**1.What are Pre-requisites to design a Framework ?**

***https://xbosoft.com/software-quality-blog/test-automation-framework-requirements/***

***https://www.infostretch.com/blog/choosing-the-right-automation-framework/***

***https://www.glowtouch.com/how-to-choose-the-right-test-automation-framework/***

[***https://www.quora.com/What-are-some-of-the-points-to-be-considered-while-developing-a-test-automation-framework***](https://www.quora.com/What-are-some-of-the-points-to-be-considered-while-developing-a-test-automation-framework)

**----+**

1) Code must be re-usable and easy to maintain.

2) We need to design the framework in such way that, if there are any major or minor changes

in the UI or application, sometimes it may result in change in the existing process flow of

application or it may change the Object properties, those changes should be incorporated

quickly.

3) Test data should be kept distinct from the code and the code should be reusable for different

sets of input data.

3a) Object repository should be kept outside the code

4) Appropriate error handling process must be included as part of the automation framework.

5) Those error should be captured in the HTML report, so that we know exactly which line of

Code is the problem.

6) The automation framework must be written in such a way that if one test script fails should

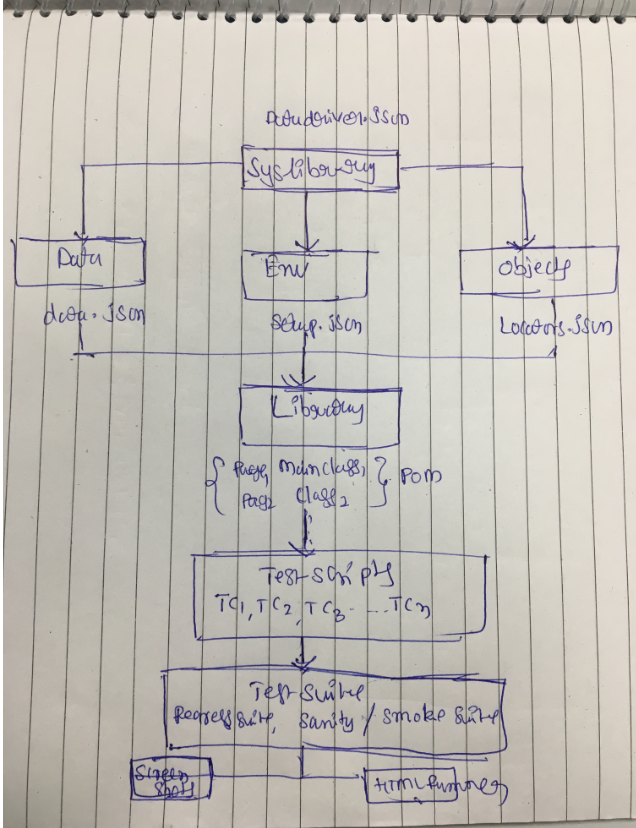
not stop the execution process. Ideally, the framework should report the errors and move on to

execute the next test case. We have to make each test case is a separate script.

**Architecture of Unit test Frame work**

The unit testing framework of Python is known as “unittest”.

It supports the sharing of setups, automation testing, shutdown code for tests, and aggregation of tests into collections, among others



**Programming Language :**

We are using Python Programming language in our project, even though selenium supports multiple languages since huge communities/groups available in net )

**Type of Framework,**

We are using Unit test Framework using with page object model

As per Page Object Model we have maintained class for every web page, each web page separate class and that holds the functionality and members of that web page,

We have separate classes for every individual test.

We have separate packages for pages & tests

**Base Class:**

It contains all the common functions used by all pages, this class responsible for loading configuration from property files/ setup.json file

Initializing web driver, implicit wait, extent reports, logs, and also creating objects of file input stream to read data from /Test data.json

**Page Object Design Model, Functions (i.e. Utils),**

Repetitive tasks will be created as functions such as (i.e. Waits, Actions, Capture screen shots, Accessing Test data and sending emails) and will use entire framework

Achieving re usability

**Setup.json**

It stores the information that reminds static throughout the framework such as (i.e. browser specific info, application url and screen shot path etc.,)

**Unittest/Pytest,**

We use unit test annotation for assertion, grouping and parallel execution etc.,

**Parameterization (i.e. Json /Excel files),**

All the historical test data will kept in side json using Apache POI we handle data driven testing.

