**HTTP Fundamentals**

1. What are the basic Features of HTTP?

* **HTTP is connectionless :** The HTTP client, i.e., a browser initiates an HTTP request and after a request is made, the client waits for the response. The server processes the request and sends a response back after which client disconnect the connection. So client and server knows about each other during current request and response only. Further requests are made on new connection like client and server are new to each other.
* **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.
* **HTTP is stateless:** As mentioned above, HTTP is connectionless and it is a direct result of HTTP being a stateless protocol. The server and client are aware of each other only during a current request. Afterwards, both of them forget about each other. Due to this nature of the protocol, neither the client nor the browser can retain information between different requests across the web pages.

1. 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.

* GET : Get is used to retrieve an information.
* POST : Method is used to submit an entity to the specified resource.
* PUT :  Replaces all current representations of the target resource with the request payload.
* PATCH: method is used to apply partial modifications to a resource.
* DELETE : This method is used to delete an specific resource .

1. What are the differences between GET and POST methods?

Get method is used to retrieve an specific resource , but POST method is used create new resource or will create a new resource.

1. What is status code in HTTP?

Status codes are issued by a server in response to a client's request made to the server

Informational response:(100-199)

Success response:(200- 299)

Redirection response :(300-309)

Client response:(400-499)

Server response:(500-599)

1. 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:** These header fields have general applicability for both request and response messages.

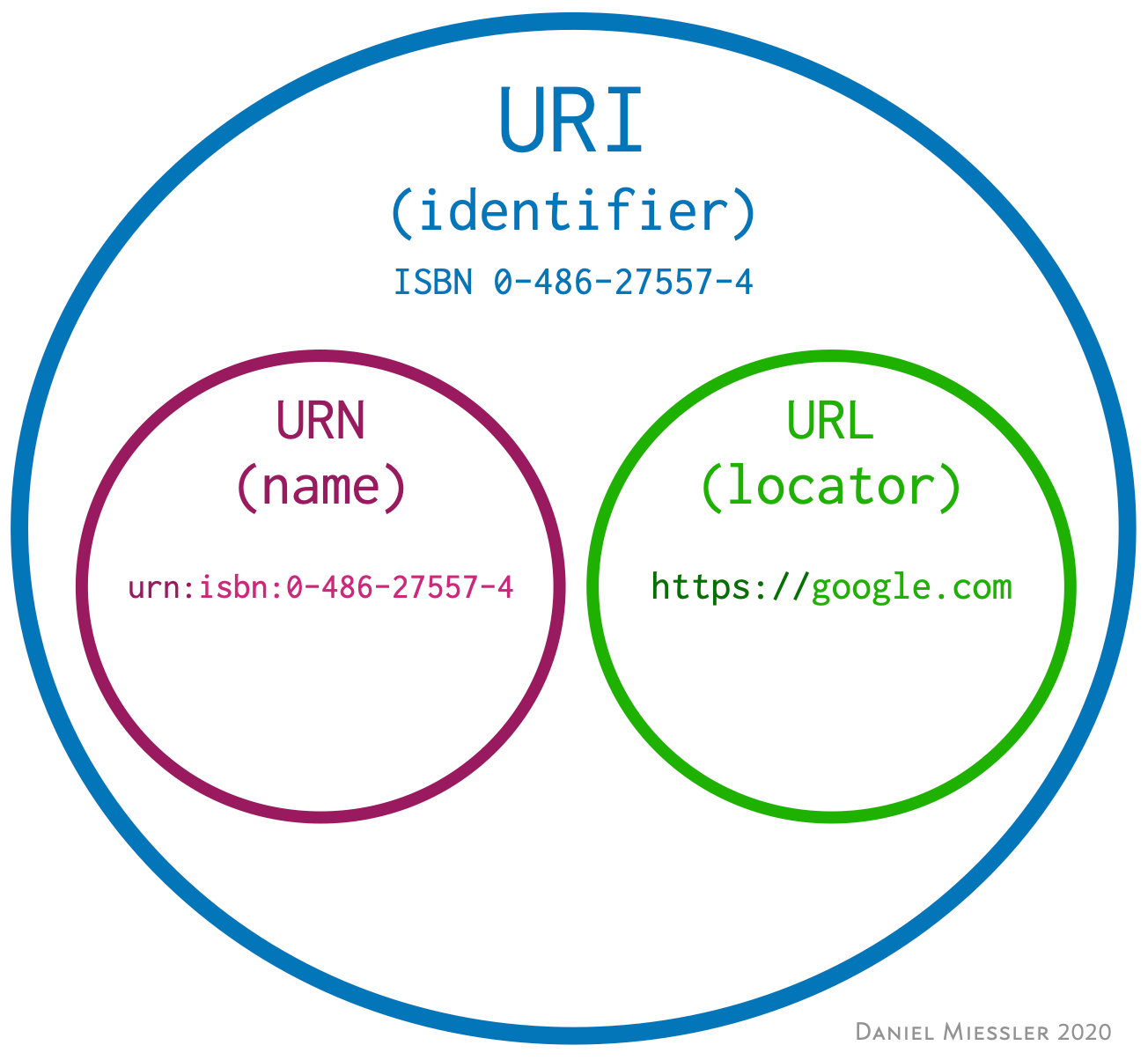
**Client Request-header:** These header fields have applicability only for request messages.

