



@JUUCUSTODIO
JULIANO CUSTÓDIO

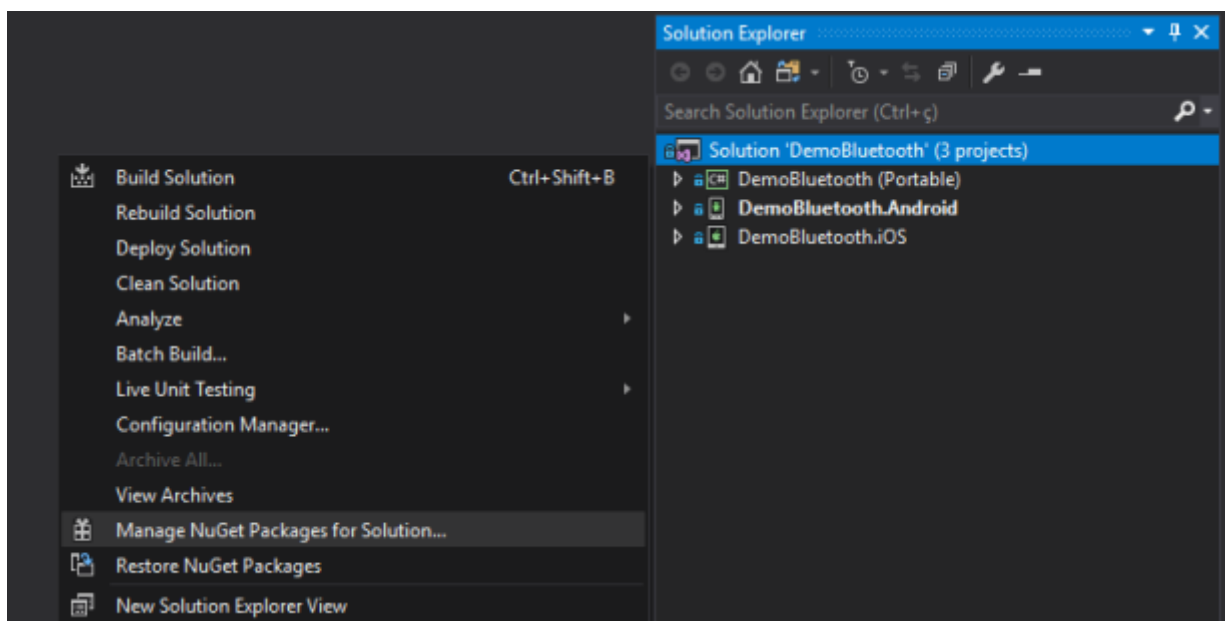
Welcome to Xamarin!

Bluetooth – Xamarin.Forms

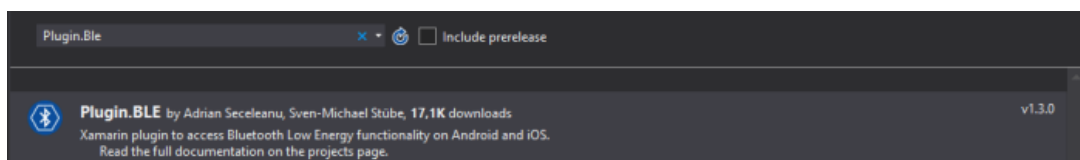
Olá, neste post irei demonstrar como sua aplicação Xamarin.Forms, pode encontrar dispositivos e conectar-se via bluetooth.

