

Лабораторная работа № 3
Автоматизоване тестування з використанням засобів
Katalon Studio
(4 години)

Мета : Отримати навички а втоматізованно го тестування я і аналізу, розробки Test Case з використанням засобів Katalon Studio .

Теоретична частина

Katalon Studio це набір інструментів для автоматизованого тестування веб та мобільних додатків. Він включає в себе повний пакет потужних засобів, які вирішують часто виникають проблеми при автоматизації UI тестів, таких як pop-up, iFrame, і wait-time. Це гнучке і зручне рішення допомагає тестувальникам тестувати краще, працювати швидше і запускати високоякісне програмне забезпечення завдяки інтелекту, яким воно забезпечує весь процес автоматизації тестування. Katalon Studio доступний для всіх тестувальників безкоштовно і може бути завантажений за посиланням <https://www.katalon.com>

Ключові особливості Katalon Studio :

- просте розгортання: один комплексний пакет містить все необхідне для розгортання потужного інструменту автоматизації тестування;
- швидке і просте налаштування: Katalon Studio не тільки легко встановлюється, але також має просту настройку середовища тестування, тестувальники можуть запускати свій перший тестовий скрипт досить швидко, використовуючи попередньо створені шаблони і тестові скрипти, такі як репозиторії об'єктів і бібліотеки ключових слів;
- результати швидше і краще: вбудовані шаблони допомагають тестувальникам швидко створювати і запускати скрипти. Вони можуть виконувати кожен крок швидко і ефективно, будь це настройка проекту, створення і виконання тестів або створення звітів;
- гнучкі режими: нові тестувальники можуть використовувати recording і ключові слова для створення автоматизованих тестів, в той час як у більш досвідчених є доступ до повної IDE для створення розширених сценаріїв;
- простота використання: навіть manual тестувальник з мінімальним досвідом програмування може з легкістю користуватися перевагами Katalon Studio ;
- крос-браузерний додаток: Katalon Studio підтримує різні платформи: Windows 32 і 64 (7, 8 і 10) і OS X 10.5+.

Детальніше про Katalon Studio :

Katalon Studio - це зручне у використанні рішення для автоматизації, яке відрізняється простотою і швидкістю. Для команди автоматизаторів досить зручно докладати якомога менше зусиль для створення нового проекту, для виконання тестів і моніторингу результатів виконання. Кожен робочий процес надає безліч можливостей і кастомізації для спрощення обслуговування і збільшення масштабу проекту:

- попередньо визначена структура артефактів: тест-кейси, тест-сьюти, тест-об'єкти і звіти. Тестувальникам більше не потрібно витрачати години на їх створення і підтримку;
- ключові слова забезпечують гнучкість в додаванні додаткових ключових слів для ефективного тестування AUT для конкретних і складних цілей тестування;
- підтримка основних потреб тестування: Web , Mobile і API;
- виконуйте відразу кілька тест-сьютів з test-suite collection;
- розширюйте поточний CI процес за допомогою режиму консолі без зайвих зусиль. Запуск командного рядка можна швидко здійснити за допомогою опції "Generate Command Line for console mode";
- з легкістю відстежуйте результати виконання у вигляді таблиці або "дерева" як під час виконання, так і після;
- докладний звіт Test Suite скорочує час аналізу результатів, ви можете експортувати його в різні формати, такі як CSV, PDF, HTML і зберегти його для подальшого використання.

Навігація по лінійному, впорядкованого робочому процесу з допомогою Katalon Studio :

Почніть :

- Вбудовані шаблони проектів: пропонуючи вбудовані шаблони для організації тест-кейсів, сховища об'єктів і ключових слів Katalon Studio значно спрощує процес тестування.
- Багатофункціональність : повністю підтримує тестування Web, Android, iOS і API на всіх операційних системах.
- Безпроблемна інтеграція інструменту: легко інтегрується з Jenkins, GIT і нативними плагінами JIRA.

створюйте :

- Автоматичне створення тестів. Записує дії і автоматично генерує скрипти з використанням вбудованих ключових слів.
- High-end скриптинг : дозволяє створювати складні тестові сценарії або настраюються ключові слова з легкістю і ефективністю.
- Розумне захоплення об'єкта: поліпшений рекордер ефективно визначає властивості об'єкта, щоб максимізувати розпізнавання.

виконуйте :

- Виконання тесту стало могутніше: запускає тест-кейси або тест-сьюти з використанням безлічі конфігурацій і наборів даних.

- Універсальність у виконанні: надає консоль інтеграції CI з різними параметрами для віддаленого виконання. Виконує тести на декількох браузерах і операційних системах локально, а також на Sauce Labs і BrowserStack.

- Гнучка обробка відмови і автоматичне повторне виконання. Включає правила виконання для автоматичної обробки складних потоків.

Отримуйте звіти :

- Звіти доступні в декількох форматах: з розширеним протоколюванням, налагоджувальними даними і скріншотами

- Звіти про виконання замовлення: інтегровані з процесом повідомлень.

- Розширені логи Selenium і Appium : с можливостями більш глибокого аналізу для поліпшення стратегії автоматизації.

Підтримуйте :

- Розумна підтримка тест-об'єктів: автоматично оновлює всі пов'язані з цим тестові приклади і комплекти при зміні об'єктів.

