



EBook Gratis

APRENDIZAJE

google-analytics-api

Free unaffiliated eBook created from
Stack Overflow contributors.

#google-
analytics-
api

Tabla de contenido

| | |
|--|-----------|
| Acerca de..... | 1 |
| Capítulo 1: Empezando con google-analytics-api..... | 2 |
| Observaciones..... | 2 |
| Versiones..... | 4 |
| Examples..... | 4 |
| Acceso a las API de Google Analytics..... | 4 |
| Introducción..... | 5 |
| Hello World Reporting API - Resto..... | 7 |
| Capítulo 2: API de informes (Analytics v4)..... | 8 |
| Observaciones..... | 8 |
| Examples..... | 8 |
| Ejemplo de informe único utilizando OAuth2 C #..... | 8 |
| Ejemplo de informe único resto..... | 9 |
| Capítulo 3: API en tiempo real..... | 11 |
| Sintaxis..... | 11 |
| Parámetros..... | 11 |
| Observaciones..... | 12 |
| Examples..... | 13 |
| Ejemplo de PHP..... | 13 |
| Capítulo 4: Autenticación..... | 16 |
| Sintaxis..... | 16 |
| Parámetros..... | 16 |
| Observaciones..... | 16 |
| Examples..... | 18 |
| OAuth2 C #..... | 18 |
| Autenticación de cuentas de servicio Vb.net..... | 19 |
| Capítulo 5: Metadata api..... | 21 |
| Sintaxis..... | 21 |
| Parámetros..... | 21 |
| Observaciones..... | 21 |

| | |
|---|-----------|
| Examples..... | 21 |
| Ejemplo de descanso..... | 21 |
| Ejemplo de Java..... | 22 |
| Ejemplo de PHP..... | 23 |
| Ejemplo de Python..... | 24 |
| Ejemplo de C #..... | 26 |
| Capítulo 6: Respuestas de error..... | 27 |
| Sintaxis..... | 27 |
| Parámetros..... | 27 |
| Observaciones..... | 28 |
| Examples..... | 28 |
| 400 invalidParameter..... | 28 |
| Creditos..... | 30 |

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [google-analytics-api](#)

It is an unofficial and free google-analytics-api ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official google-analytics-api.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con google-analytics-api

Observaciones

Las [API de Google Analytics](#) le permiten acceder a los datos dentro de Google Analytics. No debe confundirse con el [protocolo de medición](#) que se utiliza para insertar datos en [Google Analytics](#).

La API de Google Analytics se divide en partes servales.

API de informes de Google Analytics

1. [Google Analytics Core Reporting API v3](#) le brinda acceso a la mayoría de los datos de informes en Google Analytics. Con el Core Reporting API puede:
 - Cree paneles personalizados para mostrar los datos de Google Analytics.
 - Ahorre tiempo al automatizar tareas complejas de informes.
 - Integre sus datos de Google Analytics con otras aplicaciones empresariales.
2. [Google Analytics Reporting API V4](#) es el método programático más avanzado para acceder a los datos de informes en Google Analytics. Con la API de informes de Google Analytics, puede:
 - Cree paneles personalizados para mostrar los datos de Google Analytics.
 - Automatice tareas complejas de informes para ahorrar tiempo.
 - Integre sus datos de Google Analytics con otras aplicaciones empresariales.
2. La [API de informes en tiempo real](#) le permite solicitar datos en tiempo real, por ejemplo, actividad en tiempo real en una vista, para un usuario autenticado. Puede utilizar la API de informes en tiempo real para:
 - Muestra a los espectadores activos de una página y crea un sentido de urgencia para los usuarios que miran un artículo con un inventario finito.
 - Muestra el contenido más popular, como las 10 mejores páginas activas.
 - Crea y visualiza un panel de control en tiempo real.

La API de informes en tiempo real, en versión beta limitada, está disponible solo para la vista previa del desarrollador. [Regístrese](#) para acceder a la API.

3. La [API de informes de embudos multicanal](#) le permite solicitar datos de embudos multicanal para un usuario autenticado. Los datos se derivan de los datos de la ruta de conversión, que muestran las interacciones del usuario con varias fuentes de tráfico en varias sesiones antes de la conversión. Esto le permite analizar cómo múltiples canales de marketing influyen en las conversiones a lo largo del tiempo. Para obtener más detalles sobre qué datos están disponibles, lea [Acerca de los embudos multicanal](#), así como [Acerca de los datos de embudos multicanal](#). Con la API de informes de embudos multicanal puede:

- Cree informes personalizados utilizando datos de embudos multicanal. Por ejemplo, podría usar los datos de las rutas de conversión principales para informar sobre atributos como la posición relativa de las interacciones en una ruta de conversión.
- Integre los datos de embudos multicanal con los datos de su empresa. Por ejemplo, podría correlacionar los datos de conversión en línea con los datos de ventas fuera de línea y los costos de medios para llegar a una imagen más completa del ROI de marketing.
- Mostrar embudos multicanal en nuevos entornos. Por ejemplo, podría crear visualizaciones y otras presentaciones de los datos que comuniquen el valor de los diferentes canales de marketing para impulsar las conversiones.

Ayudantes

1. La [API de metadatos](#) devuelve la lista y los atributos de las columnas (es decir, dimensiones y métricas) expuestas en las API de informes de Google Analytics. Los atributos devueltos incluyen el nombre de la interfaz de usuario, la descripción, el soporte de los segmentos y más. Puede utilizar la API de metadatos para:

- Descubrir automáticamente nuevas columnas.
- Acceda a todos los atributos de dimensiones y métricas para las API de informes de Google Analytics.

Nota: Esto solo devuelve metadatos para la API de informes centrales y la API de informes. No metadatos en tiempo real.

2. [Google Analytics Embed API](#) es una biblioteca de JavaScript que le permite crear e incrustar fácilmente un panel en un sitio web de terceros en cuestión de minutos. Le brinda un conjunto de componentes conectables que pueden trabajar juntos para construir herramientas complejas, lo que lo hace simple y poderoso al mismo tiempo.

