**Types of Test Cases To Be Automated**

We know that automation testing saves a lot of time and cost. Some managers think that an automation tester needs to automate complete application and also a few teams rush to automate everything.

There are some types of test cases we must automate and also some test cases we must not automate. We shouldn’t automate all the test cases. Earlier we have discussed on [**8 Types of Test Cases Not To Be Automated**](https://www.softwaretestingmaterial.com/test-cases-not-to-be-automated/).

|  |  |
| --- | --- |
| **Types of tests need to be performed manually**  [1.Subjective Validation:](https://www.softwaretestingmaterial.com/test-cases-not-to-be-automated/#h-1-subjective-validation)  [2.New Functionalities:](https://www.softwaretestingmaterial.com/test-cases-not-to-be-automated/#h-2-new-functionalities)   * + [3.Strategic Development:](https://www.softwaretestingmaterial.com/test-cases-not-to-be-automated/#h-3-strategic-development)   + [4.User Experience:](https://www.softwaretestingmaterial.com/test-cases-not-to-be-automated/#h-4-user-experience)   + [5.Complex Functionality:](https://www.softwaretestingmaterial.com/test-cases-not-to-be-automated/#h-5-complex-functionality)   + [6.Quality Control:](https://www.softwaretestingmaterial.com/test-cases-not-to-be-automated/#h-6-quality-control)   + [7.Low return on investment:](https://www.softwaretestingmaterial.com/test-cases-not-to-be-automated/#h-7-low-return-on-investment)   + [8.Installation and setup testing:](https://www.softwaretestingmaterial.com/test-cases-not-to-be-automated/#h-8-installation-and-setup-testing) | * [**Types of Test Cases To Automate**](https://www.softwaretestingmaterial.com/test-cases-to-be-automated/#h-types-of-test-cases-to-automate)   + [1. Data-driven test cases:](https://www.softwaretestingmaterial.com/test-cases-to-be-automated/#h-1-data-driven-test-cases)   + [2. Test cases with higher complexity:](https://www.softwaretestingmaterial.com/test-cases-to-be-automated/#h-2-test-cases-with-higher-complexity)   + [3. Test case with many database updates:](https://www.softwaretestingmaterial.com/test-cases-to-be-automated/#h-3-test-case-with-many-database-updates)   + [4. The test execution rate is high:](https://www.softwaretestingmaterial.com/test-cases-to-be-automated/#h-4-the-test-execution-rate-is-high)   + [5. Smoke/Critical tests:](https://www.softwaretestingmaterial.com/test-cases-to-be-automated/#h-5-smoke-critical-tests)   + [6. Tests with several combinations:](https://www.softwaretestingmaterial.com/test-cases-to-be-automated/#h-6-tests-with-several-combinations)   + [7. Graph test cases:](https://www.softwaretestingmaterial.com/test-cases-to-be-automated/#h-7-graph-test-cases)   + [8. Higher manual execution time:](https://www.softwaretestingmaterial.com/test-cases-to-be-automated/#h-8-higher-manual-execution-time) |

### **Data-driven test cases:**

Data-driven test cases are suitable for those testing since they require multiple data sets or require several data entries such as testing that needs a username and password, filling up the forms with a multiple data variation. Let’s look at some potential examples

* You want to test different roles with different permissions. In one of my healthcare projects, I had to test around 20 roles and each role has its own set of permissions. Imagine manually logging in as each role and testing its permissions. Also, imagine running these tests as part of [smoke testing](https://www.softwaretestingmaterial.com/smoke-testing-vs-sanity-testing/) which happens frequently. So, automating these tests is ideal.
* In another scenario, I was testing an application that monitors healthcare devices such as CT and MR machines. The application would monitor several parameters maybe 50 of them and based on threshold raise an alert if something is wrong. Example: if hard drive is full raise an alarm. Now imagine testing alert for each parameter. It used to take 2 days to test all combinations however after automation, testing completed in 30 minutes.

### **2. Test cases with higher complexity:**

Test cases requiring a higher level of preparation and setup work should be automated.  
Let’s say you are testing CRM (Customer Relationship Management) application and you might want to hit CRM API, but before that, you need to set up authentication, you might need to hit some other APIs in order to finally test the CRM APIs. This might easily take some time and to test several scenarios might take a day or two.

