Functional Document for

Bing API

* **Getting Started With the Bing Ads API**

Any Bing Ads user with a developer token can begin using the Bing Ads Application Programming Interface (API)

# Downloading Reports from the Bing Ads API

# You can extract performance data from the Bing Ads API for metrics such as cost, impressions, and clicks etc. The simplest way to do it is using the report service: It enables you to pull down similar reports to that which you can find on the Reports tab of the web interface.

# [Setting Up the Development Environment](javascript:void(0))

# You need user credentials, developer token with access to Bing Ads either in production or sandbox to download a report from Bing Ads API

## [Getting a Developer Token](javascript:void(0))

* + To use Bing Ads APIs, you must have a developer token and valid user credentials.
  + If you do not yet have a Bing Ads account, go to the [Bing Ads](https://bingads.microsoft.com/Default.aspx) web application, and No account? [Create one!](https://signup.live.com/signup.aspx?wa=wsignin1.0&rpsnv=12&ct=1460111997&rver=6.5.6510.0&wp=SAPI&wreply=https:%2F%2Faccount.live.com%2Fconsent%2FManage&id=38936&bk=1460112000&uiflavor=web&uaid=72a36af9d5374eada500a2fe673f422c&mkt=EN-IN&lc=16393).(  <https://account.live.com/consent/Manage>)
  + To get a developer token for production, you must be logged into the [Bing Ads Developer Portal](https://developers.bingads.microsoft.com/Account) as a Microsoft Account user with the Super Admin role. The Super Admin may request API access for any user within their customer scope.

## [Where to Use the API Credentials](javascript:void(0))

* + Bing Ads services use Simple Object Access Protocol (SOAP) to exchange the request and response messages with the service operation
  + Each SOAP request must include the following SOAP headers, which contain the user’s credentials.

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Description** | **Data Type** | **Required** |
| ApplicationToken | The application-access token. | string | No. This header element is not used and should be null. |
| AuthenticationToken | The OAuth access token used to manage Bing Ads accounts linked to a Microsoft Account. | string | Required if the UserName and Password elements are not specified. |
| CustomerAccountId | The identifier of the account that owns the entities in the request. This header element must have the same value asthe AccountId body element when both are required. | string | Required for service operations related to ad extensions and bid estimations. As a best practice you should always specify this element for operations limited in scope to a single account per service call. |
| CustomerId | The identifier of the customer that owns the account. | string | Required for service operations related to targeting and editorial. As a best practice you should always specify this element. |
| DeveloperToken | The client application's developer access token. | string | Yes. |
| Password | The Bing Ads user's sign-in password. | string | Required if the Authentication Token element is not specified. |
| UserName | The Bing Ads user's sign-in user name. You may not set this element to a Microsoft account. | string | Required if the AuthenticationToken element is not specified. |

* [**Installing the SDK**](javascript:void(0))

The Bing Ads Java SDK depends on the libraries listed at the [Maven Repository](http://mvnrepository.com/artifact/com.microsoft.bingads/microsoft.bingads/).

Depending upon the functions used to download a report all the import statements must be included and their respective jar files also should be included

## [Using AuthorizationData](javascript:void(0))

You must initialize a new instance of ServiceClient or BulkServiceManager with AuthorizationData. The class contains properties that Bing Ads uses to authorize a user. The ServiceClient, BulkServiceManager, and ReportingServiceManager classes handle common request header fields for you, allowing you to specify the Authentication, CustomerId, AccountId, and DeveloperToken properties in the AuthorizationData object once for each service.

* The ServiceClient class handles common request header fields for you, allowing to specify the Authentication, CustomerId, AccountId, andDeveloperToken properties in the AuthorizationData object once for each service
* For example the following sample shows how to use the Customer Management service to get the current authenticated user.

