1. **What is API ?**
    - An API (Application Programming Interface) is a software intermediary that enables two applications to communicate with each other. It comprises a number of subroutine definitions, logs, and tools for creating application software.
    - In an API testing interview, you could be asked to give some API examples, here are the well-known ones: Google Maps API, Amazon Advertising API, Twitter API, YouTube API, etc.
2. **What is API Testing?**
    - In the modern development world, many web applications are designed based on three-tier architecture model. These are:
    - Presentation Tier – User Interface (UI)
    - Logic Tier – Business logic is written in this tier. It is also called Business Tier. (API)
    - Data Tier – Here information and data is stored and retrieved from a Database. (DB)
    - Ideally, these three layers (tiers) should not know anything about the platform, technology, and structure of each other. We can test UI with GUI testing tools and we can test logic tier (API) with API testing tools. Logic tier comprises of all of the business logic and it has more complexity than the other tiers and the test executed on this tier is called as API Testing.
    - API testing tests logic tier directly and checks expected functionality, reliability, performance, and security. In the agile development world, requirements are changing during short release cycles frequently and GUI tests are more difficult to maintain according to those changes. Thus, API testing becomes critical to test application logic.
    - In GUI testing we send inputs via keyboard texts, button clicks, drop-down boxes, etc., on the other hand in API testing we send requests (method calls) to the API and get output (responses). These APIs are generally REST APIs or SOAP web services with JSON or XML message payloads being sent over HTTP, HTTPS, JMS, and MQ.
3. **What are main differences between API and Web Service?**
    - All Web services are APIs but not all APIs are Web services.
    - All web services need to be exposed over web(HTTP) but All APIs need not be exposed over web(i.e. HTTP)
    - Web services might not contain all the specifications and cannot perform all the tasks that APIs would perform.
    - A Web service uses only three styles of use: SOAP, REST and XML-RPC for communication whereas API may be exposed to in multiple ways e.g. DLL files in C/C++, Jar files/ RMI in java, Interrupts in Linux kernel API etc.
    - A Web service always needs a network to operate while APIs don't need a network for operation.
4. **What are some architectural styles for creating a Web API?**
    - Bellows are four common Web API architectural styles:
        1. HTTP for client-server communication
        2. XML/JSON as formatting language
        3. Simple URI as the address for the services
        4. Stateless communication
5. **Who can use a Web API?**

  - Web API can be consumed by any clients which support HTTP verbs such as GET, PUT, DELETE, POST.
  - Since Web API services do not require configuration, they can be easily used by any client.
  - In fact, even portable devices such as mobile devices can easily use Web API, which is undoubtedly the biggest advantage of this technology.

6. **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 functionalities 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. Particularly, 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.
   - Easy Integration with GUI: API tests enable highly integrable tests, which is particularly useful if you want to perform functional GUI tests after API testing. For instance, simple integration would allow new user accounts to be created within the application before a GUI test started.
7. **Some common protocols used in API testing?**
   - Many protocols are now available to be used in API testing, such as JMS, REST, HTTP, UDDI and SOAP
8. **What are the common API testing types?**
   - Most tests fit broadly into these following nine categories:
   - Validation Testing
   - Functional Testing
   - Load testing
   - Runtime/ Error Detection
   - Security testing
   - Penetration(pen) testing
   - Fuzz testing
   - Interoperability and WS Compliance testing.
9. **What are tools could be used for API testing?**
   - Postman
   - Katalon Studio
   - SoapUI
   - Assertible
   - Tricentis Tosca
   - Apigee
   - 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

10. **What are differences between API Testing and UI Testing?**
    - API enables communication between two separate software systems. A software system implementing an API contains functions or subroutines that can be executed by another software system.
    - On the other hand, UI (User Interface) testing refers to testing graphical interface such as how users interact with the applications, testing application elements like fonts, images, layouts etc. UI testing basically focuses on look and feel of an application.
11. **What are major challenges faced in API testing?**
    - Parameter Selection
    - Parameter Combination
    - Call sequencing
    - Output verification and validation
    - Another important challenge is providing input values, which is very difficult as GUI is not available in this case.
12. **What are the testing methods that come under API testing?**
    - Unit testing
    - End to End Integration testing
    - Functional testing
    - Load testing to test the performance under load
    - Usability and Reliability testing to get consistent results
    - Security and Penetration testing to validate all types of authentication
    - Automation testing to create and run scripts that require regular API calls
13. **What is API documentation?**
    - The API documentation is a complete, accurate technical writing giving instructions on how to effectively use and integrate with an API. It is a compact reference manual that has all the information needed to work with the API, and helps you answer all the API testing questions with details on functions, classes, return types, arguments, and also examples and tutorials.
14. **What are API documentation templates that are commonly used?**
    - There are several available API documentation templates help to make the entire process simple and straightforward, such as:

- Swagger
- Miredot
- Slate
- FlatDoc
- API blueprint
- RestDoc
- Web service API specification

15. **What is a RESTFul Web Services?**
    - Mostly, there are two kinds of Web Services which should be remembered :
        1. SOAP (Simple Object Access Protocol) – an XML-based method to expose web services.
        2. REST **(Representational State Transfer)** is an architectural style for developing web services over HTTP protocol and uses HTTP method to define actions. It revolves around resource where every component being a resource that can be accessed through a shared interface using standard HTTP methods.
        3. Web services developed in the REST style are referred to as RESTful web services. These web services use 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 like JSON and a set of HTTP methods.
16. **What is a “Resource” in REST?**
    - REST architecture treats any content as a resource, which can be either text files, HTML pages, images, videos or dynamic business information.
    - REST Server gives access to resources and modifies them, where each resource is identified by URIs/ global IDs.
17. **What are the core components of an HTTP request?**
    - An HTTP request contains five key elements:
    - An action showing HTTP methods like GET, PUT, POST, DELETE.HEAD
    - Uniform Resource Identifier (URI), which is the identifier for the resource on the server.
    - HTTP Version, which indicates HTTP version, for example-HTTP v1.1.
    - Request Header, which carries metadata (as key-value pairs) for the HTTP Request message. Metadata could be a client (or browser) type, format supported by the client, format of a message body format, cache settings, and so on.
    - Request Body, which indicates the message content or resource representation.
18. **What is URI? What is the main purpose of REST-based web services and what is its format?**
    - URI stands for Uniform Resource Identifier. It is a string of characters designed for unambiguous identification of resources and extensibility via the URI scheme.
    - The purpose of a URI is to locate a resource(s) on the server hosting of the web service.
    - A URI’s format is <protocol>://<service-name>/<ResourceType>/<ResourceID>
19. **What is payload in Restful Web services?**
    - The “payload” is the data you are interested in transporting. This is differentiated from the things that wrap the data for transport like the HTTP/S Request/Response headers, authentication, etc.
20. **What is the upper limit for a payload to pass in the POST method?**
    - <GET> appends data to the service URL. But, its size shouldn’t exceed the maximum URL length. However, <POST> doesn’t have any such limit.

- So, theoretically, a user can pass unlimited data as the payload to POST method. But, if we consider a real use case, then sending POST with large payload will consume more bandwidth. It'll take more time and present performance challenges to your server. Hence, a user should take action accordingly.

21. **Enlist some of the API examples which are very well known and popular.**
    - There are several such examples, enlisted below are some most popular ones:
        1. Google Maps API: These are designed mainly for mobile and desktop use with the help of flash interface and JavaScript.
        2. Amazon Advertising API: Amazon is known for their products and thus their advertising API accesses their product to discover their functionality and thus advertise accordingly.
        3. Twitter: The API for twitter is usually in two categories, one for accessing data and the other for interacting with twitter search.
        4. YouTube: This API used for YouTube includes various functionalities including videos, live streaming, player, etc.
22. **Differentiate API testing and Unit Testing.**
    - Unit testing is usually performed by testers where every functionality is tested separately.
        1. API testing is performed by the testers for end to end testing of the functionality.
    - Unit testing have the limited scope of testing, thus basic functionalities are only considered for testing.
        1. API testing have the broader scope of testing, all issues that are functional are considered for testing.
    - Unit testing is a form of white box testing.
        1. API testing is a form of black box testing.
    - Usually, unit testing is done before the code is included in the build.
        1. API testing is performed after the build is ready for testing.
    - In Unit testing the Source code is involved in this form of testing.
        1. In API testingSource code is not involved in this form of testing.
23. **What is Rest Assured?**
24.

