***Configurer Sélénium WebDriver & SpecFlow avec Visual Studio en C #***

**Pour**



V1.0

**Zied Hannachi**

**Team Leader Test Auto**

**29/10/2020**

# 

[**Solution de test**](#_z5xc9bccs7an) **3**

[Synthèse Framework test auto](#_5qccwjkqmepr) 4

[Le framework est adapté à un processus Agile, il peut utiliser l’approche BDD ou pas suivant les besoins (automatisation des TNR Legacy ou New features).](#_a4nxqfiwb0mc) 4

[Technologies](#_t6gbwb8kmt2r) 4

[Architecture / Structure](#_udxbm9kd1xx4) 6

[**Configurer Visual Studio avec Sélénium WD**](#_3b136kxknq49) **6**

[Prérequis](#_tuf3c1sgayol) 6

[Etape 1](#_tjjtejrzuic9) 6

[Etape 2](#_ajm3smoxje) 7

[**L’architecture de la solution “AlissiaE2ETest”**](#_bnewkuamsof0) **10**

[Features](#_g1jxjremf9au) 10

[PageObjects](#_vcdsbuj24by7) 11

[StepDefinition](#_vrr1esprzezj) 12

[WrapperFactory](#_irpezx8e5wkm) 13

[Sources](#_ikeqxhh2lepv) 13

[Reports](#_mix8yd8zgk6g) 14

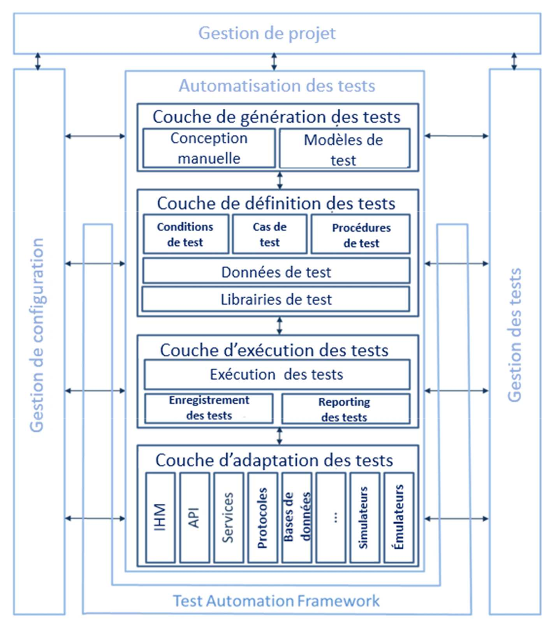
[ScreenShots](#_bb3mfspeez8k) 14

[**Intégration Continue**](#_fzkpij2fjhf) **14**

[Jenkins](#_9krz7m4hngwr) 14

# Solution de test

“AlissiaE2ETest” est une solution d’automatisation de test (IHM) de bout en bout mis en place pour Ordipat.



Architecture d’automatisation de test générique[[1]](#footnote-1)

## Synthèse Framework test auto

### Le framework est adapté à un processus Agile, il peut utiliser l’approche BDD ou pas suivant les besoins (automatisation des TNR Legacy ou New features).

* Approche BDD

BDD est une méthode de conception logicielle qui étend TDD, par l’apport de pratiques additionnelles, et qui explore et spécifie les besoins, en mettant l’accent sur la collaboration et en s’appuyant sur les tests et leur scénarisation.

### Technologies

* SpecFlow pour le BDD :

<https://specflow.org/>

[SpecFlow](https://specflow.org/) est un framework de test et un plug-in qui s’intègre avec Visual Studio, qui permet d’écrire des tests d’une manière naturelle, puis de les exécuter avec un dérouleur de tests.Il se base su un principe de BDD.

Specflow permet de créer des tests unitaires à partir de comportements écrits en syntaxe Gherkin (Given,When,Then).

* Webdriver pour les UI

La lib officielle / Selenium (en C#)

<https://selenium.dev/downloads/>

[Selenium WebDriver](https://selenium.dev/) est un framework web qui vous permet d'exécuter des tests multi-navigateurs. Cet outil est utilisé pour automatiser le test d'applications Web pour vérifier qu'il fonctionne correctement.

* ATATA

<https://atata.io/>

[Atata](https://atata.io/) est un framework d'automatisation de test web en C#/.Net pour les applications Web.