ServiceClient<ICustomerManagementService> CustomerService = new ServiceClient<ICustomerManagementService>(

authorizationData ICustomerManagementService.class);

java.lang.Long userId = null;

final GetUserRequest getUserRequest = new GetUserRequest();

getUserRequest.setUserId(userId);

User user = CustomerService.getService().getUser(getUserRequest).getUser();

* The following code block shows how to create an instance of AuthorizationData and set its Authentication, CustomerId, AccountId, andDeveloperToken properties.

static AuthorizationData authorizationData = new AuthorizationData();

authorizationData.setAuthentication(<AuthenticationGoesHere>);

authorizationData.setCustomerId("<CustomerIdGoesHere>");

authorizationData.setAccountId("<AccountIdGoesHere>");

authorizationData.setDeveloperToken("<DeveloperTokenGoesHere>");

## [Using OAuth](javascript:void(0))

The OAuth classes in the client library abstract the low level user authorization details, so you can focus at a high level on your security requirements. The client library objects take care of low level details such as formatting the authorization and redirect URIs and setting the request headers.

To use OAuth with the Bing Ads Java SDK, the Authentication property of your AuthorizationData object must be set to an Authentication-derived class such as OAuthWebAuthCodeGrant, OAuthDesktopMobileAuthCodeGrant or OAuthDesktopMobileImplicitGrant. For repeat or long term authentication, you should follow the authorization code grant flow for obtaining an access token

* At a high level you should complete the following steps to authenticate a Microsoft Account with Bing Ads using OAuth.
* [Registering Your Application](https://msdn.microsoft.com/en-us/library/bing-ads-user-authentication-oauth-guide.aspx#registerapplication) your application.
* Request user consent for your application to manage their Bing Ads accounts, by initiating either the [Implicit Grant Flow](https://msdn.microsoft.com/en-us/library/bing-ads-user-authentication-oauth-guide.aspx#implicit) or [Authorization Code Grant Flow](https://msdn.microsoft.com/en-us/library/bing-ads-user-authentication-oauth-guide.aspx#authorizationcode).
* Complete either the [Implicit Grant Flow](https://msdn.microsoft.com/en-us/library/bing-ads-user-authentication-oauth-guide.aspx#implicit) or [Authorization Code Grant Flow](https://msdn.microsoft.com/en-us/library/bing-ads-user-authentication-oauth-guide.aspx#authorizationcode) to obtain an access token that can be used to authenticate with Bing Ads services.
* For each API call to Bing Ads, use the returned access token as the AuthenticationToken element within the Bing Ads service [Service Request Header](https://msdn.microsoft.com/en-us/library/bing-ads-user-authentication-oauth-guide.aspx" \l "serviceheaders). For more information, see [Managing OAuth Tokens](https://msdn.microsoft.com/en-us/library/bing-ads-user-authentication-oauth-guide.aspx#managingoauthtokens).

## [Registering Your Application](javascript:void(0))

Before you can manage authentication for users of your Bing Ads application, you must register your application and get the corresponding client ID and client secret.

1. Go to <https://account.live.com/developers/applications>, and login with your Microsoft Account credentials when prompted.
2. Under **My apps**, click **Create application**.
3. Provide the application name and language, and accept the terms of use.
4. Specify the redirect domain for a web application, and otherwise indicate that your application is a mobile or desktop application.
5. Save your changes and take note of your client ID. Also take note of your client secret and redirect URI if you have a web application. You will use these values to manage authentication with OAuth.

* [**Managing OAuth Tokens**](javascript:void(0))

Once you have registered your application you can manage the access token for a Microsoft Account user already linked or registered with Bing Ads. For one time or short term access to manage a user's accounts, see [Implicit Grant Flow](https://msdn.microsoft.com/en-us/library/bing-ads-user-authentication-oauth-guide.aspx#implicit). The access token is short lived and will expire in minutes or hours as determined by the authentication service. Additionally, the Microsoft Account user may change their password or remove permissions for your application to authenticate on their behalf. For repeat or long term access to manage a user's accounts

### Implicit Grant Flow

For one time or short term authentication, you should follow the implicit grant flow for obtaining an access token. This is a standard OAuth 2.0 flow and is defined in detail in the [Implicit Grant section of the OAuth 2.0 spec](http://tools.ietf.org/html/rfc6749#section-4.2).

