***Application Programming Interface***

1. ***What is API?***

* *API Means Application Programming interface*
* *API is used to communicate between two or more software components.*
* *API is used to exchange data between two software components*
* *Everyday we use application like what’s up , Facebook, phone Pay ,google pay we are using API.*

1. ***What is API testing***

* *API Testing is type of software testing.*
* *Inside the API testing we focus on business logic of the functionality.*
* *API is lack of UI (User interface) means inside API we don’t have UI, we check business logic of the functionality with the help API testing.*
* *API testing is performed on message layer or business logic layer.*

1. ***What is Web Services?***

* *These are the service available over web or network that provide the way to communicate between 2 software or application*
* *Important points related to web service*

1. *Web services work over HTTP means Internet connection compulsory required.*
2. *Web service does not provide the User Interface (UI)*
3. *If an API call over HTTP protocol over network then it become the Web Services*
4. ***Difference between API and Web Service***

|  |  |
| --- | --- |
| ***Web Service*** | ***API*** |
| *it always need network to do operation* | *it may not required network for its operation* |
| *All Web Service are API* | *All API are not Web Services* |
| *Web service can be used SOAP,REST etc communication* | *API Can communicate through CRUD calls* |
| *It is not possible in case of Web Services* | *If an API get call over HTTP protocol over network then it become Web Service* |

1. ***What are the different types of API?***

*there are different types of APIs*

*1) Private API*

*2) Partner API*

*3) Open / Public API*

*4) Composite API*

*5) Internal API*

1. ***Private API***

* *Development team use these API within the team.*
* *These API is created in house by development team and it is highly personalized API according to team needs.*
* *These API cannot be used by external developer or company.*
* *these API specially build for own project or company use.*

1. ***Partner API***

* *We use Partner API to collaborate on Projects.*
* *Partner developer they can modify the API as per the requirements wise.*
* *These API allow or maintains the Organization API flows .*

1. ***Open / Public API***

* *Open API or public API is used to any organization.*
* *it is free to available in market , any company they can use these API.*
* *some Open API have free of cost and some have with charges.*
* *if u want to use Open / Public API then we have register application information in their portal.*

1. ***Composite API***

* *Composite API it allows to application to make the request in different platform (OS and Language)*
* *Composite API is used widely by organization to capture the test data from different platforms.*

1. ***Internal API***

* *Internal API is strictly used in within the organization or company.*
* *if u want to connect to internal software component or functionality then we use Internal API*

1. ***What are the different types of API protocol?***

*-API protocol it is used to exchange the data between 2 or more software components*

*there are different types of API protocol*

1. *REST API*
2. *SOAP API*
3. *GraphQL API*
4. *Webhook API*
5. ***REST API***

* *Representation State Transfer (REST) it is used on web applications.*
* *it is used to exchange the data between 2 or more software components*
* *it uses HTTP or HTTPs request to send or retrieve the data from server.*
* *REST is stateless means they don’t store any session information/ test data between the requests.*
* *REST it supports for different format as like JSON/XML/java script/html etc.*
* *it is more preferred than other API protocol.*
* *REST is architectural style, so easily we can implement SOAP protocol inside the REST.*
* *in REST service it inherits the securities from other Protocol.*

1. ***SOAP API***

* *Simple Object Access Protocol used in complex transactions that require higher level of security.*
* *SOAP uses only XML format.*
* *SOAP is protocol, we cannot implement or inherit securities from other protocol.*
* *SOAP define their own securities*
* *SOAP is slower than the REST.*
* *SOAP is preferred is less compare with REST but in transactions level API we preferred SOAP protocol.*

1. ***GraphQL API***

* *GraphQL API is mostly used in mobile and web-based applications.*
* *in GraphQL we use single endpoints for the multiple purpose.*
* *GraphQL has ability to query multiple resources in single requests*
* *GraphQL, it allows to more precise control over the test data that is retrieved format.*

1. ***Webhook API***

* *Webhook API is used in real time notifications when specific events is occurs.*
* *Webhook API is mostly used in social media applications.*

1. ***Difference between SOAP and REST?***

|  |  |
| --- | --- |
| ***SOAP*** | ***REST*** |
| *SOAP Means simple object Access protocol* | *REST means Representational State Transfer* |
| *SOAP is protocol* | *REST is architectural style* |
| *SOAP Support only (Extensive Markup Language) XML Format* | *REST supports XML or JSON format* |
| *SOAP is slower* | *REST is faster* |
| *SOAP Define own securities* | *REST service inherit the securities* |
| *It is less preferred* | *It is more preferred* |
| *SOAP can not be REST because it is protocol* | *REST can be SOAP Web service because it is concept* |
| *Not much light weight than REST* | *REST is very light weight* |