* WebDriver: Basé sur Selenium WebDriver et préserve toutes ses fonctionnalités.
* Intégration: Fonctionne sur n'importe quel moteur de test .NET (par exemple NUnit, xUnit, SpecFlow) ainsi que sur les systèmes CI comme Jenkins, Azure DevOps.
* Configurable: Définit les stratégies de recherche de composants par défaut ainsi que des paramètres supplémentaires. Atata.Configuration.Json fournit des configurations JSON flexibles.
* Rapports: Journalisation personnalisable intégrée et fonctionnalité de capture d'écran.
* Page Object Model: Fournit un modèle d'objet de page fluide unique, facile à implémenter et à gérer.
* Vérification: Un ensemble de méthodes et de déclencheurs d'assertion fluides pour la vérification des composants et des données.
* Assertion

FluentAssertion

<https://fluentassertions.com/>

Avec Fluent Assertions, les assertions sont belles, naturelles et, surtout, extrêmement lisibles.

* Report

extent report

<https://extentreports.com/>

[ExtentReports](http://extentreports.com/) est un framework de génération de rapports détaillés et interactifs, utilisé avec Selenium.

* Runtime

MsTest

<https://docs.microsoft.com/fr-fr/dotnet/core/testing/unit-testing-with-mstest>

MSTest est un framework de test unitaire fourni par Microsoft Visual Studio qui possède des fonctionnalités intégrées pour prendre en charge les tests pilotés par les données, qui peuvent être configurées très facilement.

* Langage de Scripting

C#

### Architecture / Structure

* POM (Page Object Model)

features

stepDefinitions

pageObjects

reports

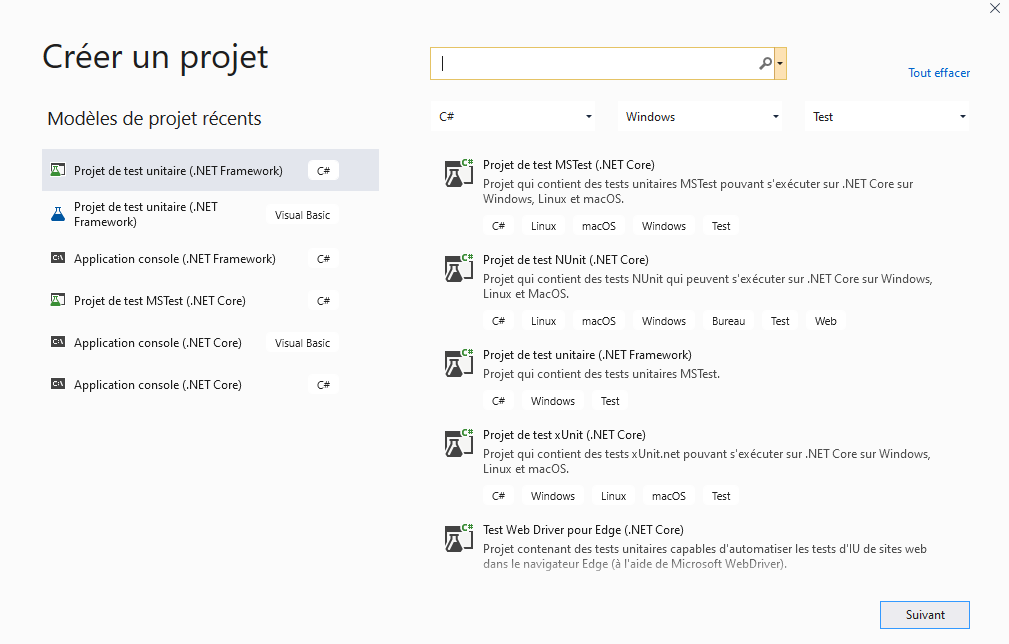
# Configurer Visual Studio avec Sélénium WD

