for Controller Layer

:: Servlet or Servlet Filter

=> Suitable technologies for View layer :: jsp, html, thymeleaf, angular, react js and etc...

(best)

(java script technologies)

↑

(outdated)

=>Instead of developing controller comp as normal servlet comp or servlet filter comp ..it is recomanded to develop as FrontController Servlet (servlet comp based FrontController DP)

=> All MVC projects of current trend are MVC with FrontController.

a

DP:: Design Pattern

=>MVC is not design Pattern .. It is an architecture to devleop Layered Applications in Java.. while developing various layers and comps in that archicture we use multiple design patterns as best practies

=> Generally in the impl of MVC architecture based web applications we can see the following Desing Patterns implementation DAO, SErvice /BusinessDelegate, Factory, FrontController, View Helper, CompositeView and etc..

FrontController

(required in Model layer)

====================

(required in controller layer)

(Required in view layer)

=>The special web comp of web application that acts as entry and exit point for all or multiple http requests and responses is called FrontController web comp.

=> Generally one web application contains only one FrontController Web comp..

=> we can take either Servlet comp or servlet filter comp or jsp comp as FrontContriller comp.. but Servlet comp is recomanded.

=> A Front Controller Web comp traps either all requests or multiple requests ---> applies common system services ---> performs navigation managment, Data Management, View Management and lastly sends the response to browser (Client) (Model Management)

System services:: The Minimum services that should be applied after trapping requests are called System services.

eg: security, logging, auditing and etc..

navigation managent:: taking the requests by trapping the requests and delegating/passing appropriate uri/url requests to appropriate Handler comps/classes is called Navigation managmeent. (In simple front controller ris right request to right handler class)

the

(Model Management) (controller class) Data Management : Passing the data(inputs) that is coming along with request to appropriate handler class and passing data(results) to appropriate view comp from handler class by keeping data in required scope (generally request scope or session scope) is called Data Management/ Model Management

View Management

: sending the results/outputs generated by Handler class by selecting appropriate view comp having loose coupling is called View Management

note1:: System services, Data Management, View Management, Navigation management logics will be placed in FrontController i.e entire flow of the Applicatiofequest to response will be under control and monitoring

of FrontController Servlet comp.

note2:: Handler class is java class and also called as controller class either can process the recieved request

directly or can take the support of Service, DAO cclasses for the same (taking service,DAO classes is good pratice)

Different popular archittecture in Java a) MVC

b) Monolithic

all services will be packed

c) SOA (SOAP based webservices) d) MicroService Architecture (extension of restful web services)

note: Architeture of the App

talks about various comps and layers in App development and their interaction among them..

While develop each comp or layer of any architecture

into single unit (jar file/war file) SOA :: Service Oriented Architecture SOAP :: Simple Object Access Protocol

we follow bunch of rules as best pratices/ design patterns

to solve the reoccurring problems

note:: MVC is the Layered architecture to develope the web applications In that we use frontcontroller, application controller design patterns

in controller layer and we use view helper,composite view design patterns in view layer

What is front Controller servlet in layman terms?

Ans) It is special servlet comp of web application having special url pattern having the ability to trap and take requests, delegating the requests to other comps of the application for processing, gathering the results from the comps and passing the results to view comps, gathering the formatted results from view comps and passing them to clients as response

=>logging keeps track of the Code flow

=> auditing keeps track of the uesr activities

(browsers)

=> Handler class::: The java class where we process the

if we use FrontController DP in our project, we will be taking only one servlet comp in the web application that is FrontController and the remaining all comps java classes who gets delegated requests from the FrontController servlet

browser (Client)

MVC Arch + FrontrController Based Java Web application (Deployed in Server)

request is called handler class , some times the handler class directly process the request or some times it takes the support of service and DAO classes for the same

In Project, we can take muliple handler classes, multiple view comps but we take single frontcontroller

(C)

handler/controller

(java classes)

request url::s1.do

(1)

FrontController (3)

request

Servlet

***.do (or)/ (2)**

delegation logic

service class (java classes)

DAO class

DB s/w

(java classes)

[System services]

(4) request processing logic results in scope

(5)

SQL Query (6)

b.logic

(11)

(10)

(9)

persisstence logic

(C) ---> controller Layer (V) ---> View Layer

(7)

(M) ---> Model Layer

(8)

SQL Query

web page

[navigation mgm

View comp

(16)

(15) response

data mgmt,

view mgmt logics]

(12) (14)

SQL Query results

execution

presentation logic (13)

(M)

(V)

take

=>Since Front Controller servlet comp should trap and either all or multiple requests, so we need to configure FrontController Servlet comp having extension match url pattern (eg: *.do) or directory match url pattern(/x/y/*) or with "/" (for All requests) (best)