API de configuración

1. La [API de administración de análisis](#) permite el acceso programático a los datos de configuración de Google Analytics. Puede crear aplicaciones para administrar de manera más eficiente cuentas analíticas grandes o complejas. Las grandes empresas con muchas propiedades pueden automatizar la configuración de la cuenta. Incluso si está creando una aplicación de informes, la API de administración le proporciona herramientas para navegar por su cuenta. Puede utilizar la API de administración de Google Analytics para:

- Enumere toda la información de cuenta, propiedad y vista de un usuario.
- Administrar propiedades, vistas y objetivos.
- Administrar permisos de usuario para una jerarquía de cuentas.
- Recupere una ID de vista para usar con la API de informes centrales.
- Determine qué objetivos están activos y acceda a sus nombres configurados.
- Gestione los vínculos entre las propiedades de Analytics y las cuentas de AdWords.
- Gestionar las audiencias de remarketing.

Las operaciones de escritura en la API de administración (por ejemplo, crear,

actualizar, eliminar, parchear) para los recursos de propiedad web, vista (perfil) y objetivo están actualmente disponibles como una vista previa del desarrollador en **versión beta limitada** . Si está interesado en utilizar estas funciones, [solicite acceso a la versión beta](#) .

2. La [API de Provisioning](#) se puede utilizar para crear nuevas cuentas de Google Analytics y habilitar Google Analytics para sus clientes a escala. Está destinado a proveedores de servicios calificados y grandes socios. Por ejemplo, podría usar la API de aprovisionamiento como parte de un nuevo proceso de incorporación de usuarios para crear una nueva cuenta de Google Analytics para un cliente y luego usar recursos adicionales de la API de administración para configurar la cuenta mediante programación y vincularla a AdWords. Todo esto puede automatizarse e iniciarse desde su propia administración o interfaz de informes.

La API de aprovisionamiento está disponible solo por invitación. Actualmente no estamos aceptando nuevos proyectos.

Versiones

Actualmente hay tres versiones de la API de Google Analytics en vivo.

La versión 2.4 de [Google Analytics V2](#) del Core Reporting API es una API solo para XML que es en su mayoría compatible con la versión 2.3. **(Legado)**

[Google Analytics V3](#), que incluye la API de informes centrales, la API de administración, la API de metadatos y varias otras API. Estas APIs devuelven Json

[Google Analytics V4](#), que actualmente solo incluye la API de informes y se considera la forma más avanzada de recuperar datos de informes de Google Analytics.

Examples

Acceso a las API de Google Analytics

Técnicamente, puede acceder a las API de Google Analytics utilizando cualquier lenguaje de programación que pueda manejar una [publicación HTTP](#) o una solicitud [HTTP Get](#) .

Dicho esto, Google también ha creado una serie de bibliotecas estándar de clientes oficiales para ayudarte con esto. Usar una biblioteca cliente estándar para el lenguaje de programación elegido puede ser mucho más fácil que codificarlo desde cero.

Las bibliotecas de clientes OFICIALES compatibles con la API de Google Analytics:

1. Biblioteca API de Google APIs - [GitHub](#)
2. Biblioteca API de .NET de Google - [GitHub NuGet](#)
3. Biblioteca API de Google APIs Python - [GitHub](#)
4. Biblioteca API Java de Google APIs - [enlace](#)

5. API Objective-C de Google APIs - [GitHub](#)

Hay más bibliotecas [aquí](#) .

Introducción

Las [API de Google Analytics](#) le permiten acceder a los datos dentro de Google Analytics. No debe confundirse con el [protocolo de medición](#) que se utiliza para insertar datos en [Google Analytics](#) .

La API de Google Analytics se divide en partes servales.

API de informes de Google Analytics

1. [Google Analytics Core Reporting API v3](#) le brinda acceso a la mayoría de los datos de informes en Google Analytics. Con el Core Reporting API puede:
 - Cree paneles personalizados para mostrar los datos de Google Analytics.
 - Ahorre tiempo al automatizar tareas complejas de informes.
 - Integre sus datos de Google Analytics con otras aplicaciones empresariales.
2. [Google Analytics Reporting API V4](#) es el método programático más avanzado para acceder a los datos de informes en Google Analytics. Con la API de informes de Google Analytics, puede:
 - Cree paneles personalizados para mostrar los datos de Google Analytics.
 - Automatice tareas complejas de informes para ahorrar tiempo.
 - Integre sus datos de Google Analytics con otras aplicaciones empresariales.
2. La [API de informes en tiempo real](#) le permite solicitar datos en tiempo real, por ejemplo, actividad en tiempo real en una vista, para un usuario autenticado. Puede utilizar la API de informes en tiempo real para:
 - Muestra a los espectadores activos de una página y crea un sentido de urgencia para los usuarios que miran un artículo con un inventario finito.
 - Muestra el contenido más popular, como las 10 mejores páginas activas.
 - Crea y visualiza un panel de control en tiempo real.

La API de informes en tiempo real, en versión beta limitada, está disponible solo para la vista previa del desarrollador. [Regístrese](#) para acceder a la API.

3. La [API de informes de embudos multicanal](#) le permite solicitar datos de embudos multicanal para un usuario autenticado. Los datos se derivan de los datos de la ruta de conversión, que muestran las interacciones del usuario con varias fuentes de tráfico en varias sesiones antes de la conversión. Esto le permite analizar cómo múltiples canales de marketing influyen en las conversiones a lo largo del tiempo. Para obtener más detalles sobre qué datos están disponibles, lea [Acerca de los embudos multicanal](#), así como [Acerca de los datos de embudos multicanal](#). Con la API de informes de embudos multicanal puede:
 - Cree informes personalizados utilizando datos de embudos multicanal. Por ejemplo,

podría usar los datos de las rutas de conversión principales para informar sobre atributos como la posición relativa de las interacciones en una ruta de conversión.

- Integre los datos de embudos multicanal con los datos de su empresa. Por ejemplo, podría correlacionar los datos de conversión en línea con los datos de ventas fuera de línea y los costos de medios para llegar a una imagen más completa del ROI de marketing.
- Mostrar embudos multicanal en nuevos entornos. Por ejemplo, podría crear visualizaciones y otras presentaciones de los datos que comuniquen el valor de los diferentes canales de marketing para impulsar las conversiones.

