UI Cross thread updates

This article will discuss how to update a UI object for Xamarin in another Thread (aside from the calling one).

For example you want to update your Label asynchronously in another Thread or class that is running in parallel to your main thread.

Example code :

The class / Thread running in parallel

public class Stock

{

…

private void Wclient\_DownloadDataCompleted(object sender, DownloadDataCompletedEventArgs e)

{

if (wclient != null)

{

//Console.WriteLine(Encoding.UTF8.GetString(e.Result));

// Console.WriteLine(this.symbol + ":" + o["quote"]["latestPrice"]);

try

{

JObject o = JObject.Parse(Encoding.UTF8.GetString(e.Result));

this.current\_Value = float.Parse(o["quote"]["latestPrice"].ToString());

Console.WriteLine(this.current\_Value);

Device.BeginInvokeOnMainThread(()=>

{

try

{

this.mylabel.Text = o["quote"]["latestPrice"].ToString();

}

catch (Exception ex)

{

Console.WriteLine(ex.ToString());

}

}

);

}

catch (Exception ex)

{

Console.WriteLine("ERROR : "+ ex.ToString());

this.current\_Value = 0;

}

}

//Console.WriteLine("Done!");

}

}

}

If we look carefully at this fragment of code:

Device.BeginInvokeOnMainThread(()=>

{

try

{

this.mylabel.Text = o["quote"]["latestPrice"].ToString();

}

catch (Exception ex)

{

Console.WriteLine(ex.ToString());

}

}

);

This basically allows us to run our update on IOS

For android we will need to use something called :

runOnUiThread

NOTE: our label object is a pass by reference variable.