**Server Response-header:** These header fields have applicability only for response messages.

**Entity-header:** These header fields define meta information about the entity-body or, if no body is present, about the resource identified by the request.

1. What is URI?

URI stand for uniform resource identifier, A Uniform Resource Identifier is a unique sequence of characters that identifies a logical or physical resource used by web technologies.



URIs provide a means of locating and retrieving information resources on a network.

Format : URI = scheme:[//authority]path[?query][#fragment]

1. 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.

Idempotency is important in APIs because a resource may be called multiple times if the network is interrupted. In this scenario, non-idempotent operations can cause significant unintended side-effects by creating additional resources or changing them unexpectedly.

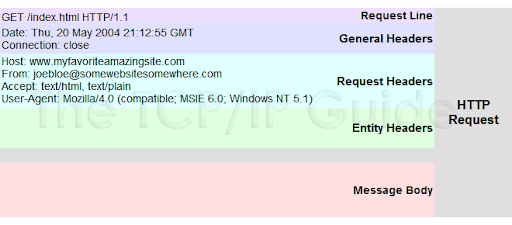
1. Explain HTTP Request & Response Messages?

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 .

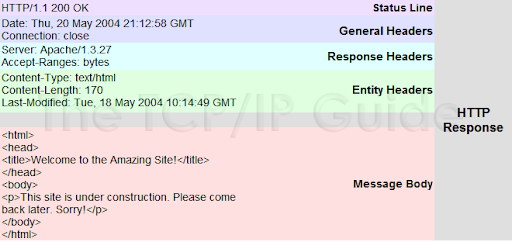
HTTP requests, and responses, share similar structure and are composed of:

* 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.

**HTTP REQUEST**



**HTTP RESPONSE**



1. What is Session State in HTTP?

Session state is a method keep track of the a user session during a series of HTTP requests. Session state allows a developer to store data about a user.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.