### **3. Test case with many database updates:**

If your test cases involve multiple database entries, it should be a good candidate for automation. Imagine a scenario that you only have one credential to login but want to test different permissions. In one of my projects we were not allowed to have more than one credential, so we changed permissions for the role and logged in with the same credentials. Performed testing. I had to test different roles but with one login credentials. I automated that by writing an automation script where the script would first change permissions, login, and execute the test case. It saved me a lot of time.

### **4. The test execution rate is high:**

Test cases that are executed in every release are suitable candidates to be automated. Smoke or sanity test cases are the best examples for higher execution.

### **5. Smoke/Critical tests:**

Smoke or critical tests are performed for those features and functionalities that are frequently used in the automation process. The smoke test includes critical test cases. Sometimes you want to make sure your production environment is always up and running and to achieve that you automate critical tests and run them every single morning.

### **6. Tests with several combinations:**

Test cases that involve different combinations or configurations are automated to save time and to avoid human errors. Don’t get confused about this with any of the above categories. Let me give you an example; Imagine an application that connects to different hardware and verifies hardware configurations and software configurations installed on it. You would store those expected parameters either in the database or XML file, connect to the device, and compare your stored parameters with the actual parameter values store on the device. This will become a challenge to test manually when you would have to test several devices with different combinations.

### **7. Graph test cases:**

Graphs are used to display data in a pictorial format. Graph testing requires plenty of data manipulations. It also includes different scenarios such as testing with plus and minus values, valid and invalid data, etc. [Automation testing](https://www.softwaretestingmaterial.com/automation-testing-tutorial/) with graphical test case not only save time but also help to maintain the accuracy of graphical data.

### **8. Higher manual execution time:**

Sometimes test execution takes a lot of time even when tests are short, not complex in nature, or don’t involve data variations. Those tests can also be automated to save time and avoid repetitive effort.

For example, I was working on a project, and application under test had many interlinked/interdependent modules which means changes in one module might affect functionality in other modes. We had to test all modules by executing all modules tests which sometimes led us to 200 test cases or more and easily take 4 to 5 days. Running those tests in one iteration for 5 days would cost a lot of time and money. So, we automated around all the interdependent tests and were able to run one iteration in just a day which saved us 4 days of time.

FRAMEWORK

### **What is a Framework*?***

A) A framework defines a set of rules or best practices that we can follow in a systematic way to achieve the desired results. So the above-mentioned test automation frameworks deal with best practices to achieve the goals of our automation project.

## **What is Selenium Framework?**

Selenium framework’s code structure helps you to reuse the code, reduce code maintenance, higher code readability, and allows multiple users to work on the same piece of the program.

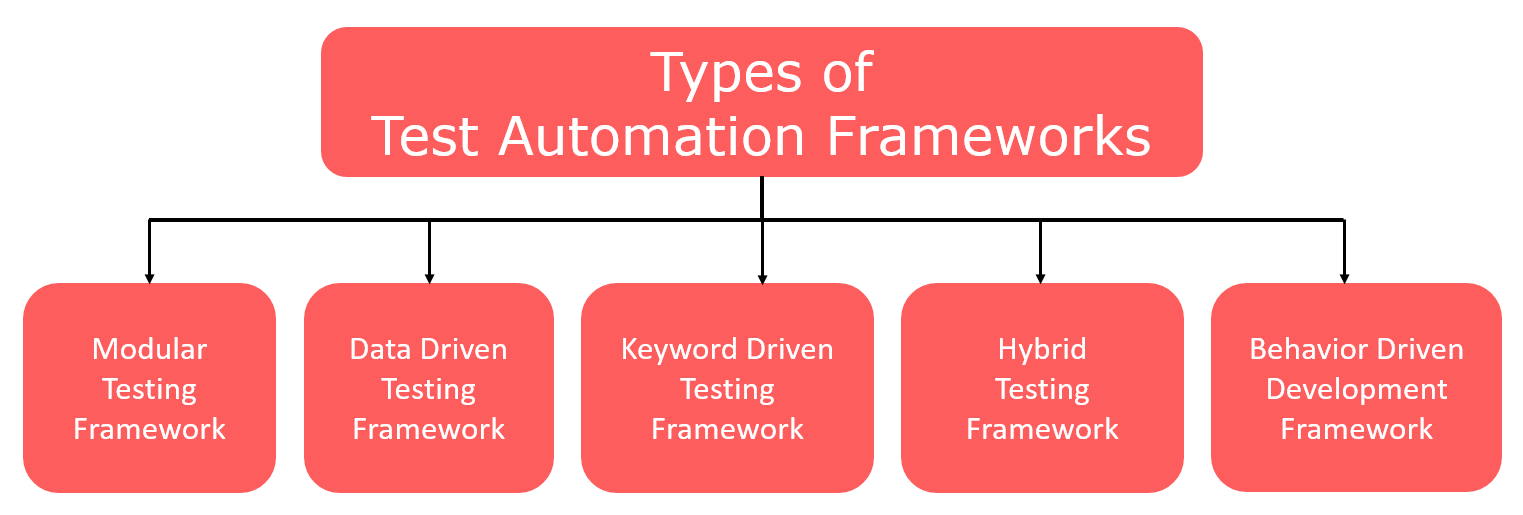
## **Why do we need the Selenium Framework?**

