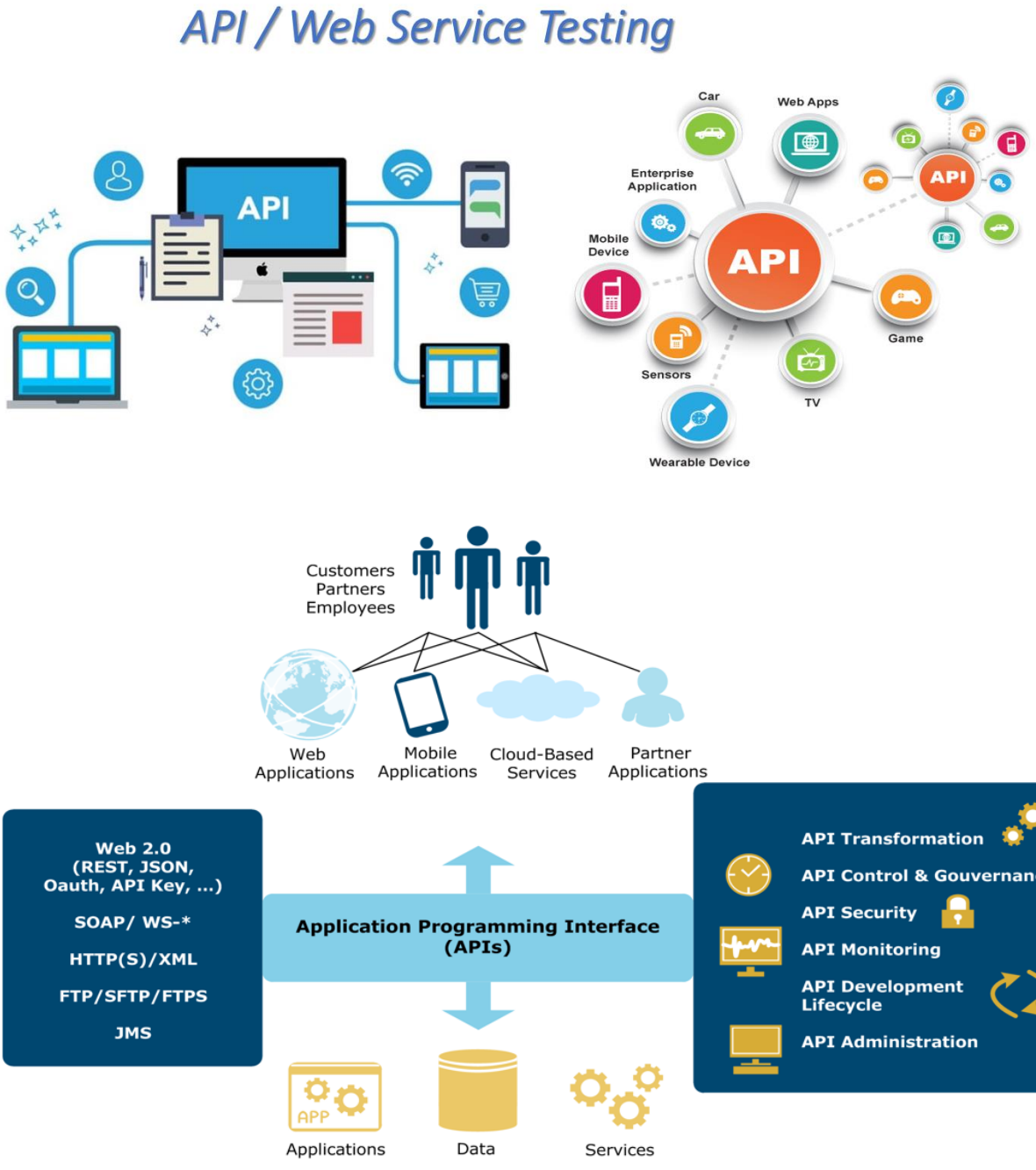
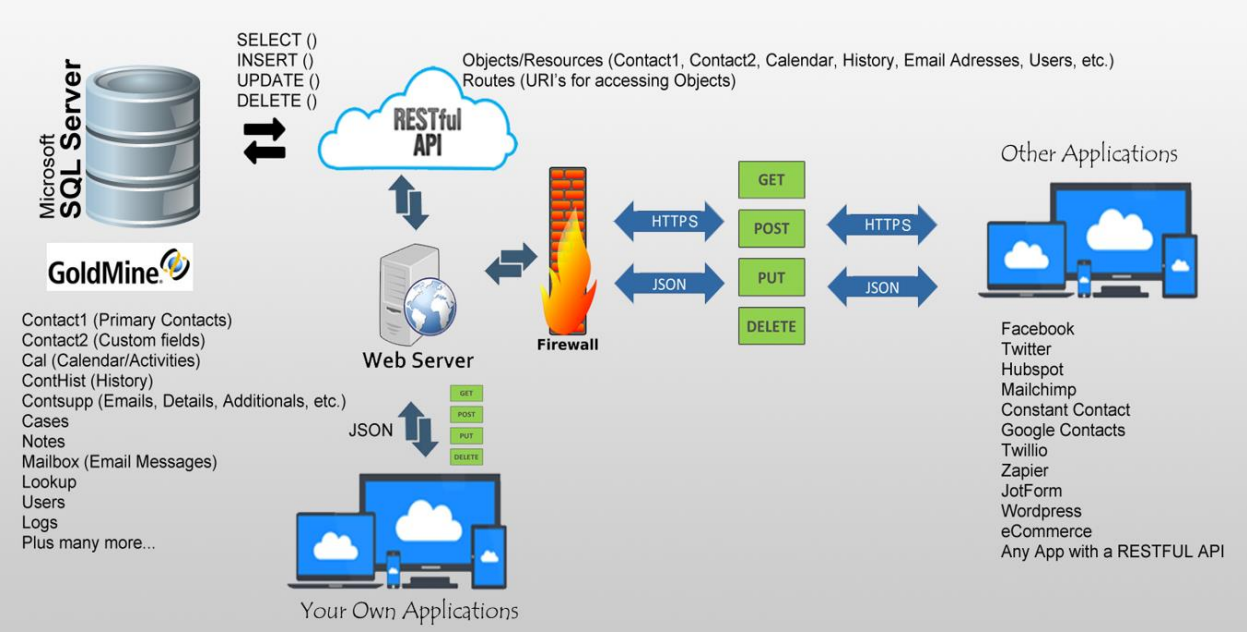
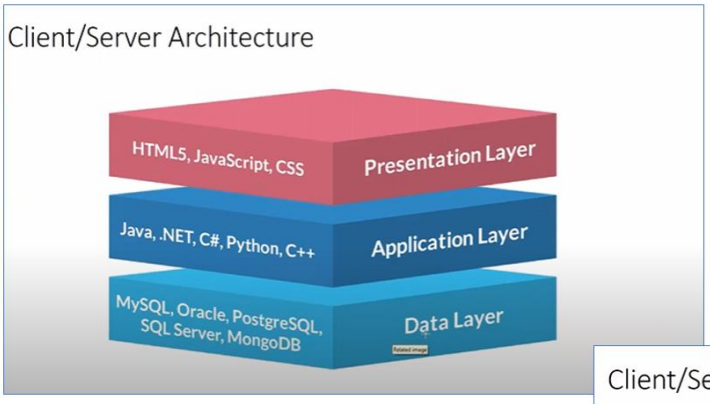
❖ **Web Service:**

