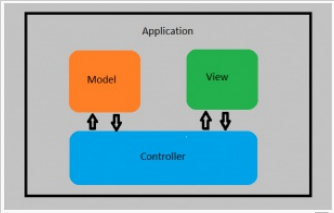
Flickr app using Model-View-Controller in Qt and Windows Phone

This article demonstrates how to create **Model-View-Controller** architecture in Qt and Windows Phone. The context in which it is described is a Flikr app.

Introduction

Model-View-Controller is a well known software design pattern. This pattern is used in both Qt and Windows Phone - in Qt we use **Model-View-Delegate** and in Windows Phone 7 we have **Model-View-ViewModel** pattern. One way of considering the parts of the pattern is:

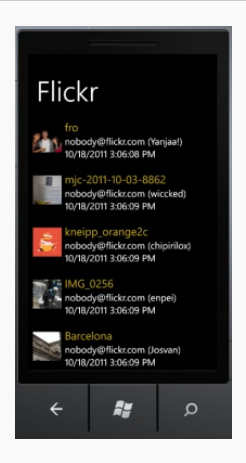
- Model : Store the data
- View : Display the data
- Controller : Communicate between Model and View to process the data.



General Representation of MVC design pattern

In this code example we implement a Flikr application for both platforms. While creating the application we will follow the MVC model and will compare the similarity and difference between the platforms in the handling the data. This project is designed to help beginners understand how the MVC pattern is used in Qt and WP7.

Flickr in Qt



Flickr in WP7

Implementation

We create empty projects for both Qt and WP7. We first add code in Qt project and will put the similar code in WP7.

Model

The Model is collection of class that can store data, separate data. Occasionally it can also communicate with any user interface. Here we will get the Flickr data and store it for further use. We are storing four information *Title*,*Image URL*, *Author* and *Date* about each Flickr item.

Qt Project (Model.qml)

Lets add a QML file in the Qt project, **Right Click** project explorer->**Add New**-> **QML** (from left panel)-> **QML File** (from right panel) , name the file name as **Model.qml** and add the below code. This is our Model class in Qt project. This class will store each model item attributes.

```
XmlListModel {
  id: feedModel
  source: "http://api.flickr.com/services/feeds/photos_public.gne?format=rss2" // flickr
  query: "/rss/channel/item" // flickr
  namespaceDeclarations: "declare namespace media=\"http://search.yahoo.com/mrss/\";"
  // Flickr
  XmlRole { name: "title"; query: "title/string()" }
  XmlRole { name: "imagePath"; query: "media:thumbnail/@url/string()" }
  XmlRole { name: "photoAuthor"; query: "author/string()" }
  XmlRole { name: "photoDate"; query: "pubDate/string()" }
}
```

WP7 Project (Model.cs)

Lets add a CS file in WP7 project, **Right Click** project explorer->**Add** -> **New Item**-> Select **Class** (from middle panel), name the

(C) Copyright Nokia 2014. All rights reserved.

file name as **Model.cs** and add the below code. This is our Model class in WP7 project. This class will hold the parsed data.

```
public class Model
  {
  public string UserName { get; set; }
  public string ImageSource { get; set; }
  public DateTime PubDate { get; set; }
  public string Author { get; set; }
  }
```

View

The view is represented by a screen where we display the data. In our case we will display the Flickr thumbnail, title, author and the date in list view.

Qt Project (View.qml)

Lets add a QML file in the Qt project, **Right Click** project explorer->**Add New**-> **QML** (from left panel)-> **QML File** (from right panel) , name the file name as **View.qml** and add the below code. This is our View class in Qt project. This class will display the Flickr data into the screen.

```
Component {
  Item {
  id: wrapper; width: wrapper.ListView.view.width; height: 86
  Item {
  id: moveMe
  Rectangle { color: "black"; opacity: index % 2 ? 0.2 : 0.4; height: 84; width:
wrapper.width; y: 1 }
  Rectangle {
  x: 6; y: 4; width: 49; height:49; color: "white"; smooth: true
  Image { source: imagePath; x: 0; y: 0
  height:49
  width:49 }
  }
  Column {
  x: 100; width: wrapper.ListView.view.width - 95; y: 15; spacing: 2
  Text { text: title; color: "white"; width: parent.width; font.pixelSize: 14;
font.bold: true; elide: Text.ElideRight; style: Text.Raised; styleColor: "black" }
  Text { text: photoDate; width: parent.width; font.pixelSize: 14; elide:
Text.ElideLeft; color: "#cccccc"; style: Text.Raised; styleColor: "black" }
  Text { text: photoAuthor; width: parent.width; font.pixelSize: 14; elide:
Text.ElideRight; color: "#cccccc"; style: Text.Raised; styleColor: "black" }
  }
  }
  }
}
```