* Easy code maintenance
* Increase in code re-usage
* Higher code readability
* Reduced script maintenance cost
* Reduced tests’ time execution
* Reduced human resources
* Easy reporting

## **Types of Test Automation Frameworks:**

Here in this post, I will explain the most popular**Types of Selenium Automation Frameworks**.

* 1. [Linear Scripting Framework](https://www.softwaretestingmaterial.com/types-test-automation-frameworks/#Linear-Scripting-Framework)
* 2. [Modular Testing Framework](https://www.softwaretestingmaterial.com/types-test-automation-frameworks/#Modular-Testing-Framework)
* 3. [Library Architecture Testing Framework](https://www.softwaretestingmaterial.com/types-test-automation-frameworks/#Library-Architecture-Testing-Framework)
* 4. [Data-driven Testing Framework](https://www.softwaretestingmaterial.com/types-test-automation-frameworks/#Data-driven-Framework)
* 5. [Keyword Driven Testing Framework](https://www.softwaretestingmaterial.com/types-test-automation-frameworks/#Keyword-Driven-Testing-Framework)
* 6. [Hybrid Testing Framework](https://www.softwaretestingmaterial.com/types-test-automation-frameworks/#Hybrid-Driven-Testing-Framework)
* 7. [Behavior Driven Development Testing Framework](https://www.softwaretestingmaterial.com/types-test-automation-frameworks/" \l "Behavior-Driven-Development-Testing-Framework)



### **Linear Scripting Framework:**

Linear Scripting Framework is a basic level test automation framework that is in the form of ‘Record and Playback’ in a linear fashion.

This framework is also known as the ‘Record and Playback’ framework.

This type of framework is used to test small-sized applications.

In this type, the creation, and execution of test scripts are done individually for each test case individually.

Testers capture each test step such as browsing, navigation, user inputs, enforcing checkpoints. Testers then play the scripts to carry out the tests.

#### **Advantages of Linear Scripting Automation Framework:**

* Can generate test scripts (Record and playback) without planning much or consume much time
* Coding knowledge is not required
* A quick way to generate test scripts

#### **Disadvantages of Linear Scripting Automation Framework:**

* Lack of reusability due to autogenerated scripts
* Hard coding the data doesn’t allow us to run with multiple data sets
* Maintenance is high – It requires a lot of effort to do even small changes.

### **Modular Testing Framework:**

In the modular testing framework, testers create test scripts module wise by breaking down the complete application under test into smaller, independent tests.

In simple words, testers divide the application into multiple modules and create test scripts individually. These individual test scripts can be combined to make larger test scripts by using a master script to achieve the required scenarios. This master script is used to invoke the individual modules to run end to end test scenarios.

The main reason for using this framework is to build an abstraction layer to safeguard the master module from any changes made in individual tests.

In this framework, testers write function libraries to use it whenever required. This is AKA modularity framework or module-based framework.

#### **Advantages of Modular Testing Framework:**

* Better scalability and easier to maintain due to breaking down the complete application into different modules
* Can write test scripts independently
* Changes in one module bring no or low impact on the other modules

#### **Disadvantages of Modular Testing Framework:**

* Takes more time to analyze the test cases and to identify reusable flows
* Due to hardcoded data in the test scripts, it’s not possible to sue multiple data sets.
* Requires coding skills to set up the framework

### **Library Architecture Testing Framework:**

Library Architecture Testing framework aka “Structured Scripting” or “Functional Decomposition”

It is based on the modular framework with some additional advantages.

