

Tracking 1.1.x

Events

If you want to track a specific event you can add to your code:

```
BMA4S.trackEvent(1000, 'test')
```

Where 1000 is your eventID and "test" is an argument of this event. Please note that you can add multiple arguments.

Using Analytics

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

Lead

If you want to send a Lead, you can add to your code:

```
BMA4S.trackLead('Lead Label', 'Lead Value')
```

Add to Cart

If you want to track an Add to Cart event, add to your code:

```
BMA4S.trackAddToCart('CartId', 'ArticleID', 'Label', 'Category', 'Currency', 12.30, 1)
```

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

Purchase

If you want to track a Purchase event, you can add to your code:

```
BMA4S.trackPurchase('Purchase ID', 'Purchase Currency', 12.30)
```

Note: Currency should be a valid 3 letters [ISO4217](#) currency (EUR,USD,..) and the last argument is the total price of this purchase.

Update Device Information

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:

```
BMA4S.updateDeviceInfo('key', 'value')
```

Replace "key" and "value" with the segmentation key name and the value you want to set.

The keys and values must match Accengage user information field names and values to allow information to be correctly updated.

Date Format

You can send a date using the updateDeviceInfo method.

Your date has to be in the following format: "yyyy-MM-dd HH:mm:ss zzz"

Manually Update Geolocation (Android only)

Geolocation is automatically updated and sent to our servers if "**android_no_geoloc**" is **false** inside **BMA4SCordovaConfig.json**

If you want to manually update the current location of the device, you can call:

```
BMA4S.updateGeolocation(latitude, longitude, altitude, accuracy);  
//Example : BMA4S.updateGeolocation(40.12, 2.12, 20, 50)
```

Views

If you want to tag each Fragment/Tabs/Activity and be able to target it, you can use the following code as soon as a view is displayed:

```
BMA4S.setView( 'Dashboard' )
```

Where your-view is the name of your view.

States (Android only)

In Accengage Interface, you can "Declare a State" (Applications Advanced Parameters). States are like browser cookies which can be used to trigger Accengage In-App notifications. You can target one or more states and combine them with Event or ViewTag.

Example: When event 5000 is triggered and state "search" contains "pizz*", display a specific In-App.

The following section explains how to declare certain states with the Accengage SDK.

Put a State

You can obtain the name of the state from the Accengage "value" field in Applications Advanced Parameters States section.

In order to put a state, write this code:

```
BMA4S.putState( 'search', 'myValue' )
```

Where "search" is the state name and "myValue" is the value you want to put for this state.

Static Lists

First of all, to use a Static List, you need to create it on the Accengage User Interface.

Add the current device to a Static List

In order to add the current device to a Static List you have to provide a JSONArray containing JSONObjects including each a listId and an optional expireAt expiration date

```
BMA4S.subscribeToLists([{'listId':'sport'},{'listId':'politic',
expireAt:1516467777000}]);
```

Remove the current device from a Static List

In order to remove the current device to a Static List you have to provide a JSONArray containing JSONObjects including each a listId

```
BMA4S.unsubscribeFromLists([{'listId':'sport'}]);
```

Get subscription status

You have 2 ways to get the subscription status of the current device.

Get subscription status of selected lists

Retrieve the status of all wanted lists, you will need to pass a function in order to retrieve the result (here it's theCallback) :

```
BMA4S.getSubscriptionStatusForLists([{'listId':'sport'}], theCallback)
```

Get all subscriptions of the current device

Retrieve the status of all lists, you will need to pass a function in order to retrieve the result (here it's theCallback) :

```
BMA4S.getListOfSubscriptions(theCallback)
```

As a result, you will get a JSONArray containing some JSONObjects :

```
[{
  "listId":"sport",
  "name":"Sport list",
  "expireAt": 0, // timestamp in ms or 0 if no expiration date
  "status":"subscribed" //can be subscribed or unsubscribed
}]
```