

AppShell.xaml

```
<?xml version="1.0" encoding="UTF-8" ?>
<Shell
    x:Class="TPMaui1.AppShell"
    xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:local="clr-namespace:TPMaui1"
    xmlns:views="clr-namespace:TPMaui1.Views"
    Shell.FlyoutBehavior="Disabled">

    <TabBar>
        <ShellContent
            Title="Notes"
            ContentTemplate="{DataTemplate local:MainPage}"
            Icon="icon_notes" />
        <ShellContent
            Title="A Propos"
            ContentTemplate="{DataTemplate local:APropos}"
            Icon="icon_about" />
        <ShellContent
            Title="Infos"
            ContentTemplate="{DataTemplate views:vPhone}"
            Icon="icon_info" />
        <ShellContent
            Title="Géo"
            ContentTemplate="{DataTemplate views:vGeolocalisation}"
            Icon="icon_info" />
    </TabBar>

</Shell>
```

MainPage.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="TPMaui1.MainPage">

    <ScrollView>
        <VerticalStackLayout
            Spacing="25"
            Padding="30,0"
            VerticalOptions="Center">

            <Image
                Source="dotnet_bot.png"
                SemanticProperties.Description="Cute dot net bot waving hi to
you!"
                HeightRequest="200"
                HorizontalOptions="Center" />

            <Label
                Text="TP n°1"
                SemanticProperties.HeadingLevel="Level1"
                FontSize="32"
                HorizontalOptions="Center" />

            <Label
                Text="TP avec .NET Multi-platform App UI"
                SemanticProperties.HeadingLevel="Level2"
```

```

App U I"
        SemanticProperties.Description="Welcome to dot net Multi platform
        FontSize="18"
        HorizontalOptions="Center" />

        <Button
            x:Name="CounterBtn"
            Text="Click me"
            SemanticProperties.Hint="Counts the number of times you click"
            Clicked="OnCounterClicked"
            HorizontalOptions="Center" />

    </VerticalStackLayout>
</ScrollView>

</ContentPage>

```

APropos.xaml

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="TPMau1.APropos"
    Title="APropos">
    <VerticalStackLayout Spacing="10" Margin="10">
        <Image Source="logo3il.png"
            SemanticProperties.Description="Cette application est une
production 3iL"
            HeightRequest="64" HorizontalOptions="CenterAndExpand"/>
        <Label FontSize="22" FontAttributes="Bold" Text="Notes 3iL"
            VerticalOptions="End" TextColor="{StaticResource Secondary}"/>
        <Label FontSize="22" Text="v1.0" VerticalOptions="End"
            TextColor="{StaticResource Secondary}"/>
        <Label Text="Application XAML/C# écrite en .NET MAUI."
            TextColor="{StaticResource Primary}"/>
        <Button Text="En savoir plus sur 3iL" Clicked="APropos_Clicked" />
    </VerticalStackLayout>
</ContentPage>

```

APropos.xaml.cs

```

namespace TPMau1;

public partial class APropos : ContentPage
{
    public APropos()
    {
        InitializeComponent();
    }

    async private void APropos_Clicked(object sender, EventArgs e)
    {
        await Launcher.Default.OpenAsync("https://www.3il-ingenieurs.fr/");
    }
}

```

vPhone.xaml

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="TPMaui1.Views.vPhone"
    Title="Info téléphone">
    <VerticalStackLayout Spacing="20" Margin="10" >
        <Label
            Text="Info Téléphone " FontSize="22"
            VerticalOptions="Center"
            HorizontalOptions="Center" />
        <Label
            Text="Welcome to .NET MAUI!"
            x:Name="lbPhoneInfo"
            VerticalOptions="Center" HeightRequest="240" FontSize="22"
            HorizontalOptions="Center" />
    </VerticalStackLayout>
    <ContentPage.ToolbarItems>
        <ToolbarItem Text="Ecran" Clicked="Ecran_Clicked"
            IconImageSource="{FontImage Glyph='+', Color=White, Size=22}" />
        <ToolbarItem Text="Batterie" Clicked="Batterie_Clicked"
            IconImageSource="{FontImage Glyph='+', Color=White, Size=22}" />
    </ContentPage.ToolbarItems>
</ContentPage>

```

vPhone.xaml.cs