**Error Screenshots, Sending Emails,**

**Version Control (i.e. Git)**

We use Git to store our code repository

**Continuous Integration (i.e. Jenkins)**

By using Jenkins we execute test cases on daily basis and also nightly execution based on the schedule,

Results/Reports will be send to every stack holders/peers

**HTML Test Runner Reports:**

For the reporting purpose, we are using HTML Test Runner Reports.

It generates beautiful HTML reports. We use the HTML reports for maintaining logs and also to include the screenshots of failed test cases in the HTML Report.

Unit test is Library known as pyunit

**What is POM?**

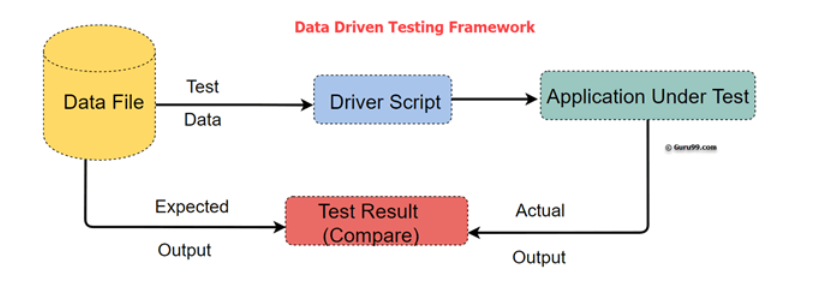
Page Object Model is a design pattern to create Object Repository for web UI elements.

Under this model, for each web page in the application, there should be corresponding page class.

This Page class will find the Web Elements of that web page and also contains Page methods which perform operations on those Web Elements.

Name of these methods should be given as per the task they are performing, i.e., if a loader is waiting for the payment gateway to appear, POM method name can be waitForPaymentScreenDisplay ().

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Python Frameworks \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



**Headless Browser Advantages :**

1

Headless browsers can do all the things you mentioned. The main discussion around headless browser these days is based around speed versus accuracy.

Some strengths of headless browsers:

* Able to run far more instances simultaneously than non-headless drivers.
* Can make use of large amounts of factory-generated or manually created test variables in Data-Driven Testing
* Run-time can be reduced by up to 50% for most tasks.
* Can be executed without taking up the screen context of a computer.

Some weaknesses:

* Hard to debug inconsistent failures on locating elements due to page loading too fast
* Unintended interactions (losing the benefit of automated UI testing vs integration or unit testing) due to speed/headless state of browser
* Code for non-headless drivers will not always work when driver is switched to htmlunit. I.e. switching between ChromeDriver and FirefoxDriver is usually pretty consistent in success rate with same code, but switching to HtmlUnit might need extra tending to.

Tradeoffs:

* Speed vs Consistency - Higher failure rate not as big of a deal with screenshots on fail or tools like saucelabs which record entire runs.
* More specific code vs more general code - Some pages/elements will need specific waits and tailoring in headless browsers, whereas it is pretty straightforward to code for nonheadless

1.What is Data Driven Framework / Explain the Architecture

In data-driven testing approach, we can use a single test to verify different sets of test cases or test data by driving the test with input and expected values from an external data source instead of using the hard-coded values every time a test is run.

Why it is Required?

Its very useful when we have similar tests that consist of the same steps but differ in the input data and expected value or the application state.

PS. – Assuming that you have [unittest,selenium and nose installed](https://scrolltest.com/selenium-testcase-with-nose-in-python/" \t "_blank).

We need to install [DDT in python](https://scrolltest.com/7-minutes-to-learn-python-programming-right-now/)to work with Data driven Testing

## ****Installing DDT****

pip install ddt

## ****Steps to Create the DDT using unit test , nose with external****[data sources(Excel)](https://scrolltest.com/working-with-excel-files-in-python/)

1. Use the @ddt decorator for the test class.  
2.  Use the @data decorator on the data-driven test methods.  
3.  Use the @unpack decorator, which unpacks tuples or lists into multiple arguments.

**Example:**

For example, we want to test the login system with multiple input fields with 1000 different data sets. To test this, you can take following different approaches:

