JDBC QUESTONS

**Fundamental Steps in JDBC**

1. Import JDBC packages.
2. Load and register the JDBC driver.
3. Open a connection to the database.
4. Create a statement object to perform a query.
5. Execute the statement object and return a query resultset.
6. Process the resultset.
7. Close the resultset and statement objects.
8. Close the connection.

## To connect to data base jdbc code :

Driver manager.get connection (“pth db name “ , “username” , “password “);

## To assaign drgc to local variable :

Connection con = dm.gc (“ “);

If you don't close the db connection then what will happen?

If we don't close the connection, it will lead **to connection memory leakage**. Unless/until application server/web server is shut down, connection will remain activate even though the user logs out.

# SQL:

\*SQL querie based on Max function

SQL MAX() aggregate function is used **to return the maximum value from the provided numerical expression**

Write Database table What if there is 1000 Table What u do.

INSERT INTO table\_name (column\_list)

VALUES

(value\_list\_1),

(value\_list\_2),

...

(value\_list\_n);

## Cartisian join :

Cartesian product result-set contains the number of rows in the first table, multiplied by the number of rows in second table.it contain where clause.

## How to connect database in spring boot.:

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

</dependency>

## For oracle :

<dependency>

<groupId>com.oracle</groupId>

<artifactId>ojdbc7</artifactId>

<version>12.1.0.1</version>

</dependency>

update salary from a table SQL query.

UPDATE employees **SET salary = salary - 1 WHERE salary >= 2000**;

Delete duplicate records from table :

To delete the duplicate rows from the table in SQL Server, you follow these steps:

* Find duplicate rows using [GROUP BY](https://www.sqlservertutorial.net/sql-server-basics/sql-server-group-by/) clause or [ROW\_NUMBER()](https://www.sqlservertutorial.net/sql-server-window-functions/sql-server-row_number-function/) function.
* Use [DELETE](https://www.sqlservertutorial.net/sql-server-basics/sql-server-delete/) statement to remove the duplicate rows.

# join in sql.

jOIN is an SQL clause used to query and access data from multiple tables, based on logical relationships between those tables

## Joins Types:

SQL specifies **five types** of JOIN : INNER , LEFT OUTER , RIGHT OUTER , FULL OUTER and CROSS .

## Primary key foreign key unique key:

A UNIQUE constraint ensures that all values in a column are different. This provides uniqueness for the column(s) and helps identify each row uniquely.

The PRIMARY KEY constraint uniquely identifies each row in a table. It must contain UNIQUE values and has an implicit NOT NULL constraint.

Foreign key helps us to build the relation between 2 tables. It can contain repeated values.

## Inner and left join :

* **(INNER) JOIN:** Retrieves records that have matching values in both tables involved in the join. This is the widely used join for queries.

**SELECT** \*

**FROM** Table\_A

**JOIN** Table\_B;

**SELECT** \*

**FROM** Table\_A

**INNER** **JOIN** Table\_B;

* **LEFT (OUTER) JOIN:**Retrieves all the records/rows from the left and the matched records/rows from the right table.

**SELECT** \*

**FROM** Table\_A A

**LEFT** **JOIN** Table\_B B

**ON** A.col = B.col;

view from SQL :

A view in SQL is a virtual table based on the result-set of an SQL statement. A view contains rows and columns, just like a real table.

13)max,count query : max count ( \* ) ;

## Subqueries :

A Subquery or Inner query or a Nested query is **a query within another SQL query and embedded within the WHERE clause**.

* Single Row Subquery. Returns zero or one row in results.
* Multiple Row Subquery. Returns one or more rows in results.
* Multiple Column Subqueries. Returns one or more columns.
* Correlated Subqueries. ...
* Nested Subqueries.

### 1. What is ORM in Hibernate?

**Hibernate ORM** stands for **Object Relational Mapping**. This is a mapping tool pattern mainly used for converting data stored in a relational database to an object used in object-oriented programming constructs.

.

