

PRESENTATION

Le musée des Célestins est un musée situé à Vichy. Le musée possède une des plus grandes collections d'œuvres impressionnistes et postimpressionnistes au monde. On peut y voir des chefs-d'œuvre de la peinture comme « Le Déjeuner sur l'herbe » et « l'Olympia » d'Édouard Manet, « la Danseuse âgée de quatorze ans » de Degas, etc... Plus de 5 000 peintures composent la collection.



- Pour chaque ŒUVRE, les informations relatives à l'ARTISTE sont mémorisées.

Exemple :



Le Déjeuner sur l'herbe est un tableau d'Édouard Manet datant de 1863.

« Édouard Manet (né à Paris le 23 janvier 1832 - mort à Paris le 30 avril 1883) est un peintre français majeur de la fin du XIXe siècle... »



- Les ŒUVRES sont exposées dans des SALLES, correspondant à des ECOLES de peinture. Chaque salle est thématique puisque elle héberge un ensemble d'œuvres ayant un ou plusieurs points communs. La composition du musée en salles est dynamique dans la mesure où elles vont évoluer au gré des expositions organisées par la direction du musée. Par exemple, au jour d'aujourd'hui, le musée est organisé autour des salles suivantes : ECOLE FRANÇAISE, ECOLE NÉERLANDAISE, Exposition MANET, Exposition DEGAS, etc...

Exemple : Salle « Exposition MANET »



Le BESOIN

La Direction du Musée des Célestins souhaite pouvoir offrir au public, une application Android qui permette de consulter l'ensemble de ses œuvres, classées par école, période ou suivant un autre critère. Par exemple, pour l'école française impressionniste, on pourra lister l'ensemble des peintres : Manet, Bastien-Lepage, Bazille, Renoir, Etc... Puis, après avoir choisi un peintre, par exemple Manet, obtenir l'ensemble de ses œuvres (photographie, puis descriptif) exposées dans le musée, ainsi que sa biographie :



Édouard Manet, né le 23 janvier 1832 à Paris (Acte de naissance détruit pendant la Commune de Paris de 1871) et mort le 30 avril 1883 dans la même ville, est un peintre et graveur français majeur de la fin du XIXe siècle. Précurseur de la peinture moderne qu'il affranchit de l'académisme, Édouard Manet est à tort considéré comme l'un des pères de l'impressionnisme : il s'en distingue en effet par une facture soignée du réel qui n'utilise pas (ou peu) les nouvelles techniques de la couleur et le traitement particulier de la lumière [...]

Olympia, 1863.



L'œuvre représente au premier plan une jeune femme blanche nue, le pied gauche encore chaussé d'une mule, allongée sur un divan et un châle de cachemire blanc, dans un intérieur décoré de tentures vertes et de tapisseries. Posée sur deux oreillers satinés, elle est accoudée sur son bras droit, la main gauche sur la naissance de ses jambes, le regard porté vers le spectateur. Au second plan, derrière le lit, à droite, une femme noire lui présente un bouquet de fleurs devant un fond vert ; un chat noir se dresse sur l'extrémité droite du lit, la queue levée [...]

L'APPLICATION

La Direction du Musée s'est adressée à l'ESN SIVY pour le développement de la future application. L'environnement de développement choisi est XAMARIN.FORMS. L'application se composera de deux parties :

- Partie SERVEUR : application de service web REST, où les données (écoles, périodes, peintres et œuvres) seront accessibles en format JSON (ressources sérialisées).
- Partie CLIENTE : application mobile Android exploitant les données JSON récupérées depuis le serveur REST.

a) L'interface à développer est la suivante :



b) Jeu de test

Dans un premier temps il s'agira de développer la partie CLIENTE (=sujet du projet actuel). Pour cela, SIVY a obtenu un JEU DE TEST RELEVANT DU DOMAINE PUBLIC, et bien que leur contenu ne corresponde pas aux œuvres exposées dans le Musée des Célestins (écoles impressionnistes), elles permettront le développement de la partie cliente, en attendant le développement de la partie serveur.

Ces données se présentent sous trois formats : JSON, HTML et images JPG

a) Données JSON : écoles, périodes, peintres et œuvres.

