

Google Ads API

快速上手指南

| | |
|-----------------------|----|
| 前言 | 1 |
| 使用Google Ads API的前提条件 | 1 |
| 获取Developer Token | 1 |
| OAuth认证密钥获取 | 3 |
| Refresh Token | 9 |
| 开始你的第一个API | 12 |

前言

本篇文档将会教你如何利用Google Ads API来获取你的相关广告的Campaigns信息及细节。为了实现这个功能，首先你需要利用OAuth2去激活Google Ads API的访问权限。之后，你需要通过使用配置文件的方式，适当的配置账号的具体信息和认证细节，从而能够利用Google Ads API进行相关的API操作。

使用Google Ads API的前提条件

为了使用Google Ads API的相关功能，首先你需要做如下的配置，使得你可以获取调用API的权限。配置完成后，你可以根据实际需求，进行API的调用了。

获取Developer Token

Developer Token用于赋予你的操作对象访问Google Ads API的权限。这个权限不与你的账号绑定，也就是说，当你通过下文提到方式申请到了需要使用的Developer Token之后，理论上你可以利用该Token对任何一个属于你的Google Ads的账号进行API访问，即使该账号没有与你申请的Token绑定，你也可以用这个Token对这个账号进行API操作。例如，你有两个经理账号使用相同的Google账号进行管理，那你为其中一个经理账号申请的Developer Token可以用来操作属于

相同Google账号下的另一个经理账号。我们会在后面的[OAuth章节](#)讨论如何限制Token对账号的访问权限。

提示 : Developer Token分成两种类型, Approved(basic) Token可以对生产环境的Google Ads账号进行API操作。Pending(test) Token只能对Google Ads测试账号进行API操作。申请Developer Token的时候, 请留意账号及Token的类型。

为了获取一个有效的Developer Token, 首先, 你需要登陆到Google Ads经理账号下进行Token的申请操作。



登陆Google Ads经理账号后, 在控制台左上角点选<[工具与设置](#)>, 之后在<[设置](#)>菜单中, 选择<[API中心](#)>。

API 联系人的电子邮件地址

公司名称

公司网址

公司类型

预期用途

主要营业地点

条款及条件

6. Brand Features. "Brand Features" means the trade names, trademarks, service marks, logos, domain names, and other distinctive brand features of each party, and in the case of Google includes the Brand Features of Google affiliates. Except where expressly stated, these terms do not grant either party any right, title, or interest in or to the other party's Brand Features. All use by you of Google's Brand Features (including any goodwill associated therewith) will inure to the benefit of Google or its affiliates. You must only use the Google Brand Features in accordance with (a) these terms and (b) the then current Google Brand Features Use Guidelines, the current version of which is at <http://www.google.com/permissions/guidelines.html>.

7. Data Protection. (A) To the extent that you access, use or otherwise process (i) any information that directly or indirectly identifies a natural person; or (ii) information that is not specifically about an identifiable individual but, when combined with other information, may directly or indirectly identify a natural person ("Personal Information") made available by Google, you will:

a. comply with all privacy, data security, and data protection laws, directives, regulations, and rules in any jurisdiction applicable to you ("Privacy Laws");

b. use or otherwise access Personal Information only for purposes which are consistent with the consent obtained from the individual to whom the Personal Information relates or as expressly permitted in these terms by Google; and

是, 我同意上述条款及条件

创建令牌

进入<API中心>之后, 请根据实际情况填写API的访问权限和相关信息, 填写完成后, 点击<创建令牌>, 页面会跳转到创建完成的界面, 此时, 你可以看到新生成的Token的具体信息和当前Token的访问级别。

