



Renato Teixeira <ra.teixeira@gmail.com>

C# WebDeveloper - COTI Informatica (Utilizando a API do Google Analytics com ASP.NET MVC e C#)

1 mensagem

Sergio Mendes <sergio.coti@gmail.com>

6 de março de 2018 02:28

Para: Bruno Orzechowski <bruno.orze@gmail.com>, Gustavo Rosa <gztavo.rosa@gmail.com>, Claudio Pimentel <iclaudp@gmail.com>, João Carlos Ribeiro da Silva <jcrs23rj@hotmail.com>, Renato Teixeira <ra.teixeira@gmail.com>, analistathiago13@hotmail.com, Alexandre Rodrigues da Silva <sxandeogrande@yahoo.com.br>, rosangela.scorzello@susep.gov.br, lfranca19@hotmail.com

Bom artigo:

Fonte: <http://www.princiweb.com.br/blog/programacao/google-apis/utilizando-a-api-do-google-analytics-com-aspnet-mvc-e-csharp.html>

Utilizando a API do Google Analytics com ASP.NET MVC e C#

Depois da boa receptividade do meu último post “[Google Maps API v3: Busca de endereço e Autocomplete](#)“, resolvi dar continuidade à posts que tratam sobre as APIs do Google. Dessa vez, irei mostrar pra vocês como utilizar a API do Google Analytics utilizando um projeto em [ASP.NET MVC](#) e C#.

O Google Analytics é uma ferramenta gratuita de análise de métricas de sites. Caso você utilize o Google Analytics em seu site, você poderá ver diversas estatísticas, como visitas, páginas com mais acessos, tecnologias utilizadas, etc. Esses dados são de extrema valia para que os clientes ou os profissionais de marketing digital, possam analisá-los e tomar ações de modo a aumentar o ROI de seus sites.

Aprendendo a utilizar a API do Google Maps, você pode exibir as estatísticas do site para o seu cliente de forma mais direta e didática. Além disso, se você tiver uma equipe de SEO na sua empresa, você pode automatizar os relatórios de SEO, que são feitos de forma manual, repetitiva e tediosa.

Vamos começar?

Obtendo a API Access

Para que possamos aumentar o nosso [limite e quota da API do Google Analytics](#), é necessário que obtenhamos uma chave da API exclusiva para o nosso projeto.

Para obter a chave, entre no [Console de APIs](#) com a sua conta do Google (a mesma que você usa para acessar o Analytics). No menu, vá em Services e habilite o Analytics API, clicando em OFF.

Após isso, vá em *API Access*. No nosso caso, iremos utilizar a opção *Simple API Access*, pois utilizaremos a versão 2.4 da API do Google Analytics. Nessa página, guarde o valor da *API key*, pois utilizaremos essa informação mais adiante.

Escolhendo a biblioteca

Para utilizarmos as APIs do Google na plataforma .NET, temos duas opções: A biblioteca [google-gdata](#) e a [biblioteca google-api-dotnet-client](#).

As diferenças dessas duas bibliotecas, é que o *google-api-dotnet-client* é beta e é recomendada pelo Google caso você queira usar a versão 3.0 da API. Já a biblioteca *google-gdata*, é recomendada pelo Google caso você queira usar a versão 2.4 da API. Nesse tutorial nós utilizaremos a biblioteca *google-gdata*.

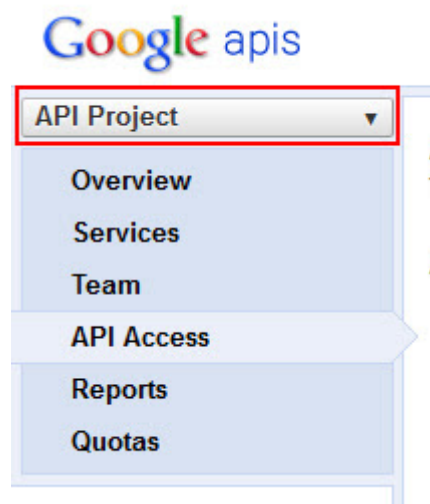
No Visual Studio, utilize o NuGet para baixar o pacote *Google.GData.Analytics*.

