**Adwords / Bing Ads tool guide**

Date: **1st May 2017**

Version: **1.0**

Table of Contents

[Adwords Setup 2](#_Toc481447755)

[Bing Ads Setup 3](#_Toc481447756)

[Refresh tokens 4](#_Toc481447757)

[Bing Ads 4](#_Toc481447758)

[Adwords 5](#_Toc481447759)

[Tool Usage 6](#_Toc481447760)

[Troubleshooting 7](#_Toc481447761)

[Adwords 7](#_Toc481447762)

[Bing Ads 7](#_Toc481447763)

# Adwords Setup

Please follow steps that you have not done yet before:

<https://developers.google.com/adwords/api/docs/guides/signup>

**Step 1. Get MCC account** (an MCC account has a menu item in ‘Accounts Setting’ called ‘AdWords API Center’) – you / your client may already have such account that you can reuse. If you are going to create new account, I’d suggest to select option ‘Manage my own accounts’ rather than ‘Manage account on behalf of customer’, I believe there will be less stringent terms and conditions from Google if you say that it is to manage other peoples

**Step 2. Get developer Token**, this token is then value for “**api.adwords.developerToken**” property. Please note that developer token needs to be approved for production data access. This is independent step that needs to be specifically requested. While the token is in Pending state, it can be used to access only the development accounts

**Step 3. Gain access to production data**, fill in and submit the form ‘Apply for Basic Access’. It will take several days for Google to process application, however its is important to get the approved developer token as the pending developer token is allowed to access only the Test Account(s).

**Step 4. Set up OAuth2 authentication (from** https://developers.google.com/adwords/api/docs/guides/first-api-call#set\_up\_oauth2\_authentication**)**

Your app will need to access user data and contact other Google services on your behalf. Authentication via [OAuth2](http://oauth.net/2/)allows your app to  operate on behalf of your account.

To enable your app to access the API, you need an OAuth2 **client ID** and **client secret**.

'Client' in this case means your client app, not an AdWords client account.

1. While logged in with your manager account credentials, open the [Google API Console Credentials page](https://console.developers.google.com/apis/credentials) (https://console.developers.google.com/apis/credentials).
2. From the project drop-down, choose Create a new project, enter a name for the project, and click Create.
3. Select Create credentials and choose OAuth client ID.
4. You may be prompted to set a product name on the Consent screen; if so, click Configure Consent Screen, supply the requested information, and click Save to return to the Credentials screen.
5. Under Application type, choose Other for this tutorial. Enter a name in the space provided. For the AdWords API, Other is the option to choose for an installed application flow. Before you build your app, you'll want to read up on the [difference between installed and web applications](https://developers.google.com/adwords/api/docs/guides/authentication#determine_your_app_type), and choose the appropriate type for the app you're building.
6. Click Create. The OAuth2 client ID and client secret appear. Copy and save these items.

These are values for:

**api.adwords.clientId**=

**api.adwords.clientSecret**=

# Bing Ads Setup

Step 1.

See <https://msdn.microsoft.com/en-us/library/bing-ads-content-api-getting-started(v=msads.90).aspx>

To use the Content API, you must have Bing Ads account and credentials. New users to Bing Ads must use a Microsoft account to access Bing Ads. Current users that haven't transitioned to a Microsoft account may use their Bing Ads user name.

To get a Bing Ads account, go to [http://bingads.microsoft.com](http://bingads.microsoft.com/). If you're not signed in to your Microsoft account, you'll be redirected to sign in to your Microsoft account or sign up for a Microsoft account. After signing in, you'll have the option to **Sign up for a new Bing Ads account**. Select the sign up option and **Continue** with the sign up process.

Next, you'll need a developer token. To get a token to use in the production environment, see [Bing Ads Developer Portal](https://developers.bingads.microsoft.com/account) (https://developers.bingads.microsoft.com/account). Click **Request Token** and provide the requested information. If you will be using the API to manage your own account, you'll receive your token immediately; however, if you'll be managing accounts for others, it may take up to five business days to get a token.

Step 2.

See <https://msdn.microsoft.com/library/bing-ads-user-authentication-oauth-guide.aspx>

Complete the ‘Register your application’ step as described to obtain ClientId of the tool:

1. Go to <https://apps.dev.microsoft.com/>, and login with your Microsoft Account credentials when prompted.
2. **Important: Do not select a platform but click directly on ‘My Applications’ link**
3. Click ‘Add an App’ button and enter the name
4. In ‘Application Secrets’ section click ‘Generate New Password’. Tool does not really need it but I believe it is important to have it.
5. In “Platform” section click ‘Add Platform’ and choose ‘Native Application’ option
6. Copy the Client ID value to **api.bing.clientI**d property
7. Click ‘Save’ button at the page bottom to save the changes

Step 3.

Go to Bing Ads account home page, click the Gear icon and choose ‘Accounts & Billing’ option

In the ‘Accounts’ copy value of ‘Account ID’ to **api.bing.accountId** property and Customer ID value to **api.bing.customerId** property.

# Refresh tokens

This section describes how to obtain the refresh token. The tool contains utility method that implements the get RefreshToken algorithm for both Adwords and Bing Ads.

## Bing Ads

Prerequisite: you must already have the DeveloperToken and have registered the tool and obtained its ClientId.

