**INTERVIEW QUESTIONS ON SPRING MVC:**

1. **What is spring mvc framework?**

Spring mvc framework provides mvc architecture, which is used for developing loosely coupled web application by using this lossely coupled we have many advantages. Like

Clear separation of roleS: controller, view handler, handler mapping, validator and etc**.**

<welcome-file>index.jsp</welcome-file>

</welcome-file-list>

<servlet> <servlet-name>Helloworld</servlet-name>

<servlet-class>org.springframework.web.servlet.dispatcherservlet</servlet-class></servlet>

<servlet-mapping>

<servlet-name>Helloworld</servlet-name>

<url-pattern>\*.ds</url-pattern>

</servlet-mapping>

</web-app>

1. **What is dispatcher servlet and contextloadlistner?**

Dispatcherservlet: like other framework where servlet will handle http req and response, spring mvc have dispatcher servlet which will do more then that like it will take the req and consults the handlr mapping to call appropriate controller now controller will take the req and select appropriate service method and set the method data and returns the view name to dispatcher servlet. Dispatcher servlet will take help of view resolver to pickup the defined viw for the req. once the view is finalized dispatcher servlet will passes the model data to view which is finally rendered on the browser.

Contectload listener will read the configuration file parse it loads the beans defined in that config.

1. **How to use java based configuration?**

**First write the dependencies and write code in web.xml**

<welcome-file>index.jsp</welcome-file>

</welcome-file-list>

<servlet> <servlet-name>Helloworld</servlet-name>

<servlet-class>org.springframework.web.servlet.dispatcherservlet</servlet-class></servlet>

<servlet-mapping>

<servlet-name>Helloworld</servlet-name>

<url-pattern>\*.ds</url-pattern>

</servlet-mapping>

</web-app>

**and write entries in xml file**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

//when you add jars it provides some content.

//default handler mapping

<bean name=*"/hello.ds"* class = *"Controller.HelloController"*/>

<! view resolver...>

<bean class=*"org.springframework.spring.view.i"*

<property name=*"prefix"* value=*" "*/>

<property name=*"prefix"* value=*".jsp"*/>"

a**nd create controller class.**

**public** **class** HelloController {

//make it a class which implements from controllerorg.springframework......you will get this by adding jars.

//add parameters req and res.

String name = req.getparameter("name");//it will reads the name file

//now create a object.

Map m = **new** Hashmap();

//put the output in this object

m.put("msg", "Helloworld" +name);

//now you need to send this object by modelandview.

ModelAndView mv = **new** ModelAnd View("success", +name);

**return** mv;

}

1. **How can we use spring to create restful web service returning JSON response?**

First add the json dependency now you can create json response from your MVC conreoller by adding **@responsebody** in conreoller class.

Or add @restcontroller in the place of @controller

@restcontroller = @controllr + @responsebody.

Example:

@**controller**

Public class classname{

@**requestmapping**(value=”/extension)

Public @**responsebody** employee methodname(){

employee emp = new employee();

return emp;

**instead of adding @responsebody add @restcontroller.**

@**Restcontroller**

Public class classname{

@**requestmapping**(value=”/extension)

Public employee methodname(){

employee emp = new employee();

return emp;

1. **Can we have multiple spring configuration files?**

Yes u can add all files in web.xml file using contextconfiglocation init parameter.

Or u can import them into exisisting configuration file you have already configured.

<servlet> <servlet-name>Helloworld</servlet-name>

<servlet-class>org.springframework.web.servlet.dispatcherservlet</servlet-class>

**<init param>**

**<param-name> contxtconfiglocation</paaram-name>**

**<param-value>**

**WEB-INF/spring-dao-hibernate.xml,**

**WEB-INF/spring-services.xml,**

**WEB-INF/spring-security.xml**

**</param-value>**

**<load-on-startup> / </load-on-startup>//we use theload-on-startup when the multiple servlet has same value it will make order and because when the servlet load the first req it consumes more time for 1st req when u use this loadonstartup the servlet will load at the time of project deployment or servlet start so it takes less time**

</servlet>

<servlet-mapping>

<servlet-name>Helloworld</servlet-name>

<url-pattern>\*.ds</url-pattern>

1. </servlet-mapping>
2. **DIFFERENCE BTWN <context:annotation-config> and <context-component-scan>?**

1)The main diff is that both tags is that <context:annotation-config> will activate applies annotations in already registered beans in application context it doesnot matter whether beans was registered by which mechanism eg: <context-component-scan>? Or it was defined in application-context.xml file itself.

<context-component-scan> does what <context:annotation-config> does, the main but addidtionlaly it will scan packages and registere the beans in application context.

1. What is the diff bten the

@**COMPONENT:**

It markes the java class as beans in which companent scanning will pick it up and pull it into application context. You need to add it on the top of class.

**@repository:**

It is the specialization of @component, it addidtionally imports the DAO’S into DI container, it also makes the unchecked exception and eligible to transit into spring datascessexception.

**@service:**

It is the specialization of @component, it doesnot have any additional behior over @component but if u us it ovr @component in service layer class it makes better intent.

**@controller:**

It is the specialization of @component by adding this to class maks spring mvc controller that beans will automatically marked into di container by adding this u can add another annotations like @requestmapping etc.

1. **What is view resolver?**

It’s a interface implemented by objects that can resolve view by name, many ways one of it by implementing the internalresourceviewrsolver which defines the prefix and suffix to resolve view name.

Ex: when controller return sthe login string prefix=”djhbjs”, suffix=”ndhdjs” file is searched and rendered.

1. **Multipartresolver?**

Spring provides multipart rsolvre to upload a file in web application there are two types of ways:

Commonmutipartresolver

Standardmultipart resolver.

1. **Spring validation support?**

By using sjr-303 annotation and any reference implementation eg: hibernate validator.

1. **Spring mvc interceptor?**

As servlet filter pre and post web req, before handle by servlet u ca use handlerintrceptor which handles pre and post req that can handle by spring mvc controller.

1. **Handle exceptions in spring mvc?**

By using handler exception resolver in dispatcher servlet.

1. **Tomcat jndi datasource?**

By configuring in bean configuration file and then inject in spring as bean dependency,

1. **Spring mvc and 3-tier architecture?**

Spring mvc is the presentatipion layer oly for of 3-tier archi and the modelview and controller concerned with the presentation tier and make use of model tier to populate the model with data from data tier.