Day 29: Plugins for Xamarin.Android

Any Xamarin series will not be complete without talking about the awesome Plugin ecosystem that Xamarin has. Xamarin’s [Github Repository](github.com/xamarin/plugins) has the whole list of Plugins for Xamarin that are developed by folks at Xamarin and also community Xamarin developers. A Plugin for Xamarin is just DLLs which are often distributed using Nuget or Xamarin Component store. The plugins abstract hard to do functionality over a simple and common API. Today, we are looking at Plugins for Xamarin.Android but a majority Xamarin plugins are, and become powerful, in cross platform scenarios, where our abstracted common API would be even more helpful.

Following are the Plugins that we will explore today. This list, by no means is “most popular” or “most useful” –

[Connectivity Plugin by James Montemagno](https://www.nuget.org/packages/Xam.Plugin.Connectivity/)

[Local Notifications by Ed Snider](https://github.com/edsnider/Xamarin.Plugins)

[Settings Plugin by James Montemagno](https://www.nuget.org/packages/rda.SocketsForPCL)

# Connectivity Plugin

Connectivity Plugin is used to detect whether you are connected to internet or not, and so you can handle the “No Network” scenarios betters. To install, the Connectivity Plugin, install the following Nuget package –

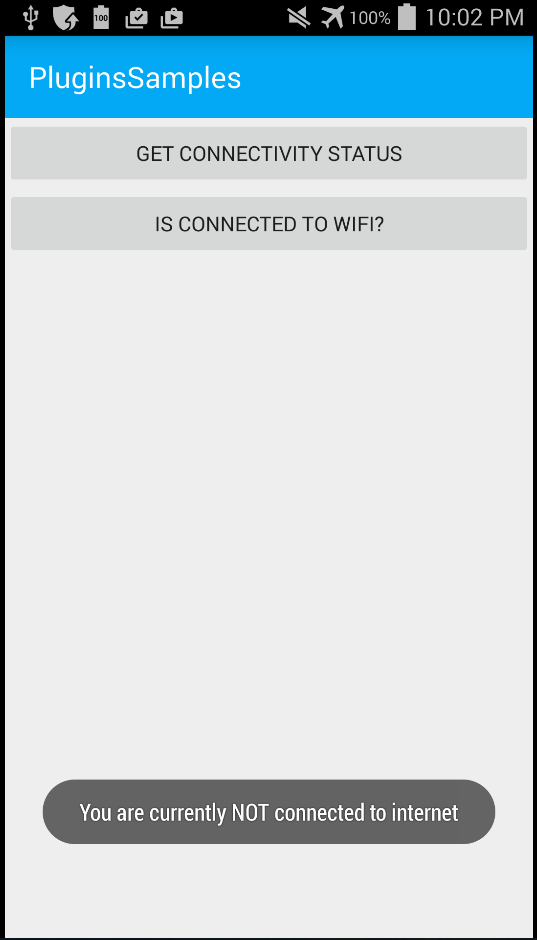
|  |
| --- |
| PM> Install-Package Xam.Plugin.Connectivity |

Once you install this plugin, you can use its common “CrossConnectivity” API to learn more about connectivity status of the device. To use Connectivity Plugin, you must enable “ACCESS\_NETWORK\_STATE” permission and “ACCESS\_WIFI\_STATE” permission in your Android Manifest file.

Once you did, let’s look at the API for the using the Plugin, here is the snippet for knowing whether you are connected to internet or not –

|  |
| --- |
| private void GetConnectivityStatusButton\_Click(object sender, EventArgs e)  {  var isConnected = CrossConnectivity.Current.IsConnected;  if (isConnected)  {  Toast.MakeText(this, "You are currently connected to internet", ToastLength.Long).Show();  }  else  {  Toast.MakeText(this, "You are currently NOT connected to internet", ToastLength.Long).Show();  }  } |