Ayudantes

1. La [API de metadatos](#) devuelve la lista y los atributos de las columnas (es decir, dimensiones y métricas) expuestas en las API de informes de Google Analytics. Los atributos devueltos incluyen el nombre de la interfaz de usuario, la descripción, el soporte de los segmentos y más. Puede utilizar la API de metadatos para:

- Descubrir automáticamente nuevas columnas.
- Acceda a todos los atributos de dimensiones y métricas para las API de informes de Google Analytics.

Nota: Esto solo devuelve metadatos para la API de informes centrales y la API de informes. No metadatos en tiempo real.

2. [Google Analytics Embed API](#) es una biblioteca de JavaScript que le permite crear e incrustar fácilmente un panel en un sitio web de terceros en cuestión de minutos. Le brinda un conjunto de componentes conectables que pueden trabajar juntos para construir herramientas complejas, lo que lo hace simple y poderoso al mismo tiempo.

API de configuración

1. La [API de administración de análisis](#) permite el acceso programático a los datos de configuración de Google Analytics. Puede crear aplicaciones para administrar de manera más eficiente cuentas analíticas grandes o complejas. Las grandes empresas con muchas propiedades pueden automatizar la configuración de la cuenta. Incluso si está creando una aplicación de informes, la API de administración le proporciona herramientas para navegar por su cuenta. Puede utilizar la API de administración de Google Analytics para:

- Enumere toda la información de cuenta, propiedad y vista de un usuario.
- Administrar propiedades, vistas y objetivos.
- Administrar permisos de usuario para una jerarquía de cuentas.
- Recupere una ID de vista para usar con la API de informes centrales.
- Determine qué objetivos están activos y acceda a sus nombres configurados.
- Gestione los vínculos entre las propiedades de Analytics y las cuentas de AdWords.
- Gestionar las audiencias de remarketing.

Las operaciones de escritura en la API de administración (por ejemplo, crear, actualizar, eliminar, parchear) para los recursos de propiedad web, vista (perfil) y

objetivo están actualmente disponibles como una vista previa del desarrollador en **versión beta limitada** . Si está interesado en utilizar estas funciones, [solicite acceso a la versión beta](#) .

2. La [API de Provisioning](#) se puede utilizar para crear nuevas cuentas de Google Analytics y habilitar Google Analytics para sus clientes a escala. Está destinado a proveedores de servicios calificados y grandes socios. Por ejemplo, podría usar la API de aprovisionamiento como parte de un nuevo proceso de incorporación de usuarios para crear una nueva cuenta de Google Analytics para un cliente y luego usar recursos adicionales de la API de administración para configurar la cuenta mediante programación y vincularla a AdWords. Todo esto puede automatizarse e iniciarse desde su propia administración o interfaz de informes.

La API de aprovisionamiento está disponible solo por invitación. Actualmente no estamos aceptando nuevos proyectos.

Hello World Reporting API - Resto

```
POST https://analyticsreporting.googleapis.com/v4/reports:batchGet?access_token={Access token from auth request}
{
  "reportRequests": [
    {
      "viewId": "XXXX",
      "dateRanges": [
        {
          "startDate": "2015-06-15",
          "endDate": "2015-06-30"
        }
      ],
      "metrics": [
        {
          "expression": "ga:sessions"
        }
      ],
      "dimensions": [
        {
          "name": "ga:browser"
        }
      ]
    }
  ]
}
```

Lea [Empezando con google-analytics-api en línea](https://riptutorial.com/es/google-analytics-api/topic/7253/empezando-con-google-analytics-api): <https://riptutorial.com/es/google-analytics-api/topic/7253/empezando-con-google-analytics-api>

Capítulo 2: API de informes (Analytics v4)

Observaciones

[Google Analytics Reporting API V4](#) es el método programático más avanzado para acceder a los datos de informes en [Google Analytics](#) . Con la API de informes de Google Analytics, puede:

- Cree paneles personalizados para mostrar los datos de [Google Analytics](#) .
- Automatice tareas complejas de informes para ahorrar tiempo.
- Integre sus datos de [Google Analytics](#) con otras aplicaciones empresariales.

Características

[Google Analytics](#) se basa en una potente infraestructura de informes de datos. Google Analytics Reporting API V4 le brinda acceso a la potencia de la plataforma [Google Analytics](#) . La API proporciona estas características clave:

- Expresiones métricas: la API le permite solicitar no solo métricas incorporadas sino también una combinación de métricas expresadas en operaciones matemáticas. Por ejemplo, puede usar la expresión `ga: goal1completions / ga: session` para solicitar las finalizaciones del objetivo por número de sesiones.
- Múltiples rangos de fechas: la API le permite en una sola solicitud obtener datos en dos rangos de fechas.
- Valor de Cohortes y Vida útil: La API tiene un amplio vocabulario para solicitar informes de valores de Cohorte y Vida útil.
- Múltiples segmentos: la API le permite obtener múltiples segmentos en una sola solicitud.

Examples

Ejemplo de informe único utilizando OAuth2 C

Este ejemplo utiliza la biblioteca oficial de Google .net Client.

```
PM> Install-Package Google.Apis.AnalyticsReporting.v4
```

Autorización requiere uno de los siguientes ámbitos de OAuth:

- <https://www.googleapis.com/auth/analytics.readonly>
- <https://www.googleapis.com/auth/analytics>

OAuth2

```
// These are the scopes of permissions you need.
string[] scopes = new string[] { AnalyticsReportingService.Scope.AnalyticsReadonly }; //
View your Google Analytics Data

UserCredential credential;
```

```

using (var stream = new FileStream(clientSecretJson, FileMode.Open, FileAccess.Read))
{
    string credPath =
System.Environment.GetFolderPath(System.Environment.SpecialFolder.Personal);
    credPath = Path.Combine(credPath, ".credentials/",
System.Reflection.Assembly.GetExecutingAssembly().GetName().Name);

    // Requesting Authentication or loading previously stored authentication for userName
    credential =
GoogleWebAuthorizationBroker.AuthorizeAsync(GoogleClientSecrets.Load(stream).Secrets,
    scopes,
    userName,
    CancellationToken.None,
    new FileDataStore(credPath, true)).Result;
}

// Create Reporting API service.
var service = new AnalyticsReportingService(new BaseClientService.Initializer()
{
    HttpClientInitializer = credential,
    ApplicationName = string.Format("{0} Authentication",
System.Reflection.Assembly.GetExecutingAssembly().GetName().Name),
});

