

Huaxing
YUAN

Joseph
ARUL

Automatisation des tests avec GitHub Copilot



SOMMAIRE

Partie I – Introduction

- Github Copilot
- Déploiement chez AXA
- Premier pas

Partie II – Mise en contexte

- Contexte de votre application
- Adoptez les bonnes pratiques
- Copilot Instruction (+exec)
- Mode Agent (+exec)

Partie III

- Perspectives
- Q&A



Avant tout - Les outils indispensables

Team .NET



Visual Studio 2026

Team JAVA



IntelliJ IDEA

Abonnement Copilot

Free

A fast way to get started with GitHub Copilot.

\$0^{USD}

Get started

Open in VS Code

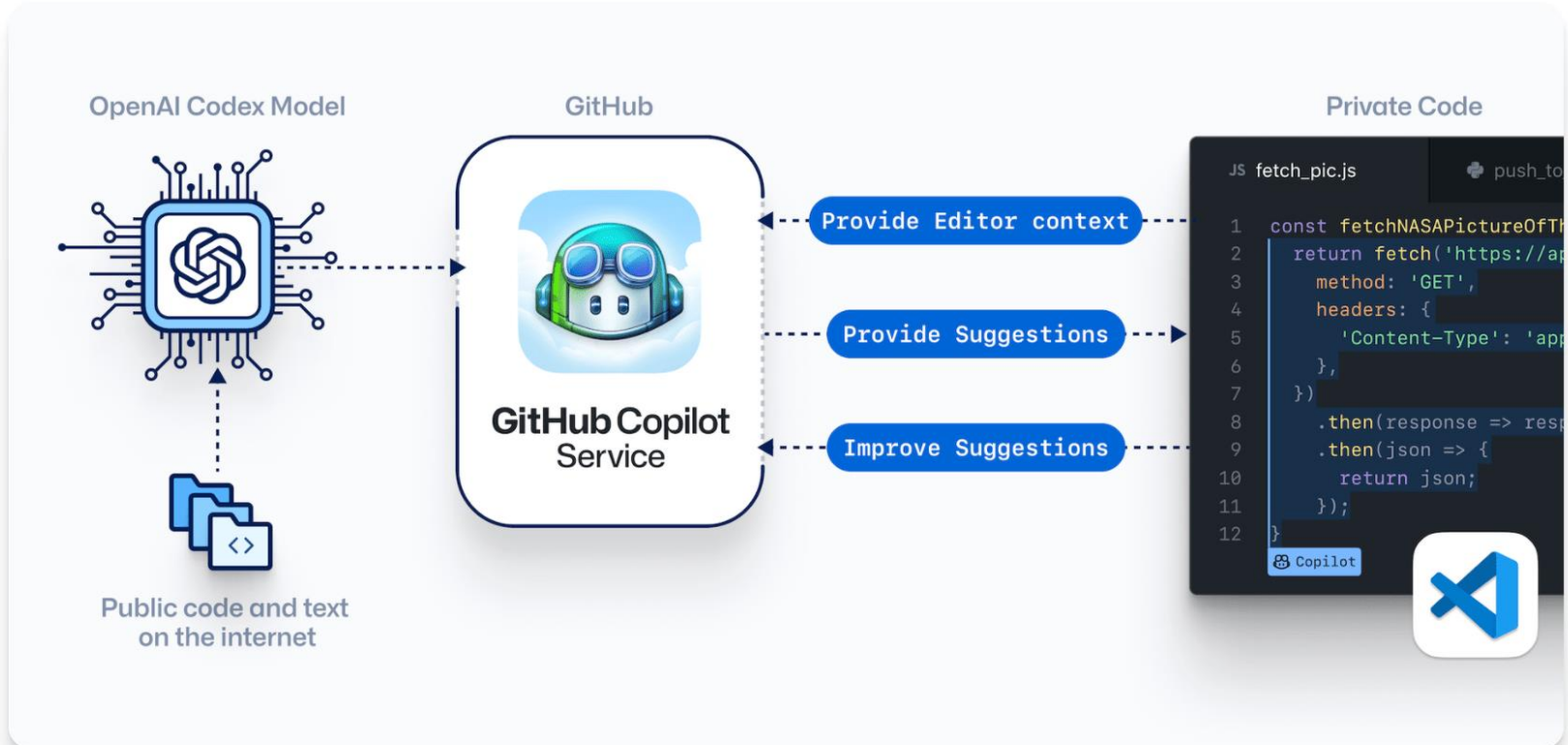
^ **What's included:**

- ✓ 50 agent mode or chat requests per month
- ✓ 2,000 completions per month
- ✓ Access to Haiku 4.5, GPT-4.1, and more

- 50 demandes en mode agent ou chat par mois
- 2 000 achèvements par mois
- Accès à Haiku 4.5, GPT-4.1...

