**1) What are web services?**

To explain web services in layman terms we can take an example of English language. As we know most of the countries have their own language which other countries can’t understand. So In this case they make use of English as the common language and this is a widely used language to bridge the gap between countries.  
In the similar way we have web services in the IT world where different languages and platforms can communicate with one another. E.g. Java applications can communicate with .NET or Angular applications can communicate with Java applications.  
In web services we have two parties, one is the service provider who provides the data/services and another is the service consumer who gets the required data/services.   
There are two types of web services, SOAP & RESTful web services. The communication between both parties is via http protocol. In SOAP the data exchange format is XML and in RESTful it uses multiple data formats E.g. JSON, Text etc.

**2) What is REST?**

REST stands for representational state transfer. HTTP is there for quite some time now and almost all the websites use it. Roy Fielding did research on how HTTP can be used for other types of applications and that’s where RESTful services was born.  
HTTP has various methods such as GET, POST, PUT, DELETE which we can use to perform an operation against a URL/URI. You can relate it to the CRUD (Create, Read, Update and Delete) operations you perform against the database which are very common in all the applications and across the industry.  
The data exchange between service provider and service consumer can be in JSON, XML or any other format which makes RESTful services Interoperable.

**3) What do you mean by REST is stateless**

This is one of the biggest advantages of REST in that it is stateless. Statelessness does not mean that it does not have state at all, but it exchanges the representational state between applications. The state is maintained at the client side instead of maintaining it at the server side.

**4) What are the different HTTP methods present in REST**

Some of the commonly used HTTP methods are as below.

**GET:** This is used to fetch/get data from the server.  
**POST:** This is used to create a new resource.  
**PUT:** This is used to modify a resource  
**DELETE**: This is used to delete a resource

#### 5) What is a resource in REST

Everything that we access is called a resource. Each resource has something called as identifier which is called as URI (uniform resource identifier)  
**E.g.** Below a  
/employees/1 – URI for accessing an employee resource with id 1.  
/employees     – URI for accessing all the employees  
  
and the action that we perform on resource is using the HTTP methods i.e. GET, POST, PUT, DELETE. We will see the meaning of each HTTP method in the questions below.

#### 6) What is the difference between POST and PUT

POST is used to create a new resource and PUT is used to update an existing resource.

#### 7) What do you mean by a resource in REST

In Object oriented programming we have objects, similarly we have resources in REST. Operations are performed against a resource. Resources can be in XML or JSON representation.

#### 8) Mention some of the implementations of REST

Some of the common implementations used to implement REST web services are Apache CXF, Jersey, REST Easy & Spring REST. Spring REST is used most widely today.

#### 9) What are the tools used to test the REST web services

There are 2 parts to REST web services, one is service consumer and another is service provider. Let’s say you are a service provider and your service is invoked by multiple consumers. Before releasing/deploying your final code you have to test it properly using some standalone tools. Two of the very common tools are Postman & SOAP UI.

#### 10) What does a REST request consist of

REST request consists of request headers and request body. Mandatories of both depends on your business logic.

#### 11) Name some of the HTTP response codes

**200 – OK**  
Operation successful  
**400 – BAD REQUEST**  
For missing Mandatory headers or any other parameter  
**404 – NOT FOUND**  
If a particular URI is not present or wrong.   
**405 – METHOD NOT ALLOWED**  
If URI has a different http method and if the invoking http method is different. E.g. if /getEmployee has a GET http method and if you are invoking the endpoint with POST http method.  
**401 – UNAUTHORIZED**  
If the user is not authenticated properly then we return 401  
**403 – FORBIDDEN**  
If the user is authenticated but he does not have rights to access the resource.

#### 12) What are the spring annotations used for HTTP methods.

@GetMapping – To fetch a resource  
@PostMapping – To create a new resource  
@PutMapping – To update an existing resource  
@DeleteMapping – To delete a resource

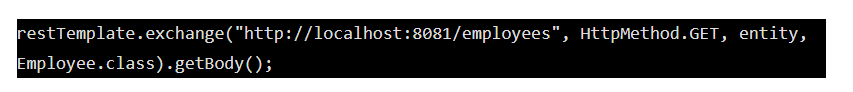
#### 13) What is @RestController?

@RestController = @Controller + @ResponseBody

This annotation is the combination of 2 annotations. By adding this we don’t need to explicitly add @ResponseBody annotation. The class annotated with @RestController has REST endpoint methods exposed so that other services can consume.

#### 14) What is RestTemplate and why do we use it?

If you want to consume a RESTful service, then you can make use of RestTemplate. We use the exchange () method to invoke the service.

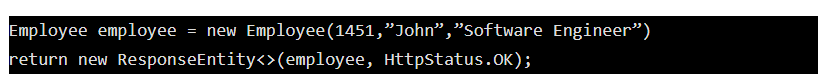


First parameter to the exchange method is the URI, second parameter is the HTTP method i.e. GET, POST, PUT or DELETE, third parameter is the entity which consists of headers & body or just header/body and fourth parameter containing the return type of the response that the service will return.