In the modular testing framework, we divide the application under test into modules whereas here we identify the common tasks and grouped them into functions. Once the functions are grouped then these groups will be kept in a library. The test scripts reuse these libraries to create new test cases.

#### **Advantages of a LIbrary Architecture Testing Framework:**

* Script maintenance is simple
* Easy to scalable
* Functions library is reusable and it can be reusable

#### **Disadvantages of a LIbrary Architecture Testing Framework:**

* Coding skills are required
* It takes more time to prepare test scripts
* A fixed set of test data is hardcoded within the scripts

### **Data-driven Framework:**

The data-driven test automation framework is focused on separating the test scripts logic and the test data from each other.

It allows us to create test automation scripts by passing different sets of test data.

The test data set is kept in the external files or resources such as MS Excel Sheets, MS Access Tables, SQL Database, XML files, etc.,

The test scripts connect to the external resources to get the test data.

By using this framework we could easily make the test scripts work properly for different sets of test data.

This framework significantly reduces the number of test scripts compared to the module-based framework.

This framework gives more test coverage with reusable tests and flexibility in the execution of tests only when required and by changing only the input test data.

It is reliable in terms of no impact on tests by changing the test data but it has its own drawbacks such as testers who work on this framework needs to have the hands-on programming knowledge to develop test scripts

#### **Advantages of a Data-Driven Framework:**

* It supports multiple data sets
* Modifying the test scripts won’t affect the test data
* No need to hardcode test data
* Saves time by executing more tests

#### **Disadvantages of a Data-Driven Framework:**

* Require coding skills
* Setting up the framework and test data takes more time
* Need experienced automation testers to design framework

### **Keyword Driven Testing Framework:**

It is also known as table-driven testing or action word based testing.

In Keyword-driven testing, we use a table format to define keywords or action words for each function or method that we would execute.

It performs automation test scripts based on the keywords specified in the excel sheet.

By using this Framework, testers can work with keywords to develop any test automation script, testers with less programming knowledge would also be able to work on the test scripts.

The logic to read keywords and call the required action mentioned in the external excel sheet is placed in the main class. Keyword-driven testing is similar to data-driven testing.

Even though to work on this framework doesn’t require much programming skills but the initial setup ( implement the framework) requires more expertise.

#### **Advantages of Keyword-Driven Frameworks:**

* No need to be an expert to write test scripts
* It is possible to reuse the code. We can point the different scripts to the same keyword
* Even though application changes, test scripts don’t change.
* Tests can be designed before developing the application
* Test scripts work independently of an application under test with basic modifications
* Not dependent on test tools

#### **Disadvantages of Keyword-Driven Frameworks:**

* Take more time to design
* The initial cost is high
* Employees with good test automation skills needed

### **Hybrid Driven Testing Framework:**

Hybrid Test automation framework is the combination of two or more frameworks mentioned above. It attempts to leverage the strengths and benefits of other frameworks for the particular test environment it manages. Most of the teams are building this hybrid driven framework in the current market.

### **Behavior Driven Development Testing Framework:**

The purpose of this Behavior Driven Development framework is to create a platform that allows everyone (such as Business Analysts, Developers, Testers, etc,) to participate actively. It requires increased collaboration between Development and Test Teams.  It doesn’t require the users to be acquainted with a programming language. We use non-technical, natural language to create test specifications. Some of the tools available in the market for Behavior Driven Development is [JBehave](http://jbehave.org/" \t "_blank), [Cucumber](https://cucumber.io/), etc.,

The frameworks stated above are some of the most popular **Test Automation Frameworks** used by the automation testers.

**Conclusion:**

* Even though we have seen different types of test automation frameworks. Popular test automation frameworks out of all are Data driven framework, Keyword driven framework, and Hybrid framework.
* In the Data-Driven Framework, we keep test data in excel sheets & use TestNG’s data provider to execute test cases.
* In Keyword driven framework, we define keywords in the excel sheet and the code will call this file to execute the test cases
* The hybrid framework is a combination of both Data-driven and Keyword driven frameworks

**How To Explain Test Automation Framework To The Interviewer**

**Language:**In our Selenium Project we are using Java language. Even though Selenium supports multiple languages, we are using Java language is just because most of the automation developers have knowledge on Selenium with Java.

