



Test Automation Mentoring Program: Advanced

API TESTING. 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 4

TASKS 4

 1. ADD API CLIENT 4

 TASK..... 4

 DEFINITION OF DONE 4

 2. IMPLEMENT API TESTS 4

 TASK..... 4

 DEFINITION OF DONE 5

 3. CHANGE API CLIENT 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 **API tests**. The goals of this module are:

- Give a possibility to use different API Client library or extend knowledge about added one.
- Use advanced features of API Client
- Evaluate different API libraries

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

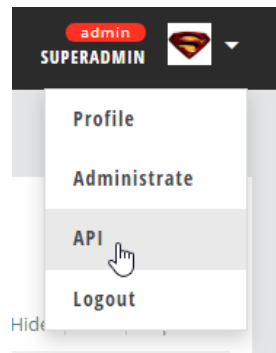
1. Launches
2. Dashboards
3. Widgets
4. Filters

Before this module:

- Mentee has a sample test project implemented.
- The project contains 2 different test runners – use one that was added in Module #2.
- The project can contain API Client added in Module #1 (during optional task #4).

Tip:

To view RP API description: Log-in => click on profile => API:



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. ADD API CLIENT

Complexity	Points	Necessity
Intermediate	3	Task is Mandatory

TASK

1. Add API Client to your project (if you don't have real API testing experience – use the one that your Mentor advises). Choose the one you haven't been working with/have lower experience/want to try
 - a. Java: Apache HTTP Client/RestAssured.
 - b. .NET: RestSharp/HttpClient/Refit
 - c. JS: Axios/Chakram/Frisby.js
2. Implement API Client infrastructure using newly added API Client. Choose the most appropriate architectural approach.
3. Configure Fiddler (or any other http sniffer) to observe traffic between Browser and deployed/web RP

DEFINITION OF DONE

Your project contains API Client with all required infrastructure. You have configured Fiddler (or any other http sniffer) to observe traffic between Browser and RP

Your project contains base test infrastructure implemented with previously added test runner.

2. IMPLEMENT API TESTS

Complexity	Points	Necessity
Intermediate	3	Task is Mandatory

TASK

Implement API tests for chosen application feature (use API documentation for this. See the TIP above):

5. Launches
6. Dashboards
7. Widgets
8. Filters

Implement at least 10 tests covering endpoints listed in RP API documentation. Minimum number of tests for each request type:

- For GET: 1 Positive test, 1 Negative tests
- For POST: 1 positive test, 2 Negative tests
- For PUT: 1 positive test, 2 Negative tests
- For PATCH: 1 positive test, 2 Negative tests
- For DELETE: 1 positive test, 1 Negative tests

For instance, list of endpoints for filters:

user-filter-controller : User Filter Controller		Show/Hide
GET	/v1/{projectId}/filter	Get permitted (own and shared) filters
POST	/v1/{projectId}/filter	Create user filter
PUT	/v1/{projectId}/filter	Update list of user filters
GET	/v1/{projectId}/filter/filters	Get list of specified user filters
GET	/v1/{projectId}/filter/names	Get available filter names
GET	/v1/{projectId}/filter/own	Get all filters for specified user who own them
GET	/v1/{projectId}/filter/shared	Get all available shared filters (except own shared filters)
GET	/v1/{projectId}/filter/{filterId}	Get specified user filter by id
PUT	/v1/{projectId}/filter/{filterId}	Update specified user filter
DELETE	/v1/{projectId}/filter/{filterId}	Delete specified user filter by id

In order to execute API calls and verify response data use Data Models (DTOs) and serialization/deserialization. Feel free to use any library for it (e.g. Jackson for Java, Newtonsoft for .NET, etc.). Choose the most appropriate architectural approaches.

DEFINITION OF DONE

Your test project contains API tests.

3. CHANGE API CLIENT

Complexity	Points	Necessity
Advanced	4	Task is Optional

TASK

Change API client. For instance (for .NET) instead of RestSharp use HttpClient. Modify API part of your framework so that such modification will not cause huge changes in business/tests layers.

DEFINITION OF DONE

Your test project able to work with 2 different API clients.

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 API login		