

رایانش ابری مفهوم نسبتاً جدیدی در عرصه‌ی فناوری اطلاعات است و در حال گسترش می‌باشد. به طور خلاصه رایانش ابری به همه چیز اعم از برنامه کاربردی (Application)، سکو ی (Platform) توسعه نرم افزار، سخت افزار و زیرساخت، به عنوان سرویس نگاه می‌کند. زیرساخت‌های موجود در مراکز داده (Data Center) به انضمام نرم‌افزارهایی که در آن قرار دارند، مجموعه‌ای را تشکیل می‌دهند که ابر نامیده می‌شود. به عبارت ساده‌تر رایانش ابری یعنی استفاده-اشتراکی از برنامه‌ها و منابع یک محیط شبکه‌ای برای انجام یک کار، بدون این که مالکیت، مدیریت منابع شبکه و سخت-افزار و برنامه‌ها، برای استفاده کننده مهم باشد. در رایانش ابری منابع کامپیوترها، برای انجام یک کار استفاده می‌شوند و داده‌های مربوط به پردازش، در هیچ کدام از کامپیوترهای شخصی ذخیره نمی‌شوند، بلکه در جای دیگری در داخل همان منابع شبکه، ذخیره می‌شوند تا در زمان و مکان دیگری قابل دسترسی باشند.

بر همین اساس شرکت‌های پیشرو در زمینه فناوری اطلاعات به ارائه سرویس‌هایی تحت عنوان خدمات رایانش ابری پرداخته اند و هدف از این سری مطالب ارائه شده، شرح مختصری بر سرویس‌های ارائه شده می باشد. در قسمت اول به معرفی سرویس های شرکت گوگل پرداخته می شود و در قسمت‌های بعدی، سرویس‌های شرکت‌های مایکروسافت و آمازون معرفی می‌گردد.

سرویس‌های رایانش ابری گوگل، در زیر دو چتر قرار دارند. گروه اول شامل مجموعه گسترده‌ای از برنامه‌های محبوب گوگل مانند Google Earth ، Google Mail ، Google Health ، Google Doc هستند که با کلیک بر روی گزینه More و Even More که در بالای صفحه اصلی گوگل قرار دارند، می‌توان به آن‌ها دسترسی پیدا کرد.

دومین محصول مبتنی بر ابر گوگل، ابزار توسعه PaaS گوگل است. این سکو در سال 2008 برای توسعه برنامه‌های تحت وب، با استفاده از زیرساخت گوگل به نام موتور Google App معرفی شد. هدف از آن قادر ساختن توسعه دهندگان برای ساخت و استقرار برنامه‌های وب بدون نگرانی از زیرساختی است که برنامه بر رویش اجرا می‌شود. برنامه‌های این موتور، با زبان‌های سطح بالا به ویژه جاوا و پایتون و در چارچوب GAE نوشته می‌شوند. گوگل به منظور گسترش این نوع برنامه‌ها یک سطح رایگان مشخص از سرویس را ارائه می‌دهد و زمانی که برنامه از سطح مشخصی از بار پردازشی، ذخیره‌سازی و پهنای باند شبکه فراتر رفت، آنگاه شارژها بر مبنای میزان استفاده محاسبه می‌شود.

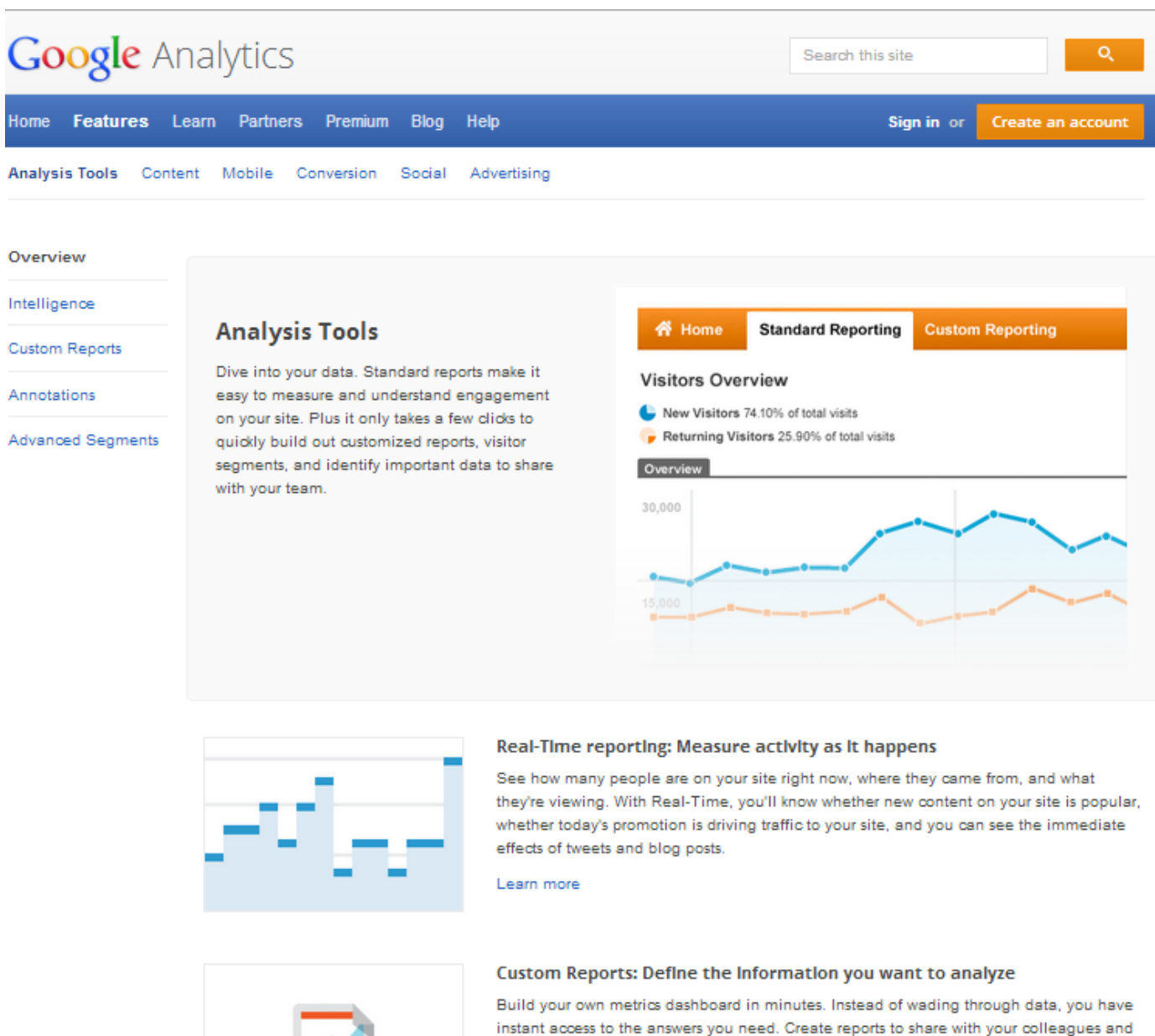
برنامه‌های GAE را باید به گونه‌ای نوشت که با زیرساخت گوگل وفق یابند. این مسئله، باعث محدودیت برنامه‌های قابل اجرا در GAE می‌گردد و علاوه بر آن، انتقال برنامه‌ها به سکوی GAE و یا انتقال از این سکو به سایر سکوها موجود دشوار می‌شود.