**Approach 1)** Create 1000 scripts one for each dataset and runs each test separately one by one.

**Approach 2)** Manually change the value in the test script and run it several times.

**Approach 3)** Import the data from the excel sheet. Fetch test data from excel rows one by one and execute the script.

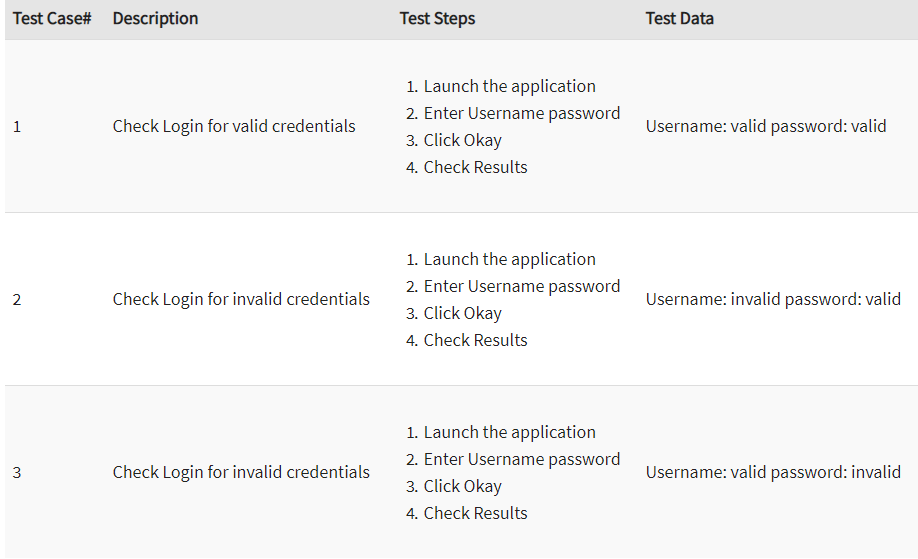
In the given three scenarios first two are laborious and time-consuming. Therefore, it is ideal to follow the third approach.

Consider you want to Test Login functionality of an application.

**Step 1)** Identify the Test Cases

* Input Correct username and password – Login Success
* Input incorrect username and correct password – Login Failure
* Input correct username and incorrect password - Login Failure

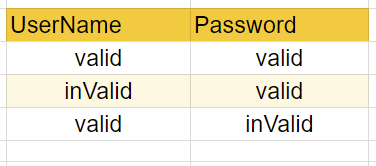
**Step 2)** Create detailed test Steps for above 3 Test Cases



**Step 3)** Create Test Script

If you observe the Test Steps Remain common through the 3 Test Steps. You need to create a Test Script to execute these steps

**Step 4)** Create an excel/csv with the Input Test Data

[](https://www.guru99.com/images/1/032318_1019_WhatisDataD3.png)

**Step 5)** Step Modify the Scrip to Loop over Input Test Data. The input commands should also be parameterized

The test script can be used to loop over following test cases just by appending test data values to Excel

* Input incorrect username and incorrect password – Login Fail
* Input correct username and password blank – Login Fail
* Input blank username and blank password– Login Fail

## Best practices of Data Driven testing:

Below given are Best testing practices for Data-Driven testing:

* It is ideal to use realistic information during the data-driven testing process
* Test flow navigation should be coded inside the test script
* Drive virtual APIs with meaningful data
* Use Data to Drive Dynamic Assertions
* Test positive as well as negative outcomes
* Repurpose Data Driven Functional Tests for Security and Performance

## Advantages of Data-Driven testing

Data-Driven offer many advantages some of them are:

1. Allows to test application with multiple sets of data values during Regression testing
2. Test data and verification data can be organized in just one file, and it is separate from the test case logic.
3. Base on the tool, it is possible to have the test scripts in a single repository. This makes the texts easy to understand, maintain and manage.
4. Actions and Functions can be reused in different tests.
5. Some tools generate test data automatically. This is useful when large volumes of random test data are necessary, which helps to save the time.
6. Data-driven testing can perform any phase of the development. A data-driven test cares are generally merged in the single process. However, it can be used in multiple test cases.
7. Allows developers and testers to have clear separation for the logic of their test cases/scripts from the test data.
8. The same test cases can be executed several times which helps to reduce test case and scripts.
9. Any changes in the test script do not effect the test data

## Disadvantages of Data Driven testing:

Some Drawbacks of Data Driven Automation Testing method are:

1. Quality of the test is depended on the automation skills of the Implementing team
2. Data validation is a time-consuming task when testing large amount of data.
3. Maintenance is a big issue as large amount of coding needed for Data-Driven testing.
4. High-level technical skills are required. A tester may have to learn an entirely new scripting language.
5. There will be more documentation. Mostly related to scripts management tests infrastructure and testing results.
6. A text editor like Notepad is required to create and maintain data files.

**Testing Activities:**

Test Planning

Test Implementation

Maintainability, Reliability, Flexibility, Efficiency, Portability, Robustness, and Usability - these are the main attributes in test automation

Reliability is the degree to which an assessment tool produces stable and consistent results.