***Note:****- For web applications, do not use implicit grant flow and instead use a client secret with the*[*Authorization Code Grant Flow*](https://msdn.microsoft.com/en-us/library/bing-ads-user-authentication-oauth-guide.aspx#authorizationcode)

1.Request user consent through a web browser control. Connect to the authorization endpoint, by using a URL in the following format. Replace CLIENT\_ID with the value configured in [Registering Your Application](https://msdn.microsoft.com/en-us/library/bing-ads-user-authentication-oauth-guide.aspx#registerapplication).

<https://login.live.com/oauth20_authorize.srf?client_id=CLIENT_ID&scope=bingads>.manage&response\_type=token&redirect\_uri=https://login.live.com/oauth20\_desktop.srf

***Note:-****For a desktop or mobile application, use https://login.live.com/oauth20\_desktop.srf as the redirect URI*

2.The user will be prompted through the Microsoft Account authorization web browser control to grant permissions for your application to manage their Bing Ads accounts.

3.The authorization service calls back to your application with the redirection URI, which includes an access token if the user authorized your application to manage their Bing Ads accounts. For example the callback URI includes an access token as follows if the user granted permissions for your application to manage their Bing Ads accounts: <https://login.live.com/oauth20_desktop.srf?vv=1550&lc=1033#access_token=ACCESS_TOKEN>.

***Note:-****If the user denied your application permissions to manage their Bing Ads accounts, the callback URI includes an error and error description field as follows:*[*https://login.live.com/oauth20\_desktop.srf?vv=1550&lc=1033#error=ERROR&error\_description=ERROR\_DESCRIPTION*](https://login.live.com/oauth20_desktop.srf?vv=1550&lc=1033#error=ERROR&error_description=ERROR_DESCRIPTION)

4.Use the returned access token as the AuthenticationToken element within Bing Ads service [Service Request Header](https://msdn.microsoft.com/en-us/library/bing-ads-user-authentication-oauth-guide.aspx" \l "serviceheaders).

## Defining Reports in the Bing Ads API

You request reports by sending SOAP objects of the specific type for the report you want to produce: For example, to produce a keyword report you would use the KeywordPerformanceReportRequest, and for an ad report you would use the AdPerformanceReportRequest etc. All of the request types derive from a base type called ReportRequest that has the following properties:

# ReportRequest Data Object

## Elements

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Description** | **Data Type** | **Required/Optional** |
| Format | The format of the report data. For example, you can request the data in comma-separated values (Csv) format or tab-separated values (Tsv) format. The default value is Csv. **Note**: All downloaded files are ZIP compressed. | [ReportFormat](https://msdn.microsoft.com/en-us/library/bing-ads-reporting-reportformat(v=msads.90).aspx) | Optional |
| Language | The language to use to generate the report headers and columns. The default is English. | [ReportLanguage](https://msdn.microsoft.com/en-us/library/bing-ads-reporting-reportlanguage(v=msads.90).aspx) | Optional |
| ReportName | The name of the report. The name is included in the report header. If you do not specify a report name, the system generates a name in the form, ReportType-ReportDateTime. The length of the report name must be less than or equal to 200. | string | Optional |
| ReturnOnlyCompleteData | Determines whether you want the service to generate the report only if all the data has been processed and is available. If **true**, the request fails if the system has not finished processing all the data based on the aggregation, scope, and time period values that you specify. However, if **false**, the request succeeds but the report will contain only the data that the system has finished processing at the time of the request (there is no indication as to whether the data is complete). The default is**false**. **Note**: When a user clicks an ad, it can take from three to four hours for the system to process the click and make it available for reporting. For more information, see [Determining When the Books Close](http://go.microsoft.com/fwlink/?LinkId=691014).  Because you cannot retrieve complete data for today, you must set this element to **false** if the end date of the custom date range specified in the Time element of the derived report object is set to today or the Time element contains one of the following predefined time values: Today, ThisWeek, ThisMonth,ThisYear | boolean | Optional |
| Aggregation | The type of aggregation to use to aggregate the report data. For example, you can aggregate the report data by day or week, or request a summary report.  The default is Summary.  The Time element specifies the time period to use for the aggregation. | [ReportAggregation](https://msdn.microsoft.com/en-us/library/bing-ads-reporting-reportaggregation(v=msads.90).aspx) | Optional |
| Columns | The list of attributes and performance statistics to include in the report. The report will include the columns in the order that you specify them. | <reportnamecolumn> array | Required |
| Filter | The filter information to use to filter the report data. | <[reportname>Filter](https://msdn.microsoft.com/en-us/library/bing-ads-reporting-keywordperformancereportfilter(v=msads.90).aspx) | Optional |
| MaxRows | The top number of data rows to return in the report. | int | Optional |
| Scope | The scope of the report. Use this element to limit the report to include data for a combination of accounts, ad groups, and campaigns. | [AccountThroughAdGroupReportScope](https://msdn.microsoft.com/en-us/library/bing-ads-reporting-accountthroughadgroupreportscope(v=msads.90).aspx) | Required |
| Sort | A list of the columns to sort, and the corresponding sort order. | <reportname>array | Optional |
| Time | The time period to use for the report. You can specify a custom date range or select a predefined date range, for example, Today or ThisWeek.  For a list of the time periods that you can specify for each aggregation type, see[Aggregation and Time](http://go.microsoft.com/fwlink/?LinkId=691012). | [ReportTime](https://msdn.microsoft.com/en-us/library/bing-ads-reporting-reporttime(v=msads.90).aspx) | Required |