1. ***What are the different tools used in API?***

* *There are famous Tools in market to test API Manually as well as by using Automation.*

1. *Postman*
2. *Soap UI*
3. *Karate*
4. *HTTP Master*
5. *Para-Soft*
6. *HP UFT*
7. *Rest Assured*
8. *Rest Pro*
9. *PyRest test*
10. *TOSCA*
11. ***What are the different testing types we performed in API?***

*we performed different testing types on API*

1. ***Functional testing***

* *by using Functional testing, we validate the application behaviour as per the requirement documents*
* *by using Functional testing, we validate the functionality of the application.*

1. ***Security testing***

* *by Security testing we test to ensure that data send in secured mode.*
* *by Security testing we validate application access functionality as per the user roles wise.*

1. ***Performance testing***

* *by using Performance testing we validate the speed and accuracy of API response.*

1. ***Integration testing***

* *by using Integration testing we validate API integration flow from one to another system.*

1. ***compatibility testing***

* *by using compatibility testing we validate application with different versions.*

1. ***What are the different types of available to create API documentation***

*there are different tools*

*1) Swagger tool*

*2) Postman tool*

*3) Apiary tool*

*4) RAML tool.*

*1****) Swagger tool***

* *Swagger/openAPI tool used to create API documentation and*
* *it allowed to developer to define the API specification in standard format.*
* *in my current project we use swagger tool to create API documentation*

*2)* ***Postman tool***

* *In Postman tool also we create API documentation*
* *but in my current project we use Postman tool to test the APIs or Web Services manually*

*3)* ***Apiary tool***

* *by using Apiary tool also, we can create API documentations*

*4)* ***RAML tool.***

* *RAML means RestFul API Modelling Language tool.*
* *RAML tool used for creating the API documentations.*
* *RAML tool is used on YAML language.*
* *it enables to developer to define the API structure of request or response parameters in human readable format.*

1. ***What is API Documentation?***

*there different important things present in API documentation*

*1) Endpoints*

*2) Request Parameter*

*3) Request and Response format*

*4) Authentication and Authorization*

*5) Response code/ status Code and Error Handling code*

*6) Usages Examples or code Snippets*

*7) Interactive documentations*

*8) Versioning and changelogs*

1. ***Endpoints***

* in API documentation we have different endpoints available to interact with API.
* Each endpoints are used for different purpose as like retrieve entity, create a entity, update the entity and delete the entity.
* To perform this operation we use different types of request as like GET Request, POST Request, PUT/PATCH Request or DELETE request.
* in API documentation we mention endpoints along with request type.

***2) Parameter***

* in API documentation they clearly mention parameter as like path parameter, query parameter, request body parameter, header parameter, cookies parameter, form multipart parameter along with key-value pair and its datatype information
* API documentation also provide more information regarding the parameters as like which parameter compulsory or optional.

***3) Request and Response format***

* in API documentation, they clearly mention we have to send request body in JSON/XML format to the server and after getting the response, server should return response body in JSON/XML format.
* Which type of request body format is accepted by endpoints and What will be response body format will return from server.
* Can we send request body in JSON or XML format for same end points?

Yes

* can we send request body in combination of JSON and XML format?

No

***4) Authentication and Authorization***

* API documentation, it should provide the instruction regarding how to add Authorization or how to use them
* it provide Authorization types of information

***5) Response code/ status Code and Error Handling code***

* in API documentation must cover various status code and error code message that API Can return in Response.
* This helps to handle different request scenario as like successful request, error code, rate limiting and authentication fail (401/403)

***6) Usages Examples or code Snippets***

* API documentation, it includes the Practical usage examples and code snippets with how to hit the API effectively.
* API documentation, they should provide dummy examples to how to use APIs.

***7) Interactive documentations***

* Interactive documentations means same tools allows to make the request directly without other tools.

***8) Versioning and changelogs***

* In API Documentation, it clearly mention the API version details and any recent changes introduced in each and every version.
* Change log it helps to track the updates .

