myob

MYOB Advanced

Native eCommerce Connector

Technical Whitepaper

15/08/2022
Version no. 2.0

Contents

Introduction 5
Commerce Connector Solution Overview 5
Entities screen (BC202000) 6
Prepare Data screen (BC501000) 7
Process Data screen (BC501500) 8
Sync History screen (BC301000) 8
Understanding key difference between Shopify and BigCommerce: 9
Comparison: 9
Shopify: 9
BigCommerce: 10
Commerce Edition Synchronized Entities of each eCommerce Platform: 10
Synchronization of Entities - BigCommerce 11
Synchronization of Entities – Shopify 11
eCommerce Features Supported – Out of the box. 12
Establish Connection Between MYOB Advanced and Shopify 13
Create a Shopify Store 14
Initial configuration for the Shopify store 20
Capture store connection information 22
Establish a connection between MYOB Advanced and the Shopify store 30
Establish Connection between MYOB Advanced and BigCommerce 31
Create a BigCommerce store and complete store configuration (optional if you have already had a store) 31
Configure myob advanced erp for integration 34
Capture store connection information 38
Establish connection between MYOB Advanced and BigCommerce 40
Standard field mappings 41
The first synchronisation 42
Automated synchronisation and Real-Time sync 45
Synchronising Customers and Customer Locations 47
Synchronising Customers 47
Synchronising Customer Locations 49
Utilising formulas in Field Mappings for Customers 49
Sync direction, Mapping, Filtering and Primary system 50
Synchronising Inventory and Product Images 50
Synchronising Sales Category – Prerequisite for synchronizing Products for BigCommerce 50

Synchronising Inventory 52
Synchronising Product Images 56
Inventory Settings 57
Synchronising product availability 58
Synchronising Sales Orders 60
Order Settings 60
For Shopify Stores: 61
For BigCommerce Stores: 64
Payment Settings 68
Shipping Settings (Shopify Stores only) 68
Sales Order Processing – Shopify 69
Raising a Sales Order: 69
Syncing the Sales Order: 72
Fulfilling the Sales Order: 74
Preparing Invoice in MYOB Advanced: 74
Processing Customer Refund (for unfulfilled orders) 75
Processing Customer Refund (for fulfilled orders) 78
Discount 81
Sales Order Processing – BigCommerce 84
Raising a Sales Order: 84
Syncing the Sales Order: 86
Fulfilling the Sales Order: 87
Preparing Invoice in MYOB Advanced: 88
Processing Customer Refund (for unfulfilled orders) 88
Processing Customer Refund (for fulfilled orders) 91
Discounts 94
Gift Wrapping and Gift Message 96
Gift Certificate/Gift Card 99
Configure Gift Certificate in MYOB Advanced 100
Enable Gift Certificate/Gift Card feature in the stores 102
Buying a Gift Certificate/Gift Card 103
Redeeming a Gift Certificate/Gift Card 105
Customer Pricing, Bulk Pricing, and Pricing Limitation (BigCommerce Only) 106
Customer Pricing 106
Bulk Pricing 107
Pricing Limitation 108
Template items, matrix items and product variants 108

Implementation Efforts Estimation 109

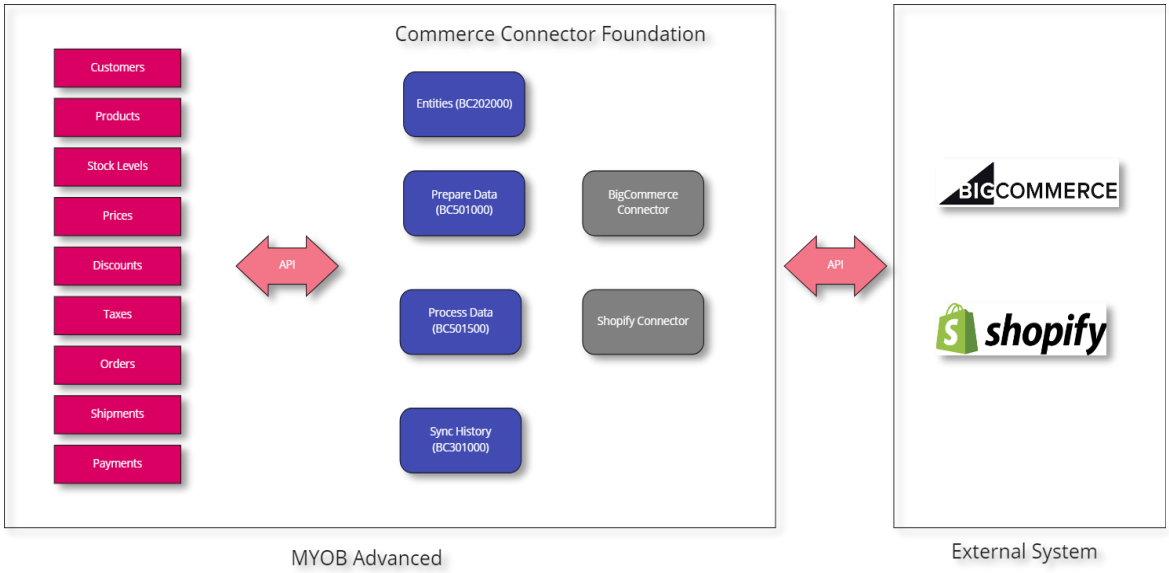
Introduction

MYOB Advanced Native eCommerce Connector provides tools for syncing data between MYOB Advanced ERP and leading eCommerce solutions, BigCommerce and Shopify. Our goal is to provide a Commerce Enabled ERP which will serve B2C and B2B organizations.

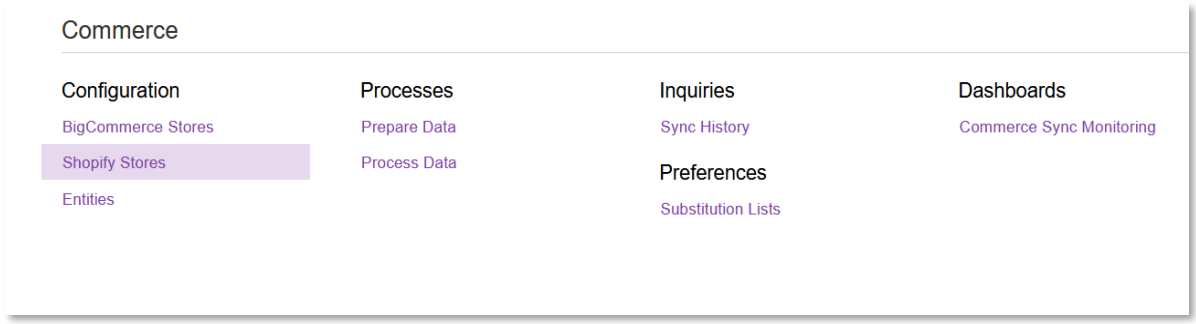
This technical whitepaper aims to provide you with the necessary information about the solution, overview of the eCommerce platforms BigComerce and Shopify, and the necessary configurations to synchronise ERP and eCommerce data.

Commerce Connector Solution Overview

The diagram below explains the overall architecture of the Commerce Connector.



The Commerce Connector is a part of MYOB Advanced, and can be found in the Commerce workspace:



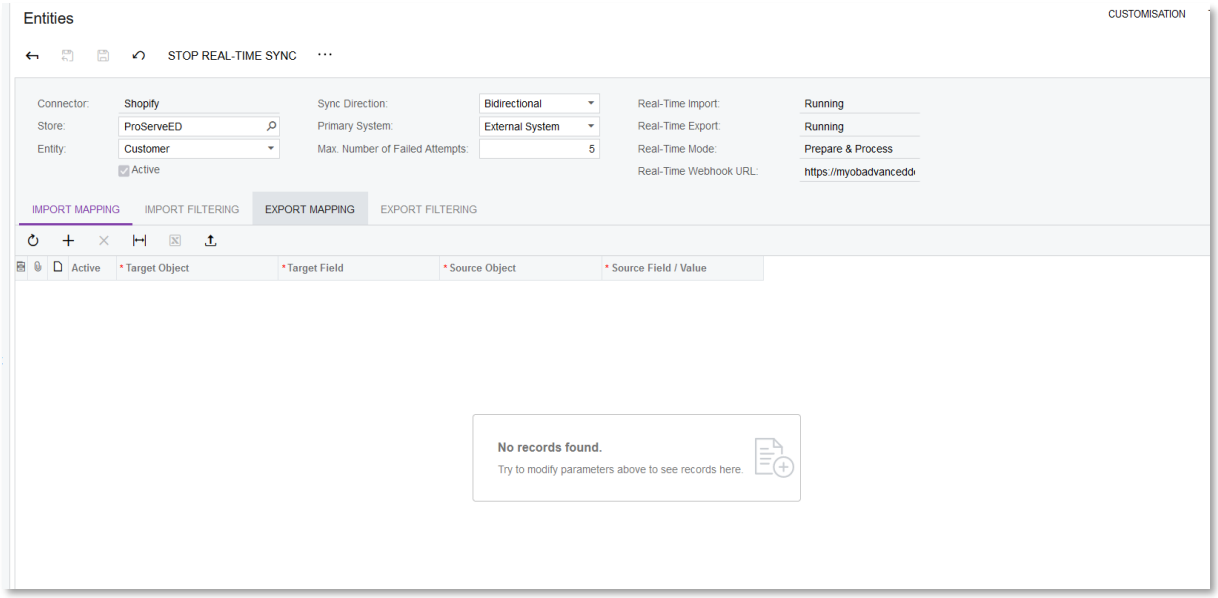
Regardless of what commerce platform the customer is using (whether it is BigCommerce of Shopify), the main functionalities of the Commerce Connector are similar, and are accessible in the following screens:

- Entities – BC202000
- Prepare Data - BC501000
- Process Data - BC501500

- Sync History - BC301000

Entities screen (BC202000)

This screen allows the user to configure Sync Direction (Bidirectional, import or export), Primary System (ERP or External System), and override standard mapping of fields, and add filters when exporting records from MYOB Advanced to external systems and vice versa.



Entities: represents an Entity that will be synced when the Commerce Connector is running, example: Stock Items, Non-Stock Items, Customers, Sales Orders, Template Items, Payments, etc.

