API GUIDE V1.0

www.betterpay.online

For any technical assistance regarding the Better Pay API, please contact support@betterpay.online

Environment

- **Production**: https://db.betterpay.online/api/payment
- **Sandbox**: Please see the **Sandbox** section

Request Parameters

All request parameters must be generated from the partner account as described below

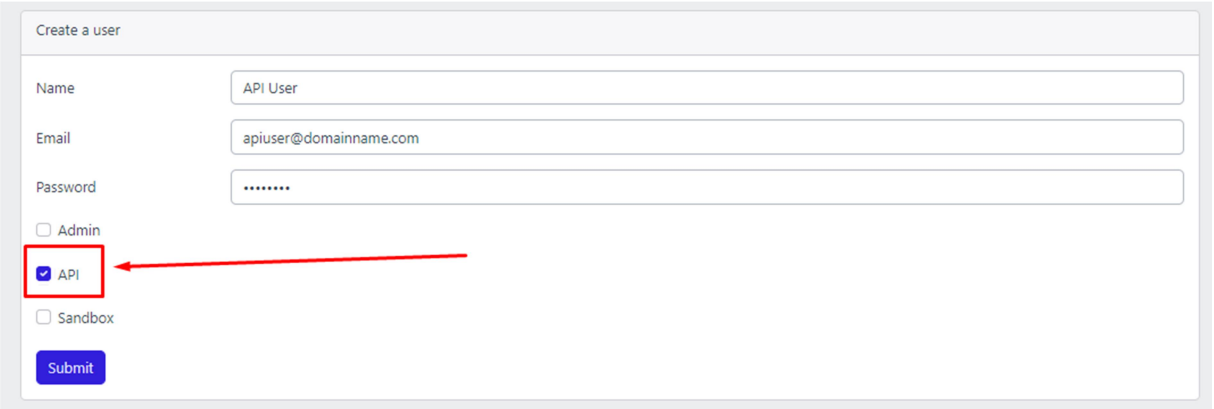
1. **API Key**: Authentication token, in the format Bearer Bearer ${token}
2. **pay_type_code**: A required parameter to connect API payment transactions to specific bank account.

Generating API Key

Position: **Header**
Content-type: **Authorization**

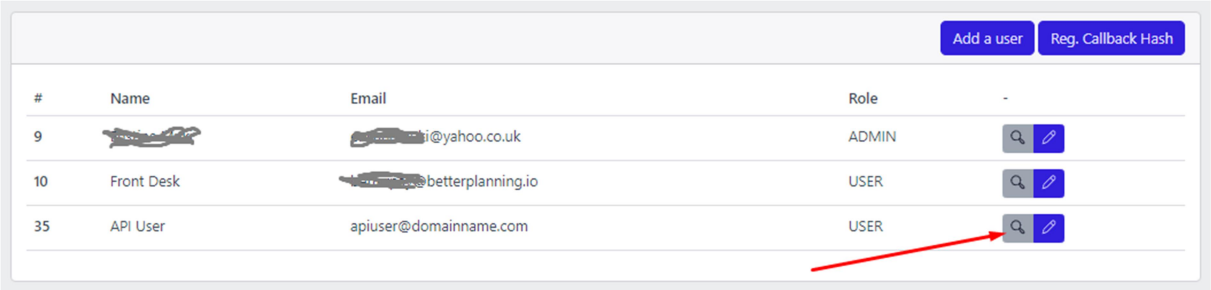
The API key connects a partner through an API user. A partner can have several API users.