WP7 Project (MainPage.xaml)

In WP7 project XAML file represents the View of the application. So we will consider the default **MainPage.xaml** file to be the our View in this case. To display the Flick data onto the screen lets add the below code inside <*Grid…*> element in **MainPage.xaml** file

```
<!--ContentPanel - place additional content here-->
  <Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
  <ListBox Height="650" HorizontalAlignment="Left" Name="listBox1"
VerticalAlignment="Top" Width="456" >
  <ListBox.ItemTemplate>
  <DataTemplate>
  <StackPanel Orientation="Horizontal" Height="132" >
  <Image Source="{Binding ImageSource}" Height="73" Width="73" VerticalAlignment="Top"
Margin="0,10,8,0"/>
  <StackPanel Width="370">
  <TextBlock Text="{Binding UserName}" Foreground="#FFC8AB14" FontSize="28" />
  <TextBlock Text="{Binding Author}" TextWrapping="Wrap" FontSize="24" />
  <TextBlock Text="{Binding PubDate}" TextWrapping="Wrap" FontSize="24" />

  </StackPanel>
  </StackPanel>
  </DataTemplate>
  </ListBox.ItemTemplate>
  </ListBox>

  </Grid>
```

Controller

(C) Copyright Nokia 2014. All rights reserved.

The Controller communicates with the front end of the Model and sends the data to View for loading.

Qt Project (Controller.qml)

Tipically **Controller.qml** is the **main.qml** in Qt project. I have just renamed the file and add some code in it. In Qt project first we have created one rectangle of specific height and width after that we have added the `Model` and `ListView` in it. This class first loads the data into Model and then display it in List View. This is the class which decides which Model to load and how the data should be displayed.

```
Rectangle {
  id: mainwindow
  width: 360//parent.width
  height:640// parent.height
  color: "green"
  Text {
  id: idText
  text: qsTr("Flickr")
  color: "black"
  font.pixelSize: 31
  font.bold: true;
  style: Text.Normal;
  styleColor: "white"
  x:140
  }
  Model {id: feedModel}
  ListView {
  id: list
  anchors.left : parent.left
  anchors.right: parent.right
  anchors.top : idText.bottom
  width: mainwindow.width ; height: mainwindow.height
  model: feedModel
  delegate: View {}
  }
}
```

WP7 Project (MainPage.xaml.cs)

For WP7 **MainPage.xaml.cs** is the controller class. Where it loads the data into Model class and then display the data in List View.

```
// Constructor
public MainPage()
  {
  InitializeComponent();

  WebClient webclient = new WebClient();
  webclient.DownloadStringCompleted += new
DownloadStringCompletedEventHandler(webclient_DownloadStringCompleted);
  webclient.DownloadStringAsync(new
Uri("http://api.flickr.com/services/feeds/photos_public.gne?format=rss2")); // Flickr
search

  }
  void webclient_DownloadStringCompleted(object sender, DownloadStringCompletedEventArgs
e)
    {
    if (e.Error != null) {
        MessageBox.Show("error");
    }


  // parsing Flickr
  XElement Xml = XElement.Parse(e.Result);

  XNamespace ns = "http://search.yahoo.com/mrss/"; // flilckr


  listBox1.ItemsSource = from flickr in Xml.Descendants("item")
  select new Model
  {
  ImageSource = flickr.Element(ns + "thumbnail").Attribute("url").Value,
  UserName = flickr.Element("title").Value,
  Author = flickr.Element("author").Value,
  PubDate = DateTime.Parse(flickr.Element("pubDate").Value)
  };

  }
```


(C) Copyright Nokia 2014. All rights reserved.

Source Code

- The full source code of Qt example is available here: File:MVCQt.zip
- The full source code of WP7 example is available here: File:MVCWP7.zip

(C) Copyright Nokia 2014. All rights reserved.