Interview Preparation – Full Stack Developer

**Topics**

|  |  |
| --- | --- |
| Self-Introduction | G |
| Roles and Responsibility |  |
| Day to Day Activities |  |
| Project – Over View & Flow |  |
| Designing Pattern |  |
| Annotation – Spring Boot |  |
| Java 8 |  |
| JSP |  |
| Spring Boot |  |
| Java |  |
| My SQL |  |
| Project Questions |  |
| SDLC – Agile, JIIRA |  |
| Maven, GIT |  |
| HTML, CSS |  |
| Question Bank To Practice |  |
| Programs to Practice – Java , Java 8 , Spring Boot |  |

Self-Introduction

As you know my name is Bharath. I currently working as a Software Engineer in Clap Hand Tech. I have 2.7 years of experience in java and J2EE Technologies.

I have good work experience in both front and back end technologies.

I have hands on experience with MVC Framework and Spring Boot

I have good work experience with My SQL Quarries

I have hands on experience with handling exception in project

I have good work experience in using and implementing designing pattern and providing solution for its concept

Moreover as a software engineer I have experience my role and responsibility in my job circle

Roles and Responsibility

I gather requirement from the functional team in terms of BRD (Business Requirement Document) and FDD (Functional Design Document)

Understanding the FDD

Updating Quarries in the Quarry log (Excel file)

Every day we have meeting with the functional team to get clarification on Quarries which are logged in the Quarry log

Sending MOM (Moment of Meeting)

Development of Code

I have best practice of Exception handling, performance and Transaction

Reviewing of Code using Sonar Cube

Pushing the code in the Repository (GIT Hub)

Support for UAT phase (If any issues occur while testing we have to resolve it)

Work closely with the Analyst and functional team to enhance the existing application

I also responsible for updating the functional and technical document for our project.

Day To Day Activates

My Daily Activities is Repetitive. Once I stepped in to the office. I check my JIRA Tool. If any task is received related to my project, I will work on it. As we following Agile Methodology which is based on Sprit, Every Sprint takes 15 days. They will create user story and assign task based on the user story through JIIRA and we will work on it. Every Days we have Scrum Meeting where we will collect the client requirement and reporting my task and clarify the doubt related to the project. Then I write business logic based on data received. Once I completed the pull the code on to the Repository i.e.., Git Hub then I update my daily work in the EOD comments on JIIRA.

Project – Overview

**E-Mart:**

I am working in Business Management Project for a client called Loyalty One, a Commercial Company in Canada.

Application- I am working in Business Management Application where Agents are End Users. It helps to analyse the sales and purchase report. It also allow the agents to generate the report as per their requirement. This application help the agent to work easily and quickly.

My application consist of 3 phases.

Report Hub: it has a default template with mandatory field. We can generate report with the help of that Mandatory fields.

Report Generator: it is based on User’s Convenience. If the Organisation receives a new requirement they will customize the reports as per their convenience.

Report Indicator: it is used to get the report in graphical data where the agent can view the data in Pie chart or Bar chart.

Here we are using data via 3rd Party Tool can TIBCO web focus report caster API it is used to schedule data as per agents convenience i.e., Daily, Weekly, Monthly, yearly.

**Flow:**

Report & Analytics - >

1. Report Hub – >

a. All Report -> it have All Report ID & Search Bar -> select an ID-> fill the Mandatory field -> Submit

(Once submitted it will be received in Agent Mail)

b. Report Organizer - it generate report based on Agent Requirement so, need to provide the Mandatory information then -> Submit

(Once submitted it will be received in Agent Mail)

c. Schedule Report - (need to provide) Report Id, from and till Date, file format (pdf/doc) and Schedule timer i.e.., daily, weekly, monthly, yearly -> Schedule.

(Once submitted it will be received to the Agent Mail on the scheduled time)

2. Report Generator –> (need to provide) Report Id, from and till Date and file format (pdf/doc) - >generate

3. Report Indicator –> (need to provide) Report Id, from & till Date along with indicator format (pie /bar) -> Generate We performed by using J-Query which is fast and high-feature java script library.