## Prérequis

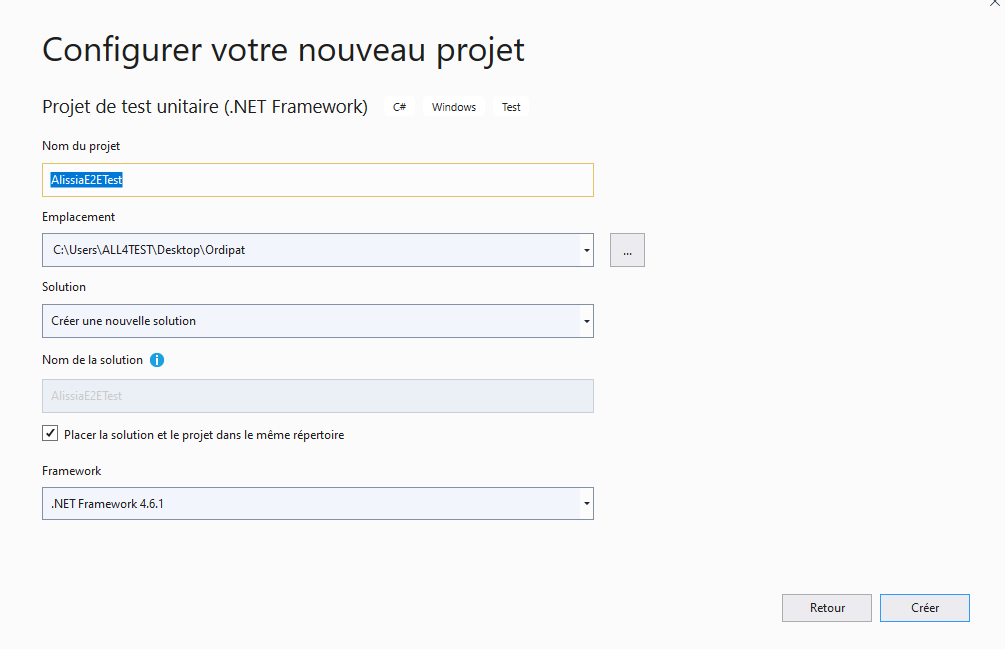
* VS 2017

## Etape 1

* Choisir “Projet de test unitaire (.Net)”

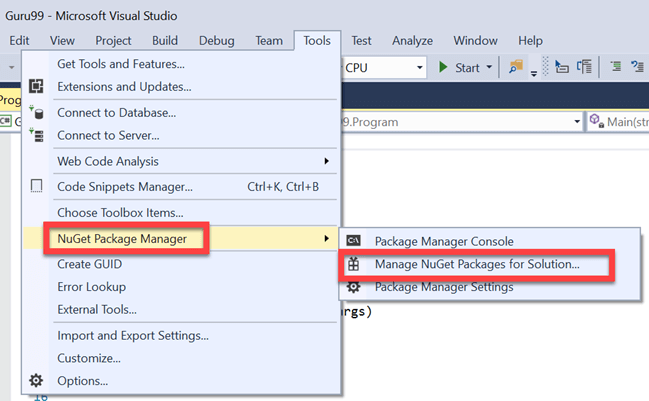


* Créer un Projet “AlissiaE2ETest”