1. ***What are the different types of Authorization in API***

* ***there are different types of Authorization in API***

***1) No Auth***

***2) Basic Auth***

***3) Digest Auth***

***4) Bearer token***

***5) OAuth 1.0***

***6) OAuth 2.0***

***7) JWT Auth***

***8) API Key***

1. ***What are the different HTTP Request?***

* *There**are 5 most important or famous HTTP Request in API*

1. ***GET Request***

*Get Request is used to retrieve the data from the server.*

*If we hit GET Request*

*it will return the status code as 200*

*It will return the status line as HTTP/1.1 200 OK*

*It will return the whole response Body or Response Payload*

*It will**return response Headers*

1. ***POST Request***

*Post Request is used to create a new Entity in server*

*If we hit POST Request*

*it will return the status code as 201*

*It will return the status line as HTTP/1.1 201 Created*

*It will return the whole response Body or Response Payload*

*It will**return response Headers*

1. ***PUT Request***

*PUT Request is used to update whole Entity in server*

*If we hit PUT Request*

*it will return the status code as 200*

*It will return the status line as HTTP/1.1 200 OK*

*It will return the whole response Body or Response Payload*

*It will**return response Headers*

1. ***PATCH Request***

*PATCH Request is used to update partial Entity in server*

*If we hit PATCH Request*

*it will return the status code as 200*

*It will return the status line as HTTP/1.1 200 OK*

*It will return the whole response Body or Response Payload*

*It will**return response Headers*

1. ***DELETE Request***

*DELETE Request is used to delete the entity from the server.*

*If we hit DELETE Request*

*it will return the status code as 204*

*It will return the status line as HTTP/1.1 204 No Content*

*It will**return response Headers*

1. ***Difference between HTTP GET and POST?***

|  |  |
| --- | --- |
| ***Get*** | ***POST*** |
| It is used to retrieve the data from server | It is used to add new entity in a server |
| Response code is 200 | Response code is 201 |
| Response status line is 200 Ok | Response status line is 201 Created |
| Not required to send request body | Compulsory we have to send request body |
| If we hit GET request 100 times it will retrieve same user 100 times | If we hit POST request 100 times then it will create 100 new users in server |
| Inside the GET Request we can pass request parameter in URI | Inside the POST request we pass request parameter in Body |
| It is not mandatory to send header with GET Request | It is mandatory to send header with POST Request |
| Inside the GET Request we can pass limited amount of data | Inside the POST Request we can pass large amount of data. |
| GET Request can be bookmarked | POST request can not be bookmarked |
| GET is used for view the data | POST used for create the data |
| GET Request is not secured | POST Request is secured |
| GET Request is idempotent means second request will be ignored until response of first request of first is delivered. | POST Request is not idempotent |

1. ***Difference between HTTP GET and PUT?***

|  |  |
| --- | --- |
| ***Get*** | ***PUT*** |
| It is used to retrieve the data from server | It is used to update entity in a server |
| Response code is 200 | Response code is 200 |
| Response status line is 200 Ok | Response status line is 200 Ok |
| Not required to send request body | Compulsory we have to send request body |
| If we hit GET request 100 times it will retrieve same user 100 times | If we hit PUT request 100 times then it will update same user 100 times in server |
| Inside the GET Request we can pass request parameter in URI | Inside the PUT request we pass request parameter in request Body |
| It is not mandatory to send header with GET Request | It is mandatory to send header with PUT Request |
| Inside the GET Request we can pass limited amount of data | Inside the PUT Request we can pass large amount of data. |
| GET Request can be bookmarked | PUT request cannot be bookmarked |
| GET is used for view the data | PUT used for update the data |
| GET Request is not secured | PUT Request is secured |
| GET request is not editable | PUT Request is editable |

1. ***Difference between HTTP GET and PATCH?***

|  |  |
| --- | --- |
| ***Get*** | ***PATCH*** |
| It is used to retrieve the data from server | It is used to update specific entity in a server |
| Response code is 200 | Response code is 200 |
| Response status line is 200 Ok | Response status line is 200 Ok |
| Not required to send request body | Compulsory we have to send request body |
| If we hit GET request 100 times it will retrieve same user 100 times | If we hit PATCH request 100 times then it will update same user 100 times in server |
| Inside the GET Request we can pass request parameter in URI | Inside the PATCH request we pass request parameter in request Body |
| It is not mandatory to send header with GET Request | It is mandatory to send header with PATCH Request |
| Inside the GET Request we can pass limited amount of data | Inside the PATCH Request we can pass large amount of data. |
| GET Request can be bookmarked | PATCH request can not be bookmarked |
| GET is used for view the data | PATCH used for update the data |
| GET Request is not secured | PATCH Request is secured |
| GET request is not editable | PATCH Request is editable |

1. ***Difference between HTTP GET and DELETE?***

|  |  |
| --- | --- |
| ***Get*** | ***DELETE*** |
| It is used to retrieve the data from server | It is used to delete the entity from server |
| Response code is 200 | Response code is 204 |
| Response status line is 200 Ok | Response status line is 204 No Content |
| If we hit GET request 100 times it will retrieve same user 100 times | If we hit DELETE request 100 times then first it will delete the entity from server then second time it will give 404 error code (Entity is not present) |
| GET Request can be bookmarked | DELETE request can not be bookmarked |
| GET is used for view the data | DELETE used for delete the data from server |
| GET Request is not secured | DELETE Request is secured |

