

# Mobile : quels sont les rappels de campagne Android et comment les utiliser ?

## Aperçu général

**À partir du SDK WalkMe Mobile version 1.6.0**, le SDK WalkMe Mobile prend en charge les rappels de campagne.

Les rappels sont une façon programmatique pour une application de s'inscrire aux événements de la campagne WalkMe. En mettant en œuvre l'interface Rappel de campagne, les applications peuvent utiliser les données transférées via les objets de rappel (y compris le type de campagne et les informations de l'utilisateur) et les utiliser aux fins de l'application, telles que la transmission de ces informations à tout système analytique ou CRM non lié à WalkMe, ou la mise à jour de toute application ou paramètres utilisateur en fonction de l'interaction de l'utilisateur avec la campagne.

## Cas d'utilisation

Les rappels de campagne peuvent être utilisés pour les raisons suivantes ainsi que beaucoup d'autres :

- Suivez l'intégration des utilisateurs à partir de n'importe quel système d'analyse
- Générez des opportunités dans un CRM en fonction de l'engagement des utilisateurs avec WalkMe Shoutout, comme « Est-ce que vous souhaitez déverrouiller une démo gratuite ? Oui/Non »
- Intégrez aux systèmes de support existants pour suivre et comparer l'utilisation/achèvement WalkThru avec les principaux sujets de ticket d'assistance

Vous trouverez ci-dessous une description de toutes les étapes requises pour mettre en œuvre les rappels de la campagne.

## Étapes pour mettre en œuvre les rappels de campagne

Pour être en mesure de s'inscrire aux rappels campagnes, mettez d'abord en œuvre cette 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);
}

```

Après avoir mis en œuvre l'interface, utilisez la méthode `setCampaignInfoListener` pour vous inscrire aux événements de rappel de campagne :

```

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

```

L'objet de rappel sera de classe `WMCampaignInfo`, qui comprend les informations suivantes :

Élément	Récupéré par	Description
CTA de campagne	<code>getCampaignCta()</code>	Le CTA de campagne avec lequel l'utilisateur de l'application a interagi. Pour tous les CTA autres que la personnalisation, le CTA retourné sera : <code>abbi://cta</code> , par exemple : <b><code>abbi://ok</code></b> ou <b><code>abbi://never</code></b>
ID de campagne	<code>getCampaignId()</code>	L'ID unique affecté à la campagne
ID du CTA de campagne	<code>getCampaignCtald()</code>	L'ID unique pour le CTA de campagne
User Data (données d'utilisateur)	<code>getUserData()</code>	Un objet contenant des informations sur l'utilisateur qui a interagi avec la campagne. Voir la description complète de l'objet ci-dessous.

Données de campagne	getCampaignData()	Un objet d'espace réservé ajouté pour l'utilisation future possible des données supplémentaires annexées à la campagne. Les données de soumission de Survey sont renseignées ici.
---------------------	-------------------	---

Ceci est la structure de l'objet WMCampaignInfo :

```
public class WMCampaignInfo {
    public String getCampaignCta()
    public String getCampaignId()
    public String getCampaignCtaId()
    public WMUserData getUserData()
    public Map<String, Object> getCampaignData()
}
```

Comme mentionné plus haut, WMCampaignInfo comprend l'objet WMUserData, qui est décrit ci-dessous :

Élément	Récupéré par	DESCRIPTION
Attributs des utilisateurs publics	getUserAttributesMap()	Tous les attributs de l'utilisateur public qui ont été définis pour l'utilisateur de l'application au moment où l'utilisateur a interagi avec la campagne. La liste comprendra toujours la clé de l'attribut et sa valeur, par exemple { « lead_id » : « 12345 » , « account_type » : « family » }

Attributs des utilisateurs privés	getPrivateUserAttributes()	Tous les attributs de l'utilisateur privé qui ont été définis pour l'utilisateur de l'application au moment où l'utilisateur a interagi avec la campagne. La liste comprendra toujours la clé de l'attribut et sa valeur, par exemple { « lead_id » : « 12345 » , « account_type » : « family » }
Horodatage de création de l'utilisateur	getUserCreationTimestamp()	L'horodatage auquel l'utilisateur a été identifié pour la première fois par WalkMe
Durée de la session	getSessionDuration()	Quelle était la durée de la session de l'utilisateur au moment où il a effectué l'interaction avec la campagne
Version Android	getDeviceVersion()	Version du système d'exploitation de l'utilisateur
ID unique de l'appareil	getDeviceId()	L'ID unique de l'appareil
Modèle de l'appareil	getDeviceModel()	Modèle de l'appareil de l'utilisateur
Orientation de l'appareil	getDeviceOrientation()	L'orientation de l'appareil au moment où l'utilisateur a interagi avec la campagne
Version de l'appli	getAppVersion()	La version de l'application utilisée par l'utilisateur
Nom de l'application	getAppName()	Le nom de l'application
Langue	getLocale()	La langue de l'appareil de l'utilisateur
Version du SDK	getSdkVer()	La version du SDK intégrée à l'application que l'utilisateur utilise
ID de la session	getSessionId()	L'ID de la session unique généré par WalkMe

Indication de nouvel utilisateur	getIsNewUser()	Est-ce la première fois que l'utilisateur est identifié par WalkMe (vrai / faux)
État des notifications push	getIsPushEnabled()	L'utilisateur a-t-il activé la notification push pour l'application (vrai / faux). Si l'application ne comporte pas de notifications push, la valeur sera « faux ».
Fuseau horaire de l'appareil	getTimezone()	Le fuseau horaire de l'utilisateur tel que défini sur l'appareil.
Réseau	getNetwork()	Le type de réseau que l'utilisateur utilise : WiFi / Cellulaire / hors ligne
Latitude de l'utilisateur	getLocationLat()	L'espace réservé pour la mise en œuvre possible de la latitude de l'emplacement
Longitude de l'utilisateur	getLocationLong()	L'espace réservé pour la mise en œuvre possible de la longitude de l'emplacement
Nom du système	getSystemName()	Le nom de l'OS de l'utilisateur : dans ce cas, sera toujours « Android »
Horodatage de l'utilisateur actuel	getTimestamp()	L'horodatage actuel de l'utilisateur

Ceci est la structure de l'objet WMUserData :

```
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()  
}
```