API IQ:

1. **What is API?**

   API is the acronym for Application Programming Interface which is a connection layer/interface between 2 or more different systems.
   This layer allows to exchange communication and data exchange between 2 or more separate software systems.
   APIs work using requests and responses. When an API requests information from a web application or web server, it will receive a response.

2. **What is a Web Service? API vs Web Service?**

   Web Service is an API that travels through the internet.
   All Web Services are APIs, but not all APIs are Web Services.

3. **What type of Web Services you know? What is the difference?**

   There are 2 types of web services.

   1. **SOAP web services**

      SOAP (Simple Object Access Protocol) - is based on transferring XML data as SOAP Messages.
      SOAP is more secure, however it is slower than REST.

   2. **RESTful web services (REST API)**
      REST (Representational State Transfer) - that uses different representations to exchange and transfer data in JSON, XML or TEXT format.
      REST is a lightweight programs and it is less secure compared to SOAP, but it is fast.

4. **Which Protocol is used by RESTful Web Services?**

   RESTful web services uses of HTTP/HTTPS protocols as a medium of communication between client and server.

5. **Most Commonly Used HTTP Methods supported By REST?**

POST – It submits information to the service for processing; it should typically return the modified or new resource → **Create**
GET -It requests a resource at the request-URL. It should not contain a request body → **Retrieve**
PUT – Replaces all current representations of the target resource with the uploaded content → **Update**
DELETE – Removes all current representations of the target resource given by a URI → **Delete**

6. Can GET request to be used instead of PUT to create a resource?

The PUT or POST method should be used to create a resource. GET is only used to request data from a specified resource.

7. What is the difference between PUT and POST request?

Using **POST** request, our intent is to create a new object on the server whereas with **PUT** request, our intent is to replace an object by another object.

POST should be used when the client sends the page to the server and then the server lets the client know where it put it. **PUT** should be used when the client specifies the location of the page.

8. Which HTTP Status codes your know?

1xx → Informational
2xx → Success (request was accepted successfully) (200→ Ok, 201→ Created, 202→ Accepted, 204→ No Content)
3xx → Redirection
4xx → Client Error (400-Bad Request, 401-Unauthorized, 403-Forbidden, 404-Not Found, 405-Method not Allowed)
5xx → Server Error (500-Internal server Error, 501-Not implemented, 502-Bad Gateway, 503-Service Unavailable)

9. What is API Testing?

Is a type of Testing which determines if the developed APIs meet expectations regarding functionality, reliability, performance and security of the application.

10. What are the advantages of API Testing?

Test for Core Functionality: API testing provides access to the application without a user interface. The core and code-level of functionality of the application will be tested and evaluated early before the GUI tests. This will help detect the minor issues which can become bigger during the GUI testing.
Time Effective: API testing usually is less time consuming than functional GUI testing. The web elements in GUI testing must be polled, which makes the testing process slower. API test automation requires less code so it can provide better and faster test coverage compared to GUI test automation. These will result in the cost saving for the testing project.
Language-Independent: In API testing, data is exchanged using XML or JSON. These transfer modes are completely language-independent, allowing users to select any code language when adopting automation testing services for the project.

11. What tools can be used to test APIs? How do you test APIs in your project?

In my project we use REST APIs and manually I test is using PostMan and to automate REST APIs I use RestAssured Library which is Java based library.
As a tester I send an API request (weather it is a GET, POST, PUT or DELETE call) and then I verify the status code, response body and checking headers.
I verify that each endpoint working as expected.
I do positive and negative testing of APIs:
Positive - I am sending valid requests, headers, parameters, and Json body and then verify that response is 200/201
Negative- I am sending invalid requests, headers, parameters, and body, expecting to the status to not be 200.

12. What is EndPoint?

An endpoint by itself is the location where a resource can be accessed
/api/createStudentProfile

An endpoint is one end of a communication channel. When an API interacts with another system, the touchpoints of this communication are considered endpoints.

13. Do you have API documentation website for your API? Any other API documentations you know?

Swagger is an open-source software framework backed by a large ecosystem of tools that helps developers design, build, document, and consume RESTful Web services.

Some of the API documentation templates:

- Swagger
- FlatDoc
- RestDoc
- API blueprint

14. What is Request and Response contains of?

HTTP request method is made up of four components:

- Request Method: Get, Post, Put, Delete
- Request URI: complete URL of the resource
- Request Header: Accept, Content-Type
- Request Body: data to be sent to the resource

HTTP response method is made up of three components:

- Response Status Code:200, 201, 400, 404, 500
- Response Header: Date, Server, Last-Modified, Content-Type
- Response Body:data that comes back to the client from the server

15. What is JSON?

- It is JavaScript Object Notation (is a minimal, readable format for structuring data.)
- It is used primarily to transmit data between a server and web application, as an alternative to XML (a lightweight version of XML)
- Represent Data in a Key : Value format

16. What is Parameters in API? Types of Parameters?

Parameters are options you can pass with the endpoint to influence the response.
In REST we 2 types of Parameters:

- Path Parameters
  As part of the URL-path (i.e. /api/resource/parametervalue)

- Query Parameters
  As a query argument (i.e. /api/resource?parameter=value)

17. What is Headers in API?

Headers provide meta-information about a request.
In my project when I send POST or PUT request in header I specify that ContentType I am using is JSON.

And whenever I receive response I verify that it contains JSON as well so then using GSON library I can perform deSerialization of JSON to Java Object.

```
RestAssured.baseURI="http://pure-ravine-92491.herokuapp.com/syntax";

    Response rsp=
    given().
            accept(ContentType.JSON).
            header("Content-Type", "application/json").
            body(map).
    when().
            post("/api/createStudentProfile");

    rsp.then().assertThat().statusCode(201).
    and().header("Content-Type", equalTo("application/json;charset=UTF-8"));

    Map<String, Object>responseMap=rsp.as(Map.class);
```

18. What is Payload?

Request and response body of every HTTP message includes request data called as Payload.

We send the payload in the POST and PUT methods but not in <GET> and <DELETE> methods.

The response payload is often a response of your GET (returning a record or a list of records), or a confirmation from a POST.

19. How do you verify a value in your Response body?

To validate value in Response we can use:

1. JUnit Assertions

```
RestAssured.baseURI="https://got-quotes.herokuapp.com";
Response rsp=
        given().
                queryParam("char", "varys").
        when().
                get("/quotes");

        int code=rsp.getStatusCode();
        Assert.assertEquals(200, code);

        String responseBody=rsp.asString();
        Assert.assertTrue(responseBody.contains("Varys"));
```

2. HamCrest Matchers

```
RestAssured.baseURI="https://got-quotes.herokuapp.com";
Response rsp=
        given().
                queryParam("char", "varys").
        when().
                get("/quotes").
        then().
                assertThat().statusCode(200).
        and().
                assertThat().body("character", equalTo("Varys"));
```

3. DeSerialization

Retrieve data from JSON and store inside Java DataStructure

Map<String, Object>responseMap=rsp.as(Map.class);

Once we have Java Object we can compare it using JUnit Assertions.

20. What are the main challenges faced in API testing?

- Selecting proper parameters and its combinations (what if you do not have correct documentation and you need to work with parameters).
- Categorizing the parameters properly (path or query).
- Proper call sequencing is required as this may lead to inadequate coverage in testing.
- Verifying and validating the output (Request & Response)
- Due to absence of GUI it is quite difficult to provide input values