



Test Automation Mentoring Program: Advanced

## BDD. 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 ..... 3

    1. ADD BDD TO YOUR PROJECT ..... 3

        TASK..... 3

        DEFINITION OF DONE ..... 4

    2. REIMPLEMENT DDT TESTS USING BDD ..... 4

        TASK..... 4

        DEFINITION OF DONE ..... 4

    3. WORKING WITH STEPS CONTEXT..... 4

        TASK..... 4

        DEFINITION OF DONE ..... 4

    4. PARALLELIZATION FOR BDD ..... 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 alternative test runner and try to use it's capabilities. The goals of this module are:

1. Give a possibility to work with BDD
2. Improve current BDD experience
3. Practice with Gherkin

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 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.

## TASKS

### 1. ADD BDD TO YOUR PROJECT

Complexity	Points	Necessity
Basic	2	Task is Mandatory

#### TASK

1. Integrate BDD framework to your project:
  - a. Java: Cucumber/JBehave
  - b. .NET: Specflow/etc.

- c. JS: Cucumber-js/etc.
2. Use BDD framework dedicated test runner for BDD tests.

## DEFINITION OF DONE

Your project contains references to new test runner.

## 2. REIMPLEMENT DDT TESTS USING BDD

Complexity	Points	Necessity
Basic	2	Task is Mandatory

### TASK

To work with this task use tests implemented in previous module (Test Runners). Rewrite these tests in BDD.

Implement test steps with different input parameter types – list and data table.

Also use below key words:

- Background (if your tests does not require it – implement new ones 😊)
- Scenario Outline
- Examples
- Given, When, Then, And, But
- Rule (if used BDD framework supports)

## DEFINITION OF DONE

Your project contains all needed tests (at least one for each input type – list data table), feature files, steps files with all required infrastructure.

## 3. WORKING WITH STEPS CONTEXT AND HOOKS

Complexity	Points	Necessity
Intermediate	3	Task is Mandatory

### TASK

1. Review implemented steps and assess them for following best practice:
  - each step should be independent and data agnostic. Steps should not be dependent on other ones.
  - test data should be shared between steps with properly organized context injection.

Reimplement steps and utilities to use Steps Context (Scenario Context) if required.

2. Create Hooks class and place some preconditions there. Use below preconditions:
  - a. Before/After TestRun
  - b. Before/After Feature
  - c. Before/After Scenario

## DEFINITION OF DONE

Your steps are organized according to BDD best practices. Your project contains Hooks.

## 4. PARALLELIZATION FOR BDD

Complexity	Points	Necessity
Advanced	3	Task is Optional

### TASK

Set up parallelization for BDD tests.

To make it possible you should configure your config file properly. However, it still doesn't guarantee you that parallel execution will go smoothly and without errors. For parallel execution there shouldn't be any static data storages. Try using async tasks/methods for that.

Also, you can use help of tags in order to configure each thread.

### DEFINITION OF DONE

Your BDD tests are able to run in parallel.

Ver.	Description of Change	Author	Date
1.0	Initial Version	Yury Karpinski	14-Oct-2020
1.1	Updated: Structure	Yury Karpinski	24-Jan-2020