### Background Completion with ReportingServiceManager

You can submit a download request and the ReportingServiceManager will automatically return results. The ReportingServiceManager abstracts the details of checking for result file completion

* **Format** – To specify if you want the report in a CSV or XML file type.
* **ReportName** – A descriptive name for the report.
* **ReturnOnlyCompleteData** – Specifies if you want to see any data at all if the books aren’t closed for the chosen period of the report: For example, if you specify a time period of Today, then the data won’t be fully processed yet; so if you supply false for this parameter, then you'll get what’s currently available; alternatively, if you specifytrue, then the report won’t be produced.

In addition to these, most of the derived types also have the following properties:

* **Aggregation** – The grouping of data according to time; valid values include Hourly,Daily, Weekly, Monthly, Yearly, and Summary. There are constraints on what values can be used in each report; one example is that some reports don’t allow Hourly aggregation. There’s also constraints about which time periods and columns can be combined with specific aggregation types: you can only use the Today, Yesterday, and Custom Date Rangetimes with Hourly aggregation; and you can’t use the TimePeriod column in Summaryreports.

**For Example Sample Code for Keyword Performance Report:-**

ReportingServiceManager = new ReportingServiceManager(authorizationData);

ReportingServiceManager.setStatusPollIntervalInMilliseconds(5000);

KeywordPerformanceReportRequest report = new KeywordPerformanceReportRequest();

report.setFormat(ReportFormat.CSV);

report.setReportName("My Keyword Performance Report");

report.setReturnOnlyCompleteData(true);

report.setAggregation(ReportAggregation.DAILY);

* **Columns –** The list of attributes and performance statistics to include in the report. The report will include the columns in the order that you specify them.. These are specific to each report.For example, the keyword performance report has columns such as Keyword, MatchType,Spend, and Impressions. The TimePeriod column can be included in most reports to show the date of a row.

**Example**

ArrayOfKeywordPerformanceReportColumn keywordPerformanceReportColumns = new ArrayOfKeywordPerformanceReportColumn(); keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.TIME\_PERIOD);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.KEYWORD);

….etc

* **Filter** – For removing specific rows from a report. Most reports have filters available, but they are specific to each one.
* **Scope** – Similar to the filter, it enables you to reduce the number of rows in the report. This is based on id’s of specific accounts, campaigns, and ad groups etc; if the scope is empty then the report will include all accounts for which the users has access.
* **Time** – The date range of stats to include in the report: Either use a preset value by specifying a PredefinedTime, or use the  CustomDateRangeStart  and CustomDateRangeEndproperties. If you use the predefined time, then the value should be one of Today,Yesterday, ThisWeek, LastWeek, ThisMonth, LastMonth, ThisYear, LastYear, LastSevenDays,LastFourWeeks, LastThreeMonths, or LastSixMonths; for custom dates, you need to supply aDate object with appropriate Day, Month, and Year fields.

