

# Associate Person to Device

Last Updated: Tuesday, June 16, 2020

At first download of an app, a customer is defined *vibes\_device\_id* and *Vibes\_person\_key*, but are unknown and cannot be matched back to a customer within an external CRM. The process of associating a person to a device is to identify a person with a unique identifier, such as a customer id or a loyalty id. This identification enables platform campaign segmentation, multiple push device support, cross channel mapping (SMS, MMS, Wallet), and to be matched back to the external CRM.

To effectively target by Person and Person custom fields, the Vibes Platform needs to have associated a device with a Person.

There are two methods to do this association:

1. [Server side association](#) (**recommended**)
2. [Client side association](#)



## Important Note

The *external\_person\_id* must be the same on the SMS/MMS and Push channels to initiate the cross channel person mapping.

## Associate Person to Device - Server Side

This association is done by a sever-to-server call sending the *vibes\_device\_id* and *external\_person\_id* to associate the device to the Person. Vibes recommends a server-to-server call to associate accounts to devices to prevent account hijacking.

To associate the *vibes\_device\_id* with an *external\_person\_id* requires the following three-step process:

1. The mobile app, via the Vibes SDK, will register the device in the Vibes Platform via the Vibes SDK. This returns the *vibes\_device\_id*.
2. The mobile app will authenticate the app user via login and send the *vibes\_device\_id* to the custom-built API.
3. The customer server will send the *external\_person\_id* and the *vibes\_device\_id* to the Vibes Associate Person call.

### Step 1: Register the Device

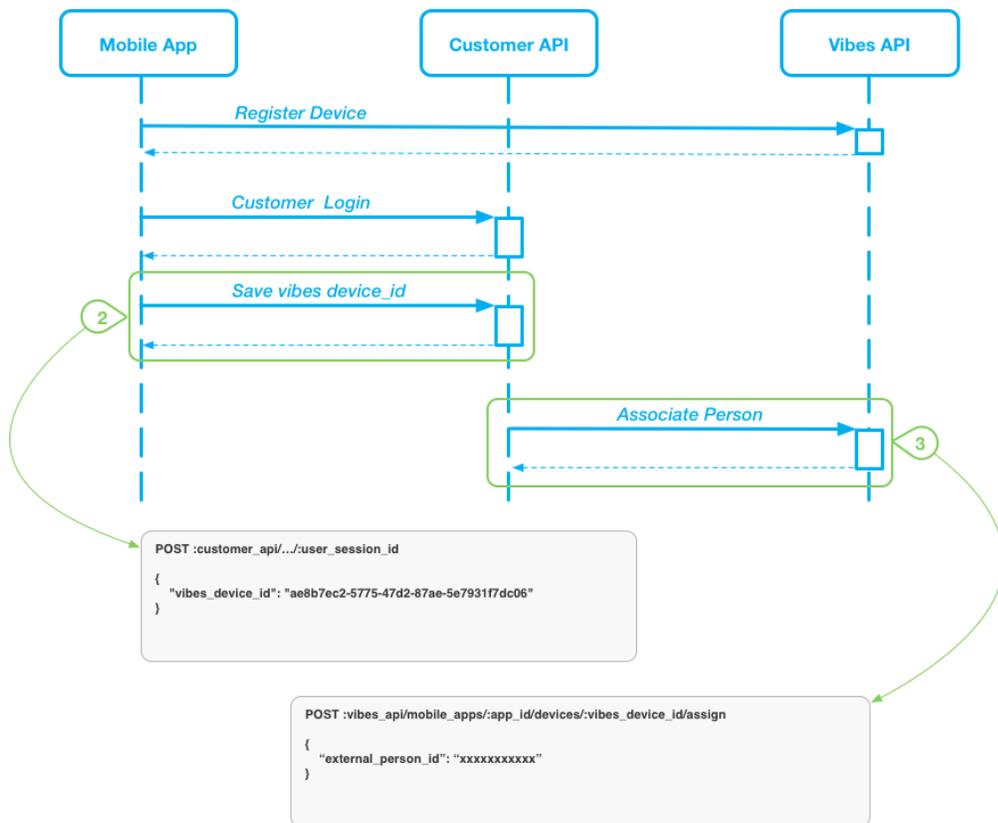
To retrieve the *vibes\_device\_id*, the mobile app will listen to the response from the Vibes SDK register device call. Review the [iOS](#) and [Android](#) integration topics for more information.