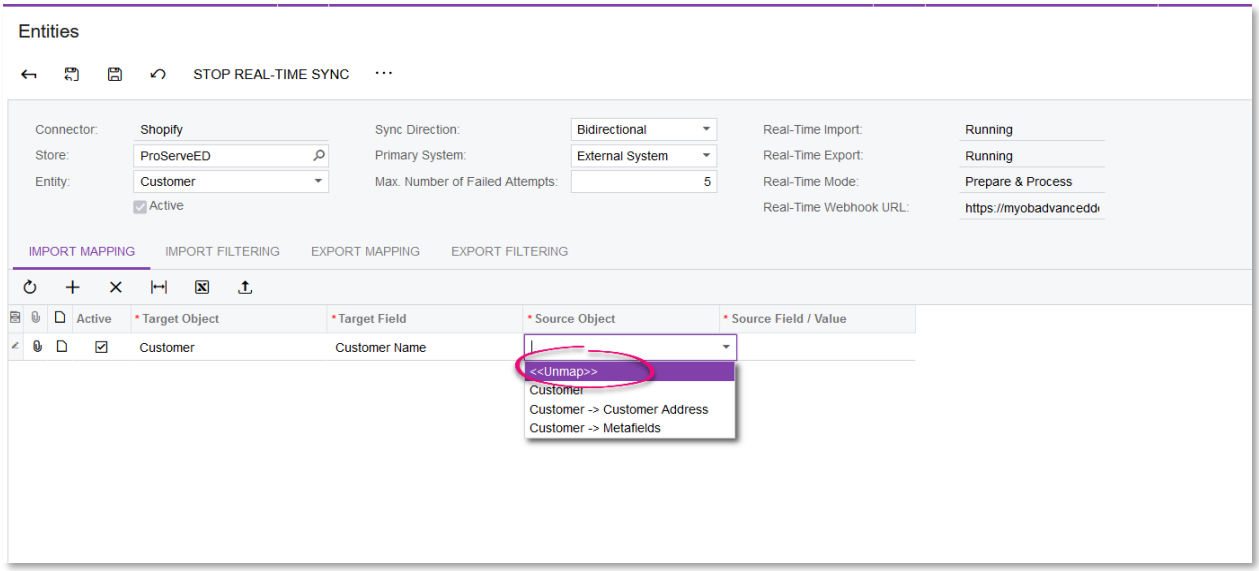
Sync Direction: determines the flow of data for a particular entity when the Commerce Connector is running. Bidirectional means data will be exported from MYOB Advanced to the commerce platform and vice versa. Import means data can only be imported from the commerce platform to MYOB Advanced. Export means data can only be exported from MYOB Advanced to the commerce platform.

Not all Sync Directions are available for all entities. For example, you can have a Bidirectional sync for Customers records, but you can only Export Stock Items from MYOB Advanced to the commerce platform.

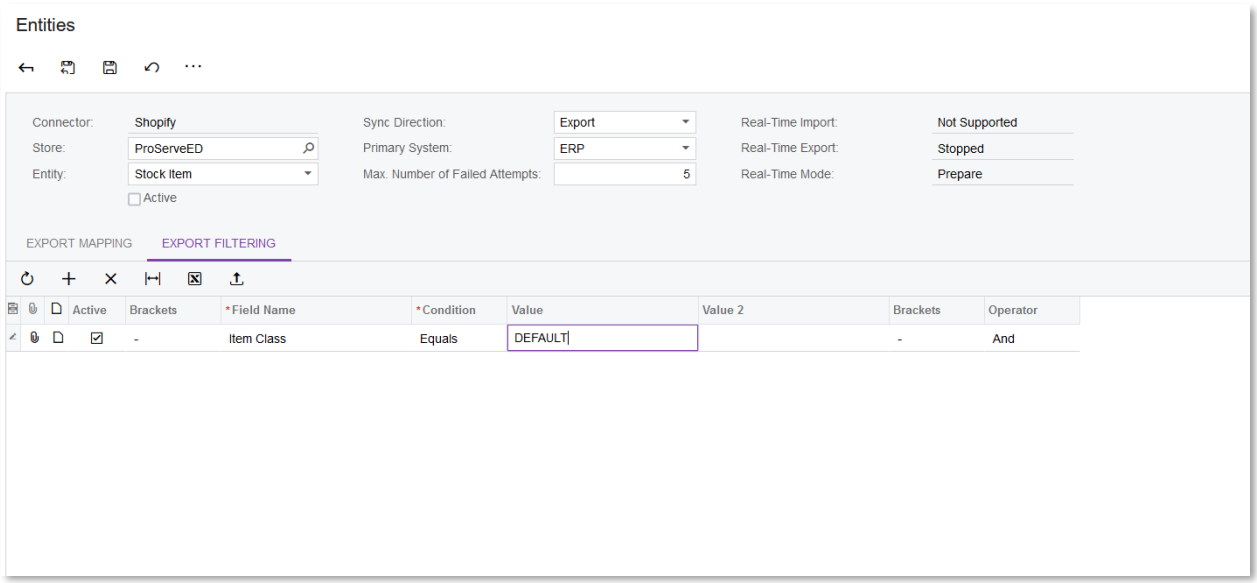
Primary System: Determines from which system the data should be picked by the Commerce Connector in case there's a conflict. For example, if a customer updates their Phone Number on the commerce platform, at the same time an MYOB Advanced user updates the customer phone number in MYOB Advanced. If the Primary System is set to External System, the Commerce Connector will instead uses the value of the commerce platform and overwrite the phone number in MYOB Advanced instead.

Import Mapping/Export Mapping: If there is no Import Mapping or Export Mapping specified, MYOB Advanced and External Systems will sync based on an in-built standard mapping. When you specify an Import Mapping or Export Mapping, the Commerce Connector override the standard mapping with the mapping specified here. The rest of the fields will be synced as per normal

If you want to not import or export a certain field, you can use the “<<Unmap>>” action. This will ensure this field will not be imported or exported when the connector is running.



Import Filtering/Export Filtering: allow the users to configure the filters for importing/exporting records between MYOB Advanced and External Systems. For example, users may only want to export Stock Items where Item Class = DEFAULT. In this case, you will configure an Export Filtering for the Stock Item entity, where Item Class Equals DEFAULT.

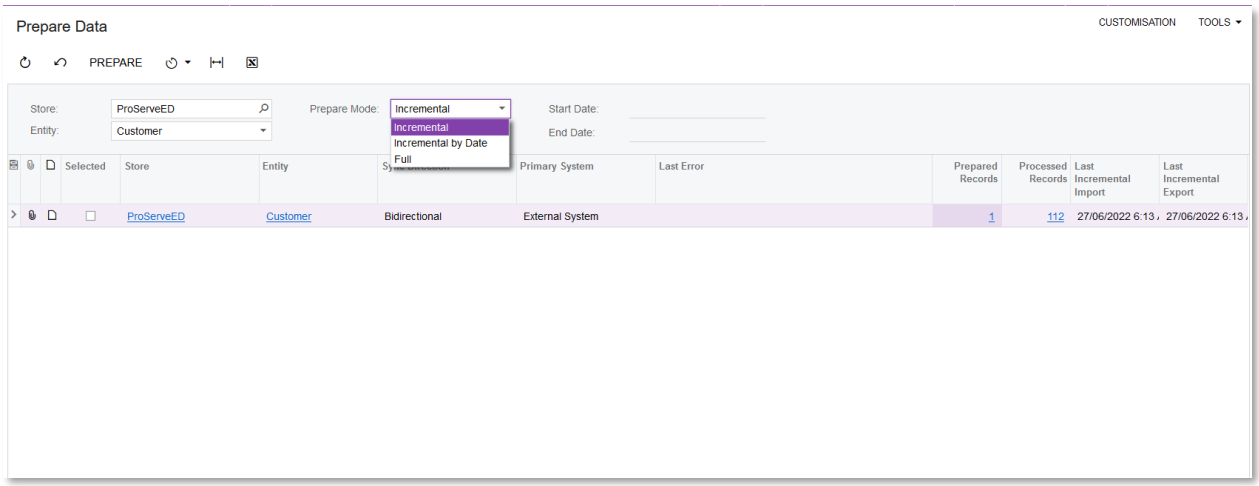


Prepare Data screen (BC501000)

This is the first part of the sync. The Prepare Data screen collates all records that fit the criteria for syncing and put them into a queue, ready for importing/exporting through the “Process Data” screen. In the Prepare Data screen, there are three Prepare Modes:

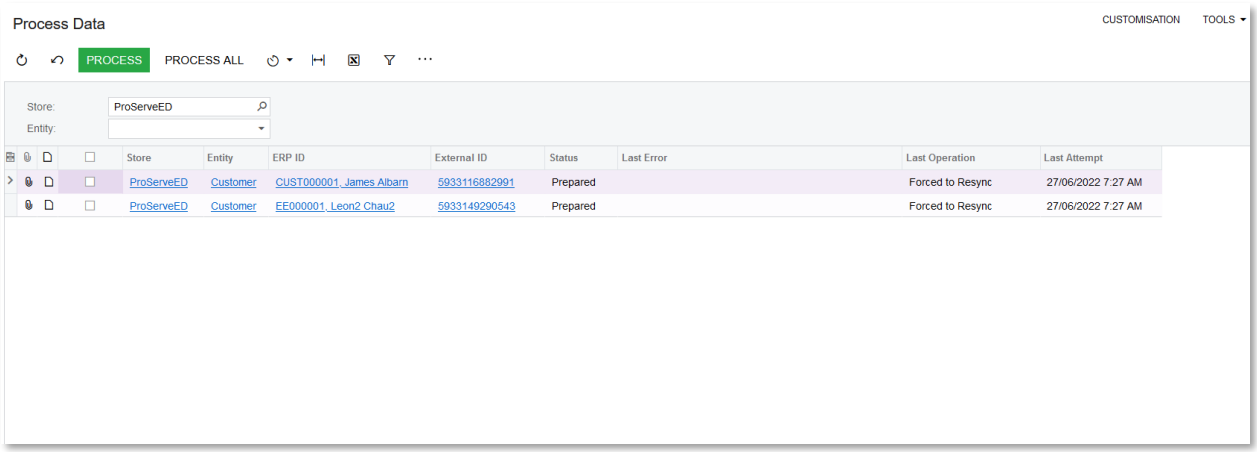
- Incremental: only gather that has not been imported or exported since the Last Incremental Import date and time or Last Incremental Export date and time.

- Incremental by Date: When this value is selected, the Start Date and End Date fields will be available. All records that have been modified within this date range will be put into a queue and ready for import or export.
- Full: all records from the start of time will be put into a queue, ready for import or export. (Except for records that are filtered out through the Import Mapping/Export Mapping on the Entities screen)



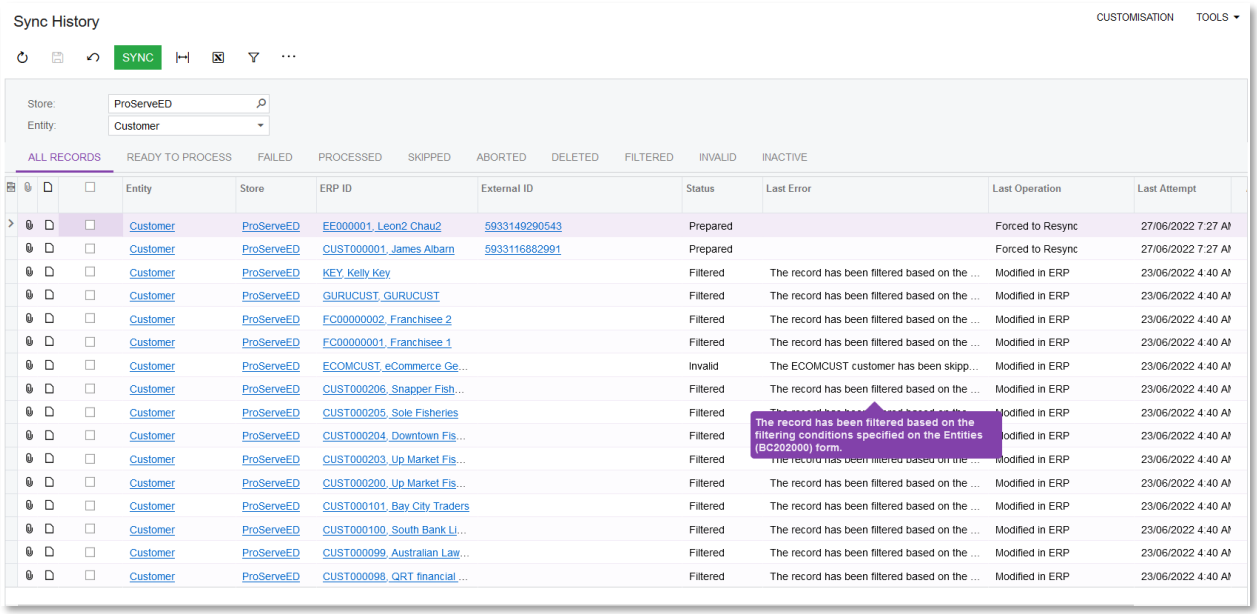
Process Data screen (BC501500)

This screen is where the actual Import or Export happens. After records are prepared through the Prepare Data screen, they will appear in this screen, ready to be imported or exported.