از میان سرویس‌های ابری رایگان ارائه شده از سوی گوگل، به معرفی سرویس آنالیز گوگل بسنده کرده و تمرکز اصلی بر روی سکوی توسعه نرم‌افزاری این شرکت (GAE) می‌باشد.

Google Analytics


به اختصار GA نامیده می‌شود و یک ابزار آماری است که تعداد و انواع بازدیدکنندگان وب-سایت و نحوه استفاده از وب-سایت را اندازه‌گیری می‌کند. این محصول بر روی بسته تحلیلی Urchin 5 که گوگل در سال 2006 آن را خریداری نمود، ساخته شده است. این سرویس رایگان عرضه می‌شود و فرآیند تحلیل را با استفاده از یک قطعه کد جاوا اسکریپت به نام Google Analytics Tracking Code با پیاده‌سازی در تگ صفحه وب انجام می‌شود.

این کد با اولین بارگذاری صفحه در سیستم-کاربران، به جمع‌آوری اطلاعات مورد نیاز پرداخته و برای پردازش به سرورهای GA باز پس می‌فرستد. این کد با کمک Cookie مرورگر اطلاعات مورد نیاز را جمع‌آوری می‌نماید.



آشنایی با Google App Engine

GAE یک سکو به عنوان سرویس می-باشد و مبتنی بر ابر گوگل است و بر روی زیر ساخت- گوگل مستقر شده است.




Home Products Solutions Pricing Customers Partners

Contact sales for enterprise level support? - or - [Try it now](#)


[Google App Engine](#) [Google Compute Engine](#) [Google Cloud Storage](#) [Google BigQuery](#) [Google Cloud SQL](#) [More Products](#)

Google App Engine


Create apps on Google's platform that are easy to manage and scale. Benefit from the same systems and infrastructure that power Google's applications.



Focus on your apps
Let us worry about the underlying infrastructure and systems.




Scale infinitely
See your applications scale seamlessly from hundreds to millions of users.




Business ready
Premium paid support and 99.95% SLA for business users.

[Try it now](#)
Need enterprise level support?
[Contact sales](#)




[Download the Data Sheet »](#)
Learn how App Engine can become the platform for your next great idea




[See Current Pricing »](#)
Pay only for what you use with no upfront costs or investments


Use cases



Websites
Build fast and secure websites on Google App Engine and never worry about servers or machines.



Business Applications
Build business applications on Google App Engine and access your applications from any device.



Mobile and gaming Apps
Build mobile backends, Chrome and Android games on Google App Engine and see it scale seamlessly.

[See all case studies »](#)

این سرویس توسعه دهندگان را قادر می‌سازد تا برنامه‌های وب ایجاد کرده و بر روی سرور-های گوگل مستقر سازند و گوگل مدیریت زیرساخت را بر عهده گیرد و اعمالی مانند نظارت، برطرف کردن اشکالات احتمالی، خوشه بندی، مدیریت و هله‌سازی ماشین‌های مجازی و غیره را انجام دهد. برای اجرای یک برنامه در GAE ابتدا باید استانداردهای سکوی گوگل رعایت شود. این استانداردها دامنه برنامه‌هایی که قابل اجرا می‌باشند را بسیار محدود می‌نماید و قابلیت حمل آن‌ها را کاهش می‌دهد.

محدودیت‌هایی که این سکوی ایجاد می‌کند، با خود مزایایی را به همراه می‌آورد که در زیر به آن‌ها اشاره می‌گردد:

وب سرویس-های پویا بر مبنای استانداردهای رایج

توسعه خودکار و توازن بار بین ماشین‌های و هله‌سازی شده که مورد استفاده وب سرویس است.

اعتبارسنجی با استفاده از API موجود در گوگل.

فضای ذخیره سازی ماندگار با قابلیت جستجو، مرتب سازی و مدیریت تراکنش.

صف کاری و زمان بندی کاری

محیط توسعه سمت مشتری (توسعه دهندگان) برای شبیه سازی GAE در سیستم محلی.