Designing Pattern

Well Proven solution for the specific problem in the code.

**Singleton Pattern:**

We create only one object for a class and the same object will be used by all the other classes.

Two Form:

1. Lazy Initialization – Creating an Object whenever required.
2. Early Initialization – It Create an Object automatically during class loader.

How to Create Singleton:

* Data Member must be private and Static.
* Constructor must be Private
* Method should be Public and Static

Used in – Multithreading, cache, logging.

Way to broke singleton:

Serialization – overcome using resolve () method

Cloning - overcome using Clone () method

Reflection – overcome using Enumerator

Advantage: Less Memory space because it use only one object for the entire class.

Annotation

Meta Data or Replacement of XML file

Spring Boot Application – Used for Auto Configuration. It is a Meta annotation that mark the main class of spring Boot Application.

Enable Auto-Configuration – it simplifies the setup process by loading the dependencies and properties

Component Scan – Scan the packages where the application is located

Configuration – It indicate a class is a configuration class. It contains n number of Bean methods. It used to register bean on to the container. Environment specified in this configuration class will be configured to the application

Auto-Wired – Automatic Dependency injection. It works only with the reference and cannot inject primitive and string values. Less Code - advantage

Component – Used to detect the spring Bean class automatically. It create object automatically and save it in the container and

Primary – when there is multiple bean of same type it check for priority.

Qualifier – It has high priority then Primary but it must have primary. It eliminate the issue of which bean have to be injected.

|  |  |
| --- | --- |
| Controller | Rest Controller |
| Used for define a spring MVC Class | Specialized version of Controller |
| Need to used @ responseBoby on Every handler method | Combination of @responseBoby and Controller So, no need to mention response boby |
| Responsible for processing the incoming request and returning the view name or ModelAndView object | Responsible for processing the incoming http request and directly returning data to be serialized into the HTTP response body |
| Cannot convert object to JSON | Convert object to Jason |

Request Mapping– It used to Map HTTP request to the specific Method in a controller class. It applied for both class level and method level

Response Entity -It represent the whole HTTP response. i.e.., Header, Body and Status Code. We have to return it from End Point.

Response Boby – It convert java object to HTTP response.

Request Boby – It convert HTTP request to java Object.

Service – It is used to organize the Business logic and Data Manipulation in a Spring Boot Application

Repository – It indicate a repository layer which helps to communicate with the DB

|  |  |
| --- | --- |
| Path Variable | Request Param |
| It Capture Values from the URL Path | Capture query parameter for the URL |
| Denoted by Curly Braces | Appears after ‘?’ (Question mark) in URL and Separated by & |
| Used to pass Dynamic values as part of URL path such as id, name | Provide additional information to the server that is not a part of the path |

Profile – It Present inside Configuration class. If more than one environment is available. The profile make only one environment to be configured with the application

Bean – It is a Method level annotation. It tells the container that the object of the class is created by the user manually.

Value – It extract value from the application.properties to the parameter, data member and constructor

Valid – It ensure the validation of the whole object i.e.., Server Side Validation

Size – It is used to define the minimum and maximum size or length of field

Past – It is used as part of form validation to ensure that a date or time in the field is relatively past to the current time or date

Not Null – It is used as part of form validation to ensure that a field is not null

Entity – It Refer to a class that represent database table. It mostly used in ORM framework such as JPA, Hibernate which used to map database tables to java object

Id – It represent a Primary Key in the Database table

Column or Join Column – used to map the field of the entity class to the corresponding column in the database table.

Table– refers to database table to store the data. Table is a structured way to store the records in rows and column

Generated Value– Auto Increment and it is used to specify the strategy for generating values of Primary key field in database entities. Types: identity , sequence, Table, Auto

Query Param- Query param is used to pass data to the server in an HTTP request (same a request Param)

Properties Source– refers to location where the spring boot reads the configuration properties for your application.

