

Mobile: What Are Android Campaign Callbacks and How Are They Used?

Brief Overview

As of WalkMe Mobile SDK version 1.6.0, the WalkMe Mobile SDK supports campaign callbacks.

Callbacks are a programmatic way for an app to register to WalkMe Campaign events. By implementing the Campaign Callbacks interface, apps can use the data transferred through the callback objects (including Campaign and user information) and utilize it for the app's purposes, such as passing that information to any analytics or CRM system unrelated to WalkMe, or updating any app or user settings according to user interaction with the Campaign.

Use Cases

Campaign callbacks can be used for the following and many more:

- Track user onboarding from any analytics system
- Generate opportunities in a CRM based on user engagement with WalkMe Shoutout - i.e. "Are you interested in a free demo? Yes / No"
- Integrate with existing support systems to track and compare WalkThru usage/completion against the top support ticket topics

Below is a description of all the required steps in order to implement campaign callbacks.

Steps To Implement Campaign Callbacks

To be able to register to campaign callbacks, first implement this interface:

```
/**
 * Interface definition for a callback to be invoked in Campaign actions.
 */
public interface WMCampaignInfoListener {
    /**
     * Called after campaign was dismissed.
     *
     * @param campaignInfo The dismissed campaign info.
     */
    void onCampaignDismissed(WMCampaignInfo campaignInfo);
}
```

```
}
```

After implementing the interface, use the `setCampaignInfoListener` method to register to the campaign callback events:

```
/**
 * Register a listener to campaign events
 *
 * @param campaignInfoListener The listener
 */
public static void setCampaignInfoListener(WMCampaignInfoListener
campaignInfoListener)
```

The callback return object will be of class `WMCampaignInfo`, which includes the following information:

Item	Retrieved By	Description
Campaign CTA	<code>getCampaignCta()</code>	The campaign CTA that the app user has interacted with. For all CTAs other than the custom one, the CTA returned will be: <code>abbi://cta</code> , for example: <code>abbi://ok</code> or <code>abbi://never</code>
Campaign ID	<code>getCampaignId()</code>	The unique ID assigned to the campaign
Campaign CTA ID	<code>getCampaignCtaId()</code>	The unique ID for the Campaign CTA
User Data	<code>getUserData()</code>	An object holding information on the user who interacted with the campaign. See full description of the object below.
Campaign Data	<code>getCampaignData()</code>	A placeholder object added for possible future use of additional data appended to the campaign. Survey submission data is populated here.

This is the structure for the `WMCampaignInfo` object:

```
public class WMCampaignInfo {

    public String getCampaignCta()
```

```

    public String getCampaignId()

    public String getCampaignCtaId()

    public WMUserData getUserData()

    public Map<String, Object> getCampaignData()
}

```

As mentioned above, `WMCampaignInfo` includes the object `WMUserData`, which is described below:

Item	Retreived By	Description
Public User Attributes	<code>getUserAttributesMap()</code>	All of the public user attributes that were set for the app user by the moment the user interacted with the campaign. The list will always include the attribute key and its value, e.g. { "lead_id" : "12345" , "account_type" : "family" }
Private User Attributes	<code>getPrivateUserAttributes()</code>	All of the private user attributes that were set for the app user by the moment the user interacted with the campaign. The list will always include the attribute key and its value, e.g. { "lead_id" : "12345" , "account_type" : "family" }
User Creation Timestamp	<code>getUserCreationTimestamp()</code>	The timestamp at which the user was first identified by WalkMe
Session Duration	<code>getSessionDuration()</code>	How long was the user session at the moment the user made the campaign interaction
Android Version	<code>getDeviceVersion()</code>	The user's OS version
Device Unique ID	<code>getDeviceId()</code>	The device unique ID
Device Model	<code>getDeviceModel()</code>	The user's device model
Device Orientation	<code>getDeviceOrientation()</code>	The device orientation at the moment the user interacted with the campaign

App Version	getAppVersion()	The app version the user is using
App Name	getAppName()	The app name
Locale	getLocale()	The user's device locale
SDK Version	getSdkVer()	The SDK version integrated with the app the user is using
Session ID	getSessionId()	The unique session ID generated by WalkMe
New User Indication	getIsNewUser()	Is this the first time the user is identified by WalkMe (true / false)
Push Notifications Status	getIsPushEnabled()	Has the user enabled push notification for the app (true / false). If the app does not feature push notifications – the value will be "false".
Device Timezone	getTimezone()	The user timezone as set on the device.
Network	getNetwork()	The network type the user is using: WiFi / Cellular / Offline
User Latitude	getLocationLat()	Placeholder for possible implementation of location latitude
User Longitude	getLocationLong()	Placeholder for possible implementation of location longitude
System Name	getSystemName()	The user's OS name – in this case will always be "Android"
Current user timestamp	getTimestamp()	The user's current timestamp

This is the structure for the `WMUserData` object:

```
public class WMUserData {

    /**
     * User Attributes in current session.
     */
    public Map<String, Object> getUserAttributesMap()
```

```
/**
 * Private User Attributes in current session.
 */
public Map<String, Object> getPrivateUserAttributes()

/**
 * Returns the User Creation Timestamp for this User.
 *
 * @return long Timestamp.
 */
public long getUserCreationTimestamp()

/**
 * Returns the User Session Duration in second.
 *
 * @return double second.
 */
public double getSessionDuration()

/**
 * Returns the Device Android version.
 *
 * @return String version number.
 */
public String getDeviceVersion()

/**
 * Returns the Device unique id.
 *
 * @return String id.
 */
public String getDeviceId()

/**
 * Returns the Device brand Model.
 *
 * @return String Model.
 */
public String getDeviceModel()

/**
 * Returns the Device Orientation PRT_REG / LSL.
 *
```

```
    * @return String Orientation.
    */
    public String getDeviceOrientation()
    /**
    * Returns the application version.
    *
    * @return String version.
    */
    public String getAppVersion()

    /**
    * Returns the application Name.
    *
    * @return String Name.
    */
    public String getAppName()

    /**
    * Returns the locale language for this instance of the Java Virtual
    Machine.
    *
    * @return String Name.
    */
    public String getLocale()

    /**
    * Returns the walkme sdk version.
    *
    * @return String version.
    */
    public String getSdkVer()

    /**
    * Returns the Session unique id.
    *
    * @return String id.
    */
    public String getSessionId()

    /**
    * Returns if the user use the app for the first time.
    *
    * @return String boolean.
```

```
*/
public String getIsNewUser()

/**
 * Returns if Push notification approved for user.
 *
 * @return String boolean.
 */
public String getIsPushEnabled()

/**
 * Returns user device timezone.
 *
 * @return String timezone.
 */
public String getTimezone()

/**
 * Returns user current network ( WIFI / 3G )
 *
 * @return String network.
 */
public String getNetwork()

/**
 * Returns user current location latitude if available
 *
 * @return String latitude.
 */
public String getLocationLat()

/**
 * Returns user current location longitude if available
 *
 * @return String longitude.
 */
public String getLocationLong()

/**
 * Returns user System Name (Android).
 *
 * @return String Android.
 */
```

```
public String getSystemName()

/**
 * Returns the Current Timestamp for this User.
 *
 * @return String Timestamp.
 */
public String getTimestamp()
}
```