=> with respect

each

MVC archicture web application Frontcontroller servlet :: 1 per projectin Controller/Handler classes :: 1 per module Service classes :: 1 per module DAO classes :: 1 per db table

Q) What is the famous architeture to devleop the Java web applications? MVC + Front Controller archictture.

note: MVC+ frontController Apps can be developed in two ways

(for multiple requets)

(for multiple requests)

What is the difference b/w FrontController and Controller/Handler class? =>Handler/controller class handles the request

by taking request from FrontController.. if service,DAO classses

are not taken then the Handler classitself process the request otherwise

it will delegate request service,DAO classes for request processing. where as

=> FrontController is servlet comp

or 1 per related 3/4 table java class

b) using Java frameworks like

a) using JEE technologies like servlet.jsp spring/spring boot mvc, struts,jsf and etc..

Limitationsof MVC + Front Controller ARchicture when it is implemented with the support of serlvet jsp technologies directly with out using frameworks

UEE Technologies) =========

is

===========

a) All the logics of the all layers must be developed manually.. This extra burden to programmer

b) We need to understand and implement entire FrontController DP manually. which is quiet complex process to perform

be

c) All rules of MVC architecture should remembered and implemented manually

The rules are a) every layer is designed to develop to specific logics, So place only those logics and do not place addtional logics

b) There can be multiple view comps, multiple model comps

but there should only one front controller comp

c) All operations and flow of executions must take place through FrontController Servlet.

handler/controller class is

=> FrontController contains navigation mgmt, view mgmt, data mgmt

and system services logics where as controller/handler class contains

either b.logic or delgation logic to service,DAO classes.

=> FrontController is 1 per web application where

as the controller/handler class is 1 per module

Need of web server / Application server In web applications /web services app Development

=> The standalone app is specific to one computer and will be operated by one user at a time... So the application will be started manually and will be stopped manually (No automation is required)

=> The web application nothing but website gets 24/7 requests from the clients of different locations

So we can not think about executing the web comps of the web application manually.. For that we need to use special piece of software that automates every activity of the web application and its web comps execution That special software is nothing but web server s/w

Example web servers :: tomcat, jetty, jonus, undertow and etc..

Application server = web server++

eg:: WebLogic, webshpere, widfly, glassfish and etc..

upto tomcat 6 (web server)

from tomcat 7(application server)

web server

application server

(using servlet.jsp technologies)

While developing MVC architecture + front controller DP based web application, we need to take care of multiple logics in mutiple layers manually by following lots of MVC rules.. especially development of FrontController Servlet comp manually is very much complex process.

of

To overcome the problems servlet,isp based MVC web annlication develonoment take the support of

web

MVC frameworks to develop MVC+ FrontController based java applications..

=> Web server is a piece of software that automates web application and its web comps execution

to

It listens to client requests continuously, takes the requests, maps the requests to right web comp of

the web application, executes the web comps dynamically, gathers the results and sends the results browsers (clients) as responses

=>Every web server / App sever software gives built-in containers like servlet container,jsp container and etc...

=> In Tomcat Web server, the servlet container name is :: CATALINA and Jsp container name is JASPER

eg:: struts -----> from apache

JSF:: Java Server Faces

JSF-> from sun MS /oracle corp (2)

ADF :: Application Development Framework

Spring MVC ----> from interface21 (pivotal team)

(1)

/spring boot MVC ---> from interface21 (Extension of spring MVC) Tapstrey from Adobe

ADF ------------> from oracle corp

[All these are java based MVC + frontController Frameworks to develop java web applications)

Advantages of developing MVC web applications using JAva MVC frameworks

of

a) These MVC frameworks give ready made FrontControllerServlet taking care multiple logics

In Struts ----> ActionServlet is built-in FrontController Servlet.

In JSF ----> FacesServlet is built-in FrontController Servlet.

In SpringMVC --> DispatcherServlet is built-in FrontController Servlet.

In SpringBootMVC --> DispatcherServlet is built-in FrontController Servlet.

spring Boot MVC = spring MVC++

=> if any module is implemented in spring style that

is called spring-<module>. eg: spring mvc, spring data jpa and etc..

=> if any module is implemented in spring boot style that

is called spring boot-<module>. eg: spring boot mvc, spring boot data jpa and etc..

b) Provides the abstraction on servlet,jsp technologies and simplfies MVC Architecture based web application development

c) Flow of execution (Navigation mgmt) is fixed according to logics placed in FrontControllerServlet

So Developers need not bother about flow of executions. (becoz the ready made FrontController servlet takes care of navigation management)

d) The pre-defined FrontController contains logics already satisfying MVC rules and guidelines

not

So Programmer need to remember and implement those rules manually

e) Gives good productivtiy in application devleopment (More work in less time, having good accuracy)

and etc..

