

Mobile : que sont les rappels de campagne iOS et comment sont-ils utilisés ?

Aperçu général

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

Utilisation des rappels de campagne

Pour être en mesure de s'inscrire aux campagnes de rappel, établissez d'abord ce protocole :

```
/**  
 * Interface definition for a callback to be invoked in Campaign actions.  
 */  
@protocol WMCampaignInfoDelegate <NSObject>
```

```
/**
 * Called after campaign was dismissed.
 *
 * @param campaignInfo The dismissed campaign info.
 */
- (void)campaignDidDismiss:(WMCampaignInfo *)campaignInfo;

@end
```

Après avoir mis en œuvre le protocole, utilisez la méthode `setCampaignInfoDelegate` pour vous inscrire aux événements de rappel de campagne :

```
/**
 * Register a delegate to campaign events
 *
 * @param delegate The delegate
 *
 */
+ (void)setCampaignInfoDelegate:(id<WMCampaignInfoDelegate>)delegate;
```

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

Élément	Récupéré par	Description
CTA de campagne	campaignCta	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 : abbi://cta , par exemple : abbi://ok ou abbi://never
ID de campagne	campaignId	L'ID unique affecté à la campagne
ID du CTA de campagne	campaignCtald	L'ID unique pour le CTA de campagne
User Data (données d'utilisateur)	userData	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	campaignData	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.
---------------------	--------------	---

```
@interface WMCampaignInfo : NSObject
```

```
@property (nonatomic,strong) NSString* campaginCta;
@property (nonatomic,strong) NSString* campaignCtaId;
@property (nonatomic,strong) NSString* campaignId;
@property (nonatomic,strong) WMUserData* userData;
@property (nonatomic,strong) NSDictionary* campaignData;
```

```
@end
```

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

Élément	Récupéré par	DESCRIPTION
Attributs des utilisateurs publics	*userAttributes	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	*privateUserAttributes	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 »}

Durée de la session	sessionDuration	Quelle était la durée de la session de l'utilisateur au moment où il a effectué l'interaction avec la campagne
Version iOS	systemVersion	Version du système d'exploitation de l'utilisateur
ID unique de l'appareil	deviceId	L'ID unique de l'appareil
Modèle de l'appareil	deviceModel	Modèle de l'appareil de l'utilisateur
Orientation de l'appareil	deviceOrientation	L'orientation de l'appareil au moment où l'utilisateur a interagi avec la campagne
Version de l'appli	appVersion	La version de l'application utilisée par l'utilisateur
Nom de l'application	appName	Le nom de l'application
Langue	locale	La langue de l'appareil de l'utilisateur
Version du SDK	sdkVer	La version du SDK intégrée à l'application que l'utilisateur utilise
ID de la session	sessionId	L'ID de la session unique généré par WalkMe
État des notifications push	isPushEnabled	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	timezone	Le fuseau horaire de l'utilisateur tel que défini sur l'appareil.
Réseau	network	Le type de réseau que l'utilisateur utilise : WiFi / Cellulaire / hors ligne

Nom du système	systemName	Le nom du système d'exploitation de l'utilisateur – dans ce cas sera toujours « iOS ».
Horodatage de l'utilisateur actuel	timestamp	L'horodatage actuel de l'utilisateur

Ceci est la structure de l'objet WMUserData :

```
@interface WMUserData : NSObject

/**
 * User Attributes in current session.
 */
@property (nonatomic,strong) NSDictionary *userAttributes;

/**
 * Private User Attributes in current session.
 */
@property (nonatomic,strong) NSDictionary *privateUserAttributes;

/**
 * User Session Duration in second.
 */
@property (nonatomic,strong) NSNumber* sessionDuration;

/**
 * User's iOS version
 */
@property (nonatomic,strong) NSString* systemVersion;

/**
 * Device Unique ID
 */
@property (nonatomic,strong) NSString* deviceId;

/**
 * Device Model.
 */
@property (nonatomic,strong) NSString* deviceModel;

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

```
*/
@property (nonatomic,strong) NSString* deviceOrientation;

/**
 * The application version.
 */
@property (nonatomic,strong) NSString* appVersion;

/**
 * The application name.
 */
@property (nonatomic,strong) NSString* appName;

/**
 * Locale language for user's device.
 */
@property (nonatomic,strong) NSString* locale;

/**
 * WalkMe SDK version.
 */
@property (nonatomic,strong) NSString* sdkVer;

/**
 * Session unique id.
 */
@property (nonatomic,strong) NSString* sessionId;

/**
 * If Push notification approved for user.
 */
@property (nonatomic,strong) NSString* isPushEnabled;

/**
 * User device timezone.
 */
@property (nonatomic,strong) NSString* timezone;

/**
 * User current network ( WIFI / 3G ).
 */
@property (nonatomic,strong) NSString* network;
```

```
/**
 * User OS System Name.
 */
@property (nonatomic,strong) NSString* systemName;

/**
 * Current Timestamp for this User.
 */
@property (nonatomic,strong) NSString* timestamp;

@end
```