### What are the advantages of Hibernate over JDBC?

* **Clean Readable Code**
* **HQL (Hibernate Query Language):** Hibernate provides HQL which is closer to Java and is object-oriented in nature. This helps in reducing the burden on developers for writing database independent queries.
* **Transaction Management:** JDBC doesn't support implicit transaction management. Whereas, Hibernate implicity provides this feature.
* **Exception Handling:** Hibernate wraps the JDBC exceptions and throws unchecked exceptions like JDBCException or HibernateException.
* **Special Features:** Hibernate supports OOPs features like inheritance, associations and also supports collections. These are not available in JDBC.

**What is the difference btn Orm and jpa ?**

Object-Relational Mapping (ORM) is the process of converting Java objects to database tables. In other words, this allows us to interact with a relational database without any SQL. The Java Persistence API (JPA) is a specification that defines how to persist data in Java applications.

### What are some of the important interfaces of Hibernate framework?

### Hibernate core interfaces are:

* Configuration
* SessionFactory
* Session
* Criteria
* Query
* Transaction

### What is a Session in Hibernate?

A session is an object that maintains the connection between Java object application and database. Session also has methods for storing, retrieving, modifying or deleting data from database using methods like persist(), load(), get(), update(), delete(), etc..

### 5. What is a SessionFactory?

SessionFactory provides an instance of **Session**. It is a factory class that gives the Session **objects** based on the configuration parameters in order to establish the connection to the database.

### What do you think about the statement - “session being a thread-safe object”?

No, Session is not a thread-safe object which means that any number of threads can access data from it simultaneously.

### 7. Can you explain what is lazy loading in hibernate?

Lazy loading is mainly used for improving the application performance by helping to load the child objects on demand.

**Hibernate Configuration File** or **hibernate.cfg.xml** is one of the most required configuration files in Hibernate.

By default, this file is placed under the src/main/resource folder.  
The file contains database related configurations and session-related configurations.  
Hibernate facilitates providing the configuration either in an XML file (like hibernate.cfg.xml) or a properties file (like hibernate.properties).

### How do you create an immutable class in hibernate?

Immutable class in hibernate creation could be in the following way. If we are using the XML form of configuration, then a class can be made immutable by markingmutable=false. The default value is true there which indicating that the class was not created by default.

Differnce B/w jDbc / Hibernate :

JDBC enables developers to create queries and update data to a relational database using the Structured Query Language (SQL). ... JDBC is database dependent i.e. one needs to write different codes for different database.

Whereas **Hibernate is database independent and same code can work for many databases with minor changes**

**Difference between jdbc and hibernate jpa ?**

| **Sr. No.** | **Key** | **JDBC** | **Hibernate** |
| --- | --- | --- | --- |
| 1 | Basic | It is database connectivity technology | It is a framework, |
| 2 | Lazy Loading | It does not support lazy loading | Hibernate support lazy loading |
| 3 | Transaction management | We need to maintain explicitly database connection and transaction. | Hibernate itself manage all transaction |
| 4. | Caching | We need to write code for implementing caching | Hibernates provides two types of caching :  First level Cache  Second level cache  No Extra code is required to use first level cache. |
| 5. | Performance | Low performance | High Performance |

**What is rollback and commit in hibernate ?**

rollback() , Hibernate **rolls-back the database transaction**. Database handles rollback, thus removing newly created object.

 When commit is called on the transaction (usually by the outermost transaction interceptor), **it flushes the session**, calls session.

**What is transaction ?**

The @Transactional annotation is **the metadata that specifies the semantics of the transactions on a method**.

Can we connect to two databases using hibernate?

To communicate with another database server, you need another hibernate configuration file similar **to hibernate.** **cfg.** **xml**. For example, let's say we have a MySQL database server and a Microsoft SQL database server that will be used in an application.

**INTRERFACES OF HIBERNATE :**

* Session interface -this is the primary interface used by hibernate applications.
* Session session = sessionFactory.openSession();

...

* SessionFactory interface-this is a factory that delivers the session objects to hibernate application. ..

Get and load in Hibernate :

In hibernate, **get() and load()** are two methods which is used to fetch data for the given identifier. They both belong to Hibernate session class. Get() method return null, If no row is available in the session cache or the database for the given identifier whereas load() method throws object not found exception.

# Hibernate / jpa annotations :

@entity : it maps java Class with db.

@ID : it maps java class variable with primary key column of DB.

@column : when variable name changes in entity classis not same as column name in db. Then we map that with this annotation.

