

Macoratti.net Xamarin

Android - Acessando um serviço REST com RestSharp



Curso de Xamarin Forms Vídeo Aulas
Desenvolva para Android, iOS e Windows Phone



Neste artigo vou mostrar como acessar um serviço REST no [Xamarin Android](#) usando a biblioteca **RestSharp** no Visual Studio 2015 e a linguagem C#.

Curso C# Vídeo Aulas
Do básico ao intermediário

Por um preço justo

Vamos iniciar com resumo sobre REST...

O que é REST ?

Representational State Transfer (REST) é um estilo de arquitetura para a construção de serviços na web. As solicitações REST são feitas em HTTP usando os mesmos verbos HTTP que os navegadores da Web usam para recuperar páginas da Web e enviar dados para os servidores. Os verbos são:

GET - esta operação é usada para recuperar dados do serviço da web.

POST - esta operação é usada para criar um novo item de dados no serviço web.

PUT - esta operação é usada para atualizar um item de dados no serviço da web.

PATCH - esta operação é usada para atualizar um item de dados no serviço da Web, descrevendo um conjunto de instruções sobre como o item deve ser modificado.

DELETE - esta operação é usada para apagar um item de dados no serviço web.

As APIs de serviço da Web que aderem ao REST são chamadas de APIs RESTful e são definidas usando:

1- Uma URI base.

2- Métodos HTTP, como GET, POST, PUT, PATCH ou DELETE.

3- Um tipo de mídia para os dados, como JavaScript Object Notation (JSON).

A simplicidade do REST ajudou a torná-lo o principal método para acessar serviços da Web em aplicativos móveis.

Além do REST podemos usar também os seguintes recursos para gerenciar serviços na web:

- [SOAP](#)
- [ASP .NET Web Services \(ASMX\)](#)
- [ASP .NET Web API](#)
- [Windows Communication Foundation \(WCF\)](#)

Para mostrar como podemos usar os recursos do REST para acessar informações na web vou criar um exemplo bem simples que acessa informações do site : <http://www.androidcodec.com> usando os recursos da biblioteca **RestSharp** que você pode consultar neste link: <http://restsharp.org/>.

Recursos usados:

- [Visual Studio Community 2015](#) ou Xamarin Studio
- [Xamarin](#)
- RestSharp - <http://restsharp.org/>
- Emulador Android virtual ou físico ([veja como emular usando o Vysor](#))

Nota: Baixe e use a versão Community 2015 do VS ela é grátis e é equivalente a versão Professional.

Criando o projeto no VS Community 2015

Abra o [VS 2015 Community](#) e clique em **New Project**;


Selecione a linguagem Visual C# e o template **Android -> Blank App(Android)**

Informe o nome um nome adequado ao seu projeto, eu vou usar o nome **Droid_RestApi**, e clique no botão **OK**;

Abra o arquivo **Main.axml** na pasta **Resources/layout** no modo **Designer** e inclua a partir da ToolBox os seguintes controles:

- **1 ScrollView**
- **1 Button - Acessar RestApi , id=btnAcessar**
- **1 TextView - id=txtResposta**