- Ефективна організація тестування: дозволяє легко керувати і підтримувати тести, дані і ключові слова.

- Спрощене взаємодія всередині команд: інтегрується з GIT, щоб члени команди могли легко ділитися артефактами і робочим навантаженням.

Katalon Studio як життєздатної альтернативи дорогим комерційним інструментам .

Katalon Studio досить хороший а в порівнянні з комерційними інструментами автоматизації тестування, такими як UFT і Test Complete з кількох причин. Він являє собою життєздатний, практично безкоштовний варіант для таких інструментів, доступних на ринку:

- ніяких витрат на ліцензування та обслуговування;
- безкоштовні апгрейди;
- крос-платформенне рішення.

Додаткові можливості Katalon Studio :

- Зменшено розмір завантаження: пакет Katalon Studio для різних ОС зменшився майже на 25%.

- Запис і відтворення для мобільних пристроїв: утиліта запису підтримує Android і iOS і здатна швидко генерувати тест-кейс з тим же UI, що і для веб-тестування.

- Параметризація тестових об'єктів. Одна з найбільш затребуваних функцій для Katalon Studio , користувачі можуть використовувати нову функцію для динамічного управління об'єктами.

- Мережеві настройки: версія 4.6 вводить можливість обходу перевірки сертифіката, а також налаштування проксі-сервера 25, що дозволяють користувачам з обмеженою мережею відповідно налаштувати Katalon Studio.

- Ключові слова для сторінок Angular / jQuery : в цю версію включені додаткові ключові слова, щоб допомогти користувачам впоратися з проблемами завантаження з використанням JS технологій.

- Покращена інтеграція Kobiton : користувачі Kobiton можуть запускати свої тести з використанням ID додатки, створеного Kobiton.

Порядок роботи

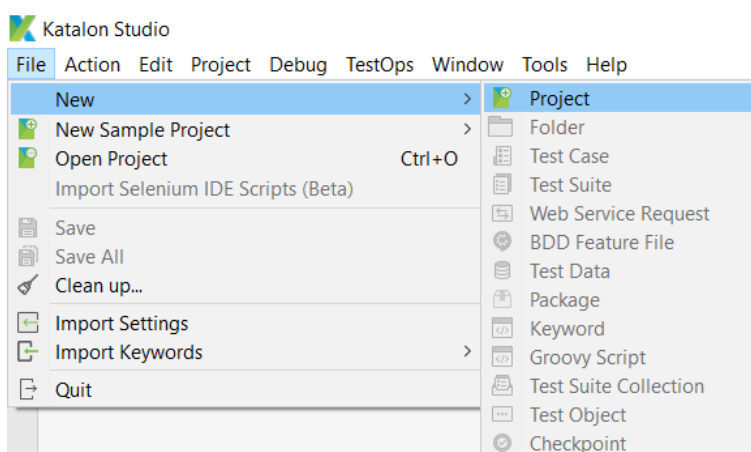
- На ваш розсуд вибрати сайт для автоматизації.
- Створити не менше трьох Тест Кейсів (Test Case).
- Створити профіль, в якому визначити глобальні властивості проекту.
- Створені Тест Кейси (Test Case) об'єднати в Тестовий Сьют (TestSuite).
- Реалізувати мінімум один власний KeyWord. Створений KeyWord застосувати в одному з Тест Кейсів (Test Case).
- Створити TestListener , в якому реалізувати мінімум один з чотирьох методів (Before / After TC, Before / After TS) на вибір. Проаналізувати роботу методу.

Установка Katalon Studio :

1. Перейти на <https://www.katalon.com/>
2. Натисніть на Download (Вам знадобиться валідний email щоб активувати Katalon Studio аккаунт)
Виберіть вашу операційну систему
- 3) Як тільки завантаження завершиться - перейдіть в відповідну папку і запустіть Katalon.exe
- 4) Розпочнеться установка.
- 5) Введіть дані вашого облікового запису.

Створим новий проект :

Перейдіть в меню File -> New -> Create Project і введіть назву проекту в який з'явився нижче вікні.



New Project
Please enter project information

Name:

Type: ☒ Web ☐ API/Web Service ☐ Mobile ☐ Generic ?

Project:

Repository URL:

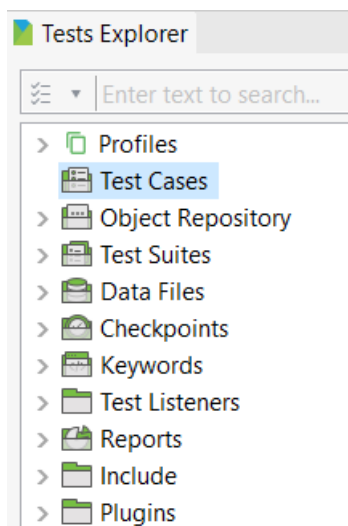
Location:

Description:

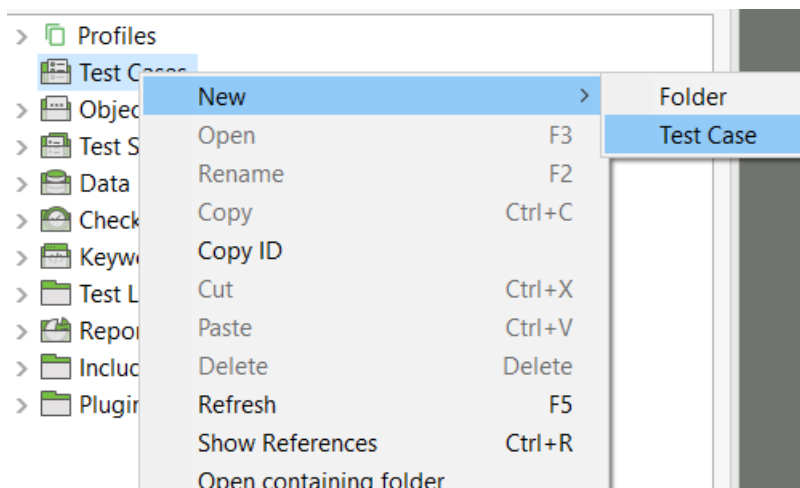
☐ Generate .gitignore file

☐ Generate build.gradle file

Ваш проект відкриється, і ви побачите наступні елементи, доступні в Test Explorer:

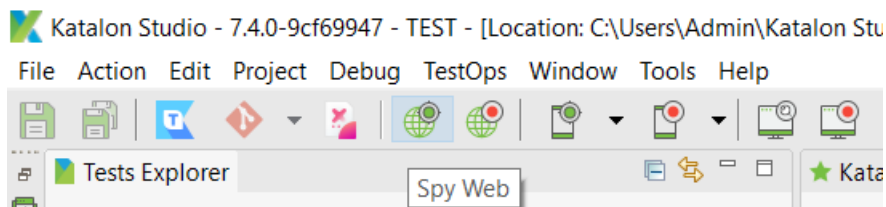


Створи тест кейси:



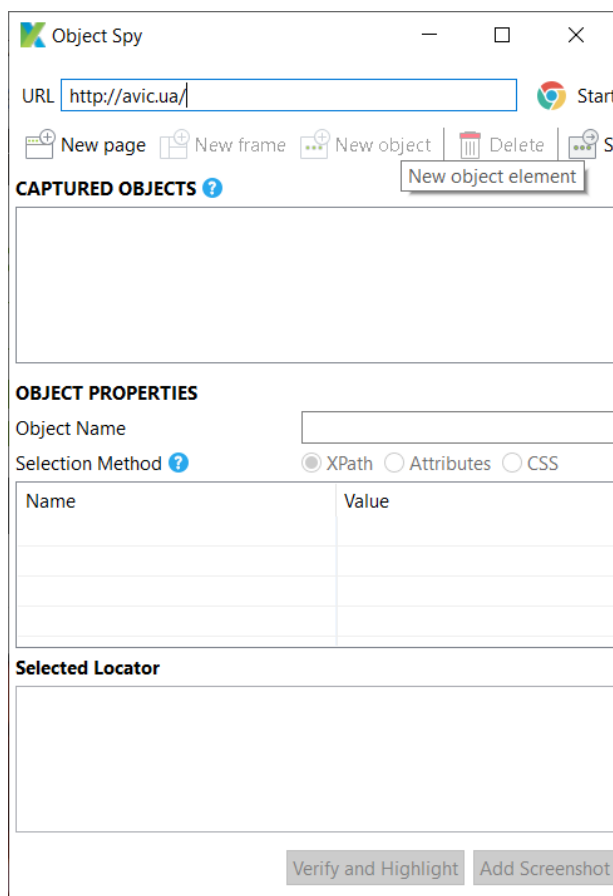
Існує багато підходів до створення тест-кейса. Це може бути як запис вашого тесту, так і генерація рядків коду після додавання об'єкта. Розглянемо другий підхід.

Перейдіть на вкладку «Object Repository», і для роботи над веб-додатком, натисніть «Spy Web».