@generated value : this annotation helps us to auto \_ increment values in DB.

@table : we use this when entity class name and class name are not same.

@qualifier : when same interfaces are implemented by more than one class.then @autowiered will be confused to create required object, this confusion will be clear by this annotation.

# WEB SERVICES :

What is Web services in simple words?

A Web service is **a software service used to communicate between two devices on a network**.

It allows multiple applications built on different programming languages to communicate with each other without any trouble. It uses the internet for direct application-to-application interaction, and also allows you to expose business logic using API.

.

@RESTCONTROLLER = using this annotation spring boot know that we are using web services.

Steps :

Go to controller package.

New – class – rest controller – finish .

What u have done in webservices

Web services are **XML-based information exchange systems that use the Internet for direct application-to-application interaction**. These systems can include programs, objects, messages, or documents.

The Web Services architecture is based upon the interactions between three roles: **service provider, service registry and service requestor**

### What are the important features of Web services?

* Used to standardized XML messaging system.
* Not tied to any one programming language or any operating system.
* Available over the internet or private networks.
* Supports loosely coupled connections between systems.
* Can be synchronous or asynchronous.
* Supports the transparent exchange of data to facilitate business integration.
* Supports communication among different apps with HTML, XML, WSDL, SOAP, etc.

### 3. What are the different components of Web Services?

Soap and rest and xml.

### Write the difference between API and Web services.

**API (Application Programming Interface)**: It acts as an interface between two devices so that they can communicate with each other without any user intervention.

**Web Service**: It facilitates interaction between two devices over a network. They are widely used for exchanging data among systems or applications.

APIs.

| **API** | **Web Service** |
| --- | --- |
| It can be online or offline. | It must use a network. |
| They are lightweight architecture. | They require SOAP to send and receive network data therefore, are not lightweight architectures. |
| It can use any design style or protocol. | It can only use SOAP but sometimes UDDI, XML, RPC, and REST also. |
| It supports HTTP/HTTPS protocol and also supports XML and JSON. | It supports HTTP protocol and also supports XML. |
| It doesn’t require any network for its operation. | It requires a network for its operation. |
| They are open source and are used for XML. | They are not open source and are used to understand JSON (JavaScript Object Notation) |

### What are RESTful Web Services?

REST (Representational State Transfer) is a stateless client-server architecture style used for developing applications that are accessible over the web. It is a type of web service whose main goal is to make web services more effective. It can be defined as the web service that uses HTTP methods for implementing the REST architecture.

### What are the advantages of RESTful web service?

There are several advantages of RESTful web services as given below:

* Platform independent.
* Simple and easy to implement and test.
* Support different formats such as JSON, XML, HTML, etc.
* Can be written in different programming languages and executed on any platform.
* Lightweight, manageable, scalable, and reusable.
* Faster and provide better performance.
* Consume less bandwidth and resources.
* A lot of automation framework is available

### 20. Which protocol is used by RESTful web services?

The protocol used by RESTful web services is HTTP.

### What do you mean by SOAP? Write its advantages.

SOAP (Simple Object Access Protocol) is an XML-based protocol that is used to access web services.

# Web service annotations :

1. @Controller. This annotation is used to make a class as a web controller, which can handle client requests and send a response back to the client. ...

## @rest controller : defines rest implementation class to perform web services.

@get mapping : using web it fetch all data from db.

@post mapping : it will save data to db using w.b

@put mapping : using w.b it will update the data in db.

@deletemapping : using w.b it will delete the data in db.

@requestbody : it map json object content to java object.

@path variable : it reads data from web service variable.

@Entity

@Table:- The @Table annotation **specifies the name of the database table to be used for mapping**.

@Id:-It maps entity class variable with primary key column of database.

@Column:  **@Column annotation is used for Adding the column the name in the table of a particular MySQL database.**

@Onetoone

@onetomany

@manytoone

@manytomany

In hibernate, **get() and load()** are two methods which is used to fetch data for the given identifier. They both belong to Hibernate session class.

Bean Scopes :

 When the Spring Framework creates a bean, it associates a scope with the bean. A scope **defines the runtime context within which the bean instance is available**.

**Spring includes 7 different Bean scopes:**

* Singleton.
* Prototype.
* Request.
* Session.
* Global session.
* Application.
* WebSocket.

