



Test Automation Mentoring Program: Advanced

UI TESTING. HOME TASK

Tech stack: .NET

Legal Notice: This document contains privileged and/or confidential information and may not be disclosed, distributed or reproduced without the prior written permission of EPAM®.

CONFIDENTIAL

TABLE OF CONTENTS

INTRODUCTION 3

MARKS 3

DEADLINE 3

PRECONDITION 3

TASKS 3

 1. CUSTOM ELEMENTS 3

 TASK..... 4

 DEFINITION OF DONE 4

 2. ACTIONS. JS EXECUTOR 4

 TASK..... 4

 DEFINITION OF DONE 4

 3. IMPLEMENT TESTS..... 4

 TASK..... 4

 DEFINITION OF DONE 4

 4. GRID/SELENOID 5

 TASK..... 5

 DEFINITION OF DONE 5

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 advanced Selenium features and try the other web-testing framework. The goals of this module are:

- Use advanced features of web-testing framework – Pure Selenium.
- Get started with Selenium Grid/Selenoid.
- Try alternative solution.

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 (\geq) 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 2 weeks. 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

1. Mentee has a sample test project implemented.
2. The project contains 2 different test runners – use one that was added in Module #2 – TEST RUNNERS.

TASKS

1. CUSTOM ELEMENTS

Complexity	Points	Necessity
Basic	2	Task is Mandatory

TASK

1. Implement custom elements architecture. Choose the most suitable architectural approach. Add logging for actions with elements.
2. Configure Waits for custom elements (use both explicit and fluent waiters). Also configure implicit wait for WebDriver.

DEFINITION OF DONE

Your project contains custom elements architecture with configured waiters and logging.

Your project contains screenshot taker utility.

2. ACTIONS. JS EXECUTOR

Complexity	Points	Necessity
Basic	2	Task is Mandatory

TASK

1. Implement Drag & Drop, Elements Resize using Actions.
2. Implement below features using Js executor:
 - a. Scroll to element
 - b. Verification if element is scrolled into view
 - c. Click using JS

Choose the most suitable architectural approach.

DEFINITION OF DONE

Your project contains custom elements with implemented JS and Actions.

3. IMPLEMENT TESTS

Complexity	Points	Necessity
Intermediate	3	Task is Mandatory

TASK

Implement tests using advanced features added in tasks #1 and #2.

For instance, if chosen Dashboards feature, you can verify that dashboard widgets can be resized:

- a. Selected size is saved.
- b. The other widgets move while resizing.
- c. The Content of widget resizes as well.

NOTE: Drag & Drop, Elements Resize can be used only in dashboards. So, if you chose any other feature - to work with such functionality you should create appropriate pages. You can limit number of tests by 1 in this case.

DEFINITION OF DONE

Test project contains required tests. All tests work and passing.

4. GRID/SELENIOD

Complexity	Points	Necessity
Intermediate	3	Task is Optional

TASK

Set up the tests to run using either Selenium Grid or Selenium for multiple browsers

DEFINITION OF DONE

Test project is set up to run using Selenium Grid or Selenium.

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 alternative framework	Yury Karpinski	27-March-2022