

Tracking

Events

To use the additional parameters of the custom events for targeting purposes, they must be formatted as a valid json string. Use a timestamp to represent a date.

If you want to track a specific event :

```
TrackEvent(1000, "{ \"string-type-field\": \"value\", \"bool-type-field\": true, \"number-type-field\": 7201, \"date-type-field\": 1498233848 }")
```

Where 1000 is the eventId.

Custom Events

It is possible to build custom events and track them via the following method : `Accengage.TrackCustomEvent(long type, dictionary<string, object> parameters);`

CustomEvent type is supposed to be a number strictly higher than 1000 (long-typed).

CustomEvent parameters are designed to be a dictionary composed of key/value couples.

Keys are strings regardless of whether you are on iOS or Android

On the other hand, values type change depending on whether you are on iOS or Android platform.

Android

In Android we are expecting to pass value in the following types :

- Boolean
- Int32
- Int64
- float
- Double
- Java.Util.Date
- String

If you try to pass other kind of data, it will be treated as a String

```
Dictionary<string, object> parameters = new Dictionary<string, object>();  
parameters["keyBoolean"] = true;  
parameters["keyInteger"] = 23;  
parameters["keyDouble"] = 2.0;  
parameters["keyDate"] = new Java.Util.Date();  
parameters["keyString"] = "Test String";  
Accengage.TrackCustomEvent(1001, parameters);
```

iOS

In iOS we are expecting to pass value in the following types from the Foundation Framework :

- NSDate
- NSNumber

- NSString
- bool

If you try to pass other kind of data, it will be ignored.

```
Dictionary<string, object> parameters = new Dictionary<string, object>();
parameters.add("keyNSString", new NSString("foo bar"));
parameters.add("keyNSNumber", new NSNumber(2));
parameters.add("keyBool", true);
Accengage.TrackCustomEvent(1001, parameters);
```

Analytics

You can track specific events like "Add to Cart", "Purchase" and "Lead". Here is how to use each of these events.

Lead

```
TrackLead("lead label", "lead value");
```

Add to Cart

First of all you will need to create an Item using the AccengageItem class.

```
AccengageItem accengageItem = new
AccengageItem("idItem", "labelItem", "categoryItem", 19.50, "EUR", 1)
```

Currency should be a valid 3 letters [ISO4217](#) currency (EUR,USD,..) and the 2 last arguments are price and quantity

Now you just have to add it to the Cart.

```
TrackCart("cartId", accengageItem, "EUR");
```

Purchase

```
TrackPurchase("purchaseId", "EUR", 19.5, listItem);
```

Update Device Info

You can create a device profile for each device in order to qualify the profile (for example, registering whether the user is opt in for or out of some categories of notifications).

A device profile is a set of key/value that are uploaded to Accengage server. In order to update information about a device profile, add the following lines to your code:

```
UpdateDeviceInfo(listUdi);
```

Where listUdi is a List<KeyValuePair<string, string>> containign all the informations to update.

New API to Update Device Information

You have the ability to use update device information using the following method : public static void UpdateDeviceInformation(string action, string key, object value);

Action are string-typed. Possible values are "set", "delete", "increment" and "decrement". Using other values would be ineffective.

Android

In Android we are expecting to pass value in the following types :

- Boolean
- Int32
- Int64
- Double
- Java.Util.Date
- String

If you try to pass other kind of data, it will be treated as a String

```
Accengage.UpdateDeviceInformation("set", "firstname", "John");  
Accengage.UpdateDeviceInformation("set", "lastname", "Doe");  
Accengage.UpdateDeviceInformation("set", "age", 35);  
Accengage.UpdateDeviceInformation("increment", "number_of_visit", 1.0);  
Accengage.UpdateDeviceInformation("decrement", "article_left", 1.0);
```

You can only operate "increment" or "decrement" actions on suitable value Double in Android

iOS

In iOS we are expecting to pass value in the following types from the Foundation Framework :

- NSDate
- NSNumber
- NSString
- bool

If you try to pass other kind of data, it would be ignored.

```
Accengage.UpdateDeviceInformation("increment", "number", new NSNumber(3));  
Accengage.UpdateDeviceInformation("set", "text", new NSString("foo"));
```

You can only operate "increment" or "decrement" actions on suitable value NSNumber in iOS

Subscription Tag

Subscription tag is a new API used to mark user interest for a particular topic : public static void SetDeviceTag(string category , string identifier, Dictionary<string, object> parameters)

Android

In Android we are expecting to pass value in the following types :

- Boolean
- Int32
- Int64
- Java.Util.Date
- String

If you try to pass other kind of data, it will be treated as a String

```
Dictionary<string, object> parameters = new Dictionary<string, object>();
parameters["club_league_1"] = "corinthians";
parameters["club_league_2"] = "figueirense";
Accengage.SetDeviceTag("Soccer", "Brazil", parameters);
```

iOS

In iOS we are expecting to pass value in the following types from the Foundation Framework :

- NSDate
- NSNumber
- NSString
- bool

```
Dictionary<string, object> parameters = new Dictionary<string, object>();
parameters.add("club_league_1", new NSString("corinthians"));
parameters.add("club_league_2", new NSString("figueirense"));
Accengage.SetDeviceTag("Soccer", "Brazil", parameters);
```

A device tag can contain up to 5 parameters and all parameter key should be different.

As mentioned with the name of the method, setDeviceTag is an update of the Subscription Tag, it will override then the previous information.

You can set an expiration date for your Subscription Tag by adding the parameter "exp" to the Device Tag where the value is a timestamp for the expiration date :

```
parameters["exp"] = "1543512073";
```

You can unsubscribe a user to a particular topic with the following code :

```
Accengage.DeleteDeviceTag("Soccer", "Brazil");
```

Important

Please note that this feature is only available through Accengage's API

Views

You can identify each views of your application that the SDK can refer to using.

```
TrackScreenDisplay("MyView");
```

where MyView is the name of your view.

iOS

This method must take place in the viewDidAppear and is to be used with TrackScreenDismiss in the viewDidDisappear. Look the [Documentation](#).