پشتیبانی از محیط اجرا جاوا و پایتون.

هنگامی که یک برنامه در GAE مستقر گردید، با استفاده از نام دامنه دلخواه یا با استفاده از آدرس تجاری Google Apps قابل دستیابی است. موتور Google Apps در حال حاضر برنامه‌هایی که در جاوا و پایتون نوشته شده است را پشتیبانی می‌کند و علاوه بر آن از زبان‌های ماشین مجازی جاوا و چندین چارچوب تحت وب پایتون که WSGI و CGI را پشتیبانی می‌کنند نیز با محیط GAE سازگاری دارند.

برنامه‌هایی که در GAE اجرا می‌شوند از سیستم عامل مستقل هستند یا به گفته گوگل بر روی Sand Box اجرا می‌شوند. این ویژگی GAE را قادر می‌سازد، سیستم را بهینه کند تا تقاضاهای وب، با بار ترافیکی فعلی منطبق شوند. همچنین برنامه‌ها را قادر می‌سازد با امنیت بالاتری کار کنند، زیرا تنها می‌توانند به کامپیوترهایی متصل شوند که آدرس‌های مشخصی دارند و سرویس‌ها را با استفاده از پروتکل Http و یا Https از پورت‌های شناخته شده پاسخ دهند. از طرف دیگر برنامه‌ها نیز به این میزان محدود شده که تنها فایل‌ها را بخوانند. آن‌ها حق نوشتن فایل به صورت مستقیم بر روی سیستم‌ها را ندارند و برای دستیابی به داده، باید از ذخیره داده در Cache یا سرویس ماندگار دیگری استفاده نمایند.

GAE یک سیستم انبار داده توزیع شده دارد که از پرس و جوها و تراکنش‌ها پشتیبانی می‌نماید. این انبار داده غیر رابطه‌ای است، اما اشیاء داده یا موجودیت‌هایی که خصوصیات لازم را دارند، ذخیره می‌نماید. به همین علت در پرس و جوها می‌توان از فیلتر نوع موجودیت بهره برد و آن‌ها را به ترتیب خصوصیات مرتب نمود.

در نهایت توجه به مدل قیمت‌گذاری گوگل قابل توجه است. گوگل برای تشویق توسعه دهندگان در نوشتن برنامه با استفاده از GAE، استقرار و توسعه برنامه را تا میزان مشخصی از منابع رایگان کرده است و با عبور از مقدار رایگان باید هزینه را به ازای مصرف پرداخت نمود. بر اساس جدول ارائه شده در سایت شرکت گوگل به ازای تجاوز از میزان مصرف رایگان، سیستم هزینه گذاری بر اساس تعرفه‌های زیر، اقدام به محاسبه حق شارژ می‌نماید و بدیهی است برای آگاهی از آخرین تعرفه‌ها و کسب اطلاعات بیشتر، مراجعه به [صفحه سایت شرکت گوگل](#) توصیه می‌شود:

مبلغ به ازای هر یک ساعت استفاده از CPU معادل 0.08 دلار

داده ذخیره شده به ازای هر گیگابایت در ماه معادل 0.18 دلار

پهنای باند خروجی به ازای هر گیگابایت معادل 0.12 دلار

پهنای باند ورودی رایگان

هزینه دریافت هر ایمیل معادل 0.0001 دلار

به منظور ذخیره اطلاعات در منبع داده پایدار، از API استفاده می‌گردد که به ازای تعداد تراکنش‌هایی که تبادل می‌گردد، هزینه پرداخت می‌شود. از آنجایی که بنا به تعداد تبدلات و نوع حافظه پایداری که استفاده می‌گردد، هزینه متغیر است، خواننده محترم برای رویت لیست مذکور به منبع ذکر شده، ارجاع داده می‌شود.

منبع سهمیه	سهمیه پیش فرض رایگان به ازای هر برنامه
مصرف CPU	28 ساعت به ازای هر برنامه در روز

منبع سهمیه	سهمیه پیش فرض رایگان به ازای هر برنامه
منبع ذخیره پایدار داده	1 گیگابایت به ازای هر برنامه در ماه
پهنای باند ورودی	1 گیگابایت به ازای هر برنامه در روز
پهنای باند خروجی	1 گیگابایت به ازای هر برنامه در روز
تراکنش با منبع داده Datastore	50 هزار تراکنش برای خواندن و نوشتن به ازای هر برنامه در ماه
تراکنش با منبع داده Blobstore	5 گیگابایت به ازای هر برنامه در روز
ایمیل دریافتی	100 دریافت به ازای هر برنامه در روز

