# Utiliser Xamarin.Essentials pour récupérer le niveau de batterie

# Démonstration 1 du module 8

|  |
| --- |
| Manipuler les différents types de layouts (StackLayout, RelativeLayout et GridLayout) et comprendre les propriétés d’alignement et de répartition de l’espace vide |

Déroulement

* Effectuer un clic droit sur la solution et sélectionner **Gérer les packages NuGet pour la solution**
* Cliquer sur l’onglet **Parcourir** et cocher la case **Inclure la version préliminaire**
* Rechercher le package **Xamarin.Essentials** et installer celui-ci pour tous les projets
* Ouvrir le fichier **MainActivity.cs** du projet Android et ajouter la ligne suivante dans la méthode OnCreate :

base.OnCreate(savedInstanceState);

Xamarin.Essentials.Platform.Init(this, savedInstanceState);

* Ajouter la méthode suivante dans la MainActivity :

public override void OnRequestPermissionsResult(int requestCode, string[] permissions, [GeneratedEnum] Android.Content.PM.Permission[] grantResults)

{

Xamarin.Essentials.Platform.OnRequestPermissionsResult(requestCode, permissions, grantResults);

base.OnRequestPermissionsResult(requestCode, permissions, grantResults);

}

* Ajouter un Label nommé « Orientation » dans la page MainPage.xaml :

<Label x:Name="Orientation"></Label>

* Ouvrir le contrôleur MainPage.cs et ajouter dans le constructeur un appel au module DeviceDisplay pour connaître l’orientation de l’écran et qui met à jour le champ Orientation :

switch (orientation)

{

case ScreenOrientation.Landscape:

this.Orientation.Text = "Paysage";

break;

case ScreenOrientation.Portrait:

this.Orientation.Text = "Portrait";

break;

case ScreenOrientation.Unknown:

this.Orientation.Text = "Orientation inconnue";

break;

}

* Souscrire à l’évènement ScreenMetricsChanged pour être tenu informé du changement d’orientation :

public MainPage()

{

InitializeComponent();

var metrics = DeviceDisplay.ScreenMetrics;

var orientation = metrics.Orientation;

switch (orientation)

{

case ScreenOrientation.Landscape:

this.Orientation.Text = "Paysage";

break;

case ScreenOrientation.Portrait:

this.Orientation.Text = "Portrait";

break;

case ScreenOrientation.Unknown:

this.Orientation.Text = "Orientation inconnue";

break;

}

DeviceDisplay.ScreenMetricsChanged += this.OnScreenMetricsChanged;

}

private void OnScreenMetricsChanged(object sender, ScreenMetricsChangedEventArgs e)

{

var metrics = e.Metrics;

var orientation = metrics.Orientation;

switch (orientation)

{

case ScreenOrientation.Landscape:

this.Orientation.Text = "Paysage";

break;

case ScreenOrientation.Portrait:

this.Orientation.Text = "Portrait";

break;

case ScreenOrientation.Unknown:

this.Orientation.Text = "Orientation inconnue";

break;

}

}