1) Create the API User

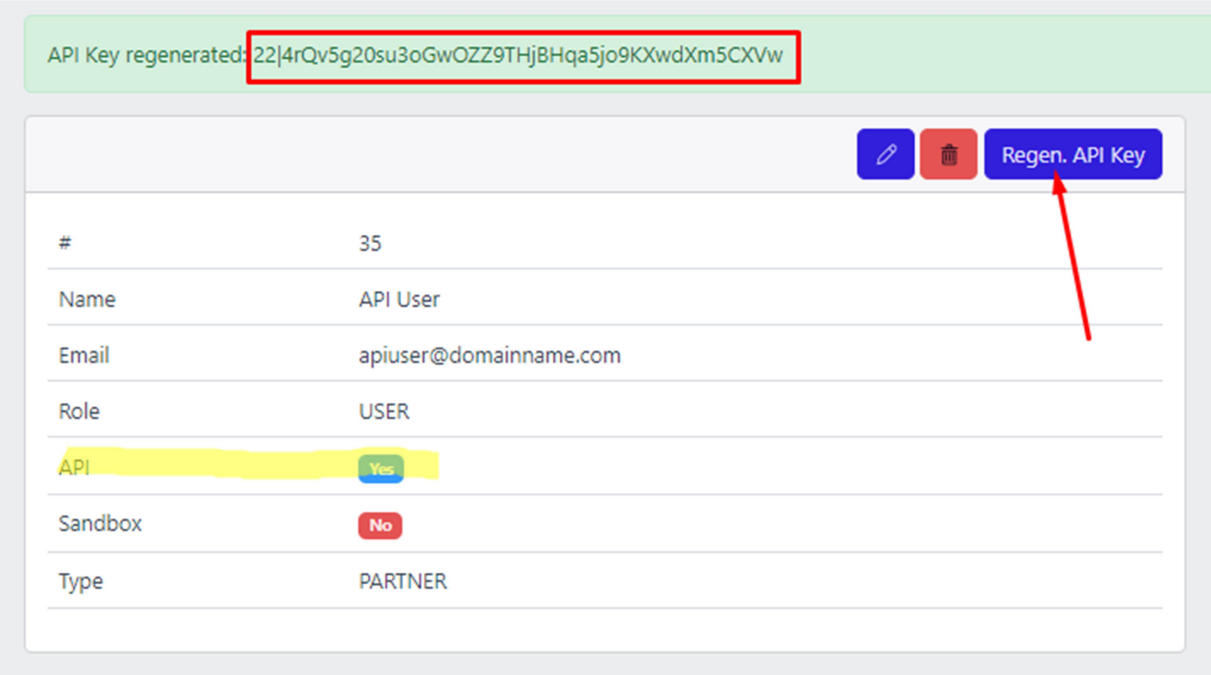


When creating an API user, make sure the API box is checked. Without this, you won't have the option to generate the API Key.

2) Generating the API Key



Click the view icon of the specific user on the list of users as illustrated above. This brings you to the page below



- Make sure that the API value is '**Yes**'
- Click the '**Regen. API Key**' button
- The **API Key** will be generated for this user and displayed in green

NB: You must copy this key out. If this is missing or compromised, you must regenerate another key. The key is not displayed anywhere in the betterpay account.

Here are some few other points to note:

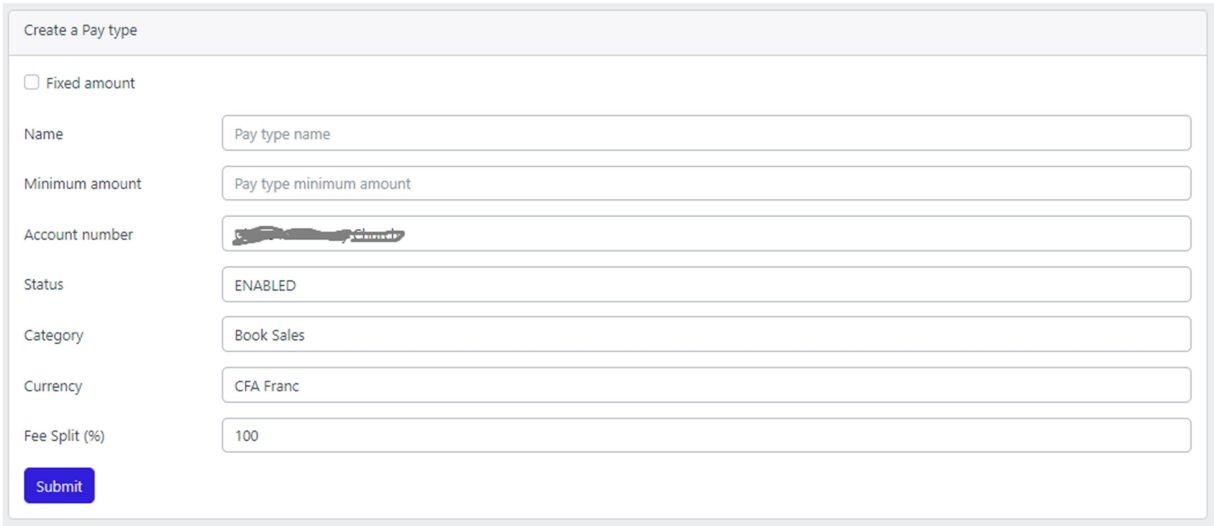
- An existing user can be turned to an API User. That will have no impact on their privileges.
- The API option can also be turned off on any user. If this is done, the API Key is lost. If the user's API option is reactivated, a new key must be generated