Sync History screen (BC301000)

All records that have been processed through the Process Data screen, will appear in the Sync History screen. This screen is very useful, not only because it shows information related to the status of the sync, what operations were carried out during the last sync (e.g. Inserted Externally, Modified Externally, Modified Internally, etc.), whether a record is synced, skipped, filtered or deleted, but also the corresponding error message in case a sync was not successful, to help users troubleshoot.



Understanding key difference between Shopify and BigCommerce:

Both Shopify and BigCommerce are very strong platforms. MYOB remains agnostic when recommending systems.

Comparison:

- Both platforms are cloud intuitive, cloud-based, SaaS eCommerce platforms.
- Shopify offers native POS functionality and an iOS POS App. BigCommerce integrates with 3rd party POS systems.
- BigCommerce offers more marketplace integrations (Amazon, etc) and they also acquired Feedonomics, a marketplace aggregator.
- Out-of-the-box Shopify does not support customer specific pricing. BigCommerce does. However, the customer needs to be on the Enterprise subscription of BigCommerce to access this feature.
- Many free or inexpensive themes exist for both platforms to help the customers change the look and feel of their store.
- Both platforms offer a large library of resources, documentation, forums and self-help tools online. BigCommerce typically serves midmarket/enterprise customers. Shopify's customer base covers a very wide range customers – from startups through to enterprise.

Shopify:

- Known for being more oriented to B2C merchants - *Shopify will be releasing their new B2B Edition in 2022.*
- Very large marketplace of add-on apps
- Shopify offers additional services to eComm merchants outside of the eCommerce platform.
- Such as a fulfillment network, representation on the Shop app, integrations with social platforms and Headless capabilities
- Shopify Plus merchants have access to Shopify Flow, a visual workflow manager.

- Shopify's front-end performance is the best in the industry. Meaning, Shopify website pages load faster than other platforms.

BigCommerce:

- Features support both B2C and B2B merchants
- The BigCommerce Marketplace offers hundreds of apps providing extended functionality.
- BigCommerce includes many features in their core which Shopify does not. Instead, Shopify relies on 3rd party app builders to solve these challenges.
- BigCommerce offers functionality to build headless eCommerce sites/apps. BigCommerce offers an “Open SaaS” platform. Which is very flexible. Strong BigCommerce agencies can flex the solution into custom
- solutions for merchants.
- BigCommerce was built to serve larger, more complex customers.

Commerce Edition Synchronized Entities of each eCommerce Platform:

Entity	Shopify	BigCommerce
Customer Price Classes	No*	Yes
Customers	Yes	Yes
Customer Locations	Yes	Yes
Sales Category	Yes***	Yes
Stock Item	Yes	Yes
Non-Stock Item	Yes	Yes
Template Item	Yes**	Yes
Base Sales Price	No*	Yes
Price List	No*	Yes
Product Images	Yes	Yes
Product Availability	Yes	Yes
Sales Orders	Yes	Yes
Payments	Yes	Yes
Shipments	Yes	Yes
Refunds	Yes	Yes

***Price Class, Base Sales Price, Price List:** Shopify does not currently support these entities in their core. “Sales” price can be applied with MSRP and Default pricing fields on the Stock Items or Non-Stock Items screen. The MSRP will become Shopify’s “Compare At Price” and the Default price will become Shopify’s “Price”. When MSRP is higher than Default Price (Compare At Price is higher than Price), Shopify will show as though the item is “on sale”.

****Template Item:** Shopify’s core solution only support up to 3 variants on a Product with Variants. This means, a Template Item can only have 3 attributes (for example, Size, Colour and Material). Add-on’s are available which will add more variants/attributes but we have not tested our solution with Shopify Add-ons

*****Sales Category**: Shopify handles Sales Categories/Product Categories in a very unique manner. Our solution will export MYOB Advanced Sales Categories as Shopify’s Product Tags.

Synchronization of Entities - BigCommerce

Entity	Direction	Modes	Filtering	Mapping	Custom Fields
Customer Price Classes	Bi-directional	Manual, Schedule, Realtime	Import/Export	Import/Export	No
Customers	Bi-directional	Manual, Schedule, Realtime	Import/Export	Import/Export	Yes
Customer Locations	Bi-directional	Manual, Schedule, Realtime	Import/Export	Import/Export	Yes
Sales Category	Bi-directional	Manual, Schedule, Realtime	Import/Export	Import/Export	No
Stock Item	Export	Manual, Schedule, Realtime	Export	Export	Yes
Non-Stock Item	Export	Manual, Schedule, Realtime	Export	Export	Yes
Base Sales Price	Export	Manual, Schedule	No	No	No
Price List	Export	Manual, Schedule	No	No	No
Product Images	Export	Manual, Schedule	No	No	No
Product Availability	Export	Manual, Schedule, Realtime	No	No	No
Sales Orders	Bi-directional	Manual, Schedule, Realtime	Import/Export	Import/Export	Yes
Payments	Import	Manual, Schedule	Import	Import	Yes
Shipments	Export	Manual, Schedule, Realtime	Export	Export	Yes
Refunds	Import	Manual, Schedule	No	No	No

Synchronization of Entities – Shopify

Entity	Direction	Modes	Filtering	Mapping	Custom Fields
Customers	Bi-directional	Manual, Schedule, Realtime	Import/Export	Import/Export	Yes
Customer Locations	Bi-directional	Manual, Schedule, Realtime	Import/Export	Import/Export	Yes

Stock Item	Export	Manual, Schedule, Realtime	Export	Export	Yes
Non-stock Item	Export	Manual, Schedule, Realtime	Export	Export	Yes
Template Item	Export	Manual, Schedule, Realtime	Export	Export	Yes
Product Images	Export	Manual, Schedule	No	No	No
Product Availability	Export	Manual, Schedule, Realtime	No	No	No
Sales Orders	Import	Manual, Schedule, Realtime	Import	Import	Yes
Payments	Import	Manual, Schedule	Import	Import	Yes
Shipments	Export	Manual, Schedule, Realtime	Export	Export	Yes
Refunds	Import	Manual, Schedule, Realtime	No	No	No

eCommerce Features Supported – Out of the box.

Feature	Shopify	BigCommerce
Support B2C merchants	Yes	Yes
Support B2B merchants	No*	Yes
Gift Card Support	Yes	Yes
Customer Specific Pricing	No	Yes (only with BigCommerce Enterprise Subscription)
Kit Assembly Support	Yes	Yes
Products with variants via Matrix Inventory	Yes	Yes
Control Product Availability based on Inventory Levels	Yes	Yes
Coupons and Discount code support (sync to SO)	Yes	Yes
Volume based discounts	No	Yes
Export offline orders of multiple Order Types to eCommerce "My Account" section	No	Yes
Pay by Terms, COD, in-store pickup	Yes	Yes
Payment information sync	Yes	Yes
Flexible Shipping Options	Yes	Yes

Tracking numbers export to eCommerce Orders	Yes	Yes
Export offline orders to the eCommerce "My Account > Orders" page	No	Yes**
Manage catalogue structure from ERP	Yes***	Yes
Guest customer checkout support	Yes	Yes
Product information management (title, description, images, pricing, videos, SEO, attributes)	Yes****	Yes
Custom field support	Yes	Yes
Supply inventory from multiple warehouses	Yes	Yes
Display inventory levels per warehouse/location on eCommerce	Yes	No
Native POS application support	Yes	No
Sell personalised items	Yes	Yes
Sell Gift Wrapping service and submit Gift Messages	No	Yes
Map Data from ERP to eCommerce Customer/Meta fields	Yes	Yes

*B2B merchant support is possible if customer specific pricing is not required.

**Merchant chooses which Order Types should be exported to BigCommerce. The products on the order are required to reside on the BigCommerce database, even if not visible to the public. This is required so BigCommerce can render product information (title, short description, images, etc.) on the "My Account > Orders" page.

***Shopify's catalogue structure is unique – Sales Categories will be exported as Tags

**** Videos are not supported in the Shopify connector. Shopify's REST API does not accept Youtube and Vimeo URLs

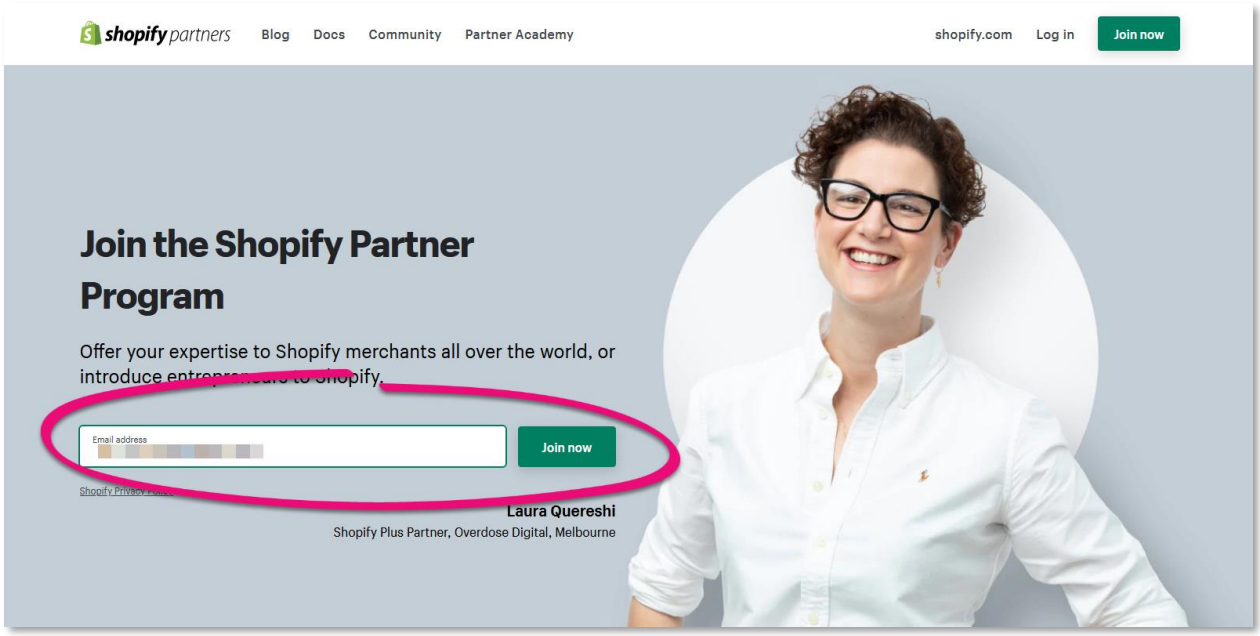
Establish Connection Between MYOB Advanced and Shopify

NOTE: in a normal scenario, you will likely configure Shopify integration for your customer's existing store. The integration process is similar. However, this guide also covers the setup of a Shopify store using Shopify partner's program so that you can set up a store for testing purposes without being limited by the two weeks trial duration by Shopify.