**Type of Framework:** In our project, we are using [Data-driven Framework](https://www.softwaretestingmaterial.com/data-driven-framework-selenium-webdriver/) by using [Page Object Model design pattern](https://www.softwaretestingmaterial.com/page-object-model/) with Page Factory.

**POM:**As per the Page Object Model, we have maintained a class for every web page. Each web page has a separate class and that class holds the functionality and members of that web page. Separate classes for every individual test.

**Packages:** We have separate packages for Pages and Tests. All the web page related classes come under the **Pages** package and all the tests related classes come under **Tests** package.

**Test Base Class:**Test Base class (TestBase.java) deals with all the common functions used by all the pages. This class is responsible for loading the configurations from properties files, Initializing the WebDriver, Implicit Waits, Extent Reports, and also to create the object of FileInputStream which is responsible for pointing towards the file from which the data should be read.

**Utility Class (AKA Functions Class):**Utility class (TestUtil.java) stores and handles the functions (The code which is repetitive in nature such as waits, actions, capturing screenshots, accessing excels, sending email, etc.,) which can be commonly used across the entire framework. The reason behind creating a utility class is to achieve reusability. This class extends the TestBase class to inherit the properties of TestBase in TestUtil.

**Properties file:**This file (**config.properties**) stores the information that remains static throughout the framework such as browser-specific information, application URL, screenshots path, etc.

All the details which change as per the environment and authorization such as URL, Login Credentials are kept in the config.properties file. Keeping these details in a separate file makes it easy to maintain.

**Screenshots:** Screenshots will be captured and stored in a separate folder and also the screenshots of failed test cases will be added to the extent reports.

**Test Data:** All the historical test data will be kept in an excel sheet (controller.xlsx). By using ‘controller.xlsx’, we pass test data and handle data-driven testing. We use [Apache POI](https://www.softwaretestingmaterial.com/handling-excel-files-using-apache-poi/) to handle excel sheets.

**TestNG:** Using TestNG for Assertions, Grouping, and Parallel execution.

**Maven:** Using Maven for build, execution, and dependency purpose. Integrating the TestNG dependency in the POM.xml file and running this POM.xml file using Jenkins.

**Jenkins:** By using Jenkins CI (Continuous Integration) Tool, we execute test cases on a daily basis and also for nightly execution based on the schedule. Test Results will be sent to the peers using Jenkins.

**Extent Reports:** For the reporting purpose, we are using Extent Reports. It generates beautiful HTML reports. We use the extent reports for maintaining logs and also to include the screenshots of failed test cases in the Extent Report.

API Interview questions

**1. What is an API?**

API is an acronym and it stands for **A**pplication **P**rogramming **I**nterface. API is a set of routines, protocols, and tools for building Software Applications. APIs specify how one software program should interact with other software programs.

In simple words, API stands for **A**pplication **P**rogramming **I**nterface. API acts as an interface between two software applications and allows the two software applications to communicate with each other. API is a collection of software functions that can be executed by another software program.

**2. What is API Testing?**

API testing is a type of [software testing](https://www.softwaretestingmaterial.com/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 the Business logic layer of the [software architecture](https://www.softwaretestingmaterial.com/software-architecture/). API testing can be performed on any software system which contains multiple APIs.

**3. What are the common API Testing Types?**

API testing typically involves the following practices:

* Unit testing
* Functional testing
* Load testing
* Runtime/ Error Detection
* Security testing
* UI testing
* Interoperability and WS Compliance testing
* Penetration testing
* Fuzz testing

**4. 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

**5. Difference between API and Web services?**

**Web services:**

1. All web services are APIs  
2. All web services need to be exposed over web(HTTP)  
3. A Web service uses only three styles of use: SOAP, REST and XML-RPC for communication  
4. A Web service always needs a network to operate

**APIs:**

1. All APIs are not web services  
2. All APIs need not be exposed over web(i.e. HTTP)  
3. API uses multiple ways for communication e.g. DLL files in C/C++, Jar files/ RMI in java, Interrupts in Linux kernel API etc.  
4. APIs don’t need a network for operation

**6. What is Soap?**

SOAP stands for Simple Object Access Protocol. It is an XML based messaging protocol. It helps in exchanging information among computers.

**7. What is Rest API?**

REST stands for Representational State Transfer. It is a set of functions helping developers in performing requests and receive responses. Interaction is made through HTTP Protocol in REST API.