Common Property sources in spring boot:

1. Command Line Argument
2. Java System Properties
3. Properties Files
4. Configuration in Embedded servlet - Tomcat
5. Property source in the Spring Environment

Model Attributes– these are used to pass data from the controller to the view. Model Attributes are commonly used in web application to send data from the server side to client side

Java 8

**Lambda Expression:**

It Provide Function Programming Features.

It is an anonymous Function which implementation of Functional interface.

It reduces LOC (Line of Code).So, we no need to write again for implementation.

Syntax: Argument list -> {Body}

**Features of Java 8:**

* For Each in iterative Interface
* Functional Interface and Lambda Expression
* Default & Static methods in Interface
* Stream API
* Time API
* IO Improvement
* Concurrency Improvement
* Collection Improvement

**Functional Interface :**

it is used to achieve Functional Programming approach. It Provide SAM i.e., it may have default & static methods but allow only one Abstract Method. It is in util. function package.

It have 7 Major interface-

* Predicate – test()
* Bi-Predicate – test()
* Function – Apply()
* Bi-Function – Apply()
* Consumer – Accept()
* Bi- Consumer-Accept()
* Supplier – Get()

**Optional function:**

it is used to overcome Null Pointer Expection. It is many predefined methods such as isPresent () ,get(), isNullable, optional.of

**Method Reference:**

 It is Compact & Easiest form of Lambda Expression. It is used to refer methods of functional Interface. It use  :: Operator

Type- 1. Static, 2. Instance, 3. Constructor

**Stream:**

It is an Interface. It is one of the Feature in Java 8.

It is support collection framework.it is a Parallel Processing i.e., Pipeline Process.

We can re-use collection values but stream values cannot be re-use(it perform only one operation as it is in pipeline process).

Stream allows to perform operation such as filtering, mapping, sorting and reducing elements of collection easily.

Syntax:

Source-> Intermediate-> Terminal

(Intermediate it Optional but it have only one terminal and it is mandatory)

**IO Stream:**

It is a fundamental part of java IO Classes, they are used to read from and write to various source, such as files, in memory buffers etc..,

Two Types:

1. Byte Stream – Handling Binary Data such as read and writing bytes
2. Character Stream – Handling character data, provide higher level abstraction for reading and writing text.

**Buffer Reader Stream:**

It is a class that provide buffering capacities to read character from the character-input stream.

It reduces the number of actual read operation by reading large chunks of data at a time and storing them in the buffer

ReadLine () helps to read a Line at a time and return it as a String

JSP

(Java Server Page)

It is a Server Side Technology

It is a web based technology used to create dynamic and plat form independent web pages

JSP tags are used to add Java into HTML.

It is used to create Web Application

|  |  |
| --- | --- |
| **Servlet** | **JSP** |
| It is a Java Code | It is HTML Based Compilation code |
| It does not have inbuilt implicit object | It have inbuilt implicit object |
| It support all type of protocols | It support only HTTP Protocol |
| All packages are at the top of the program | We can import packages anywhere in the program |
| Faster | Slower- as the first step of JSP lifecycle is translation of JSP into java |
| It does not have pre-define templates | It has pre-defined templates |
| It is Html in Java | It is java in html so, easy to write |

**Web Application:**

It is used to create websites it may have static html page but the contents must be dynamic

Two Type:

Static Web Application: it have Static contents where every user can view the same information in each individual pages.

Dynamic Web Application: it allow the user it deliver content in the web browser and information may change based on the users

**MVC:**

Model View Controller is a Framework that separate the Presentation layer, business layer and database

View: it represent the presentation layer. i.e.., UI/UX – User Interface

Controller: it holds the Business Logics and act as an interface between view and the model. It intercept all the incoming request.

Model: it represent the state of the application. It is also called as Data Access Layer. It has Bean, Session, etc..,

Web Controller: it holds all the JSP and Servlet file of the controller.