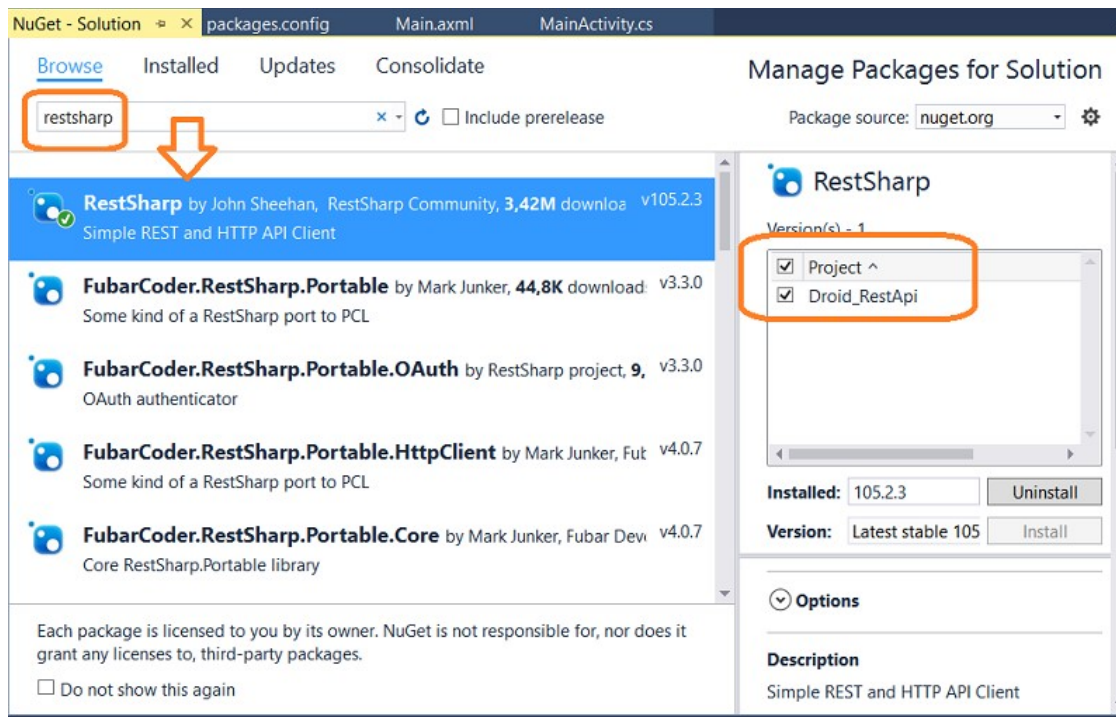
Abaixo vemos o leiaute no emulador do Xamarin e ao lado o respectivo código XML gerado :

	<pre><?xml version="1.0" encoding="utf-8"?> <ScrollView xmlns:android="http://schemas.android.com/apk/res/android" android:layout_width="match_parent" android:layout_height="wrap_content" android:minWidth="25px" android:minHeight="25px"> <LinearLayout android:orientation="vertical" android:layout_width="match_parent" android:layout_height="match_parent" android:minWidth="25px" android:minHeight="25px" android:background="@android:color/holo_orange_light"> <Button android:text="Acessar RestApi" android:layout_width="match_parent" android:layout_height="wrap_content" android:id="@+id/btnAcessar" /> <TextView android:text="" android:textAppearance="?android:attr/textAppearanceMedium" android:layout_width="match_parent" android:layout_height="match_parent" android:id="@+id/txtResposta" android:textColor="@android:color/black" android:background="#d3ffce" /> </LinearLayout> </ScrollView></pre>
--	---

Vamos agora incluir uma referência a biblioteca **RestSharp** em nosso projeto.

No menu **Tools** clique em **Nuget Package Manager** e a seguir em [Manage Nuget Packages for Solution](#);

Selecione a guia **Browse** e digite **restsharp**. Localize a biblioteca e instale-a no projeto:



Agora podemos usar os recursos da **RestSharp**.

Abra o arquivo **MainActivity** na raiz do projeto e inclua o código abaixo neste arquivo.

```
using Android.App;
using Android.Widget;
using Android.OS;
using RestSharp;

namespace Droid_RestApi
{
    [Activity(Label = "RestAPI", MainLauncher = true, Icon = "@drawable/icon")]
    public class MainActivity : Activity
    {
        ProgressDialog dialog;
        TextView txtv1;

        protected override void OnCreate(Bundle bundle)
        {
            base.OnCreate(bundle);

            // Set our view from the "main" layout resource
            SetContentView (Resource.Layout.Main);
            Button button = FindViewById<Button>(Resource.Id.btnAcessar);
            txtv1 = FindViewById<TextView>(Resource.Id.txtResposta);

            button.Click += Button_Click;
        }

        private void Button_Click(object sender, System.EventArgs e)
        {
            dialog = new ProgressDialog(this);
            dialog.SetMessage("Aguarde...");
            dialog.SetProgressStyle(ProgressDialogStyle.Spinner);
            dialog.Show();

            var cliente = new RestClient("http://www.androidcoded.com");
            var request = new RestRequest("/", Method.GET);
```

```

request.AddHeader("Content-type", "application/json");
cliente.ExecuteAsync(request, response =>
{
    RunOnUiThread(delegate
    {
        dialog.Dismiss();
        var responseFacebook = response.Content;
        txtv1.Text = responseFacebook.ToString();
        Toast.MakeText(this, "reposta", ToastLength.Long);
    });
});
}
}
}