### Step 2: Send *vibes\_device\_id* to Customer API

After the device is registered, the customer application needs to save or pass through the *vibes\_device\_id* property that was received in the *registerDevice* response.

This will require a custom-built API to pass the *vibes\_device\_id* or add the *vibes\_device\_id* property in custom *customer\_login* call.

The following is an example customer API call.



## Response Types

The following are possible responses from the customer endpoint:

| Status Code | Definition   | Response Type |
|-------------|--|---------------|
| 204         | The <i>vibes_device_id</i> saved.  | Success       |
| 403         | (Forbidden) The session has expired or the token is invalid.                           | Failure       |
| 422         | There is a validation problem with the request. The resulting error will give details. | Failure       |



**Note:** This is just a suggested API structure. You can implement this API call in whatever way makes sense for your system, as long as the mobile user is securely authenticated.

## Step 3: Send *vibes\_device\_id* and *external\_person\_id* to Vibes Associate Person API

This call assigns a given device to a specific Person. The caller supplies a *person\_key* or *external\_person\_id* to associate the device to. If both are supplied, the *person\_key* will be used and the *external\_person\_id* will be ignored.

- If a *person\_key* is supplied, and a Person with that *person\_key* does not exist, that will result in an error.
- If an *external\_person\_id* is supplied and a Person with that *external\_person\_id* does not exist, then the *external\_person\_id* will be assigned to the Person currently associated with the device.
- In either case, if the indicated Person exists and the device is not already associated with them, it will be moved to be associated with them.
- If the indicated Person is already associated with the device, then nothing will be done and a success response will be returned.

Use the [Associate Push Device to Person API](#) to assign a given device to a specific Person as defined by their *person\_key* or *external\_person\_id*.

## Associate Person to Device - Client Side

To effectively target by Person and Person custom fields, the Vibes Platform needs to have associated a device with a Person. By default, devices can only be associated server side — a 403 forbidden response will be returned if the association is done directly from the device. If you want to enable a client-side association for your app, you must contact Vibes to enable this setting.

## Android

Use the following code to directly update the Person to the device. The *external\_person\_id* is your internal reference to the Person.

You can add the code wherever it makes the most sense for your application.

Kotlin:

```
Vibes.getInstance().associatePerson(externalPersonId, object : VibesListener<Void> {
    override fun onSuccess(value: Void?) {
        //Handle Success
    }
    override fun onFailure(error: String) {
        //Handle Failure
    }
})
```

Java:

```
Vibes.getInstance().associatePerson(externalPersonId, new VibesListener() {
    @Override
    public void onSuccess() {
        //Handle Success
    }
    @Override
    public void onFailure(String error text) {
        //Handle Failure
    }
});
```

## iOS

Use the following code to directly update the Person to the device. The *external\_person\_id* is your internal reference to the Person.

You can add the code wherever it makes the most sense for your application.

```
let externalPersonId = "external user id"
Vibes.shared.associatePerson(externalPersonId: externalPersonId)
```

Delegate method:

```
func didAssociatePerson(error: Error?) {
    if let error = error {
        // there was an error associating the person with this device
    } else {
        // the person was successfully associated with this device
    }
}
```

## iOS Objective-C

Use the following code to directly update the Person to the device. The *external\_person\_id* is your internal reference to the Person.

You can add the code wherever it makes the most sense for your application.

```
NSString *externalPersonId = @"external user id";
[[Vibes shared] associatePersonWithExternalPersonId:externalPersonId];
```

Delegate method:

```
- (void)didAssociatePersonWithError(NSError *error) {
    if (error == nil) {
        // there was an error associating the person with this device
    } else {
        // the person was successfully associated with this device
    }
}
```