```

Solicitud de informes

```

// Create the DateRange object.
DateRange June2015 = new DateRange() { StartDate = "2015-01-01", EndDate = "2015-06-30" };
DateRange June2016 = new DateRange() { StartDate = "2016-01-01", EndDate = "2016-06-30" };
List<DateRange> dateRanges = new List<DateRange>() { June2016, June2015 };

// Create the ReportRequest object.
// This should have a large number of rows
ReportRequest reportRequest = new ReportRequest
{
    ViewId = ConfigurationManager.AppSettings["GoogleAnalyticsViewId"],
    DateRanges = dateRanges,
    Dimensions = new List<Dimension>() { new Dimension() { Name = "ga:date" }, new Dimension()
{ Name = "ga:usertype" } },
    Metrics = new List<Metric>() { new Metric() { Expression= "ga:users" }, new Metric() {
Expression = "ga:sessions" } },
    PageSize = 1000,
};

List<ReportRequest> requests = new List<ReportRequest>();
requests.Add(reportRequest);

var getReport = new GetReportsRequest() { ReportRequests = requests };
var response = service.Reports.BatchGet(getReport).Execute();

```

Ejemplo de informe único resto

Las solicitudes de API son HTTP POST con el token de acceso adjunto al final del punto final de la API.

Autorización requiere uno de los siguientes ámbitos de OAuth:

- <https://www.googleapis.com/auth/analytics.readonly>
- <https://www.googleapis.com/auth/analytics>

Tenga en cuenta que al publicar los datos use `ContentType = "application/Json";`

```
https://analyticsreporting.googleapis.com/v4/reports:batchGet?Access_token={from auth}
{
  "reportRequests": [
    {
      "viewId": "XXXX",
      "dateRanges": [
        {
          "startDate": "2015-06-15",
          "endDate": "2015-06-30"
        }
      ],
      "metrics": [
        {
          "expression": "ga:sessions"
        }
      ],
      "dimensions": [
        {
          "name": "ga:browser"
        }
      ]
    }
  ]
}
```

Lea API de informes (Analytics v4) en línea: <https://riptutorial.com/es/google-analytics-api/topic/7265/api-de-informes--analytics-v4->

Capítulo 3: API en tiempo real

Sintaxis

- GET [https://www.googleapis.com/analytics/v3/data/realtime?ids=\[Analytics view Id\] & metrics = \[Métricas en tiempo real\] & access_token = \[Access_token_from_Authentication\]](https://www.googleapis.com/analytics/v3/data/realtime?ids=[Analytics view Id] & metrics = [Métricas en tiempo real] & access_token = [Access_token_from_Authentication])
- GET [https://www.googleapis.com/analytics/v3/data/realtime?ids=\[Analytics view Id\] & metrics = \[Métricas en tiempo real\] & dimensiones = \[Métricas en tiempo real\] & access_token = \[Access_token_from_Authentication\]](https://www.googleapis.com/analytics/v3/data/realtime?ids=[Analytics view Id] & metrics = [Métricas en tiempo real] & dimensiones = [Métricas en tiempo real] & access_token = [Access_token_from_Authentication])

Parámetros

| (Requerido) Nombre del parámetro | Descripción |
|--|--|
| ids | ID de tabla única para recuperar datos de Analytics. El ID de la tabla tiene el formato ga: XXXX, donde XXXX es el ID de la vista (perfil) de Analytics. |
| métrica | Una lista separada por comas de las métricas de Analytics. Por ejemplo, 'rt: activeUsers'. Se debe especificar al menos una métrica. |
| (Opcional) Nombre del parámetro | Descripción |
| dimensiones | Una lista separada por comas de dimensiones en tiempo real. Por ejemplo, 'rt: medium, rt: city'. |
| filtros | Una lista separada por comas de filtros de dimensiones o métricas que se aplicarán a los datos en tiempo real. |
| max-resultados | El número máximo de entradas para incluir en este feed. |
| ordenar | Una lista separada por comas de dimensiones o métricas que determinan el orden de clasificación de los datos en tiempo real. |
| (Estándar) Nombre del parámetro | Descripción |
| llamar de vuelta | Nombre de la función de devolución de llamada de JavaScript que maneja la respuesta. Utilizado en las solicitudes JavaScript JSON-P. |
| bonitaImprimir | Devuelve la respuesta en un formato legible por humanos si es |

| (Requerido) Nombre del parámetro | Descripción |
|-------------------------------------|---|
| | verdadero. Valor por defecto: verdadero. Cuando esto es falso, puede reducir el tamaño de la carga útil de respuesta, lo que podría conducir a un mejor rendimiento en algunos entornos. |
| quotaUsuario | Le permite imponer cuotas por usuario desde una aplicación del lado del servidor, incluso en los casos en que la dirección IP del usuario es desconocida. Esto puede ocurrir, por ejemplo, con aplicaciones que ejecutan trabajos cron en App Engine en nombre de un usuario. Puede elegir cualquier cadena arbitraria que identifique de forma única a un usuario, pero está limitada a 40 caracteres. Anula el uso de userIp si se proporcionan ambos. Obtenga más información sobre el uso de límites. |
| userIp | Le permite imponer cuotas por usuario al llamar a la API desde una aplicación del lado del servidor. Obtenga más información sobre el uso de límites. |

Observaciones

La API de informes en tiempo real, en versión beta limitada, está disponible solo para la vista previa del desarrollador. [Regístrese](#) para acceder a la API.

La [API de informes en tiempo real](#) le permite solicitar datos en tiempo real, por ejemplo, actividad en tiempo real en su propiedad, para un usuario autenticado.

Puede utilizar la API de informes en tiempo real para:

- Muestra a los espectadores activos de una página y crea un sentido de urgencia para los usuarios que miran un artículo con un inventario finito.
- Muestra el contenido más popular, como las 10 mejores páginas activas.
- Crea y visualiza un panel de control en tiempo real.

