

# Chapter 6.3:

## Soft Asserts

### Introduction

In this chapter, we will discuss the Difference Between Hard Asserts & Soft Asserts and answer the question Should Automation Engineers Use Hard and/or Soft Asserts.

### Difference Between Hard Asserts & Soft Asserts

The difference between Hard Asserts and Soft Asserts is a Hard Assert stops executions after a fail and move to the next annotation. It does not matter if the next annotation is a Test Annotation or a Configuration Annotation.

We saw in the previous chapter, steps 3, 4, and 5 did not execute because they were inside the same annotation that failed. Therefore, those steps were skipped and execution picked up at the next Test Method which starts at step 6.

Someone might consider using a try-catch block for the Hard Assert but it won't work either. I'm going to show you. Here's the try-catch block which will try the assertions and catch the AssertionError. Let's Run. Look all of the Test Methods Passed although the AssertionError states The Welcome Link Is Not Correct On The Home Page expected false but found true. Scroll the console. Steps 3, 4, and 5 were skipped again. The Results tab shows all Test Methods Passed. That's not accurate.

A Soft Assert is different from a Hard Assert. It continues execution after a failed assertion and moves to the next statement line. It was designed to keep executing even when a verification step fails.

To use Soft Assert, first, we declare SoftAssert which is a class in TestNG then our object reference softassert equals new SoftAssert. Next, we replace all of the Asserts with our object reference softassert. The assertion methods (assertEquals, assertFalse, and assertTrue) will stay the same.

Recall from the previous video, that the softassert class had only 2 methods: assertAll and doAssert. We need to use assertAll every time for softassert to work. If assertAll is not used then our test will pass and throw no AssertionError. I'm going to run without assertAll then run with assertAll to show you the difference. Run.

All Test Methods Passed and we know Steps 3 and 5 did not Pass. SoftAssert kept executing although the verification step Failed. Now, let's add assertAll to the end of the Test Method: softassert.assertAll and Run. One of our Test Methods Failed which is testHomePageVerification. Do you see both Assertions? The Welcome Link Is Not Correct On The Home Page expected false but found true. The Dashboard Is Not Correct On The Home Page expected true but found false. All 8 steps show up. The Results tab shows 1 Test Method Failed with the same 2 messages. The Welcome Link Is Not Correct On

The Home Page expected false but found true. The Dashboard Is Not Correct On The Home Page expected true but found false. The Welcome Link and Dashboard are not correct on the Home page.

Here's a diagram that shows assertAll. On the left, we see a few assertion methods and assertAll on the right. The arrows are pointing to assertAll because they indicate an AssertionError gets stored into assertAll if there is a failure. That's why we place the assertAll method at the end of our Test Method.

## Hard Asserts vs Soft Asserts

Hard Asserts versus Soft Asserts. Which one should we use for automation? We should use both asserts but it depends on our test. Sometimes it's good to have a hard assertion and other times it's not good to have a hard assertion. The same with soft assertions. Let's consider these print statements.

Open Browser, Open Application, Sign Into The Application, Go To Home Page and Verify Home Page, Go To Search Page and Verify Search Page, Search For User, and Sign Out of the Application. We know Hard Asserts stop if a verification step fails and will not execute the next statement but Soft Assert will execute the next statement.

If we had a failure after opening the browser then we would not want to continue executing the next step. There is no reason. With that scenario, it's best to implement a Hard Assert. The same with opening an application. There is no reason to keep executing our Test Script if it's a failure opening the application. The next step which is Sign Into The Application would return a failure.

However, we would not implement a Hard Assert after Step 4, Go To Home Page and Verify Home Page. If there's a failure on the Home Page, we still want to verify Step 5, Go To Search Page and Verify Search Page. To sum up assertions, we want to implement Hard Asserts in our code where it does not make sense to keep executing our test after a verification failure. We implement Soft Asserts in our code when there is a failure and we want to continue executing our test. Next, we will cover Dependency Testing.