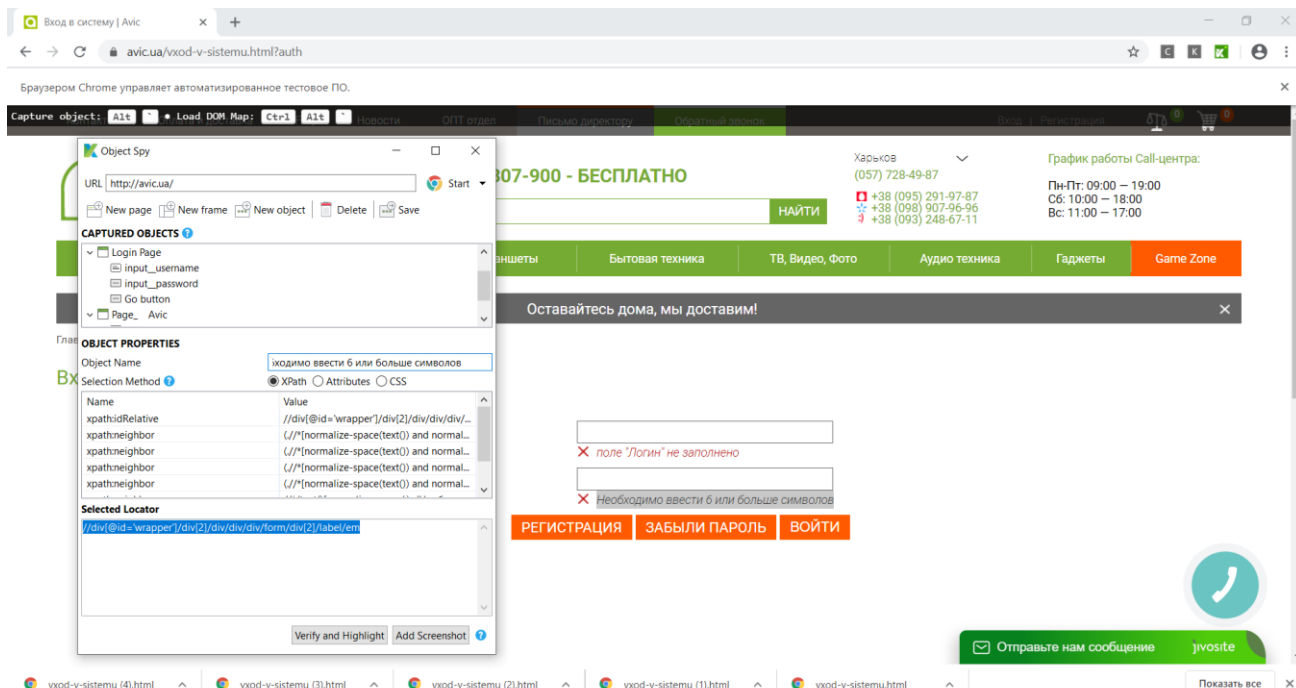


З'явиться вікно Object Spy. Введіть стартовий URL і виберіть, в якому браузері його відкрити.

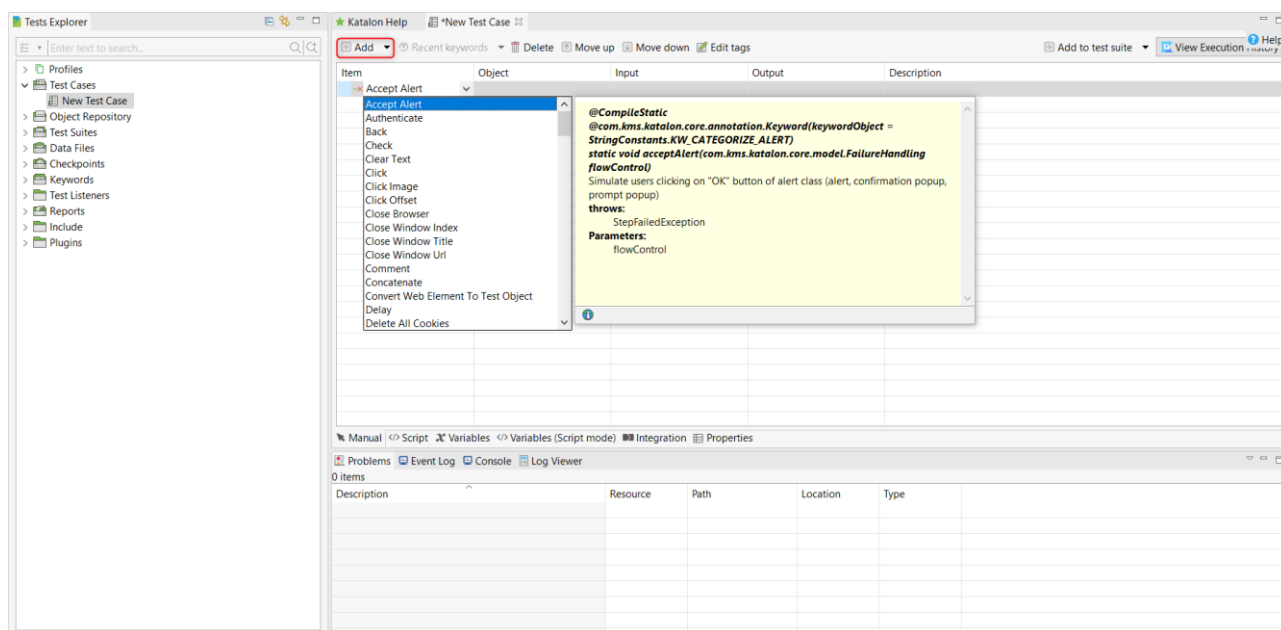
Це запустить URL в обраному браузері. Переходимо по <https://avic.ua/> використовуючи Chrome.



Візьміть всі об'єкти, необхідні для тесту, натиснувши «Alt + ~», тепер об'єкти будуть відображатися у вікні сховища об'єктів під "Captured Objects":



Щоб додати кроки тестування натисніть «Add», і в такій рядку клацніть стовпець «item», щоб додати ключове слово / метод / дію, яке ви маєте намір виконати.