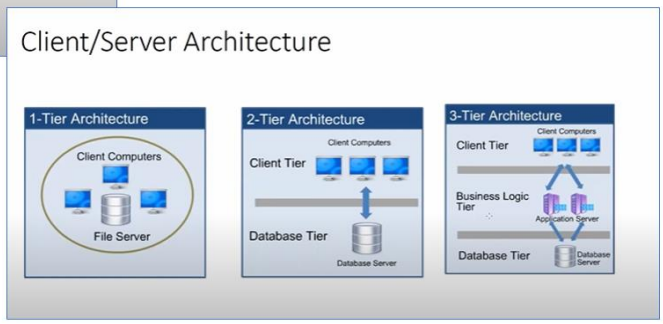
- Service available over the web.

- **Project Application:**
  - Front End(UI)
  - Backend(Database)
  - Service(API)





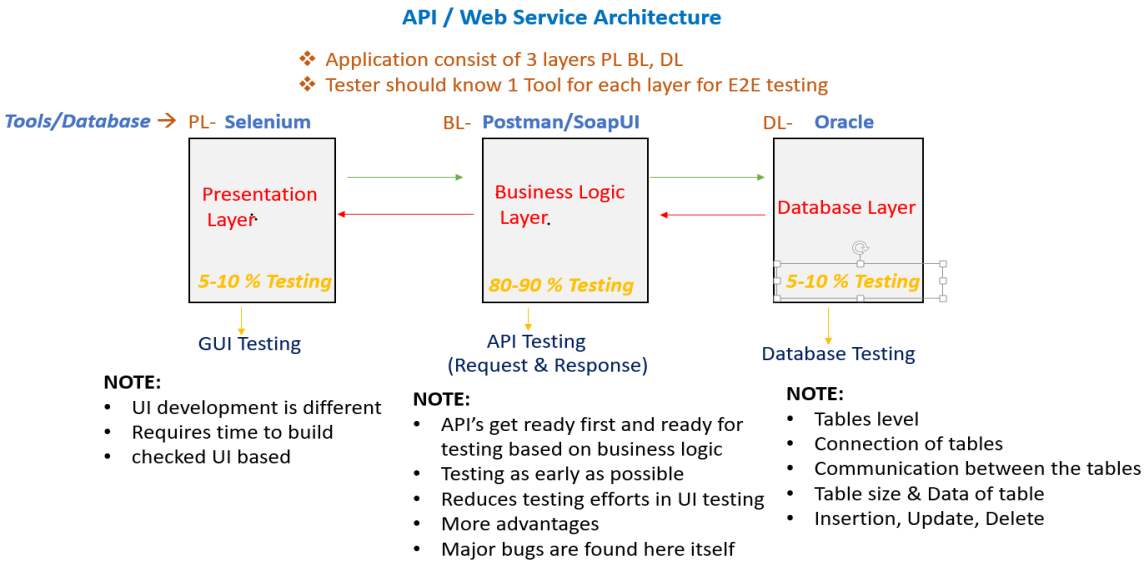




What is client and Server?

Client: A client is computer hardware device or software that accesses a service made available by a server. The server is often (but not always) located on a separate physical computer.

Server: A server is a physical computer dedicated to run services to serve the needs of other computers. Depending on the service that is running, it could be a file server, database server, home media server, print server, or web server.



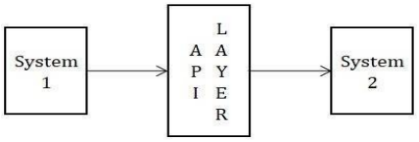
❖ **What is web service?**

- Service available over web.
- Enables Communication between applications over the web.
- Platform independent Communication
  - Kitchen (Chef)(Service
  - Provider)
  - You (Service
  - Consumer)
  - Waiter (API)
- Using Web Service two different applications can talk to each other & exchange data/information.
- An API is the acronym for Application Programming Interface, which is a software Intermediary that allows two applications to talk to each other.
- Each time you use an APP Like Facebook, send an instant message or check the weather on your phone, you are using an API.

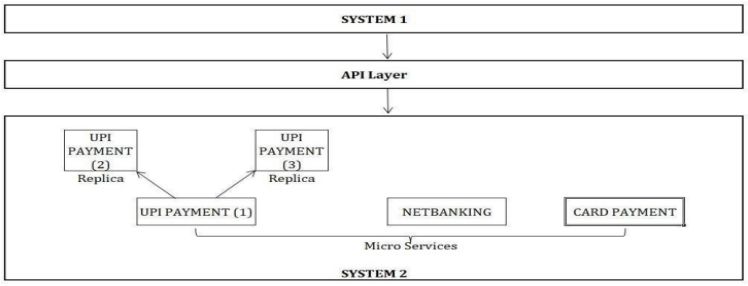
❖ **API- Application Programming Interface.**

- API is use to communicate between two systems.
- It is simply know as sending the request from one system to another system and getting the required response.
- For Ex. Communication between IRCTC and OTP. Here we send request from IRCTC App to get the response as OTP.