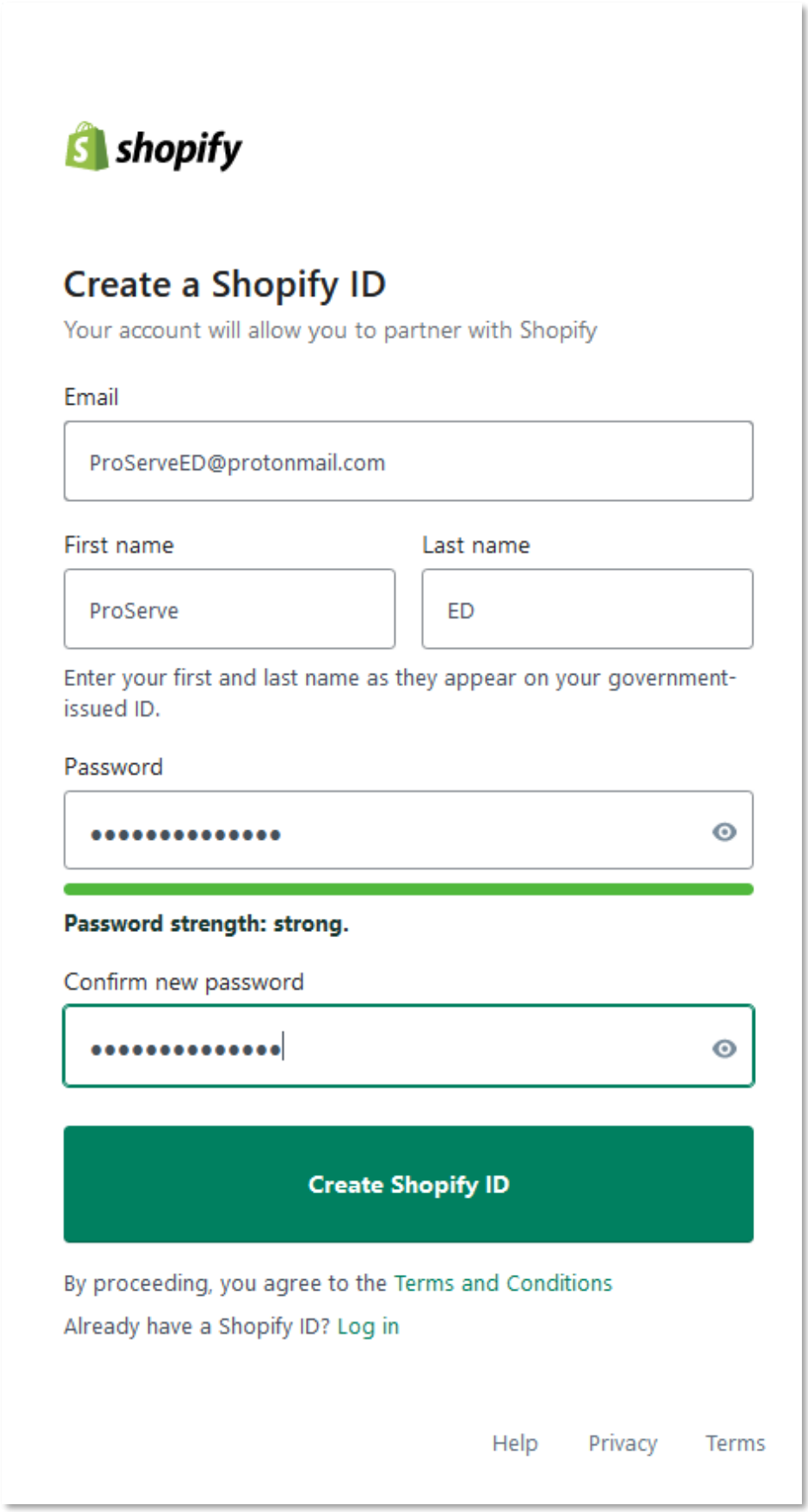
If you have already had a Shopify store up and running, you can skip the "Create a Shopify Store" part.

Create a Shopify Store

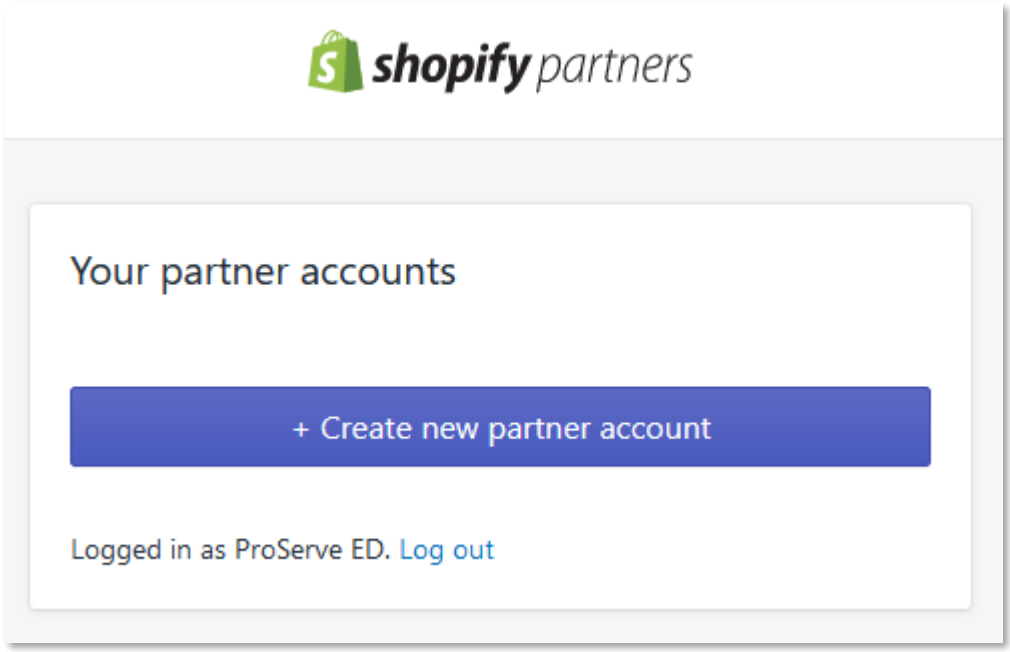
1. To set up a test store using Shopify Partners program, go to: https://www.shopify.com.au/partners - enter your email address and click join now to register as a partner:



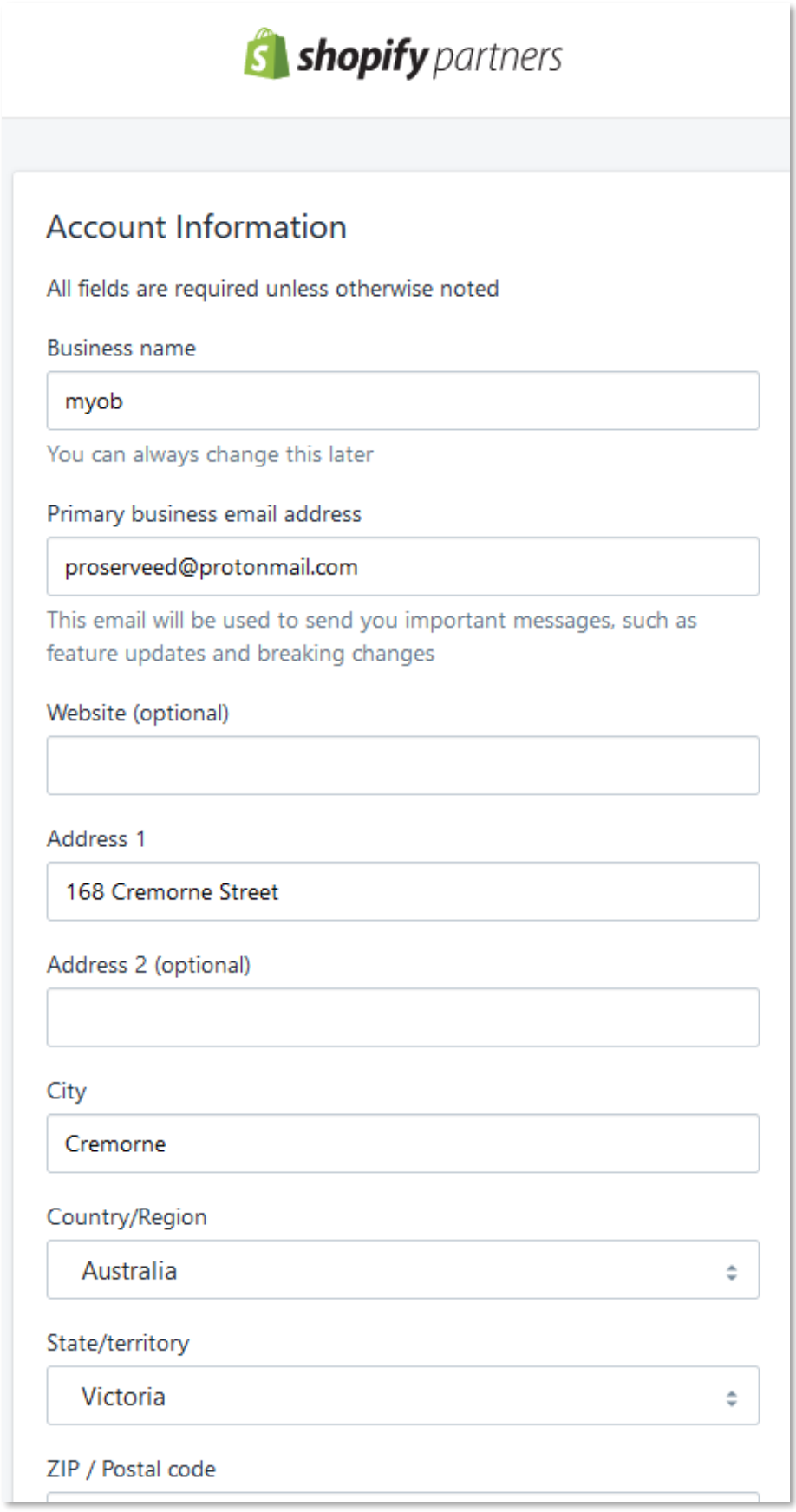
2. Enter your first name, last name, email address and password to create a new Shopify ID. Click Create Shopify ID when finish.


shopify
Create a Shopify ID
Your account will allow you to partner with Shopify
Email
ProServeED@protonmail.com
First name
ProServe
Last name
ED
Enter your first and last name as they appear on your government-issued ID.
Password
Password strength: strong.
Confirm new password
Create Shopify ID
By proceeding, you agree to the Terms and Conditions
Already have a Shopify ID? Log in
Help
Privacy
Terms