Отже, натискаємо для виконання відповідних завдань:

Item : вибрати, що робити;

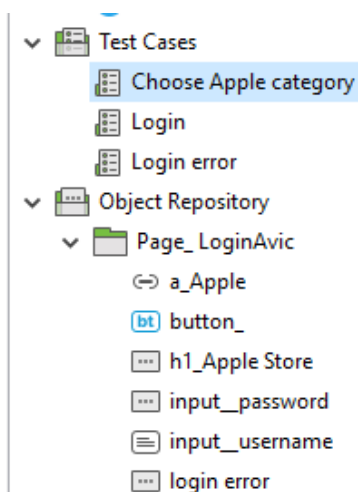
Object : де , на якому об'єкті;

Input : що подавати на крок;

Output : де зберігати значення, що повертається.

Аналогічним чином, продовжуємо додавати все кроки. Гарна ідея - додати деякий опис на кожен крок, щоб при майбутньому використанні тест-

кейса чітко розуміти, що тут і навіщо. Для цього клацніть стовпець опису і введіть опис.



TC1 Login to avic.com

Item	Object	Input	Output	Description
1 - Open Browser		"https://avic.ua/vxod-v-sistemu.htm"		
2 - Set Text	input_username	"Marharyta"		
3 - Set Encrypted Text	input_password	"RigbBhfdqOBDK95asqKeHw=="		
4 - Click	button_			

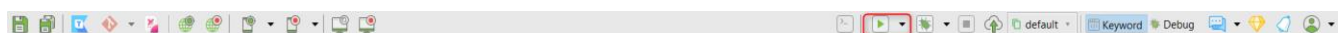
TC2 Check login error

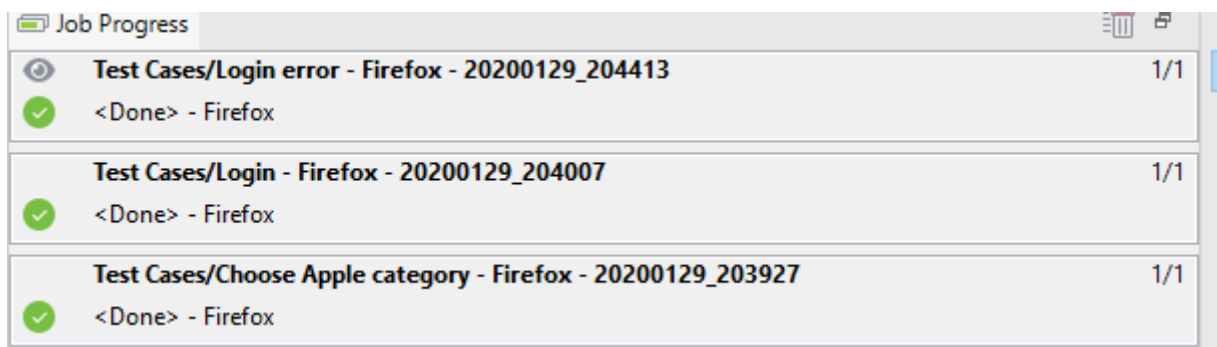
Item	Object	Input	Output	Description
1 - Open Browser		"https://avic.ua/vxod-v-sistemu.htm"		
2 - Set Text	input_username	"Marharyta"		
3 - Set Encrypted Text	input_password	"RigbBhfdqOBDK95asqKeHw=="		
4 - Click	button_			
5 - Verify Element Visible	a_Apple			
6 - Click	a_Apple			
7 - Verify Element Visible	h1_Apple Store			

TC3 Choose Apple category

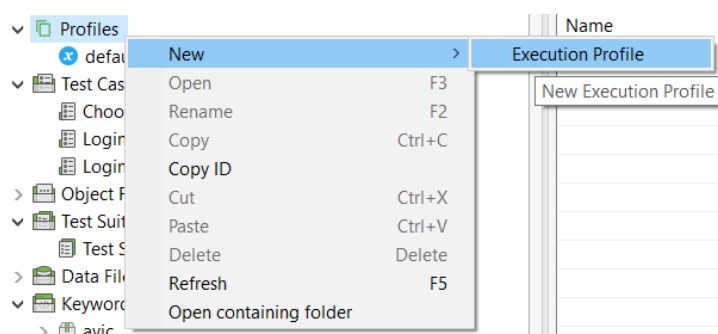
Item	Object	Input	Output	Description
1 - Open Browser		"https://avic.ua/vxod-v-sistemu.htm"		
2 - Set Text	input_username	"Marharyta"		
3 - Set Encrypted Text	input_password	"RigbBhfdqOBDK95asqKeHw=="		
4 - Click	button_			

Run created TC Check result.