- **Advantages of API:**
  - **API provides the security.**
    - While communicating between two systems, API will provide API layer in between them which helps to secure out data.



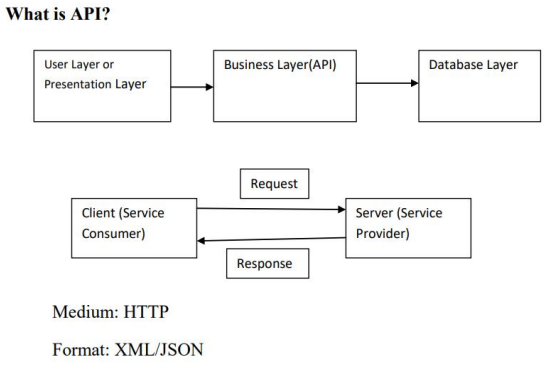
  - **To avoid the data breaching.**
    - As we know API provides the API layer for security purpose, so it also avoids Data hacking or breaching.
  - **To increase the performance and Balances the load.**



    - In API, Developer already created a Replica's of Micro cervices which helps to increase the performance of the system by sharing the request with replicas in case of load on system 2 increases. Here API will decide how much load to be shared with each micro service and replica. As this balances the load, the performance will get automatically increases.
  - **API helps in Data hiding.**
    - API helps to hide the data also.
  - **API helps for Proper communication between two systems.**
    - When first system is sending the request to second system, then the request should go for second system only not the third one. For ex. User wants to do a payment for Amazon order by using Gpay app. Then the payment request should go for Gpay only. This is nothing but the proper communication. In between this communication process, API will provide API layer for Security purpose.
  - **API also checks and authenticates the data which we are passing.**
  - **API tests core functionality.**
  - **API is time effective.**
    - We can hit lot of API's at a time with less time.
  - **Language Independent.**

- API is Language independent i. e API reads multiple languages like XML, JSON, HTML, TEXT etc.

- **Easy interaction with GUI.**



❖ **Difference between API and Web Service?**

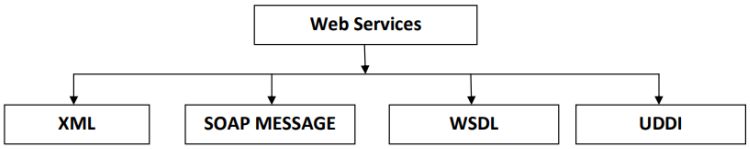
Web Service	**API**
All web services are API.	All APIs are not web service.
Web service might not perform all operations.	API would perform all operations.
Web service needs network for its operation.	API doesn't need a network for its operation.
It supports XML.	Responses are formatted using Web API's MediaTypeFormatter into XML, JSON, or any other given format
It can be used by any client who understands XML.	It can be used by a client who understands JSON or XML.
It provides supports only for the HTTP protocol.	It provides support for the HTTP/s protocol: URL Request/Response Headers, etc.

Two Types of Web Services:

- SOAP Service

- REST Service

❖ **Architecture of Web Services:**



- **XML: (Extensible/Extreme Markup Language)**
  - It is machine readable code which is software and hardware independent.
  - It is used to perform the communication between different technology and different platforms.
  - XML can hold data like.. (HTML, HEADER,BODY, DIV, FAULT).

- **SOAP MESSAGE:**
  - **SOAP stands for simple object access protocol.**
  - SOAP is XML based protocol for exchanging information between the web services
  - SOAP is a communication Protocol
  - SOAP Provides data transport for Web Services.
  - All the information/message exchange happens over a common format: XML
  - SOAP has defined Structure/Format: SOAP Message
  - SOAP Message consists of: **Envelope, Header, Body and Fault.**

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:tem="http://tempuri.org/">
  <soapenv:Header/>
  <soapenv:Body>
    <tem:Add>
      <tem:intA>?</tem:intA>
      <tem:intB>?</tem:intB>
    </tem:Add>
  </soapenv:Body>
</soapenv: Envelope>
```

  - **Envelope: (Mandatory)**

    - Every SOAP message has a root envelope element
    - Envelope is mandatory part of SOAP Message
    - The SOAP Envelope indicates the start and the end of the message so that the receiver knows when entire message has been received.
    - The SOAP envelope solves the problem of knowing when you are done receiving a message and are ready to process it.

  - **Header: (Optional)**

    - Contains any optional attributes of the message used in processing the message.

  - **Body: (Mandatory)**

    - Contains the XML data comprising the message being sent.

- The SOAP Body is a mandatory element which contains application defined XML data being exchanged in the SOAP Message.
- The body must be contained within the envelope and must follow any headers that might be defined for the message.

- **Fault: (Optional)**
  - An optional fault element that provides information about errors that occurred while processing the message.
  - When an error occurs during processing the response to a SOAP Message is called a SOAP fault element in the body of the message, and the fault is returned to sender of the SOAP message.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Client</faultcode>
      <faultstring>System.Web.Services.Protocols.SoapException: Server was unable
to read request. ---> System.InvalidOperationException: There is an error in XML
document (6, 36). ---> System.FormatException: Input string was not in a correct
format.
  at System.Number.StringToNumber(String str, NumberStyles options,
NumberBuffer&amp;amp; number, NumberFormatInfo info, Boolean parseDecimal)
  at System.Number.ParseInt32(String s, NumberStyles style, NumberFormatInfo
info)
  at System.Xml.XmlConvert.ToInt32(String s)
  at
Microsoft.Xml.Serialization.GeneratedAssembly.XmlSerializationReader1.Read1_Ad
d()
  at
Microsoft.Xml.Serialization.GeneratedAssembly.ArrayOfObjectSerializer.Deserialize(
XmlSerializationReader reader)
  at System.Xml.Serialization.XmlSerializer.Deserialize(XmlReader xmlReader, String
encodingStyle, XmlDeserializationEvents events)
  --- End of inner exception stack trace ---
  at System.Xml.Serialization.XmlSerializer.Deserialize(XmlReader xmlReader, String
encodingStyle, XmlDeserializationEvents events)
```

at System.Xml.Serialization.XmlSerializer.Deserialize(XmlReader xmlReader, String encodingStyle)
at System.Web.Services.Protocols.SoapServerProtocol.ReadParameters()
--- End of inner exception stack trace ---
at System.Web.Services.Protocols.SoapServerProtocol.ReadParameters()
at System.Web.Services.Protocols.WebServiceHandler.CoreProcessRequest()</faultstring>
<detail/>
</soap:Fault>
</soap:Body>
</soap:Envelope>