**What is Test Case?**

A TEST CASE is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly.

Q #1) What is Automation?

Automation is any action which can reduce human efforts.

Q #2) What all things can you automate?

Regression test suite

Smoke / Sanity test suite

Build deployment

Test data creation

Automating behind the GUI like testing of APIs and methods

Q #3) How do you identify the test cases which are suitable for automation?

Identify the appropriate test cases for automation is the most important step towards automation.

Q #4) Can you achieve 100% automation?

100% automation would be difficult to achieve because there would be many edge test cases and some cases which are executed seldom. Automating these cases which are not executed that often will not add value to the automated suite.

Q #5) Currently I do not have any automation in place in my project, now I want to implement automation, what would be my steps?

First, identify which type of testing/test cases you want to automate

Identify the tool

Design the framework

Create the utility files and environment files

Start scripting

Identify and work on the reporting

Allocating time for enhancing and maintaining the scripts.

Q #6) How do you decide which tool you have to use?

Concluding which tool is best suitable for the project requires a lot of brainstorming and discussions.

Q #7) Once you identify the tool what would be your next steps?

Once we finalize the tool, our next step would be to design the framework.

Q #8) What is a framework?

A framework is a set of a structure of the entire automation suite. It is also a guideline if followed can result in a structure which is easy to maintain and enhance. These guidelines include:

Coding standards

Handling the test data

Maintaining and handling the elements (object repository in QTP)

Handling of environment files and properties file

Reporting of data

Handling logs

Q #9) What are the attributes of a good framework?

The characteristics are:

Modular – The framework should be adaptable to change. Testers should be able to modify the scripts as per the environment or login information change

Reusable – The commonly used methods or utilities should be written in a common file which is accessible to all the scripts.

Consistent – The suite should be written in a consistent format by following all the accepted coding practices.

Independent – The scripts should be written in such a way that they are independent of each other. In case one test fails, it should not hold back remaining test cases (unless it is a login page)

Logger – It is good to have implemented the logging feature in the framework. This would help in case our scripts run for longer hours (say nightly mode), if the script fails at any point of time, having the log file will help us to detect the location and the type of error.

Reporting – It is good to have reporting feature automatically embedded into the framework. Once the scripting is done, we can have the results and reports sent via an email.

Integration – Automation framework should be such that it is easy to integrate it with other application like continuous integration or triggering the automated script as soon as the build is deployed.

Q #10) Can you do without a framework?

Frameworks are guidelines and not mandatory rules, so we can do without a framework, but if we create it and follow it, enhancing and maintaining would be easy to implement.

Q #11) What are the different types of automation tool you are aware of?

Open source tool like Selenium, JMeter

Paid tools like QTP, Load Runner, Ranorex, RFT, and Rational Robot.

Q #12) What generally is the structure of a framework?

Normally the structure should have – (It would differ from project to project)

A “src” (source) folder having the actual test scripts

A”lib” (library) folder having all the libraries and common methods

A “class” folder having all the class file (in-case using java)

A “log” folder having the log file(s)

A file/folder having all the web element Ids

A file containing the URL, environment and login information.

Q #13) Where you maintain information like URL, login, password?

This information should always be maintained in a separate file.

Q #14) Why do you want to keep this kind of information in a separate file and not directly in code?