Autorización

Las llamadas a la API en tiempo real de Google Analytics requieren autorización con al menos uno de los siguientes ámbitos (lea más acerca de la autenticación y la autorización).

| Alcance | acceso permitido |
|---|--|
| https://www.googleapis.com/auth/analytics | Acceso completo a las cuentas de Google Analytics hasta el nivel de acceso de los usuarios autenticados. |
| https://www.googleapis.com/auth/analytics.readonly | Acceso de solo lectura a las cuentas de Google Analytics de usuarios |

| Alcance | acceso permitido |
|---------|------------------|
| | autenticados |

Examples

Ejemplo de PHP

Utiliza la [biblioteca de cliente PHP](#)

```
/**
 * 1.Create and Execute a Real Time Report
 * An application can request real-time data by calling the get method on the Analytics
 service object.
 * The method requires an ids parameter which specifies from which view (profile) to retrieve
 data.
 * For example, the following code requests real-time data for view (profile) ID 56789.
 */
$optParams = array(
    'dimensions' => 'rt:medium');

try {
    $results = $analytics->data_realtime->get(
        'ga:56789',
        'rt:activeUsers',
        $optParams);
    // Success.
} catch (apiServiceException $e) {
    // Handle API service exceptions.
    $error = $e->getMessage();
}

/**
 * 2. Print out the Real-Time Data
 * The components of the report can be printed out as follows:
 */

function printRealtimeReport($results) {
    printReportInfo($results);
    printQueryInfo($results);
    printProfileInfo($results);
    printColumnHeaders($results);
    printDataTable($results);
    printTotalsForAllResults($results);
}

function printDataTable(&$results) {
    if (count($results->getRows()) > 0) {
        $table .= '<table>';

        // Print headers.
        $table .= '<tr>';

        foreach ($results->getColumnHeaders() as $header) {
            $table .= '<th>' . $header->name . '</th>';
        }
        $table .= '</tr>';
    }
}
```



```

// Print table rows.
foreach ($results->getRows() as $row) {
    $stable .= '<tr>';
    foreach ($row as $cell) {
        $stable .= '<td>'
            . htmlspecialchars($cell, ENT_NOQUOTES)
            . '</td>';
    }
    $stable .= '</tr>';
}
$stable .= '</table>';

} else {
    $stable .= '<p>No Results Found.</p>';
}
print $stable;
}

function printColumnHeaders(&$results) {
    $html = '';
    $headers = $results->getColumnHeaders();

    foreach ($headers as $header) {
        $html .= <<<HTML
<pre>
Column Name      = {$header->getName()}
Column Type      = {$header->getColumnType()}
Column Data Type = {$header->getDataType()}
</pre>
HTML;
    }
    print $html;
}

function printQueryInfo(&$results) {
    $query = $results->getQuery();
    $html = <<<HTML
<pre>
Ids              = {$query->getIds()}
Metrics          = {$query->getMetrics()}
Dimensions       = {$query->getDimensions()}
Sort             = {$query->getSort()}
Filters          = {$query->getFilters()}
Max Results     = {$query->getMax_results()}
</pre>
HTML;

    print $html;
}

function printProfileInfo(&$results) {
    $profileInfo = $results->getProfileInfo();

    $html = <<<HTML
<pre>
Account ID          = {$profileInfo->getAccountId()}
Web Property ID     = {$profileInfo->getWebPropertyId()}
Internal Web Property ID = {$profileInfo->getInternalWebPropertyId()}
Profile ID          = {$profileInfo->getProfileId()}
Profile Name        = {$profileInfo->getProfileName()}

```

```

Table ID                = {$profileInfo->getTableId()}
</pre>
HTML;

    print $html;
}

function printReportInfo(&$results) {
    $html = <<<HTML
    <pre>
Kind                    = {$results->getKind()}
ID                     = {$results->getId()}
Self Link              = {$results->getSelfLink()}
Total Results         = {$results->getTotalResults()}
    </pre>
HTML;

    print $html;
}

function printTotalsForAllResults(&$results) {
    $totals = $results->getTotalsForAllResults();

    foreach ($totals as $metricName => $metricTotal) {
        $html .= "Metric Name = $metricName\n";
        $html .= "Metric Total = $metricTotal";
    }

    print $html;
}

```

Versión original copiada de [documentación oficial](#).

Lea API en tiempo real en línea: <https://riptutorial.com/es/google-analytics-api/topic/7305/api-en-tiempo-real>

Capítulo 4: Autenticación

Sintaxis

- GET [https://accounts.google.com/o/oauth2/auth?client_id= {clientid} & redirect_uri = urn:ietf:wg:oauth: 2.0: oob & scope = {Scopes} & response_type = code](https://accounts.google.com/o/oauth2/auth?client_id={clientid} & redirect_uri = urn:ietf:wg:oauth:2.0:oob & scope = {Scopes} & response_type = code)
- POST <https://accounts.google.com/o/oauth2/token> code = {Código de la llamada anterior} & client_id = {ClientId} & client_secret = {ClientSecret} & redirect_uri = urn:ietf:wg:og: outh & grant_type = permission_code
- POST <https://accounts.google.com/o/oauth2/token> client_id = {ClientId} & client_secret = {ClientSecret} & refresh_token = {De la llamada anterior} & grant_type = refresh_token

Parámetros

| Parámetro | Descripción |
|----------------------------|---|
| Identificación del cliente | Desde la consola de desarrolladores de Google identifica tu proyecto o aplicación. |
| secreto | Desde la consola de desarrolladores de Google identifica tu proyecto o aplicación. |
| redirect_uri | Desde la ubicación de Google Developer donde se debe devolver la autenticación. En el caso de aplicaciones nativas, <code>urn:ietf:wg:oauth:2.0:oob</code> se puede utilizar para denotar localhost |

Observaciones

Para acceder a cualquier API de Google, debe identificarse como desarrollador e identificar su proyecto. Lo hacemos creando un nuevo proyecto en la [consola de Google Developers](#) .

Cuando cree su proyecto, si desea acceder a las API de Google Analytics, debe habilitar las API a las que desea acceder.

- **API de informes** : acceso a la API de informes de Google Analytics v4.
- **API de análisis** : acceso a todo lo demás.

Ahora debe decidir cómo desea acceder a los datos.

Con Google Data hay dos tipos de datos públicos y privados.