**DEPENDANCY INJECTION :**

Dependency Injection is the main functionality provided by [Spring](https://www.geeksforgeeks.org/introduction-to-spring-framework/) IOC(Inversion of Control). The Spring-Core module is responsible for injecting dependencies through either Constructor or Setter methods.

**Need for Dependency Injection:**  
Suppose class One needs the object of class Two to instantiate or operate a method, then class One is said to be **dependent** on class Two.

**Types of Spring Dependency Injection:**   
There are two types of Spring Dependency Injection. They are: 

**Setter Dependency Injection (SDI)**: This is the simpler of the two DI methods. In this, the DI will be injected with the help of setter and/or getter methods.

**Constructor Dependency Injection (CDI)**: In this, the DI will be injected with the help of [constructors](https://www.geeksforgeeks.org/constructors-in-java/).

# MICROSERVICES

### What do you mean by Microservice?

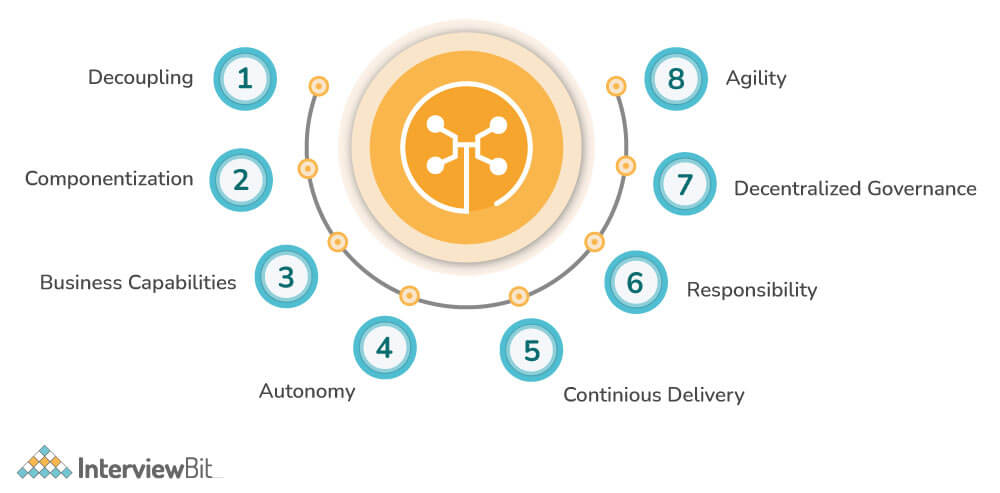
Microservices, also known as Microservices Architecture, is basically an SDLC approach in which large applications are built as a collection of s3mall functional modules. It is one of the most widely adopted architectural concepts within software development.

it helps deliver large, complex applications promptly, frequently, and reliably. Applications are modeled as collections of services, which are:

* Maintainable and testable
* Loosely coupled
* Independently deployable
* Designed or organized around business capabilities
* Managed by a small team

### Write main features of Microservices.

Some of the main features of Microservices include:



* **Decoupling**: Within a system, services are largely decoupled. The application as a whole can therefore be easily constructed, altered, and scalable
* **Componentization**: Microservices are viewed as independent components that can easily be exchanged or upgraded
* **Business Capabilities**: Microservices are relatively simple and only focus on one service
* **Team autonomy**: Each developer works independently of each other, allowing for a faster project timeline
* **Continuous Delivery**: Enables frequent software releases through systematic automation of software development, testing, and approval
* **Responsibility:** Microservices are not focused on applications as projects. Rather, they see applications as products they are responsible for
* **Decentralized Governance:** Choosing the right tool according to the job is the goal. Developers can choose the best tools to solve their problems
* **Agility:** Microservices facilitate agile development. It is possible to create new features quickly and discard them again at any time

### Write main components of Microservices.

Some of the main components of microservices include:

* Containers, Clustering, and Orchestration
* IaC [Infrastructure as Code Conception]
* Cloud Infrastructure
* API Gateway
* Enterprise Service Bus
* Service Delivery

### 3. What are the benefits and drawbacks of Microservices?

**Benefits:**

* Self-contained, and independent deployment module.
* Independently managed services.
* In order to improve performance, the demand service can be deployed on multiple servers.
* It is easier to test and has fewer dependencies.
* A greater degree of scalability and agility.
* Simplicity in debugging & maintenance.
* Better communication between developers and business users.
* Development teams of a smaller size.

