


Mobile: How To Add User Attributes

To create User Attributes, you must both add them to the Mobile Console and your developers must use the WalkMe Mobile SDK API to set them up in your app.

Adding User Attributes in the Mobile Console

This step is mandatory for all User Attributes. Add the Attributes reference in the Console to use Attributes in segments and Rule-Based Goals.

- In the Mobile Console, click *SETTINGS > App Settings > USER ATTRIBUTES*.
- Click *NEW ATTRIBUTE*.
- Set the Attribute name and its privacy level:



- - If this is a Codeless User Attribute - make sure the name you set matches the key on the app storage (see below)
 - If this is an API User Attribute - Make sure to provide the Attribute name and type to your developers so they can configure matching ones in the API call.

Private Attributes

To protect your end-users' privacy and your app's sensitive data, you can set a User Attribute as private. This setting prevents WalkMe from storing the private User Attribute data, and this data will not be taken into account when WalkMe Mobile AI provides predictions about your end-users'

behavior, nor will it be saved for analytics purposes.

Codeless User Attributes

Starting from version 1.18.0, the WalkMe SDK is able to fetch data from the device's SharedPreferences (Android) or UserDefaults (iOS), and store it in User Attributes.

This means these User Attributes can be set without calling the WalkMe SDK API, making setting up these attributes a lot quicker and frictionless. All you need to do is just check with your developers if the values you want to use already exist in the SharedPreferences / UserDefaults.

Codeless Attributes in iOS

In iOS, WalkMe can easily access any key set on the UserDefaults. In the console, simply set the attribute with the same name as the UserDefaults key.

Codeless Attributes in Android

In Android, the SharedPreferences is not a single pool of variables, but instead broken down into different sections (each section is called SharedPreferences).

In the console, to set up a User Attribute by one of the SharedPreferences keys, it needs to be referenced according to this convention: [SharePreferences section name].[key name] , *for example*: telcosp.username .

Steps For Adding User Attributes in the WalkMe Mobile SDK

iOS

Your developers should implement the following code:

```
/** Regular User Attributes */
+(void)setUserAttributeWithKey:(NSString *)key andValue:(id)value;
+(void)setUserAttributes:(NSDictionary<NSString*,id> *)attributes;

/** Private User Attributes */
+(void)setPrivateUserAttributeWithKey:(NSString *)key andValue:(id)value;
+(void)setPrivateUserAttributes:(NSDictionary<NSString*,id> *)attributes;

/** Clearing Private User Attributes (if needed) */
+(void)clearPrivateUserAttributes;
```

Android

Your developers should implement the following code:

```
/** Regular (non private) User Attributes */  
public static void setUserAttribute(String key, Object value)  
public static void setUserAttributes(Map<String,Object> attributes)  
  
/** Private User Attributes */  
public static void setPrivateUserAttribute(String key, Object value)  
public static void setPrivateUserAttributes(Map<String,Object> attributes)  
  
/** Clearing Private User Attributes (if needed) */  
public static void clearPrivateUserAttributes()
```

Multi Value User Attributes

The same API used for setting a single value user attribute can also be used for setting a User Attribute that represents an array.

iOS multi value user attribute example

```
[ABBI setUserAttributeWithKey:@"numbers" andValue:@[@"1", @"2"]];
```

Android multi value user attribute example

```
ABBI.setUserAttribute("numbers", new ArrayList<>(Arrays.asList("1", "2",  
"3")));
```

Testing User Attributes

To test that a User Attribute was added successfully, you can check that it appears in the “User Attributes” section available in the [App Info screen](#) in Power Mode.