

DOCUMENTATION Application Musée

Classe métiers :

```
public class PeriodList
{
    public List<Period> periods { get; set; }
}

public class Period
{
    public string name { get; set; }
    public List<PeriodCollection> collection { get; set; }
}

public class PeriodCollection
{
    public string name { get; set; }
    public string file { get; set; }
}
```

```
public class Painter
{
    public string name { get; set; }
    public string years { get; set; }
    public string location { get; set; }
    public string shortinfoLink { get; set; }
    public string bioLink { get; set; }
    public string image { get; set; }
    public List<Painting> artworks { get; set; }
}

public class Painting
{
    public string name { get; set; }
    public string year { get; set; }
    public string picture { get; set; }
    public string thumbnail { get; set; }
}
```

Ce code est utilisé pour binder une listview à une collection, par la suite on récupérer un élément de la listview puis le cast en un objet de la liste initial :

```
[XamlCompilation(XamlCompilationOptions.Compile)]
public partial class CreateGroupsPage : ContentPage
{
    // Constructeur : liste de ressources "Period"
    public CreateGroupsPage (List<Period> listOfPeriods)
    {
        InitializeComponent ();

        // Binding de la ListView
        this.categories.ItemsSource = listOfPeriods;
    }

    // Choix d'un élément de la liste (Ex: "Italian Renaissance")
    private async void OnCategorySelected(object sender, SelectedItemChangedEventArgs e)
    {
        // Si rien n'a été sélectionné
        if (this.categories.SelectedItem == null)
            return;

        // Transtypage en référence de classe Period
        var selectedPeriod = e.SelectedItem as Period;

        // Élément désélectionné de la liste
        this.categories.SelectedItem = null;

        // Navigation : page "CreatePainters"
        var page = new CreatePaintersPage(selectedPeriod);
        await Navigation.PushAsync(page);
    }
}
```

```
public CreatePaintersPage(Period UnePeriode)
{
    InitializeComponent();

    this.painters.ItemsSource = GetLesPeintres(UnePeriode);
}
```

Ici on vient chercher une liste de peintre via une liste de fichier en xml :

```
private List<Painter> GetLesPeintres(Period UnePeriode)
{
    List<Painter> aRetourner = new List<Painter>();

    var assembly = typeof(CreateGroupsPage).GetTypeInfo().Assembly;
    Stream stream;

    foreach(PeriodCollection NomDePeintre in UnePeriode.collection)
    {
        stream = assembly.GetManifestResourceStream("ArtLookUp.Data." + NomDePeintre.file);

        using (var reader = new StreamReader(stream))
        {
            var jsonString = reader.ReadToEnd();

            Painter p = new Painter();
            p = JsonConvert.DeserializeObject<Painter>(jsonString);

            if (p == null)
                throw new Exception("Error! JSON file does not contain a list of periods!");

            aRetourner.Add(p);
        }
    }

    return aRetourner;
}
```

Ici un event est utilisé pour créer une nouvelle page :

```
public PainterArtworksPage(Painter UnPeintre)
{
    InitializeComponent();

    this.artworks.ItemsSource = UnPeintre.artworks;
}

private async void OnPaintingSelected(object o, SelectedItemChangedEventArgs e)
{
    await Navigation.PushAsync(new PaintingViewPage((Painting)this.artworks.SelectedItem));
}
```

```
public PainterDetailPage(Painter UnePainter)
{
    InitializeComponent();

    this.BindingContext = UnePainter;
}

private async void OpenArtworks(object o, SelectedItemChangedEventArgs e)
{
    await Navigation.PushAsync(new PainterArtworksPage((Painter)this.BindingContext));
}

private async void OpenBiography(object o, SelectedItemChangedEventArgs e)
{
    await Navigation.PushAsync(new PainterBiographyPage((Painter)this.BindingContext));
}
```