API 访问权限

目前已为您启用开发者令牌, 提供的是测试帐号访问权限。在“访问权限级别”部分申请基本访问权限, 即可开始与正式帐号互动。

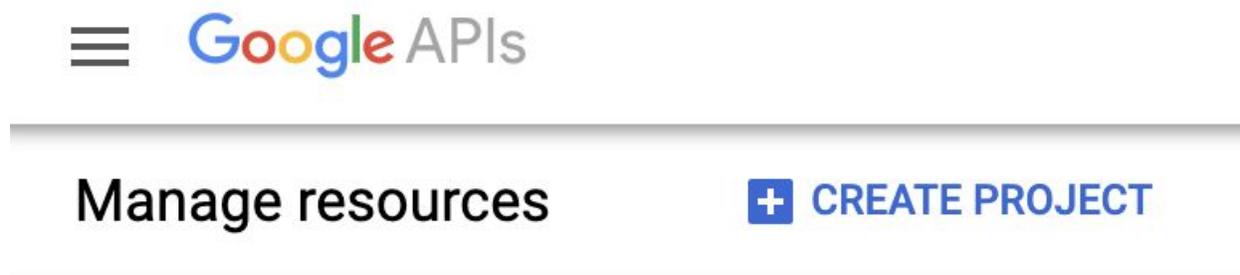
| | | |
|--------|--|-----------------------------------|
| 开发者令牌 | <div style="background-color: black; width: 100px; height: 15px;"></div> | 重置令牌 |
| 访问权限级别 | 测试帐号 | 申请基本访问权限 <input type="checkbox"/> |

提示: 所有新创建的Developer Token均为测试(test) Token, 该Token只能对Google Ads测试账号进行API操作。如果你想利用该Developer Token对生产环境的账号进行API操作, 请点击<申请基本访问权限>, 按照实际情况填写表格, 后台工作人员会进行审核, 审核完成后, 会将测试账号升级成基本账号。

OAuth认证密钥获取

当成功获取了Developer Token之后, 该Token理论上可以对任何一个Google Ads账号进行API操作。为了进行权限控制, 你需要利用OAuth的方式配置API操作权限。

首先，你需要先激活Google Ads API，点击[Google API Console](#)进入API管理控制台。



点击<CREATE PROJECT>，创建一个新的API项目。

New Project

Project name *
Google Ads API OAuth ?

Project ID: sinuous-branch-253307. It cannot be changed later. [EDIT](#)

Organization *
google.com ▼ ?

Select an organization to attach it to a project. This selection can't be changed later.

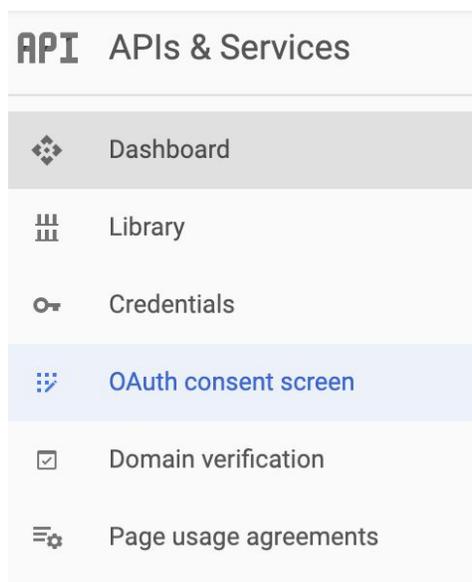
Location *
google-default BROWSE

Parent organization or folder

Create Googleplex internal App Engine project ?

[CREATE](#) [CANCEL](#)

根据实际情况，配置当前API项目的相关信息。并点击<CREATE>。

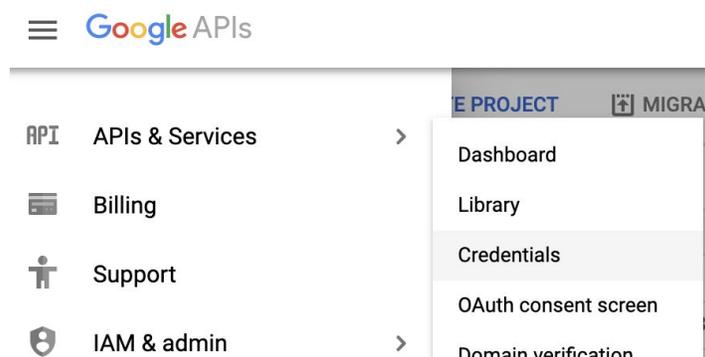


创建完成新的API项目之后，选择该项目下的<APIs & Services>，之后选择进入<OAuth consent screen>菜单下，创建新的Consent。