**JDBC Connection:**

It is used to create and manage data from a database

Java API provide three programming activities

1. Connect to the Database
2. Send Query and update Statement on to the Database
3. Process the Result rendered as a response from the Databases

8 Steps to Connect using JDBC:

1. Import Packages
2. Load Driver Manager
3. Register Manager
4. Get Connection
5. Create statement
6. Execute Query
7. Process data
8. Close Connection

**Life Cycle of JSP:**

1. Translation of JSP
2. Compilation of JSP
3. Class Loader
4. Instance Creation
5. Initialization
6. Process data
7. Destroy object

**Bean Class or POJO Class:**

It is standard Java Class with attributes, getter and setter. It has special form of data in the server and it is mainly used to encapsulate the data.

**Implicit Object:**

These are the object provided by the JSP to the developer where they can call them without being explicitly declared.

Nine Implicit Object: Out, Request, Response, Config, Session, Application, Page, Page Content, Exception

**Ways to Achieve Servlet:**

1. Implementing Servlet Interface
2. Extending Generic Servlet
3. Extending HTTP Servlet

**Hierarchy of Servlet:**

Servlet - > Generic Servlet -> Http Servlet

**Class Loader:**

It is responsible to load a class in JVM during run time. As it a part of JRE, it does not need to know the underlying file to run a java program.

**HTTP Available Methods:**

1. Get
2. Put
3. Post
4. Patch
5. Delete

|  |  |
| --- | --- |
| **Generic Servlet** | **Http Servlet** |
| It is Independent | It is dependent but by implementing generic servlet we can make it dependent |
| It does not support Session and send redirect | It support Session and send redirect |
| Service Method is Abstract | Service Method is not Abstract |
| It implement all methods of Serialization and servlet interface | It implement only HTTP Specified method |
| It is Faster | It is Slower |

**Three Tags used in JSP**

1. Script let Tag
2. Expression Tag
3. Declaration Tag

|  |  |
| --- | --- |
| **Get** | **Post** |
| Limited Amount of Data can be transferred | Large Amount of data can be transferred |
| It is not secured as it sent data via header through URL | It is secured as it sent data through Body |
| It can be Bookmarked | It cannot be Bookmarked |
| It is idempdent | It is not idempdent |
| Mostly using as it is efficient | Less Efficient So, less in use |

**HTTP Session:**

It is used to create a session between client and the server. The container create a session ID for each user which help to identifier the corresponding user

Two Task:

1. Bind Object
2. View and manipulate the data for the HTTP session.

**Http Servlet:**

It is in javax.servlet. Http package, it is a protocol implementation as it support only http request

**Exception Handling in JSP:**

< Error Page > tag at the top of the JSP can handle the Exceptions.

**Life Cycle Method in JSP:**

1. Init ()
2. Service ()
3. Destroy ()
4. Get Servlet info ()
5. Get Servlet Config ()

**Patch ():**

It perform a partial update in a resource. It has less Band Width.

**Put ():**

It modify a resource fully and it has High Band Width.

**Idempotent:**

It refers to an operation that can be applied multiple times without changing the result beyond the initial application.

Spring BooT

|  |  |
| --- | --- |
| **Spring** | **Spring Boot** |
| It is a frame work which provide comprehensive programming for java based application | It is a module that provide RAD(Rapid Application Development) using Spring Framework |
| Auto Configuration is not Possible So, configuration is Manuel | Auto Configuration is Possible |
| Used to create Loosely coupling and reusable application | Used to create stand alone or enterprise application |
| It doesnot support inn-memory DB | It support in-memory database |
| It doesnot have inbuilt server like Tomcat | It have inbuilt server |

**Dependency Injection:**

It is a fundamental Aspect of Spring module in which the container inject object on to another object.

**Types:**

1. Setter Injection
2. Constructor Injection
3. Field Injection

Advantage:

1. Avoid Tightly Coupling
2. Simplifies Unit –Testing

|  |  |
| --- | --- |
| **Loosely Coupling** | **Tightly Coupling** |
| It is Independent | It is Dependent |
| Used in Disturbed System | Used in Parallel Processing System |
| Each Processor have own memory module | All Processor share same memory module |
| Class depend on interface rather them class | One class depend on other class |
| More Flexible and best for reusable of code | Less Flexible and reduce reusable of code |