Test Automation Mentoring Program: Advanced

TEST RUNNERS. HOME TASK

Tech stack: .NET, Java, JavaScript

TABLE OF CONTENTS

INTRODUCTION

You will be working with Report Portal application deployed locally on your workstation during all the mentoring program.

The scope of this task is to work with alternative test runner and try to use it's capabilities. The goals of this module are:

- Give a possibility to use the other test runner
- Evaluate different test runners
- Use advanced features of test runner
- Improve Tools Evaluation skill

Work primarily with feature of report portal which was chosen previously:

1. Launches
2. Dashboards
3. Widgets
4. Filters

MARKS

Points	Mark	Description
1	0.5	Unsatisfactory
2	1	Unsatisfactory
3	1.5	Unsatisfactory
4	2	Satisfactorily
5	2.5	Satisfactorily
6	3	Satisfactorily
7	3.5	Good
8	4	Good
9	4.5	Excellent
10	5	Excellent

Home task is DONE if more than (>=) 4 points received.
Mentor can take away points if Mentee's code quality does not meet his/her requirements.

DEADLINE

It's expected that Mentee completes a task in 1 week. The deadline can be adjusted in case of:

1. Vacation
2. Illness
3. Business trip
4. Exceptional case

Before adjusting the deadline please notify and confirm with Coordinator about changes.

PRECONDITION

In order to start working with this task you should generate Demo data in Report portal:

1. Go to Settings -> demo data
2. Specify postfix -> click Generate demo data

You will be working with this data.

TASKS

1. ADD ALTERNATIVE TEST RUNNER TO YOUR PROJECT

Complexity	Points	Necessity
Basic	2	Task is Mandatory

TASK

1. Choose the test runner you haven't been working with/have lower experience/want to try
    a. Java – Junit4/Junit5/TestNG
    b. .NET – MsTest/nUnit/xUnit
    c. JS – Jasmine/Mocha/etc. NOTE – since JS changes pretty fast – please work with ANY tool you would like to.
2. Implement base test infrastructure with preconditions and postconditions (test layer) using newly added runner.

DEFINITION OF DONE

Your project contains references to new test runner.

Your project contains base test infrastructure with preconditions and postconditions implemented with newly added test runner.

2. DATA DRIVEN TESTS

Complexity	Points	Necessity
Intermediate	4	Task is Mandatory

TASK

Here we want you to compare BOTH added test runners. First thing is DDT. To complete this task, you may require implementing some Page objects in Business layer of your framework. Here we are not focused on POs organization, however, try to use best practice here.

Implement several (at least 3) DDT tests which are suitable for your feature.

E.g. for **launches** - test to verify that launch (Click *Launches* on the left menu) contains expected data (TOTAL; PASSED; FAILED; SKIPPED; PRODUCT BUG; AUTO BUG; SYSTEM ISSUE; TO INVESTIGATE). Think for other scenarios in scope of your feature.

Do this for BOTH test runners you've added.

Prepare at least 5 data sets for each implemented test. Design and implement test data storing utility and place it in Core Layer.

DEFINITION OF DONE

Your test layer contains DDT tests. Test Data stored according to chosen strategy. DDT implemented for BOTH test runners.

3. PARALLELIZATION

Complexity	Points	Necessity
Intermediate	3	Task is Mandatory

TASK

Set up your added tests in previous task to run in parallel at different levels of parallelization (implement one or both options):

1. Class level
2. Test level

Also make possible to setup number of threads (if test runner supports this).

Do this for BOTH test runners.

DEFINITION OF DONE

Your tests are able to run in parallel for both test runners.

4. BONUS TASK

Complexity	Points	Necessity
Advanced	1	Task is Optional

Implement parallelization for DDT – so EACH DDT test should be run in several threads, for example if you have 5 data sets in your tests you should be able to utilize 5 threads, so that each data set will use own thread. NOTE – not every test runner supports this, so there’s no requirement to implement this for both test runners.

Ver.	Description of Change	Author	Date
1.0	Initial Version	Yury Karpinski	14-Oct-2020
1.1	Updated: Structure	Yury Karpinski	24-Jan-2020
1.2	Removed soft assertions	Yury Karpinski	27-March-2022
2.0	Merged with BDD	Yury Karpinski	27-July-2022