# **SELECTIVE INTERVIEW QUESTIONS**

**1. What is difference between maven vs Gradle:**

Both having main purpose to support dependency management and automated build.

Maven:

1. Maven contains all dependency details in well manner structure inside pom.xml and it is purely xml based

2.To achieve inbuilt functionality we need to add plugin separately in maven and we need to specify specific goal for it which supported by maven build tool

Gradle:

1. One of the first things we can note about Gradle is that it’s not using XML files, unlike Ant or Maven. Cause internally using Groovy script which more technically handy any technology can integrate it easily

2. in gradle every build and plugin we need to specify as a task and in one single shot you can run all the task without specify separate goal for each

As we compare 2 main points except these 2 all difference just syntatically changed but having the same features , even if you will mark build.gradle we are specifying there apply plugin: 'maven' so we can say gradle is just a layer above ant and maven with much more manageable way if you observe we are following same structure and command just syntax changed for both.

**2. How to generate WADL for Rest application**

To generate WADL you need to write separate controller class with All your API details for better refer below link.

https://javattitude.com/2014/05/26/wadl-generator-for-spring-rest/

As part of our project we are not using WADL cause we are consuming our service only not any third party so from UI or from Ansible script we are getting the endpoint URL .which is more easy suppose i developed bunch of end point which i want to provide other user then only i need to bind up all endpoint and then i need to provide WADL . as we are only using it for our internal call so WADL structure not required

**3. Why we are using Rest**

First try to understand what is Rest ,Rest is an architectural way of desine web services , through which you can consume and expose any of your service to other user with multiple mediaType it can be JSON ,XML ,HTML or PLAIN\_TEXT

Assume i have multiple modules as part of my project for example

1.EMS-API

2.HRMS-API

so think from HRMS-API i want to access any of method from EMS-API how can i do that

Except Rest we have 2 way 1)SOAP service 2)JMS right ?

And you know we need to follow lot of steps to develop any of above 2 , so as we know to access any other service simple i should aware on URL,REQUEST and RESPONSE that's enough to access any other API so this easy and handy features we are getting using Rest API

i can use normal spring mvc as well but main problem i can't expose my service and one more thing we are not developing fully MVC based application which provides by MVC (MODEL,VIEW AND CONTROLER) , cause MODEL and CONTROLER is enough for us to expose any service through rest and that view things mapped by separate UI team through any of third party api like angular or ajax or jquery .

**4. Spring vs spring boot**

spring Vs Spring Boot , let's compare basic In short, Spring framework helps you build web applications. It takes care of dependency injection, handles transactions, implements an MVC framework and provides foundation for the other Spring frameworks (including Spring Boot)

While you can do everything in Spring without Spring Boot, Spring Boot helps you get things done faster:

simplifies your Spring dependencies, no more version collisions

can be run straight from a command line without an application container

build more with less code - no need for XML, not even web.xml, auto-configuration

useful tools for running in production, database initialization, environment specific config files, collecting metrics

if you will observe multiple repository it supports with 0 config and lot of inbuilt features it supports like data caching , it will give us monitor our application in multiple environment etc..

**5. Why we should go for lambda expressions?**

Lambda expression just simply help us to ignore instantiate anonymous inner class and it helps us to write code in easy manner with less boiler plate code like below and it specific for functional interface

if you observe stream api of java 8 maximum utility method using the lambda expression that's why our code is smooth and readable and faster

package com.acc.dev.Core;

public class LambdaDem {

public static void main(String[] args) {

// without lambda

MyInterface1 interface1 = new MyInterface1() {

@Override

public void test() {

System.out.println("calling 1 without lambda");

}

};

interface1.test();

// with lambda

MyInterface1 interface2 = () -> {

System.out.println("Calling with lambda");

};

interface2.test();

}

}

**7. How to create enterprise application project using maven.**

To develop any enterprise application using maven means we must have instaled maven and set class path else maven plugin should avilable in our developer IDE .