- **WSDL: Web Service description Language**
  - WSDL Stands for Web Service description Language.
  - WSDL is standard format for describing a web Service.
  - WSDL is pronounced as ‗Wiz-dull' and spelled as ‗W-S-D-L'.
  - WSDL definition describes how to access a web service and what operations it will Perform.
  - WSDL is often used in combination with SOAP and XML schema to provide web services over the Internet.
  - A client program connecting to web service can read the WSDL to determine what functions are available on the server.
  - WSDL is an XML document with a at the root and the child elements,
    - <types>,
    - <message>,
    - <portType>,
    - <binding>.
  - **<definition>:**
    - Element must be the root element of all WSDL documents.
    - It defines the name of the web service.
    - The definitions element is container for all the other elements.
  - **<types>:**

    - WSDL <types> element take care of defining the data types that are used by the web service.

    - WSDL allows the types to be defined in separate elements so that the types are reusable with multiple web services.

```
<wsdl:types>
<s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
```

```
<s:element name="Add">
<s:complexType>
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="intA" type="s:int"/
>
<s:element minOccurs="1" maxOccurs="1" name="intB" type="s:int"/
>
</s:sequence>
</s:complexType>
</s:element>
```

- **<message>:**

  - The <message > element describes the data being exchanged between the web service providers and consumers.

  - Each Web Service has two Messages: Input and Output.

```
<wsdl:message name="AddSoapIn">
<wsdl:part name="parameters" element="tns:Add"/>
</wsdl:message>
<wsdl:message name="AddSoapOut">
<wsdl:part name="parameters" element="tns:AddResponse"/>
</wsdl:message>
```

- **<portType>:**

  - <portType> can combine one request and one response message into a single request/response operation. This is most commonly used in SOAP Services.

  - A PortType can define multiple operations.

```
<wsdl:portType name="CalculatorSoap">
<wsdl:operation name="Add">
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap
.org/wsdl/">Adds two integers. This is a test
WebService. ©DNE Online</wsdl:documentation>
<wsdl:input message="tns:AddSoapIn"/>
<wsdl:output message="tns:AddSoapOut"/>
</wsdl:operation>
<wsdl:operation name="Subtract">
<wsdl:input message="tns:SubtractSoapIn"/>
```

```
<wsdl:output message="tns:SubtractSoapOut"/>
</wsdl:operation>
<wsdl:operation name="Multiply">
<wsdl:input message="tns:MultiplySoapIn"/>
<wsdl:output message="tns:MultiplySoapOut"/>
</wsdl:operation>
<wsdl:operation name="Divide">
<wsdl:input message="tns:DivideSoapIn"/>
<wsdl:output message="tns:DivideSoapOut"/>
</wsdl:operation>
</wsdl: portType>
```

- **<binding>:**

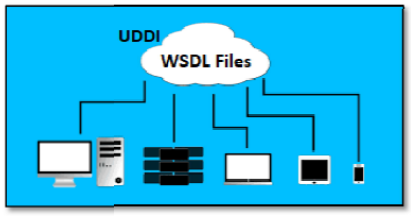
- The <binding> element provides specific details on how a portType operation will actually be transmitted over the web.

- The bindings can be made available via multiple transports including HTTP GET,HTTP POST or SOAP.

```
<wsdl:binding name="CalculatorSoap12" type="tns:CalculatorSoap">
<soap12:binding transport="http://schemas.xmlsoap.org/soap/http"
/>
<wsdl:operation name="Add">
<soap12:operation soapAction="http://tempuri.org/Add" style="doc
ument"/>
<wsdl:input>
<soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="Subtract">
<soap12:operation soapAction="http://tempuri.org/Subtract" style
="document"/>
<wsdl:input>
<soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
```

WSDL File/Service: <definitions>, <types>, <message>, <port Type>, <binding>

- **UDDI: Universal Description Discovery and Integration**
  - Universal description discovery integration is an XML based standard for publishing and finding the web service.
  - A web Service Providers publishes his web service (through WSDL) on an online directory from where consumers can query & search the web services
  - This is online registry/directory is called UDDI.
  - It is a global repository or online directory where all the web services are stored and any organization can search the service the under the UDDI.
  - Also we can say that all the WSDL files are stored under the UDDI.



  - **2 Types of Web Services:**
    - SOAP Service (SOAPUI Tool)
    - REST Service (Soapui and Postman)

    - **SOAP Service :(SOAPUI Tool):**
      - Developer will provide WSDL file http://www.dneonline.com/calculator.asmx?WSDL
      - Unit Testing Document
      - Time taken for the response=5Secs
      - **Test Cases:**
        - Validating SOAP response = Pass
        - Validating DATA and count of data in response = Pass
        - Validating Tagname/Attributes Present in the response=EC=AddResult=>PassValidating Status code in response.
        - Validating different status codes=EC-200-OK=Pass .
        - Validating time taken for the response=EC=5Secs=Pass
        - Applying Assertions for Verification.
        - **Validating the SOAP Service by passing Test data=Re-Testing**
        - Verify the SOAP Service by passing one digit number EC=one digits numbers should be added=Pass

- Verify the SOAP Service by passing two digit number EC=Two digits numbers should be added=Pass
- Verify the SOAP Service by passing Four/Five digit number EC=Four/Five digits numbers should be added=Pass
- Verify the SOAP Service by passing one digit number at 1st place and two digit number at 2nd place one digit number at 1st place and two digit number at 2nd place should be added=Pass
- Verify the SOAP Service by Passing zero at both place EC =Zero number should be added=Pass
- **Validating Negative Test cases**
- Verify the SOAP Service by passing decimal digit numbers EC=Decimal digits numbers should not be added=Pass
- Verify the SOAP Service by passing String/Fractional digit number EC=Number should not be added=Pass
- Verify the SOAP Service by passing Null/Blank values EC=Should not be added=Pass
- Verify the SOAP service by Passing wrong WSDL file EC=Pass
- **Xpath Assertion:**
- Xpath is a query language for selecting Nodes(Tagnames) from an XML document.
- Xpath can also use to compute values from the content of an xml document.
- To find something is present in the response.
- https://ws.footballpool.dataaccess.eu/info.wso?WSDL