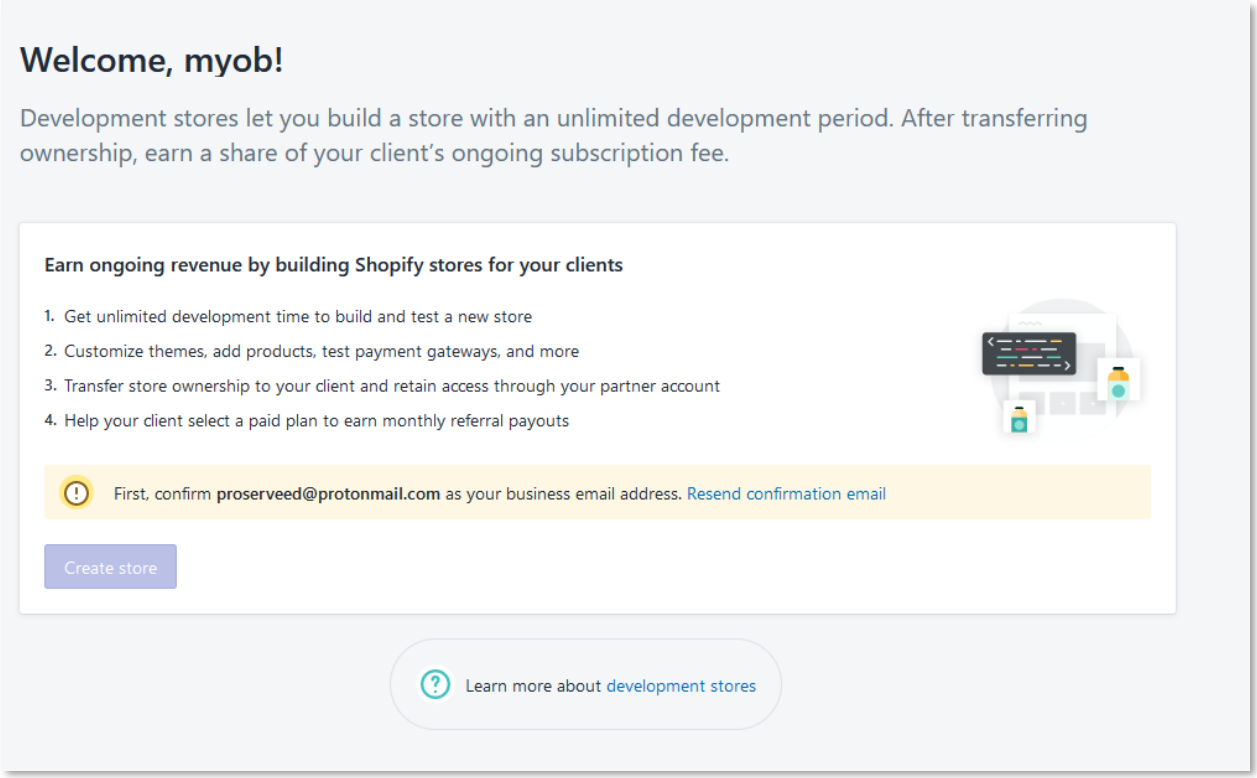
3. Click Create New Partner Account:



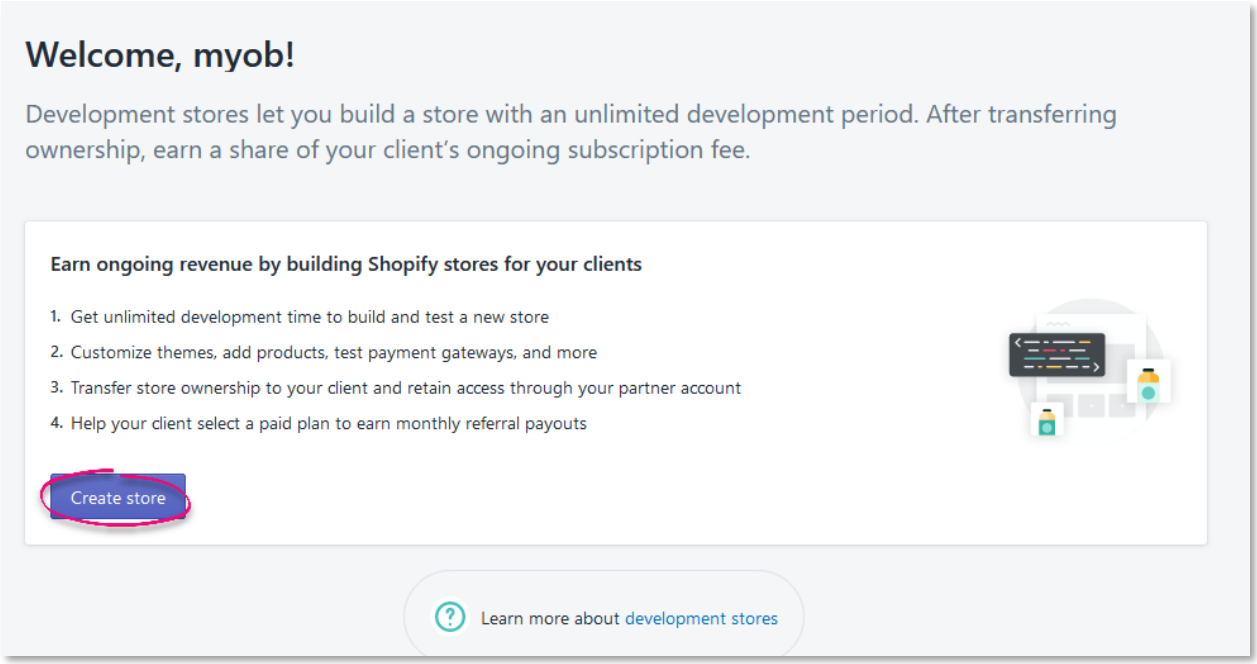
4. You will be prompted to the next screen where you will need to fill in your details for the Shopify Partner Account. Fill in the details as required. Once you finish, click “View Your Dashboard” at the bottom of the screen.


shopify partners
Account Information
All fields are required unless otherwise noted
Business name
myob
You can always change this later
Primary business email address
proserveed@protonmail.com
This email will be used to send you important messages, such as feature updates and breaking changes
Website (optional)
Address 1
168 Cremorne Street
Address 2 (optional)
City
Cremorne
Country/Region
Australia
State/territory
Victoria
ZIP / Postal code

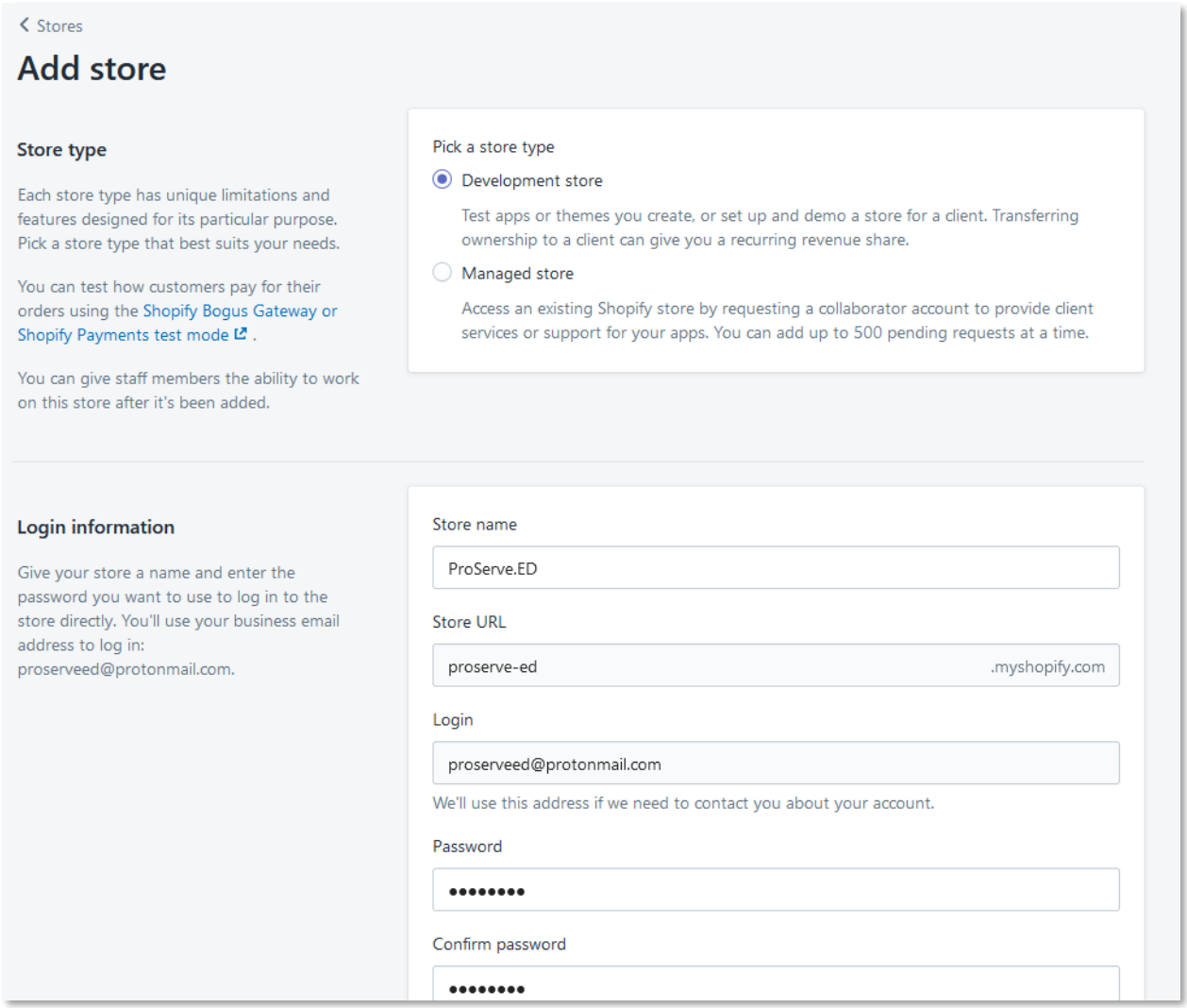
5. Once you can see your dashboard, before you can create a new store, you need to verify the email address that you've registered with Shopify. Go into your mailbox and find the email from Shopify to verify your email address.