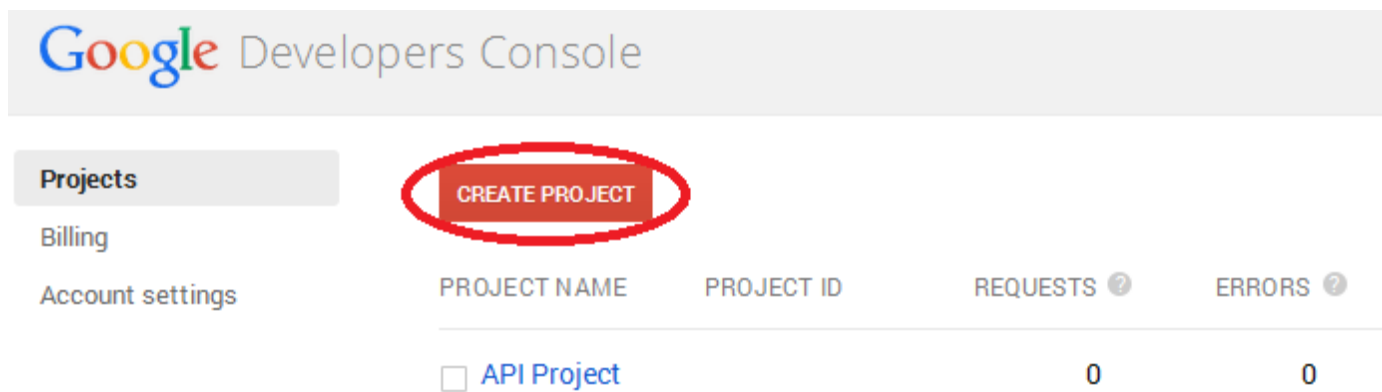
در زمان نگارش این مطلب، آخرین نگارش API مخصوص [Google Analytics](https://www.google.com/analytics/)، نگارش سوم آن است و ... کار کردن با آن دارای مراحل خاصی است که حتما باید رعایت شوند. در غیر اینصورت عملا در یک برنامه‌ی وب یا سرویس ویندوز قابل اجرا نخواهند بود. زیرا در حالت متداول کار با API مخصوص Google Analytics، ابتدا یک صفحه‌ی لاگین به Gmail باز می‌شود که باید به صورت اجباری، مراحل آن را انجام داد تا مشخصات تأیید شده‌ی اکانت در حال استفاده‌ی از API، در پوشه‌ی AppData ویندوز برای استفاده‌های بعدی ذخیره شود. این مورد برای یک برنامه‌ی دسکتاپ معمولی مشکل ساز نیست؛ زیرا کاربر برنامه، به سادگی می‌تواند صفحه‌ی مرورگری را که باز شده‌است، دنبال کرده و به اکانت گوگل خود وارد شود. اما این مراحل را نمی‌توان در یک برنامه‌ی وب یا سرویس ویندوز پیگیری کرد، زیرا عموما امکان لاگین از راه دور به سرور و مدیریت صفحه‌ی لاگین به Gmail وجود ندارد یا بهتر است عنوان شود، بی‌معنا است. برای حل این مشکل، گوگل راه حل دیگری را تحت عنوان اکانت‌های سرویس، ارائه داده است که پس از ایجاد آن، یک فایل X509 Certificate برای اعتبارسنجی سرویس، در اختیار برنامه نویس قرار می‌گیرد تا بدون نیاز به لاگین دستی به Gmail، بتواند از API گوگل استفاده کند. در ادامه نحوه‌ی فعال سازی این قابلیت و استفاده از آن را بررسی خواهیم کرد.

ثبت برنامه‌ی خود در گوگل و انجام تنظیمات آن

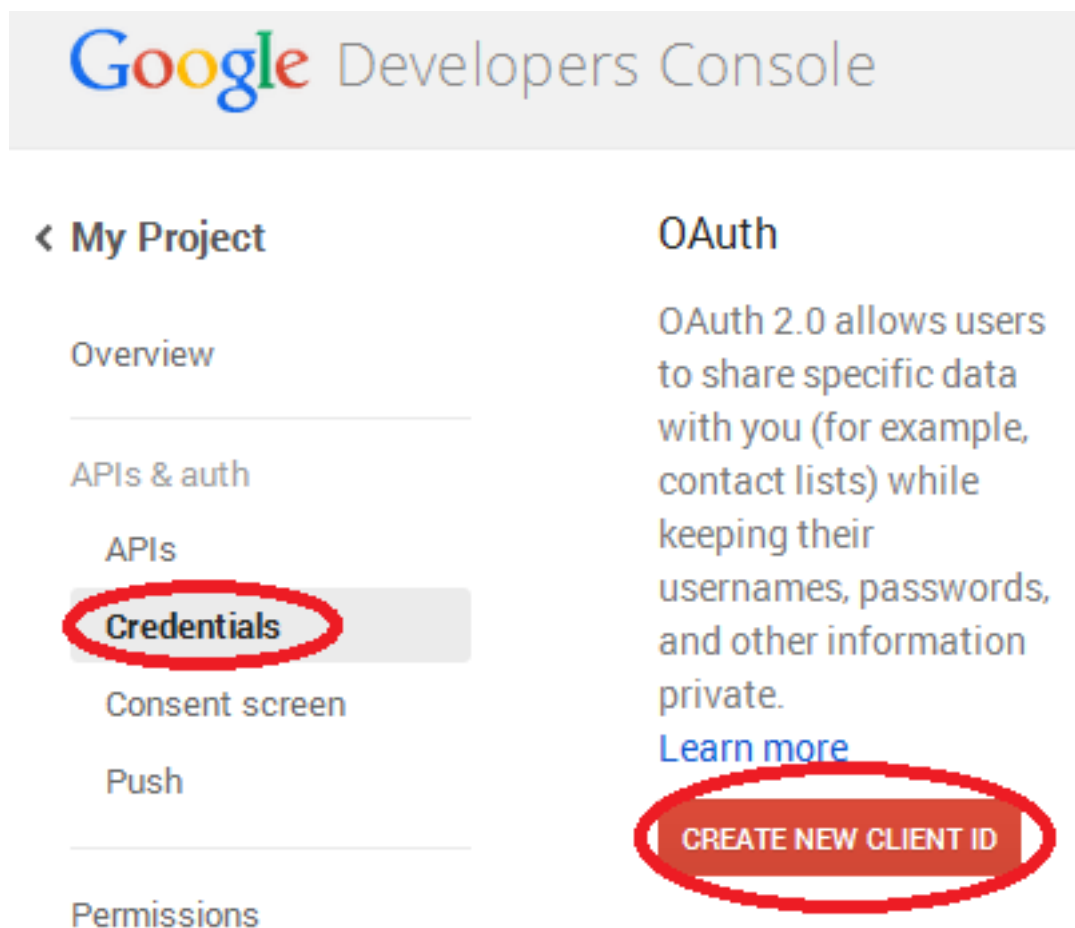
اولین کاری که برای استفاده از نگارش سوم Google Analytics API باید صورت گیرد، ثبت برنامه‌ی خود در Google Developer Console است. برای این منظور ابتدا به آدرس ذیل وارد شوید:

<https://console.developers.google.com>

سپس بر روی دکمه‌ی Create project کلیک کنید. نام دلخواهی را وارد کرده و در ادامه بر روی دکمه‌ی Create کلیک نمایید تا پروفایل این پروژه ایجاد شود.