1. ***Difference between HTTP POST and PUT?***

|  |  |
| --- | --- |
| ***POST*** | ***PUT*** |
| It is used to add new entity in a server | It is used to update entity in a server |
| Response code is 201 | Response code is 200 |
| Response status line is 201 Created | Response status line is 200 Ok |
| If we hit POST request 100 times then it will create 100 new users in server | If we hit PUT request 100 times then it will update same user 100 times in server |
| POST request can not be bookmarked | PUT request cannot be bookmarked |
| POST used for create the data | PUT used for update the data |
| POST is non Idempotent | PUT request is idempotent so if we send same request multiple times that should be equivalent to first request. |

1. ***Difference between HTTP POST and PATCH?***

|  |  |
| --- | --- |
| ***POST*** | ***PATCH*** |
| It is used to add new entity in a server | It is used to update specific entity in a server |
| Response code is 201 | Response code is 200 |
| Response status line is 201 Created | Response status line is 200 Ok |
| If we hit POST request 100 times then it will create 100 new users in server | If we hit PATCH request 100 times then it will update same user 100 times in server |
| POST request can not be bookmarked | PATCH request cannot be bookmarked |
| POST used for create the data | PATCH used for update the data |
| POST is non Idempotent | PATCH request is idempotent so if we send same request multiple times that should be equivalent to first request. |

1. ***Difference between HTTP POST and DELETE?***

|  |  |
| --- | --- |
| ***POST*** | ***DELETE*** |
| It is used to add new entity in a server | It is used to delete the entity from server |
| Response code is 201 | Response code is 204 |
| Response status line is 201 Created | Response status line is 204 No Content |
| Compulsory we have to send request body | we never send request body in DELETE Request |
| It is mandatory to send header with POST Request | It is not mandatory to send header with DELETE Request |
| If we hit POST request 100 times then it will create 100 new users in server | If we hit DELETE request 100 times then first it will delete the entity from server then second time it will give 404 error code (Entity is not present) |
| POST used for create the data | PUT used for delete the data |
| POST is non Idempotent | PUT request is idempotent so if we send same request multiple times that should be equivalent to first request. |

1. ***Difference between HTTP PUT and PATCH?***

|  |  |
| --- | --- |
| ***PUT*** | ***PATCH*** |
| It is used to update entity in a server | It is used to update partially entity in a server |

1. ***Difference between HTTP PUT and DELETE?***

|  |  |
| --- | --- |
| ***PUT*** | ***DELETE*** |
| It is used to update entity in a server | It is used to delete entity from server |
| Response code is 200 | Response code is 204 |
| Response status line is 200 Ok | Response status line is 204 No Content |
| Compulsory we have to send request body | We never send request body in Delete request |
| If we hit PUT request 100 times then it will update same user 100 times in server | If we hit DELETE request 100 times then first it will delete the entity from server then second time it will give 404 error code (Entity is not present) |
| Inside the PUT request we pass request parameter in request Body | We never send request body in Delete request |
| It is mandatory to send header with PUT Request | It is not mandatory to send header with PUT Request |
| Inside the PUT Request we can pass large amount of data. | Inside the DELETE Request we never pass data. |
| PUT used for update the data | DELETE request is used to delete the data |

1. ***Difference between HTTP PATCH and DELETE?***

|  |  |
| --- | --- |
| ***PATCH*** | ***DELETE*** |
| It is used to update entity in a server | It is used to delete entity from server |
| Response code is 200 | Response code is 204 |
| Response status line is 200 Ok | Response status line is 204 No Content |
| Compulsory we have to send request body | We never send request body in Delete request |
| If we hit PATCH request 100 times then it will update same user 100 times in server | If we hit DELETE request 100 times then first it will delete the entity from server then second time it will give 404 error code (Entity is not present) |
| Inside the PATCH request we pass request parameter in request Body | We never send request body in Delete request |
| It is mandatory to send header with PATCH Request | It is not mandatory to send header with PUT Request |
| Inside the PATCH Request we can pass large amount of data. | Inside the DELETE Request we never pass data. |
| PATCH used for update the data | DELETE request is used to delete the data |

1. ***What are the different points we verify in API?***

* ***We verify different points in HTTP Request***

1. *Status code*
2. *Status Line*
3. *Response body*
4. *Response Headers*
5. *Response cookies*
6. *Response time*
7. *Authorization check*
8. *Error code in case in API return any code*
9. *Data Accuracy*