**Drawbacks:**

* Due to the complexity of the architecture, testing and monitoring are more difficult.
* Lacks the proper corporate culture for it to work.
* Pre-planning is essential.
* Complex development.
* Requires a cultural shift.
* Expensive compared to monoliths.
* Security implications.
* Maintaining the network is more difficult.

### 9. Explain how you can override the default properties of Spring boot projects.

By specifying properties in the application.properties file, it is possible to override the default properties of a spring boot project.    
  
**Example:**  
In Spring MVC applications, you need to specify a suffix and prefix. You can do this by adding the properties listed below in the application.properties file.

* For suffix – spring.mvc.view.suffix: .jsp
* For prefix – spring.mvc.view.prefix: /WEB-INF/

### What are the challenges that one has to face while using Microservices?

The challenges that one has to face while using microservices can be both functional and technical as given below:   
  
**Functional Challenges:**

* Require heavy infrastructure setup.
* Need Heavy investment.
* Require excessive planning to handle or manage operations overhead.

**Technical Challenges:**

* Microservices are always interdependent. Therefore, they must communicate with each other.
* It is a heavily involved model because it is a distributed system.
* You need to be prepared for operations overhead if you are using Microservice architecture.
* To support heterogeneously distributed microservices, you need skilled professionals.

### 16. Explain how independent microservices communicate with each other.

Communication between microservices can take place through:

* HTTP/REST with JSON or binary protocol for request-response
* Websockets for streaming.
* A broker or server program that uses advanced routing algorithms.

# MVC ARCHITECTURE :

### What is the full form of MVC?

MVC is generally abbreviated as Model View Controller.

### What do you understand by Model View Control?

MVC is a software architecture or application design model containing 3 interconnected verticals or portions. These 3 portions are the model (data associated with the application), the view (which is the user interface of an MVC application), and the controller (the processes that are responsible for handling the input).

### Explain Model, View and Controller in brief

A model can be defined as the data that will be used by the program.

A view is a way of displaying objects (user interfaces) within an application. This is the particular vertical through which end users will communicate.

A controller is the third vertical which is responsible for updating both models and views. It accepts input from users as well as performs the equivalent update. In other words, it is the controller which is responsible for responding to user actions.

### Explain some benefits of using MVC?

**Some common benefits of MVC are –**

* **Support of multiple views:**
* **Faster development process:**
* **SEO-friendly development:**
* HTML, CSS and JavaScript than that of traditional WebForms.
* **Lightweight**

### How will you maintain the sessions in MVC?

The sessions of an MVC can be maintained by 3 possible ways:

* viewdata
* temp data and
* view bag

### How will you define the 3 logical layers of MVC?

**The 3 logical layers of MVC can be defined as follows:**

* Model logic which acts as a business layer
* View logic which acts as a display layer
* Controller logic acts as input control

### How to execute any MVC project? Explain its steps.

**For executing an MVC project, the steps followed are –**

1. Receive the first request for the application
2. Then, the routing is performed
3. Then, the MVC request handler is created
4. After that, the controller is created and executed
5. Then, the action is invoked
6. Then, the results are executed

### What is Spring MVC?

The Spring MVC or Spring Web MVC can be defined as a framework that provides a “Model View Controller” (MVC) architecture in the application

### Explain briefly the GET and POST Action types?

The GET Action Type is implemented for requesting the data from a particular resource. Using these GET requests, a developer can pass the URL (that is compulsory).  
The POST Action Type is implemented for submitting the data that needs to be handled to a certain resource. Using these POST requests, a developer can move with the URL, which is essential along with the data.

### What are the areas of benefits in using MVC?

**The area of benefits of using MVC is:**

* Unit testing becomes much easier.
* It permits its users to shape views, models, and controllers into 3 distinct operational sections within an application.
* It becomes easy to assimilate with other areas produced by another application.

