

.Net MAUI

Introduction

.NET MAUI (Multi-platform App UI) est le framework de Microsoft pour développer des applications natives multiplateformes en C#.

Il permet de cibler Android, iOS, Windows et MacOS avec un seul et même projet.

✅ Successeur direct de Xamarin.Forms.

Objectif

- Créer des applications mobiles et desktop avec un seul code source.
- Utiliser XAML pour l'interface et C# pour la logique métier.
- Déployer facilement sur plusieurs plateformes.

Prérequis

- Windows 11 (pour Android/Windows)
- Visual Studio 2022 17.3+ avec workload .NET MAUI installé.
- SDK Android, Xcode (pour iOS/MacOS).
- Compte développeur Apple (nécessaire pour déployer sur iOS).

Installation

Installer Visual Studio

- Télécharger Visual Studio Community : <https://visualstudio.microsoft.com>

Workload à installer

- "Mobile development with .NET" (inclut .NET MAUI)
-

Vérification

Ouvrir un terminal :

- `dotnet workload list`

.NET MAUI doit apparaître dans la liste.

Créer un projet MAUI

Commande CLI

`dotnet new maui -n NomDuProjet`

Via Visual Studio

- Créer un nouveau projet → .NET MAUI App → Choisir le type (Blazor, MVU, classique)

Concepts clés

XAML	Définir l'interface utilisateur avec balisage XML
Binding	Lier les données au front-end sans code lourd
MVVM	Architecture typique (Model-View-ViewModel)
Cross-platform	Une seule base de code pour plusieurs OS

.Net MAUI

Structure d'un projet .NET MAUI

MainPage.xaml	Page principale (UI avec XAML)
MainPage.xaml.cs	Code-behind en C#
App.xaml	Définition globale (styles, ressources)
App.xaml.cs	Logique d'application
Platforms/	Spécificités par plateforme (Android, iOS, Windows, MacCatalyst)
Resources/	Images, polices, styles partagés

Exemple simple

Interface (MainPage.xaml)

```
<VerticalStackLayout>
  <Label Text="Bonjour, .NET MAUI!"FontSize="32"HorizontalOptions="Center" />
  <Button Text="Cliquez-moi"Clicked="OnButtonClicked" />
</VerticalStackLayout>
```

Logique (MainPage.xaml.cs)

```
private void OnButtonClicked(object sender, EventArgs e)
{
    (sender as Button).Text = "Merci d'avoir cliqué!";
}
```

Déployer l'application

- Sur Windows : exécuter localement.
- Sur Android : utiliser l'émulateur Android ou un téléphone branché en USB.
- Sur iOS : nécessitera un Mac ou un Mac dans le cloud.

Bonnes pratiques

- ✓ Respecter le modèle MVVM pour séparer logique métier et interface.
- ✓ Utiliser BindingContext pour connecter les ViewModels aux vues.
- ✓ Mutualiser au maximum les ressources et styles.
- ✓ Tester les comportements spécifiques à chaque plateforme.

Limitations connues

- Déploiement sur iOS demande macOS (Xcode requis).
- Certaines fonctionnalités peuvent varier légèrement selon la plateforme.
- .NET MAUI est encore jeune, certaines librairies tierces sont en cours de migration.