ADICIONANDO O NUGET PACKAGE

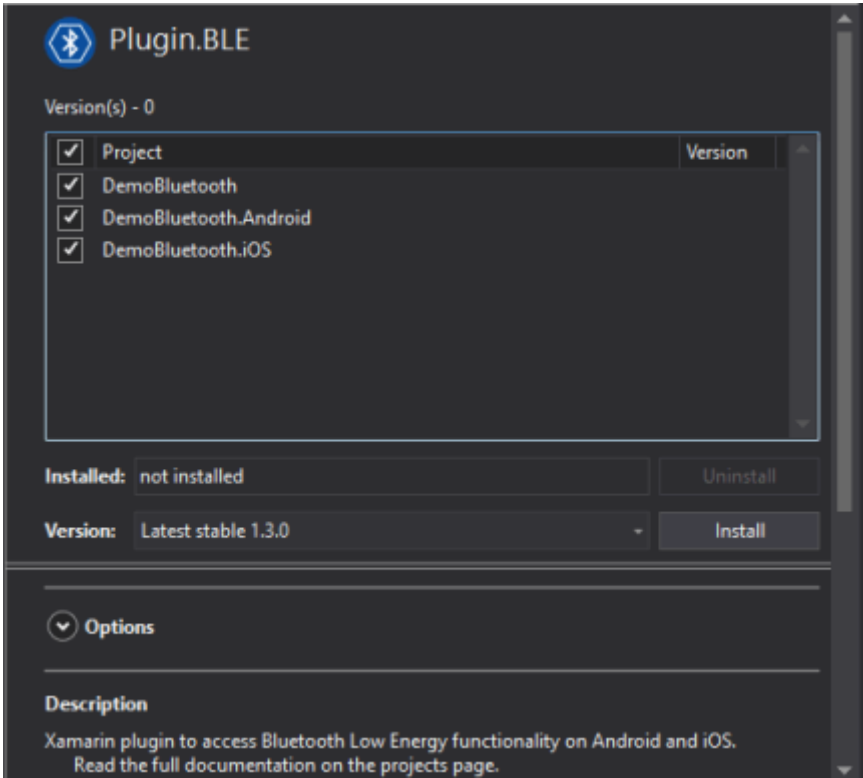
Clique com o botão direito em cima de sua Solution e selecione “Manage NuGet Packages for Solution...”.



Digite “Plugin.BLE” e selecione o plugin como demonstrado na imagem a seguir.



Selecione todos os projetos e clique no botão “Install”.



Xaml

Crie um botão para acionar o Scanner e uma lista onde será exibido todos os dispositivos encontrados. Adicione 2 labels, uma para aparecer o Nome do dispositivo e outra para o Address, como demonstrado a seguir:

1	<?xml version="1.0" encoding="utf-8" ?>
2	<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
3	xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
4	x:Class="DemoBluetooth.MainPage">
5	<ContentPage.Content>
6	<StackLayout>
7	<Button Text="Search" Clicked="searchDevice"/>
8	<ListView x:Name="DevicesList"
9	IsPullToRefreshEnabled="true"
10	CachingStrategy="RecycleElement"

11	ItemSelected="DevicesList_OnItemSelected">
12	<ListView.ItemTemplate>
13	<DataTemplate>
14	<ViewCell>
15	<StackLayout>
16	<Label Text="{Binding NativeDevice.Name}"></Label>
17	<Label Text="{Binding NativeDevice.Address}" ></Label>
18	</StackLayout>
19	</ViewCell>
20	</DataTemplate>
21	</ListView.ItemTemplate>
22	</ListView>
23	</StackLayout>
24	</ContentPage.Content>
25	</ContentPage>

view raw **MainPage.xaml** hosted with ❤ by **GitHub**

C#

Crie as variáveis adapter, bluetoothBLE, list e device e inicialize no método construtor como demonstrado a seguir.

1	namespace DemoBluetooth
2	{
3	public partial class MainPage
4	{
5	
6	IAdapter adapter;
7	IBluetoothLE bluetoothBLE;
8	ObservableCollection<IDevice> list;
9	IDevice device;
10	
11	public MainPage()
12	{
13	InitializeComponent();

14	
15	bluetoothBLE = CrossBluetoothLE.Current;
16	adapter = CrossBluetoothLE.Current.Adapter;
17	
18	list = new ObservableCollection<IDevice>();
19	DevicesList.ItemsSource = list;
20	
21	}
22	}
23	}

[view raw MainPageInitialize.cs](#) hosted with ❤ by [GitHub](#)

Em seguida crie o método “searchDevice”, que irá verificar se o Bluetooth do dispositivo está ativado, se sim irá scanear os dispositivos, a cada dispositivo encontrado irá adicionar na variável list criada anteriormente.

Vale ressaltar que o ScanMode está configurado para “Balanced”, isso significa que este modo possui um desgaste da bateria balanceado.

1	...
2	
3	private async void searchDevice(object sender, EventArgs e)
4	{
5	if (bluetoothBLE.State == BluetoothState.Off)
6	{
7	await DisplayAlert("Atenção", "Bluetooth desabilitado.", "OK");
8	}
9	else
10	{
11	list.Clear();
12	
13	adapter.ScanTimeout = 10000;
14	adapter.ScanMode = ScanMode.Balanced;
15	
16	
17	adapter.DeviceDiscovered += (obj, a) =>
18	{
19	if (!list.Contains(a.Device))
20	list.Add(a.Device);
21	};
22	
23	await adapter.StartScanningForDevicesAsync();
24	
25	}
26	
27	}

28	
29	...