The pay_type_code

Position: **Body**

The Pay Type is a key element on the Better Pay platform. It allows users to specify:

3) The bank account to which payments are sent
4) Currency of the payment
5) Amount per transaction (fixed or open with minimum)
6) Mobile Charges
7) Market must be **DISAPORA** *(this is necessary to test card payments)*

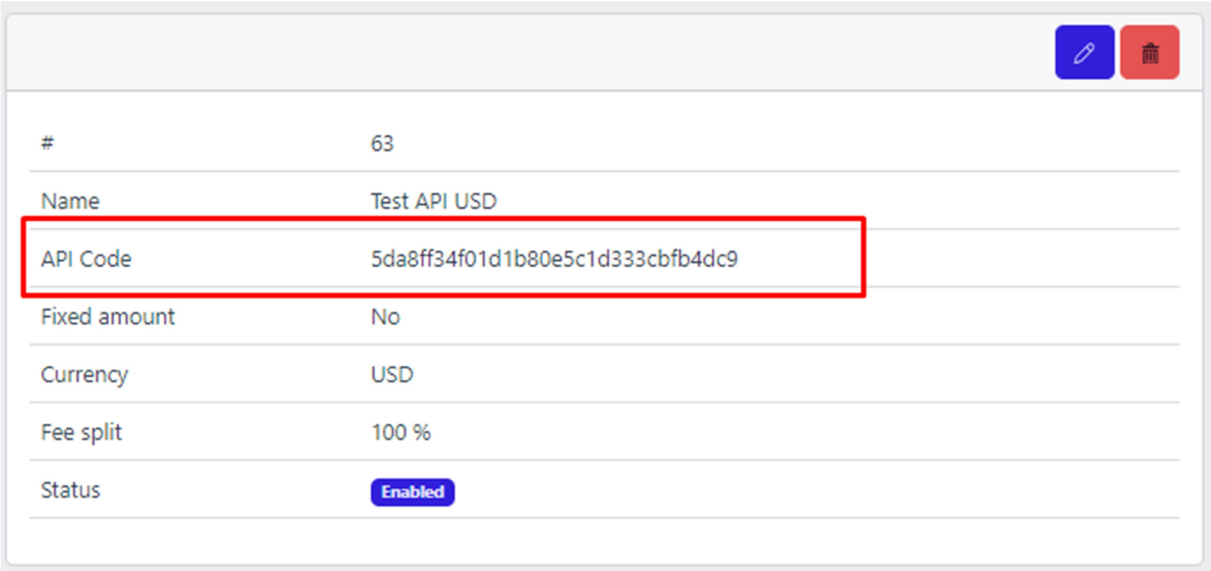


To get the **pay_type_code** for your online shop, go to **Pay Types** -> **Add a pay type** and fill the form.

Here are a couple of key points to note:

- The currency selected must be the currency used on your shop
- Make sure your Financial Institution has attached your bank account to your better pay account.
- If you are running a shop where buyers are expected to pay different amounts, DO NOT activate the **Fixed amount** option. Rather set a minimum.
- For mobile wallet, you have to decide who pays the operator charges (Fee Split). 100% means that the end user pays the charges. For instance, if the amount is 10,000fcfa, the payer pays 10,200 fcfa. If you set the split at zero, the user pays 10,000fcfa and you receive 9,800fcfa in your account. If you set it to 30%, it implies you pay 70% of the charges.