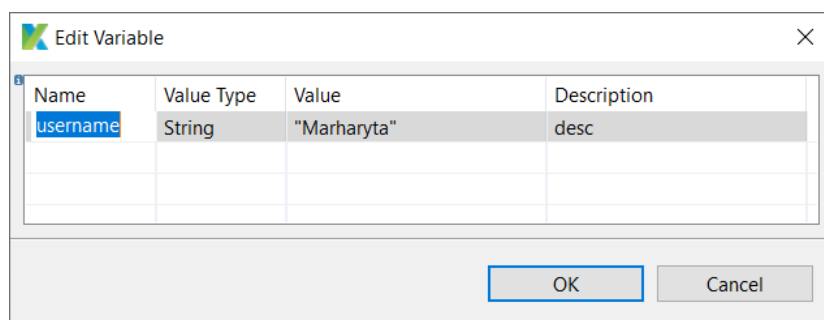




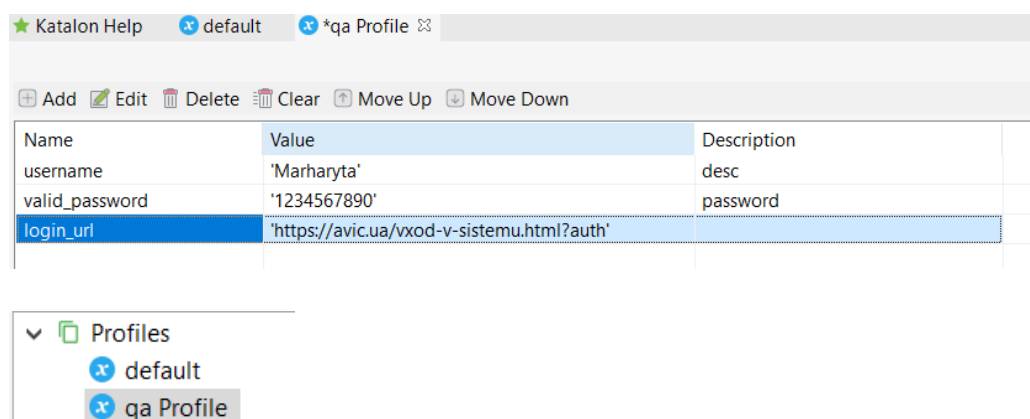
Створюємо Профіль, в якому визначаємо глобальні властивості проекту (AddProfile):



Введіть дані в профіль (Enter data to profile):



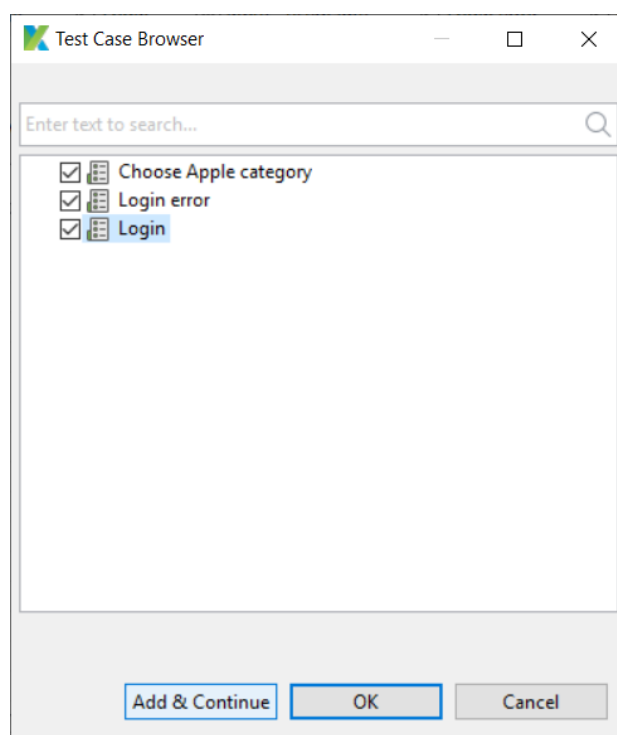
Введіть значення (Enter value):



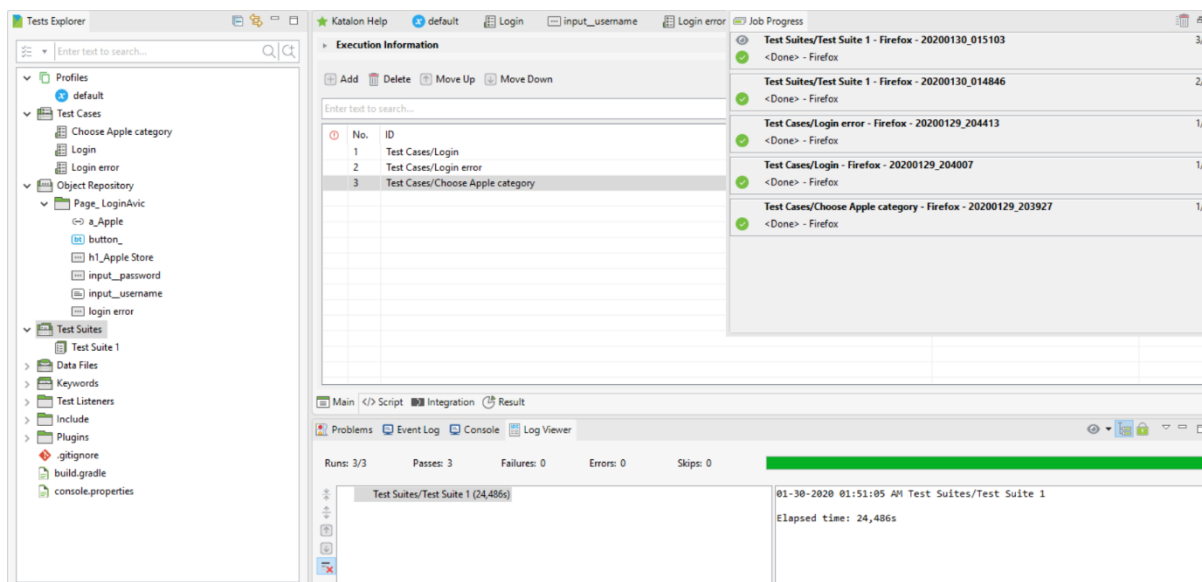
Створені Тест Кейси об'єднуєм в Тестовий Сьют (Create TestSuite1 and add our test case):