```

using Microsoft.Maui.Devices;

namespace TPMaui1.Views;

public partial class vPhone : ContentPage
{
    public vPhone()
    {
        Routing.RegisterRoute($"{nameof(vEcran)}", typeof(vEcran));
        Routing.RegisterRoute($"{nameof(vBatterie)}", typeof(vBatterie));
        InitializeComponent();
        Phone_Info();
    }

    public void Phone_Info()
    {
        System.Text.StringBuilder sb = new System.Text.StringBuilder();
        sb.AppendLine($"Modèle: {DeviceInfo.Current.Model}");
        sb.AppendLine($"Constructeur: {DeviceInfo.Current.Manufacturer}");
        sb.AppendLine($"Plateform: {DeviceInfo.Current.Platform}");
        sb.AppendLine($"version: {DeviceInfo.VersionString}");
        sb.AppendLine($"Nom: {DeviceInfo.Name}");
        sb.AppendLine($"Matériel: {Get_Idiom()}");
        lbPhoneInfo.Text = sb.ToString();
    }

    public String Get_Idiom()
    {
        if (DeviceInfo.Current.Idiom == DeviceIdiom.Desktop)
            return "PC";
        if (DeviceInfo.Current.Idiom == DeviceIdiom.Tablet)
            return "Tablette";
        if (DeviceInfo.Current.Idiom == DeviceIdiom.Phone)
            return "Téléphone";
        if (DeviceInfo.Current.Idiom == DeviceIdiom.TV)

```

```

        return "Télévision";
    return "Inconnu";
}

private async void Ecran_Clicked(object sender, EventArgs e)
{
    await Shell.Current.GoToAsync(nameof(vEcran));
}

private async void Batterie_Clicked(object sender, EventArgs e)
{
    await Shell.Current.GoToAsync(nameof(vBatterie));
}
}

```

vBatterie.xaml

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="TPMaui1.Views.vBatterie"
    Title="Info Batterie">
    <VerticalStackLayout>
        <Label
            Text="Alimentation!"
            VerticalOptions="Center" HorizontalOptions="Center"
            HeightRequest="30" FontSize="22" TextColor="{StaticResource Primary}"
        />
        <Button Text="Type alimentation" Clicked="Alimentation_click"
            HeightRequest="50" FontSize="22" TextColor="White"/>
        <Label
            Text=""
            x:Name="lbAlim"
            HorizontalOptions="Center" HeightRequest="30" FontSize="20"
            TextColor="{StaticResource Secondary}" />

        <Label
            Text="..." x:Name="lbEtat"
            VerticalOptions="Center" HorizontalOptions="Center"
            HeightRequest="80" FontSize="22" TextColor="{StaticResource
Secondary}" />
    </VerticalStackLayout>
</ContentPage>

```

vBatterie.xaml.cs

```

namespace TPMaui1.Views;

public partial class vBatterie : ContentPage
{
    public vBatterie()
    {
        InitializeComponent();
        // Définition des événement
        Battery.Default.BatteryInfoChanged += Battery_InfoChanged;
        Battery.Default.EnergySaverStatusChanged += Battery_ModeEcoChanged;
    }
}

```

```

private async void Battery_ModeEcoChanged(object sender,
EnergySaverStatusChangedEventArgs e)
{
    // Événement lié au passage (ou en fin) en économie d'énergie
    if (Battery.Default.EnergySaverStatus == EnergySaverStatus.On)
    {
        await DisplayAlert("Alerte", "Passage en économie d'énergie", "OK");
    }
    if (Battery.Default.EnergySaverStatus == EnergySaverStatus.Off)
    {
        await DisplayAlert("Info", "Mode économie d'énergie arrêté", "OK");
    }
}
public void Battery_InfoChanged(object sender, BatteryInfoChangedEventArgs e)
{
    String sState = "";
    if (e.State == BatteryState.Full) { sState = "Batterie pleine"; }
    if (e.State == BatteryState.Charging) { sState = "Batterie en charge"; }
    if (e.State == BatteryState.Discharging) { sState = "Batterie déchargée"; }
}
    if (e.State == BatteryState.NotCharging) { sState = "Batterie non en
charge"; }
    if (e.State == BatteryState.NotPresent) { sState = "Batterie absente"; }
    lbEtat.Text = sState + Environment.NewLine;
    lbEtat.Text += "Niveau batterie: " + (e.ChargeLevel *
100).ToString("0.0") + "%";
}

private void Alimentation_click(object sender, EventArgs e)
{
    lbAlim.Text = TypeAlimentation();
}
private String TypeAlimentation()
{
    switch (Battery.Default.PowerSource)
    {
        case (BatteryPowerSource.AC):
            return "Chargeur 220 V";
        case (BatteryPowerSource.Battery):
            return "Sur batterie";
        case (BatteryPowerSource.Usb):
            return "Sur USB";
        case (BatteryPowerSource.Wireless):
            return "Par Sans fil";
    }
    return "inconnu";
}
}

```

vEcran.xaml

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
x:Class="TPMau1.Views.vEcran"
Title="Info Ecran">
<VerticalStackLayout Spacing="20" Margin="10" >
<Label
Text="Info Ecran"
x:Name="lbInfo"
HorizontalOptions="Center" HeightRequest="30" FontSize="22"
TextColor="{StaticResource Primary}" />

```

```

        <Label
            Text=""
            x:Name="lbInfoEcran"
            HorizontalOptions="Center" HeightRequest="240" FontSize="22"
            TextColor="{StaticResource Secondary}" />
    </VerticalStackLayout>

