**Spring Boot Interview Questions**

**What is springboot and benefits ?**

RAD :- Rapid application development

**What**?

* Spring boot spring framework for RAD
* Auto configuration
* embedded application server.(tomcat,Jetty)

**Why**?

* it provides RAD

**HOW**

* Its provide efficient , fast and standalone you can just run it without boiler plate code extra configuration and dependencies

**What is RAD?**

* Modified waterfall model
* Bussiness modeling
* Data modeling
* Process modeling
* Application generation
* Testing and turnover

**Change port**

* Server.port properties

**Change embedded server**

* Change the dependency from the tomcat to jetty

**@value**

* Get property file value

**@RestController**

* Create web service
* Restful controller
* Combination of controller and response body

**@GetMapping**

* Get web service method

**@RestController vs @Controller**

* human readable html css js
* Jason XML base client readable rest api client
* web service base app
* web application
* model object to view template
* return response to client

**@GetMapping Vs @RequestMapping**

* Sure when handle only get request.
* get post put all the type

**What is use of profile?**

* multiple env dev pp prod
* Spring.profile.active=deb
* app properties create multiple property

**@Springbootannotation**

**Disable default web behaviour**

* Spring.main.web.application.type=none

**EnableAutoConfiguration(excludeconfiguration=true)**

**Actuator**

Additional feature

* Monitor
* Manage
* auditing health
* metric gathering
* Adding dependency spring-boot-starter-actuator in pom.xml what bean created,
* CPU usage and http hits

**/actuator/health**

Status :- health

**InMemoryHttpTraceRepository** require to create bean

**Steps to deploy**

* War/Jar
* Springboot maven plugin

**Advantages yaml over properties**

* Clarity , readability ,hierarchical configuration supports maps lists scaler
* YamlMapFactorybean

**Transaction Management**

* **Proxy: -** Spring Boot implicitly creates a proxy for the transaction annotated methods.
* **Wrapper :-**  for such methods, the proxy acts like a wrapper,
* **Create Transaction At Beginning: -** which takes care of creating a transaction at the beginning of the method call and committing the transaction after the method is executed.

**@Qualifier**

By including the @Qualifier annotation, together with the name of the specific implementation we want to use, in this example Foo, we can avoid ambiguity when Spring finds multiple beans of the same type.