#### 15) What is ResponseEntity?

Being a RESTful service provider if we want to return a response body that contains Data as well as the response code then we make use of ResponseEntity

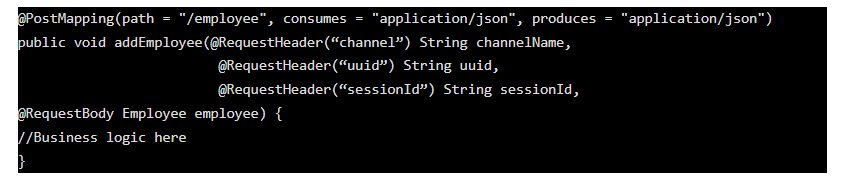
E.g. let’s say we have a URI or a method getEmployee which returns an Employee object. Along with the Employee object we want to return a response code say 200, then we would return as



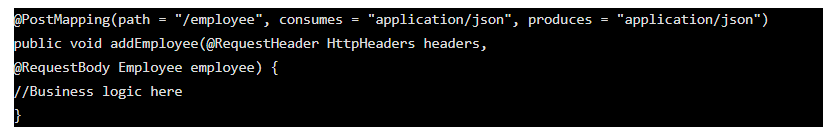
#### 16) @RequestHeader OR @RequestHeaders annotation

Both the annotations are used to receive values from the request headers.

-> @RequestHeader annotation is used to get each header one by one whatever is required.



**@RequestHeaders** are used to get all the headers into one object. This can be used when there are too many headers and you want all instead of fetching it one by one.

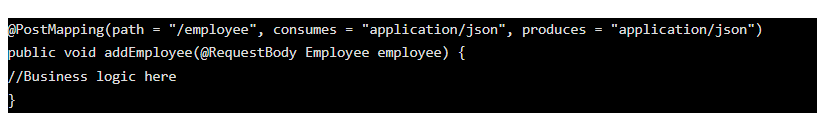


#### 17) What do you mean by Accept and Content-Type header

Both are keys that are present in the headers sent from the client (E.g. browsers) to the service  
**Accept:** It contains the value which specifies what response format the client is expecting. E.g. application/json  
**Content-Type:** It contains the value which specifies which format of the data sent to the service.

#### 18) Produces and consumes in REST

Produces specifies the MIME media type that is sent to the client and Consumes specifies the MIME media type that is accepted or consumed from the client.  
E.g. The consumer of the below service has to send the request in the JSON format.



#### 19) What are the best practises in developing RESTful services?

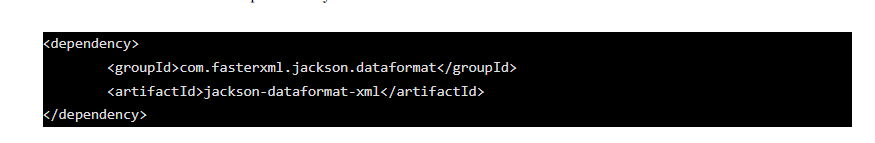
As we know, there are two parties involved in REST communication. Service consumer and Service provider. Service provider is the one who has exposed the web service and consumers are the one who are accessing the web service.  
  
While developing we need to make the best use of HTTP methods, make semantic use of methods properly.E.g. If you are providing data to the consumer then use HTTP GET, if you are creating a new resource then make use of HTTP POST.  
  
Also make the best use of HTTP Response codes. E.g. If the operation is successful send a 200 response. If the request format is not proper or some of the header is missing we need to send a 400 response. I did see some developers sending 500 Internal Server which is wrong. We need to send the correct response codes  
  
Avoid sending sensitive data in the URIs and headers.  
While designing the URIs, make use of plural forms. For example  
/emloyees/123 – CORRECT  
/employee/123 – WRONG

#### 20) ****What are HttpMessageConverters in Spring REST and why does it return JSON format by default?****

HttpMessageConverters are used to convert the object to JSON/XML and vice versa. While developing REST service in Spring Boot, the default data format that is being returned in JSON. This is because it registers the default MessageConvertor i.e. Jackson.

#### 21) How to add XML support for our endpoint?

As we know that JSON is the default data format that REST service returns, the question is how we send the response in XML format if the client requests it. If we are using spring boot then we need to add below dependency.



#### 22) What is Swagger in the REST web services world?

As per the Open API/Swagger specification “The OpenAPI Specification (OAS) defines a standard, language-agnostic interface to RESTful APIs which allows both humans and computers to discover and understand the capabilities of the service without access to source code, documentation, or through network traffic inspection”  
  
Swagger is a document or contract which is either .yml or .json format which has below details  
-The list of endpoints present in our service and their respective HTTP method.  
-Header that we need to pass to the consumer.  
-The data definition i.e. request body that a service consumer needs to send, and in return what the endpoint/service provider will return.   
-Http Response code being returned for different scenarios.  
  
Usually the service provider will prepare the document and share it with one or more service consumers, so that both parties work independently based on the agreed contract. Sometimes service providers & service consumers will discuss together to design the final contract/swagger file.