- Los datos públicos no son propiedad de un usuario. La API de metadatos es una API pública en la que no es necesario iniciar sesión para acceder a esos datos.
- La API de informes contiene una información de Google Analytics de los usuarios que es privada, no puede verla a menos que el usuario le haya dado permiso para acceder a ella.

Si solo está accediendo a datos públicos, todo lo que necesita hacer es crear una clave de API pública y podrá acceder a la API en cuestión. Si va a acceder a los datos privados del usuario, deberá crear las credenciales de OAuth2 o las credenciales de la cuenta de servicio.

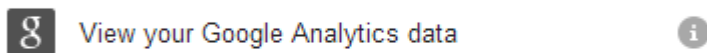
Autorización OAuth2

Para acceder a los datos privados del usuario, debemos tener el permiso del propietario de los datos para acceder a ellos. [OAuth2](#) nos permite solicitar ese acceso a un usuario.

Probablemente has visto antes a OAuth2 antes.

Google Analytics Windows ▾

This app would like to:



Google Analytics Windows and Google will use this information in accordance with their respective terms of service and privacy policies.

Cancel

Accept

La aplicación "Google Analytics Windows" solicita acceso para ver los usuarios de "Google Analytics Data"

1. Las ventanas de Google Analytics son el nombre del proyecto que se creó en la consola de desarrolladores de Google.
2. Los datos de Google Analytics son el alcance de los permisos que solicitamos.

Ámbito Necesitamos decirle al usuario lo que pretendemos hacer. La API de Google Analytics tiene dos ámbitos que puede usar.

1. <https://www.googleapis.com/auth/analytics.readonly>
2. <https://www.googleapis.com/auth/analytics>

Es mejor solicitar solo los alcances que necesita. Si solo va a leer los datos de un usuario, solo necesita solicitar el alcance de solo lectura.

Cuentas de servicio de autorizaciones.

[Las cuentas de servicio](#) son diferentes en que son aprobadas previamente. Si crea credenciales de cuenta de servicio, usted, como desarrollador, puede tomar el correo electrónico de la cuenta de servicio y agregarlo como usuario en su cuenta de Google Analytics. **En el nivel de cuenta**, esto le otorgará acceso a los datos a la cuenta de servicio. No necesitará abrir la ventana de autenticación y solicitar acceso. La cuenta de servicio tendrá acceso a los datos mientras sea un usuario en la cuenta de Google Analytics.

Conclusión

Se necesita autenticación para acceder a la mayoría de los datos expuestos por la API de Google Analytics.

No puede usar el inicio de sesión / inicio de sesión del cliente y la contraseña para acceder a cualquier API de Google a partir de mayo de 2015. Debe usar la autenticación abierta.

**

Examples

Oauth2 C

Ejemplo utiliza la [biblioteca del cliente Dotnet de las API de Google](#) .

PM> Install-Package Google.Apis.AnalyticsReporting.v4

```
/// <summary>
/// This method requests Authentcation from a user using Oauth2.
/// Credentials are stored in System.Environment.SpecialFolder.Personal
/// Documentation https://developers.google.com/accounts/docs/OAuth2
/// </summary>
/// <param name="clientSecretJson">Path to the client secret json file from Google
Developers console.</param>
/// <param name="userName">Identifying string for the user who is being
authenticated.</param>
/// <returns>DriveService used to make requests against the Drive API</returns>
public static AnalyticsReportingService AuthenticateOauth(string clientSecretJson, string
userName)
{
    try
    {
        if (string.IsNullOrEmpty(userName))
            throw new Exception("userName is required.");
        if (string.IsNullOrEmpty(clientSecretJson))
            throw new Exception("clientSecretJson is required.");
        if (!File.Exists(clientSecretJson))
            throw new Exception("clientSecretJson file does not exist.");

        // These are the scopes of permissions you need.
        string[] scopes = new string[] { AnalyticsReportingService.Scope.AnalyticsReadOnly
}; // View your Google Analytics Data

        UserCredential credential;
        using (var stream = new FileStream(clientSecretJson, FileMode.Open,
FileAccess.Read))
        {
            string credPath =
System.Environment.GetFolderPath(System.Environment.SpecialFolder.Personal);
            credPath = Path.Combine(credPath, ".credentials/",
System.Reflection.Assembly.GetExecutingAssembly().GetName().Name);

            // Requesting Authentication or loading previously stored authentication for
userName
```

```

        credential =
GoogleWebAuthorizationBroker.AuthorizeAsync(GoogleClientSecrets.Load(stream).Secrets,
                                            scopes,
                                            userName,

CancellationToken.None,
                                            new
FileDataStore(credPath, true)).Result;
    }

    // Create Drive API service.
    return new AnalyticsReportingService(new BaseClientService.Initializer()
    {
        HttpClientInitializer = credential,
        ApplicationName = string.Format("{0} Authentication",
System.Reflection.Assembly.GetExecutingAssembly().GetName().Name),
    });
    }
    catch (Exception ex)
    {
        Console.WriteLine("Create Oauth2 DriveService failed" + ex.Message);
        throw new Exception("CreateOauth2DriveFailed", ex);
    }
}

```

Autenticación de cuentas de servicio Vb.net

Ejemplos de uso de [Install-Package Google.Apis.AnalyticsReporting.v4](#)

```

Public Shared Function getServiceInitializer() As BaseClientService
    Dim serviceAccountCredentialFilePath = "Path to Json service account key file"
REM from Google Developers console
    Dim myKeyEMail = "XXX@developer.gserviceaccount.com" REM from Google Developers console
    Dim scope =
Google.Apis.AnalyticsReporting.v4.AnalyticsReportingService.Scope.AnalyticsReadOnly

    Try

        Dim credential
        Using stream As New FileStream(serviceAccountCredentialFilePath, FileMode.Open,
FileAccess.Read)

            credential = GoogleCredential.FromStream(stream).CreateScoped(scope)

        End Using

        Dim initializer As New BaseClientService.Initializer()
        initializer.HttpClientInitializer = credential
        initializer.ApplicationName = "SRJCGMail"

        Dim service As New AnalyticsReportingService(initializer)
        Return service

    Catch ex As Exception
        Console.WriteLine(ex.Message)
        Return Nothing
    End Try

