Mobile: Action Steps

Brief Overview

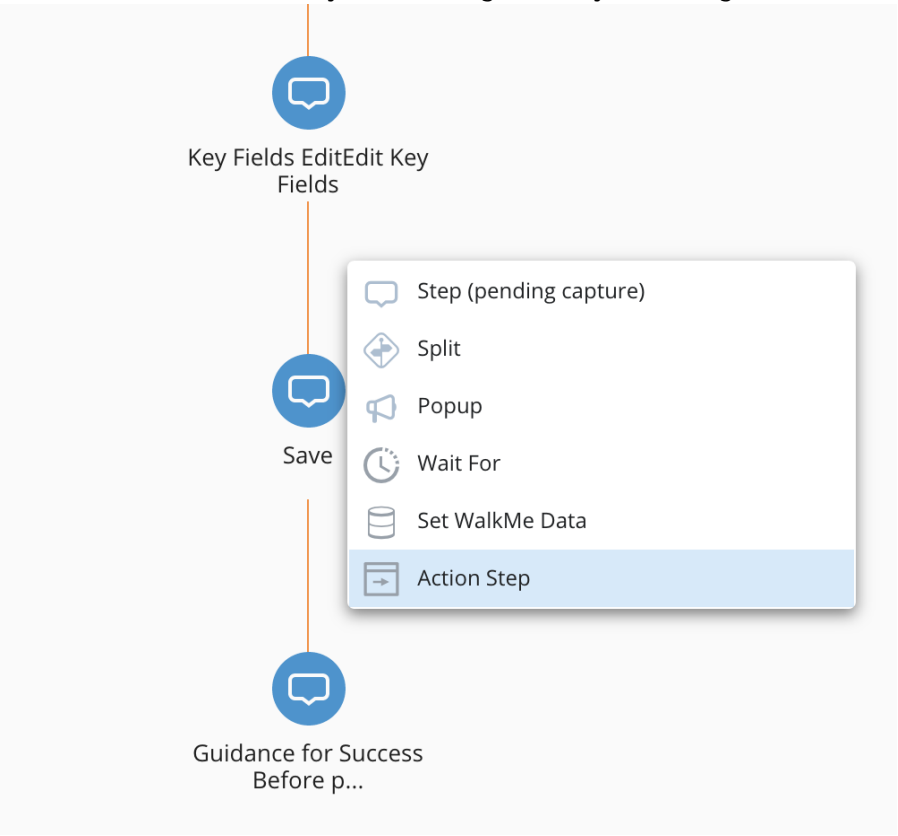
Action Steps are a step type that are very useful for automation use cases. The ability to take action on behalf of end-users improves productivity and time to value for app users. Previously, WalkMe Mobile supported balloon and banner steps as well as auto-tap steps. The first Action Step type added is “Scroll” and in the future, more action types will be added.

Use Cases

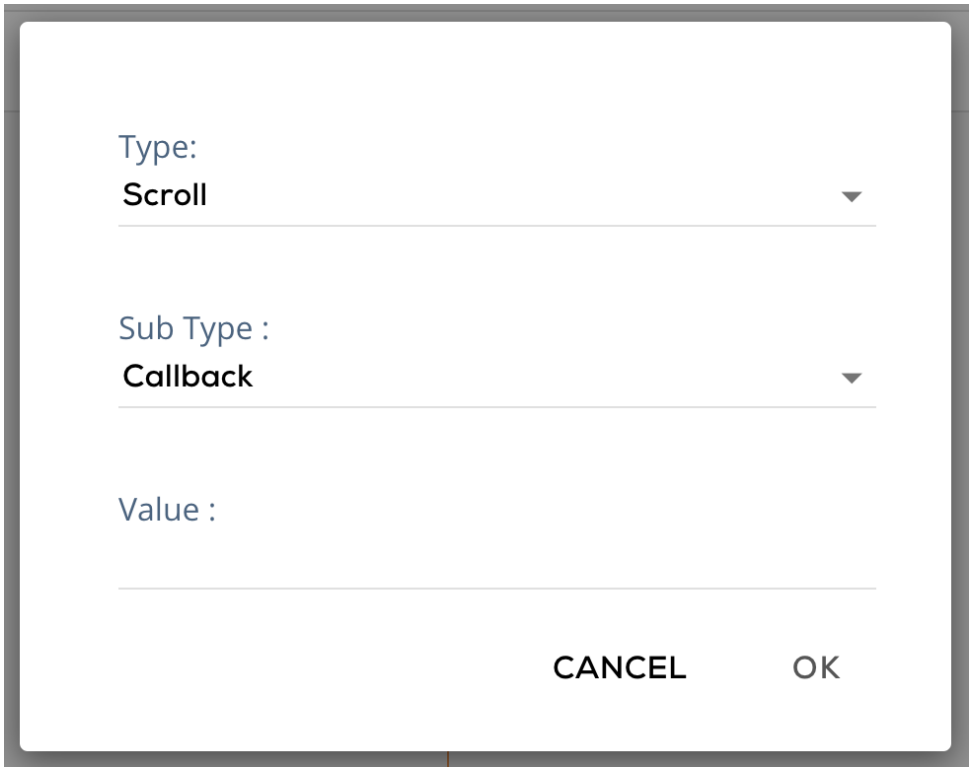
- Auto-scroll to a specific element “type”

Steps to add Action Step

1. In the Mobile Console, create or modify an existing SWT by hovering over the + and clicking “Action Step”



2. Select “Type”, “Sub Type” and “Value”



Type	Sub Type	Value
Scroll	Callback	Type of element (i.e. UILabel)

How to Enable Action Steps

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

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

- 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);
}
```

After implementing the protocol, register to the campaign callback events:

- 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)
```

Lastly, implement the campaign action method in order to return the desired value for the action:

- 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);
        }
    }
}
```

Requirements/Limitations

- iOS SDK version 2.9.7 and Android SDK version 2.10.9
- Regular captured step must follow immediately after the “scroll” action step. This provides the scroll action context to scroll to the appropriate container where the element exists.