* What are the Basic features of http?
* Http is **connectionless**: Client and server knows each other only during the request-response flow time. After the response reaches the client, the client disconnect the connection. **Keep alive header** can be used to keep the connection live
* Http is **stateless**: Http is connectionless because it is stateless. Neither the client nor the browser can retain information between different requests across the web pages. In case if we want to retain the information, **state cookies** can be used
* Http is **media-independent:** It means, any type of data can be sent by HTTP as long as both the client and the server know how to handle the data content. It is required for the client as well as the server to specify the content type using appropriate **MIME-type**.
* What are request methods in http?
  + - HTTP defines a set of **request methods** to indicate the desired action to be performed for a given resource
    - 1) **GET**: request for a representation of the resource. Used to retrieve the data
    - 2**) HEAD**: Request is similar to GET, but we are requesting for a response without response body
    - 3) **POST**: Used to submit an entity to the specified resource, often causing a change in state or side effects on the server.
    - 4) **PUT**: changes all current representation of a target resource with a request payload.
    - 5) **DELETE**: deletes a specifies resource.
    - 6) **PATCH**: to apply partial modification to a resource
* WHAT IS THE DIFFERENCE BETWEEN GET AND POST METHODS?
* **GET is used to request data from a specified resource.**
  + **It can be cached, can remain in browser history,can be bookmarked, should never be used while dealing with sensitive data,have length restrictions,are only used to request data**
* **POST is used to send data to a server to create/update a resource.**
  + It cannot be cached, donot remain in browser history, cannot be bookmarked, don’t have length restrictions
* WHAT IS STATUS CODE IN HTTP?
* The Status-Code element in a server response, is a 3-digit integer where the first digit of the Status-Code defines the class of response and the last two digits do not have any categorization role
* 1xx: informational
* 2xx: success
* 3xx: redirection
* 4xx: client error
* 5xx: server error
* WHAT ARE THE HEADER FIELDS IN HTTP?
* HTTP header fields provide required information about the request or response, or about the object sent in the message body. There are four types of HTTP message headers:
  + General-header: have general applicability for both request and response messages.
  + Client Request-header: only applicable to request
  + Server Response header: only applicable to response
  + Entity header: define meta information about the entity-body or, if no body is present, about the resource identified by the request.
* WHAT IS URI?
* is a string of characters that identifies a resource on the web either by using location, name or both. It allows uniform identification of the resources
  + Is used to distinguish one resource from other regardless of the method used.
  + doesn’t contains the protocol specification
  + is the superset of URL
* WHAT ARE IDEMPOTENT METHODS AND WHY DO WE CALL THEM?
* An idempotent HTTP method is a HTTP method that can be called many times without different outcomes. It would not matter if the method is called only once, or ten times over. The result should be the same
* GET,HEAD,PUT and DELETE are idempotent methods
* EXPLAIN HTTP REQUEST AND RESPONSE MESSAGES?
* HTTP messages are how data is exchanged between a server and a client. There are two types of messages: requests sent by the client to trigger an action on the server, and responses, the answer from the server.
* Both share similar structure:
  + A **start-line** describing the requests to be implemented, or its status of whether successful or a failure. This start-line is always a single line
  + An *optional set of***HTTP headers** specifying the request, or describing the body included in the message
  + **A blank line** indicating all meta-information for the request has been sent.
  + An optional **body** containing data associated with the request (like content of an HTML form) or the document associated with a response. The presence of the body and its size is specified by the start-line and HTTP headers.
* WHAT IS SESSION STATE IN HTTP?
* is a method to keep track of the a user session during a series of HTTP requests
* The HTTP protocol is stateless, which means that HTTP has no built-in way to keep track of a user as they navigate from one webpage to another. As a result, there are a number of other methods used to maintain state. These include session state, cookies, hidden form fields , passing variables through the querystring, and form posts.
* WHAT IS HTTPS?
* Hypertext transfer protocol secure (HTTPS) is the **secure** version of HTTP, which is the primary protocol used to send data between a web browser and a website. HTTPS is encrypted in order to increase security of data transfer. This is particularly important when users transmit sensitive data, such as by logging into a bank account, email service, or health insurance provider
* EXPLAIN REST AND RESTFUL?
* **Representational State Transfer** is basically an architectural style of development having some principles:
  + It should be stateless
  + It should access all the resources from the server using only URI
  + It does not have inbuilt encryption
  + It does not have session
  + It uses one and only protocol- HTTP
  + For performing CRUD operations, it should use HTTP verbs such as get, post, put and delete
  + It should return the result only in the form of JSON or XML, atom, OData etc. (lightweight data )
* A service based on REST is called a "**RESTFUL** service".
* MENTION WHAT ARE THE HTTP METHODS SUPPORTED BY REST?
* GET
* POST
* PUT
* PATCH
* DELETE
* HEAD
* OPTIONS
* EXPLAIN THE ARCHITECTURAL STYLE FOR DEVELOPING WEB API?