</ContentPage>

```

vEcran.xaml.cs

```

namespace TPMau1.Views;

public partial class vEcran : ContentPage
{
    public vEcran()
    {
        InitializeComponent();

        lbInfoEcran.Text = "Résolution: " +
            DeviceDisplay.Current.MainDisplayInfo.Height.ToString() + " x " +
            DeviceDisplay.Current.MainDisplayInfo.Width.ToString() +
            Environment.NewLine;
        lbInfoEcran.Text += "Densité: " +
            DeviceDisplay.Current.MainDisplayInfo.Density.ToString() + Environment.NewLine;
        lbInfoEcran.Text += "Orientation: " +
            DeviceDisplay.Current.MainDisplayInfo.Orientation.ToString() +
            Environment.NewLine;
        lbInfoEcran.Text += "Rotation: " +
            DeviceDisplay.Current.MainDisplayInfo.Rotation.ToString() + Environment.NewLine;
        lbInfoEcran.Text += "V. Refresh: " +
            DeviceDisplay.Current.MainDisplayInfo.RefreshRate.ToString() +
            Environment.NewLine;
    }
}

```

vGeolocalisation.xaml

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="TPMau1.Views.vGeolocalisation"
    Title="Géolocalisation">
    <VerticalStackLayout>
        <Label
            Text="Welcome to .NET MAUI!"
            VerticalOptions="Center"
            HorizontalOptions="Center" />
        <Button Text="Me géolocaliser" Clicked="Geolocalisation_click"
            HeightRequest="50" FontSize="22" TextColor="White"
            x:Name="btLocaliser"/>
        <Button Text="Annuler" Clicked="Annulation_click"
            HeightRequest="50" FontSize="22" TextColor="White"
            x:Name="btAnnuler"/>
        <Label
            Text=""
            x:Name="lblongitude"
            HorizontalOptions="Center" HeightRequest="50" FontSize="22"
            TextColor="{StaticResource Secondary}" />
        <Label
            Text=""

```

```

        x:Name="lblatitude"
        HorizontalOptions="Center" HeightRequest="50" FontSize="22"
        TextColor="{StaticResource Secondary}" />
    <Label
        Text=""
        x:Name="lbaltitude"
        HorizontalOptions="Center" HeightRequest="50" FontSize="22"
        TextColor="{StaticResource Secondary}" />
</VerticalStackLayout>
</ContentPage>

```

vGeolocalisation.xaml.cs

```

namespace TPMau1.Views;

public partial class vGeolocalisation : ContentPage
{
    private CancellationTokenSource _cancelTokenSource; // Token qui représente
    la requete de localisation en cours
    private bool _isCheckingLocation; // Varianle qui sera à true lorsqu'une
    demande de géolocalisation est en cours

    public vGeolocalisation()
    {
        InitializeComponent();
        btAnnuler.IsEnabled = false;
    }
    private async void Geolocalisation_click(object sender, EventArgs e)
    {
        await GetCurrentLocation();
    }
    public async Task GetCurrentLocation()
    {
        btAnnuler.IsEnabled = true;
        btLocaliser.IsEnabled = false;
        lblongitude.Text = "longitude: ";
        lblatitude.Text = "latitude: ";
        lbaltitude.Text = "altitude: ";
        try
        {
            _isCheckingLocation = true;
            GeolocationRequest gr = new
            GeolocationRequest(GeolocationAccuracy.Medium, TimeSpan.FromSeconds(10));
            _cancelTokenSource = new CancellationTokenSource();
            Location loc = await Geolocation.Default.GetLocationAsync(gr,
            _cancelTokenSource.Token);
            if (loc != null)
            {
                lblongitude.Text = "longitude: " +
                loc.Longitude.ToString("0.0000");
                lblatitude.Text = "latitude: " + loc.Latitude.ToString("0.0000");
                lbaltitude.Text = "altitude: " + (loc.Altitude.ToString() ??
                "Null");
            }

            btAnnuler.IsEnabled = false;
            btLocaliser.IsEnabled = true;
        }
        catch (Exception ex) { }
        finally { _isCheckingLocation = false; }
    }
}

```

```

private void Annulation_click(object sender, EventArgs e)
{
    if (_isCheckedLocation && _cancelTokenSource != null &&
_cancelTokenSource.IsCancellationRequested == false)
        _cancelTokenSource.Cancel();
    btAnnuler.IsEnabled = false;
    btLocaliser.IsEnabled = true;
}
}

```

AndroidManifeste.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
    <application android:allowBackup="true" android:icon="@mipmap/appicon"
android:roundIcon="@mipmap/appicon_round"
android:supportsRtl="true"></application>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.BATTERY_STATS" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-feature android:name="android.hardware.location" android:required="false"
/>
    <uses-feature android:name="android.hardware.location.gps"
android:required="false" />
    <uses-feature android:name="android.hardware.location.network"
android:required="false" />
</manifest>

```