[view raw MainPageSearchDevice.cs](#) hosted with ❤ by [GitHub](#)

E por último, crie o método “DevicesList_OnItemSelected” que irá se conectar com o dispositivo que foi selecionado na lista.

1	...
2	
3	private async void DevicesList_OnItemSelected(object sender, SelectedItemChangedEvent
4	{
5	device = DevicesList.SelectedItem as IDevice;
6	
7	var result = await DisplayAlert("AVISO", "Deseja se conectar a esse dispositivo?", "Conec
8	
9	if (!result)
10	return;
11	
12	//Stop Scanner
13	await adapter.StopScanningForDevicesAsync();
14	
15	try
16	{
17	await adapter.ConnectToDeviceAsync(device);
18	
19	await DisplayAlert("Conectado", "Status:" + device.State , "OK");
20	
21	}
22	catch (DeviceConnectionException ex)
23	{
24	await DisplayAlert("Erro", ex.Message, "OK");
25	}
26	
27	}
28	
29	...

[view raw MainPageDevicesList_OnItemSelected.cs](#) hosted with ❤ by [GitHub](#)

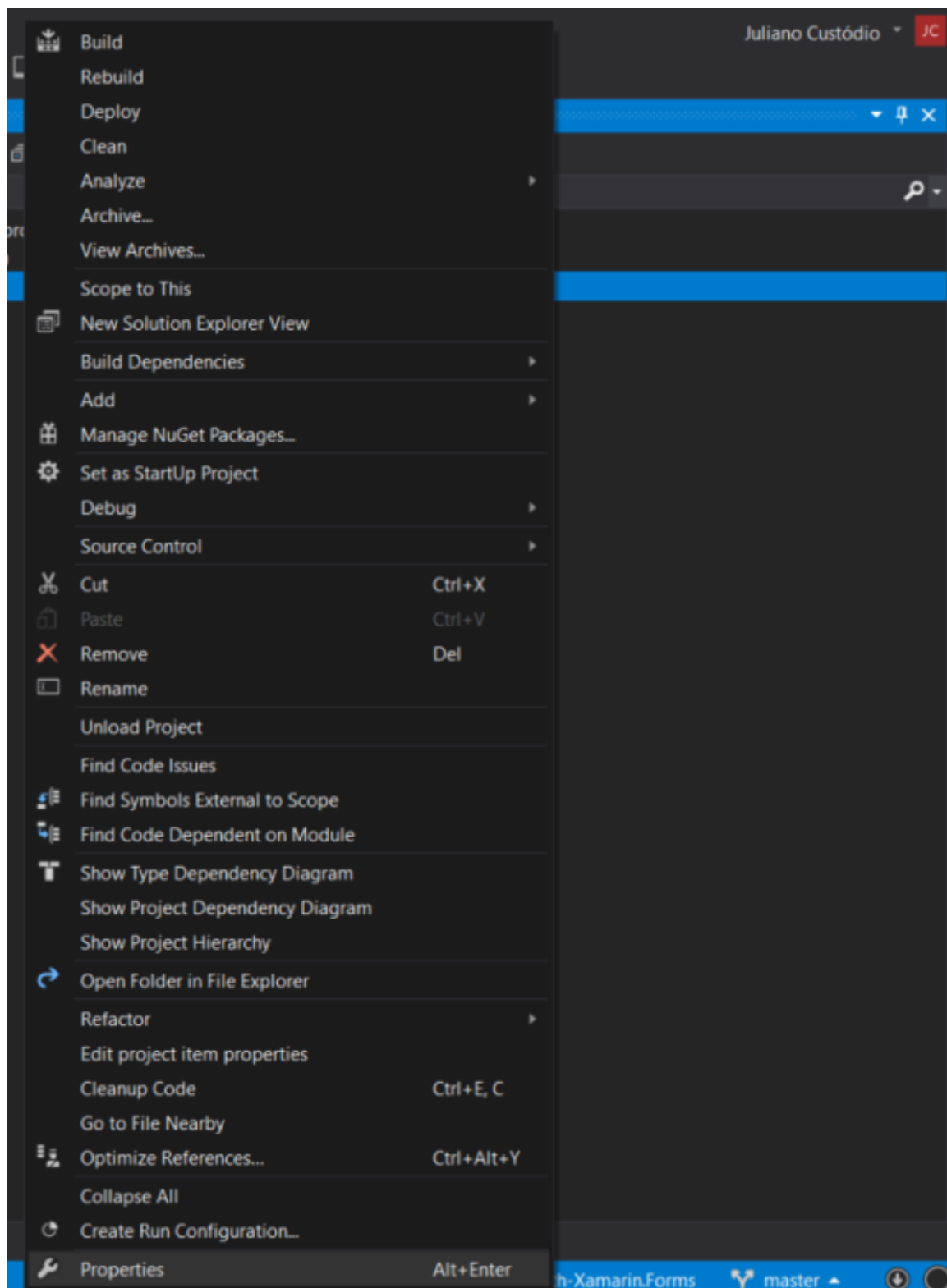
MainPage.xaml.cs

1	using System;
2	using System.Collections.ObjectModel;
3	using Plugin.BLE;
4	using Plugin.BLE.Abstractions.Contracts;
5	using Plugin.BLE.Abstractions.Exceptions;
6	using Xamarin.Forms;
7	
8	namespace DemoBluetooth
9	{
10	public partial class MainPage
11	{
12	IAdapter adapter;
13	IBluetoothLE bluetoothBLE;
14	ObservableCollection<IDevice> list;
15	IDevice device;
16	
17	public MainPage()
18	{
19	InitializeComponent();
20	
21	bluetoothBLE = CrossBluetoothLE.Current;
22	adapter = CrossBluetoothLE.Current.Adapter;
