Selenium Test Recording Best Practices

ℹ **Selenium Test Recording Best Practices**
[A test MUST...] [verify] [be maintainable] [be independent] [leave a clean test-environment] [be sturdy] [run on all browsers]

see Selenium IDE Guide for Testers for specific Selenium Best Practices

A test MUST...

verify

- A test is as good as its design
- A test should be constantly improved
- A test should verify expected results as well as process execution
- A test should test a specific function, feature or defect

How ?

- Try to find new tests or new verifications to complete tests
- Use defects (not discovered by Selenium tests) to add new verifications
- Each action specified in the test description should be followed by a *verify**
- Create short tests (> 20 commands)

be maintainable

- A test is to be writen before/during dev.
- A test shouldn't need important maintenance for slight application changes.
- A test should be clearly structured and commented to be maintainable by all.

How ?

- Testers need participate earlier in the process to define needs/impacts and assess maintenance cost.
- Carefully choose locators and verification elements.
- One action = One comment (echo command in selenium)
- TODO : Use typical script template

be independent

- A test case is can be run by itself (**isolated**)
- A test is unaffected by the presence, absence, or results of other tests
- A test suite can be executed in any order (**order-independent**)

How ?

- Start all tests at the same location : initial state (i.e. "Home" logged-out for Portal)
- Clean data
- End the test where it started

leave a clean test-environment

- A test should be able to be executed as many times as needed
- Problems should be easy to analyze : Analyze differences with initial state

How ?

- All test-data creation and modification is to be erased by the test to enable new test-data.
- A failed test generally leaves test-data. Clean test-environment eases analysis.
- Use normalized names for test-data. example : "test-32-NewPortal". When analyzing, it is easier to find which test the element is from.

be sturdy

- A test should not be blocked by a tooling inefficiencies
- A test should not be impacted by low network response-rates or random causes

- A test should not depend on the test environment (unless specified)

How ?

- Run tests in converted java to verify correct implementation and report problems
- Once a test passes or detects the targeted defect, improve speed with *waitFor** and **consider it 'Done' if the test can be run 5 times full speed**.
- Avoid verifications depending on browsers, databases, OS...

run on all browsers

- A test should run successfully on multiple browsers.

How ?

- Run scripts in all browsers before
- To run in maven cmd you need to add parameters "-Dselenium.browser = ..."