## Etape 2

* Configurer Sélénium Web Driver

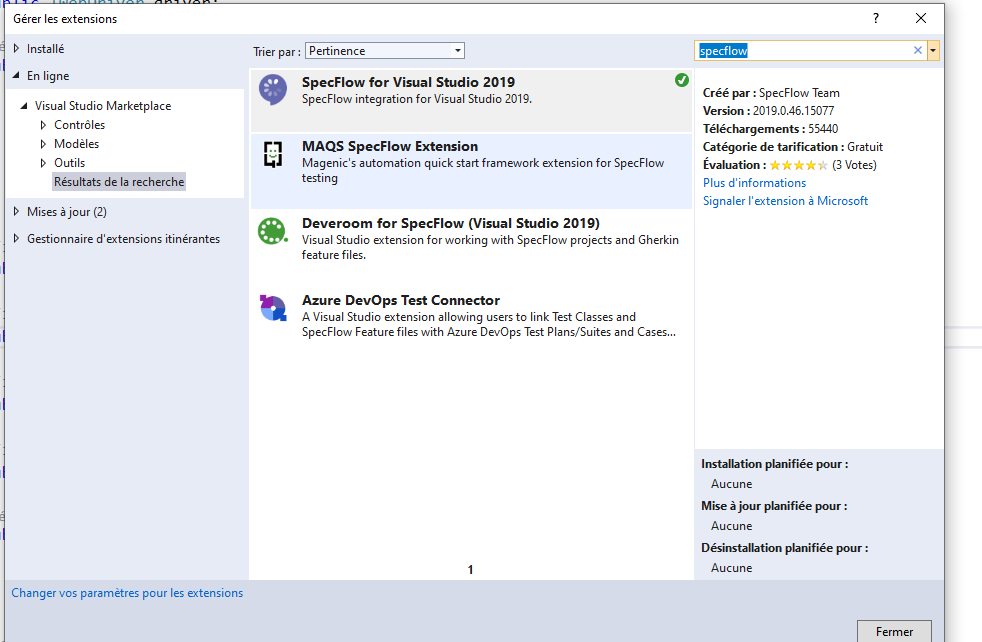


* Ajouter le pluguin SpecFlow

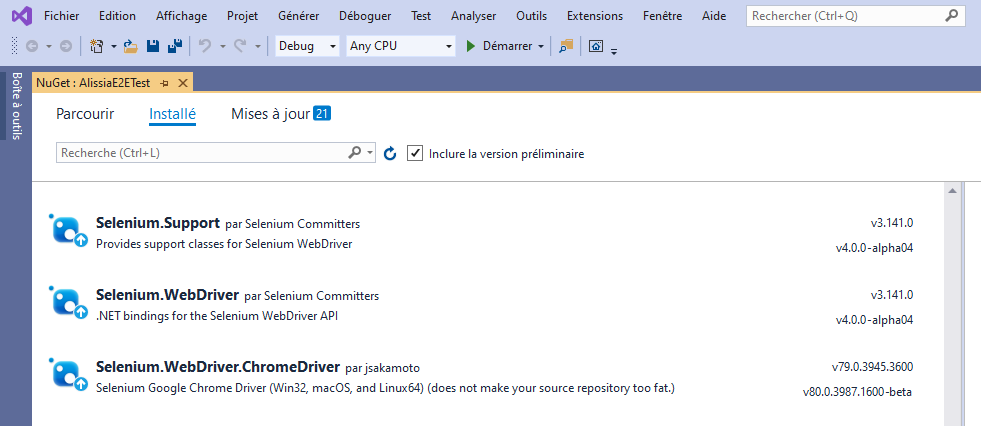
Outils🡺 Extentions 🡺 En ligne

Taper **SpecFlow**

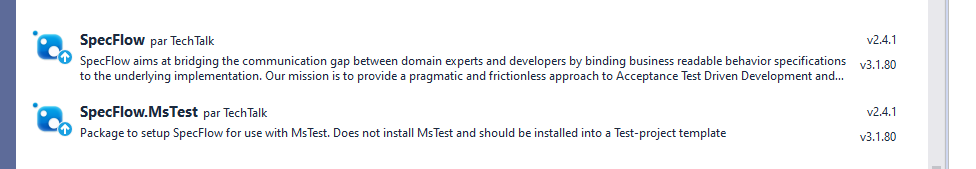
**Choisir Specflow for VS « version »**



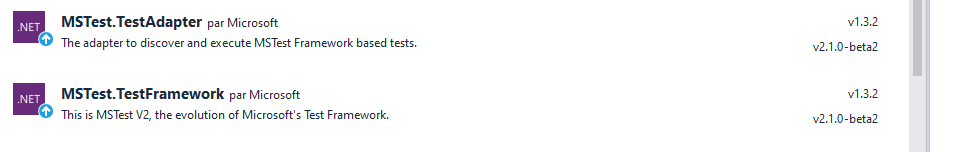
* Rechercher Selenium et installer



* Rechercher SpecFlow et installer

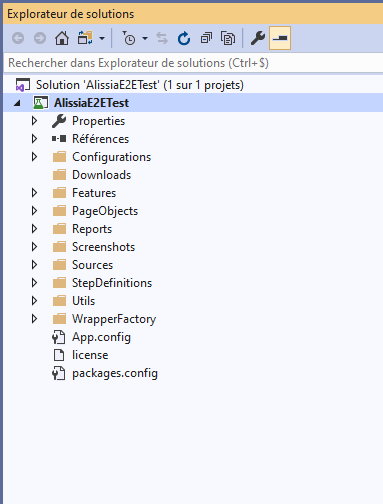


* Rechercher MsTest et installer



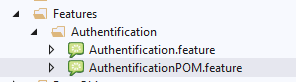
# 

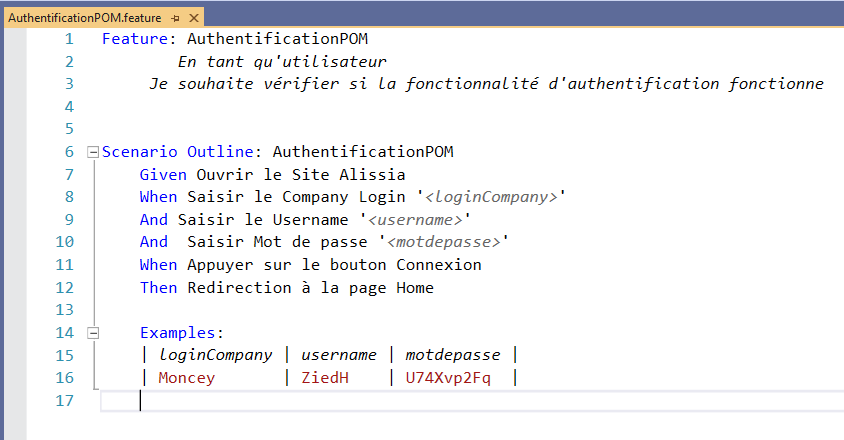
# L’architecture de la solution “AlissiaE2ETest”



## Features

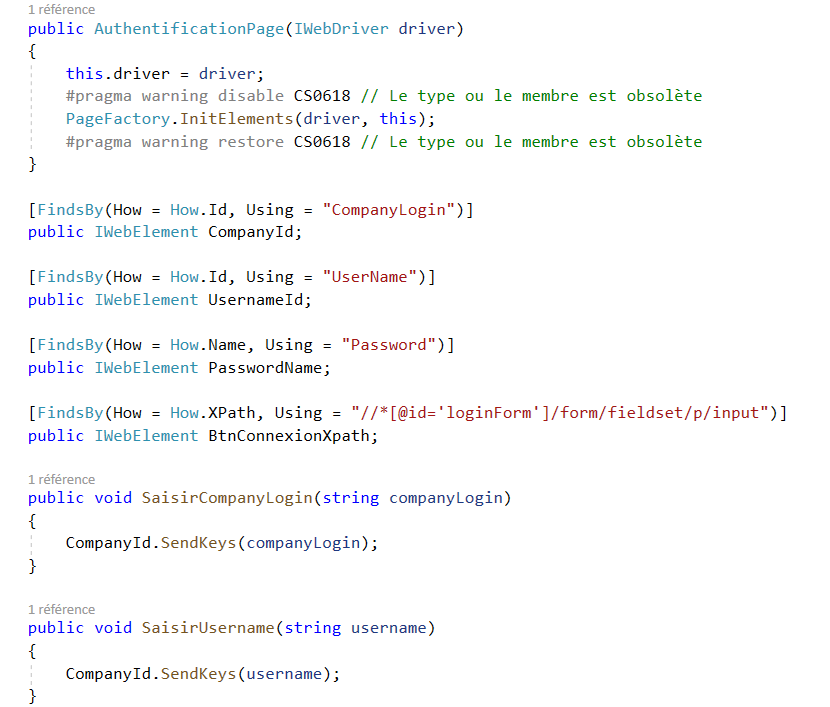
Rédaction des cas de test en Gherkin





## PageObjects

Identification des éléments et création des méthodes



## StepDefinition

Classe de définition des étapes

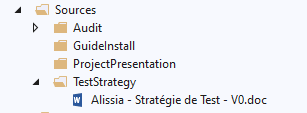


## WrapperFactory

Design Patter Factory Navigator

## Sources

Liste des documents de la solution



## Reports

Le rapport ExttentReport

## ScreenShots

Les captures d’écran quand il y a un bug

# Intégration Continue

## Jenkins

* [Jenkins](https://jenkins.io/) est une plateforme de CI/CD, très extensible grâce à ses plugins.

1. ISTQB, Syllabus Niveau Avancé, <http://www.cftl.fr/wp-content/uploads/2019/06/Avance_TA_E_Syllabus-FR-2016_Final_v_1_0.pdf> [↑](#footnote-ref-1)