**Mobile UET SDK Integration Guide**

Contents App ID

[Legal Notice 2](#_Toc85719520)

[Shorthand 3](#_Toc85719521)

[Mobile UET SDK 4](#_Toc85719522)

[Scenarios 4](#_Toc85719523)

[Adding the UET SDK to your Project 5](#_Toc85719524)

[Add UET SDK Library 5](#_Toc85719525)

[Interface specification 6](#_Toc85719526)

# Legal Notice

This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2022 Microsoft. All rights reserved.

Microsoft is a trademark of the Microsoft group of companies. All other trademarks are property of their respective owners.

# Shorthand

advertiser: the caller of the UET SDK

customer: advertiser’s customer

# Mobile UET SDK

Universal Event Tracking (UET) is a Microsoft framework which captures user engagement data on your website, enabling features like conversion tracking, audience targeting and auto bidding. Please refer to [Microsoft Ads help documentation](https://help.ads.microsoft.com/#apex/ads/en/56681/2-500) for more info on UET and the use-cases it powers.

For the mobile platform, we provide a SDK to fire the UET events for the APPs.

## Scenarios

Two scenarios of Mobile UET SDK.

1 When a user opens the native app directly which integrated The Mobile UET SDK, it can track any conversion they defined.

A screenshot of a cell phone

Description automatically generated

2 When a user browses a website and wants to buy a product displayed on it. And we can track the conversion which came from this website by passing the WEB\_USER\_ID parameter.

Graphical user interface, text, application

Description automatically generated

## Adding the UET SDK to your Project

UET SDK provide the ability to send the events to the Bing Ads server. For the first step, integrate to your project. Call the SDK interface in right format to send the info.

### Add UET SDK Library

Create a directory in your app path of project. Put the package *uetsdk.jar* in the directory.

Graphical user interface, application

Description automatically generated

Implemented the UET SDK and the library depended. There is one package UET SDK used called *okhttp3*. Its version is 3.11.0 or after. Modify the file *build.gradle* in module wise. Add the implementation in the file as in red font the below.

Graphical user interface, text

Description automatically generated

dependencies **{** implementation 'androidx.appcompat:appcompat:1.3.1'  
 implementation 'com.google.android.material:material:1.2.1'  
 implementation 'androidx.constraintlayout:constraintlayout:2.0.1'  
 testImplementation 'junit:junit:4.+'  
 androidTestImplementation 'androidx.test.ext:junit:1.1.2'  
 androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'  
 **implementation('com.squareup.okhttp3:okhttp:3.11.0')  
 implementation files('libs/uetsdk.jar')**  
**}**

### Interface specification

To complete the data, we need to send tagid, event action, event value, pagetype, product id, category id, search term, purchased product details,revenue, and so on. You can send these info to Bing Ads server by calling the UET SDK. There is a class *LogSendingManager* in the UET SDK. Fire the event by the inteface *sendEvent*.

public void sendEvent(Map<String, String> requiredParam, Map<String, String> optionalParam)

Firstly, you should create a *logSendingManager*. Set the android context to it. Example as below.

LogSendingManager logSendingManager = new LogSendingManager();  
logSendingManager.setContext(MainActivity.this);

Call the interface using the parameter of your case.

requiredParam: the mandatory parameter in the call. The below are the key in the map and its value meaning list.

String *TAG\_ID*：the tag id you used for the event track which is provided by

the UET.

String *EVENT\_*ACTION: the customer action in the context. Values as below:

ADD\_PAYMENT\_INFO: the customer adds the payment info for

a product.

ADD\_TO\_CART: the customer adds a product to the cart.

ADD\_TO\_WISHLIST: the customer adds a product into the

wish list.

PURCHASE: the customer adds purchase a product.

BEGIN\_CHECKOUT: the customer to checkout.

VIEW\_ITEM: the customer views a product.

VIEW\_ITEM\_LIST: the customer views a list of products.

VIEW\_SEARCH\_RESULTS: the customer views search result.

… and so on

String *EVENT\_VALUE*: Event value (custom value passed by advertiser)

String *APP\_ID*: the APP ID for the APP who call the SDK. It is allocated by

the Bing Ads server.

e.g.

HashMap<String, String> rParam = new HashMap<>();  
rParam.put(ParameterConf.*TAG\_ID*, "BINGADSTESTTAG123");  
rParam.put(ParameterConf.*EVENT\_ACTION*, MicrosoftAnalytics.Event.*ADD\_TO\_CART*);  
rParam.put(ParameterConf.*APP\_ID*, "dddcacd1a081e657ec965800d8687eeb");

rParam.put(ParameterConf.*EVENT\_VALUE*, "120");

OptionalParam: the parameter in you want to send. Frequently used as below.

String *WEB\_MUID*:MUID from web cookie ID  
String *WEB\_ANID* : ANID from web cookie ID

String *CUSTOMER\_USER\_ID*: the unique value of a user in the advertiser system.   
String *MID*: GUID generated by UET tag.

String *EVENT\_CATEGORY*:advertiser can use events category for conversion

tracking or remarketing.

String *EVENT\_LABEL*: advertiser’s value of a event label.  
String *VARIABLE\_REVENUE*：advertiser’s value to track variable revenue.  
String *VARIABLE\_REVENUE\_CURRENCY*：currency to track variable revenue.