# JSON :

**What is JSON? Explain.**

**Answer:** JSON is the abbreviation of JavaScript Object Notation. It is one of the simplest data interchange format, independent of programming language and platform. Its lightweight text-based structure makes it easily readable. It is derived from JavaScript for presenting simple data in the form of key-value pairs.

**What is meant by JSON objects?**

**Answer:** An object is defined as a set of key-value pairs. A JSON starts with a left brace “{“ and ends with another right brace “}”. Every key is followed by a colon “:” and the key-value pairs are separated from each other by using a comma “,”. So, basically, the JSON object is a collection of keys along with their values arranged in a pre-specified JSON format.

**What is the extension of the JSON file?**

**Answer:** A JSON file has an extension of “.json”. Being in a text-based format, a JSON file can be viewed or edited using any text editor like notepad

**What are the limitations of JSON?**

* As the data gets complex with several nested or hierarchical structures, it becomes complex for human readability.
* JSON is not suitable for handling very complex large data.
* JSON doesn’t have support for handling multimedia formats such as rich text or images.
* It doesn’t support comments.

**What are the advantages of JSON over XML?**

* JSON is lighter and faster than the XML.
* JSON has object types but XML doesn’t define objects as types.

JSON data can be easily accessed as a JSON object using JavaScript. On the other hand, the XML data need to be parsed and allocated to the variables using APIs

## Postman

Postman is a free, HTTP Client based software application primarily used to perform API testing. It supports testing of HTTP requests by utilizing GUI (Graphical User Interface) which can be executed and the responses can be validated.

### What is a collection in Postman?

### A collection in Postman helps to group similar requests. It helps in systematically arranging the requests into folders.

### Why do we use Postman?

### We use Postman for the below reasons:

* Firstly, Postman is free software that is useful for API testing.
* It can send HTTP requests of various types (such as GET, POST, PUT, PATCH, etc) and gives the ability to save environments for future use.
* It helps in managing the end-to-end lifecycle of the API -

### What are the different types of API requests supported in Postman?

Postman supports the following type of requests:

* GET
* POST
* PUT
* PATCH
* DELETE
* COPY

**What do you mean by Agile or Agile Methodology or Agile Process?**

 It is an iterative approach and each iteration is specially designed to be small and manageable so that it can be delivered in a specific given period of time. Agile methodologies are open to changing requirements over time and encourage constant feedback from end-users.

it is the most popular approach because, in this process, customers are also involved so that they can get updates regarding their product and also make sure whether or not they are meeting their requirements.

### Which process you used in Agile Methodology?

Different types of Agile methods or frameworks widely used in the world for software development and project development are listed below:

* **Scrum:** It is used to establish hypotheses, test them, reflect on the experience, and also make adjustments. It heavily depends on feedback, self-management, small teams, and work broken out into sprints. It relies on incremental development.

### What are advantages and disadvantages of Agile Process.

**Advantages**

There are several advantages of using the Agile Process as given below:

* Adapt well with changing requirements
* Face-to-face conversation with team members and customers
* Focuses on technical excellence and good design
* Fast and continuous development.
* Enables collaboration and interaction between client and project team.
* Ensure and promote customer satisfaction .
* Faster feedback from customers or end-users.
* Quick identification and elimination of errors found in the code
* Division of agile project into sprints or iterations i.e., short and repeatable phases typically 1-4 weeks long
* Quick delivery of products
* Easy to manage with more flexibility

**Disadvantages**

There are several disadvantages of using Agile Process as given below:

* Lack of formal documentation and designing
* Difficult to estimate resource requirement and effort
* Not good for small development projects
* Costly as compared to other development methodologies
* Requires more time and energy from everyone
* Risk of ever-lasting project

## SERVLETS :

### What is a Servlet?

A servlet is a small Java program that runs within a Web server. Servlets receive and respond to requests from Web clients, usually across HTTP, the HyperText Transfer Protocol.

### How do you write a servlet that is part of a web application?

To write a servlet that is part of a web application:  
Create a Java class that extends javax.servlet.http.HttpServlet.   
Import the classes from servlet.jar (or servlet-api.jar).   
These will be needed to compile the servlet.