- **REST Service (Soapui and Postman):**
  - Developer will provide URL https://reqres.in/
  - Unit Testing Document
  - Time taken for the response=5Secs

  - **Test Cases:**

    - Validating REST Response=Pass
    - Validating Data and Count of Data in the REST Response=Pass.
    - Validating tagnames/attributes present in the response EC=Id, Email, first_name, last-named, Avatar=Pass
    - Validating different Status Codes EC=200-OK=Pass.
    - Validating time taken for the response EC=5secs=Pass.
    - Applying assertion for verification.

- **Validating the functionality by passing Test data (Re-Testing).**
- Verify by Passing Parameter as 1 in the GET Method EC=Status Code-200-Ok & GET Method will show response=Pass.
- Verify by Passing Parameter as 2 in the GET Method EC=Status Code-200-Ok & GET Method will show response=Pass.
- Verify by Passing Parameter as 3 in the GET Method EC=Status Code-200-Ok & GET Method will show response=Pass.
- Verify by Passing Parameter as 10/100/1000 in the GET Method EC=Status Code as 200-Ok & GET Method will show response=Pass.
- **Validating negative test cases!**
- Verify by passing decimal values in GET Method EC=Status Code-400-Bad Request& GET Method will not show the response=Fail-Defect.
- Verify by passing parameter as Null/Blank value in GET Method EC=Status Code as 400-Bad Request& GET Method will not show the response=Fail-Defect.
- Verify by passing parameter as Character/String value in GET Method EC=Status Code-400-Bad Request& GET Method will not show the response=Fail-Defect.
- Verify by Passing wrong URL/URI in GET Method EC=Status Code-404- Not Found & GET Method will not show the response=Pass.
- Verify by passing wrong Authorization/bearer Token/API Key/Basic Auth(Username& Password) in GET Method EC=Status Code-401-UnAuthorization & GET Method will not show the response.
- Verify by changing GET Method into another method EC=Status Code-405-Method Not Allowed & GET Method will not show the response=Pass

❖ **Classification of Web Services:**

SOAP Service	**REST Service**
SOAP stands for Simple Object Access Protocol.	REST stands for Representational State Transfer.

SOAP services used for web-based application only.	Rest services used for web based, Mobile based, Desktop based application.
SOAP services are largely based on XML.	REST services use multiple standards like HTTP, JSON, URL, and XML
SOAP uses XML for request & response.	Rest uses HTTP/URL for request & response like HTML, Text, JSON, XML etc.
For testing SOPA service we require WSDL file.	For testing REST service, we require URL/ URI.
SOAP (designed) is a protocol.	REST (designed) is an architectural style.
Performance wise Requires more bandwidth.	Requires fewer bandwidth.
SOAP supports SSL security.	REST supports SSL & HTTPS security.
It is taking more time to respond.	It is faster for response.

- **Type of Request/ Methods:**
  - **GET Method-** GET is used to get data from a resource –similar select to statement
  - **POST Method-** POST is used to send data to a server to create a resource – Similar Insert to statement
  - **Patch Method-** requests are to make partial update on a resource - Similar update to statement
  - **PUT Method-** PUT is used to send data to a server to update a resource - Similar update to statement
  - **DELETE Method-** The DELETE method deletes the specified resource - - Similar delete to statement

- **What is CRUD?**

  - Most REST APIs implement CRUD: **Create, Retrieve, Update, and Delete.**
  - We can map these operations into CRUD.
    - POST—Create
    - GET—Retrieve
    - PUT / PATCH—Update
    - DELETE—Delete

- **Different status code:**
  - Successful responses (2XX)
  - Server errors (4XX)
  - Client errors (5XX)
  - **Theses status code will define by developer.**

- **For Successful Response(2XX):**
  - **200-OK: When we get Successful data.** The request has succeeded. The meaning of the success depends on the HTTP method **– ex. GET**
  - **201- Created:** when we get Successful data. The request has succeeded and a new resource has been created as a result. This is typically the response sent after **POST requests.**
  - **202- Accepted:** If data will sent to server for storing purpose- ex**. PUT / PATCH.**
  - **204- No Content:** If unique data sent to server then it sent no content. **Ex. POST, PUT, PATCH.**
  - **Server errors (4XX)- Server 2**
  - **400-Bad Request: when URL wrong or end point missing**. The server could not understand the request due to invalid syntax/Data.
  - **401- Unauthorised: when session got expired, passing invalid token/ username/pass.** If invalid authorization will be provided in request.
  - **403- Forbidden:** The client does not have access rights to the content. It will go to some intermediate state.
  - **404-Page not found: when we are trying to access the URL but URL not present.** The server cannot find the requested resource. In the browser, this means the URL is not recognized.
  - **405-Method not allowed:** The request method is known by the server but has been disabled and cannot be used.
  - **internal errors (5XX)- Server 1**
  - **500- Internal server Error: when any server down or network issue.** The server has encountered a situation it doesn't know how to handle.
  - **503-Service not available:** The server is not ready to handle the request. Common causes are a server that is down for maintenance or that is overloaded.
  - **503-Method not Implemented:** The request method is not supported by the server and cannot be handled.

- **What Is End Point In API?**

  - Endpoint is one end of a communication channel. When an API interacts with another system, the touch points of this communication are considered endpoints For APIs, an endpoint can include a URL of a server or service.
  - Address where API is hosted on the server.
  - HTTP Methods which are commonly used to communicate with REST API‘s are: GET, POST, PUT, PATCH, DELETE.