1. What is HTTPS?

Hypertext Transfer Protocol Secure (HTTPS) is an extension of the [Hypertext Transfer Protocol](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) (HTTP). It is used for [secure communication](https://en.wikipedia.org/wiki/Secure_communications) over a [computer network](https://en.wikipedia.org/wiki/Network_operating_system), and is widely used on the Internet In HTTPS, the [communication protocol](https://en.wikipedia.org/wiki/Communication_protocol) is encrypted using [Transport Layer Security](https://en.wikipedia.org/wiki/Transport_Layer_Security) (TLS) or, formerly, Secure Sockets Layer (SSL). The protocol is therefore also referred to as HTTP over TLS, or HTTP over SSL.The main purpose of HTTPS are [authentication](https://en.wikipedia.org/wiki/Authentication) of the accessed [website](https://en.wikipedia.org/wiki/Website), and protection of the [privacy](https://en.wikipedia.org/wiki/Information_privacy) and [integrity](https://en.wikipedia.org/wiki/Data_integrity) of the exchanged data while in transit. It protects against [man-in-the-middle attacks](https://en.wikipedia.org/wiki/Man-in-the-middle_attack), and the bidirectional [encryption](https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation) of communications between a client and server protects the communications against [eavesdropping](https://en.wikipedia.org/wiki/Eavesdropping) and [tampering](https://en.wikipedia.org/wiki/Tamper-evident#Tampering). , HTTPS signals the browser to use an added encryption layer of SSL/TLS to protect the traffic. SSL/TLS is especially suited for HTTP, since it can provide some protection even if only one side of the communication is [authenticated](https://en.wikipedia.org/wiki/Authentication). with HTTP transactions over the Internet, where typically only the [server](https://en.wikipedia.org/wiki/Web_server) is authenticated (by the client examining the server's [certificate](https://en.wikipedia.org/wiki/Public_key_certificate)). HTTPS creates a secure channel over an insecure network. Web browsers know how to trust HTTPS websites based on [certificate authorities](https://en.wikipedia.org/wiki/Certificate_authority) that come pre-installed in their software. Certificate authorities are in this way being trusted by web browser creators to provide valid certificates.

**Introduction to API**

1. Explain REST and RESTFUL?

REST :

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.In rest architecture the client and server communicate each other using HTTP protocol,ie the client send an HTTP request to an server and the server on receiving the request will process the request and send back the response to the client.

There are certain characteristics to rest

* Client-server: REST applications have a server that manages application data and state. The server communicates with a client that handles the user interactions. A clear separation of concerns divides the two components. This means you can update and improve them in independent tracks.
* Stateless: servers don’t maintain any client state. Clients manage their application state. Their requests to servers contain all the information required to process them.
* Cacheable: servers must mark their responses as cacheable or not. So, infrastructures and clients can cache them when possible to improve performance. They can dispose of non-cacheable Information, so no client uses stale data
* Uniform interface: this constraint is REST’s most well known feature or rule, depending on who you ask. Fielding says “The central feature that distinguishes the REST architectural style from other network-based styles is its emphasis on a uniform interface between components.” REST services provide data as resources, with a consistent namespace.
* Layered system: components in the system cannot “see” beyond their layer. So, you can easily add load-balancers and proxies to improve security or performance.
* Code on demand.

**RESTFUL :**

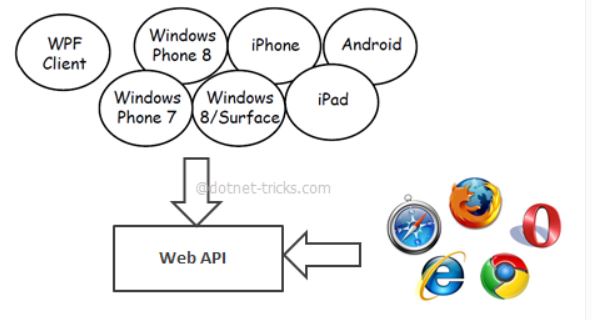
Restful Web Services are basically REST Architecture based Web Services. In REST Architecture everything is a resource. RESTful web services are light weight, highly scalable and maintainable and are very commonly used to create APIs for web-based applications.

1. Mention what are the HTTP methods supported by REST?

* GET : Get is used to retrieve an information.
* POST : Method is used to submit an entity to the specified resource.
* PUT :  Replaces all current representations of the target resource with the request payload.
* PATCH: method is used to apply partial modifications to a resource.
* DELETE : This method is used to delete an specific resource .

1. Explain the architectural style for creating web API?

WEB API : Today, a web-based application is not enough to reach it's customers. People are very smart, they are using iphone, mobile, tablets etc. devices in its daily life. These devices also have a lot of apps for making the life easy. Actually, we are moving from the web towards apps world. So, if you like to expose your service data to the browsers and as well as all these modern devices apps in fast and simple way, you should have an API which is compatible with browsers and all these devices. Web API is the great framework for exposing your data and service to different-different devices. Moreover Web API is open source an ideal platform for building REST-ful services



* Client – server
* Stateless
* Caching
* Custom interface
* Layered interface
* Mobility aware

1. Explain the RESTFul Web Service?

RESTful Web Services are basically REST Architecture based Web Services.  RESTful web services are light weight, highly scalable and maintainable and are very commonly used to create APIs for web-based applications. Web services based on REST Architecture are known as RESTful web services. These webservices uses HTTP methods to implement the concept of REST architecture. A RESTful web service usually defines a URI, Uniform Resource Identifier a service, provides resource representation such as JSON and set of HTTP Methods.  Restful Web Service, expose API from your application in a secure, uniform, stateless manner to the calling client.