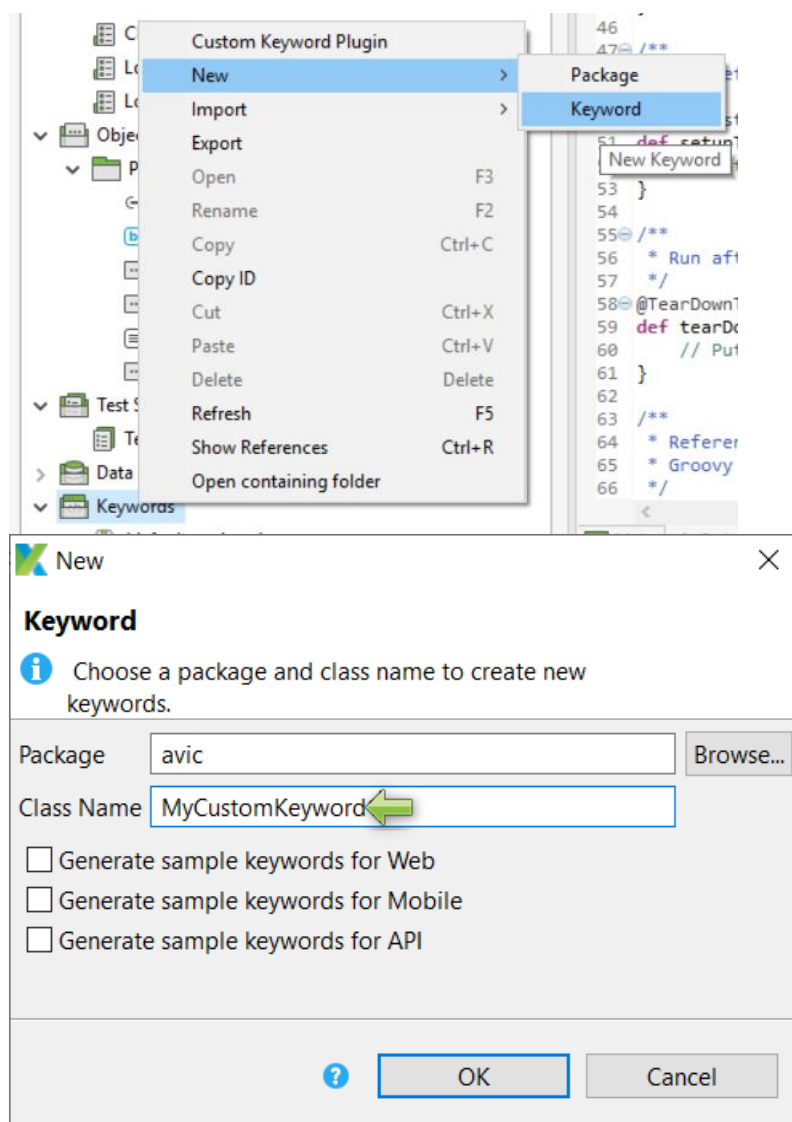
Вибрати всі тести (Choose all test case):



Запустіть тестовий пакет (Run test Suite):



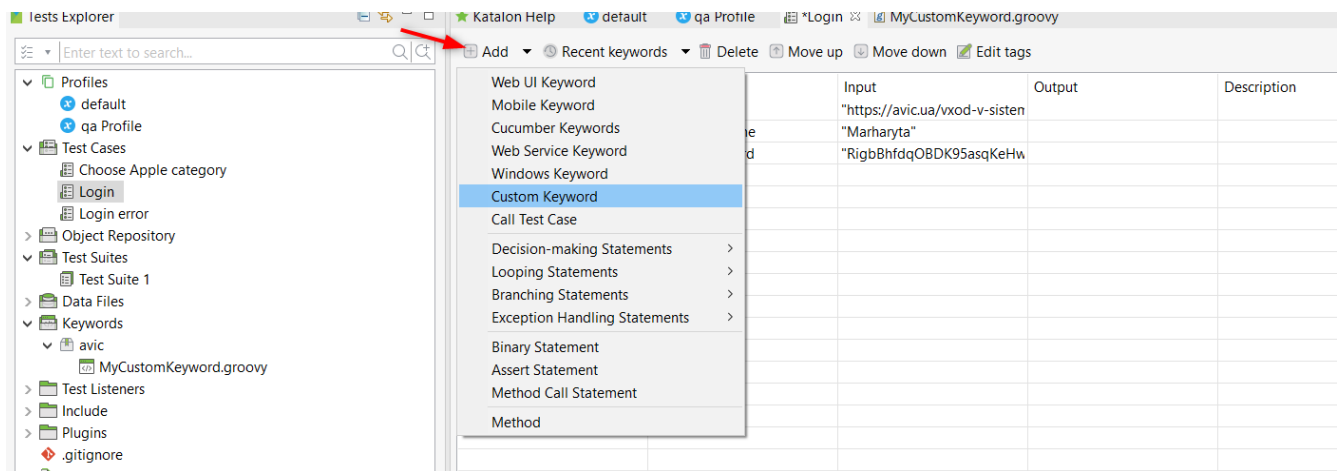
Реалізуємо власний KeyWord. Створений KeyWord застосовуємо в одному з Тест Кейсів.



```
Katalon Help  default  qa Profile  *Login  MyCustomKeyword.groovy
```

```
1 package avic
2
3 import static com.kms.katalon.core.checkpoint.CheckpointFactory.findCheckpoint
22
23 public class MyCustomKeyword {
24     @Keyword
25     def myCustomPrint(String msg) {
26         //print a simple message to console
27         println 'the message is: ' + msg
28     }
29 }
30 }
```