### 3. What are some of the advantages of Servlets?

* Servlets provide a number of advantages over the other approaches
* A Servlet is convenient in modifying regular HTML.
* We can write the servlet code into the JSP.
* Servlets includes the feature of multithreading of java.
* We can make use of exception handling.
* Servlets have a separate layer of business logic in the application
* Easy for developers to show and process the information.
* Servlets provide a convenient way to modify HTML pages.

*Compared to other technologies, servlets have a number of advanta*ges:

* Platform and vendor independence
* Integration
* Efficiency
* Scalability
* Robustness and security

### What do you mean by Servlet chaining?

Servlet Chaining is a way where the output of one servlet is piped to the input of another servlet, and the output of that servlet can be piped to the input of yet another servlet and so on. Each servlet in the pipeline can either change or extend the incoming request. The response is returned to the browser from the last servlet within the servlet chain. In the middle, the output out of each servlet is passed as the input to the next servlet, so every servlet within the chain has an option to either change or extend the content.

# JSP :

### What is JSP?

JSP stands for Java Server Pages. This technology is used to create dynamic web pages in the form of [HyperText Markup Language](https://www.interviewbit.com/html-interview-questions/) (HTML). They have embedded Java code pieces in them. They are an extension to the Servlet Technology and generate Servlet from a page. It is common to use both servlets and JSP pages in the same web apps.

### What are some of the advantages of using JSP?

* Better performance and quality as JSP is a specification and not a product.
* JSP pages can be used in combination with servlets.
* JSP is an integral part of J2EE, a complete platform for Enterprise-class applications.
* JSP supports both scripting and element-based dynamic content

**the Scripting elements are:**

| **Element** | **Description** |
| --- | --- |
| <% ... %> | Scriptlet used to embed scripting code. |
| <%= ... %> | Expression, used to embed Java expressions when the result shall be added to the response. Also used as runtime action attribute values. |
| <%! ... %> | Declaration used to declare instance variables and methods in the JSP page implementation class. |

**What the different types of JSTL tags are ?**

* Core Tags.
* Formatting tags.
* SQL tags.
* XML tags.
* JSTL Functions.

JavaServer Pages Standard Tag Library

JSTL, which stands for **JavaServer Pages Standard Tag Library**, is a collection of custom JSP tag libraries that provide common Web development functionality.

# SPRING BOOT :

## What is Spring boot?

Sprint boot is a Java-based spring framework used for Rapid Application Development (to build stand-alone microservices). It has extra support of auto-configuration and embedded application server like tomcat, jetty, etc.

### What are the advantages of using Spring Boot?

The advantages of Spring Boot are listed below:

* Easy to understand and develop spring applications.
* Spring Boot is nothing but an existing framework with the addition of an embedded HTTP server and annotation configuration which makes it easier to understand and faster the process of development.
* Increases productivity and reduces development time.
* Minimum configuration.
* We don’t need to write any XML configuration, only a few annotations are required to do the configuration.

### What are the Spring Boot key components?

Below are the four key components of spring-boot:

* Spring Boot auto-configuration.
* Spring Boot CLI.
* Spring Boot starter POMs.
* Spring Boot Actuators.

### 3. Why Spring Boot over Spring?

Below are some key points which spring boot offers but spring doesn’t:

* Starter POM.
* Version Management.
* Auto Configuration.
* Component Scanning.
* Embedded server.
* InMemory DB.
* Actuators

### How does Spring Boot works?

Spring Boot automatically configures your application based on the dependencies you have added to the project by using annotation. The entry point of the spring boot application is the class that contains @SpringBootApplication annotation and the main method.

Spring Boot automatically scans all the components included in the project by using @ComponentScan annotation.

### 6. What does the @SpringBootApplication annotation do internally?

The @SpringBootApplication annotation is equivalent to using @Configuration, @EnableAutoConfiguration, and @ComponentScan with their default attributes. Spring Boot enables the developer to use a single annotation instead of using multiple.

### What is the purpose of using @ComponentScan in the class files?

 You need to add the @ComponentScan annotation for your class file to scan your components added to your project.

### 8. How does a spring boot application get started?

### What are starter dependencies?

Spring boot starter is a maven template that contains a collection of all the relevant transitive dependencies that are needed to start a particular functionality.  
Like we need to import spring-boot-starter-web dependency for creating a web application.