- **GET:** The GET is used to extract information from the given server using a given URI. While using GET request, it should only extract data and should have no other effect on data. No Payload/Body required.
- **End Point Request URL can be constructed as below:**
- URI=(Base URL/End Point)/Resource/(Query/Path)Parameters
- URI= URL+Resourse+Parameter Value

- **What Is Payload In API?**
  - A payload in API is the actual data that is sent with the CURD method in HTTP. It is the important information that we submit to the server when we are making an API request.
  - The payload can be sent or received in various formats, including JSON, XML, and TEXT etc.

- **Resources:**
  - Resources represents API/Collection which can be accessed from the servers
  - Google.com/maps
  - Google.com/images
  - Google.com/search

- **Path Parameters:** are variable parts of the URL Path.They are typically used to point to a specific resource within a collection, such as user id identified by ID
  - https://www.google.com/images/1123343
  - https://www.google.com/docs/1123343
  - https://www.amazon.com/orders/112

- **Query Parameters:**
  - Query Parameter is used to sort/filter the resources.
  - Query Parameter is identified with —"?"
  - https://www.amazon.com/orders?sort_by=2/20/2022

- **End Point Request URL can be constructed as below:**
  - URI=(Base URL/End Point)/Resource/(Query/Path)Parameters
  - URI= URL+Resourse+Parameter Value

- **URL & URI**
  - **URL (uniform resource locator):** A URL is nothing but the address of a given unique resource on the Web. With the security protocol.

- **URN (Uniform Resource Name):** A URN is nothing but the specific name by which we can find a recourse on the Web.
- URI**(Uniform Resource Identifier):** It is combination of the URL and URN.
- EX: https://reqres.in/api/users?page=2
- Where
- Base URL/Domain/Endpoint – https://reqres.in
- URI- https://reqres.in/api/users?page=2
- URI contains: URL & other part
- **contains**
- 1. parameters/Resource = /api/users
- 2. Parameter values/ Query String = after? mark ex. page=2
- URI= URL +Resource +Parameter value.

- **POST Method/Request:**
  - POST Method Service-URL/URI=https://reqres.in/api/register
  - Body Payload:

```
{
 "email": "eve.holt@reqres.in",
 "password": "pistol"
}
```

  - Unit Testing Document
  - Time Taken for the Response=5Sec
  - Username & Password(Basic Auth)/API Key/Bearer Token-dalablfblf.

➢ **Test Cases:**

- Validating REST Response=Pass.
- Validating Data and Count of data in the Response=Pass.
- Validating Tagname/Attributes Present in the Response=EC=>Id, Token=Pass.
- Validating Different Status Codes=>EC=Status Code-201-Created=Fail-Defect.
- Validating Time Taken for the response=5Sec=Pass.
- Applying Assertion for Verifications.
- **Validating Functionality by passing Test data (Re-Testing).**
- Verify the POST Method/Request by Passing george.bluth@reqres.in with Password(“abcdef”)=> then EC=For this email ID and Password (“pistol”) will be inserted in the server.=Pass
- Verify the POST Method/Request by Passing janet.weaver@reqres.in with Password then EC=For this email ID and Password will be inserted in the server.=Pass

- Verify the POST Method/Request by Passing charles.morris@reqres.in with Password (“abcdef”) =>EC=For this email ID and Password will be inserted in the server.=Pass
- Verify the POST Method/Request by Passing lindsay.ferguson@reqres.in with Password=>EC=For this email ID and Password will be inserted in the server.=Pass
- Verify the POST Method/Request by Passing rachel.howell@reqres.in with Password (“abcdef”) =>EC=For this email ID and Password will be inserted in the server =Pass
- **Validating Negative Test cases!**
- Verify POST Method/Request by passing invalid email id vctc@reqres.in=>EC=400-Bad Request=Pass.
- Verify the POST Method by Passing Null/Blank email id=>EC=400-Bad Request=Pass.
- Verify the POST Method/Request by Passing only emailID rachel.howell@reqres.in Without Password=>EC=400-Bad Request=Pass.
- Verify POST Method /Request by Passing invalid email domain rachel.howell@gmail.com =EC=400- Bad Request=Pass.
- Verify POST Method/Request by Passing valid id (id=1) with Password=EC=400-Bad Request=Pass.
- Verify POST Method Request/Request by Passing wrong URL/URI=>EC=404-Not Found=Pass.
- Verify by changing POST Method/Request into Another Method/Request=EC=405-Method Not Allowed=Pass.

❖ **PUT Method/ Request-**

- PUT method Service –URL/ URI= https://reqres.in/api/users/2
- Body/ Payload-

```
{
 "name": "morpheus",
 "job": "zion resident"
}
```

- Username & Password/ API Key/ Barrier token - tdcgbshdnjhgfvbdhcnd
- Time take to response = 60 sec
- Unit testing documents.

❖ **Delete method/ request-**

- GET method Service –URL/ URI= https://reqres.in/api/users/2
- Username & Password/ API Key/ Barrier token - tdcgbshdnjhgfvbdhcnd
- Time take to response = 60 sec
- Unit Testing documents-

➢ **Agenda: JSON Path Count, JSON Path Existence, JSON Path Match, Postman**

- GET METHOD: https://reqres.in/api/users/2
- Xpath (XML document-SOAP Service) similarly JSON Path (JSON document -REST Service)
- JSON Path Count:
- https://jsonpathfinder.com/

➢ **What Is Authentication?**

- Authentication is the act of validating that users are whom they claim to be
- This is the first step in any security process.
- **Complete an authentication process with:**
- **Passwords.** Usernames and passwords are the most common authentication factors. If a user enters the correct data, the system assumes the identity is valid and grants access.
- **One-time pins: Grant** access for only one session or transaction.
- **Authentication apps:** Generate security codes via an outside party that grants access.
- **Biometrics:** A user presents a fingerprint or eye scan to gain access to the system.

➢ **What Is Authorization?**

- Authorization in system security is the process of giving the user permission to access a specific resource or function.
- Giving someone permission to download a particular file on a server or providing individual users with administrative access to an application are good examples of authorization.
- In secure environments, authorization must always follow authentication. Users should first prove that their identities are genuine before an organizations administrators grant them access to the requested resources.