ReportTime rt=new ReportTime();

String time\_period = context.TIME\_PERIOD;

ReportTimePeriod rt\_d=ReportTimePeriod.valueOf(time\_period);

rt.setPredefinedTime(rt\_d);

System.out.println("Time period-------"+time\_period);

//rt.setPredefinedTime(ReportTimePeriod.context.TIME\_PERIOD);

report.setTime(rt);

### Background Completion with ReportingServiceManager

You can submit a download request and the ReportingServiceManager will automatically return results. The ReportingServiceManager abstracts the details of checking for result file completion, and you don't have to write any code for results polling.

ReportingServiceManager = new ReportingServiceManager(authorizationData);

ReportingServiceManager.setStatusPollIntervalInMilliseconds(5000);

ReportRequest reportRequest = getKeywordPerformanceReportRequest();

ReportingDownloadParameters reportingDownloadParameters = new ReportingDownloadParameters();

reportingDownloadParameters.setReportRequest(reportRequest);

reportingDownloadParameters.setResultFileDirectory(new File(FileDirectory));

reportingDownloadParameters.setResultFileName(ResultFileName);

reportingDownloadParameters.setOverwriteResultFile(true);

// You may optionally cancel the downloadFileAsync operation after a specified time interval.

File resultFile = ReportingServiceManager.downloadFileAsync(

reportingDownloadParameters,

null).get(3600000, TimeUnit.MILLISECONDS);

System.out.println(String.format("Download result file: %s\n", resultFile.getName()));

* **Error Handling**

If Any Exception like Api Error, Batch Error, OperationError, OAuth tokens, ReportOperation Error etc are occurred in the code those exception are Caught by the flowing block of code

catch (ExecutionException ee) {

Throwable cause = ee.getCause();

if (cause instanceof AdApiFaultDetail\_Exception) {

AdApiFaultDetail\_Exception ex = (AdApiFaultDetail\_Exception)cause;

System.out.println("The operation failed with the following faults:\n");

for (AdApiError error : ex.getFaultInfo().getErrors().getAdApiErrors())

{

System.out.println("AdApiError\n");

System.out.println(String.format("Code: %d\nError Code: %s\nMessage: %s\n\n",

error.getCode(), error.getErrorCode(), error.getMessage()));

}

} else if (cause instanceof ApiFaultDetail\_Exception) {

ApiFaultDetail\_Exception ex = (ApiFaultDetail\_Exception)cause;

System.out.println("The operation failed with the following faults:\n");

for (BatchError error : ex.getFaultInfo().getBatchErrors().getBatchErrors())

{

System.out.println(String.format("BatchError at Index: %d\n", error.getIndex()));

System.out.println(String.format("Code: %d\nMessage: %s\n\n", error.getCode(), error.getMessage()));

}

for (OperationError error : ex.getFaultInfo().getOperationErrors().getOperationErrors())

{

System.out.println("OperationError\n");

System.out.println(String.format("Code: %d\nMessage: %s\n\n", error.getCode(), error.getMessage()));

}

} else {

ee.printStackTrace();

}

} catch (OAuthTokenRequestException ex) {

System.out.println(String.format("Couldn't get OAuth tokens. Error: {0}. Description: {1}",

ex.getDetails().getError(), ex.getDetails().getDescription()));

ex.printStackTrace();

} catch (ReportingOperationInProgressException ex) {

System.out.println("The result file for the reporting operation is not yet available for download.");

ex.printStackTrace();

} catch (ReportingOperationCouldNotBeCompletedException ex) {

System.out.println(String.format("ReportingOperationCouldNotBeCompletedException Message: {0}", ex.getMessage()));

ex.printStackTrace();

} catch (InterruptedException ex) {

ex.printStackTrace();

}

* **Sample Code to download a report from Bing API (Keyword Performance Report)**

**final** java.lang.String FileDirectory = context.STAGING\_DIR\_PATH;

// The name of the report file.

