What is Data Driven Testing? Learn to create Framework

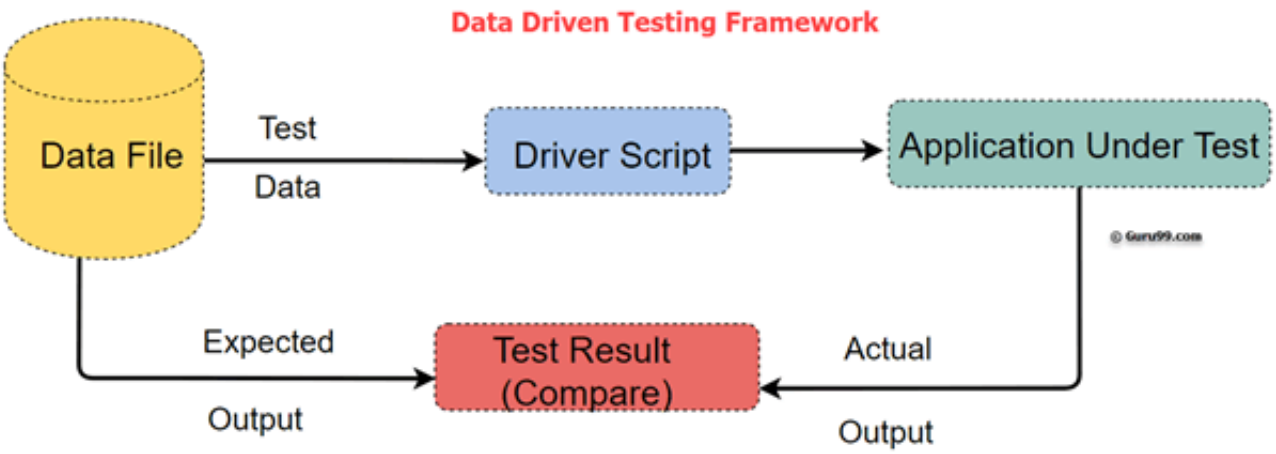
Thomas Hamilton November 12, 2022

Data Driven Testing

Data Driven Testing is a software testing method in which test data is stored in table or spreadsheet format. Data driven testing allows testers to input a single test script that can execute tests for all test data from a table and expect the test output in the same table. It is also called table-driven testing or parameterized testing.

Data Driven Framework

Data Driven Framework is an automation testing framework in which input values are read from data files and stored into variables in test scripts. It enables testers to build both positive and negative test cases into a single test. Input data for multiple data sources like .xls, .xml, .csv and databases.



In this tutorial, you will learn

Why Data Driven Testing?

Data Driven Testing is important because testers frequently have multiple data sets for a single test and creating individual tests for each data set can be time-consuming. Data driven testing helps keeping data separate from test scripts and the same test scripts can be executed for different combinations of input test data and test results can be generated efficiently.

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.

Thus, the third approach is nothing but a Data-Driven framework.

How to create a Data Driven Automation Framework

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

Test Case#	Description	Test Steps	Test Data	Expected Results
1	Check Login for valid credentials	1. Launch the application 2. Enter Username password	Username: valid password: valid	Login Success

		3. Click Okay 4. Check Results		
2	Check Login for invalid credentials	1. Launch the application 2. Enter Username password 3. Click Okay 4. Check Results	Username: invalid password: valid	Login Fail
3	Check Login for invalid credentials	1. Launch the application 2. Enter Username password 3. Click Okay 4. Check Results	Username: valid password: invalid	Login Fail

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

```
// This is Pseudo Code
```

```
// Test Step 1: Launch Application
driver.get("URL of the Application");

// Test Step 2: Enter Username
txtbox_username.sendKeys("valid");

// Test Step 3: Enter Password
txtbox_password.sendKeys("invalid");

// Test Step 4: Check Results
If (Next Screen) print success else Fail
```

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

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

```
// This is Pseudo Code
// Loop 3 Times
for (i = 0; i & lt; = 3; i++) {
    // Read data from Excel and store into variables
    int input_1 = ReadExcel(i, 0);
    int input_2 = ReadExcel(i, 1);

    // Test Step 1: Launch Application
    driver.get("URL of the Application");
```

```
    // Test Step 2: Enter Username
    txtbox_username.sendKeys(input_1);
    // Test Step 3: Enter Password

    txtbox_password.sendKeys(input_2);
    // Test Step 4: Check Results
    If(Next Screen) print success
    else Fail
}
```

Above are just 3 test cases. 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

And so on

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.

Conclusion:

- Data-driven is a test automation framework which stores test data in a table or spread spreadsheet format.
- In Data-driven test automation framework, input data can be stored in single or multiple data sources like xls, XML, csv, and databases.
- To create an individual test for each data set is a lengthy and time-consuming process. Data Driven Testing framework resolves this issue by keeping the data separate from Functional tests.
- In Data Driven Testing, it is an ideal option to use realistic information
- It allows testing application with multiple sets of data values during Regression testing
- Drawback of this method is that it is depended on the automation skills of the Implementing team

You Might Like:

- **What is Software Testing? Definition**
- **50 BEST Software Testing Tools List (Dec 2022 Update)**
- **What is Thread Testing in Software Testing?**
- **What is Loop Testing? Methodology, Example**
- **Non Destructive Software Testing (NDT): What is, Test Strategy**

← Prev

Report a Bug

Next →

About

About Us

Advertise with Us

Write For Us

Contact Us

Career Suggestion

SAP Career Suggestion Tool

Software Testing as a Career

Interesting

eBook

Blog