Que comece a diversão

Com o pacote *Google.GData.Analytics* referenciado ao seu projeto, vamos escrever algumas linhas de código que irão autenticar o nosso projeto na API do Google Analytics.

Para autenticar, basta criar um objeto do tipo *AnalyticsService*, e informar o nome da sua aplicação, seu usuário (e-mail do Gmail) e a sua senha.

O nome da aplicação você irá pegar no Console de APIs. Ele fica no canto esquerdo da tela, acima de *Overview*. Clicando em cima do nome, você pode renomeá-lo se preferir.



O nosso código de autenticação ficará assim:

```
1  string usuario = "email@gmail.com";
2  string senha = "suasenha";
3
4  var service = new AnalyticsService("API Project");
5
6  service.setUserCredentials(usuario, senha);
```

Dai-me os dados, Google

A obtenção de dados do Google Analytics é bem tranquila. Antes de eu dar um exemplo para obter o número de visitas de um determinado período de tempo, é bom que você tenha a referência das métricas e dimensões disponíveis na API.

Com as dimensões e métricas em mãos, você saberá como obter qualquer dado disponível na API. Você pode usar a [Referência de Dimensões e Métricas para Analytics](#), e para testar na prática essas dimensões e métricas, você pode usar o Google Analytics Query Explorer 2. Recomendo fortemente que você estude a referência utilizando a [Query Explorer 2](#), pois a união dos dois torna o aprendizado muito mais prático e dinâmico.

Vamos então pegar todas as visitas e page views do nosso site, de um determinado período. Para isso, nós:

1. Instanciamos um objeto *DataQuery*, que contém o endereço do Feed da API do Google Analytics com o parâmetro *key*, que você deve colocar a sua *API Key*.

2. Definimos o atributo "ids". Para obter o valor desse atributo, entre no [Query Explorer 2](#) logado com a sua conta do Google (a mesma que você usa para acessar o Analytics), clique em "Authorize Access", e o valor do atributo ids irá aparecer no formulário;
3. Definimos as dimensões e métricas (separados por vírgula);
4. Definimos o período: Data de início e término;
5. Por fim, a ordenação dos dados.

```
1 string apiKey = "";
2 string feedURL = "https://www.googleapis.com/analytics/v2.4/data?key=" + apiKey
3
4 var query = new DataQuery(feedURL);
5
6 query.Ids = "";
7 query.Dimensions = "ga:date";
8 query.Metrics = "ga:visits,ga:pageviews";
9
10 query.GAStartDate = DateTime.Now.AddDays(-16).ToString("yyyy-MM-dd");
11 query.GAEndDate = DateTime.Now.AddDays(-1).ToString("yyyy-MM-dd");
12
13 query.Sort = "ga:date";
14 query.StartIndex = 1;
15
16 var dataFeedVisitas = service.Query(query);
```

Agora que nós temos todos os dados, vamos exibí-los:

```
1 var builder = new StringBuilder();
2
3 foreach (DataEntry dado in dataFeedVisitas.Entries)
4 {
5     var visitas = dado.Metrics[0].Value;
6     var pageViews = dado.Metrics[1].Value;
7     var data = dado.Dimensions[0].Value;
8
9     builder.Append("Data: " + data);
10    builder.Append("Visitas: " + visitas);
11    builder.Append("Page views: " + pageViews);
12 }
13
14 Response.Write(builder.ToString());
```

Maravilha! Agora temos os dados que queríamos. Observe que para pegarmos os dados das métricas, usamos o *Metrics*, e para pegarmos as dimensões, usamos o *Dimensions*.

O código que coloquei aqui nesse post irá funcionar tanto em Web Forms quanto em MVC. Deixo no [meu GitHub](#) um projeto em [ASP.NET MVC](#) e [C#](#) com o código mais bem organizado, caso vocês queiram baixar para referência ou melhorá-lo.

--

Sergio da Silva Mendes



"Me contaram e eu esqueci, Vi e entendi, Fiz e aprendi."
Confúcio - Filósofo Chinês