تنها نکته‌ی مهم این قسمت، بخاطر سپردن نام پروژه است. زیرا از آن جهت اتصال به API گوگل استفاده خواهد شد. پس از ایجاد پروژه، به صفحه‌ی آن وارد شوید و از منوی سمت چپ صفحه، گزینه‌ی Credentials را انتخاب کنید. در ادامه در صفحه‌ی باز شده، بر روی دکمه‌ی Create new client id کلیک نمایید.



در صفحه‌ی باز شده، گزینه‌ی Service account را انتخاب کنید. اگر سایر گزینه‌ها را انتخاب نمائید، کاربری که قرار است از API استفاده کند، باید بتواند توسط مرورگر نصب شده‌ی بر روی کامپیوتر اجرا کننده‌ی برنامه، یکبار به گوگل لاگین نماید که این مورد مطلوب برنامه‌های وب و همچنین سرویس‌ها نیست.

Create Client ID

APPLICATION TYPE

☐ Web application

Accessed by web browsers over a network.

☒ Service account

Calls Google APIs on behalf of your application instead of an end-user. [Learn more](#)

☐ Installed application

Runs on a desktop computer or handheld device (like Android or iPhone).

Create Client ID

Cancel

در اینجا ابتدا یک فایل مجوز p12 را به صورت خودکار دریافت خواهید کرد و همچنین پس از ایجاد client id، نیاز است، ایمیل آنرا جایی یادداشت نمایید:

OAuth

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private.

[Learn more](#)

CREATE NEW CLIENT ID

Compute Engine and App Engine [Learn more](#)

Client ID	29908.apps.googleusercontent.com
Email address	29908@developer.gserviceaccount.com

Download JSON

Service Account

Client ID	1071908-ti1vi55flpnkp3	mq.apps.googleusercontent.com
Email address	1071908-ti1vi55flpnkp3o	nq@developer.gserviceaccount.com
Public key fingerprints	811e1d9bfd8ed	3b976cd516b55

Generate new key

Download JSON

Delete

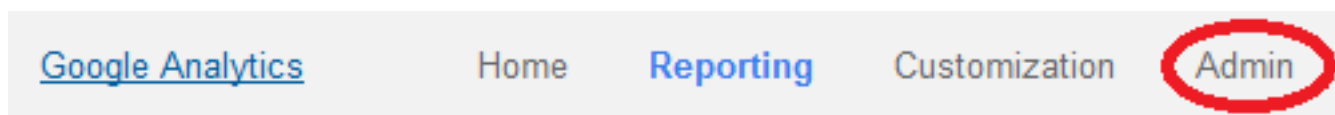
از این ایمیل و همچنین فایل p12 ارائه شده، جهت لاگین به سرور استفاده خواهد شد. همچنین نیاز است تا به برگه‌ی APIs پروژه‌ی ایجاد شده رجوع کرد و گزینه‌ی Analytics API آنرا فعال نمود:

< My Project	NAME	QUOTA	STATUS
Overview	Analytics API	0%	ON
APIs & auth	BigQuery API	0%	ON
APIs	Google Cloud SQL		ON
Credentials			

تا اینجا کار ثبت و فعال سازی برنامه‌ی خود در گوگل به پایان می‌رسد.

دادن دسترسی به Client ID ثبت شده در برنامه‌ی Google Analytics

پس از اینکه Client ID سرویس خود را ثبت کردید، نیاز است به اکانت Google Analytics خود وارد شوید. سپس در منوی آن، گزینه‌ی Admin را پیدا کرده و به آن قسمت، وارد شوید:



در ادامه به گزینه‌ی User management آن وارد شده و به ایمیل Client ID ایجاد شده در قسمت قبل، دسترسی خواندن و آنالیز را اعطاء کنید:

ACCOUNT
www.dotnettips.info

Account Settings

User Management

AdSense Linking

All Filters

Change History

Email	Account Permissions
1. 1075flpvi5cebn g4j3mq@developer.gse rviceaccount.com	Read & Analyze
2.	Manage Users, Edit, Collaborate, Read & Analyze

Add permissions for:

User e-mail that is registered in Google accounts

☐ Notify this user by email

Add Cancel

در صورت عدم رعایت این مساله، کلاینت API، قادر به دسترسی به Google Analytics نخواهد بود.

