

モバイル：アクションステップ

概要

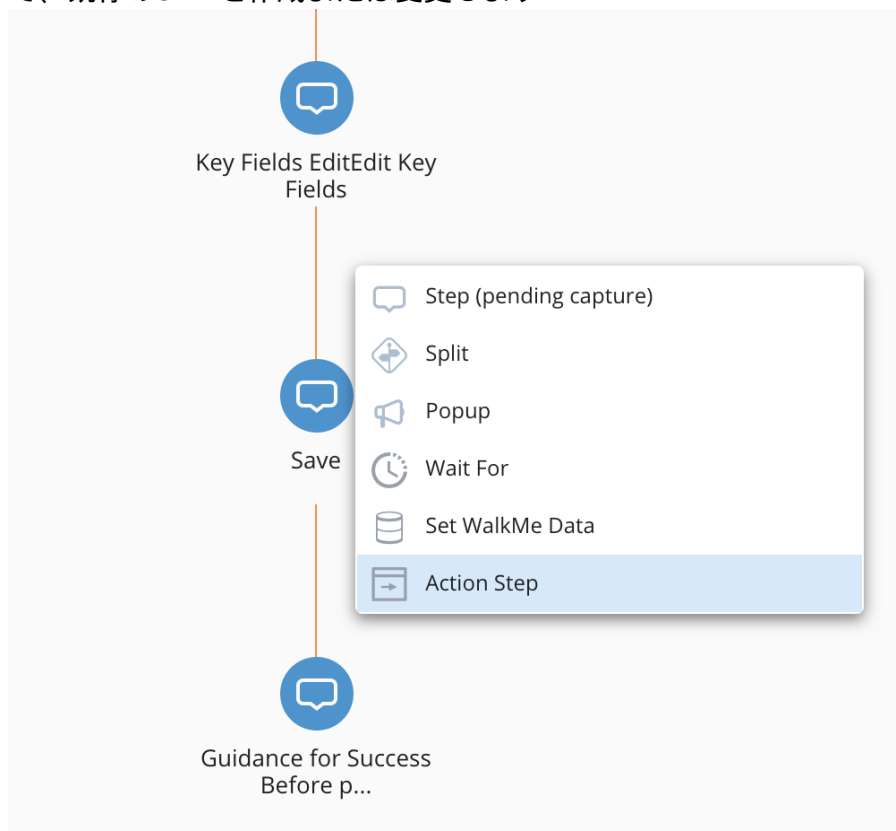
アクションステップは、自動化のユースケースに非常に便利なステップタイプです。エンドユーザーに代わってアクションを実行する機能により、アプリユーザーの生産性とTime to Value（価値実現までの時間）が向上します。以前WalkMe Mobileは、バルーンとバナーのステップに加え、自動タップのステップをサポートしていました。最初に追加されたアクションステップタイプは「スクロール」で、今後はさらにアクションタイプが追加される予定です。

ユースケース

- 特定の要素の「タイプ」に自動スクロール

アクションステップを追加する手順

1. Mobileコンソールで、+にカーソルを合わせ[Action Step（アクションステップ）]をクリックして、既存のSWTを作成または変更します



2. [Type（タイプ）][Sub Type（サブタイプ）][Value（値）]を選択します

Type:

Scroll

Sub Type :

Callback

Value :

CANCEL

OK

タイプ	サブタイプ	値
スクロール	コールバック	エレメントのタイプ (UILabelなど)

アクションステップを有効にする方法

以下は、アクションステップのコールバックを実装するために必要なすべての手順の説明です。

キャンペーンのコールバックに登録するには、まずこのプロトコルを実装します。

- iOS

```
/**
 * 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.
 */
@optional
- (void)campaignDidDismiss:(WMCampaignInfo *)campaignInfo;
/**
 * Called right before the campaign is about to be shown (will not be
 * called on Power Mode preview)
 *
 * @param campaignInfo The shown campaign info.
 */
@optional
- (void)campaignWillShow:(WMCampaignInfo *)campaignInfo;
/**
 * Return a value for SWT Action step of type "Callback"
 *
 * @param type the action type (ex. scroll).
 * @param campaign The shown campaign info.
 * @param args array of values from the Value field on the console
 * (comma separated values).
 */
@optional
- (void)campaign:(WMCampaignInfo *)campaign didPerformAction:(NSString
*)actionType withArgs:(NSArray<NSString *> *)args andCompletion:(void
(^)(__nullable id result))completion;
@end
```

- Android

```
public interface WMCampaignInfoListener {  
    /**  
     * Called right before the campaign is about to be shown (will not be  
     * called on Power Mode preview)  
     *  
     * @param campaignInfo The shown campaign info.  
     */  
    void onCampaignPresented(WMCampaignInfo campaignInfo);  
    /**  
     * Called after campaign was dismissed.  
     *  
     * @param campaignInfo The dismissed campaign info.  
     */  
    void onCampaignDismissed(WMCampaignInfo campaignInfo);  
    /**  
     * Called on action step  
     *  
     * @param campaignInfo The campaign info.  
     * @param actionType The action type.  
     * @param args String array of arguments,  
     * @param completionStepListener The listener to receive the result.  
     */  
    void onCampaignAction(WMCampaignInfo campaignInfo, String actionType,  
        String[] args, WMCampaignActionListener completionStepListener);  
}
```

プロトコルを実装した後、キャンペーンのコールバックイベントに登録します。

- iOS

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

- Android

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

最後に、アクションに必要な値を返すために、キャンペーンアクションメソッドを実装します。

- iOS

```
- (void)campaign:(WMCampaignInfo *)campaign didPerformAction:(NSString *)actionType withArgs:(NSArray<NSString *> *)args andCompletion:(void (^)(id _Nullable))completion {
    if (args.firstObject) {
        NSLog(@"Value for action type: %@ with args: %@", type, args.firstObject); // Value for action type: scroll with args: ProductCardTypeClass
    }
    NSIndexPath *indexPath = [NSIndexPath indexPathForRow:4 inSection:1];
    if (completion) {
        completion(indexPath);
    }
}
```

- Android

```
@Override
public void onCampaignAction(WMCampaignInfo campaignInfo, String actionType, String[] args, ABBI.WMCampaignActionListener completionStepListener) {
    if (args != null && args.length > 0) {
        Integer index = null;
        // Calculate the index based on the provided args
        index = myCardIndexCalculator();
        if (completionStepListener != null) {
            completionStepListener.onActionCompleted(index);
        }
    }
    else {
        if (completionStepListener != null) {
            completionStepListener.onActionCompleted(null);
        }
    }
}
```

要件 / 制限

- iOS SDKバージョン2.9.7およびAndroid SDKバージョン2.10.9
- 通常のキャプチャステップは、「スクロール」アクションステップの直後に続く必要があります。これにより、エレメントが存在する適切なコンテナにスクロールするためのスクロールアクションコンテキストが提供されます。