Cet atelier se basera sur le modèle GPT4.1

Introduction – GitHub Copilot



Introduction – GitHub Copilot

Chat

GitHub Copilot Chat

créer une classe nommer LoginAction

créer une classe nommer LoginAction

GitHub Copilot

2 references

```
C# LoginAction.cs Apply
```

```
using AxaFrance.WebEngine;  
using AxaFrance.WebEngine.Web;  
using OpenQA.Selenium;  
  
namespace InstructionsDemo.Actions  
{  
    /// <summary>  
    /// Shared action to perform login  
    /// </summary>  
    public class LoginAction : SharedA  
    {  
        public override Variable[]? Ret
```

What additional methods or properties should be included in the 'LoginAction' class?

+ Active Document X

verifier si le message d'erreur affiche dans checkpoint

GPT-4.1

Code Completion

```
//créer une classe nommer LoginAction  
public class LoginAction : AxaFrance.We  
{  
    public override AxaFrance.WebEngine  
    public override void DoAction()  
{  
    // Navigates to the login page  
    Browser.Navigate().GoToUrl(GetF  
    var login = new PageModels.Page  
    login.UserName.SetValue(GetPar  
    login.UserName.SetSecure(GetPar  
    login.ButtonLogin.Click();  
}  
}  
public override bool DoCheckpoint()  
{  
    var login = new PageModels.Page  
    if (login.ErrorMessage.Exists(5  
    {  
        Information.AppendLine("Err  
        return false;  
    }  
    return true;  
}  
}
```

Bugs Correction

```
new Actions.LoginAction();
```

using OpenQA.Selenium.Interactions;
OpenQA.Selenium.Interactions.Actions
Generate type 'LoginAction'
Change 'Actions' to 'Action'.
Convert to top-level statements
Fix with Copilot
Suppress or configure issues

Mode Plan/Agent

Update "positive/negative" test terminology to "passing/nonPassing"

Overview: Update "positive/negative" test terminology to "passing/nonPassing"

Progress: 20%

Last Updated: 2025-12-02 15:25:57

Plan Steps

- Update C# backend entity properties from PositiveTests/NegativeTests to PassingTests/NonPassingTests
- Update Razor component comments from "positive/negative" to "passing/nonPassing"
- Update backend worker test generation logic to use new terminology
- Verify resource files already contain correct Passing/NonPassing keys
- Build and verify no compilation errors

Sécurité

- Chiffrement des données
- Stockage des prompts
- Audits & certifications
- Pentests

Confidentialité des données

- Données collectées
- Traitement des données

Propriété intellectuelle

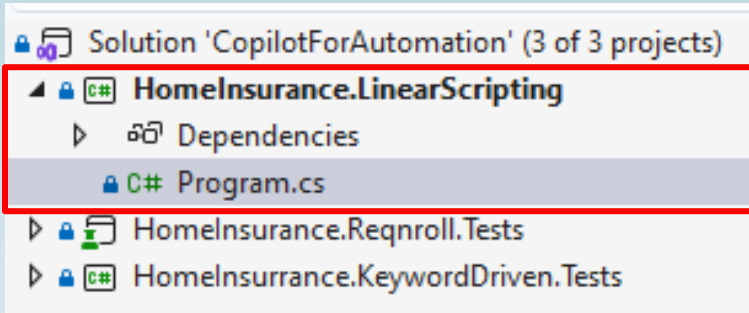
- Utilisation pour l'entraînement
- Propriété du code généré
- Filtrage des similitudes

Atelier 1 - Premier bas (10 min)

Team .NET

Cloner le repo:

<https://github.com/huaxing-yuan/CopilotForAutomation>



Team JAVA

Cloner le repo:

<https://github.com/josepharul82/project-jtia-linear.git>



<https://axafrance.github.io/webengine-dotnet/demo/home-insurance/>

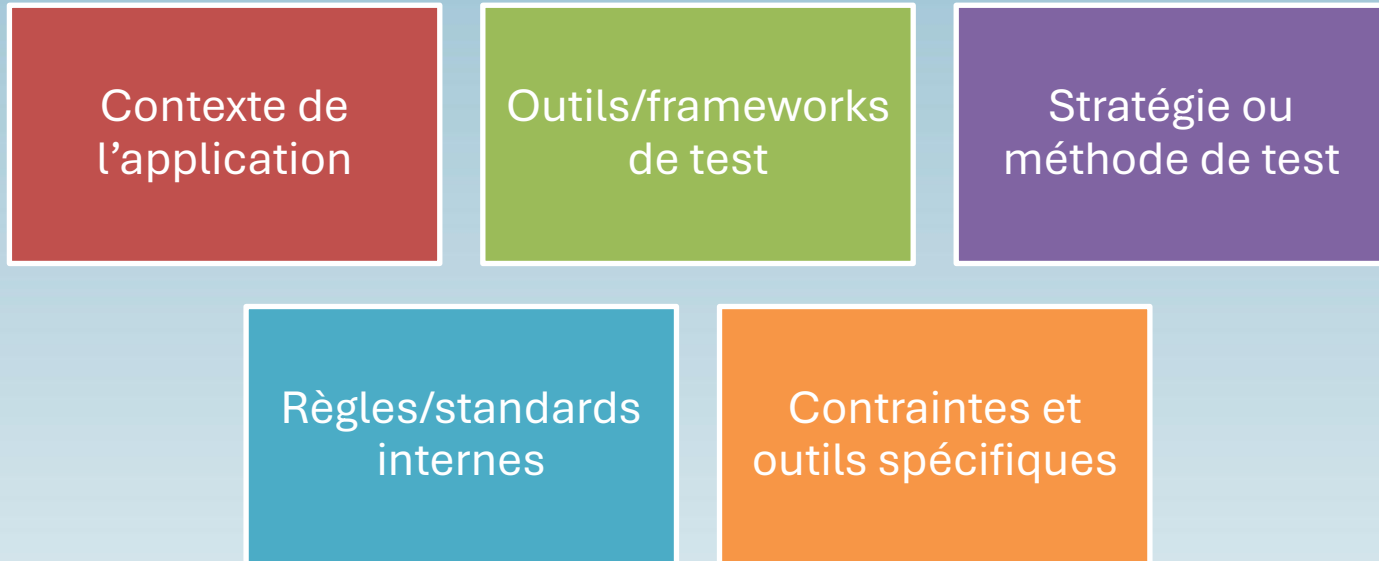
Ça fonctionne !

Mais nous allons tendre vers plus l'efficience

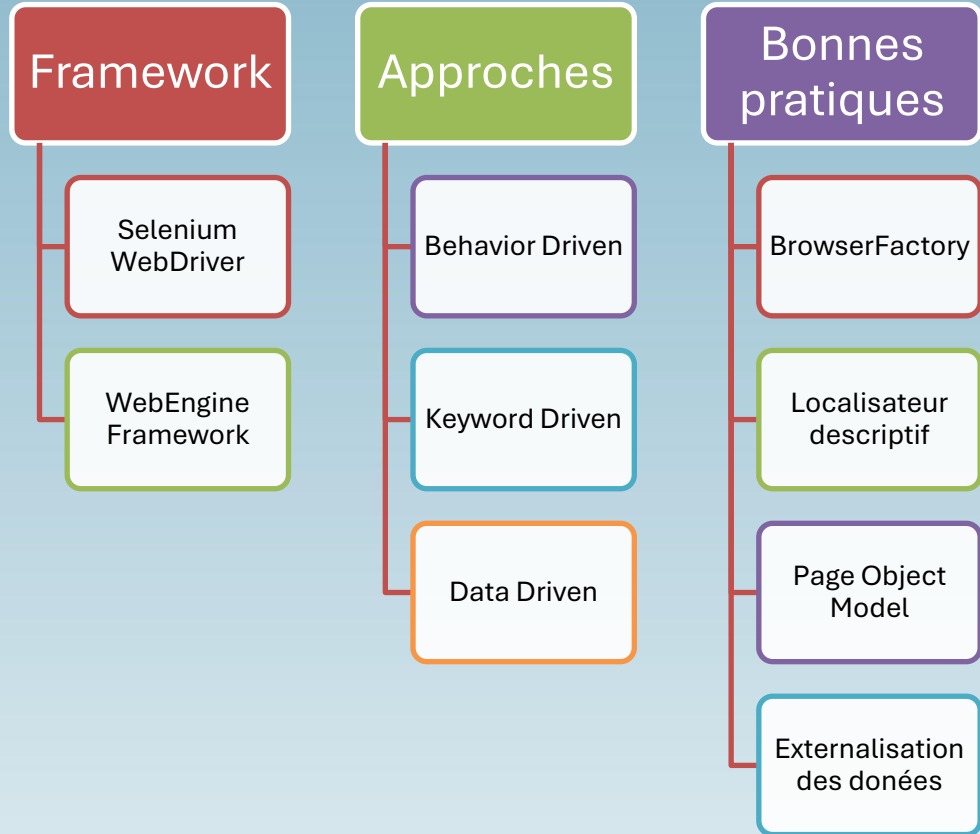
**En accompagnant Copilot avec un contexte précis
et des consignes adaptées.**