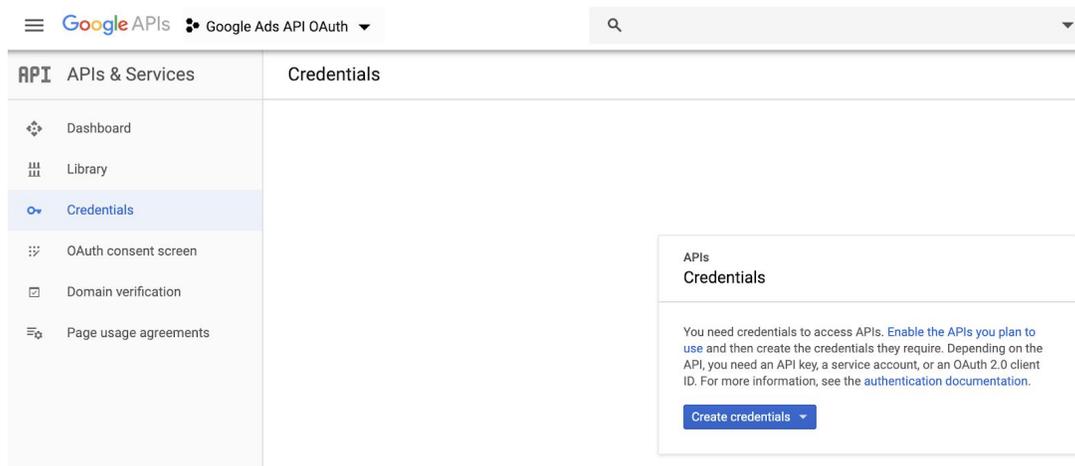
Application name [?]
The name of the app asking for consent

Application logo [?]
An image on the consent screen that will help users recognize your app

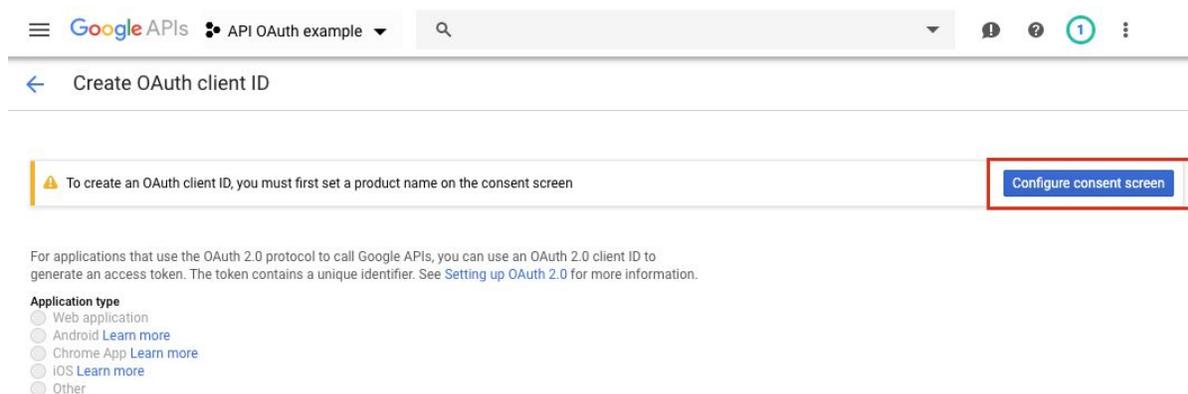
Consent name的部分，只需要修改<Application Name>的部分即可，其他需要配置的内容我们可以维持默认值。