**final** java.lang.String ResultFileName = "keyword\_performance\_report\_"+AccountId+"\_"+TalendDate.getDate("yyyyMMdd")+".csv";

// The report file extension type.

// final ReportFormat ReportFileFormat = ReportFormat.CSV;

//String folder = FileDirectory.substring(0, FileDirectory.lastIndexOf('\\'));

String folder = context.STAGING\_DIR\_PATH;

File dir = **new** File(folder);

**if** (!dir.exists())

{

System.out.println("The output folder does not exist. Ensure that the " + "folder exists and try again.");

**return**;

}

**try**

{

AuthorizationData authorizationData=(AuthorizationData)context.OAUTH\_DATA;

System.out.println("AuthorizationData retrived----- "+authorizationData);

authorizationData.setAccountId(AccountId);

ReportingServiceManager = **new** ReportingServiceManager(authorizationData);

ReportingServiceManager.setStatusPollIntervalInMilliseconds(5000);

KeywordPerformanceReportRequest report = **new** KeywordPerformanceReportRequest();

report.setFormat(ReportFormat.CSV);

report.setReportName("My Keyword Performance Report");

report.setReturnOnlyCompleteData(**true**);

report.setAggregation(ReportAggregation.DAILY);

System.out.println("Report param step 1"+report.toString());

ArrayOfKeywordPerformanceReportColumn keywordPerformanceReportColumns = **new** ArrayOfKeywordPerformanceReportColumn();

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.ACCOUNT\_NAME);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.ACCOUNT\_NUMBER);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.ACCOUNT\_ID);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.TIME\_PERIOD);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.CAMPAIGN\_NAME);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.AD\_GROUP\_NAME);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.KEYWORD);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.KEYWORD\_ID);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.AD\_ID);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.AD\_TYPE);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.DESTINATION\_URL);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.CURRENT\_MAX\_CPC);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.CURRENCY\_CODE);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.DELIVERED\_MATCH\_TYPE);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.AD\_DISTRIBUTION);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.IMPRESSIONS);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.CLICKS);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.CTR);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.AVERAGE\_CPC);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.SPEND);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.AVERAGE\_POSITION);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.CONVERSIONS);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.CONVERSION\_RATE);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.COST\_PER\_CONVERSION);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.AVERAGE\_CPM);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.PRICING\_MODEL);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.BID\_MATCH\_TYPE);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.DEVICE\_TYPE);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.QUALITY\_SCORE);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.QUALITY\_IMPACT);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.KEYWORD\_RELEVANCE);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.LANDING\_PAGE\_RELEVANCE);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.LANDING\_PAGE\_USER\_EXPERIENCE);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.LANGUAGE);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.HISTORIC\_QUALITY\_SCORE);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.HISTORIC\_KEYWORD\_RELEVANCE);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.HISTORIC\_LANDING\_PAGE\_RELEVANCE);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.HISTORIC\_LANDING\_PAGE\_USER\_EXPERIENCE);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.BUSINESS\_LISTING\_ID);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.BUSINESS\_LISTING\_NAME);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.BUSINESS\_CATEGORY\_ID);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.BUSINESS\_CATEGORY\_NAME);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.CAMPAIGN\_STATUS);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.ACCOUNT\_STATUS);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.AD\_GROUP\_STATUS);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.KEYWORD\_STATUS);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.NETWORK);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.TOP\_VS\_OTHER);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.DEVICE\_OS);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.ASSISTS);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.EXTENDED\_COST);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.REVENUE);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.RETURN\_ON\_AD\_SPEND);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.COST\_PER\_ASSIST);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.REVENUE\_PER\_CONVERSION);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.REVENUE\_PER\_ASSIST);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.BOUNCE\_RATE);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.TOTAL\_VISITS);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.AVERAGE\_PAGES\_PER\_VISIT);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.AVERAGE\_DURATION\_PER\_VISIT);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.TRACKING\_TEMPLATE);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.CUSTOM\_PARAMETERS);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.FINAL\_URL);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.FINAL\_MOBILE\_URL);

keywordPerformanceReportColumns.getKeywordPerformanceReportColumns().add(KeywordPerformanceReportColumn.FINAL\_APP\_URL);

report.setColumns(keywordPerformanceReportColumns);

