

Brightpearl

Upgrading your app to be multi-currency ready

Version 2.0

Updated February 2019

Chris Tanner

Background	3
Making payments against an order	3
Deprecated API messages	4
New, multi-currency capable API messages	4

Background

Brightpearl now supports multi-currency accounting, and there are some changes we've made to the Brightpearl API in order to do this. In order to make your app to make use of multi-currency accounting, you'll need to adopt some of these new and changed API messages into your integration.

All existing API messages will remain backwards-compatible for a while, and we'll announce separately once a deprecation date has been determined, but we strongly suggest you upgrade your app as soon as possible to keep inline with the rest of Brightpearl's functionality - read the "Guide to multi-currency accounting in Brightpearl".

Making payments against an order

Order payments were introduced back in 2016 with version 4.90. If your app is still using the deprecated Sales Receipt POST to send payments you will need to switch to using the customer-payment POST for sales (supplier-payment POST for purchases) to make your app multi-currency.

Using the Sales Receipt POST all payments are created in the base currency, regardless of the order currency. Using customer-payment POST payments are sent and created in the order currency. The order payments will handle creating the necessary accounting in Brightpearl using the relevant currency.

Adding order payments allowed us to decouple order processing and payment processing from accounting - not all merchants use Brightpearl for accounting, but still want to show their orders as paid. It also means that app developers don't need to consider accounting in their integrations.

As an app developer, you no longer have to consider accounting in order to mark orders as paid. Rather than using an account nominal code, you can now use any payment method code you like. If you use a payment method code that exists (which is the normal route), and that payment method has a bank account for the payment currency, then the accounting transactions will be made for you.

Always send the customer-payment POST in the same currency as the order. Until multi-currency is activated on an account, this will be converted to base currency accounting journals. As soon as multi-currency is activated, foreign currency customer payments will create foreign currency sales receipts.

Deprecated API messages

Old endpoint	New endpoint
accounting-service/sales-receipt POST	accounting-service/invoice/sales-receipt *
accounting-service/journal POST	accounting-service/journal-entry The journal GET and SEARCH will return multi-currency journals.

* Note that if you are using sales-receipt POST to pay sales orders, you should change your app to use the customer-payment POST.

New, multi-currency capable API messages

Endpoint	Purpose
accounting-service/journal-entry POST	Create a journal, eg JJ, BR, BP, BT
accounting-service/invoice/sales-receipt POST	Pay a sales invoice/credit directly **
accounting-service/invoice/purchase-payment POST	Pay a purchase invoice/credit directly **
accounting-service/invoice/sales-invoice POST	Create a SI journal ***
accounting-service/invoice/sales-credit POST	Create a SC journal ***
accounting-service/invoice/purchase-invoice POST	Create a PI journal ***
accounting-service/invoice/purchase-credit POST	Create a PC journal ***
accounting-service/customer-payment POST SEARCH	Pay a Sales Order or Sales Credit
accounting-service/supplier-payment POST SEARCH	Pay a Purchase Order or Purchase Credit
accounting-service/currency/{currency-id}/exchange-rate GET PUT POST	Add a new exchange rate for a currency
accounting-service/sale-payment-total GET	Determine the amount and currency of customer-payments made to date on a sales order or sales credit
accounting-service/purchase-payment-total GET	Determine the amount and currency of supplier-payments made to date on a purchase order or purchase credit

** Do not use these endpoints to pay invoices that relate to orders. Use customer-payment POST or supplier-payment POST. These endpoints are designed for paying “quick invoices” or “supplier bills” that do not relate to an order.

*** If you are wanting to invoice an order, do not use the accounting endpoints. You will end up with just the accounting entries and the order will not be shown as invoiced. Use one of the following order-service endpoints instead:

- sales-order-close POST
- sales-credit-close POST
- purchase-order-close POST
- purchase-credit-close POST