完成Consent的配置之后，返回上层菜单，依次选择<APIs & Services>，之后选择<Credentials>。



进入Credentials界面后，选择<Create credentials>。



提示：如果你在创建OAuth Credentials之前，忘记创建Consent Name，点击<Create Credentials>之后，会提示你需要先创建Consent才可以继续创建OAuth密钥。在这里可以直接点击<Configure consent screen>跳转到Consent的配置部分。请参考上文进行Consent的配置。

API key

Identifies your project using a simple API key to check quota and access

OAuth client ID

Requests user consent so your app can access the user's data

Service account key

Enables server-to-server, app-level authentication using robot accounts

Help me choose

Asks a few questions to help you decide which type of credential to use

在credentials的种类中，选择<OAuth client ID>。

For applications that use the OAuth 2.0 protocol to call Google APIs, you can use an OAuth 2.0 client ID to generate an access token. The token contains a unique identifier. See [Setting up OAuth 2.0](#) for more information.

Application type

- Web application
- Android [Learn more](#)
- Chrome App [Learn more](#)
- iOS [Learn more](#)
- Other

Name 

Google Ads API

Create

Cancel

<Application type>请选择<Web application>或者<Other(Installed Application)>这两种类型中的一种。关于两种类型具体的区别，可以参考这个[链接](#)。<Name>部分，可以自由输入容易记忆的名称即可。点击<Create>，即可生成一个权限的OAuth信息。本文档示例中，我们使用<Other>做为<Application type>。

OAuth client

The client ID and secret can always be accessed from Credentials in APIs & Services

 OAuth is limited to 100 [sensitive scope logins](#) until the [OAuth consent screen](#) is published. This may require a verification process that can take several days.

Here is your client ID

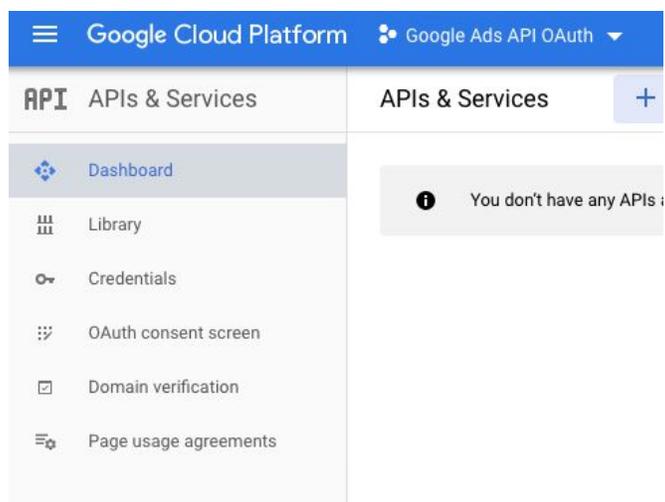
[Redacted client ID] 

Here is your client secret

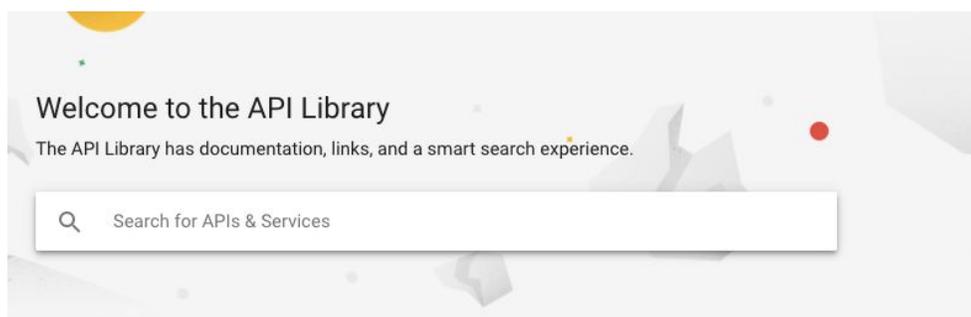
[Redacted client secret] 