<dependency>

<groupId> org.springframework.boot</groupId>

<artifactId> spring-boot-starter-web </artifactId>

</dependency>

### 10. What is Spring Initializer?

### Spring Initializer is a web application that helps you to create an initial spring boot project structure and provides a maven or gradle file to build your code. It solves the problem of setting up a framework when you are starting a project from scratch.

### What is Spring Boot dependency management?

Spring Boot dependency management is used to manage dependencies and configuration automatically without you specifying the version for any of that dependencies.

### 15. Can we create a non-web application in Spring Boot?

Yes, we can create a non-web application by removing the web dependencies from the classpath along with changing the way Spring Boot creates the application context.

### 16. Is it possible to change the port of the embedded Tomcat server in Spring Boot?

Yes, it is possible. By using the **server.port** in the **application.properties**.

### Explain @RestController annotation in Sprint boot?

It is a combination of @Controller and @ResponseBody, used for creating a restful controller. It converts the response to JSON or XML. It ensures that data returned by each method will be written straight into the response body instead of returning a template.

### 22. What is the difference between @RestController and @Controller in Spring Boot?

@Controller Map of the model object to view or template and make it human readable but @RestController simply returns the object and object data is directly written in HTTP response as JSON or XML.

### What is the difference between RequestMapping and GetMapping?

RequestMapping can be used with GET, POST, PUT, and many other request methods using the method attribute on the annotation. Whereas getMapping is only an extension of RequestMapping which helps you to improve on clarity on request.

### How to get the list of all the beans in your Spring boot application?

Spring Boot actuator “/Beans” is used to get the list of all the spring beans in your application.

### What is dependency Injection?

The process of injecting dependent bean objects into target bean objects is called dependency injection.

* Setter Injection: The IOC container will inject the dependent bean object into the target bean object by calling the setter method.
* Constructor Injection: The IOC container will inject the dependent bean object into the target bean object by calling the target bean constructor.
* Field Injection: The IOC container will inject the dependent bean object into the target bean object by Reflection API.

### What is an IOC container?

IoC Container is a framework for implementing automatic dependency injection. It manages object creation and its life-time and also injects dependencies into the class.

ANNOTATIONS :

@Requestparam : @RequestParam **used for accessing the values of the query parameters**

**@pathvariable :**  @PathVariable used for accessing the values from the URI template.

**@Autowired**hen we use @Autowired annotation, the spring container auto-wires the bean by matching data-type.

**@Configuration:** It is a class-level annotation.

**@ComponentScan:** It is used when we want to scan a package for beans.

**@Bean:** It is a method-level annotation. It tells the method to produce a bean to be managed by Spring Container.

**@Controller:** The @Controller is a class-level annotation. It is a specialization of **@Component**. It marks a class as a web request handler.

**@Service:** It is also used at class level. It tells the Spring that class contains the **business logic**.

**@Repository:** It is a class-level annotation. The repository is a **DAOs** (Data Access Object) that access the database directly. The repository does all the operations related to the database.

**@RequestMapping:** It is used to map the **web requests**.

**@GetMapping:** It maps the **HTTP GET** requests on the specific handler method.

* **@PostMapping:** It maps the **HTTP POST**requests on the specific handler method.
* **@PutMapping:** It maps the **HTTP PUT** requests on the specific handler method.
* **@DeleteMapping:** It maps the **HTTP DELETE** requests on the specific handler method.
* **@PatchMapping:** It maps the **HTTP PATCH**requests on the specific handler method
* **@RequestBody:** It is used to **bind** HTTP request with an object in a method parameter.
* **@ResponseBody:** It binds the method return value to the response body. It tells the Spring Boot Framework to serialize a return an object into JSON and XML format.
* **@PathVariable:** It is used to extract the values from the URI. It is most suitable for the RESTful web service, where the URL contains a path variable.
* **@RequestParam:** It is used to extract the query parameters form the URL. It is also known as a**query parameter**. It is most suitable for web applications.
* **@RequestHeader:** It is used to get the details about the HTTP request headers. We use this annotation as a **method parameter**.
* **@RestController:** It can be considered as a combination of **@Controller** and **@ResponseBody**annotations**.** The @RestController annotation is itself annotated with the @ResponseBody annotation.
* **@RequestAttribute:** It binds a method parameter to request attribute.