    - In order to test REST APIs, We have RESTAssured library. It is developed by JayWay Company and it is a really powerful catalyzer for automated testing of REST-services. REST-assured provides a lot of nice features, such as DSL-like syntax, XPath-Validation, Specification Reuse, easy file uploads and with those features we will handle automated API testing much easier.
    - Rest Assured has a gherkin type syntax which is as BDD (Behavior Driven Development):
    - Also, you can get JSON response as a string and send it to the JsonPath class and use its methods to write more structured tests.
25. **How to Make a POST Request with RestAssured?**

- Rest AP IURL – URL of the Rest API
- API Body – Body of the Rest API. Example: {“key1″:”value1″,”key2″:”value2”}
- setContentType() – Pass the “application/json”, “application/xml” or “text/html” etc. headers to setContenType() method.
- Authentication credentials – Pass the username and password to the basic() method or if there is no authentication leave them blank basic(“”,””)

26. **What are the different Response code in RestAssured?**
    - **1xx: Informational :**Communicates transfer protocol-level information.
    - 2xx: Success:Indicates that the client’s request was accepted successfully.
    - 3xx: Redirection:Indicates that the client must take some additional action in order to complete their request.
    - 4xx: Client Error:This category of error status codes points the finger at clients.
    - 5xx: Server Error :The server takes responsibility for these error status codes.
27. **What can we have in given() in Rest Assured?**
    - Given() can have headers(), parameters()[Path parameter with {},Query . Parameter with ? Header paremeter with cookies, body(), ContentType(),relaxedHTTPSValidation(),etc
    - In Post we should not send the data in url
28. **What can we have in When() in Rest Assured?**
    - when() can hit the resource with get(),post(),put(),delete()
29. **What can have in Then() in Rest Assured?**
    - will get the response so we can assert with mwthod assertThat() for statusCode(),body() .extract() will extract the information
30. **What are SOAP Web services?**
    - The SOAP (Simple Object Access Protocol) is defined as an XML-based protocol. It is known for designing and developing web services as well as enabling communication between applications developed on different platforms using various programming languages over the Internet. It is both platform and language independent.
31. **SOAP or Rest APIs, which method to use?**
    - **SOAP is the heavyweight choice for Web service access. It provides the following advantages/disadvantage when compared to REST:**
        1. SOAP is not very easy to implement and requires more bandwidth and resources.
        2. SOAP message request is processed slower as compared to REST and it does not use web caching mechanism.
        3. WS-Security: While SOAP supports SSL (just like REST) it also supports WS-Security which adds some enterprise security features.
        4. WS-Atomic Transaction: Need ACID Transactions over a service, you’re going to need SOAP.
        5. WS-ReliableMessaging: If your application needs Asynchronous processing and a guaranteed level of reliability and security. Rest doesn’t have a standard messaging system and expects clients to deal with communication failures by retrying.
        6. If the security is a major concern and the resources are not limited then we should use SOAP web services. Like if we are creating a web service for

    payment gateways, financial and telecommunication related work, then we should go with SOAP as here high security is needed.

- **REST is easier to use for the most part and is more flexible. It has the following advantages when compared to SOAP:**
    1. Since REST uses standard HTTP, it is much simpler.
    2. REST is easier to implement, requires less bandwidth and resources.
    3. REST permits many different data formats whereas SOAP only permits XML.
    4. REST allows better support for browser clients due to its support for JSON.
    5. REST has better performance and scalability. REST reads can be cached, SOAP based reads cannot be cached.
    6. If security is not a major concern and we have limited resources. Or we want to create an API that will be easily used by other developers publicly then we should go with REST.
    7. If we need Stateless CRUD operations then go with REST.
    8. REST is commonly used in social media, web chat, mobile services and Public APIs like Google Maps.
    9. RESTful service returns various MediaTypes for the same resource, depending on the request header parameter “Accept” as application/xml or application/json for POST and /user/1234.json or GET /user/1234.xml for GET.
    10. REST services are meant to be called by the client-side application and not the end user directly.