Once the Pay Type is created, click its view button and there you will find the code (called **API Code**)



Please note that the API Code is same as Pay_type_code

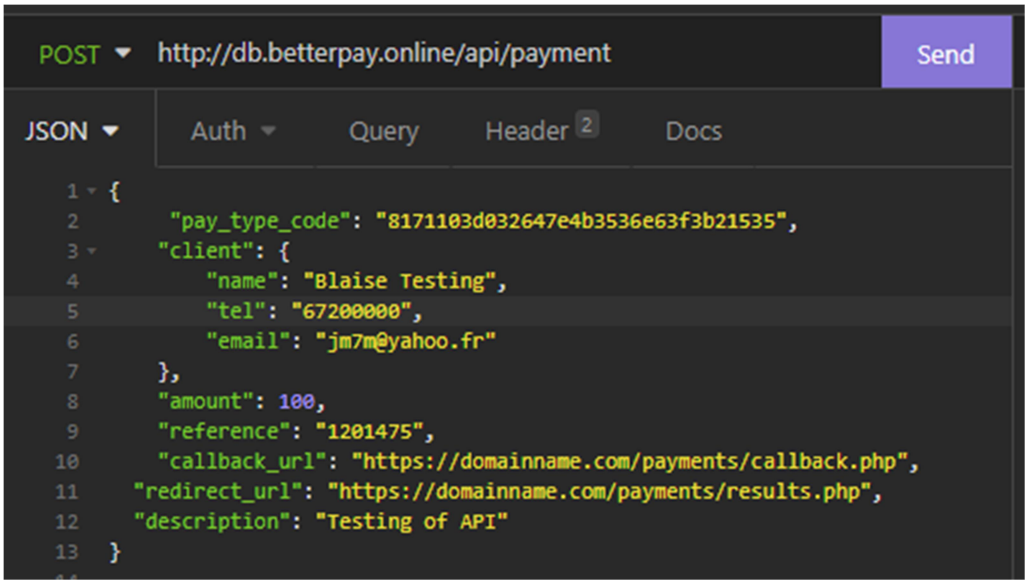
Other request parameters

- client (Name, tel,email)
- amount (do not specify the currency here)
- reference (unique value from shop)
- callback_url

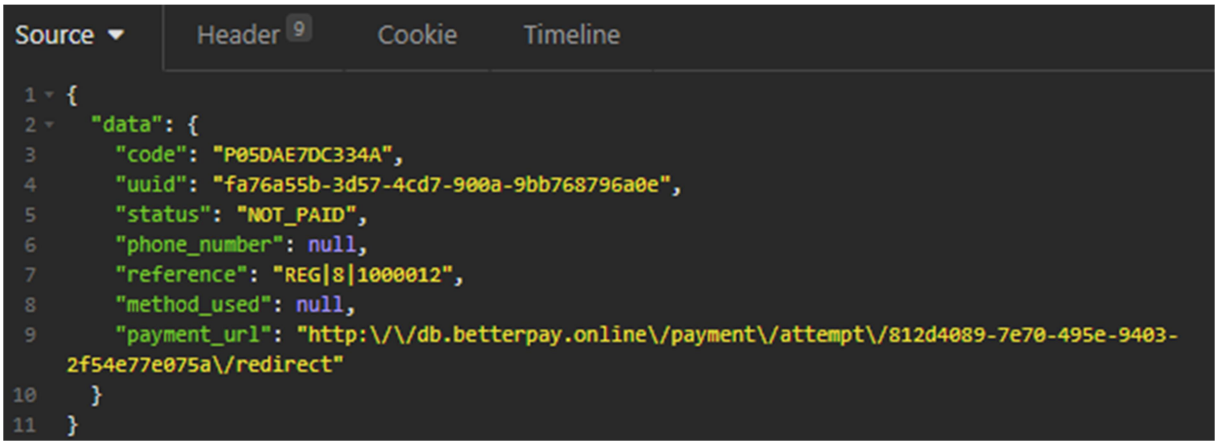
- description
- redirect_url (where the user is redirected after successful or failed payment. The user is redirected with a GET parameter: **success** with value of either *True* or *False* depending on the final payment action

NB: Don't update your payment status based on this value. It can be manipulated. Use the parameters posted to the callback url

