Fiche de Procédure : Les Bases de .NET MAUI

1. Qu'est-ce que .NET MAUI ?

- .NET MAUI (Multi-platform App UI) est un framework multiplateforme développé par Microsoft permettant de créer des applications mobiles et de bureau natif avec C# et XAML, à partir d'une base de code partagée unique.
- Avec .NET MAUI, vous pouvez cibler Android, iOS, macOS et Windows dans un seul projet, simplifiant le développement et la maintenance des applications multiplateformes.

2. Fonctionnalités principales

- Projet unique : Un seul projet partagé pour toutes les plateformes, avec gestion centralisée des ressources, du manifeste d'application et du point d'entrée.
- Contrôles d'interface riches : Large collection de contrôles pour afficher des données, lancer des actions, naviguer, sélectionner, etc.
- Moteur de mise en page : Création de pages et de layouts avancés pour des interfaces adaptatives.
- Navigation avancée : Plusieurs types de pages (navigation, tiroirs, onglets…) pour des expériences utilisateur enrichies1.
- Liaison de données : Support du data binding pour des architectures MVVM élégantes et maintenables.
- API multiplateforme : Accès aux fonctionnalités natives de l'appareil (GPS, capteurs, batterie, réseau, presse-papiers, fichiers, etc.) via des API unifiées.
- Fonctionnalités graphiques : Canvas de dessin, manipulation d'images et de formes, transformations graphiques.
- Rechargement à chaud : Modifiez le code ou le XAML et visualisez instantanément le résultat sans recompiler l'application.
- Intégration avec les outils .NET : Utilisation de Visual Studio, NuGet, C# moderne, tests unitaires, etc.

3. Exemples de fonctionnalités natives accessibles

- Accès aux capteurs (accéléromètre, gyroscope, boussole)
- Vérification de la connectivité réseau
- Informations sur l'appareil et la batterie

- Sélection et partage de fichiers
- Notifications, presse-papiers, géolocalisation
- Authentification web, ouverture de liens, envoi de SMS ou d'e-mails

4. Structure d'un projet .NET MAUI

- /Platforms/ : Dossiers spécifiques à chaque plateforme (Android, iOS, MacCatalyst, Windows)
- /Resources/ : Images, polices, fichiers de configuration partagés
- /App.xaml & App.xaml.cs : Définition de l'application, ressources globales, navigation
- /MainPage.xaml & MainPage.xaml.cs : Page principale (interface et logique)
- /ViewModels/ : (optionnel) Logique métier et liaison de données (MVVM)
- /Services/ : (optionnel) Accès aux API, gestion des données, etc.

5. Exemple minimal d'application

App.xaml.cs

```
public partial class App : Application
{
    public App()
    {
        InitializeComponent();
        MainPage = new MainPage();
    }
}
```

MainPage.xaml

```
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             x:Class="MonApp.MainPage">
    <VerticalStackLayout>
        <Label Text="Bonjour, .NET MAUI !"
               VerticalOptions="Center"
               HorizontalOptions="Center" />
        <Button Text="Cliquez-moi" Clicked="OnButtonClicked"/>
    </VerticalStackLayout>
```

```
</ContentPage>
```

MainPage.xaml.cs

```
private void OnButtonClicked(object sender, EventArgs e)
{
    DisplayAlert("Bienvenue", "Vous avez cliqué sur le bouton.",
"OK");
}
```

6. Bonnes pratiques et astuces

- Utilisez la liaison de données (data binding) et le pattern MVVM pour séparer l'interface de la logique métier.
- Centralisez les ressources (images, polices, chaînes) dans le dossier `/Resources/`.
- Profitez du rechargement à chaud pour accélérer le développement d'UI.
- Testez régulièrement sur plusieurs plateformes pour garantir la cohérence de l'expérience utilisateur.
- Utilisez les API multiplateformes pour accéder aux fonctionnalités natives sans écrire de code spécifique à chaque OS.

7. Ressources pour progresser

- Documentation officielle Microsoft (.NET MAUI)
- Tutoriels, guides et exemples sur Microsoft Learn
- Communauté GitHub, forums et Stack Overflow
- Exemples de projets open source utilisant .NET MAUI

Astuce : .NET MAUI permet de mutualiser un maximum de code et d'UI pour cibler plusieurs plateformes, tout en gardant la possibilité d'accéder aux fonctionnalités spécifiques de chaque appareil si besoin.