استفاده از نگارش سوم Google Analytics API در دات نت

قسمت مهم کار، تنظیمات فوق است که در صورت عدم رعایت آن‌ها، شاید نصف روزی را مشغول به دیباگ برنامه شوید. در ادامه نیاز است پیشنیازهای دسترسی به نگارش سوم Google Analytics API را نصب کنیم. برای این منظور، سه بسته‌ی نیوگت ذیل را توسط کنسول پاورشل نیوگت، به برنامه اضافه کنید:

```
PM> Install-Package Google.Apis
PM> Install-Package Google.Apis.auth
PM> Install-Package Google.Apis.Analytics.v3
```

پس از نصب، کلاس GoogleAnalyticsApiV3 زیر، جزئیات دسترسی به Google Analytics API را کیسوله می‌کند:

```
using System;
using System.Linq;
using System.Security.Cryptography.X509Certificates;
using Google.Apis.Analytics.v3;
using Google.Apis.Analytics.v3.Data;
using Google.Apis.Auth.OAuth2;
using Google.Apis.Services;

namespace GoogleAnalyticsApiV3Tests
{
    public class AnalyticsQueryParameters
    {
        public DateTime Start { set; get; }
        public DateTime End { set; get; }
        public string Dimensions { set; get; }
        public string Filters { set; get; }
        public string Metrics { set; get; }
    }

    public class AnalyticsAuthentication
    {
        public Uri SiteUrl { set; get; }
        public string ApplicationName { set; get; }
        public string ServiceAccountEmail { set; get; }
        public string KeyFilePath { set; get; }
        public string KeyFilePassword { set; get; }

        public AnalyticsAuthentication()
        {
            KeyFilePassword = "notasecret";
        }
    }

    public class GoogleAnalyticsApiV3
    {
        public AnalyticsAuthentication Authentication { set; get; }
        public AnalyticsQueryParameters QueryParameters { set; get; }

        public GaData GetData()
        {
            var service = createAnalyticsService();
            var profile = getProfile(service);
            var query = service.Data.Ga.Get("ga:" + profile.Id,
                QueryParameters.Start.ToString("yyyy-MM-dd"),
                QueryParameters.End.ToString("yyyy-MM-dd"),
                QueryParameters.Metrics);
            query.Dimensions = QueryParameters.Dimensions;
            query.Filters = QueryParameters.Filters;
            query.SamplingLevel = DataResource.GaResource.GetRequest.SamplingLevelEnum.HIGHERPRECISION;
            return query.Execute();
        }

        private AnalyticsService createAnalyticsService()
        {
            var certificate = new X509Certificate2(Authentication.KeyFilePath,
```

```

Authentication.KeyFilePassword, X509KeyStorageFlags.Exportable);
    var credential = new ServiceAccountCredential(
        new ServiceAccountCredential.Initializer(Authentication.ServiceAccountEmail)
        {
            Scopes = new[] { AnalyticsService.Scope.AnalyticsReadonly }
        }.FromCertificate(certificate));

    return new AnalyticsService(new BaseClientService.Initializer
    {
        HttpClientInitializer = credential,
        ApplicationName = Authentication.ApplicationName
    });
}

private Profile getProfile(AnalyticsService service)
{
    var accountListRequest = service.Management.Accounts.List();
    var accountList = accountListRequest.Execute();
    var site = Authentication.SiteUrl.Host.ToLowerInvariant();
    var account = accountList.Items.FirstOrDefault(x =>
        x.Name.ToLowerInvariant().Contains(site));
    var webPropertyListRequest = service.Management.Webproperties.List(account.Id);
    var webPropertyList = webPropertyListRequest.Execute();
    var sitePropertyList = webPropertyList.Items.FirstOrDefault(a =>
        a.Name.ToLowerInvariant().Contains(site));
    var profileListRequest = service.Management.Profiles.List(account.Id, sitePropertyList.Id);
    var profileList = profileListRequest.Execute();
    return profileList.Items.FirstOrDefault(a => a.Name.ToLowerInvariant().Contains(site));
}
}
}

```

در اینجا در ابتدا بر اساس فایل p12 ایی که از گوگل دریافت شد، یک X509Certificate2 ایجاد می‌شود. پسورد این فایل مساوی است با ثابت notasecret که در همان زمان تولید اکانت سرویس در گوگل، لحظه‌ای در صفحه نمایش داده خواهد شد. به کمک آن و همچنین ServiceAccountEmail ایمیلی که پیشتر به آن اشاره شد، می‌توان به AnalyticsService لاگین کرد. به این ترتیب به صورت خودکار می‌توان شماره پروفایل اکانت سایت خود را یافت و از آن در حین فراخوانی service.Data.Ga.Get استفاده کرد.

مثالی از نحوه استفاده از کلاس GoogleAnalyticsApiV3

در ادامه یک برنامه‌ی کنسول را ملاحظه می‌کنید که از کلاس GoogleAnalyticsApiV3 استفاده می‌کند:

```

using System;
using System.Collections.Generic;
using System.Linq;

namespace GoogleAnalyticsAPIV3Tests
{
    class Program
    {
        static void Main(string[] args)
        {
            var statistics = new GoogleAnalyticsApiV3
            {
                Authentication = new AnalyticsAuthentication
                {
                    ApplicationName = "My Project",
                    KeyFilePath = "811e1d9976cd516b55-privatekey.p12",
                    ServiceAccountEmail = "10152bng4j3mq@developer.gserviceaccount.com",
                    SiteUrl = new Uri("http://www.dotnettips.info/")
                },
                QueryParameters = new AnalyticsQueryParameters
                {
                    Start = DateTime.Now.AddDays(-7),
                    End = DateTime.Now,
                    Dimensions = "ga:date",
                    Filters = null,
                    Metrics = "ga:users,ga:sessions,ga:pageviews"
                }
            }.GetData();
        }
    }
}

```

```
foreach (var result in statistics.TotalsForAllResults)
{
    Console.WriteLine(result.Key + " -> total:" + result.Value);
}
Console.WriteLine();

foreach (var row in statistics.ColumnHeaders)
{
    Console.Write(row.Name + "\t");
}
Console.WriteLine();

foreach (var row in statistics.Rows)
{
    var rowItems = (List<string>)row;
    Console.WriteLine(rowItems.Aggregate((s1, s2) => s1 + "\t" + s2));
}

Console.ReadLine();
    }
}
```