23	
24	list = new ObservableCollection<IDevice>();
25	DevicesList.ItemsSource = list;
26	
27	}
28	
29	private async void searchDevice(object sender, EventArgs e)
30	{
31	if (bluetoothBLE.State == BluetoothState.Off)
32	{
33	await DisplayAlert("Atenção", "Bluetooth desabilitado.", "OK");
34	}
35	else
36	{
37	list.Clear();
38	

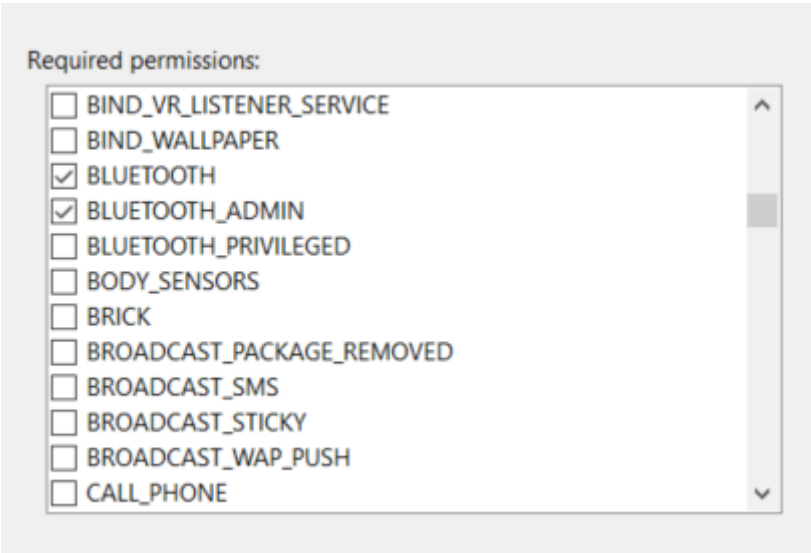
39	adapter.ScanTimeout = 10000;
40	adapter.ScanMode = ScanMode.Balanced;
41	
42	
43	adapter.DeviceDiscovered += (obj, a) =>
44	{
45	if (!list.Contains(a.Device))
46	list.Add(a.Device);
47	};
48	
49	await adapter.StartScanningForDevicesAsync();
50	
51	}
52	
53	}
54	
55	private async void DevicesList_OnItemSelected(object sender, SelectedItemChangedEvent
56	{
57	device = DevicesList.SelectedItem as IDevice;
58	
59	var result = await DisplayAlert("AVISO", "Deseja se conectar a esse dispositivo?", "Conec
60	
61	if (!result)
62	return;
63	
64	//Stop Scanner
65	await adapter.StopScanningForDevicesAsync();
66	
67	try
68	{
69	await adapter.ConnectToDeviceAsync(device);
70	
71	await DisplayAlert("Conectado", "Status:" + device.State , "OK");
72	
73	}
74	catch (DeviceConnectionException ex)
75	{
76	await DisplayAlert("Erro", ex.Message, "OK");
77	}
78	
79	}
80	}
81	}

ANDROID

Edite o manifesto para adicionar algumas permissões, para isso clique com o botão direito no projeto .android e selecione Properties.



No Android Manifest selecione as seguintes permissões.



