

Utilisation des dll pour faire fonctionner le lecteur Rfid sur le Cone²

1) Ajouter les dll dans le projet

Il faut ajouter les références suivantes dans le projet :

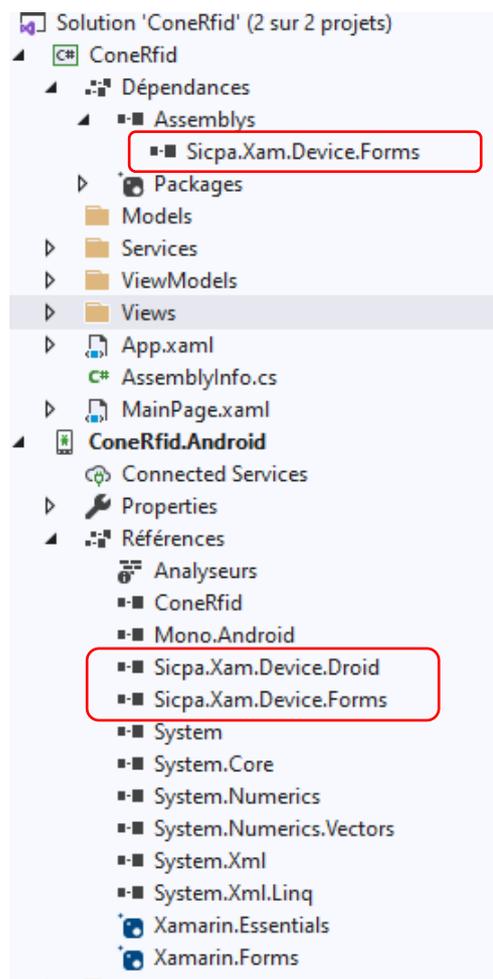
Dans la partie Forms

Sicpa.Xam.Device.Forms

Dans la partie Android

Sicpa.Xam.Device.Droid

Sicpa.Xam.Device.Forms



Note : Pour que les explications des méthodes apparaissent dans IntelliSense, il faut mettre les fichiers suivant avec les fichiers dll :

Sicpa.Xam.Device.Droid.xml

Sicpa.Xam.Device.Forms.xml

2) Utilisation

a) Dans la MainActivity de la partie Android

Il faut "tout simplement" déclarer la classe AgridentLecteur de la dll, dans l'instanciation du DependencyService.

```
//on enregistre une classe AgridentLecteur de type ILecteurRfid dans le dependencyService
DependencyService.RegisterSingleton<Sicpa.Xam.Device.Forms.Interfaces.ILecteurRfid>(new
Sicpa.Xam.Device.Droid.Rfid.Agrident.AgridentLecteur());
```

b) Dans le ViewModel de la partie Forms, qui contrôle le lecteur Rfid

- ❖ On déclare une interface ILecteurRfid qui contiendra l'objet lecteurRfid utilisé dans la couche Android

```
//cela permet d'avoir accès à toutes les propriétés et méthodes d'un objet lecteurRfid
private Sicpa.Xam.Device.Forms.Interfaces.ILecteurRfid lecteurRfid;
```

- ❖ Dans le constructeur :

- On récupère l'objet lecteurRfid de la couche Android par l'intermédiaire du DependencyService
- On se connecte au lecteurRfid
- On s'abonne à l'événement EvTagLu de lecteurRfid

```
//constructeur
1 référence
public RfidViewModel()
{
    //--RFID-- on récupère l'objet lecteurRfid utilisé dans la couche Android par l'intermédiaire du DependencyService
    lecteurRfid = DependencyService.Get<Sicpa.Xam.Device.Forms.Interfaces.ILecteurRfid>();

    //--RFID-- On se connecte au lecteur. En fait on active le receiver et son evenement
    lecteurRfid.Connexion();
    //--RFID-- on s'abonne à l'événement EvTagLu du lecteurRfid
    lecteurRfid.EvTagLu += OnEvTagLu;

    LireRfidCommand = new Command(LectureRfidCommand);
    NumRfid = "-----";
}
```

- ❖ On crée la méthode OnEvTagLu où on récupère les infos du tag dans l'objet infosTag

```
//méthode qui est déclenchée quand un tag est lu (événement EvTagLu)
1 référence
private void OnEvTagLu(object sender, Sicpa.Xam.Device.Forms.Models.InfosRfid infosTag)
{
    //on récupère l'info .IsoRfid (correspond au numéro à 15 chiffres) de l'objet infosTag
    NumRfid = infosTag.IsoRfid;
    var duration = TimeSpan.FromMilliseconds(100);
    Vibration.Vibrate(duration); //besoin d'une permission vibrate dans le manifest
}
```

- ❖ On lance la lecture d'un tag dans le code du bouton de la vue
On récupère une exception si l'appli AgridentWedge n'est pas installée

1 référence

```
private void LectureRfidCommand()
{
    NumRfid = "--((((--)))--";
    try
    {
        //--RFID-- lance la lecture d'un tag
        lecteurRfid.LectureTag();
    }
    catch (Sicpa.Xam.Device.Forms.Exceptions.AgridentExecption ex)
    {
        Console.WriteLine(ex.Message);
    }
}
```

L'interface ILecteurRfid

```
/// <summary>
/// Interface représentant un lecteur RFID
/// </summary>
1 référence
public interface ILecteurRfid
{
    /// <summary>
    /// lance une lecture de tag
    /// </summary>
    1 référence
    void LectureTag();
    /// <summary>
    /// Connexion à AgridentWedge
    /// </summary>
    1 référence
    void Connexion();
    /// <summary>
    /// Déconnexion à AgridentWedge
    /// </summary>
    1 référence
    void Deconnexion();
    /// <summary>
    /// Active l'antenne Rfid
    /// </summary>
    1 référence
    void RFOn();
    /// <summary>
    /// Desactive l'antenne Rfid
    /// </summary>
    1 référence
    void RFOff();
    /// <summary>
    /// Événement quand un tag est lu
    /// </summary>
    2 références
    EventHandler<InfosRfid> EvTagLu { get; set; }
}
```

La classe InfosRfid

```
public class InfosRfid
{
    /// <summary>
    /// Numéro rfid 15 chiffres
    /// </summary>
    2 références
    public string IsoRfid { get; set; }
    /// <summary>
    /// Code pays, ou fabricant, 3 chiffres
    /// </summary>
    2 références
    public string CodePays { get; set; }
    /// <summary>
    /// Numéro animal 12 chiffres
    /// </summary>
    2 références
    public string NumeroAnimal { get; set; }
    /// <summary>
    /// Type tag, HDX ou FDX
    /// </summary>
    2 références
    public string TypeTag { get; set; }
    /// <summary>
    /// Numéro Iso étendu
    /// </summary>
    2 références
    public string IsoEtendu { get; set; }
    /// <summary>
    /// Code de l'extention de l'Iso etendu
    /// </summary>
    2 références
    public string CodeEtendu { get; set; }
    /// <summary>
    /// Un tag à été lu
    /// </summary>
    2 références
    public bool TagLu { get; set; }
}
```