```

Vamos entender o código :

Usamos um **ProgressDialog()** para exibir uma caixa diálogo e uma mensagem de texto mostrando um indicador de progresso da operação:

```

dialog = new ProgressDialog(this);
dialog.SetMessage("Aguarde...");
dialog.SetProgressStyle(ProgressDialogStyle.Spinner);
dialog.Show();

```

A seguir criamos uma instância da classe **RestClient()** passando a url de destino para o construtor da classe para que a comunicação seja estabelecida. Depois criamos uma instância da classe **RestRequest()** informando o tipo de requisição (GET) e definimos o cabeçalho da requisição como sendo do tipo '*application/json*':

```

var cliente = new RestClient("http://www.androidcoded.com");
var request = new RestRequest("/", Method.GET);
request.AddHeader("Content-type", "application/json");

```

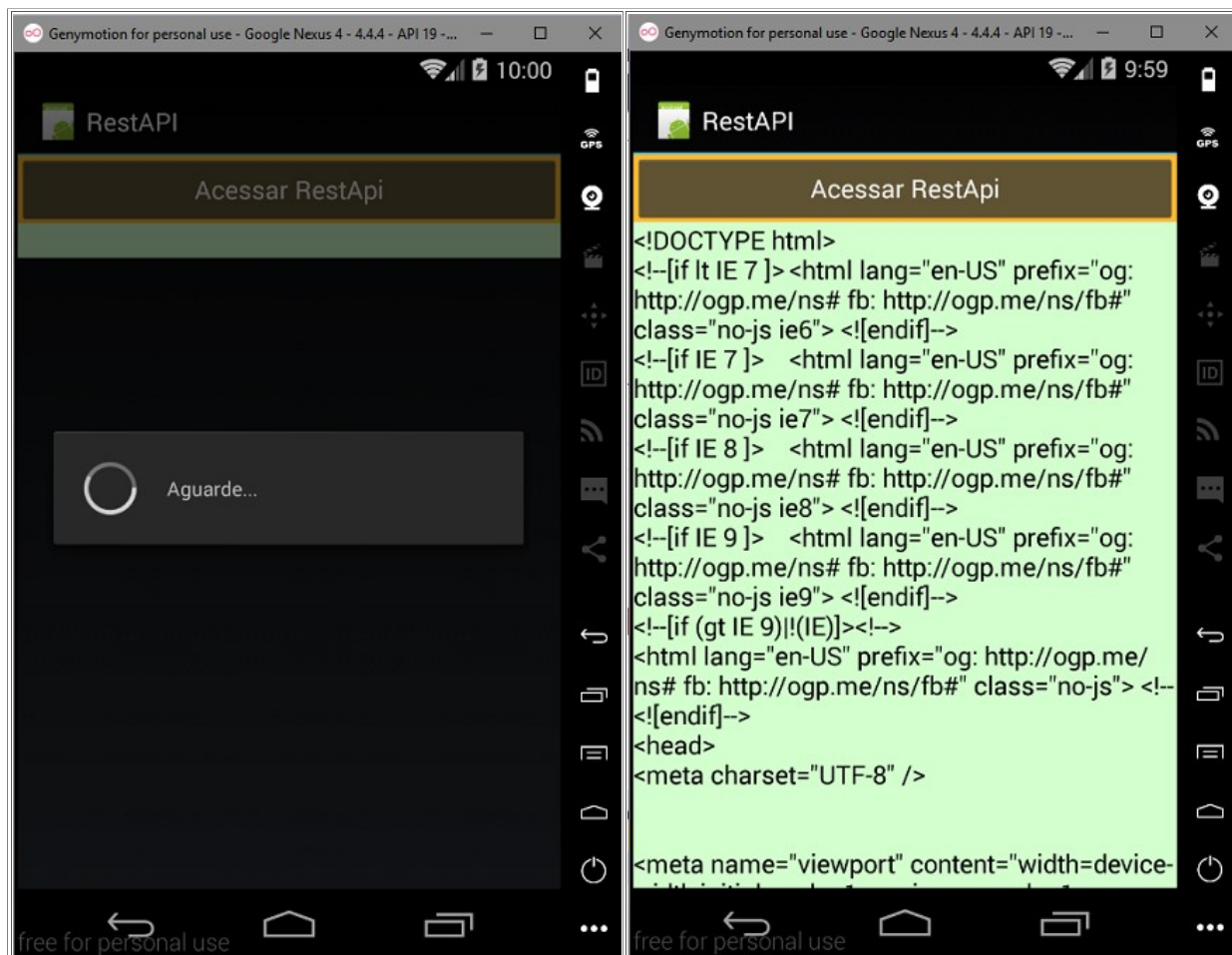
Concluindo usamos o método **ExecuteAsync()** onde vamos usar o método **runOnUiThread** para ter acesso a thread principal do Android e obter e exibir a resposta no **TextView**:

```

cliente.ExecuteAsync(request, response =>
{
    RunOnUiThread(delegate
    {
        dialog.Dismiss();
        var responseSite = response.Content;
        txtv1.Text = responseSite.ToString();
        Toast.MakeText(this, "reposta", ToastLength.Long);
    });
});