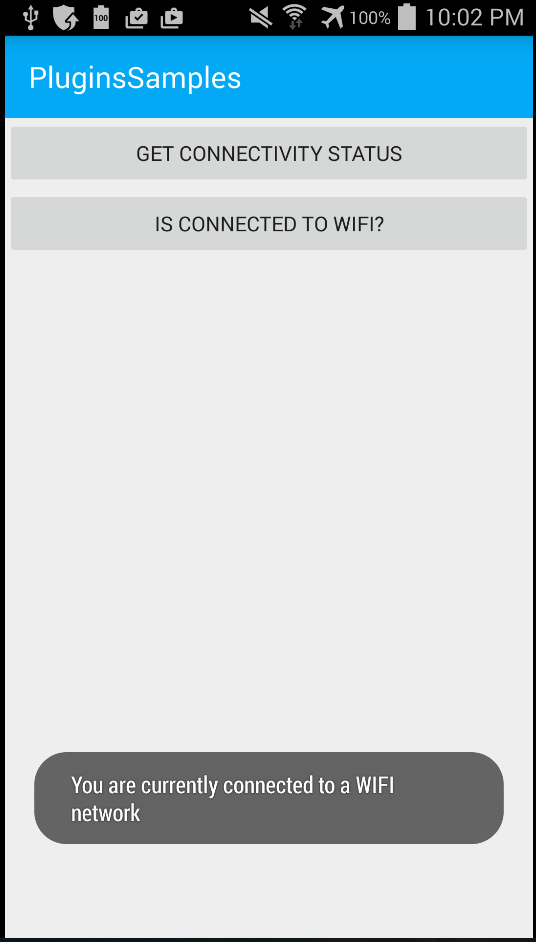
In the sample above, the CrossConnectivity.Current.IsConnected API tells us whether or not the device is connected to internet.



Now, let’s learn how we can find out whether or not we are connected to a WIFI Network –

|  |
| --- |
| private void IsConnectedToWifiButton\_Click(object sender, EventArgs e)  {  var connectionTypes = CrossConnectivity.Current.ConnectionTypes;  bool isWifi = connectionTypes.Any(connectionType => connectionType == ConnectionType.WiFi);  if (isWifi)  {  Toast.MakeText(this, "You are currently connected to a WIFI network", ToastLength.Long).Show();  }  else  {  Toast.MakeText(this, "You are currently NOT connected to a WIFI network", ToastLength.Long).Show();  }  } |

In the snippet above, we are calling the CrossConnectivity.Current.ConnectionTypes API which gives us list of connections that the device currently has. From that list, we are finding out if any of the connection type is of type WIFI using ConnectionType enum.



# Local Notifications Plugin

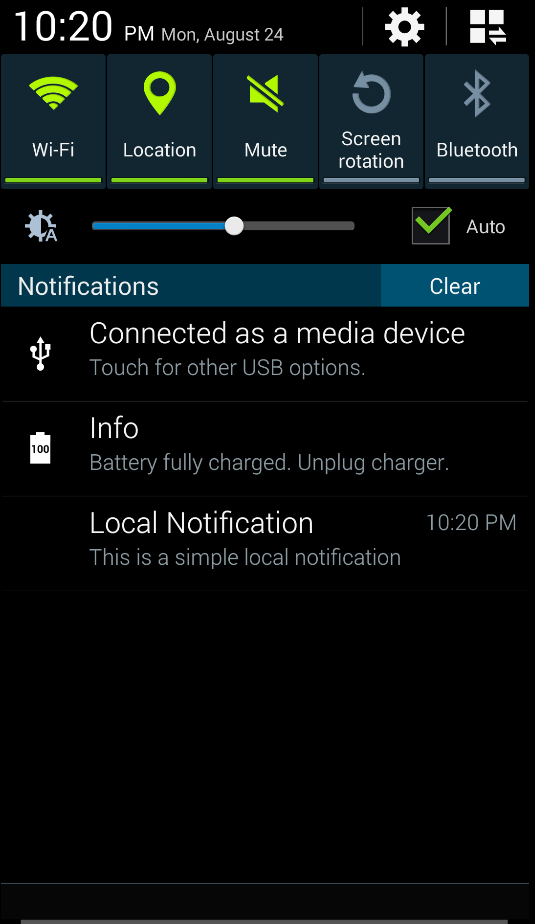
The local notifications plugin simplifies the API for the local notifications that we looked at couple of days ago. To install this plugin via nuget, run the following command in your package manager console –

|  |
| --- |
| PM> Install-Package Xam.Plugins.Notifier |

Once installed, to show a local notification, simple use the below API –

|  |
| --- |
| private void ShowNotificationButton\_Click(object sender, System.EventArgs e)  {  Notifier.Current.Show("Local Notification", "This is a simple local notification");  } |