System.out.println("Report param step 2");

AccountThroughAdGroupReportScope keywordPerformanceReportScope = **new** AccountThroughAdGroupReportScope();

ArrayOflong accArray=**new** ArrayOflong();

accArray.getLongs().add(authorizationData.getAccountId());

ArrayOflong campaignArray=**new** ArrayOflong();

// campaignArray.getLongs().add(authorizationData.getCampaignId());

keywordPerformanceReportScope.setAccountIds(accArray);

keywordPerformanceReportScope.setAdGroups(**null**);

keywordPerformanceReportScope.setCampaigns(**null**);

report.setScope(keywordPerformanceReportScope);

System.out.println("Report param step 3");

ReportTime rt=**new** ReportTime();

String time\_period = context.TIME\_PERIOD;

ReportTimePeriod rt\_d=ReportTimePeriod.valueOf(time\_period);

rt.setPredefinedTime(rt\_d);

System.out.println("Time period-------"+time\_period);

//rt.setPredefinedTime(ReportTimePeriod.context.TIME\_PERIOD);

report.setTime(rt);

System.out.println("Report param step 4");

ReportingDownloadParameters reportingDownloadParameters = **new** ReportingDownloadParameters();

reportingDownloadParameters.setReportRequest(report);

reportingDownloadParameters.setResultFileDirectory(**new** File(FileDirectory));

reportingDownloadParameters.setResultFileName(ResultFileName);

reportingDownloadParameters.setOverwriteResultFile(**true**);

System.out.println("Report param step 5 "+reportingDownloadParameters.getReportRequest());

File resultFile = ReportingServiceManager.downloadFileAsync(

reportingDownloadParameters, **null**).get();

System.out.println("Report param step 6");

System.out.println(String.format("Download result file: %s\n", resultFile.getName()));

} **catch** (ExecutionException ee) {

Throwable cause = ee.getCause();

**if** (cause **instanceof** AdApiFaultDetail\_Exception) {

AdApiFaultDetail\_Exception ex = (AdApiFaultDetail\_Exception)cause;

System.out.println("The operation failed with the following faults:\n");

**for** (AdApiError error : ex.getFaultInfo().getErrors().getAdApiErrors())

{

System.out.println("AdApiError\n");

System.out.println(String.format("Code: %d\nError Code: %s\nMessage: %s\n\n",

error.getCode(), error.getErrorCode(), error.getMessage()));

}

} **else** **if** (cause **instanceof** ApiFaultDetail\_Exception) {

ApiFaultDetail\_Exception ex = (ApiFaultDetail\_Exception)cause;

System.out.println("The operation failed with the following faults:\n");

**for** (BatchError error : ex.getFaultInfo().getBatchErrors().getBatchErrors())

{

System.out.println(String.format("BatchError at Index: %d\n", error.getIndex()));

System.out.println(String.format("Code: %d\nMessage: %s\n\n", error.getCode(), error.getMessage()));

}

**for** (OperationError error : ex.getFaultInfo().getOperationErrors().getOperationErrors())

{

System.out.println("OperationError\n");

System.out.println(String.format("Code: %d\nMessage: %s\n\n", error.getCode(), error.getMessage()));

}

} **else** {

ee.printStackTrace();

}

} **catch** (OAuthTokenRequestException ex) {

System.out.println(String.format("Couldn't get OAuth tokens. Error: {0}. Description: {1}",

ex.getDetails().getError(), ex.getDetails().getDescription()));

ex.printStackTrace();

} **catch** (ReportingOperationInProgressException ex) {

System.out.println("The result file for the reporting operation is not yet available for download.");

ex.printStackTrace();

} **catch** (ReportingOperationCouldNotBeCompletedException ex) {

System.out.println(String.format("ReportingOperationCouldNotBeCompletedException Message: {0}", ex.getMessage()));

ex.printStackTrace();

} **catch** (InterruptedException ex) {

ex.printStackTrace();

} /\*catch (IOException ex) {

ex.printStackTrace();

} catch (URISyntaxException ex) {

ex.printStackTrace();

}\*/