**Interview Questions – Spring , Spring Boot & Microservices**

**1.What is spring boot**

* **Spring Boot is an open source Java-based framework used to create a micro Service.**
* **It is used to build stand-alone and production ready spring applications.**

**2. Features of spring boot**

1. **Auto Configuration - spring boot scans the classpath and identify the jars available in the class path, based on that it will do the configuration.**

**For eg : in our project we are using JPA, so since h2 is available in class path it will automatically prepared the data source connection**

1. **Starters in the build tools - if we want to create spring boot application from the scratch we need to take care lot of dependencies, instead of that spring boot comes up with this starters where we need to add what are the starters we need**

**Starters for our project, then this dependencies automatically taken care by spring boot.**

* **'org.springframework.boot:spring-boot-starter-actuator'**
* **'org.springframework.boot:spring-boot-starter-data-jpa'**
* **Includes Restful, Apache Tomcat 'org.springframework.boot:spring-boot-starter-web'**
* **.springframework.cloud:spring-cloud-starter-config:2.2.3.RELEASE'**
* **Cache fn -'org.springframework.boot:spring-boot-starter-cache'**
* **test- t'org.springframework.boot:spring-boot-starter-test'**

1. **Spring boot CLI - Command Line Interfaces allow you to quickly start the spring boot application. Also used to work with scripting languages**
2. **Spring Initializer - it is available online. We don’t need to create app from scratch we can download the skeleton for our project by using spring initializer**
3. **Actuator - is the mechanism where we can monitor the health of our applications.like many sessions,connections,data resources and if we want to analyses the log**

**3. Difference between Spring and Spring boot**

**4.what are the frequent annotations used in springboot**

1. **@SpringBootApplication**
2. **@EnableAutoConfiguration- enables spring boot auto configuration mechanism. Automatically load configurations based on class path jars**
3. **@ComponentScan -the @ComponentScan annotation is used to find beans and the corresponding injected with @Autowired annotation.**
4. **@Configuration - allow to register extra beans in the context or import additional configuration classes**
5. **@RequestController - Rest API Implementation- @RestController is a special controller used in RESTFul web services and the equivalent of @Controller + @ResponseBody.**
6. **@RequestMapping - To map the path to the REST API.**
7. **@EnableCaching- enable caching mechanism**
8. **@Test - enable Junit testing**
9. **@Controller @RestController @EnableWebMvc- enable spring mvc + embedded tomcat - @Controller is a common annotation that is used to mark a class as Spring MVC Controller**
10. **@EnableWebSecurity-enable**
11. **@EnableTransactionManagement- enable transaction management**

**5.how do change the tomcat port number**

**In the application.properies file:**

**server.port= <port number>**

**6. Where do you configure application related properties?**

**We will define all the configuration related parameters for the spring boot application in application.properties file.**

**Also spring boot supports .properties, .yaml files**

**If we are have the profile we can have different property file for different environment like dev,QA,Production**

**7. How to provide Context path for your application**

**You can define the context path in application.properties using Server.servlet.context-path in application.properties file**

**Server.servlet.context-path=/my-custom-path**

**8. How to define profiles in spring boot**

**Spring profiles provide the way to segregate parts of your application configuration and make it available only in certain environments**

**Any @Component or @Configuration can be marked with @profile to limit when it is loaded**

**spring .profile.active=prod**

**9.Spring Initializer**

* **Spring initializer is used to create a skeleton of a project by selecting the required dependencies and projection info like maven/gradle,version etc.**

**10.what are the spring boot starters why we need it**

**Spring Interview Questions**

**1.What is dispatcher servlet**

* **DispatcherServlet is a default front controller available in spring.**
* **Front controller will accept all the request from client (through common url: /drugdisease.com/\*), and make decisions that which controller class can handle each incoming request and redirect the request to respective controller. Get the response and return the same to client.**
* **Add dispatch servlet name and mapping web.xml file, (right click project, and create servlet)**
* **Whenever the dispatcher servlet initialize it is looking for dispatcherServletname-servlet.xml under web-inf folder.**
* **Dispatcherservlet initialized only at the time of first request coming.**
* **Also, dispatcherservlet create webApplication context container which will be created on server start up and closed at the time of server shut down.**

**In spring boot**

* **The Spring Boot autoconfiguration registers and configures the *DispatcherServlet* automatically**. Therefore, we don’t need to register the *DispatcherServlet* manually.
* By default, the *spring-boot-starter-web* starter configures *DispatcherServlet* to the URL pattern “/”. So, we don't need to complete any additional configuration for the above *DispatcherServlet* example in the *web.xml* file.
* However, we can customize the URL pattern using *server.servlet.*\* in the *application.properties* file:
* server.servlet.context-path=/demospring.mvc.servlet.path=/baeldung

**2. what is an IOC container**

* IOC- Inversion of control is core of the spring framework.
* It is responsible for creating ,managing and injec the objects called beans.
* And manages their entire bean life cycle.
* Two types of IOC container there,

a. BeanFactory - it has basic functionalities. It is Interface, it is Legacy

b. ApplicationContext - it provides all the features that provides BeanFactory and additional features also. It is Interface, it is advanced.

c. web application context- it is extended to application context. It is interface. Designed to handle the java web-based application.

**3. what is an application context**

**4. what is a web application context**

1. **what is view resolver**

**Microservices:**

[**https://www.guru99.com/microservices-interview-questions.html**](https://www.guru99.com/microservices-interview-questions.html)

**Spring Cloud?**