```

End Function

Lea Autenticación en línea: <https://riptutorial.com/es/google-analytics-api/topic/7268/autenticacion>

Capítulo 5: Metadata api

Sintaxis

- HTTP GET [https://www.googleapis.com/analytics/v3/metadata/ {reportType} / columns? Key = {APIKey}](https://www.googleapis.com/analytics/v3/metadata/{reportType}/columns?Key={APIKey})
- HTTP GET [https://www.googleapis.com/analytics/v3/metadata/ {reportType} / columns? Access_token = {Access_token}](https://www.googleapis.com/analytics/v3/metadata/{reportType}/columns?Access_token={Access_token})

Parámetros

| Nombre del parametro | Descripción |
|----------------------|---|
| tipo de informe | Tipo de informe. Valores permitidos: ga . Donde ga corresponde a la API de informes centrales . |

Observaciones

La [API de metadatos](#) devuelve la lista y los atributos de las columnas (es decir, dimensiones y métricas) expuestas en las [API de informes de Google Analytics](#) (v2, v3 y v4). Los atributos devueltos incluyen el nombre de la interfaz de usuario, la descripción, el soporte de los segmentos y más.

Puede utilizar la API de metadatos para:

- Descubrir automáticamente nuevas columnas.
- Acceda a todos los atributos de dimensiones y métricas para las API de informes de Google Analytics

Esta es la misma lista que se encuentra en el [Explorador de dimensiones y métricas](#) .

Nota: Las dimensiones y métricas de los embudos en tiempo real y multicanal no están disponibles actualmente.

Examples

Ejemplo de descanso

Las llamadas a la [API de metadatos](#) son con HTTP Get:

Usando la clave de API pública

OBTENER <https://www.googleapis.com/analytics/v3/metadata/ga/columns?key=>

{YOUR_API_KEY}

Uso del token de acceso de OAuth2 o autenticación de cuenta de servicio

OBTENER

https://www.googleapis.com/analytics/v3/metadata/ga/columns?access_token={Authenticated_Access-Token}

Ejemplo de Java

utiliza la [biblioteca de Java Client](#)

```
/**
 * 1. Execute a Metadata Request
 * An application can request columns data by calling the list method on the Analytics service
 object.
 * The method requires an reportType parameter that specifies the column data to retrieve.
 * For example, the following code requests columns for the ga report type.
 */

try {
    Columns results = getMetadata(analytics);
    // Success

} catch (GoogleJsonResponseException e) {
    // Catch API specific errors.
    handleApiError(e);
} catch (IOException e) {
    // Catch general parsing network errors.
    e.printStackTrace();
}

/**
 * 2. Print out the Columns data
 * The components of the result can be printed out as follows:
 */

private static Columns getMetadata(Analytics analytics) throws IOException {
    String reportType = "ga";
    return analytics.metadata()
        .columns()
        .list(reportType)
        .execute();
}

private static void printMetadataReport(Columns results) {
    System.out.println("Metadata Response Report");
    printReportInfo(results);
    printAttributes(results.getAttributeNames());
    printColumns(results.getItems());
}

private static void printReportInfo(Columns results) {
    System.out.println("## Metadata Report Info ##");
    System.out.println("Kind: " + results.getKind());
    System.out.println("Etag: " + results.getEtag());
}
```

```

System.out.println("Total Results: " + results.getTotalResults());
System.out.println();
}

private static void printAttributes(List<String> attributeNames) {
    System.out.println("## Attribute Names ##");
    for (String attribute : attributeNames) {
        System.out.println(attribute);
    }
}

private static void printColumns(List<Column> columns) {
    System.out.println("## Columns ##");

    for (Column column : columns) {
        System.out.println();
        System.out.println("Column ID: " + column.getId());
        System.out.println("Kind: " + column.getKind());

        Map<String, String> columnAttributes = column.getAttributes();

        for (Map.Entry<String, String> attribute: columnAttributes.entrySet()) {
            System.out.println(attribute.getKey() + ": " + attribute.getValue());
        }
    }
}
}

```

Nota: primera versión copiada de [Metadata.list](#)

Ejemplo de PHP

Utiliza la [biblioteca de cliente PHP](#)

```

/**
 * 1. Execute a Metadata Request
 * An application can request columns data by calling the list method on the Analytics service
 object.
 * The method requires an reportType parameter that specifies the column data to retrieve.
 * For example, the following code requests columns for the ga report type.
 */

try {

    $results = $analytics->metadata_columns->listMetadataColumns('ga');
    // Success

} catch (apiServiceException $e) {
    // Handle API service exceptions.
    $error = $e->getMessage();
}

/**
 * 2. Print out the Columns data
 * The components of the result can be printed out as follows:
 */

```

```

function printMetadataReport($results) {
    print '<h1>Metadata Report</h1>';
    printReportInfo($results);
    printAttributes($results);
    printColumns($results);
}

function printReportInfo(&$results) {
    $html = '<h2>Report Info</h2>';
    $html .= <<<HTML
<pre>
Kind                = {$results->getKind()}
Etag                = {$results->getEtag()}
Total Results      = {$results->getTotalResults()}
</pre>
HTML;
    print $html;
}

function printAttributes(&$results) {
    $html = '<h2>Attribute Names</h2><ul>';
    $attributes = $results->getAttributeNames();
    foreach ($attributes as $attribute) {
        $html .= '<li>'. $attribute . '</li>';
    }
    $html .= '</ul>';
    print $html;
}

function printColumns(&$results) {
    $columns = $results->getItems();
    if (count($columns) > 0) {
        $html = '<h2>Columns</h2>';
        foreach ($columns as $column) {
            $html .= '<h3>' . $column->getId() . '</h3>';
            $column_attributes = $column->getAttributes();
            foreach ($column_attributes as $name=>$value) {
                $html .= <<<HTML
<pre>
{$name}: {$value}
</pre>
HTML;
            }
        }
    } else {
        $html = '<p>No Results Found.</p>';
    }
    print $html;
}

```

Nota: versión original copiada de [metadata.list](#)