1.Go to file section

2.Create Maven project

3.choose maven archtype as WebApp

4.specify your group id , artifact id and version

5.finish project

6.add required dependency

7.Start deveopmenet ur application

**8. What is maven goals?**

Maven goals is nothing it just inform to your maven build tool what exact task will done by your plugin

for example assume i added plugin wsdl2java how maven know what is the role of this plugin by seeing the goal section and phase section he will identify like below

<plugin>

    <groupId>org.apache.cxf</groupId>

    <artifactId>cxf-codegen-plugin</artifactId>

    <version>${cxf.version}</version>

    <executions>

        <execution>

            <id>generate-sources</id>

            <phase>generate-sources</phase>

            <configuration>

                <sourceRoot>${project.build.directory}/generated/cxf</sourceRoot>

                <wsdlOptions>

                    <wsdlOption>

                        <wsdl>${basedir}/src/main/resources/myService.wsdl</wsdl>

                    </wsdlOption>

                </wsdlOptions>

            </configuration>

            <goals>

                <goal>wsdl2java</goal>

            </goals>

        </execution>

    </executions>

</plugin>

**9. Maven life cycle?**

maven life cycle catagorized in multiple section and each life cycle having specific role. like

mvn:validate -> helps us to validate pom.xml

mvn:compile - compile the source code of the project

mvn: clean ->will help us to delete all generated file from target folder

mvn:package - take the compiled code and package it in its distributable format, such as a JAR.

mvn:verify - run any checks on results of integration tests to ensure quality criteria are met

mvn: install -> will help us to download all pom.xml dependency and store in m2 repositiry

mvn:generate Source-> helps us to generate source code from plugins like wsdl2java etc.

mvn:test -> will help us to run our junit test case

mvn:build-> like here u can run any other specific life cycle commands

like this some more phase is there suppose assume am running mvn:install means all the initial phase will execute first like validate,clean and package and all

**10. How to write test case for your rest api.**

To write junit test case for rest api multiple third party is there like mockMvc , power mock and mockito

As am using mockMvc and it's quite easy to work

assume i have a end point http://localhost:8080/saveEmployee who will take input as Employee object and will give me some response as it is saving employee object means it is one POST HTTP method call

steps

1.inject mockMvc in ur test class

2.inject webApplicationContext in your test class

3.Before run your test method just initialize ur mockMvc using @Before annotation

4.write a test method with return type void

5.gather request and convert it to jsonString use either Gson or use ObjectMapper

5.then mention your method type which u r going to perform by passing the appropiate end point url and pass the request string

Like mockMvc.perform(POST("http://localhost:8080/saveEmployee"))

6. Mention the mediaType whether it is json or xml or plain text

7. It will return us MvcResult as a raw data then map it your appropriate Response type

8. then pick any Boolean value from response and compare with assert

Example:

@RunWith(SpringRunner.class)

public class SpringDataJpaApplicationTests {

private MockMvc mockMvc;

private ObjectMapper om = new ObjectMapper();

@Autowired(required = true)

private WebApplicationContext webApplicationContext;

@Before

public void setUp() {

mockMvc = MockMvcBuilders.webAppContextSetup(webApplicationContext).build();

}

@Test

public void addUser() throws Exception {

UserRequest request = new UserRequest();

List<User> users = new ArrayList<>();

User user1 = new User();

user1.setName("Basanta");

user1.setAddress("Hyderabad");

User user2 = new User();

user2.setName("Santosh");

user2.setAddress("Bangalore");

users.add(user1);

users.add(user2);

request.setUsers(users);

String requestJson = om.writeValueAsString(request);

MvcResult result = mockMvc

.perform(post("/Services/saveUser").content(requestJson).contentType(MediaType.APPLICATION\_JSON\_VALUE))

.andExpect(status().isOk()).andReturn();

String resultContent = result.getResponse().getContentAsString();

UserResponse response = om.readValue(resultContent, UserResponse.class);

Assert.assertTrue(response.isStatus() == true);

}

**11. What is default mode of autowiring in spring?**

As we are using annotation approach so by default it will consider byType

**12. Why JMS if we have Already web Services**

You will find lot of difference in google, as far as i know there is main 2 reason

1. We should go for JMS if our request payload contains huge data

2. Using JMS u can't achieve interoperable nature as its name itself self-explanatory i.e Java messaging service

And in both the cases u can achieve asynchronous and synchronous mechanism