DispatcherServlet of spring MVC/spring Boot MVC web application

==========================

================================

=> it is spring MVC supplied readymade servlet comp

org.springframework.web.servlet.DispatcherServlet

(spring-webmvc-<ver>.jar)

=> Once we add spring-boot-starter-web we get lots of jar files related to MVC module like spring-web-<ver>.jar, spring-web-mvc-<ver>.jar and etc..

In that spring-weer.jar file contains the readymade "DispatcherServlet" comp

as front controller servlet comp with the url pattern "/"

=> It is given as FrontController Servlet having ready made navigation logics(control the flow),

view management logics (controls the view comps) and Data mangement logics (decides the data flow)

=> In spring MVC application we need to cfg explicitly with SErvletcontainer having

either directory match url pattern (like /x/y/*) or extenstion match url pattern (like *.do) or global url ("/") (manuall work)

(To take multiple requests)

(DispatcherServlet)

=> Spring Boot MVC application it is automatically registered with servlet Container

having url pattern "/" and having load on startup enabling

(To take all requests)

of

takes takes all the requests (recomanded)

=> On the deployment spring MVC/spring Boot MVC App the following operations takes place

All these are

server startup

or deployment

activities in spring MVC

or spring Boot MVC

applications.

on DispatcherServlet

a) Becoz of "load-on-startup" that is enabled the servlet container creates

DispatcherServlet class obj either on the deployment of the web application (for hot deployment) or during the server startup (for cold deployment)

hot deployment :: deploying the web application when the server is in running mode cold deployment :: deploying the web application when the server is in stopped mode

of

internally b) The init() method DispatcherServlet(DS) executes which creates IOC container of type WebApplicationContext (ApplicationContext type) container

c) This Web applicationContext performs pre-instantiation of singleton scope spring beans

(loc

like controller/handler classes, service classes, DAO clases, DataSources, Handler Mappings, ViewResolver and etc.. perfoms the necessary Dependency Injections and keeps them in the

Internal cache of IOC container. (This process keeps all the objs ready in the deployment itself)

note:: only standalone App contains main(-) method... So in standlone Apps we create IOC container in the main(-) method.. web applicaitons (spring/boot MVC Apps) does not contain main(-) method The DispatcherServlet takes are creating IOC container in those web applications.

note:: In spring MVC applications, DispatcherServlet, view comps(jsps) are taken care by Servlet,jsp containers.. where as controller/handler classes, service classes, DAO classes, Handler mappings, view resolvers and etc,, are java classes acting as spring beans who will be taken care by the DispatcherServlet created IOC container.

note:: all web applications must be deployed in web server /application server like tomcat, weblogic, wildfly

and etc.. to make them accessible for 24/7.

Server to

WebServer or Application listens client requests continuosly, takes the requets, maps the requests to the web comps of web application, executes the web comps dynamically, gathers the results and sends them browser responses.

WebServer /application server

Spring MVC web application (Deployed app)

if we enable <load-on-startup> on servlet comp, the servlet container creates the object for servlet comp class either during server startup or (for hot deployment) during the deployment of the web application.. this called pre-instantiation or eager instantiation of the servlet comp

hot deployment:: deploying the web application when the server is in running mode

cold deployment :: deploying the web application when the server is in stopped mode

note:: In spring boot MVC app, the pre-defined DispatcherServlet will be registered with Servlet container using Programatic approach using sc.addServlet(-,-). In thaprocess the DispatcherServlet gets "/" url pattrn and enabled <load-on-startup>

note: DispatchcerServlet uses HandlerMapping comp as the helper comp to pickup right handler method of hander/controller class for request processing

note:: DispatcherServlet uses ViewResolver comp as the helper comp to pick the right view comp (jsp file) for formatting the results

WebServer /Application server

automates the web application and

its web comps execution to provide 24/7 access to web application.

=>SErvletContainer is managing DispatcherServlet =>DispatcherServlet creates the IOC container i.e ServletContainer is managing the IOC container indirectly

(Dispatcher

jsp1 jsp2

DS

=>The DispatcherServlet created IOC container manages the spring beans like handler classes, service classes, DAO classes and etc..

=>Jsp Container which internally uses Servlet container

will manage the execution of the jsp comps(jsp files)

webcontainer = servlet container + jsp container

Handler Mapping

Servlet) IOC containe view Resolver

DAO class

Service class

controller class

DS=DispatcherServlet

Jsp container

Sevletcontainer

JRE/JVM

Application Server = web server ++

How many ways are there to confiure /map servlet comp with servlet Cotnainer?

ans) Three approaches are there

approach1) Declarative approach (using web.xml cfgs)

approach2) annotation approach (using@WebServlet).

approach3) Programatic approach/Dynamic Servlet Registration (using sc.addServlet(-,-))

(Using addSEervlet(-) on ServletContext object)

note:: Spring Boot MVC uses approach3 to register DispatcherServlet with SErvletContainer having the url pattern "/" and by enabling load-on-startup