```

Executando o projeto usando o emulador **Genymotion** iremos obter o seguinte resultado:



Pegue o projeto aqui : [Droid RestApi.zip](#) (sem as referências)

Porque a palavra da cruz é loucura para os que perecem; mas para nós, que somos salvos, é o poder de Deus. Porque está escrito: Destruirei a sabedoria dos sábios, E aniquilarei a inteligência dos inteligentes.

1 Coríntios 1:18,19

[Veja os Destaques e novidades do SUPER DVD Visual Basic \(sempre atualizado\) : clique e confira !](#)

Quer migrar para o VB .NET ?

- Veja mais sistemas completos para a plataforma .NET no [Super DVD .NET](#) , confira...
- [Curso Básico VB .NET - Vídeo Aulas](#)

Quer aprender C# ??

- Chegou o [Super DVD C#](#) com exclusivo material de suporte e vídeo aulas com curso básico sobre C#.
- [Curso C# Basico - Video Aulas](#)

Quer aprender os conceitos da Programação Orientada a objetos ?

- [Curso Fundamentos da Programação Orientada a Objetos com VB .NET](#) NEW

Quer aprender o gerar relatórios com o ReportViewer no VS 2013 ?

- [Curso - Gerando Relatórios com o ReportViewer no VS 2013 - Vídeo Aulas](#) NEW

Referências:

- [Seção VB .NET do Site Macoratti.net](#)
- [Super DVD .NET - A sua porta de entrada na plataforma .NET](#)
- [Super DVD Vídeo Aulas - Vídeo Aula sobre VB .NET, ASP .NET e C#](#)
- [Super DVD C# - Recursos de aprendizagens e vídeo aulas para C#](#)
- [Seção C# do site Macoratti.net](#)
- [Seção ASP .NET do site Macoratti .net](#)
- [Curso Básico VB .NET - Vídeo Aulas](#)
- [Curso C# Básico - Vídeo Aulas](#)
- [Curso Fundamentos da Programação Orientada a Objetos com VB .NET](#) NEW
- [Macoratti .net | Facebook](#)
- [macoratti - YouTube](#)
- [Jose C Macoratti \(@macoratti\) | Twitter](#)
- [VB.NET 2005 - Controles - Macoratti.net](#)
- [Seção de Jogos do site Macoratti .net](#)
- [Xamarin - Desenvolvimento Multiplataforma com C# ... - Macoratti.net](#)
- [Xamarin - Apresentando Xamarin.Forms - Macoratti.net](#)
- [Xamarin.Forms - Olá Mundo - Criando sua primeira ... - Macoratti.net](#)
- [Xamarin.Forms - Olá Mundo - Anatomia da aplicação - Macoratti.net](#)
- <https://developer.xamarin.com/api/type/Android.App.AlertDialog/>
- [Xamarin Android - Tratando eventos de forma declarativa](#)
- [Seção Mobile/Xamarin do site Macoratti .net](#)
- <https://developer.android.com/reference/android/app/Activity.html>
- <https://developer.xamarin.com/api/type/Android.Widget.ProgressBar/>

[José Carlos Macoratti](#)