Run the tool as:

**$ java -cp Adwords-1.0-executable.jar tools.GetBingRefreshToken --DeveloperToken '<DeveloperToken>' --ClientId '<ClientId>'**

The tool uses JavaFX to run simple web client to perform tasks required to obtain the RefreshToken.   
After startup, a simple web client window appears. The tool will connect to login.live.com and will display a Microsoft login dialog. Log-in with the account credentials that belongs to the provided DeveloperToken.

Next the consent dialog appears asking you to confirm access of the tool to the account data. After confirmation action the web client windows closes and tool exits.

Return back to the console output (terminal window where you started the tool, the output will look similar to this:

##################################################################

Token refresh time: 05/01/2017 18:25:34

New access token:



**Your Refresh token:**

**MCfj!Z4rgc7RE1OPNypCCERsZD0Ay85EfTUO0kGhJxWZa4IBthbpUOfahrh1jdX0NN6ajsa6dTLS2Q5iWrNReh6Qfid74xK91hOVVdiC4OcSqoaTZLthvTyNgZz0zLJt74LkCHJP!Vs2MrIU2BpNzr4kpE4lO0tgFgMdi2yKmC0yC9xmYjdvAjpOBlzoKfFidOxKg63R33J8w3TcCXTSOrZcuOnNUiLvCnECxU9I6YhItU\*58PjkXJmsFT4SDltJy4IRX\*hQx\*6HNNPe\*QfTaFaYDVpKiM!X3lRD6gfRe86exJnVp0vMZhxwv6gEHv1BFVHf7W\*CMZ7orymgzlHHygT0**$

##################################################################

Copy the refresh token to config file to property **api.bing.refreshToken**

## Adwords

Prerequisite: you must already have the adwords MCC account and its developer token and you must have also register the tool as native / other application and obtained it’s clientId, and stored these values in configuration file (properties: api.adwords.developerToken, api.adwords.clientId).

**$ java -cp Adwords-1.0-executable.jar tools.GetAdwordsRefreshToken --config <path-to-ads.properties>**

The tool will prepare and print the custom URL that you need to paste into browser. If necessary (if prompted), log-in to the adwords account with credentials that belong to the DeveloperToken. Browser will then display a consent dialog whre you hav eto confirm access to the account by the tool.  
The confirmation page will print a unique code. Copy the code and paste it to the console and it <Enter>.  
The tool will consume this unique code and generate the RefreshToken.  
The sample screen output should look similar to this:

Paste this url in your browser:

https://accounts.google.com/o/oauth2/auth?access\_type=offline&client\_id=897796100522-lpivmon0dipt7v94sbmjjbcfcm1t8598.apps.googleusercontent.com&redirect\_uri=urn:ietf:wg:oauth:2.0:oob&response\_type=code&scope=https://www.googleapis.com/auth/adwords

Type the code you received here:

4/Frj6VbF69INxiDXKc3SCqavn1TvdLvE6IzM4CB0LI1k

Your refresh token is: 1/WkB6pmb1MoeChbR-mbL2za6ouSyalrfj2dL2n1kNV8s

In your ads.properties file, modify:

**api.adwords.refreshToken=1/WkB6pmb1MoeChbR-mbL2za6ouSyalrfj2dL2n1kNV8s**

# Tool Usage

Run tool as:

**java -jar Adwords-1.0-executable.jar --config <ads.properties path> --refreshtoken "refreshtoken" --obsWindow 8 --lag 24 --dbName js871 --provider [adwords|bing] --debug true --removeDownloadedReportFile true**

or the minimum parameters (using all values from config file)

**java -jar Adwords-1.0-executable.jar --config ads.properties --provider [adwords|bing]**

Parameter explanation:

--**refreshToken** Refresh token for the client

--**obsWindow** Observation window. Program should download the statistics for the duration of XXX hours

--**lag** number of hours the collection is delayed

--**dbName** Name of the database to save results in (collection name is "costs\_adwords")

--**domainName** Name of the subdomain within AdWords account for which to download data from.

--**config** full or relative path to ads.propeties file with all configuration entries

--**provider** [adwords|bing|all] tell the tool which API to use. 'all' means both (all) implemented API providers

--**reportDirectory** directory (used by Bing Ads where to store temporary downloaded files)

--**removeDownloadedReportFile** [true|false] indicate if to keep or delete downloaded temporary files

--**loadFromFile** <fileName> a CSV file from which to load the data (do not download from API but use the CSV file instead). This parameter is not compatible with '--provider all' option

.

(Important note: Although I took care to paste the correct ASCII code, it is possible that ‘-‘ char is converted by MSWord to EM-dash (non-ASCII) character. Please consider this option if copy & paste the command from this document and receiving unusual error, like “Cannot find class –jar”)

# Troubleshooting

Set **--debug true** command line option.

Check the SOAP message(s) returned from Adwords / Bing Ads API, look for any string detailing the SOAPFault (error) that occurred.

## Adwords

Check the page <https://developers.google.com/adwords/api/docs/common-errors> for explanation of the error and possible course of action to fix it.

## Bing Ads

https://msdn.microsoft.com/en-us/library/bing-ads-error-handling-guide(v=msads.100).aspx#Common Errors