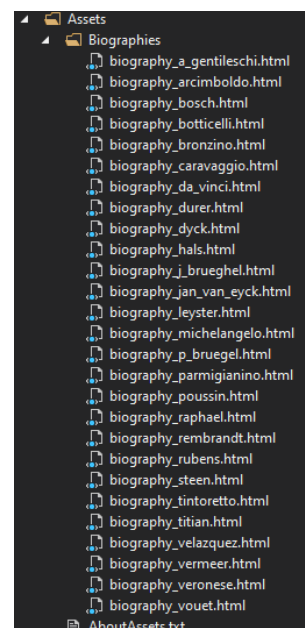
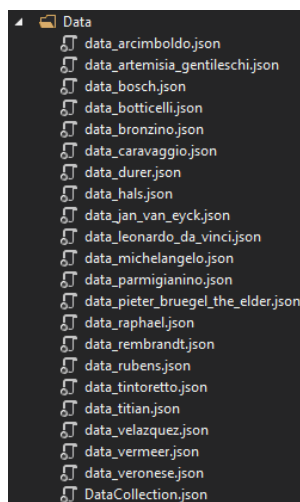
Par exemple :



REMARQUE : ces fichiers JSON seront ajoutés à la solution, dans un dossier « Data » à créer dans la « partie » C#/XAML

b) Données HTML : biographies de chaque peintre

```
// biography_botticelli.html
<!DOCTYPE html>
<style>
  body {
    background-color: #764040;
  }
  div.container p {
    text-align: justify;
    font-family: Bookman;
    font-size: 14px;
    font-style: normal;
    font-weight: normal;
    text-decoration: none;
    text-transform: none;
    color: #000000;
  }
</style>
<div class="container">
```



...

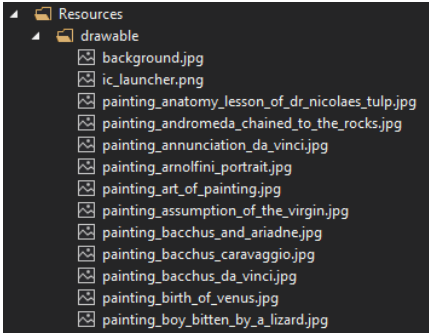
```
<p><strong>Early years</strong></p>
<p style="text-align: justify;">
  Details about Botticelli's personal life are sparse, but his prolific professional career is well
  documented due to the stunning heights of his fame.</p>
<p style="text-align: justify;">Botticelli's career started at age 14, where he was apprenticed to Fra
  Filippo Lippi who had a noticeable effect on his painting style and many historians are keen to note
  that he adopted Lippi's intimate and detailed artistic technique [...]
  &nbsp;
</p>

<!-- Etc... -->
</div>
```

REMARQUE : ces fichiers HTML seront ajoutés dans un dossier « Biographies » à créer dans la « partie » ANDROID :

c) Images JPG :

Les images concernant les peintres, leurs œuvres, etc... Seront ajoutées dans le dossier « Resources/drawable » de la partie Android de la solution :



ELEMENTS TECHNIQUES

a) « WelcomePage » - Page d'accueil

- Chargement et lecture des données JSON depuis le fichier « DataCollection.json »

```
// Attribut. Objet ressource "PeriodList" : ensemble des données JSON désérialisées.
private PeriodList dataPeriodList = new PeriodList();

// Constructeur
1 référence
public WelcomePage()
{
    InitializeComponent();

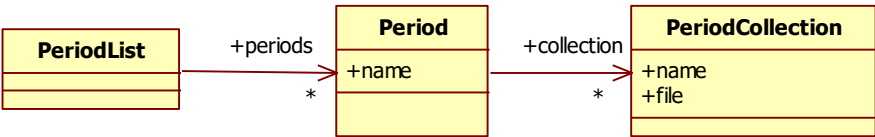
    // Chargement des données JSON ("embedded resources")
    var assembly = typeof(WelcomePage).GetTypeInfo().Assembly;
    Stream stream = assembly.GetManifestResourceStream("NomApp.Data.DataCollection.json");

    // Lecture données JSON et chargement de la collection --
    using (var reader = new StreamReader(stream))
    {
        var jsonString = reader.ReadToEnd();

        // Etc...
    }
}
```

```
"periods": [
  //Italian Renaissance
  {
    "name": "Italian Renaissance",
    "collection": [
      {
        "name": "Sandro Botticelli",
        "file": "data_botticelli.json"
      },
      {
        "name": "Leonardo da Vinci",
        "file": "data_leonardo_da_vinci.json"
      },
      { ... }
    ]
  },
  //Northern Renaissance
  { ... }
]
```

- Desérialisation en objet RESSOURCE « PeriodList ».



- Affichage de la page suivante (« liste des périodes ») en passant à son constructeur, la collection des périodes : `this.dataPeriodList.periods`

b) « CreateGroupsPage » - Affichage de la liste des PERIODES

- La collection de ressources « Period » doit être liée (« bindée ») à une « ListView » XAML (afficher uniquement le nom de la période).
- Sélection d'une PERIODE et affichage dans la page suivante, de la LISTE DES PEINTRES correspondants. On transmet au constructeur la ressource « Period » sélectionnée dans la liste.

c) « CreatePaintersPage » - Affichage de la liste des PEINTRES pour la période précédemment sélectionnée.