- **DIFF. TYPES OF AUTHORIZATIONS-**
  - Basic Auth
  - Digest
  - Token
  - OAuth1
  - OAuth2
  - No auth
  - AWS signature
  - **Basic Auth**
    - In Basic we have to pass only Username and Password.
  - **Digest Auth**
    - In Digest auth we have to pass Only Username and Password same as that of Basic Auth but Digest Auth is more secure than Basic auth.
  - **OAuth 1.0**
    - In OAuth 1.0 we have to pass have to pass Consumer Key, Consumers secrete, Access Token, Token Secrete. All this details will be provided by developer side.
  - **OAuth 2.0**
    - In OAuth 2.0 we have to pass have to pass Grant type, Client Id, Client Secrete. All this details will be provided by developer side.
  - **Token**
    - Here we have to just put the value of token. Token value may be the combination of integer and character values. While entering the token value, we need to mention Bearer keyword. Bearer means the identification for that token. Token is mostly used type of authorizations.

Questions on API testing:

- **What are the different methods present in API?**
  - There are different methods present in API: **GET, POST, PUT, DELETE, PATCH etc.**
- **What are different operations performed in API?**
  - **Below are the operations performed in API:**
  - GET- used to fetch the data
  - POST- Used to create data
  - PUT- Used to update data
  - DELETE- used to delete data
- **What is difference between PUT and PATCH?**
  - PUT- we can update all the fields as well as single field
  - PATCH- we can update single/ partial fields

- **What are main differences between API and Web Service?**
  - Api call internally and webservices call over the internet
  - The only difference is that a Web service facilitates interaction between two machines over network.
  - An API acts as an interface between two different applications so that they can communicate with each other. Web service also uses SOAP, REST, and XML- RPC as a means of communication.

- **What are the advantages of API Testing?**
  - Api provides the security.
  - API checks the authentication and the data that we are passing.
  - Can transfer the load to diff micro services.
  - API helps to avoid data breaching.
  - Test for Core Functionality.
  - Time Effective- we can hit lots of APIs within less time.
  - Language-Independent- like Json, XML, html, text.
  - Easy Integration with GUI.

- **What is different Test Environment in project?**
  - Generally we will have below four test Environments:
  - DEV- where developers works.
  - SIT/QA- where Testers works.
  - UAT- where Testers and Client works.
  - PROD- It's a live environment.

- **What are the test environments of API?**
  - Global- Global has large scope (used to pass variables between diff collections)
  - Local – Local has small scope (Used to pass variable from one request to another)
  - We are using QA/UAT environment in which we are using Global and Local environment for API methods.

- **What must be checked when performing API testing?**
  - Error codes, data which are coming (Retrieval data), Time.

- **What are differences between API Testing and UI Testing?**
  - API doesn't provide the GUI (Graphical User interface) but UI provides.

- **Where we pass the data in post?**
  - We pass the data in Body-> Raw-> in the form of Json, XML. Html, text

- **What are tools could be used for API testing?**
  - Postman
  - Swagger
  - SoapUI

- **What are common API errors that often founded?**

- These are the common error getting during API testing

  201-created

  200-ok

  400-Bad request

  401-Unauthorised

  403- Forbidden

  404- Page not found

  500- Internal server error

  503-service not available

- **Any examples why error code generates?**

  200- When we get successful data.

  201- When we create data into database.

  400- URL wrong or end point missing.

  401- When session got expired, passing invalid token/ username/pass.

  404- When we are trying to access the URL but URL not present.

  405- Method not allowed.

  500- Any server down or network issue.

- **What are the collections?**
  - Collections are used to store the services (API methods)
  - By using collection we can run all the methods at the same time.
  - We can Import/Export Collection.

- **What is mean bearer token?**
  - Bearer token is one of the Authentication pass in headers
  - Bearer means identification for the token.

- **Can we run collection?**
  - Yes, we can run the collection and collection methods at the same time, but before we run the previous or old collection we have to update the authentication.

- **What is mean by end points/service URL?**
  - End points are the different service URLs which are used to hit the URL with domain URL.

- **What is mean API?**
  - Application programming interface
  - API stands for Application programming interface.
  - Used to communicate between two systems.
  - It simply knows as sending the request and getting the response.

- **What are headers?**
  - Headers is nothing but the what kind of request it is

    {content-type= application json/application xml/application text }

- **What is bearer?**
  - Bearer is the identifier for particular token used for the Authentication.

- **Types of API**
  - REST API- Uses Postman tool (Representational state transfer)
  - SOAP API- Uses SOAPUI tool (simple object access protocol)

- **Concept under REST:**
  - REST- REST is an architecture used to create rest API.
  - REST Assured- To automate rest API we need rest assured libraries.
  - RESTFUL- when we automate rest API it called as restful services.

- **What is the difference between '/' and '?'**
  - /- Path parameter
  - ?- Query parameter

- **What is producer and consumer?**
  - Producer- who produce the data.
  - Consumer- who consumes the data.

- **What is URI?**
  - URI- Unique resource identifier
  - URI= URL+ENDPOINT
  - Eg. https://www.amazon.com+/login/home

- **What are diff ways to pass the data/ scripting languages?**

  a) JSON:

  ```
  {
  "name": "Suraj ",
  "email": "Suraj123@gmail.com",
  "gender": "Male",
  "status": "Active"
  }
  ```

  b) XML:

  ```
  <name>suraj</name>
  <email>suraj@gmail.com</email>
  ```

  c) String

  d) Text

  e) Html

  f) Javascript

- **What are headers?**
  - Headers mean what kind of data we are passing.

    a. Authorization

    b. Content Type

    c. Language

- **What we pass in http request?**

    a. URI

    b. Headers

    c. Payload

- **What are different authorizations?**

    a. Basic Auth

    - Pass the username and pass.

    b. Digest

    - Whenever we are passing username and pass it will get convert in # keys.
    - It means your username/pass will secured get server side too.

    c. Oauth1

    - **Oauth1 required below things:**

        1. Consumer Key
        2. Consumer Secret
        3. Access Token
        4. Secret Token

    - Above info will get from developers.

    d. Oauth2

    - **Oauth2 required below things:**

        1. Client Id
        2. Client Secret
        3. Grant type

    - Above info will get from developers.

    f. Bearer Token

    g. NoAuth

- **What are Oauth1 and Oauth2?**
  - Oauth1- this auth uses when we need third party logins.
  - Oauth2- this auth uses when we have single URL and different endpoints.