记录下来新创建的OAuth的<client ID>和<client secret>的信息，稍后的配置中，我们会用到这两个信息。

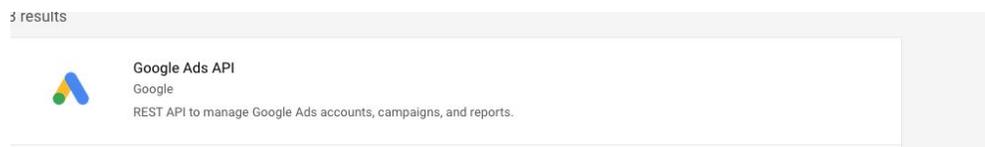
当前的配置，为你生成了对应的OAuth的密钥信息，但是，并没有让你的OAuth可以调用Google Ads API的功能。为了激活这个功能，我们需要在当前账号下，激活Google Ads API。



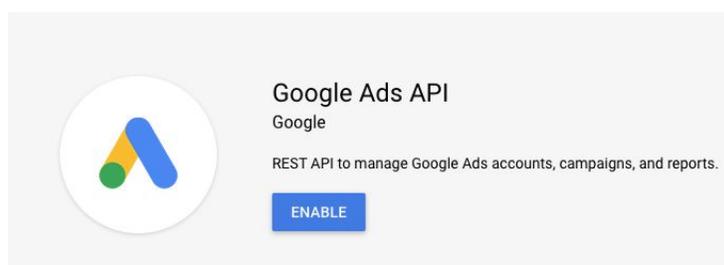
在当前的<APIs & Services>界面，选择<Library>。



在搜索框里，输入<Google Ads API>。



点选<Google Ads API>，进入这个API的管理界面。



在Google Ads API管理界面里，点击<[ENABLE](#)>，完成之后，你之前创建的OAuth密钥信息，就可以用来进行Google Ads API的相关操作了。

Refresh Token

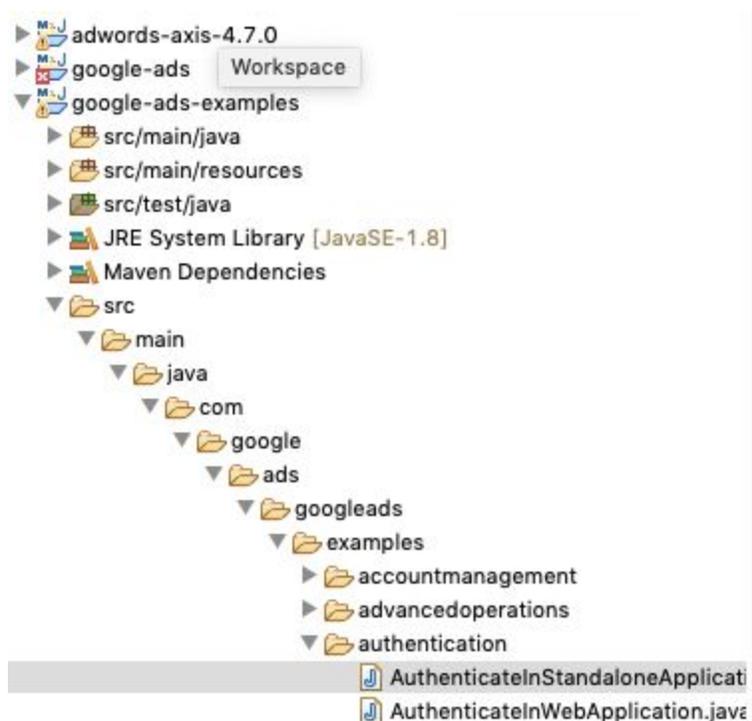
当成功创建了OAuth的相关密钥之后，我们需要为这一对密钥创建Refresh Token(更新令牌)，该Token在之后会嵌入到我们的API调用中，从而完成正确的认证授权操作。