* REST, or Representational State Transfer, is an architectural style for providing standards between computer systems on the web, making it easier for systems to communicate with each other. REST-compliant systems, often called RESTful systems, are characterized by how they are stateless and separate the concerns of client and server
  + Seperation of client and server: code on the client side can be changed at any time without affecting the operation of the server, and the code on the server side can be changed without affecting the operation of the client.
  + Statelessness: the server does not need to know anything about what state the client is in and vice versa
  + Communication between client and server:  clients send requests to retrieve or modify resources, and servers send responses to these requests
  + Making requests: REST requires that a client make a request to the server in order to retrieve or modify data on the server. A request generally consists of:
    - Http verb
    - Header
    - Path to a resource
    - Optional message body containing data
  + Sending responses: consist of
    - Content-type
    - Response code
* EXPLAIN RESTFUL WEB SERVICES
* **Restful Web Services** is a lightweight, maintainable, and scalable service that is built on the REST architecture. Restful Web Service, expose API from your application in a secure, uniform, stateless manner to the calling client. The calling client can perform predefined operations using the Restful service. The underlying protocol for REST is HTTP. REST stands for REpresentational State Transfer
* The key elements of a RESTful implementation are as follows:
  + **Resources**
  + **Request Verbs**
  + **Request Headers**
  + **Request Body**
  + **Response Body**
  + **Response Status codes**
* Restful methods: GET,POST,PUT,PATCH,DELETE,HEAD,OPTIONS
* EXPLAIN WHAT IS A RESOURCE IN REST?
* REST architecture treats every content as a resource. These resources can be Text Files, Html Pages, Images, Videos or Dynamic Business Data. REST Server simply provides access to resources and REST client accesses and modifies the resources. Here each resource is identified by URIs/ Global IDs. REST uses various representations to represent a resource where Text, JSON, XML. The most popular representations of resources are XML and JSON.
* WHICH PROTOCOL IS USED BY RESTFUL SERVICES?
* http protocol
* WHAT IS MESSAGING IN RESTFUL WEB SERVICES?
* RESTful Web Services make use of HTTP protocols as a medium of communication between client and server. A client sends a message in form of a HTTP Request and the server responds in the form of an HTTP Response. This technique is termed as Messaging. These messages contain message data and metadata i.e. information about message itself.
* HTTP request: consists of Verb, URI, http version,Request header, Request Body
* HTTP response: consists of status/response code, http version,Response header, response body
* STATE THE CORE COMPONENTS OF HTTP REQUEST?
* http version: indicates version
* request body: represents message content
* request header: Contains metadata, such as cache settings and client type, for the HTTP request message
* URI : identifies the resource on the server
* Verb: indicates http methods like POST,GET,PUT etc
* STATE THE CORE COMPONENTS OF HTTP RESPONSE?
* http version: indicates version
* response body: represents message content
* response header: Consists of metadata, like content length and server length, for the HTTP response message
* status/response code: Indicates the server status for the requested resource
* WHAT DO YOU UNDERSTAND ABOUT PAYLOAD IN RESTFUL WEB SERVICE?
* A payload in Restful web services is the actual data pack that is sent with the GET method in HTTP. It is the crucial information that you submit to the server when you are making an API request. The payload can be sent or received in various formats, including JSON.
* EXPLAIN THE CACHING MECHANISM.
* Caching refers to storing the server response in the client itself, so that a client need not make a server request for the same resource again and again. A server response should have information about how caching is to be done, so that a client caches the response for a time-period or never caches the server response.
* Following are the headers which a server response can have in order to configure a client's caching:
  + Date
  + Last modified
  + Cache- control
  + Expires
  + Age
* LIST THE MAIN DIFFERENCE BETWEEN REST AND SOAP
* SOAP stands for Simple Object Access Protocol whereas REST stands for Representational State Transfer
* SOAP is a protocol whereas REST is an architectural pattern.
* SOAP uses service interfaces to expose its functionality to client applications while REST uses Uniform Service locators to access to the components on the hardware device
* SOAP needs more bandwidth for its usage whereas REST doesn’t need much bandwidth
* SOAP only works with XML formats whereas REST work with plain text, XML, HTML and JSON
* SOAP cannot make use of REST whereas REST can make use of SOAP.
* ENLIST ADVANTAGES AND DISADVANTAGES OF STATELESSNESS
* Advantages:
  + Web services can treat each method request independently.
  + Web services need not maintain the client's previous interactions. It simplifies the application design.
  + As HTTP is itself a statelessness protocol, RESTful Web Services work seamlessly with the HTTP protocols.
  + As the server does not need to manage any session, deploying the services to any number of servers is possible, and so scalability will never be a problem
  + As the service calls (requests) can be cached by the underlying application, the statelessness constraint brings down the server's response time, that is, it improves performance with regard to response time
* Disadvantages:
  + Web services need to get extra information in each request and then interpret to get the client's state in case the client interactions are to be taken care of.
  + Using cookies is less efficient than any in-memory system would be. Webmasters have the responsibility of manipulating and maintaining the cookies