And the result would look like –



# Settings Plugin

Settings Plugin is another awesome plugin created by James Montemagno. Settings are my go to way to store application level “settings” when writing Xamarin Applications. To install Settings plugin, simply run the following command in your package manager console –

|  |
| --- |
| PM> Install-Package Xam.Plugins.Settings |

Once installed, You need to create a “Settings” file in your Xamarin.Android Application –

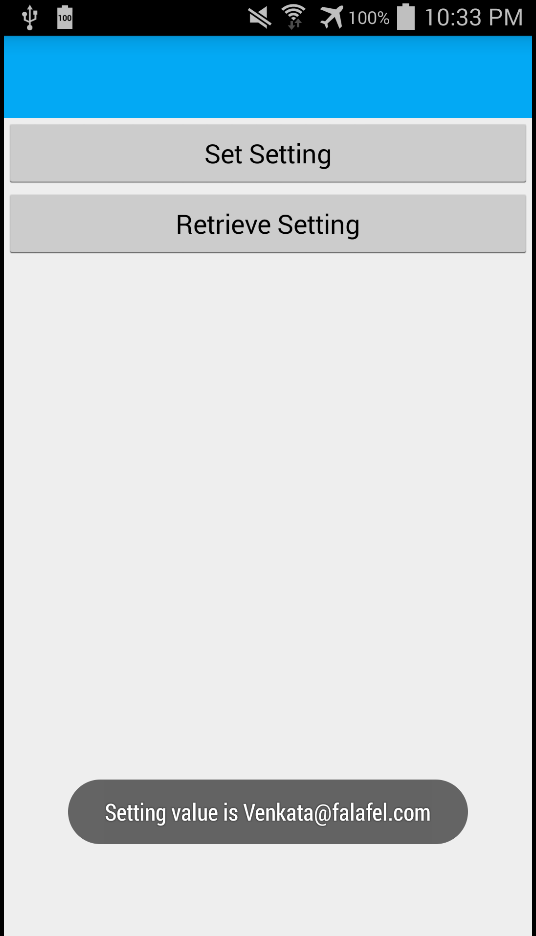
|  |
| --- |
| // Helpers/Settings.cs  using Refractored.Xam.Settings;  using Refractored.Xam.Settings.Abstractions;  namespace PluginsSamples  {  /// <summary>  /// This is the Settings static class that can be used in your Core solution or in any  /// of your client applications. All settings are laid out the same exact way with getters  /// and setters.  /// </summary>  public static class ApplicationSettings  {  private static ISettings AppSettings  {  get  {  return CrossSettings.Current;  }  }  #region Setting Constants  private static readonly string SettingsDefault = string.Empty;  #endregion  public static string LoginNameSetting  {  get  {  return AppSettings.GetValueOrDefault<string>("LoginName", SettingsDefault);  }  set  {  AppSettings.AddOrUpdateValue<string>("LoginName", value);  }  }  }  } |

This class would have our Application Level settings. For example, I am storing “LoginNameSetting” above and we are calling the Plugin’s AppSettings.GetValueOrDefault<> and AppSettings.AddOrUpdateValue<> methods to get and set the value. And finally, to use this API we would something like –

|  |
| --- |
| private void SetSettingButton\_Click(object sender, EventArgs e)  {  ApplicationSettings.LoginNameSetting = "Venkata@falafel.com";  } |

And to retrieve the setting –

|  |
| --- |
| private void RetrieveSettingButton\_Click(object sender, EventArgs e)  {  Toast.MakeText(this, "Setting value is " + ApplicationSettings.LoginNameSetting, ToastLength.Long).Show();  } |



These settings are persisted across app closes and device restarts.

James Montemagno also put together a fantastic [video](https://plus.google.com/events/c4uvjot4v4urdju6vl4sal773jg) on how to create your own Plugins for the Xamarin Ecosystem, give it a watch if not for how to create, you would definitely understand the power of plugins a lot more.

That’s it for today, see you all tomorrow.