Assert Class in TestNG

In TestNG, the Assert class is a utility that provides a set of methods to validate conditions in your tests. These conditions can determine whether a test passes or fails. The Assert class is typically used in Java Selenium UI automation to validate that the expected results match the actual results.

Key Methods of the Assert Class

1. **assertEquals()**
   - **Usage:** Validates that two values are equal.
   - **Signature:** public static void assertEquals(Object actual, Object expected)
   - **Example:** Assert.assertEquals(actualTitle, expectedTitle);
2. **assertNotEquals()**
   - **Usage:** Validates that two values are not equal.
   - **Signature:** public static void assertNotEquals(Object actual, Object expected)
   - **Example:** Assert.assertNotEquals(actualPrice, incorrectPrice);
3. **assertTrue()**
   - **Usage:** Validates that a condition is true.
   - **Signature:** public static void assertTrue(boolean condition)
   - **Example:** Assert.assertTrue(isElementDisplayed);
4. **assertFalse()**
   - **Usage:** Validates that a condition is false.
   - **Signature:** public static void assertFalse(boolean condition)
   - **Example:** Assert.assertFalse(isErrorDisplayed);
5. **assertNull()**
   - **Usage:** Validates that an object is null.
   - **Signature:** public static void assertNull(Object object)
   - **Example:** Assert.assertNull(deletedElement);
6. **assertNotNull()**
   - **Usage:** Validates that an object is not null.
   - **Signature:** public static void assertNotNull(Object object)

    - **Example:** Assert.assertNotNull(activeSession);

7. **fail()**
    - **Usage:** Forces a test to fail.
    - **Signature:** public static void fail(String message)
    - **Example:** Assert.fail("Test case failed intentionally");

Soft Assert vs Hard Assert

- **Hard Assert:**
    - **Definition:** A Hard Assert in TestNG is an assertion that stops test execution immediately when the assertion fails. If a hard assert fails, the subsequent steps or code in that particular test case are not executed.
    - **Usage:** Suitable when a failure in the assertion means the rest of the test should not be executed.
- **Soft Assert:**
    - **Definition:** A Soft Assert in TestNG allows the test to continue execution even after an assertion fails. All failures are recorded, and the test does not stop. After all assertions are done, you need to call softAssert.assertAll() to check if any assertions have failed.
    - **Usage:** Useful when you want to continue executing the test case and collect all assertion failures at the end.

FedEx Ecommerce Application Testing Examples

Let's consider five different scenarios where you might apply these assertions in the context of testing the FedEx Ecommerce application.

1. Validating Page Title (Hard Assert with assertEquals)

```
@Test
public void validateHomePageTitle() {
    WebDriver driver = new ChromeDriver();
    driver.get("https://www.fedex.com/");

    String actualTitle = driver.getTitle();
    String expectedTitle = "FedEx | Tracking, Shipping, and Locations";
    Assert.assertEquals(actualTitle, expectedTitle, "Home page title does not match!");

    driver.quit();
}
```

2. Validating Error Message (Soft Assert with assertTrue)

```
@Test
public void validateLoginErrorMessage() {
    WebDriver driver = new ChromeDriver();
    driver.get("https://www.fedex.com/login");

    driver.findElement(By.id("loginButton")).click();

    SoftAssert softAssert = new SoftAssert();

    boolean isErrorDisplayed = driver.findElement(By.id("errorMessage")).isDisplayed();
    softAssert.assertTrue(isErrorDisplayed, "Error message should be displayed when no
credentials are provided.");

    String actualErrorMsg = driver.findElement(By.id("errorMessage")).getText();
    String expectedErrorMsg = "Please enter your credentials.";
    softAssert.assertEquals(actualErrorMsg, expectedErrorMsg, "Error message text does not
match!");

    softAssert.assertAll();

    driver.quit();
}
```

3. Validating Dropdown Values (Hard Assert with assertNotNull)

```
@Test
```

```
public void validateShippingDropdownValues() {
    WebDriver driver = new ChromeDriver();
    driver.get("https://www.fedex.com/shipping");

    WebElement dropdown = driver.findElement(By.id("shippingOptions"));
    Select select = new Select(dropdown);

    List<WebElement> options = select.getOptions();
    Assert.assertNotNull(options, "Dropdown options should not be null!");

    Assert.assertEquals(options.size(), 5, "Shipping dropdown should have 5 options.");

    driver.quit();
}
```

4. Validating Element Absence (Soft Assert with assertFalse)

```
@Test
public void validateTrackingNumberFieldAbsence() {
    WebDriver driver = new ChromeDriver();
    driver.get("https://www.fedex.com/shipping");

    SoftAssert softAssert = new SoftAssert();

    boolean isTrackingFieldPresent = driver.findElements(By.id("trackingNumber")).size() > 0;
    softAssert.assertFalse(isTrackingFieldPresent, "Tracking Number field should not be present
in the Shipping page.");

    softAssert.assertAll();

    driver.quit();
}
```

5. Forcing Test Failure (Using fail() method)

```
@Test
public void forceFailureForDebugging() {
    WebDriver driver = new ChromeDriver();
    driver.get("https://www.fedex.com/");

    try {
        // Simulate some condition that should cause the test to fail
        if (driver.findElement(By.id("nonExistentElement")).isDisplayed()) {
            Assert.fail("This element should not be present, failing the test.");
        }
    } catch (NoSuchElementException e) {
        System.out.println("Element not found, test continues...");
    } finally {
        driver.quit();
    }
}
```

Summary

- **Assert Class**: Provides essential methods to validate test conditions, ensuring that the expected outcomes align with the actual results.
- **Hard Assert**: Stops the test execution immediately upon failure.
- **Soft Assert**: Allows the test to continue even if some assertions fail, collecting failures to be reported at the end.
- **FedEx Ecommerce Examples**: Showcased how to apply these concepts in real-world scenarios, including validating page titles, error messages, dropdown options, element absence, and forced failures.

This detailed explanation should provide you with a solid understanding of how to use assertions effectively in your UI automation tests using TestNG and Java Selenium, specifically within the context of the FedEx Ecommerce application.