* **JSON:-**
* **(**JavaScript Object Notation**)-** is a standard text-based format for representing structured data and data inter-change format
* It is commonly used for transmitting data in web applications
* It is easy for humans to read and write and for machines to prase and generate
* For example, sending some data from the server to the client, so it can be displayed on a web page, or vice versa
* The purpose of JSON is to store and transmit data objects consisting of attribute value and arrays and also other serializable values
* It is independent language
* It supports array, object, string, number and values.
* In JSON, data is represented in key-value pairs, and curly braces hold objects, where a colon is followed after each name. The comma is used to separate key-value pairs. Square brackets are used to hold arrays, where each value is comma-separated.
* JSON defines only two data structures: objects and arrays.
* An object is a set of name-value pairs

2. What is Json arrays?

* The array index begins with 0.
* The square brackets [...] are used to declare JSON array.
* JSON array are ordered list of values.
* JSON arrays can be of multiple data types.
* JSON array can store string, number, boolean, object or other array inside JSON array.
* In JSON array, values must be separated by comma.
* Arrays in JSON are almost the same as arrays in JavaScript.

3.Controller and Restcontroller?

* **Controller:-**
* The controller controls the data flow into model object and updates the view whenever data changes
* It keeps view and model separate
* In Spring framework, all the requests sent by the dispatcher servlet normally directed to a controller class.This controller class is mapping those requests to each process & execute the requested inputs. In a project there can be multiple controllers defined for different purposes.All these controllers refers to the same dispatcher servlet
* In a Controller mapping there are two types of mapping as GET & POST. Normally there can be many GET methods in a controller while one POST method is employed. GET request method is used to get the requests from the user do the desired work & output results into a view( jsp pages)
* The @Controller is a annotation to mark class as Controller Class in Spring
* **RestController:-**
* The @RestController annotation in Spring is nothing but a combination of @Controller and @ResponseBody annotation.
* It was added into Spring to make the development of RESTful Web Services in Spring framework easier.
* It used for which eliminates the need to annotate every request handling method of the controller class with the @ResponseBody annotation.

4.Soap and Restful Webservices?

* **Soap:- (Simple Object Access Protocol**)

It is a protocol specification for exchanging structured information in the implementation of web services in computer networks.\

SOAP is a messaging protocol. Messages (requests and responses) are **XML documents over HTTP**.

**The XML contract is defined by the** (Web Services Description Language). It provides a set of rules to define the messages, bindings, operations, and location of the service.

The XML used in SOAP can become extremely complex. For this reason, it is best to use SOAP with a framework like [JAX-WS](https://www.baeldung.com/jax-ws) or Spring

The SOAP web service endpoint class will handle all the incoming requests for the service. It will initiate the processing and will send the response back.

**Restful webservice:-**

Representational state transfer (REST) is **a software architectural style that defines a set of constraints to be used for creating Web services**

The main goal of RESTful web services is **to make web services more effective**. RESTful web services try to define services using the different concepts that are already present in HTTP. ... We can build REST services with both XML and JSON. JSON is more popular format with REST. The key abstraction is a resource in REST

Create a simple Spring Boot web application and write a controller class files which is used to redirects into the HTML file to consumes the RESTful web services.

Rest webservices are not protocol

5.What is the difference between web application and webservice application?

|  |  |
| --- | --- |
| web application | webservice application |
| These are accessed by web browsers | These are accessed by services, applications and sysytems |
| Developed using browser-oriented programming, scripting, frameworks and servers | Developed using standard programming languages |
| User interfaces are provided | User interfaces are not provided |
| Interaction models are not provided | Interactin models are Provided |
| Building blocks are page death,linkrot | Here, SOAP,WSDL |
| Operation modes are synchronous | Operation modes are synchronous and asynchronous |

