

Katalon Studio

Tests web scriptés

Michaël MATHIEU

7 septembre 2025

Résumé

Scripter des tests permet de garantir une stabilité et une confiance pour les mises en production.

1 Katalon

Il existe deux manières d'utiliser Katalon : la version *Recorder* qui se présente sous la forme d'un plugin dans le navigateur et la version *Studio* qui permet de scripter les tests. C'est cette dernière version que ce document se propose de présenter.

Seules les fonctionnalités de base, actives avec la version gratuite, seront utilisées.

2 Page à tester

Pour illustrer l'utilisation de Katalon Studio, nous utiliserons la page <https://michaelmathieu.net/resources/katalon/login/> dont le script est :

```
<?php
    $logged = false;
    if ($_SERVER["REQUEST_METHOD"] === "POST") {
        $email = filter_input(INPUT_POST, "email", FILTER_SANITIZE_EMAIL);
        $password = filter_input(INPUT_POST, "password");
        $logged = $email === "toto@mail.com" && $password === "1234"; // check credentials
        if (!$logged) {
            echo '<span style="color:red" id="lblLoginError">Wrong credentials</span>';
        }
    }

    if ($logged) {
        echo '<a href="." id="lnkLogout">Logout</a>';
    } else {
        echo <<<EOL
            <form method="POST">
                <input type="email" id="email" name="email" placeholder="toto@mail.com"><br>
                <input type="password" id="password" name="password" placeholder="1234"><br>
                <input type="submit">
            </form>
        EOL;
    }
?>
```

3 Katalon Studio

La seule version que l'on peut télécharger est la version *Enterprise*¹. Il faut se créer un compte et être connecté pour utiliser Katalon Studio. Après 30 jours d'essai, les fonctionnalités payantes ne seront plus disponibles. Bonne nouvelle, ces dernières ne sont pas indispensables et ce document ne les abordent pas.

3.1 Préambule

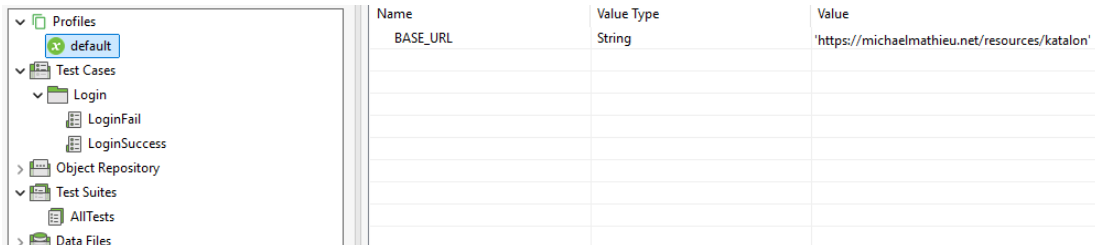
Après avoir installé Katalon Studio, commencez par télécharger les drivers relatifs au web : Katalon Studio → Tools → Update Web Drivers → Chrome

Définir des constantes pour vos tests

A minima, l'URL où votre projet est déployé doit être en constante afin de pouvoir très facilement modifier l'URL de **tous** vos tests avec une seule modification.

Les constantes se trouvent dans l'explorateur de tests dans Profiles → default

Dans l'exemple ci-dessous, nous avons une constante globale **BASE_URL** qui pointe sur <https://michaelmathieu.net/resources/katalon>



Name	Value Type	Value
BASE_URL	String	'https://michaelmathieu.net/resources/katalon'

3.2 Tester votre application

Chacun des tests suivants peuvent être exécuté individuellement.
Notez l'utilisation de la constante globale `internal.GlobalVariable.BASE_URL`.

3.2.1 Tester le scénarii nominal

```
import com.kms.katalon.core.testobject.ConditionType
import com.kms.katalon.core.testobject.TestObject
import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI

WebUI.openBrowser(internal.GlobalVariable.BASE_URL + '/login')

TestObject inputEmail = new TestObject()
    .addProperty("id", ConditionType.EQUALS, "email")
WebUI.setText(inputEmail, "toto@mail.com")

TestObject inputPassword = new TestObject()
    .addProperty("id", ConditionType.EQUALS, "password")
WebUI.setText(inputPassword, "1234")

TestObject btnSubmit = new TestObject()
    .addProperty("type", ConditionType.EQUALS, "submit")
WebUI.click(btnSubmit)

TestObject lnkLogout = new TestObject()
    .addProperty("id", ConditionType.EQUALS, "lnkLogout")
WebUI.verifyElementVisible(lnkLogout)

TestObject lblLoginError = new TestObject()
    .addProperty("id", ConditionType.EQUALS, "lblLoginError")
WebUI.verifyElementNotPresent(lblLoginError, 5)

WebUI.closeBrowser()
```

3.2.2 Tester le scénarii alternatif

Il est important de ne pas uniquement tester le cas nominal ($\Rightarrow$ ne pas se contenter de s'assurer que cela fonctionne si les entrées sont correctes), mais également les scénarios alternatifs ($\Rightarrow$ les erreurs s'affichent-elles correctement quand les entrées sont erronées?). Dans notre cas, on pourrait tester :