- **What is WSDL file?**
  - WSDL basically an XML document contains all the details about web service and all API request.

- **What is Web service?**
  - Whenever we are hitting any service over the internet it known as webservice.
  - Webservice is any piece of software that makes it available over the internet and uses a standardized XML messaging system.

- **What is UDDI?**
  - Universal description discovery integration.
  - -UDDI is an XML based standard for describing, publishing and finding the webservices.

- **What are diff soap elements/components?**

  a. Envelop – It is beginning and end of message.

  b. Header – Header elements contain header information.

  c. Body – body element contains call and response information.

  d. Fault – Fault contain error and status information.

- **What is diff WSDL element/component?**

  a. Type- Define the data types used by the webservices.

  b. Message – Define the data element for each operation.

  c. Port Type- Describe the operation that can be performed and message involve.

  d. Binding- Defines the protocol and data format for each port type.

- **What is means URI(API)**
  - URI= URL+endpoints(resource)
  - url: www.facebook.com
  - endpoints: /login/home
  - URI- unified resource identifier

- url- unified resource locator.

- **Difference between monolithic and miscroservice?**
  - Monolethic - all api available under one service.
  - Microservice- for api have different microservice.

- **What are different API gatways?**
  a. SSL certificate
  b. Routing
  c. Adapter
  d. Cache
  e. Load balancer
- **What are different assertions present in SOAPUI?**
  - Below are the different assertions present in SOAPUI:
    - **Contains -** checks for the existence of a specified string
    - **Not Contains -** checks for the non-existence of a specified string
    - **Response SLA** - validates that the last received response time was within the defined limit.Applicable to Script TestSteps and TestSteps that send requests and receive responses.
    - **Invalid HTTP Status Codes -** checks that the target TestStep received an HTTP result with a status code not in the list of defined codes. Applicable to any TestStep that receives HTTP messages
    - **Valid HTTP Status Codes** - checks that the target TestStep received an HTTP result with a status code in the list of defined codes. Applicable to any TestStep that receives HTTP messages.Etc.

- **What is API Testing?**
  - API testing is a type of software testing that involves testing APIs directly and also as a part of integration testing to check whether the API meets expectations in terms of functionality, reliability, performance, and security of an application.
  - In API Testing our main focus will be on Business logic layer of the software architecture.
  - API testing can be performed on any software system which contains multiple APIs.

- **Name some of the common protocols used in API Testing?**
  - Some of the protocols using in API Testing are as follows:
  - HTTP
  - REST
  - SOAP
  - JMS

- UDDI

- **What are the advantages of API Testing?**
  - API Testing is time effective when compared to GUI Testing. API test automation requires less code so it can provide faster and better test coverage.
  - API Testing helps us to reduce the testing cost. With API Testing we can find minor bugs before the GUI Testing. These minor bugs will become bigger during the GUI Testing. So finding those bugs in API testing will be cost effective to the company.
  - API Testing is language independent.
  - API Testing is quite helpful in testing Core Functionality. We can test the APIs without a user interface. In GUI Testing, we need to wait until the application is available to test the core functionalities.
  - API Testing helps us to reduce the risks.

- **What exactly needs to be verified in API testing?**
  - Basically, on API Testing, we send a request to the API with the known data and we analyses the response.

    1. Data accuracy
    2. HTTP status codes
    3. Response time
    3. Error codes in case API returns any errors
    4. Authorization checks
    5. Non-functional testing such as performance testing, security testing etc.

- **Name some tools used for API Testing?**
  - Some of the tools used for API Testing are as follows:
  - Postman, Katalon Studio,SoapUI,AssertibleTricentis ,ToscaApigee,JMeter,Rest-Assured ,Karate DSL,API Fortress,Parasoft,HP QTP(UFT),vREST,Airborne,API Science,APIary Inspector,Citrus Framework,Hippie-Swagger,HttpMaster Express,Mockbin,Ping API,Pyresttest,Rest Console,RoboHydra Server,SOAP Sonar,Unirest,WebInject etc.

- **List some most used templates for API documentation?**
  - Some of the API documentation templates are as follows.
    - Swagger
    - FlatDoc
    - RestDoc
    - API blueprint
    - Slate

- Miredot
- Web service API Specification.

- **Name some of the API examples which are quite popular**
  - **Some of the popular API examples are:** Google MapsAPI, YouTube, Twitter, Amazon Advertising API etc.

- **Difference between API testing and Unit Testing?**
  - **UNIT TESTING:**
  - Unit testing is conducted by Development Team.
  - Unit testing is a form of White box testing.
  - Unit testing is conducted prior to the process of including the code in the build
  - Source code is involved in Unit testing.
  - In unit testing, the scope of testing is limited, so only basic functionalities are considered for testing.
  - **API TESTING:**
  - API testing is conducted by QA Team.
  - API testing is a form of Black box testing.
  - API testing is conducted after the build is ready for testing.
  - Source code is not involved in API testing.
  - In API testing, the scope of testing is wide, so all the issues that are functional are considered for testing.

- **What are the main challenges faced in API testing?**
  - **Some of the challenges we face while doing API testing are as follows:**
  - Selecting proper parameters and its combinations.
  - Categorizing the parameters properly.
  - Proper call sequencing is required as this may lead to inadequate coverage in testing.
  - Verifying and validating the output.
  - Due to absence of GUI it is quite difficult to provide input values.

- **What are the types of bugs we face when performing API testing?**
  - Issues observed when performing API testing are:
    - Stress, performance, and security issues
    - Duplicate or missing functionality
    - Reliability issues
    - Improper messaging
    - Incompatible error handling mechanism
    - Multi-threaded issues

- Improper errors

- **How to send input data in GET Method?**
  - Ans: Using Query Parameters
  - POST: A post request is used to send data to the server, for example, Customer information, File, Upload, etc. using HTML forms.

- **How to send input data in POST Method?**
  - Ans: Using Body Payload

- **Which type of Authorization is mostly used in organization? Why?**
  - Token Authorization is mostly used in any organization because if it expires then we can easily regenerate the new token by simple clicking on regenerate the token. Token expiry limit will be decided by Developer. Token also provides more security.

Status Codes