Contexte à fournir à GitHub Copilot pour des tests efficaces

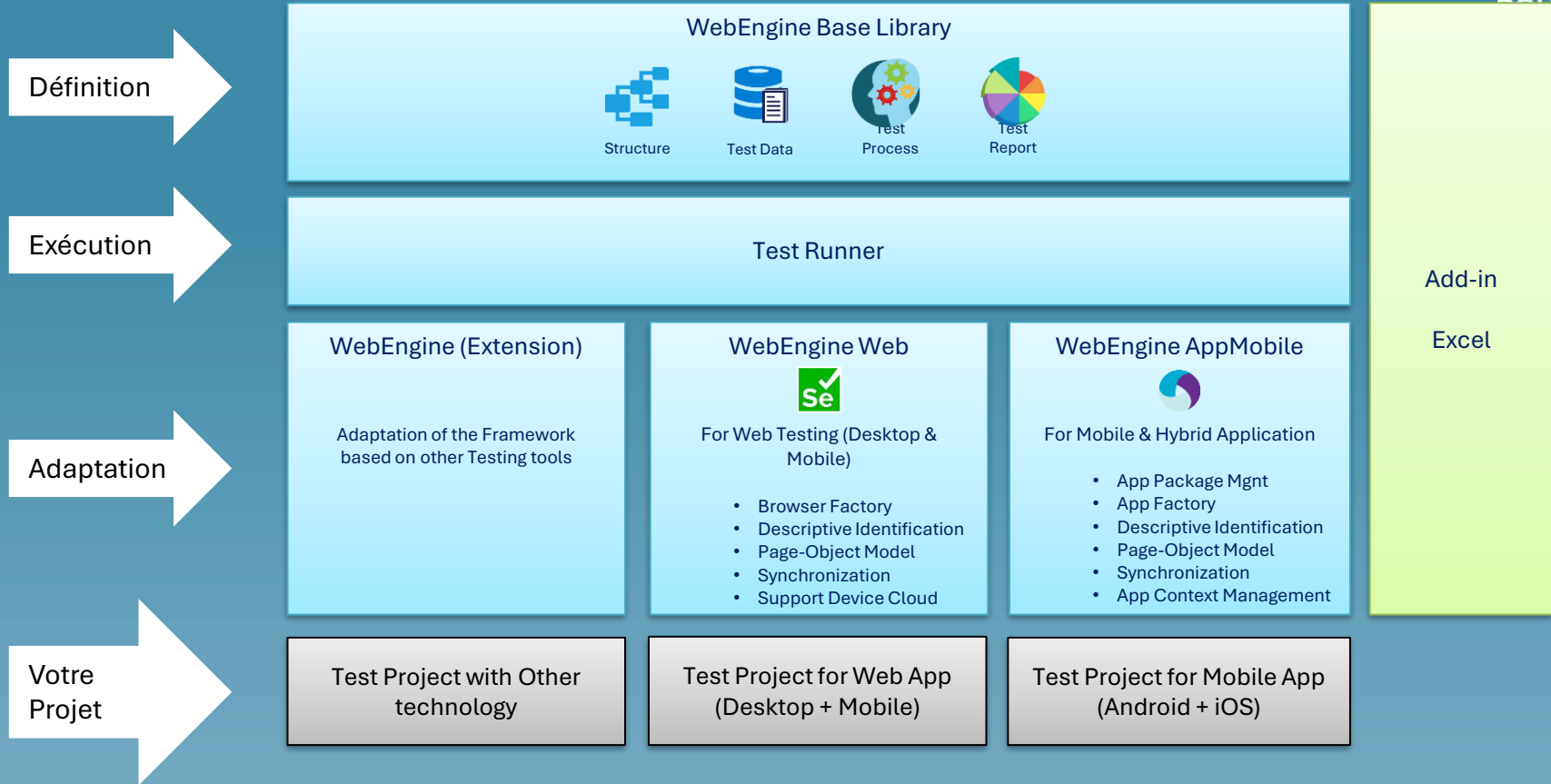
- Pour obtenir des suggestions précises et adaptées avec GitHub Copilot lors de l'écriture de tests, il est essentiel de lui fournir un maximum de contexte sur votre projet. Voici les principaux éléments à préciser avant de commencer :



Exemple : Framework de Test et Approche de test chez AXA



Architecture Framework WebEngine





Browser Factory

Browser Factory simplifie la gestion du WebDriver et vous assure d'exécuter votre script sur n'importe quelle plateforme : Windows, MacOS, Android et iOS.

En local ou à distance.

```
var driver = BrowserFactory.GetDriver(Platform, BrowserType);
```

App Factory

Similaire avec BrowserFactory mais pour les tests d'applications natives.



Element Description

Identification descriptive d'un élément WEB

Identifier des éléments via la combinaison d'attributs HTML ou de propriétés DOM.