چند نکته

ApplicationName همان نام پروژه‌ای است که ابتدای کار، در گوگل ایجاد کردیم.
KeyFilePath مسیر فایل مجوز p12 است که گوگل در حین ایجاد اکانت سرویس، در اختیار ما قرار می‌دهد.
ServiceAccountEmail آدرس ایمیل اکانت سرویس است که در قسمت ادمین Google Analytics به آن دسترسی دادیم.
SiteUrl آدرس سایت شما است که هم اکنون در Google Analytics دارای یک اکانت و پروفایل ثبت شده است.
توسط AnalyticsQueryParameters می‌توان نحوه‌ی کوئری گرفتن از Google Analytics را مشخص کرد. تاریخ شروع و پایان گزارش گیری در آن مشخص هستند. در مورد پارامترهایی مانند Dimensions و Metrics بهتر است به مرجع کامل آن در گوگل مراجعه نمائید:

[Dimensions & Metrics Reference](#)

برای نمونه در مثال فوق، تعداد کاربران، سشن‌های آن و همچنین تعداد بار مشاهده‌ی صفحات، گزارش‌گیری می‌شود.

برای مطالعه بیشتر

[Using Google APIs in Windows Store Apps](#)

[How To Use Google Analytics From C# With OAuth](#)

[#Google Analytics's API v3 with C](#)

[NET Library for Accessing and Querying Google Analytics V3 via Service Account.](#)

[#Google OAuth2 C](#)

نظرات خوانندگان

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۰:۲۸ ۱۳۹۳/۰۲/۲۴

سلام و تشکر از مقاله خوبتان.
به نظر شما بهتر است که یک اکانت در آنالیتیکس تعریف کنیم و تمام وبسایت‌های خودمان را به این اکانت اضافه کنیم یا اینکه برای هر وبسایت یک اکانت مجزا بسازیم؟

نویسنده: وحید نصیری
تاریخ: ۱۰:۳۶ ۱۳۹۳/۰۲/۲۴

- برای متد `getProfile` نوشته شده تفاوتی نمی‌کند. چون تمام اکانت‌های موجود را جهت یافتن آدرس سایت مدنظر جستجو می‌کند تا شماره پروفایل آن‌را برای کوئری گرفتن استخراج کند.
+ اکانت `Client ID` ساخته شده، برای تمام سرویس‌های متفاوت گوگل کاربرد دارد؛ و منحصر به `Google Analytics` نیست. فقط باید در قسمت `APIs` آن پروژه، سرویس موردنظر را فعال کرد.
- همچنین مزیت این روش نسبت به پروتکل‌های قدیمی گوگل که از نام کاربری و کلمه‌ی عبور `Gmail` استفاده می‌کردند، عدم نیاز به ذخیره سازی اطلاعات حساس اکانت گوگل خود در فایل کانفیگ برنامه است. بنابراین نیازی نیست تا اکانت‌های متفاوت گوگلی را برای این‌کار خاص ایجاد کنید.

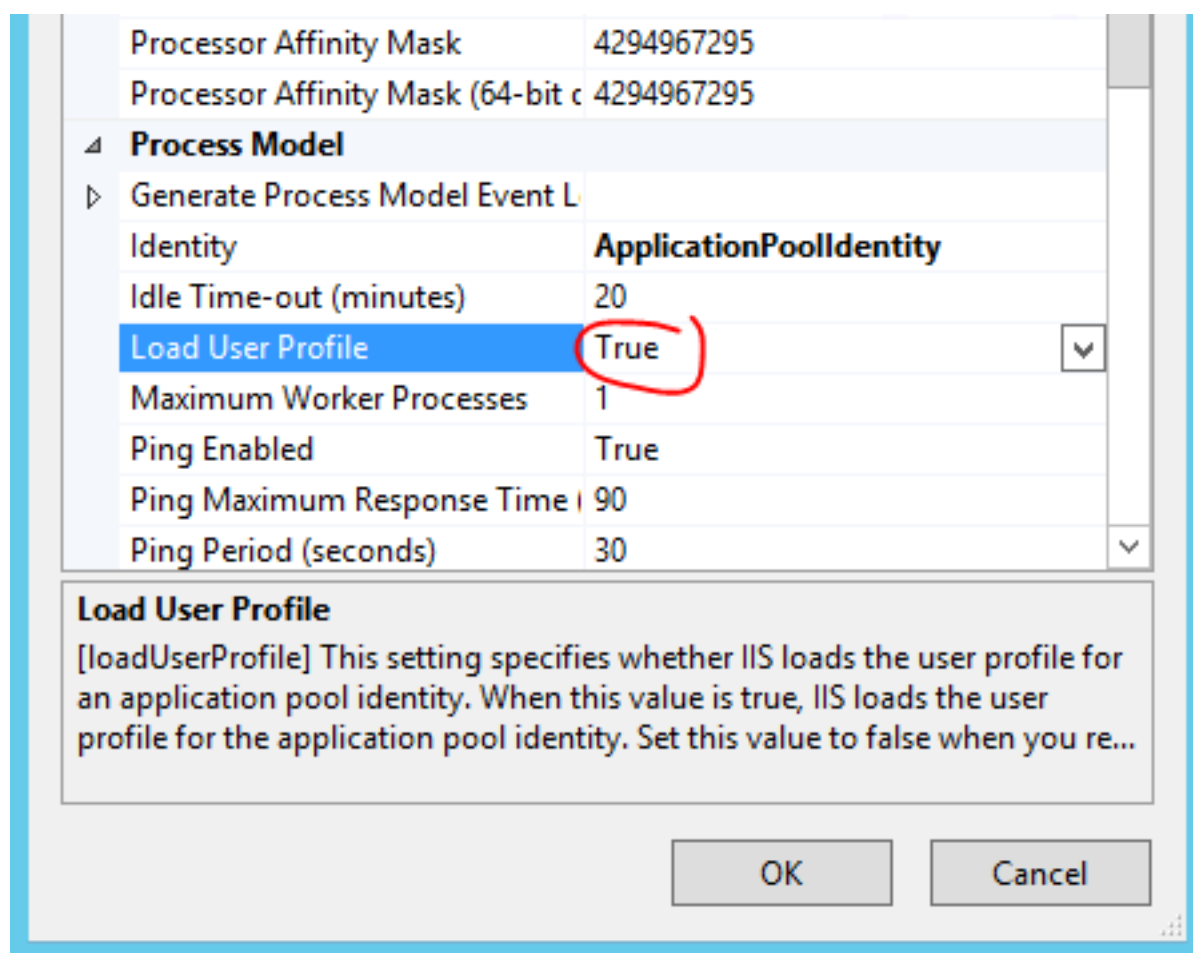
نویسنده: ایلیا اکبری فرد
تاریخ: ۱۲:۳۱ ۱۳۹۳/۰۲/۲۴

