



Assessor+ Generator Troubleshooting



- **Missing command from PageObject method**
 - **Description:** a command recorded with Selenium IDE is not transformed into code (e.g., mouseOver)
 - **Solution:** follow the logic of the target method w. r. t. the app functionality and manually add the missing command
- **Wrong text within sendKeys command**
 - **Description:** the text of a sendKeys command is weird (e.g., it contains <tags>) or different from the expected one
 - **Solution:** check the value passed to the associated PageObject method and manually change it
- **Wrong method signature**
 - **Description:** the signature of a PageObject method is wrong (e.g., it presents too many parameters, some of them null)
 - **Solution:** check how these parameters are used within the method, in particular those regarding sendKeys commands, that may be assigned to null values, and change their values properly





- **Wrong locator**

- **Description:** the locator used to find an element does not exist or does not work
- **Solution:** Since some elements may be visible but interactively hidden by other elements, the locator might need to be changed, even if it worked in Selenium IDE. Manually replace the locator by inspecting the page

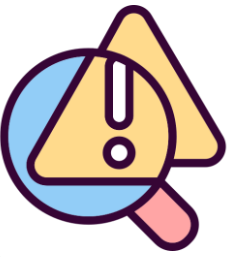
- **Slow test processing**

- **Description:** the test is logically correct, but some functionalities are slow to process (e.g., add a new user in a database), then the expected element is not shown in time and the test fails
- **Solution:** Manually add a `Thread.sleep(milliseconds)` before the failing instruction

- **Useless waitForElementLoaded command**

- **Description:** the test waits for an element that is not present or which is not needed to wait
- **Solution:** check if the element to load should exist in the page, by inspecting it. If not, or if it is not needed, remove the `WaitForElementLoaded` and any other reference to that element





- **Wrong base URL**

- **Description:** the test opens a page different from the expected one
- **Solution:** check the URL associated with the command to open the browser within the failing test and fix it

- **A test fail at the end of correct execution**

- **Problem:** the test is completed correctly but fails at the end of the execution
- **Solution:** This is likely due to a `driver.close()` instruction placed at the end of the test. Remove that instruction, since it is automatically performed at the end of each test, otherwise the test will try to close the browser instance twice

- **Assert fail**

- **Problem:** an assert fails although the test is correct
- **Solution:** check that the element being asserted exists and that the expected text exactly matches the actual one, correcting any tabulation characters if needed (e.g., new line)