API Payment Request



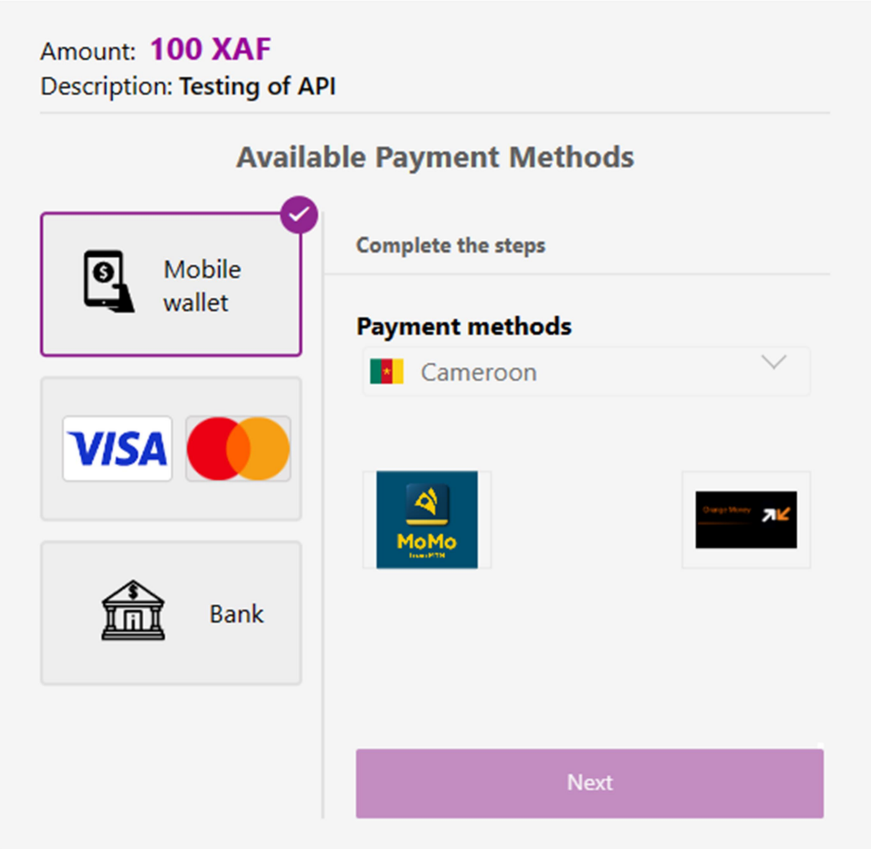
When you create a payment request, betterpay platform returns a series of parameters which will help you proceed with payment.



Both code and uuid are unique values and could be used to update payment status.

The payment_url is another unique element. Based on your code logic, you may use this as well to validate payments. These are all returned in the callback signal as seen below.

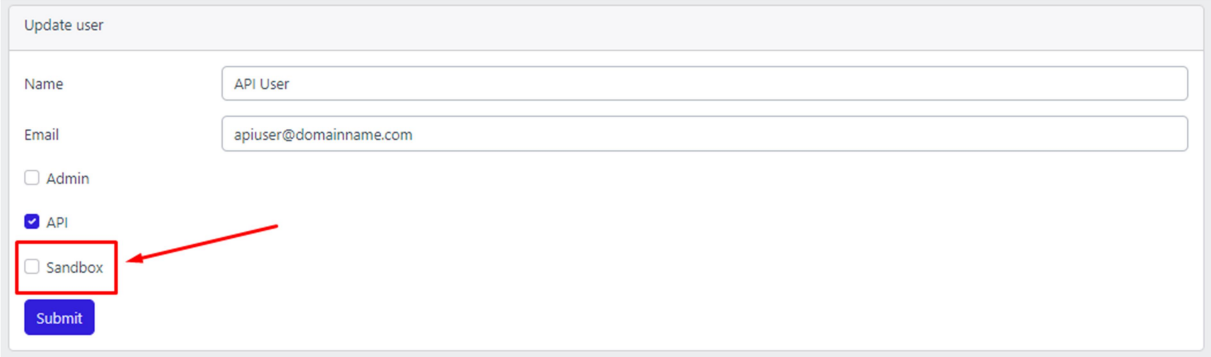
NB: You are required to redirect the payer to the **payment_url** payment authorization URL to proceed with payment. This takes them to the page with different payment methods:



- Mobile Wallet (Available in 16 African Countries)
- Credit and Debit Cards
- Bank Payments

Sand Box

The Sandbox environment allows the coder to test run his code before production. All you have to do is enable the Sandbox option for the API User. We've seen how to create the API User on the pages earlier. Click the ***Edit*** button of the API User and check the Sandbox check box:



That's all you need to do. Any transaction pushed through the sandbox user will run on the sandbox platform.