String *KEYWORD*：the keyword used by the user to search.

String *SCREEN\_HEIGHT*：device screen height.  
String *SCREEN\_WIDTH*：device screen width.  
String *DEVICE\_OS\_VERSION*：device OS version.  
String *DEVICE\_TYPE*: device type.  
String *DEVICE\_MODEL*: device model.

String *BRAND*: device brand.

String *CPU\_ABI*: device CPU brand.

String *OS\_TIME*: device OS time.

String *TIME\_ZONE*: device OS time zone.

String *LANGUAGE*: device OS language.

String *COUNTRY*: device OS country.

String *ECOMM\_TOTALVALUE*: total value of a event such as purchase, view item,

and so on.

String *PAGETYPE*: the allowed values are – home, searchresults, category,

product, cart, purchase, other.

String *COUPON*: customer’s coupon info.

String *TAX*: tax of a product.

String *SHIPPING*: shipping info of a product.

String *TRANSACTION\_ID*: transaction id in the advertiser’s system.

String *SEARCH\_TERM*: the search term customer used.

String *TRAVEL\_ORIGINID*: customer’s travel origin id.

String *TRAVEL\_DESTID*: customer’s travel destination id.

String *TRAVEL\_STARTDATE*: customer’s travel start date.

String *TRAVEL\_ENDDATE*: customer’s travel end date.

String *TRAVEL\_TOTALVALUE*: customer’s travel total value.

String *CONTENT\_TYPE*: customer’s content type for the action begin\_checkout

and view\_items.

String *LOCATION\_ID*: customer’s event location id.

e.g.

HashMap<String, String> oParam = new HashMap<>();  
oParam.put(ParameterConf.*CUSTOMER\_USER\_ID*, "MSBingAds001");  
oParam.put(ParameterConf.*MID*, "a17afa22-df45-4360-8ba6-6092aab02058");  
oParam.put(ParameterConf.*EVENT\_CATEGORY*, "cart");  
oParam.put(ParameterConf.*EVENT\_VALUE*, "120");  
oParam.put(ParameterConf.*VARIABLE\_REVENUE\_CURRENCY*, "USD");  
oParam.put(MicrosoftAnalytics.Param.*PAGETYPE*, ParameterConf.*PAGE\_TYPE\_PURCHASE*);

complicated param: the parameter with many description.

String *ITEMS*: item typically is a list of products used for ecommerce

events. Advertiser used several properties of them to describe

a product. Items is a array of item with specified format.

brand: Brand of the item

category: Item category

creative\_name: Name of a creative used

creative\_slot: Name of the creative slot

id: Unique ID/SKU/Item group id for the item

location\_id: Location of the item

name: Item name

price: Purchase price of the item

quantity: Item quantity

Each item is packed with properties “&” delimited, then the

whole thing is packed again with “,” delimited.

{"id": "P12345", "price":"30", "quantity":"2"} ->

id=P12345&price=30&quantity=2

{"id": "P23456", "price":"60"} -> id=P23456&price=60

Items value: "id=P12345&price=30&quantity=2,id=P23456&price=60"

Note: Do not use “&” and “,” in your single value. Use “and”

and “comma” instead.

String *PRODID*: Unique ID/SKU/Item group id for the product. PRODID can be an

array, elements are “,” delimited.

e.g. There is an event for the add to cart with two products A and B. There parameter may as bellow.

LogSendingManager logSendingManager = new LogSendingManager();  
logSendingManager.setContext(MainActivity.this);  
HashMap<String, String> rParam = new HashMap<>();  
rParam.put(ParameterConf.*TAG\_ID*, "HOOTESTTAG123");  
rParam.put(ParameterConf.*EVENT\_ACTION*, MicrosoftAnalytics.Event.*ADD\_TO\_CART*);  
rParam.put(ParameterConf.*EVENT\_VALUE*, "120");  
rParam.put(ParameterConf.*APP\_ID*, "dddcacd1a081e657ec965800d8687eeb");  
  
HashMap<String, String> oParam = new HashMap<>();  
oParam.put(ParameterConf.*CUSTOMER\_USER\_ID*, "MSBingAds001");  
oParam.put(ParameterConf.*MID*, "a17afa22-df45-4360-8ba6-6092aab02058");  
oParam.put(ParameterConf.*EVENT\_CATEGORY*, "cart");  
oParam.put(ParameterConf.*VARIABLE\_REVENUE*, "120");  
oParam.put(ParameterConf.*VARIABLE\_REVENUE\_CURRENCY*, "USD");  
oParam.put(MicrosoftAnalytics.Param.*PAGETYPE*, ParameterConf.*PAGE\_TYPE\_PURCHASE*);  
oParam.put(MicrosoftAnalytics.Param.*PRODID*, "A#1,B#2");  
oParam.put(ParameterConf.*ITEMS*, "id=P12345&price=30&quantity=2,id=P23456&price=60&quantity=1");  
logSendingManager.sendEvent(rParam, oParam);

Keys in the two parameter map can be found in class ***ParameterConf*** or ***MicrosoftAnalytics.Param***. If there is property or something else not in the class, you can add it in the optionalParam with the same format.

String newKey = “MyName”;

String newValue = “MyValue”;  
oParam.put(newKey, newValue);