Interaction synchronisée

Synchroniser automatiquement le script de test avec l'application et attend que l'objet soit chargé ou cliquable

Localisateurs disponibles :

Id, Name, TagName, ClassName, InnerText, LinkText, CssSelector, Xpath

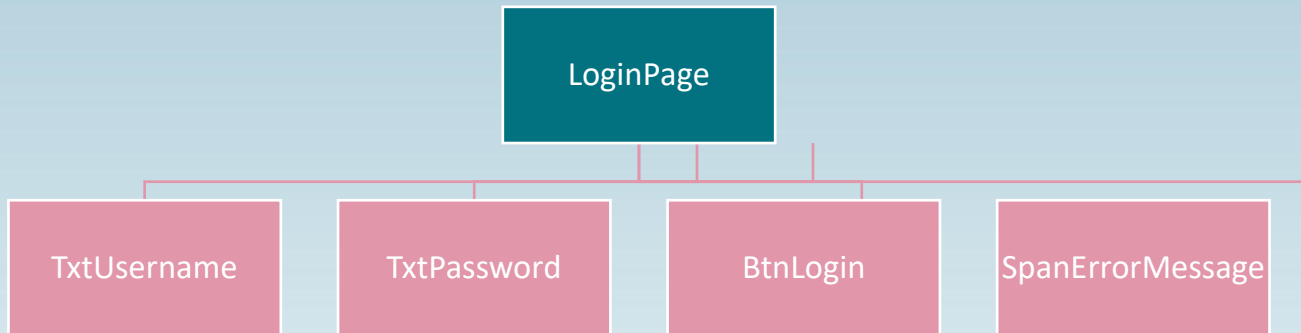
Si les localisateurs ci-dessus ne suffisent pas, vous pouvez aussi utiliser **HtmlAttributes**



Page-Object Model

Les éléments IHM utilisés dans un écran ou un page web peuvent être organisés dans des modèles d'objets représentant page.

- Le Page Object Model se charge d'identifier l'élément et de s'assurer de son interaction
- Le Script de test se focalise sur la logique du test.



The image shows a web application interface on the left and its Page Object Model (POM) implementation on the right. The web page is titled "AUTHENTIFICATION CENTRALISÉE" and contains a login form with fields for "Identifiant" (username) and "Mot de passe" (password), and a "Valider" button. The POM implementation is in C# and defines a `LoginPage` class that inherits from `PageModel`. It contains three `WebElementDescription` objects: `Login` (for the username input), `Password` (for the password input), and `ButtonValider` (for the validation button). Blue arrows connect the elements in the web page to their corresponding descriptions in the POM.

Web Page Elements:

- 1. Username input field: `<input type="text" name="user" id="user" maxlength="10" class="input">`
- 2. Password input field: `<input type="password" name="PASSWORD" id="password" maxlength="15" class="input">`
- 3. Validation button: `<input type="image" name="Valider" src="images/valider.gif" alt="Valider">`
- 4. Label "Identifiant" (points to the username input field)

POM Implementation:

```
namespace WebEngineDemo
{
    1 reference
    public class LoginPage : PageModel
    {
        0 references
        public LoginPage(WebDriver driver) : base(driver)
        {
            I
        }

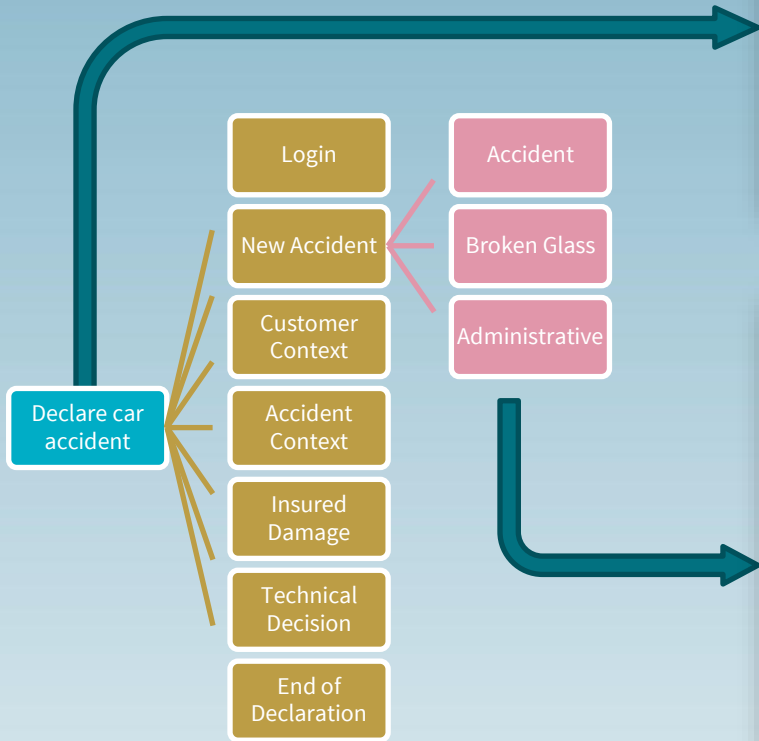
        0 references
        public WebElementDescription Login { get; set; } = new WebElementDescription
        {
            Id = "user"
        };

        0 references
        public WebElementDescription Password { get; set; } = new WebElementDescription
        {
            Id = "password"
        };

        0 references
        public WebElementDescription ButtonValider { get; set; } = new WebElementDescription
        {
            TagName = "input",
            Name = "Valider"
        };

        0 references
        public WebElementDescription ErrorMessage { get; set; } = new WebElementDescription
        {
            TagName = "message",
            TagName
        }
    }
}
```