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

**SOAP:**  
1. SOAP is a protocol through which two computers communicate by sharing XML document  
2. SOAP supports only XML format  
3. SOAP does not support caching  
4. SOAP is slower than REST  
5. SOAP is like a custom desktop application, closely connected to the server  
6. SOAP runs on HTTP but envelopes the message

**REST:**  
1. REST is a service architecture and design for network-based software architecture  
2. REST supports different data formats  
3. REST supports caching  
4. REST is faster than SOAP  
5. REST client is just like a browser and uses standard methods An application has to fit inside it  
6. REST uses the HTTP headers to hold meta information

**9. What are the common tests that are performed on APIs?**

Some of the common tests we perform on APIs are as follows.

1. Verify whether the return value is based on the input condition. The response of the APIs should be verified based on the request.  
2. Verify whether the system is authenticating the outcome when the API is updating any data structure  
3. Verify whether the API triggers some other event or request another API  
4. Verify the behavior of the API when there is no return value

**10. 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 GUI Testing. So finding those bugs in the 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.

**11. 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 analyze the response.  
1. Data accuracy  
2. HTTP status codes  
3. Response time  
3. Error codes in case API return any errors  
4. Authorization checks  
5. Non-functional testing such as performance testing, security testing

**12. Name some tools used for API Testing?**

Some of the tools used for API Testing are as follows:

* [Postman](https://www.getpostman.com/)
* [Katalon Studio](https://www.katalon.com/)
* [SoapUI](https://www.soapui.org/)
* [Assertible](https://assertible.com/)
* [Tricentis Tosca](https://www.tricentis.com/software-testing-tools/)
* [Apigee](https://apigee.com/)
* [JMeter](https://jmeter.apache.org/)
* [Rest-Assured](http://rest-assured.io/)
* [Karate DSL](https://github.com/intuit/karate)
* [API Fortress](http://apifortress.com/)
* [Parasoft](https://www.parasoft.com/)
* [HP QTP(UFT)](https://software.microfocus.com/)
* [vREST](https://vrest.io/)
* [Airborne](https://github.com/brooklynDev/airborne)
* [API Science](https://www.apiscience.com/)
* [APIary Inspector](https://help.apiary.io/tools/api-inspector/)
* [Citrus Framework](https://citrusframework.org/)
* [Hippie-Swagger](https://github.com/CacheControl/hippie-swagger)
* [HttpMaster Express](https://www.httpmaster.net/)
* [Mockbin](http://mockbin.org/)
* [Ping API](https://ping-api.com/)
* [Pyresttest](https://github.com/svanoort/pyresttest)
* [Rest Console](https://github.com/ahmadnassri/restconsole)
* [RoboHydra Server](http://robohydra.org/)
* [SOAP Sonar](http://www.crosschecknet.com/products/soapsonar.php)
* [Unirest](https://www.npmjs.com/package/unirest)
* [WebInject](http://www.webinject.org/)

**13. 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.

**14. Name some of the API examples which are quite popular.**

Some of the popular API examples are

* Google Maps API
* YouTube
* Twitter
* Amazon Advertising API

**15. Difference between API testing and Unit Testing?**

**UNIT TESTING:**

* Unit testing is conducted by the 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

**16. 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 the absence of GUI, it is quite difficult to provide input values

**17. 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

**18. How is UI testing is not similar to API testing?**

UI (User Interface) testing is to test the graphical interface part of the application. Its main focus is to test the look and feel of an application. On the other hand, API testing enables the communication between two different software systems. Its main focus is in the business layer of the application.

**19. Name some most commonly used HTTP methods?**

Some of the HTTP methods are

**GET:** It enables you to retrieve data from a server  
**POST:** It enables you to add data to an existing file or resource in a server  
**PUT:** It lets you replace an existing file or resource in a server  
**DELETE:** It lets you delete data from a server  
**PATCH:** It is used to apply partial modifications to a resource  
**OPTIONS:** It is used to describe the communication options for the target resource  
**HEAD:** It asks for a response identical to that of a GET request, but without the response body

**20. Can you use GET request instead of PUT to create a resource?**

No, GET request only allows read only rights. It enables you to retrieve data from a server but not create a resource. PUT or POST methods should be used to create a resource.

**21. What is the difference between PUT and POST methods?**

PUT and POST methods are sometimes confused in regards to when each should be used. 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