Ejemplo de Python

Utiliza la [biblioteca del cliente de Python](#)

```

# 1. Execute a Metadata Request
# An application can request columns data by calling the list method on the Analytics service
object.
# The method requires an reportType parameter that specifies the column data to retrieve.
# For example, the following code requests columns for the ga report type.

try:
    results = service.metadata().columns().list(reportType='ga').execute()

except TypeError, error:
    # Handle errors in constructing a query.
    print ('There was an error in constructing your query : %s' % error)

except HttpError, error:
    # Handle API errors.
    print ('Arg, there was an API error : %s : %s' %
           (error.resp.status, error._get_reason()))

# 2. Print out the Columns data
# The components of the result can be printed out as follows:

def print_metadata_report(results):
    print 'Metadata Response Report'
    print_report_info(results)
    print_attributes(results.get('attributeNames'))
    print_columns(results)

def print_report_info(columns):
    print "Metadata Report Info"
    if columns:
        print 'Kind           = %s' % columns.get('kind')
        print 'Etag           = %s' % columns.get('etag')
        print 'Total Results = %s' % columns.get('totalResults')

def print_attributes(attributes):
    if attributes:
        print 'Attribute Names:'
        for attribute in attributes:
            print attribute

def print_columns(columns_data):
    if columns_data:
        print 'Columns:'

        columns = columns_data.get('items', [])

        for column in columns:
            print
            print '%15s = %35s' % ('Column ID', column.get('id'))
            print '%15s = %35s' % ('Kind', column.get('kind'))

            column_attributes = column.get('attributes', [])

            for name, value in column_attributes.iteritems():
                print '%15s = %35s' % (name, value)

```

Nota: versión original copiada de [metadata.list](#)

Ejemplo de C

Utiliza la [biblioteca del cliente .Net](#)

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

```
var metadataService = new AnalyticsMetaDataService(new BaseClientService.Initializer()
    {
        ApiKey = {Public API KEY},
        ApplicationName = "Metadata api",
    });

var result = Service.Metadata.Columns.List("ga").Execute();
```

Lea Metadata api en línea: <https://riptutorial.com/es/google-analytics-api/topic/7297/metadata-api>

Capítulo 6: Respuestas de error

Sintaxis

- 400 invalidParameter Indica que un parámetro de solicitud tiene un valor no válido.
- 400 badRequest Indica que la consulta no fue válida. Por ejemplo, faltaba la identificación de los padres o la combinación de dimensiones o métricas solicitadas no era válida.
- 403 insuficientes Permisos Indica que el usuario no tiene permisos suficientes para la entidad especificada en la consulta.
- 403 dailyLimitExceeded Indica que el usuario ha superado la cuota diaria (por proyecto o por vista (perfil)).
- 403 useLimits.userRateLimitExceededUnreg Indica que la aplicación debe registrarse en la Consola de API de Google.
- 403 userRateLimitExceeded Indica que se ha excedido el límite de velocidad de usuario. El límite de velocidad máxima es de 10 qps por dirección IP.
- 403 rateLimitExceeded Indica que se han excedido los límites globales o globales de la tasa del proyecto. Reintentar usando retroceso exponencial. Debe reducir la velocidad a la que está enviando las solicitudes.
- 403 quotaExceeded Indica que se han alcanzado las 10 solicitudes simultáneas por vista (perfil) en la API de informes centrales.
- 429 RESOURCE_EXHAUSTED AnalyticsDefaultGroupCLIENT_PROJECT-1d Indica que las solicitudes por día por cuota de proyecto se han agotado.
- 429 RESOURCE_EXHAUSTED AnalyticsDefaultGroupCLIENT_PROJECT-100s Indica que las solicitudes por 100 segundos por cuota de proyecto se han agotado.
- 429 RESOURCE_EXHAUSTED AnalyticsDefaultGroupUSER-100s Indica que las solicitudes por 100 segundos por usuario por cuota de proyecto se han agotado.
- 429 RESOURCE_EXHAUSTED DiscoveryGroupCLIENT_PROJECT-100s Indica que las solicitudes de descubrimiento por cada 100 segundos de cuota se han agotado.
- 500 internalServerError Se produjo un error de servidor interno inesperado. No vuelva a intentar esta consulta más de una vez.
- 503 backendError Server devolvió un error. No vuelva a intentar esta consulta más de una vez.

Parámetros

| Nombre | Descripción |
|-------------------|--|
| dominio | Ubicación del error ex: global |
| razón | Motivo del error. |
| mensaje | Un mensaje explicando el error y una posible solución. |
| Tipo de ubicación | El tipo de ubicación del error ex: paramater |
| ubicación | La ubicación real del error. |

Observaciones

Si una solicitud es exitosa, la API devuelve un código de estado HTTP 200 junto con los datos solicitados en el cuerpo de la respuesta.

Si se produce un error con una solicitud, la API devuelve un código de estado HTTP y el motivo de la respuesta según el tipo de error. Además, el cuerpo de la respuesta contiene una descripción detallada de lo que causó el error.

Examples

400 invalidParameter

Es importante leer la respuesta de error que devuelve el servidor de API de Google Analytics. En muchos casos le pueden decir exactamente lo que está mal.

```
400 invalidParameter

{
  "error": {
    "errors": [
      {
        "domain": "global",
        "reason": "invalidParameter",
        "message": "Invalid value '-1' for max-results. Value must be within the range: [1, 1000]",
        "locationType": "parameter",
        "location": "max-results"
      }
    ],
    "code": 400,
    "message": "Invalid value '-1' for max-results. Value must be within the range: [1, 1000]"
  }
}
```

En este caso el mensaje:

"mensaje": "Valor no válido '-1' para resultados máximos. El valor debe estar dentro del rango: [1, 1000]",

Nos dice que enviamos un valor de -1 para el parámetro max-results que no es válido, solo podemos enviar un valor de 1-1000.

Lea Respuestas de error en línea: <https://riptutorial.com/es/google-analytics-api/topic/7376/respuestas-de-error>

Creditos

| S. No | Capítulos | Contributors |
|-------|------------------------------------|---|
| 1 | Empezando con google-analytics-api | Community , DalmTo , K48 , MarkeD |
| 2 | API de informes (Analytics v4) | DalmTo |
| 3 | API en tiempo real | DalmTo |
| 4 | Autenticación | DalmTo , Matt |
| 5 | Metadata api | DalmTo |
| 6 | Respuestas de error | DalmTo |