为了生成正确的Refresh Token，我们需要调用OAuth Client Library来进行操作，根据你习惯使用的编程语言的不同，请选择对应的OAuth Client Library来执行这个操作。同时，根据你在OAuth配置界面选择的OAuth类型(Web或者其他)，请选择对应的Library获取Refresh Token。当前支持的所有Library可以点击这个[链接](#)获取。

本文档中，我们使用Java作为语言，因为我们OAuth密钥的类型为<[Other](#)>，所以本范例选择<[Installed Application flow](#)>作为生成Refresh Token的Library示范。

首先，让我们先从Github下载Java版本的[Google Ads API Client Library](#)。下载完成之后，选择你熟悉的IDE打开这个Java项目。本文档使用eclipse做为IDE示范。

提示：当前Java Library使用Maven，请保证你的IDE已经预装Maven对应的插件，否则可能会有报错。



在刚才加载的Java项目中，按上图找到<AuthenticateInStandaloneApplication.java>这个执行函数。我们可以利用这个函数生成Refresh Token。

```
// The console will be used when running this example in some
// set the clientId and clientSecret in the lines below.
clientId = "INSERT_CLIENT_ID_HERE";
clientSecret = "INSERT_CLIENT_SECRET_HERE";
// Optional: If your application knows which user is trying to
// to the user's email address so that the Google Authenticat
```

找到这个函数中的<clientId>和<clientSecret>这两个变量，替换成我们刚刚在OAuth密钥环节生成的<clientId>和<clientSecret>。保存并执行这个函数。

```
Paste this url in your browser:
https://accounts.google.com/o/oauth2
Type the code you received here:
```

执行后，你会在console输出窗口，看到类似于上图的一段话，将输出的URL复制黏贴到任何一个网页浏览器窗口。该操作要求使用Google Ads的访问权限，请确保使用的网页浏览器登录了Google Ads的管理员账号。

提示：具体调用的函数与你刚才创建的OAuth Credentials的类型相关。如果你创建的是<Other>，请使用<AuthenticateInStandaloneApplication.java>，如果你创建的是<Web Application>，请使用<AuthenticateInWebApplication.java>。



URL会跳转到新的网页认证界面，在这个界面里，直接点击<允许>。完成认证授权过程。

提示：当前的账号授权绑定，所需要访问的账号，是你Google Ads用于生成Developer Token的经理账号，请确认当前的浏览器使用了正确的的Google账号登陆。例如，你用于生成Developer Token的Google Ads的经理账号是abc@google.com，请保证这里关联的账号，也是abc@google.com。如果API开发人员没有Google Ads的经理账号，请联系经理账号拥有者，由拥有者在网页完成授权，再把下图产生的授权码提供给API开发人员。



请复制以下代码，切换到您的应用，然后粘贴此代码：



授权结束后，页面会跳转至上图所示的界面中。将当前页面中出现的代码，复制粘贴回刚才的Java函数的console端口。

```
Paste this url in your browser:
https://accounts.google.com/o/oauth2
Type the code you received here:
4/rQEozg1kF5ibpfbF4K5sCVghWysMCY
Your refresh token is: 1//0e4AUc
```

这时，console界面就会生成一个Refresh Token。这个Token与你的OAuth相关信息绑定，请妥善保管这个新生成的Token，我们稍后的配置会用到。至此，Google Ads API所需要使用的Refresh Token也生成完毕。

开始你的第一个API

当你按照上面的步骤完成相关操作后，你就获得调用Google Ads API所必须的5个认证信息：

1. clientId -> 来自于OAuth Credentials配置
2. clientSecret -> 来自于OAuth Credentials配置
3. Refresh Token -> 使用OAuth Credentials通过Client Library生成
4. Developer Token -> 来自于Google Ads控制台界面
5. LoginCustomerId -> 你的Google Ads的经理账号ID。