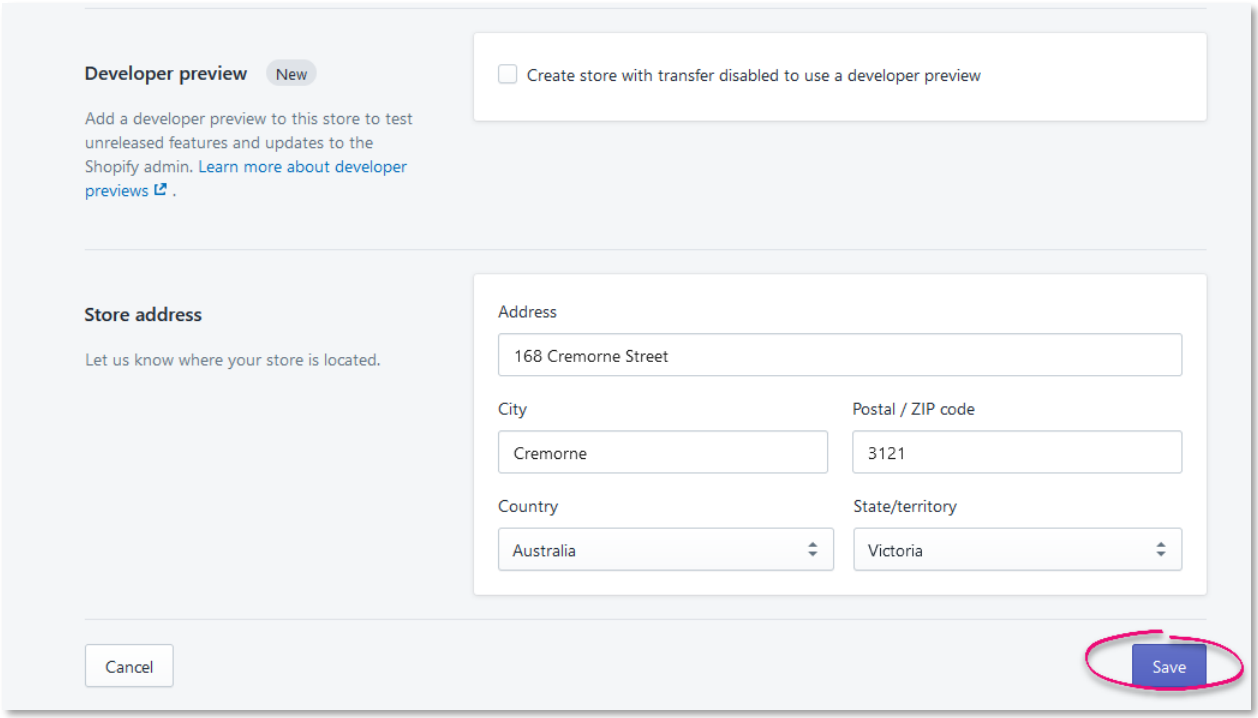
6. Once you have verified your email, go back to Shopify. You can now create a new store. Click "Create Store" to create a new Shopify store:



7. In the Add Store page, enter the store name, login information



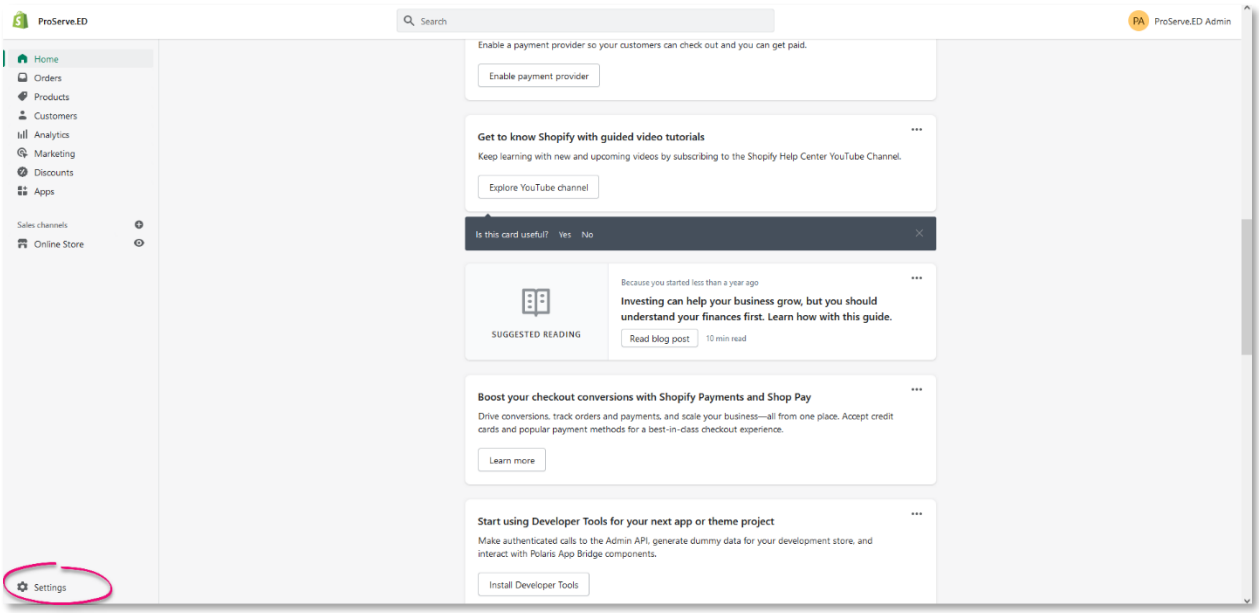
8. Enter the store address and press Save:



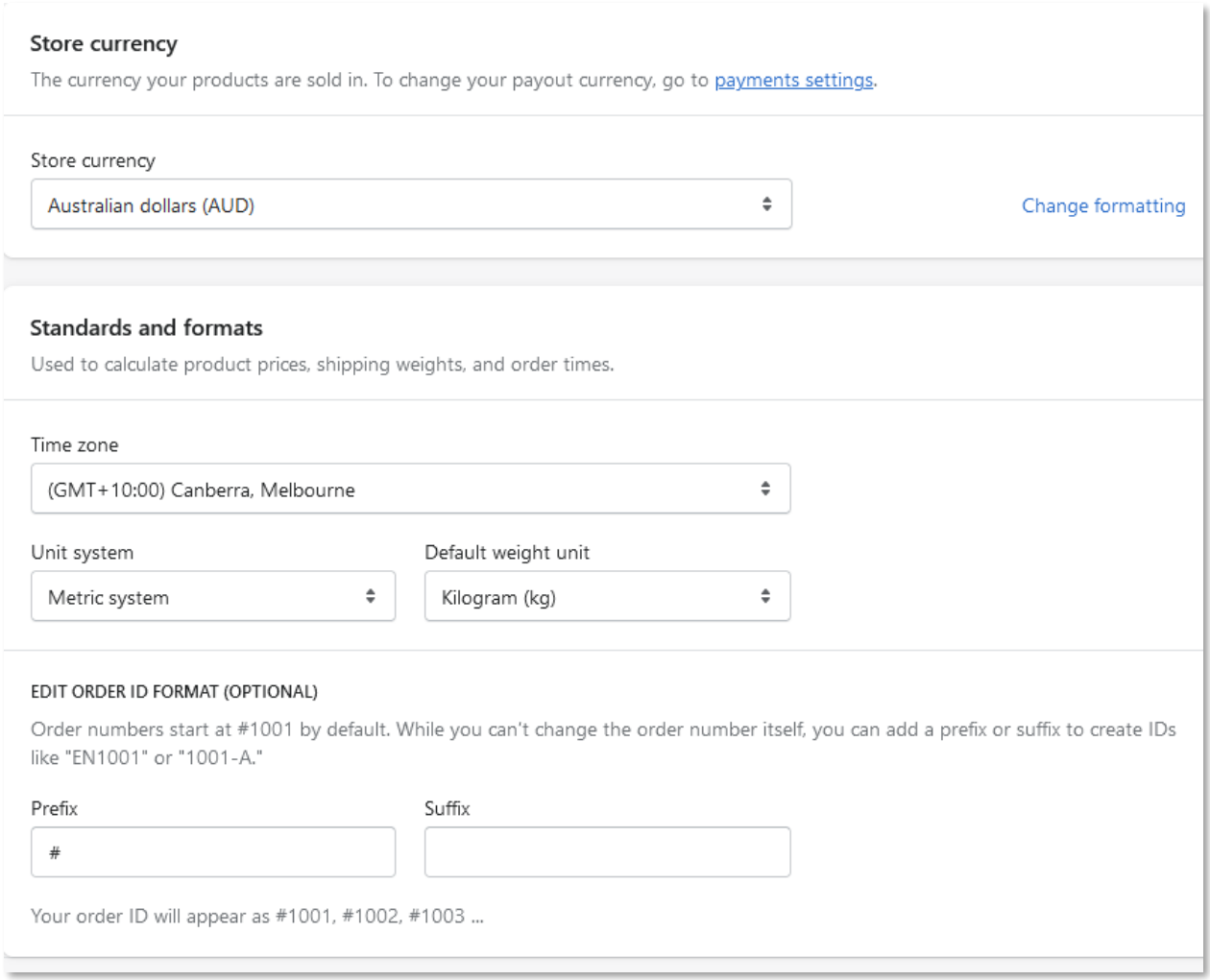
9. Your online store will now be created. To access the store's admin page, in the browser, enter the https://<storeURL>/admin - the store URL is set up in step 7 (see screenshot).

Initial configuration for the Shopify store

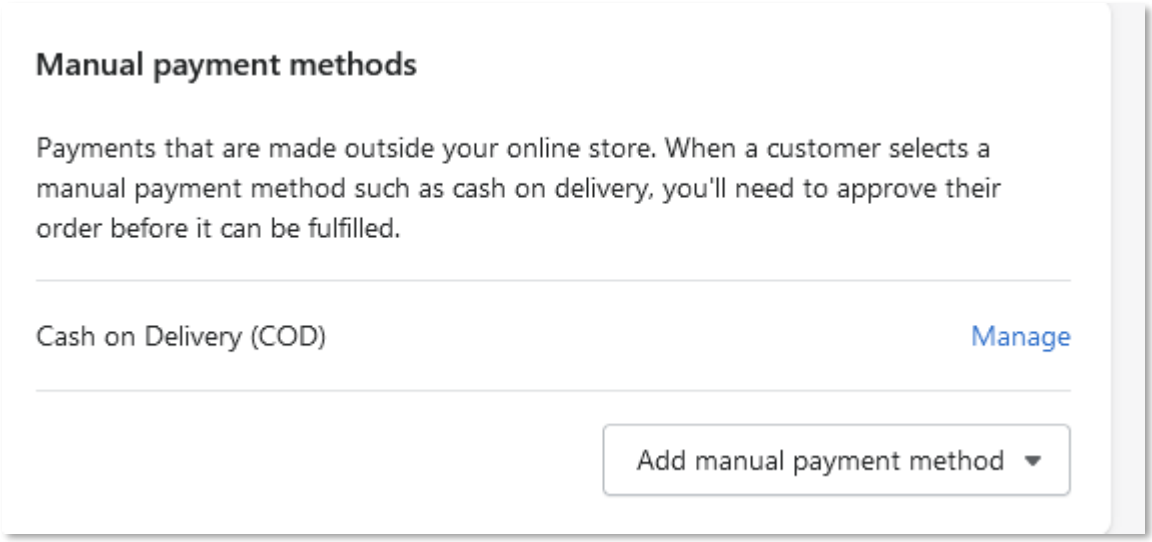
1. Access your Shopify store admin page (https://<storeURL>/admin)
2. Click Settings at the bottom left of the screen:



3. In the Store Details tab, scroll down to the bottom and set the store currency, time zone and Unit of Measure for your store accordingly:

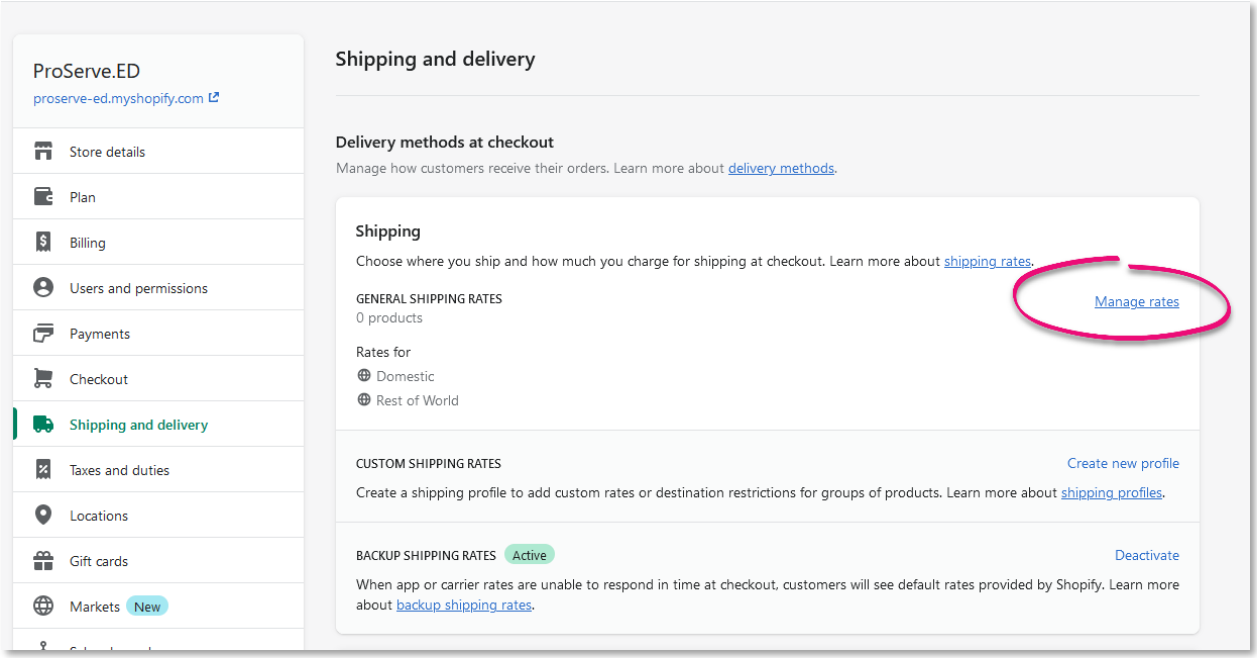


4. Click on the Payments tab, and scroll down to the Manual Payment Methods section, add a new manual payment method for “Cash On Delivery (COD)”. Click “Activate Cash On Delivery (COD)”:



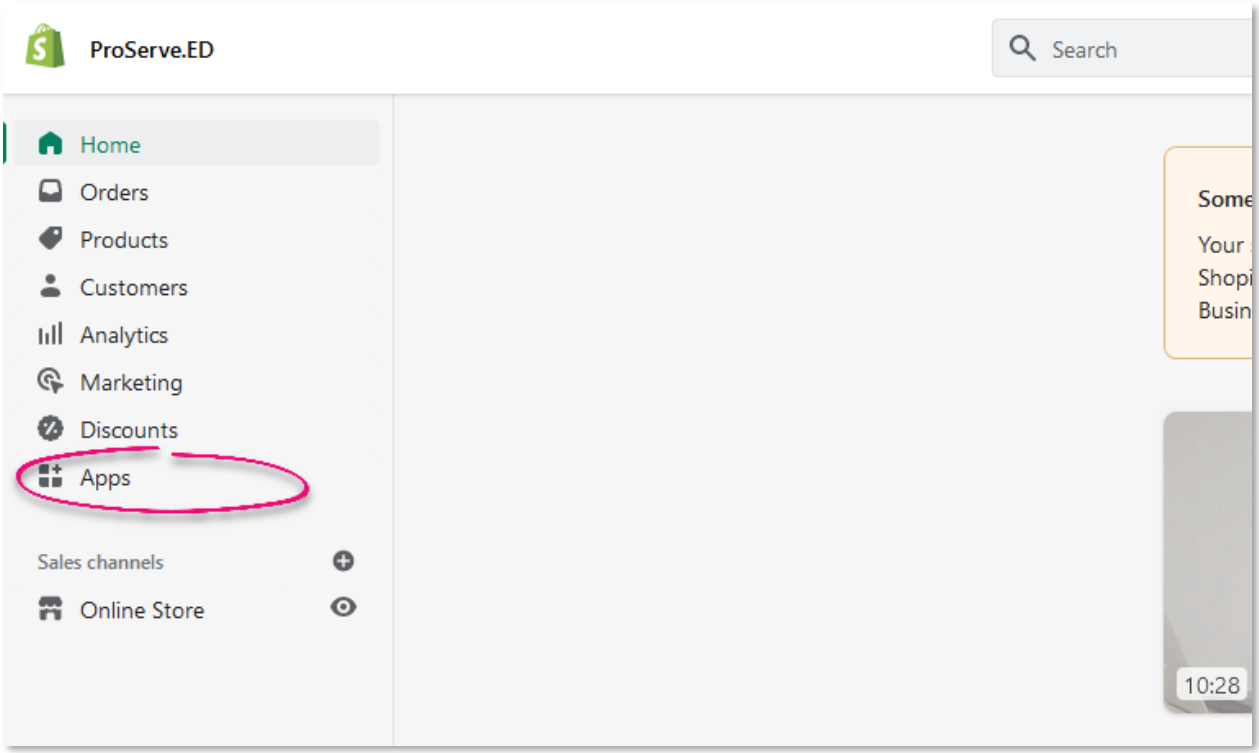
5. Click on the Shipping and delivery tab, you should already see that there are two shipping zones: Domestic and Rest of the world. If required, click on Manage Rates to

manage the shipping rates. Otherwise, the basic configuration of the store to support the integration between Shopify and MYOB Advanced is finished.

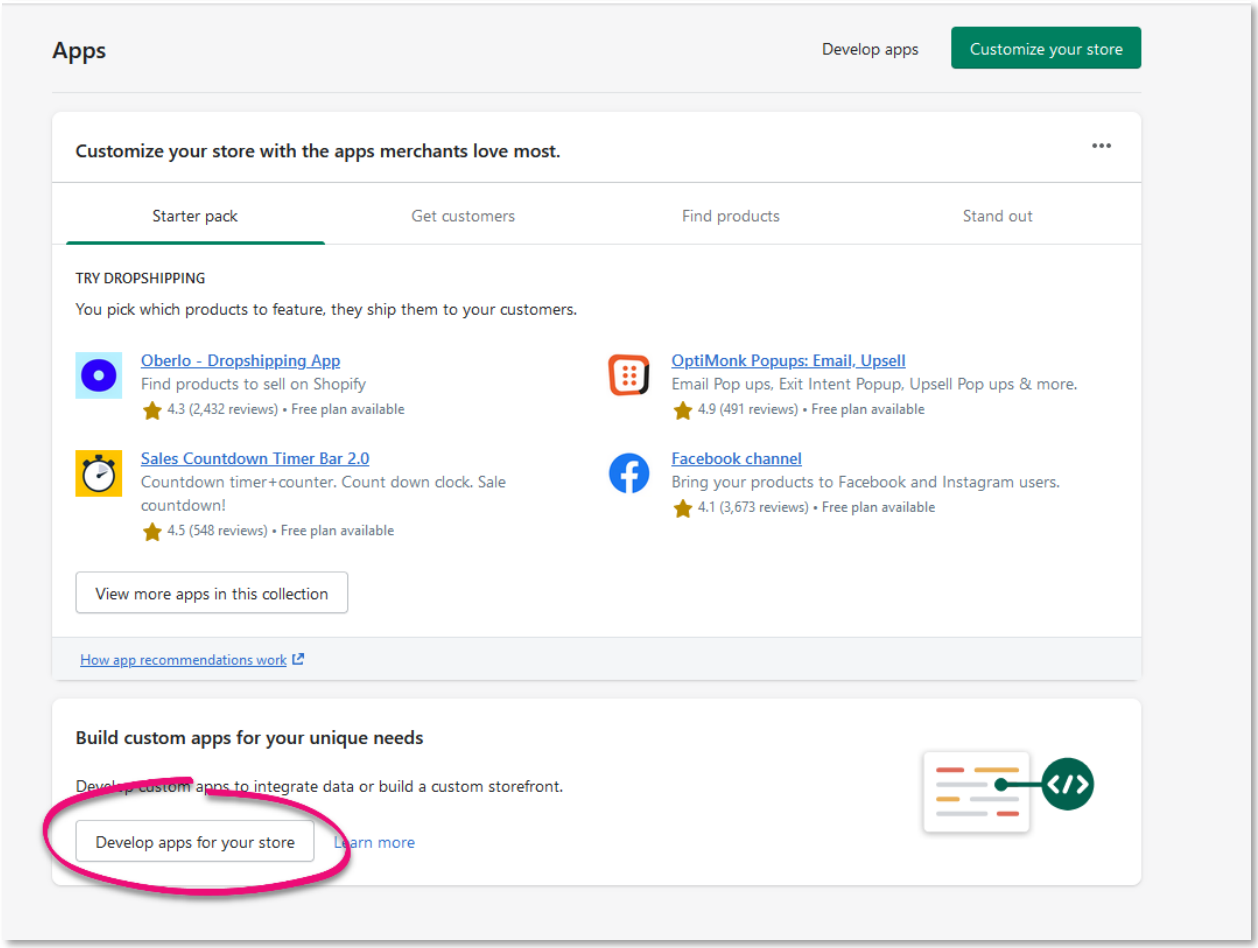


Capture store connection information

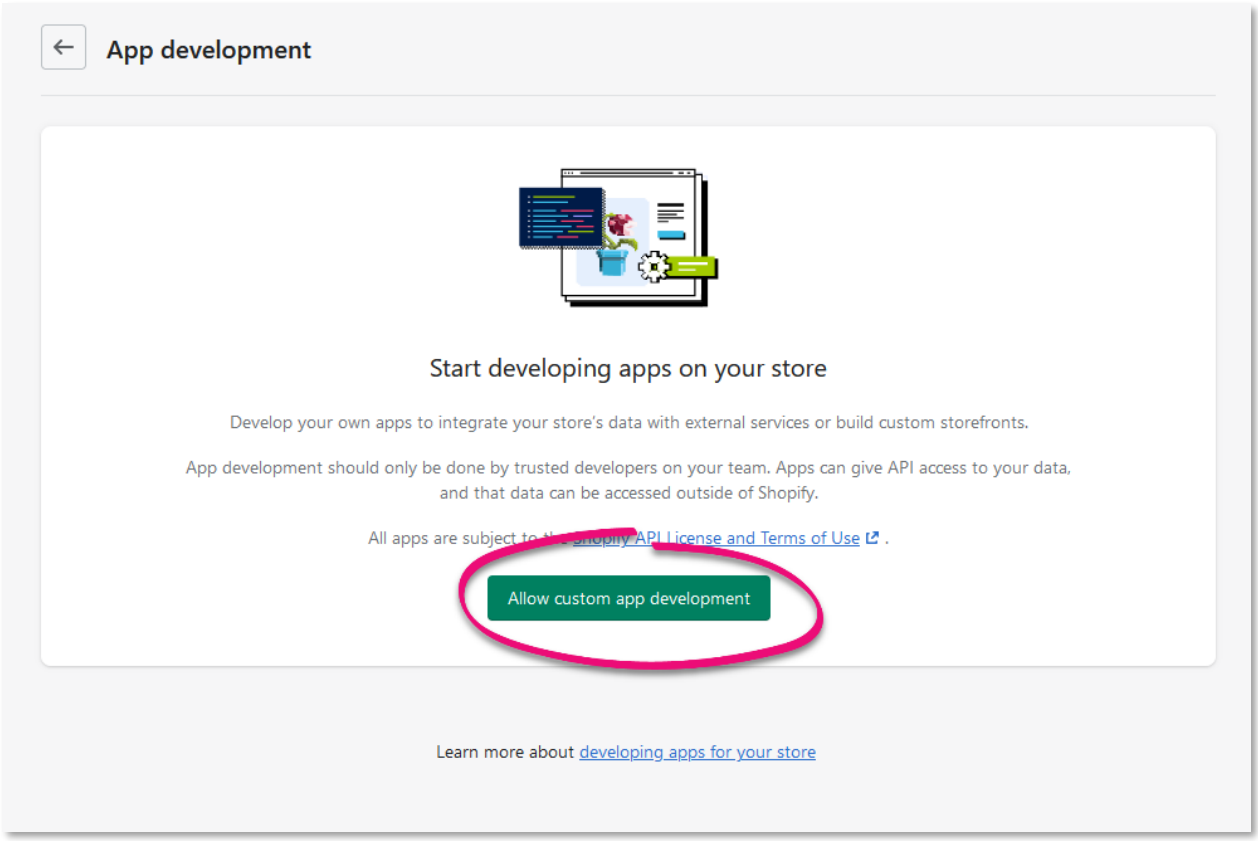
1. Sign into the Shopify admin dashboard
2. On the menu on the left, click on Apps