- le format de l'email
- si les credentials sont corrects
- etc.

```
import com.kms.katalon.core.testobject.ConditionType
import com.kms.katalon.core.testobject.TestObject
import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI

WebUI.openBrowser(internal.GlobalVariable.BASE_URL + '/login')

TestObject btnSubmit = new TestObject()
                        .addProperty("type", ConditionType.EQUALS, "submit")
WebUI.click(btnSubmit)

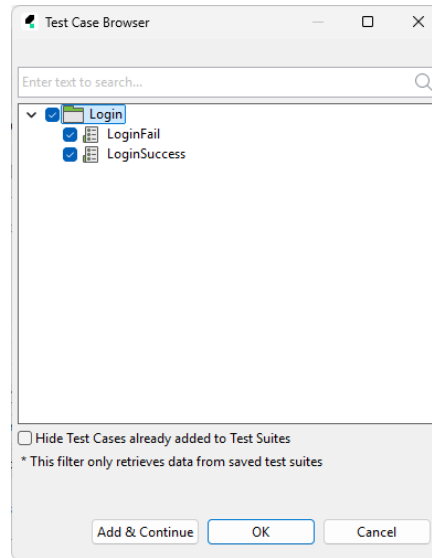
TestObject lnkLogout = new TestObject()
                        .addProperty("id", ConditionType.EQUALS, "lnkLogout")
WebUI.verifyElementNotPresent(lnkLogout, 5)

TestObject lblLoginError = new TestObject()
                        .addProperty("id", ConditionType.EQUALS, "lblLoginError")
WebUI.verifyElementVisible(lblLoginError)

WebUI.closeBrowser()
```

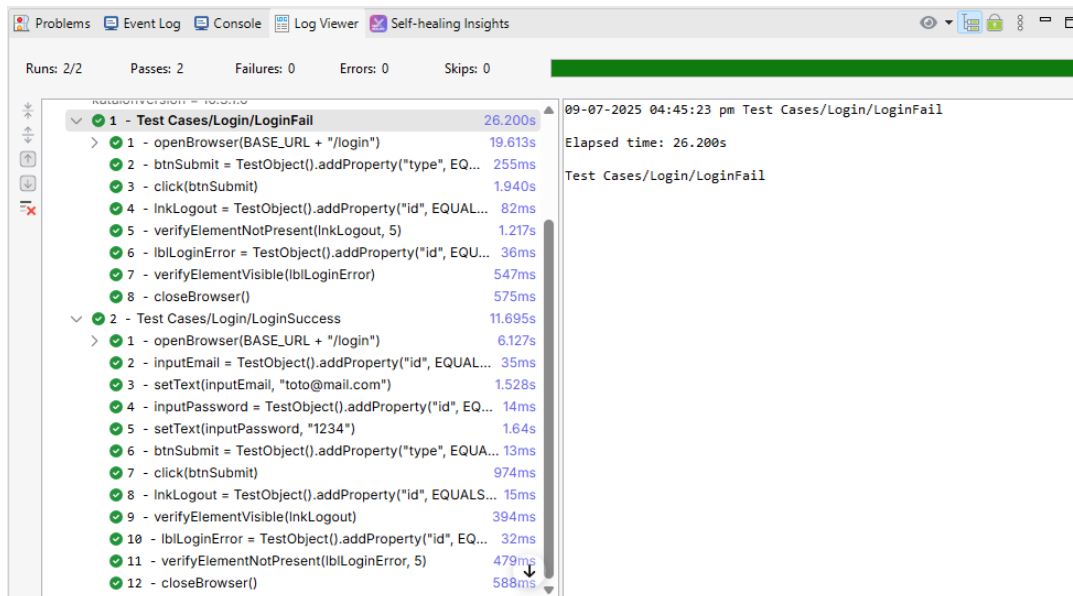
3.2.3 Exécuter tous vos tests d'un seul click

Pour cela, vous devez créer un *Test Suite* et y ajouter tous vos tests (bouton *Add*).



Ensuite, cliquez sur le bouton *Run*.

Après un certain temps (cela peut être long..), vous aurez une vue globale du résultat de tous les tests :



3.2.4 Pour aller plus loin

Voici une liste des différents types de vérifications que l'on peut effectuer :

Vérification	méthode(s)
Elément présent dans le DOM	<code>verifyElementPresent(to, timeout)</code>
Elément non présent dans le DOM	<code>verifyElementNotPresent(to, timeout)</code>
Elément visible	<code>verifyElementVisible(to)</code>
Valeur d'un attribut (par exemple <code>value</code>)	<code>verifyElementAttributeValue(to, attributeName, attributeValue, timeout)</code>
Texte exact d'un élément (<code>span</code> , <code>label</code> , <code>div</code>)	<code>verifyElementText(to, expectedText)</code>
Texte présent quelque part dans la page	<code>verifyTextPresent(text, isRegex)</code>
Options sélectionnées (dropdown)	<code>verifyOptionPresentByValue(...)</code> , <code>verifyOptionsPresent(...)</code> , <code>verifyOptionSelectedByValue(...)</code> , etc.
Liens accessibles (URLs) à vérifier	<code>verifyLinksAccessible(links)</code>