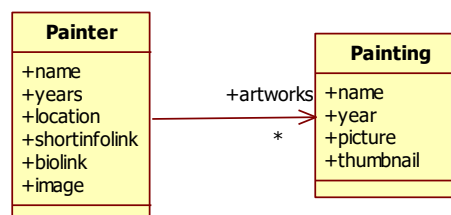
- Une collection `List<Painter>` `listOfPainters` doit être initialisée à partir des données JSON : peintres pour la période traitée. Pour chaque PEINTRE :

```
// On récupère le fichier JSON du Peintre.
var assembly = typeof(WelcomePage).GetTypeInfo().Assembly;
Stream stream = assembly.GetManifestResourceStream("NomApp.Data."+ c.file);

// Parcours des données du fichier JSON (Ex: "data_boticelli.json")
Painter dataPainter;
using (var reader = new System.IO.StreamReader(stream))
{
    var jsonString = reader.ReadToEnd();

    // Désérialisation
    // ...
}

// Ajout du peintre désérialisé à la collection à "binder"
// ...
```



- Sélection d'un PEINTRE et affichage dans la page suivante, de sa FICHE (= portrait, dates, lieu de naissance et court descriptif). Cette page présente deux boutons permettant de consulter sa BIOGRAPHIE et sa liste d'ŒUVRES. On transmet au constructeur la ressource « Painter » sélectionnée dans la liste.

d) « PainterDetailPage » - Fiche du PEINTRE précédemment sélectionné.

- Il suffit de « binder » la ressource au contexte de la page.
- Affichage, suivant le bouton choisi, de la biographie ou de la liste d'œuvres. Dans les deux cas, on transmettra au constructeur correspondant, la ressource « Painter » traitée.

e) « PainterBiographyPage » - Biographie détaillée du PEINTRE précédemment sélectionné.

- Les données seront récupérées depuis les fichiers HTML. Pour cela, on utilisera le composant XAML « WebView » :
`<WebView x:Name="webviewBio" VerticalOptions="FillAndExpand" HorizontalOptions="FillAndExpand" />`
- La source de données de ce composant est constitué de la concatenation de : file:///android_asset/Biographies/ Avec le nom du fichier HTML correspondant au peintre traité.

f) « PainterArtWorksPage » - Liste des OEUVRES du PEINTRE précédemment sélectionné.

- Il s'agira de lier (« binder ») la collection de ressources « Painting » à une « ListView ».
- L'utilisateur a la possibilité de cliquer sur une des œuvres afin d'afficher son détail, dans la page suivante. On transmettra à son constructeur la ressource « Painting » sélectionnée.

« PainterArtWorksPage » - Liste des OEUVRES du PEINTRE précédemment sélectionné.

- L'image correspondant à l'œuvre précédemment sélectionnée sera affichée dans un composant XAML :

```
<Grid Padding="20">
    <local:PinchToZoom>
        <local:PinchToZoom.Content>
            <Image Source="{Binding picture}" />
        </local:PinchToZoom.Content>
    </local:PinchToZoom>
</Grid>
```

- REMARQUE : « PinchToZoom » est une classe permettant de gérer un « zoom » sur l'image de l'œuvre. Le code correspondant, récupéré de sources externes, est fourni :

```

0 références
public static class DoubleExtensions
{
    2 références
    public static double Clamp(this double self, double min, double max) {...}
}

1 référence
public class PinchToZoom : ContentView
{
    private double currentScale = 1;
    private double startScale = 1;
    private double xOffset = 0;
    private double yOffset = 0;

    0 références
    public PinchToZoom() {...}

    1 référence
    private void PinchGesture_PinchUpdated(object sender, PinchGestureUpdatedEventArgs e) {...}

    1 référence
    public void OnPanUpdated(object sender, PanUpdatedEventArgs e) {...}
}

```

- On doit pouvoir activer le mode « paysage » afin de mieux consulter le détail de l'œuvre. Pour cela, on ajoutera le code C# suivant, sur les deux événements « OnAppearing » et « OnDisappearing »

```

protected override void OnAppearing()
{
    base.OnAppearing();
    MessagingCenter.Send(this, "unlockLandscape");
}

0 références
protected override void OnDisappearing()
{
    base.OnDisappearing();
    MessagingCenter.Send(this, "lockLandscape");
}

```

ANNEXE

Classes représentant les ressources

```

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; }
}

```