Actions réutilisables - test piloté par mots-clés



```
2 references | Huaxing YUAN, 29 days ago | 1 author, 1 change
public class DeclarationSinistre_Auto : TestCaseWeb
{
    1 reference | Huaxing YUAN, 29 days ago | 1 author, 1 change
    public DeclarationSinistre_Auto()
    {
        TestSteps = new TestStep[]
        {
            new TestStep{ Action = nameof(SeConnecter)},
            new TestStep{ Action = nameof(NouveauSinistre_Contrat)},
            new TestStep{ Action = nameof(NouveauSinistreAuto_ContextClient)},
            new TestStep{ Action = nameof(NouveauSinistreAuto_DommageAssure)},
            new TestStep{ Action = nameof(NouveauSinistreAuto_DecisionTechnique)},
            new TestStep{ Action = nameof(FinDeclaration)},
        };
    }
}
```

Scénario de test (utilisant mots-clés action)

```
[Description("Authentication WAC")]
3 references | Huaxing YUAN, 29 days ago | 1 author, 1 change
public class SeConnecter : SharedActionWeb
{
    0 references | Huaxing YUAN, 29 days ago | 1 author, 1 change
    public override Variable[] RequiredParameters => new Variable[0];
    0 references | Huaxing YUAN, 29 days ago | 1 author, 1 change
    public override void DoAction()
    {
        //Environnement de Test => URL de l'application
        var environnement = GetParameter(ParameterList.ENVIRONNEMENT);
        var solarisURL = EnvironmentVariables.Current.GetValue($"URL_{environnement}");
        Browser.Navigate().GoToUrl(solarisURL);
        Page.LoginMac loginPage = new Page.LoginMac(Browser);


        //Login via username et mot de passe.
        string username = GetParameter(ParameterList.LOGIN);
        string password = GetParameter(ParameterList.PASSWORD);
        loginPage.TxtUserName.SendKeys(username);
        loginPage.TxtPassword.SendKeys(password);
        loginPage.BtnValider.Click();
        //Vérifier sans message d'erreur de connexion.
        if (loginPage.LblInformation.Exists(4))
        {
            var text = loginPage.LblInformation.GetText();
            if (text.Contains("renouveler"))
            {
                ActionResult = Result.CriticalError;
                Information.AppendLine("Connexion KO.");
                Information.AppendLine(text);
                Screenshot("Connexion");
                return;
            }
        }
        ActionResult = Result.Passed;
    }
}
```

Mots-clés d'action (implémentation)

Demo : Fournir à Copilot vos contextes

Moi:

Je donne tous les contextes pour l'approche et les éléments du framework WebEngine

 GitHub Copilot

> 1 reference

OK, tout est clair. Je suis prêt à vous aider à automatiser les tests avec Selenium, WebEngine, et l'approche BDD ReqnRoll, en respectant les consignes sur l'utilisation de **BrowserFactory**, **WebElementDescription**, et les Page Models. Indiquez le scénario ou la page à automatiser pour commencer.

Moi:

Des prompts simples de l'automatisation, et Copilot va bien travailler dans ce sens

Fournir à Copilot vos contextes

- Comment simplifier et accélérer le développement ?
- Comment éviter de fournir le contexte complet à chaque demande ?
- Comment partager le contexte avec les membres de l'équipe ?

Copilot Instruction

Copilot Instruction permet à l'utilisateur de définir des règles permanentes, des préférences ou un contexte personnalisé pour l'IA.

- Les contextes que l'on peut fournir dans les instructions incluent :
 - Style de code
 - Choix du framework
 - Conventions de nommage
 - ...

Context for Copilot - Test Automation

We are working on Test Automation using Selenium WebDriver customized with WebEngine Framework. Speak with me in French.

Framework repository: <https://github.com/AxaFrance/webengine-dotnet>

We use following approaches to write test cases:

- Linear Scripting : for unit or very simple scenarios. We translate test scenario to test scripts directly.
- Behavior-Driven Development : where the test scenarios are written in gherkin format with `Reqno11` (specflow which is no more maintained).
- Keyword-Driven Testing : where the test scenario are written using keywords that represent actions or verifications in the application under test.

You will analyze the structure of the projet to determine the approche, tell me which approach is used. For linear scripting DO NOT use SharedActions or TestCaseWeb/TestCaseApp classes.