6. Response body and response entity?

* **Response body:-**
* The response body consists of the resource data requested by the client.
* Response body Annotation indicating a method parameter should be bound to the body of the HTTP request
* @ResponseBody annotation can be put on a method and indicates that the return type should be written straight to the HTTP response body and not placed in a Model, or interpreted as a view name
* @ResponseBody is a Spring annotation which binds a method return value to the web response body. It is not interpreted as a view name. It uses HTTP Message converters to convert the return value to HTTP response body, based on the content-type in the request HTTP header
* **Response Entity:-**
* ResponseEntity represents the whole HTTP response: status code, headers, and body. As a result, we can use it to fully configure the HTTP response. If we want to use it, we have to return it from the endpoint
* ResponseEntity is a generic type.
* ResponseEntity is used when you need to change HTTP headers or HTTP status code based upon your business logic or incoming request.
* ResponseEntity wraps the original object as its body which is optional. If you want to return an object or null, ResponseEntity will work in either way

7. What is ddl and ddl-auto?

* **Ddl:-**
* A data definition language (DDL) is a computer language used to create and modify the structure of database objects in a database. These database objects include views, schemas, tables, indexes
* **Data Definition Language (DDL) commands:**
* CREATE to create a new table or database.
* ALTER for alteration.
* Truncate to delete data from the table.
* DROP to drop a table.
* RENAME to rename a table.
* **Ddl-auto:-**
* ddl-auto property is Spring Data JPA specific and is their way to specify a value that will eventually be passed to Hibernate under the property it knows, hibernate
* auto is a hibernate configuration property. It is used to validate and exports schema DDL to the database
* ddl-auto explicitly and the standard Hibernate property values are none , validate , update , create , and create-drop .
* **JPA repository:-**
* The Java Persistence API (JPA) is the standard way of persisting Java objects into relational databases.
* The JPA consists of two parts: a mapping subsystem to map classes onto relational tables as well as an EntityManager API to access the objects, define and execute queries
* The Spring Data JPA framework can then inspect that contract, and automatically build the interface implementation
* pagingAndSortingRepository:-
* PagingAndSortingRepository is an extension of CrudRepository to provide additional methods to retrieve entities using the pagination and sorting abstraction. It provides two methods
* Page findAll(Pageable pageable) – returns a Page of entities meeting the paging restriction provided in the Pageable object.
* Iterable findAll(Sort sort) – returns all entities sorted by the given options. No paging is applied here.
* **CRUDrepository:-**
* CrudRepository provides generic CRUD operation on a repository for a specific type. CrudRepository is a Spring data interface and to use it we need to create our interface by extending CrudRepository
* The CrudRepository interface provides methods for CRUD operations, so it allows you to create, read, update and delete records without having to define your own methods.
* What are mappings?
* **One-to-one mapping:-**
* The One-To-One mapping represents a single-valued association where an instance of one entity is associated with an instance of another entity. In this type of association one instance of source entity can be mapped atmost one instance of target
* **One-to-many mapping:-**
* A one-to-many relationship between two entities is defined by using the @OneToMany annotation in Spring Data JPA
* It declares the mappedBy element to indicate the entity that owns the bidirectional relationship.
* **Many-to-one mapping:-**
* A many-to-one relationship is where one entity contains values that refer to another entity a column or set of columns that has unique values.
* **Many-to-one mapping:-**
* @ManyToMany annotation to create a many-to-many relationship between two entities. In a bi-directional association, the @ManyToMany annotation is used on both the entities

10. What are relations in parent child table is-a, uses-a,has-a?

* An IS-A relationship is inheritance. The classes which inherit are known as sub classes or child classes.
* HAS-A relationship is composition and  HAS A relationship means the class is using another class, so it has it as a member

11.What is transient in JPA?

* Transient is a variables modifier used in serialization. At the time of serialization, if we don't want to save value of a particular variable in a file, then we use transient keyword
* @Transient Annotation in JPA is used to indicate that a fiels is not to be persisted or ignore fields to save in the database
* @Transient exist in javax.persistent package.It is used to annotate a property or field of nan entity class,mapped superclass or embedded class