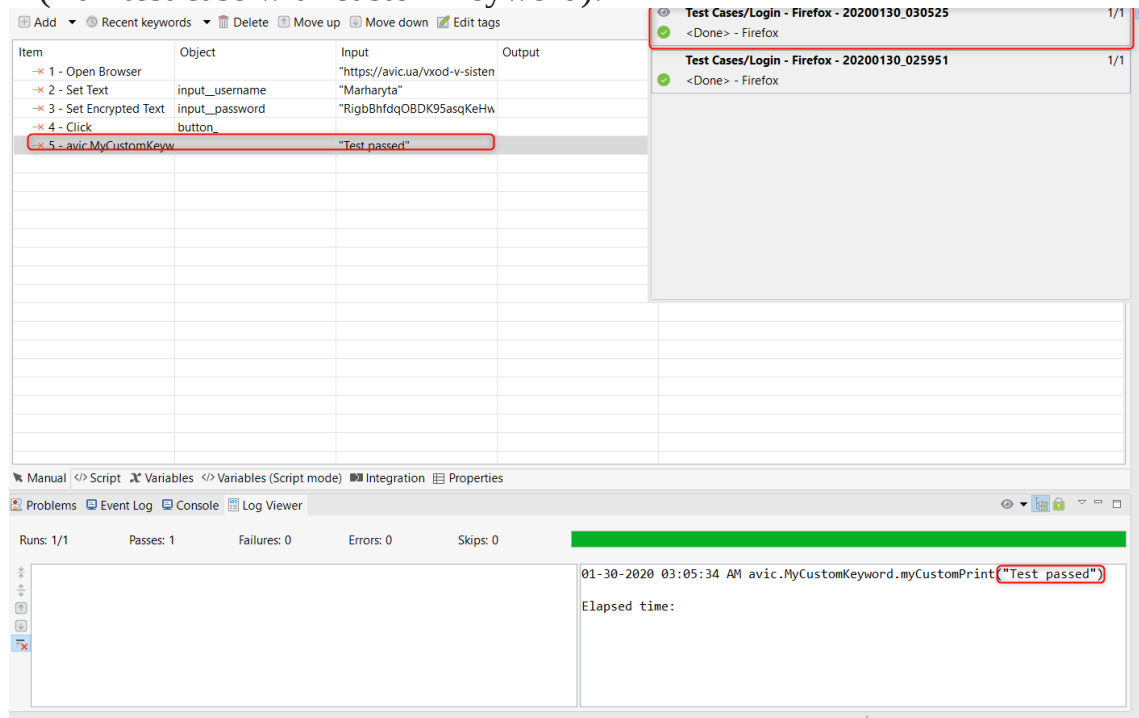
Відкрийте Test Case і додайте своє ключове слово (Open TC and add custom keyword):



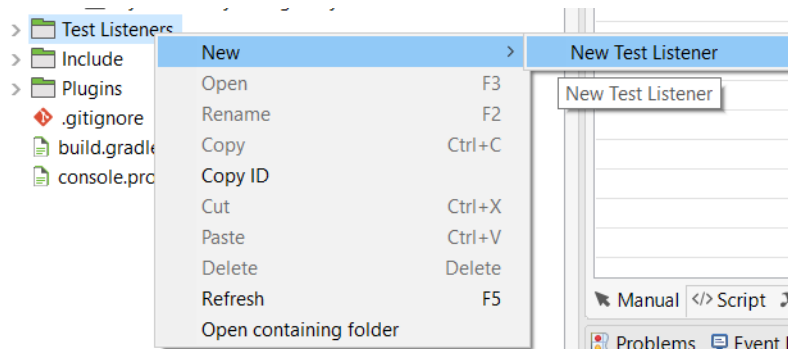
Введіть значення « пройдений тест » в параметр msg (Enter “test passed” value to parameter msg):

[illegible]

Запустіть тестовий приклад з призначенням для користувача ключовим словом (Run test case with custom keyword):



Створюємо TestListener , в якому реалізуємо Before / After TC метод:



Create 2 method : BeforeAfter TC

```
class CDP_TestListener {
    /**
     * Executes before every test case starts.
     * @param testCaseContext related information of the executed test
    case.
     */
    @BeforeTestCase
    def sampleBeforeTestCase(TestCaseContext testCaseContext) {
        WebUI.openBrowser("");
        WebUI.maximizeWindow();
        WebUI.navigateToUrl(GlobalVariable.login_url)
    }
    /**
     * Executes after every test case ends.
     */
}
```

