



Test Automation Mentoring Program: Advanced

TEST AUTOMATION PLANNING. HOME TASK

Tech stack: .NET, Java, JavaScript

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

TASKS 4

 1. DEPLOY LOCAL REPORT PORTAL USING DOCKER 4

 TASK 4

 DEFINITION OF DONE 4

 2. OVERVIEW THE APPLICATION UNDER TEST 4

 TASK 4

 DEFINITION OF DONE 4

 3. ESTIMATIONS 5

 TASK 5

 DEFINITION OF DONE 5

 4. CREATE TEST PLAN (TEST AUTOMATION) AND AUTOMATED TESTING STRATEGY 5

 TASK 6

 DEFINITION OF DONE 6

INTRODUCTION

You will be working with Report Portal application either deployed locally on your workstation or using rp.epam.com during all the mentoring program.

Consider you joined a new project and your application under test is Report Portal. Customer as usual wants firstly to have some basic definitions of your work (let's be focused on test automation). So the main goal and output of these module will be automated testing plan document and estimations.

In order to make it doable we will narrow our focus are and limit report portal to just only features (instead of covering all the application we will do this for just one feature) from the list below:

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

And you will be working further with just this feature (not only in this module but in all modules from this program).

The goals of this module are:

- Overview the Application Under Test.
- Learn how to create test plan/test strategy.
- Learn how to estimate test automation efforts.

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.

TASKS

1. DEPLOY LOCAL REPORT PORTAL USING DOCKER

Complexity	Points	Necessity
Basic	1	Task is Mandatory

TASK

1. Download and install Docker (use Personal free subscription)
2. Set up Report Portal container with Docker ([instruction for Windows](#) / [for Mac](#)).

DEFINITION OF DONE

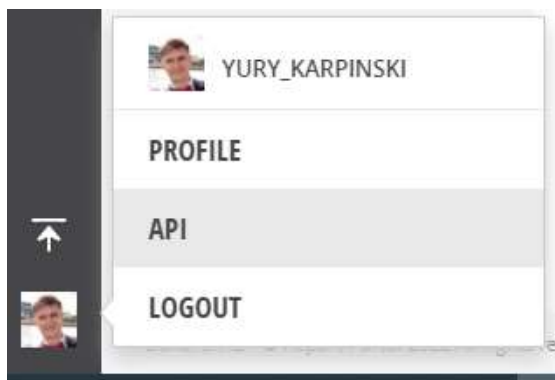
You're able to run Report Portal locally (<https://localhost:8080>)

2. OVERVIEW THE APPLICATION UNDER TEST

Complexity	Points	Necessity
Basic	2	Task is Mandatory

TASK

1. Create your personal project in RP
2. Create I mock data for this project (Settings => Demo Data)
3. Play with U
4. Observe API documentation (Account => API)



IMPORTANT: It's supposed that you will be working with one of application features (areas) chosen at this stage. E.g. if you choose Dashboards – it means that you will be working with them during all the program (estimate, create test plan, create API and UI tests in next modules). Here we define list of features for application:

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

You should choose one of them to work further.

DEFINITION OF DONE

Personal project with mock data created. Feature chosen.

3. ESTIMATIONS

Complexity	Points	Necessity
Intermediate	3	Task is Mandatory

TASK

Overview the feature chosen in previous task (Launches, Dashboards, Widgets, Filters). The goal is to cover this feature by test automation from scratch. Think about testing approaches, user journeys, test cases.

Scope to be estimated:

1. TAF creation.
 - a. Creation of git repository
 - b. Creation of core framework layer (Configuration, test data, logger)
2. API tests creation.
 - a. API part of business layer
 - b. API client integration
 - c. DTOs
 - d. Tests (consider total number of test cases as 10)
3. UI tests creation.
 - a. UI part of business layer
 - b. Elements
 - c. Page objects
 - d. Tests (consider total number of test cases as 10)
4. CI/CD setup.
 - a. Build pipeline
 - b. Test execution pipeline
5. Integration with 3-d party tools
 - a. Integration with project management tool (e.g. Jira, Azure DevOps, Trello)
 - b. Integration with reporting tool (Allure, RP)
 - c. Integration with communication tool (Slack, Teams)
 - d. Integration with test execution tool (Sauce Labs, BrowserStack)

Now estimate your efforts on high level using one of estimation strategies described in the video. Use your teammates or Mentor for planning poker if needed. Estimate in story points (Decide with your team what is story point. Consider by default: 1 SP = 1 man-day).

Provide the result of your work in .docx/.xlsx document which contains work breakdown with estimates.

DEFINITION OF DONE

You have high level estimations.

4. CREATE TEST PLAN (TEST AUTOMATION) AND AUTOMATED TESTING STRATEGY

Complexity	Points	Necessity
Advanced	4	Task is Mandatory

TASK

Again, work with feature chosen in Task#2.

Using EPAM Test plan and test strategy templates create a test plan document (specific for test automation) which describe how would you test the feature. Areas to be in scope of which should be in scope:

1. TAF. Think about tools to be used, framework layers, approaches of working with test data
2. API tests. Think about tools to be used, testing approaches
3. UI tests. Think about tools to be used, testing approaches
4. CI/CD. Think about tools to be used, tests execution, schedule/triggers, quality gates
5. Integrations with 3-d party (reporting, test execution). Think about integration of automated tests into agile process.

Test plan should contain at least below sections:

- Scope of Work
 - To be covered by Automated testing
 - Not to be covered by Automated Testing
- Resources
 - Test Automation Team
 - Tools for test automation
- Test Strategy (test automation)
 - DoR (test automation)
 - DoD (test automation)
 - Automated testing types
 - Automated testing levels
- Risks in test automation

Use EPAM templates for documents:

[ECL | Guideline: Test Strategy \(sharepoint.com\)](#)

[ECL | Template: Test Plan \(sharepoint.com\)](#)

DEFINITION OF DONE

Test plan created.

Ver.	Description of Change	Author	Date
1.0	Initial Version	Yury Karpinski	26-July-2022