با سلام.
همه چیز بر روی کلاینت درست کار میکند ولی وقتی سایت را بر روی سرور آپلود میکنم خطای زیر روی می‌دهد:
`System.Security.Cryptography.CryptographicException: An internal error occurred`

نویسنده: وحید نصیری
تاریخ: ۱۳:۲۷ ۱۳۹۳/۰۲/۲۴

این سه مورد را بررسی کنید:
- آیا مسیر فایل `p12` را درست تنظیم کرده‌اید؟ در یک برنامه‌ی وب باید به این نحو باشد:
`Server.MapPath("~/folder/" + filename)`

- آیا در `application pool` شما `load user profile` فعال است؟ اگر خیر، `crypto subsystem` کار نخواهد کرد:



- تنظیم زیر را هم امتحان کنید:
 بجای

```
var certificate = new X509Certificate2(Authentication.KeyFilePath, Authentication.KeyFilePassword, X509KeyStorageFlags.Exportable);
```

بنویسید:

```
var certificate = new X509Certificate2(Authentication.KeyFilePath, Authentication.KeyFilePassword, X509KeyStorageFlags.MachineKeySet | X509KeyStorageFlags.PersistKeySet | X509KeyStorageFlags.Exportable);
```

نویسنده: ایلیا اکبری فرد
 تاریخ: ۱۴۰۲/۰۲/۲۴

حالت سوم پاسخ شما برای مورد من درست کار کرد. تشکر فراوان.

نویسنده: وحید نصیری
 تاریخ: ۱۵۰۲۸ ۱۳۹۳/۰۲/۲۹

یک نکته‌ی تکمیلی

اگر کتابخانه‌ی Google.Apis.Analytics.v3 را بر روی یک سیستم دات نت 4 اجرا کنید، احتمالاً خطای ذیل را دریافت خواهید کرد:

Could not load type 'System.Net.HttpStatusCode' from assembly System.Net

علت اینجا است که دات نت 4 نیاز به وصله‌ی [KB2468871](#) دارد تا بتواند [portable libraries](#) را بارگذاری کند.

نویسنده: کامران
تاریخ: ۱۳۹۳/۰۵/۲۱ ۲:۵۲

از این ایراد میگیره میگه null هست، من اکانت ساختم پراپرتی هم درست کردم

```
Line 77: var webPropertyListRequest = service.Management.Webproperties.List(account.Id);  
Object reference not set to an instance of an object.
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۲۱ ۱۰:۱۱

«من اکانت ساختم پراپرتی هم درست کردم»
کافی نیست. تمام مراحل باید انجام شوند.
این خطاها صرفا به معنای غیرمعتبر بودن یکی از مراحل است که سبب شده‌اند اطلاعات آن مرحله خاص قابل دریافت نباشد.

نویسنده: کامران
تاریخ: ۱۳۹۳/۰۵/۲۱ ۱۰:۵۳

نه اینکه فقط اکانت و پراپرتی درست کرده باشم، همه مراحل رو انجام دادم اما از خطی که خطا میده متوجه شدم که پراپرتی‌های من رو پیدا نمیکنه یعنی خالی برم‌گردونه، قبلا هم کار کرده بودم همین مشکل پیش میومد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۲۱ ۱۱:۰۹

در متد `getProfile` :
- اگر `account` نال هست یعنی `SiteUrl` مشخص شده در ابتدای برنامه، در [اکانت آنالیتیکس](#) شما موجود نیست. روی سطر `accountList.Items` یک `break-point` قرار دهید تا لیست واقعی اکانت‌های ثبت شده را مشاهده کنید.
- اگر به مرحله‌ی اجرای `accountList` نمی‌رسید، یعنی `ServiceAccountEmail` معتبری وارد نشده‌است.

نویسنده: کامران
تاریخ: ۱۳۹۳/۰۵/۲۱ ۱۲:۳۲

متوجه اشکال شدم.
من کدهای شما رو کپی کردم و نخوندم چه کردید، کدها رو که بررسی کردم متوجه شدم شما توی لیست اکانتها که واکنشی میکنید اکانتی که میخوایم باید همنام یا شبیه آدرس سایت (`SiteUrl`) باشه و طبیعتا من وقتی توی گوگل آنالیز اکانت و پروفایلها رو میسازم نام دلخواه میدم، شما بجای استفاده از آی دی از اسم استفاده کردید که از روی اسم آی دی رو در بیارید برای همین چون اسم اکانت با آدرس سایتی یکی نبود چیزی پیدا نمیکرد :

```
var account = accountList.Items.FirstOrDefault(x => x.Name.ToLowerInvariant().Contains(site));
```

برای پراپرتی هم به همین صورت.