When writing test case, follow these guidelines for every approaches :

- Do not use hardcoded locators such as xpath or css selectors in test script when possible. Use ElementDescription instead.
- Add Web elements into PageObject Models and refer it in test actions.
- Apply PageObject Model design pattern to separate UI element locators to test script.
- when suggest test script, use native functions from WebEngine Framework if possible.
- **DO NOT** create ElementDescription directly in SharedActions, add them to appropriate PageModels

For Keyword-Driven, you should use SharedActions to implemented reusable action groups. For Gherkin, you can also use shared action inside step definition to maximum reuse.

- Use parameters in actions, and use parameter list to store all parameters used in the test case.
- In actions, perform actions using PageObject Models and their elements.

For a Keyword-Driven test project, it should be structured as follow based on current active project, using current projet's namespace.

- folder PageModels: store all page object models
- folder Actions: store all SharedActions (which should be reused in different test cases)
- folder TestCases: store all test scenarios
- folder TestData: store all test data files, such as XML or Excel files folders should be inside project folder, not the solution folder.

📁 Votre projet

📁 .github

📄 copilot-instructions.md

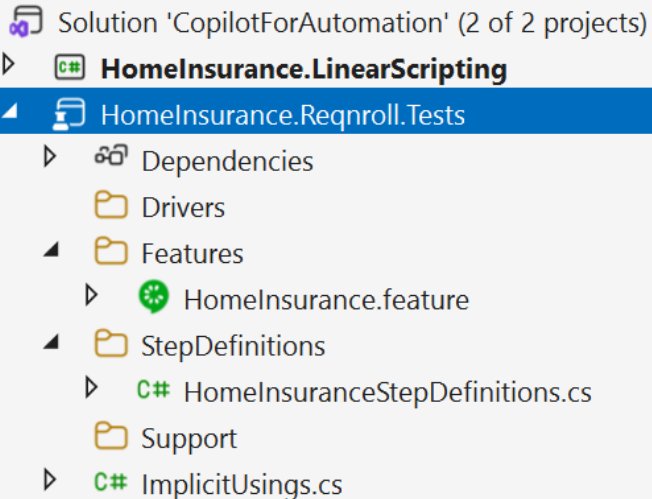
C#: <https://github.com/AxaFrance/webengine-dotnet/blob/main/copilot-instructions.md>

JAVA: <https://github.com/josepharul82/project-jtia-bdd/blob/main/.github/copilot-instructions.md>

Atelier 3 – Automatiser un parcours bout en bout (20 min)

- Souscription d'une assurance habitation pour un appartement
- Gherkin
- Branch master = projet vide, branche correction = projet généré

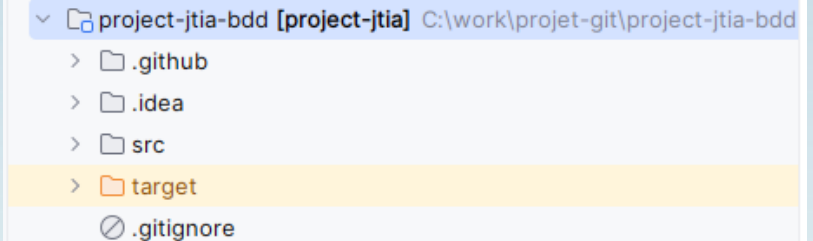
Team .NET



Team JAVA

Cloner le repo:

<https://github.com/josepharul82/project-jtia-bdd.git>

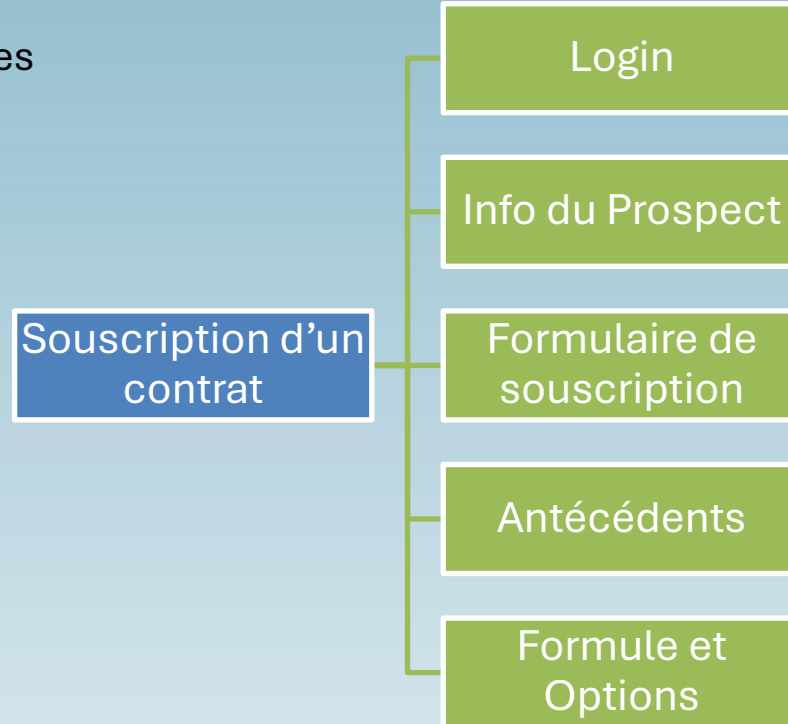


Atelier 3 – Automatiser un parcours bout en bout (20 min)