It support all http methods such as GET , POST , PUT , PATCH , DELETE to perform operation.It enables web applications that are built on various programming languages to communicate with each otherWith the help of Restful services, these web applications can reside on different environments, some could be on Windows, and others could be on Linux.  no matter what the environment is, the end result should always be the same that they should be able to talk to each other. Restful web services offer this flexibility to applications built on various programming languages and platforms to talk to each other.

1. Explain what is a “Resource” in REST?

A resource in REST is a similar Object in Object Oriented Programming or is like an Entity in a Database. Once a resource is identified then its representation is to be decided using a standard format so that the server can send the resource in the above said format and client can understand the same format.These resources can be text , html ,json, xml , csv, image , video . Thus rest does not put any restrictions on the type of resource need to be used , it supports any format unless the client or the server is familier.

A good resource representation should be :

**Understandability:** Both the Server and the Client should be able to understand and utilize the representation format of the resource

**Completeness:**  Format should be able to represent a resource completely. For example, a resource can contain another resource. Format should be able to represent simple as well as complex structures of resources

**Linkablity:**  A resource can have a linkage to another resource, a format should be able to handle such situations.

1. Which protocol is used by RESTful web services?

REST use HTTP (Hypertext transfer protocol) for communication between client and server.The REST architectural style is designed to use a stateless communication. protocol, typically HTTP. In the REST architecture style, clients and servers exchange representations of resources by using a standardized interface and protocol.

1. 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.

1. State the core components of an HTTP Request?
   1. HTTP Version – Indicates version
   2. Request Body – Represents message content
   3. URI – Identifies the resource on the server
   4. Verb – Indicates HTTP methods such as GET, POST, and PUT
2. State the core components of an HTTP response?
   1. HTTP Version – Indicates the present version of HTTP
   2. Response Body – Represents the response message content
   3. Response Header – Consists of metadata, like content length and server length, for the HTTP response message.
   4. Status/Response Code – Indicates the server status for the requested resource
3. What do you understand about payload in RESTFul web service?

Request data which is present in the body part of every HTTP message is referred to as 'Payload'. In Restful web service, the payload can only be passed to the recipient through the POST method.

1. Explain the caching mechanism?

Caching is a technique to speed up data lookups (data reading). Instead of reading the data directly from it source, which could be a database or another remote system, the data is read directly from a cache on the computer that needs the data.

By caching we will be able to store 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.

1. List the main differences between SOAP and REST?

|  |  |
| --- | --- |
| SOAP | REST |
| * SOAP stands for Simple Object Access Protocol * SOAP is a protocol. SOAP was designed with a specification. It includes a WSDL file which has the required information on what the web service does in addition to the location of the web service. * SOAP cannot make use of REST since SOAP is a protocol and REST is an architectural pattern. * SOAP can only work with XML format | * REST stands for Representational State Transfer * REST is an Architectural style in which a web service can only be treated as a RESTful service if it follows the constraints of being * Client Server,Stateless,Cacheable,Layered System,Uniform Interface * REST can make use of SOAP as the underlying protocol for web services, because in the end it is just an architectural pattern. * REST permits different data format such as Plain text, HTML, XML, JSON, etc. But the most preferred format for transferring data is JSON |

1. Enlist advantages and disadvantages of ‘Statelessness’.

**Advantages**

* 1. web services can treat each method request independently.
  2. Web services need not maintain the client's previous interactions. It simplifies the application design.
  3. As HTTP is itself a statelessness protocol, RESTful Web Services work seamlessly with the HTTP protocols.

**Disadvantages**

* 1. 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.