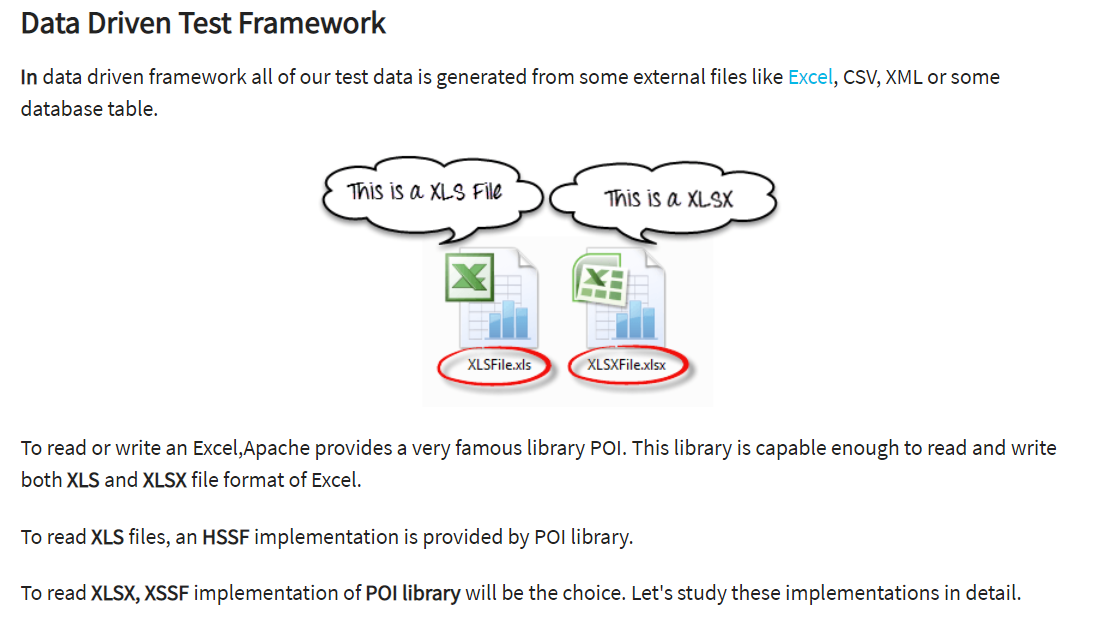
URL, Login, and passwords are the kind of fields which are used very often and these changes as per the environment and authorization. In case we hardcode it into our code, we have to change it in every file which has its reference. In case there are say more than 100 files, then it becomes very difficult to change in all the 100 files and hence can lead to errors. So this kind of information is maintained in a separate file so that updating becomes easy.

Q #15) What are the different types of frameworks?

Different types of framework available are:

Keyword driven framework

Data Driven framework



Hybrid Framework

Linear Scripting

Q #16) Can you tell some good coding practices while automation?

Some of the good coding practices include:

Add appropriate comments

Identify the reusable methods and write it in a separate file

Follow the language-specific coding conventions

Maintain the test data in a separate file

Run your scripts regularly

Q #17) Any kind of test which you think should not be automated?

Tests which are seldom executed

Exploratory testing

Usability testing

Test which is executed fairly quickly when done manually

Q #18) Do you think that testing can be done only at the UI level?

Today as we are moving to Agile mode, testing is not limited to the UI layer. Early feedback is imperial for an agile project. If we concentrate only on the UI layer, we are actually waiting until the UI is developed and available to test. Rather we can test even before the UI is actually developed. We can directly test the APIs or the methods using tools like Cucumber and FitNesse.

In this way, we are giving the feedback much early and even are testing before the UI is developed. Following this approach will help us to test only the GUI aspect of small cosmetic changes or some validations on the UI and will help the developers by giving more time to fix the bugs.

Q #19) How do you select which automation tool is best suited for you?

Selecting the automation tool depends upon various factors like:

The scope of the application which we want to automate

Management overhead like cost and budget

Time to learn and implement the tool

Type of support available for the tool.

Limitation of the tool

Q #20) What do you think holds testers back to do automation? Is there a way to overcome it?

The major hurdle for testers is to learn programming/coding when they want to automate. Since testers do not code, adapting to coding is a bit challenging for testers. We can overcome it by:

Collaborating with developers when automating

Considering that automation is the responsibility of the whole team and not only of the testers

Giving a dedicated time and focus on automation.

Getting proper management support.

You can save these automation testing interview questions as pdf and print for further reading.

Conclusion:

Most of the test automation interview questions are centered on the framework you develop, so it is recommended that you create and understand your test framework thoroughly. When I am interviewing, and the candidate has answered my question on the framework, I also prefer asking language specific question (core java in my case).

The questions start from basics of java to write the logic of some basic scenario like –

How would you extract a set of text from a given line?

How would you extract URL?

In any web page, at any frame, the number of links and its content change dynamically, how would you handle it?

How do you handle images and flash objects?

How do you find a word in a line?

Answers to all these test automation interview questions are very much specific to the tool/language you are using for automating. So before you go for the interview, brush up your programming skills.

In case you did not get a chance to create your framework and someone else have created it, make some time to understand it thoroughly before sitting for the interview.

**When will you automate a test?**

Automation in preferred in following cases

Repetitive Tasks

Smoke and Sanity Tests

Test with multiple data set

Regression test cases

**On what basis you can map the success of automation testing?**

By following criteria, the success of automation testing can be mapped

1) Defect Detection Ratio

2) Automation execution time and time savings to release the product

3) Reduction in Labour & other costs

**Type of Objects in a programming language:**

1. Attributes – color, gender, Nationality etc…

2. Actions-Walk, see, smell etc…