- Etape 1: Copier le fichier d'instruction dans votre projet
- Etape 2: Implémenter le page Login
- Etape 3: Implémenter le page Prospect avec le mode AGENT
- Etape 4: Implémenter le page Formulaire de souscription

Démo – Test Piloté par Mots-clés (avec Mode Agent)

- Découpage de scénario en mots-clés
- Piloté par données
- Génération des Jeux de données

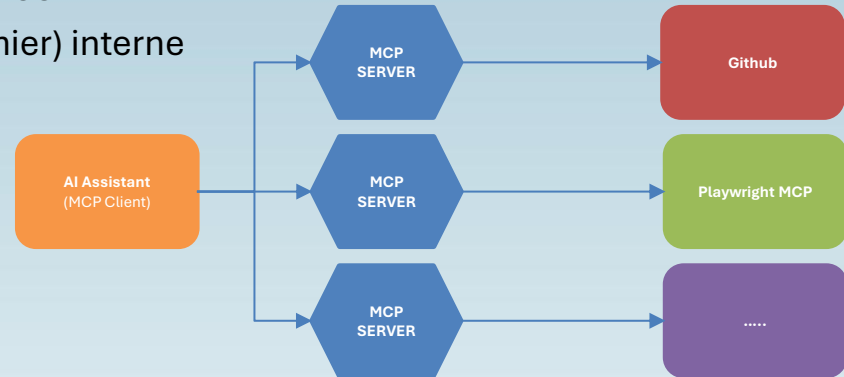


Perspective – Serveur MCP

- Copilot = assistant de programmation local
- Avec MCP: Copilot = assistant intégré dans votre écosystème
 - Un serveur MCP peut exposer :
 - Des **outils** (“exécuter un test”, “générer un fichier”, “déployer sur un environnement”)
 - Des **ressources** (logs, infos système, fichiers, endpoints)
 - Des **prompts personnalisés** (guides internes, conventions, règles QA)...

– Serveur MCP dans le domaine automatisation de test


- Récupération d’info depuis un système (Web, API, Fichier) interne
- Connecter avec Selenium / Playwright, ...
- Intégration avec les autres outils de Build ou DevOps
- Lancer les tests, mettre à jour des jeux de données















MERCI

ANNEXE



 **CopilotForAutomation** Public Pin Watch 0

master 1 Branch 0 Tags Add file Code

	Local	Codespaces
 huaxing-yuan Add project files.		
 Homelnsurance.LinearScripting Add project files.	 Clone ?	
 Homelnsurance.Reqnroll.Tests Add project files.	HTTPS SSH GitHub CLI	
 Homelnsurance.KeywordDriven.Tests Add project files.	<input type="text" value="https://github.com/huaxing-yuan/CopilotForAutor"/> 	
 .gitattributes Add .gitattributes, .gitignore, and README.md.	Clone using the web URL.	
 .gitignore Add .gitattributes, .gitignore, and README.md.	 Open with GitHub Desktop	
 CopilotForAutomation.slnx Add project files.	Open with Visual Studio	
 README.md Add .gitattributes, .gitignore, and README.md. 28 minutes ago	 Download ZIP	

josepharul82 / project-jtia-bdd

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

project-jtia-bdd Public Pin Watch 0 Fork 0 Star 0

main 1 Branch 0 Tags Add file Code About

Josepharul82 add home details page cb28a51 · 16 hours ago 3 Commits

.github	initial commit	16 hours ago
src/test	add home details page	16 hours ago
.gitignore	initial commit	16 hours ago
README.md	initial commit	16 hours ago
pom.xml	initial commit	16 hours ago
prompt.txt	initial commit	16 hours ago

About
No description, website, or topics provided.
Readme
Activity
0 stars
0 watching
0 forks

Releases
No releases published
[Create a new release](#)

josepharul82 / project-jtia-linear

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

project-jtia-linear Public Pin Watch 0 Fork 0 Star 0

main 1 Branch 0 Tags Add file Code About

Josepharul82 add home detail page 1a59730 · 16 hours ago 3 Commits

src/test/java/fr/axa/automation/sample	add home detail page	16 hours ago
.gitignore	initial commit	17 hours ago
README.md	initial commit	17 hours ago
pom.xml	initial commit	17 hours ago
prompt.txt	initial commit	17 hours ago

About
No description, website, or topics provided.
Readme
Activity
0 stars
0 watching
0 forks

Releases
No releases published