提示： LoginCustomerId对应的Google Ads的经理账号，并不一定必须是你创建Developer Token所使用的经理账号，但是这两个经理账号必须是隶属于相同管理员的账号，即相同的登陆邮箱。

将对应的Google Ads API Library下载到本地，并导入你的编译软件中。这里我们使用[Java Maven](#)作为范例。其他编程语言的Library，可以在这个[链接](#)下载到。

提示： 因为本范例使用Java Maven，所以我们需要在根目录下创建ads.properties文件。如果使用其他编程语言，请参考上面链接，查看对应语言的Library的README文件，搜索配置文件的存储位置和编写方式。本示例中，我们将Java Library包下载到文件夹/Users/zhangsan/Desktop/google-ads-java，对应的ads.properties文件需要放置在该文件夹的根目录中，即/Users/zhangsan/

```
##### Configuration file start #####
#Tue Sep 10 20:58:45 CST 2019
api.googleads.refreshToken=
api.googleads.clientSecret=
api.googleads.clientId=
api.googleads.developerToken=
# Required for manager accounts only: Specify the login customer ID used to
# authenticate API calls. This will be the customer ID of the authenticated
# manager account. You can also specify this later in code if your application
# uses multiple manager account + OAuth pairs.
#
api.googleads.loginCustomerId=
##### Configuration file end #####
```

使用任何文本编译工具，例如nano, vim等，打开ads.properties文件。你会看到类似上图的输出。根据我们在之前步骤获取到的信息，请自行填充ads.properties文件里的对应内容。填写完成后，进行保存。

```

macbookpro:google-ads-examples gasliu$ mvn exec:java -Dexec.mainClass="com.google.ads.googleads.examples.accountmanagement.GetAccountInformation" -Dexec.args="--customerId 58XXXXXXXX56"
Scanning for projects...

-----< com.google.api-ads:google-ads-examples >-----
Building Google Ads API client library for Java examples 4.1.1-SNAPSHOT
[ jar ]

--- exec-maven-plugin:1.6.8:java (default-cli) @ google-ads-examples ---
: An illegal reflective access operation has occurred
: Illegal reflective access by com.google.protobuf.UnsafeUtil (file:/Users/gasliu/.m2/repository/com/google/protobuf/protobuf-java/3.5.1/protobuf-java-3.5.1.jar) to field java.nio.Buffer.address
: Please consider reporting this to the maintainers of com.google.protobuf.UnsafeUtil
: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
: All illegal access operations will be denied in a future release
>-18 18:07:25,462 INFO [Gax-3] logging.RequestLogger (RequestLogger.java:157) - SUCCESS REQUEST SUMMARY. Method: google.ads.googleads.v2.services.CustomerService/GetCustomer, Endpoint: googleads.g
is.com:443, CustomerID: null, RequestID: Zfb7nQDqEMM2oKlY1Q0C2w, ResponseCode: OK, Fault: null.
r with ID 58XXXXXXXX56, descriptive name 'Gus Test User Account', currency code 'USD', timezone 'America/Phoenix', tracking URL template '' and auto tagging enabled 'false' was retrieved.

BUILD SUCCESS
-----
Total time: 4.992 s
Finished at: 2019-09-18T18:07:25+08:00
-----

```

配置完成之后，你就可以开始发出自己的第一个Google Ads API了<[GetAccountInformation](#)>，我们使用这个API去获取58XXXXXXXX56这个账号的基本信息。从而，我们完成了第一个Google Ads API的所有配置并成功发起了第一个Google Ads的API操作。

提示：58XXXXXXXX56这个账号并不是LoginCustomerId，即，这个账号不是你的Google Ads经理账号，而是一个普通用户账号。也就是说，当你配置了经理账号后，只要你知道用户账号ID，你就可以利用API去过去该经理账号下所有管理维护的用户账号。换言之，你不能用你的经理账号，去对其他经理账号下的用户账号发送API请求。