

Let Users Share Your App

- [Step 1 - Add a button to start share](#)
- [Step 2 - Add a Custom JS Action to the button](#)
- [Step 3 - Recording Referrers into new user's data](#)
- [Alternative Route](#)

We will explain here the fullest solution, where you also track which users do share your app, and which particular new user came from which referrer. If you need a simple share function, without tracking, you will also be told, where to stop.

Step 1 - Add a button to start share



This buttons are often denoted with something like this -

or this

You can find a list of suitable Share buttons, which you can use is here - <https://fontawesome.com/icons?d=gallery&q=share>

Here is an Article on [How to Add Icons](#) to buttons.

Step 2 - Add a Custom JS Action to the button

Below is the sample code to add into the Custom JS -

```
const share = function(){
  navigator.share({
    title: 'mobsted.com',
    text: 'Check out mobsted platform',
    url: 'https://mobsted.com/?utm_referral_objectid=#Object:objid#',
  })
  .then(() => console.log('Successful share'))
  .catch((error) => console.log('Error sharing', error));
}

if (navigator.share) {
  share()
} else {
  alert('Sharing does not work');
}
```

NOTE the URL link above (line 5) - needs to lead to the place in your app, where you want to land all the new users. It can be a certain screen in your app, or a Common Access Link from the Security section of the app, or a screen in a landing-page kind of app, so not the same one from where sharing is made. All depends on the task at hand.

An example of such URL could be - <https://a15ef654.mobsted.com/pwa/?appid=17>

Title and text (lines 3 & 4) - what the native sharing function of a user's device will show during the process.

STOP HERE IF YOU ARE NOT TRACKING REFERRERS IN YOUR SYSTEM

If you do need to track referrers & pass the sharing user's data for marketing/tracking:

It is done using UTM marks, so note the following in the code above (line 5 again):

"?" denotes start of the URM mark

"utm_referral_objectid" is just a name you give this UTM mark, any name here will do, as long as it has no spaces

#Object:objid# is a hashtag to get the right column from Objects. But you can use anything, like a phone number, or email or good option - #Object:uuid# (which is the random hash).

Step 3 - Recording Referrers into new user's data

This step is only needed when tracking referrers. Each new object will receive this incoming UTM mark with some user data you choose from the user who was a referrer.


Now,

On a welcome/landing screen of a needed app, where you will be sending all new users, you need to add similar to the following.

In this example, we add a new Action to the Screen record the incoming UTM on the screen itself, but you can do it on any element (depends on what you want, like to record all shares and link opens or only those who made a certain action).

The Action in the example will be "Save to Objects Column":

Actions of Screen



Copy actions
Choose source component

ADD NEW ACTION

Save to object column action settings

Data

Column name	Value
CustomColumn1	#Route:query:utm_referral_objectid#

Add item

Choose event type

ADD EXECUTION CONDITIONS

Cancel

Save

NOTE - the column MUST EXISTS already in Objects if the welcome/landing app. This column will store the the referrer. In the example above - it is named CustomColumn1.

Use the hashtag designed to pull in data from the URL itself, in this case the UTM marks we created. Full options for this hashtag [are explained in Data Refs in App](#) article > Section 4 > "Current URL Route".

The general form is #Route:query:**custom_name**#. In the case on the picture above we called that **custom_name** as "utm_referral_objectid", which takes the UTM variable created on Step 2 (JS code line 5).

Alternative Route

You can place sharing function to a custom button, not the element taken from the palette on your left.

In this case you will need to take care of Errors in JS and change:

Button's JS Action (recommended)	Custom JS
----------------------------------	-----------

```
if (navigator.share) {  
  share()  
} else {  
  alert('Sharing not works');  
}
```

```
if (navigator.share) {  
  document.getElementById("btnShare").onclick = share  
} else {  
  document.getElementById("btnShare").onclick = () => alert('Sharing does not work');  
}
```

In the Custom JS case, you need to create buttons or link on your own and also you must take into account different naming for each element, so name each sharing Custom JS block's button as "btnShare", "btnShare1", "btnShare2" and so on, other wise only the first script will work.

Thats all. Ask us anything we missed here.