NB: There won't be any change in the betterpay **payment_url** structure

Callback Parameters

After a complete (failed or successful) payment, betterpay platform will post some parameters to the callback url specified in the request parameters:

x-callback-hash
Location: Header
More about this parameter below.

Other callback parameters:


```
{
  "code": "S581187900EA1",
  "uuid": "dcc623d8-b63d-46fc-a835-5e03e952d28a",
  "status": "PAID",
  "phone_number": "23[redacted]",
  "reference": "DAM|100|1000012",
  "sandbox": true,
  "method_used": "MTN_MOMO",
  "payment_url": "https://db.betterpay.online/payment/attempt/5e4dcd34-b6b4-4c74-b3b8-d7b687d385db/redirect"
}
```

(Please note that the request parameter image above and this callback parameter image are from different transactions. Do not compare their values)

I mentioned earlier that the *code, uuid, payment_url* and the *reference* values in the request command are returned in the callback. *Method_used* , Sanbox and *status* have been added. The value of the *status* parameter is either PAID or NOT_PAID. Sandbox is either *true* or *false*. Use this to update your payment status accordingly.

x-callback-hash

Each payment transaction has a unique callback hash. Use this extra security layer to verify the authenticity of any callback data posted to your platform.

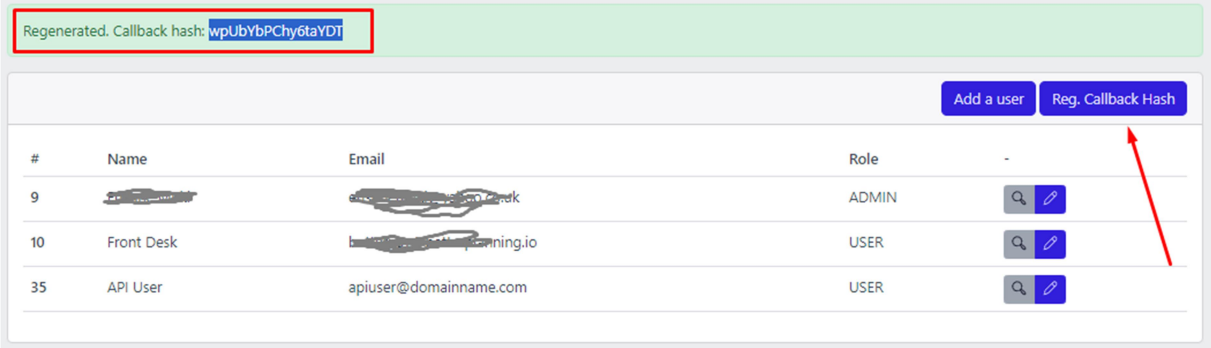
For each payment request, calculate the callback hash value and store locally. Before proceeding to process any callback data received, making sure the callback hash in the callback parameter matches the stored value.

The callback hash is made up of two elements:

1) The code in the payment request. This is unique per transaction.
2) The unique hash generated from your better pay account.

Let's see how to get the hash from your account

Go to your account and click **Users** :



On the list of users, click **Reg. Callback Hash** and copy the hash generated to your program.

Please note that each time a new hash is regenerated, you must update your script with the new hash.

You have to concatenate these two values in a sha256 hash to get the unique hash:

```
$callbackhash = hash("sha256", $code, wpUbYbPChy6taYDT)
```

Get the $code from the payment request data:

Sample Callback PHP script

```
//Do your db connection

// Retrieve the raw POST data
 $jsonBetterPayData = file_get_contents('php://input');

// Decode the JSON data into a PHP associative array
$data = json_decode($jsonBetterPayData, true);

// get the callback harsh from header
$hashBetterPayHeaders = getallheaders();
$xhash=$hashBetterPayHeaders ['X-Callback-Hash'];

if ($data !== null)
{
    // Access the data and perform operation
   $code=$data['code'];
   $uuid=$data['uuid'];
   $status=$data['status'];
   $phone_number=$data['phone_number'];
   $reference=$data['reference'];
   $method_used=$data['method_used'];
   $payment_url=$data['payment_url'];
   $sandbox=$data['sandbox'];

//Do whatever you want with the data. Use the $xhash value to authenticate the source of the data
 }
```