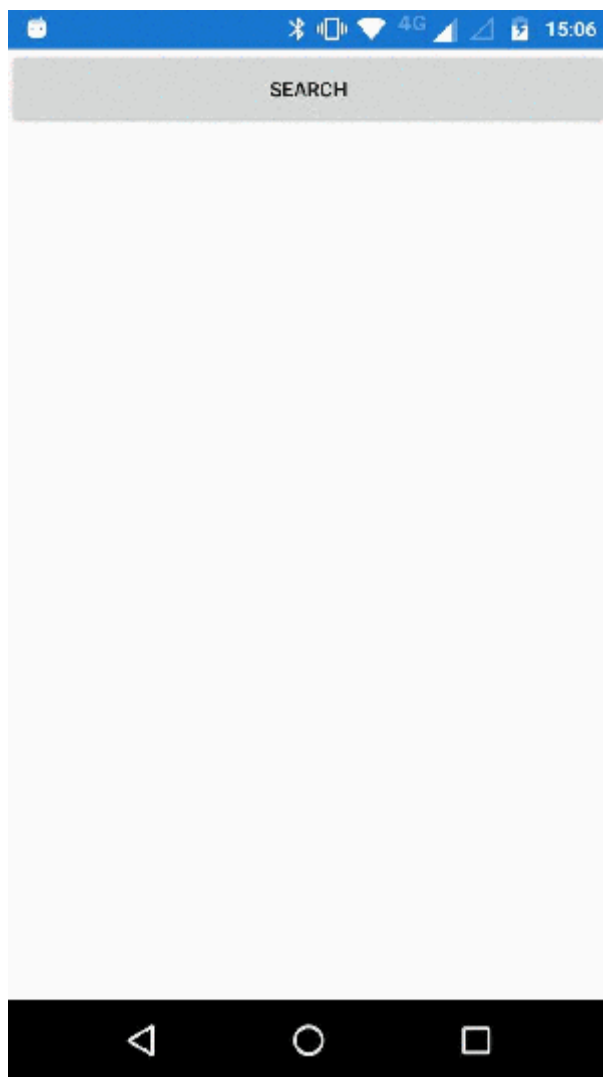
IOS

No projeto .iOS edite o arquivo Info.plist e adicione a seguinte permissão dentro de “dict” como demonstrado a seguir.

1	...
2	<key>NSBluetoothPeripheralUsageDescription</key>
3	<string>Esse aplicativo precisa acessar o Bluetooth.</string>
4	
5	</dict>
6	</plist>

view raw Info.plist hosted with ❤ by GitHub

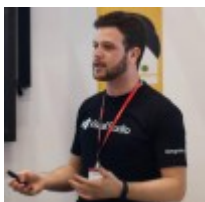
Resultado



Esse e todos os exemplos deste blog encontram-se disponíveis no GitHub.



(<https://github.com/juucustodio/Bluetooth-Xamarin.Forms>).



Publicado por Juliano Custódio

É formado em Análise e Desenvolvimento de Sistemas pela Faculdade de Tecnologia de Sorocaba, Software Developer na empresa Algorama, faz parte da lista de autores do Planet Xamarin, fundador da comunidade Xamarin Sorocaba, palestrante e escreve artigos sobre Xamarin.

[Ver todos os posts de Juliano Custódio](#)

2 comentários em “Bluetooth – Xamarin.Forms”

1.

cyberlacs disse:

19/02/2018 às 17:59

Amigo estou com problemas de incompatibilidade no meu projeto xamarin e o plugin BLE. Criei um projeto Xamarin que nesta versão do visual studio não temos a opção PCL e somente .Net Framework onde quase tudo não tem compatibilidade, gostaria de saber como faço para que esta biblioteca funcione em meu projeto, você tem ideia de como fazer isto ?

Outra dúvida é como enviar para um dispositivo informações, no meu caso quero enviar somente um char 'L' ou 'E' gostaria de obter também esta informação.

Obrigado por compartilhar 😊

Responder

Juliano Custódio disse:

23/02/2018 às 11:30

Olá,

No momento existe uma issue no repositório do projeto plugin BLE, se quiser acompanhar essa issue o link é: <https://github.com/xabre/xamarin-bluetooth-le/issues/267>

Caso você necessite de uma solução imediata, recomendo baixar o projeto no github e utilizar esse método para realizar a conversão.

https://blogs.msdn.microsoft.com/premier_developer/2017/10/27/converting-pcl-portable-class-libraries-libraries-to-net-standard-class-libraries/

Da uma olhada no readme do projeto, lá tem exemplos de como vc pode trocar informações com o dispositivo: <https://github.com/xabre/xamarin-bluetooth-le/blob/master/README.md>

Espero ter ajudado. 😊

Responder

Ⓜ AUTOMATTIC.