

بالاخره گوگل کار تهیه API مخصوص ابزار [Analytics](#) خود را [به پایان رساند](#) و اکنون برنامه نویسی‌ها می‌توانند همانند سایر سرویس‌های گوگل از این ابزار گزارشگیری نمایند.

خلاصه کاربردی این API ، دو صفحه تعاریف پروتکل ( + ) و ریز مواردی ( + ) است که می‌توان گزارشگیری نمود. هنوز کتابخانه [google-gdata](#) جهت استفاده از این API به روز رسانی نشده است؛ بنابراین در این مقاله سعی خواهیم کرد نحوه کار با این API را از صفر بازنویسی کنیم. مطابق صفحه تعاریف پروتکل، سه روش اعتبارسنجی جهت دریافت اطلاعات API معرفی شده است که در اینجا از روش ClientLogin که مرسوم‌تر است استفاده خواهیم کرد. مطابق مثالی که در آن صفحه قرار دارد، اطلاعاتی شبیه به اطلاعات زیر را باید ارسال و دریافت کنیم:

```
POST /accounts/ClientLogin HTTP/1.1
User-Agent: curl/7.15.1 (i486-pc-linux-gnu) libcurl/7.15.1
OpenSSL/0.9.8a zlib/1.2.3 libidn/0.5.18
Host: www.google.com
Accept: */*
Content-Length: 103
Content-Type: application/x-www-form-urlencoded
accountType=GOOGLE&Email=userName@google.com&Passwd=myPasswr&source=curl-tester-1.0&service=analytics

HTTP/1.1 200 OK
Content-Type: text/plain
Cache-control: no-cache
Pragma: no-cache
Date: Mon, 02 Jun 2008 22:08:51 GMT
Content-Length: 497
SID=DQ...
LSID=DQAA...
Auth=DQAAAG8...
```

در دات نت فریم ورک، این کار را به صورت زیر می‌توان انجام داد:

```
string getSecurityToken()
{
    if (string.IsNullOrEmpty(Email))
        throw new NullReferenceException("Email is required!");

    if (string.IsNullOrEmpty>Password)
        throw new NullReferenceException("Password is required!");

    WebRequest request = WebRequest.Create("https://www.google.com/accounts/ClientLogin");
    request.Method = "POST";

    string postData = "accountType=GOOGLE&Email=" + Email + "&Passwd=" + Password +
"&service=analytics&source=vahid-testapp-1.0";
    byte[] byteArray = Encoding.ASCII.GetBytes(postData);

    request.ContentType = "application/x-www-form-urlencoded";
    request.ContentLength = byteArray.Length;

    using (Stream dataSt = request.GetRequestStream())
    {
        dataSt.Write(byteArray, 0, byteArray.Length);
    }

    string auth = string.Empty;
    using (WebResponse response = request.GetResponse())
    {
        using (Stream dataStream = response.GetResponseStream())
        {
            using (StreamReader reader = new StreamReader(dataStream))
            {
                string responseFromServer = reader.ReadToEnd().Trim();
```

```

string[] tokens = responseFromServer.Split('\n');
foreach (string token in tokens)
{
    if (token.StartsWith("SID="))
        continue;

    if (token.StartsWith("LSID="))
        continue;

    if (token.StartsWith("Auth="))
    {
        auth = token.Substring(5);
    }
    else
    {
        throw new AuthenticationException("Error authenticating Google user " +
Email);
    }
}
}
}
return auth;
}

```

همانطور که ملاحظه می‌کنید به آدرس <https://www.google.com/accounts/ClientLogin> ، اطلاعات postData با متد POST ارسال شده (دقیقا مطابق توضیحات گوگل) و سپس از پاسخ دریافتی، مقدار نشانه Auth را جدا نموده و در ادامه عملیات استفاده خواهیم کرد. وجود این نشانه در پاسخ دریافتی به معنای موفقیت آمیز بودن اعتبار سنجی ما است و مقدار آن در طول کل عملیات باید نگهداری شده و مورد استفاده مجدد قرار گیرد.

سپس مطابق ادامه توضیحات API گوگل باید لیست پروفایل‌هایی را که ایجاد کرده‌ایم پیدا نمائیم:

```

string getAvailableProfiles(string authToken)
{
    return fetchPage("https://www.google.com/analytics/feeds/accounts/default", authToken);
}

```

متد fetchPage را از [پیوست](#) این مقاله می‌توانید دریافت نمائید. خروجی یک فایل xml است که با انواع و اقسام روش‌های موجود قابل آنالیز است، از کتابخانه‌های XML دات نت گرفته تا Linq to xml و یا [روش serialization](#) که من روش آخر را ترجیح می‌دهم. مرحله بعد، ساخت URL زیر و دریافت مجدد اطلاعات مربوطه است:

```

string url =
string.Format("https://www.google.com/analytics/feeds/data?ids={0}&metrics=ga:pageviews&start-
date={1}&end-date={2}", id, from, to);
return fetchPage(url, auth);

```

و سپس آنالیز اطلاعات xml دریافتی، جهت استخراج تعداد بار مشاهده صفحات یا pageviews استفاده شده در این مثال. لیست کامل مواردی که قابل گزارشگیری است، در صفحه [Dimensions & Metrics Reference](#) گوگل ذکر شده است.

فایل‌های کلاس‌های مورد استفاده را [از اینجا](#) دریافت نمائید.

مثالی در مورد نحوه استفاده از آن:

```

CGoogleAnalytics cga = new CGoogleAnalytics
{
    Email = "username@gmail.com",
    Password = "password",
    From = DateTime.Now.Subtract(TimeSpan.FromDays(1)),
    To = DateTime.Now.Subtract(TimeSpan.FromDays(1))
};

```

```
List<CGoogleAnalytics.SitePagePreviews> pagePreviews =  
    cga.GetTotalNumberOfPageViews();  
  
foreach (var list in pagePreviews)  
{  
    //string site = list.Site;  
    //int pw = list.PagePreviews;  
}
```

## نظرات خوانندگان

نویسنده: نیما  
تاریخ: ۱۳۸۸/۰۲/۱۲ ۱۰:۵۸:۰۰

سلام استاد نصیری  
با تشکر از مطلب مفید شما. آیا میشه مثلا اون قسمت Map گزارش گوگل رو گذاشت توی سایت؟  
ممنون از شما

نویسنده: وحید نصیری  
تاریخ: ۱۳۸۸/۰۲/۱۲ ۱۲:۳۹:۰۰

سلام،  
API فوق امکان استخراج اطلاعات مربوط به کشورهای مختلف و بازدید کننده‌های مختلف را نیز می‌دهد. (به صفحه Dimensions & Metrics Reference مراجعه کنید)  
پس از استخراج داده‌ها، رسم نمودار و نقشه با خودتان است. فقط اطلاعات و آمار خام را به شما می‌دهد.