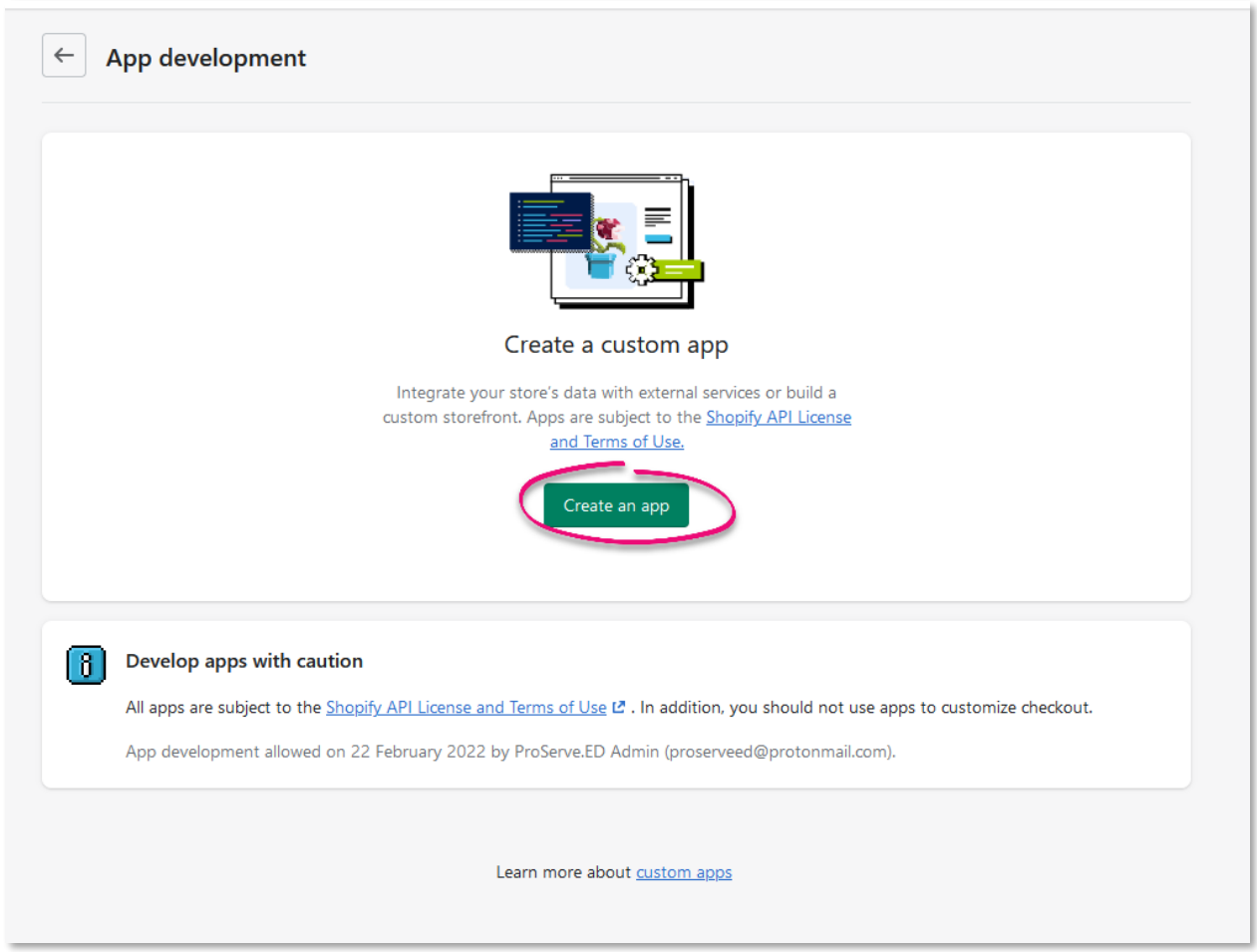
3. Click on Develop apps for your store:



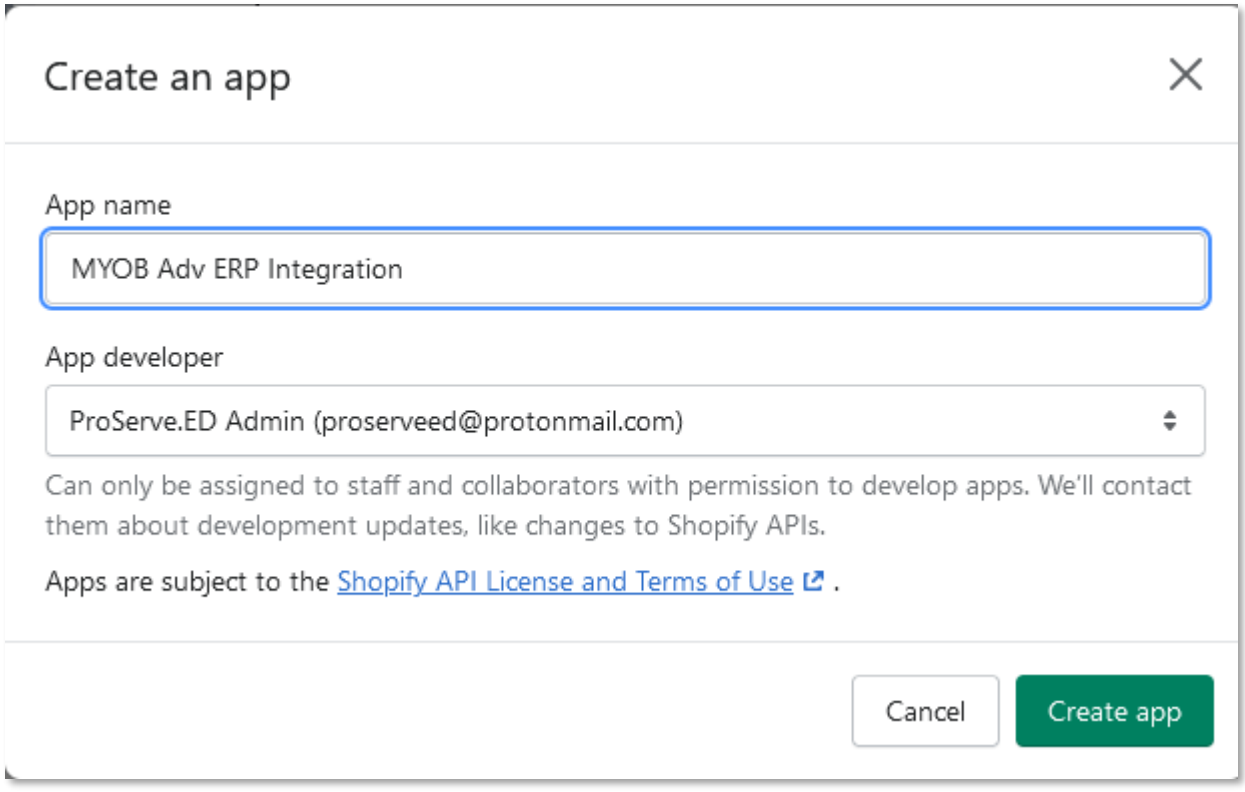
4. Click “Allow Custom App Development”. Shopify will prompt you to the next screen where you’ll have to agree on some the terms and conditions. Click “Allow Custom App Development” again to agree.



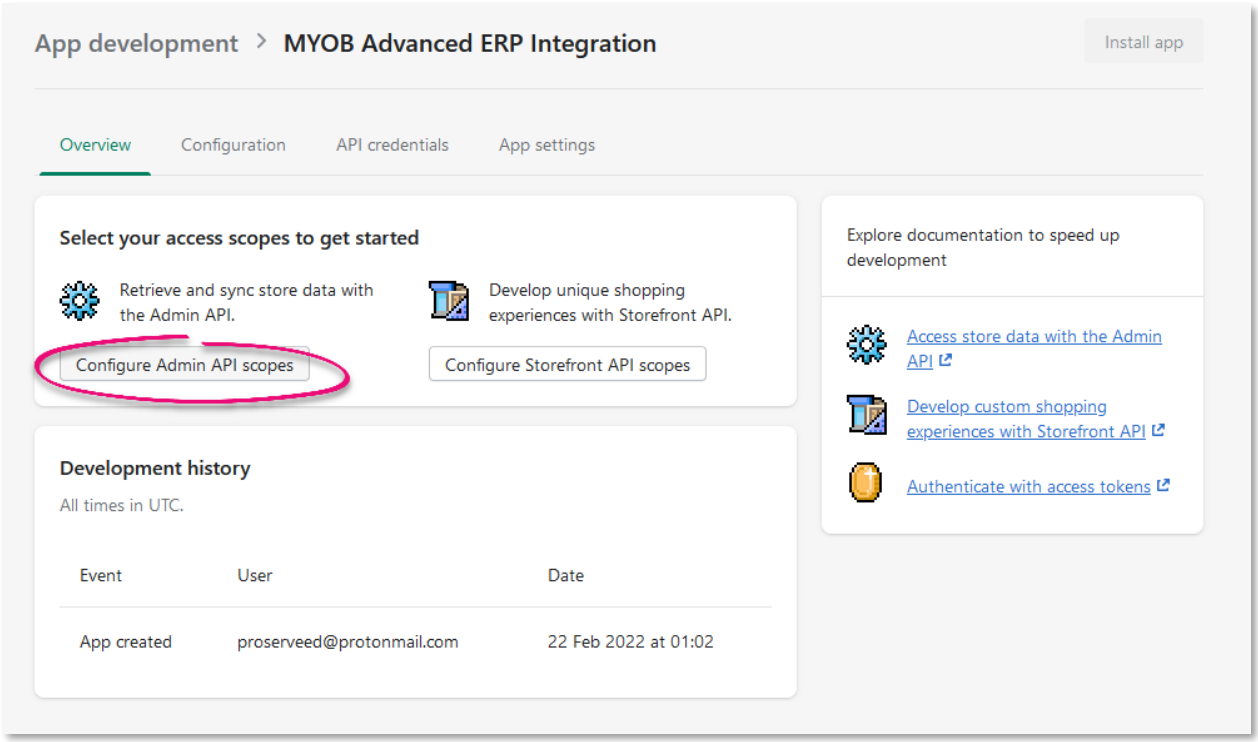
5. In the App Development page, click “Create an app”



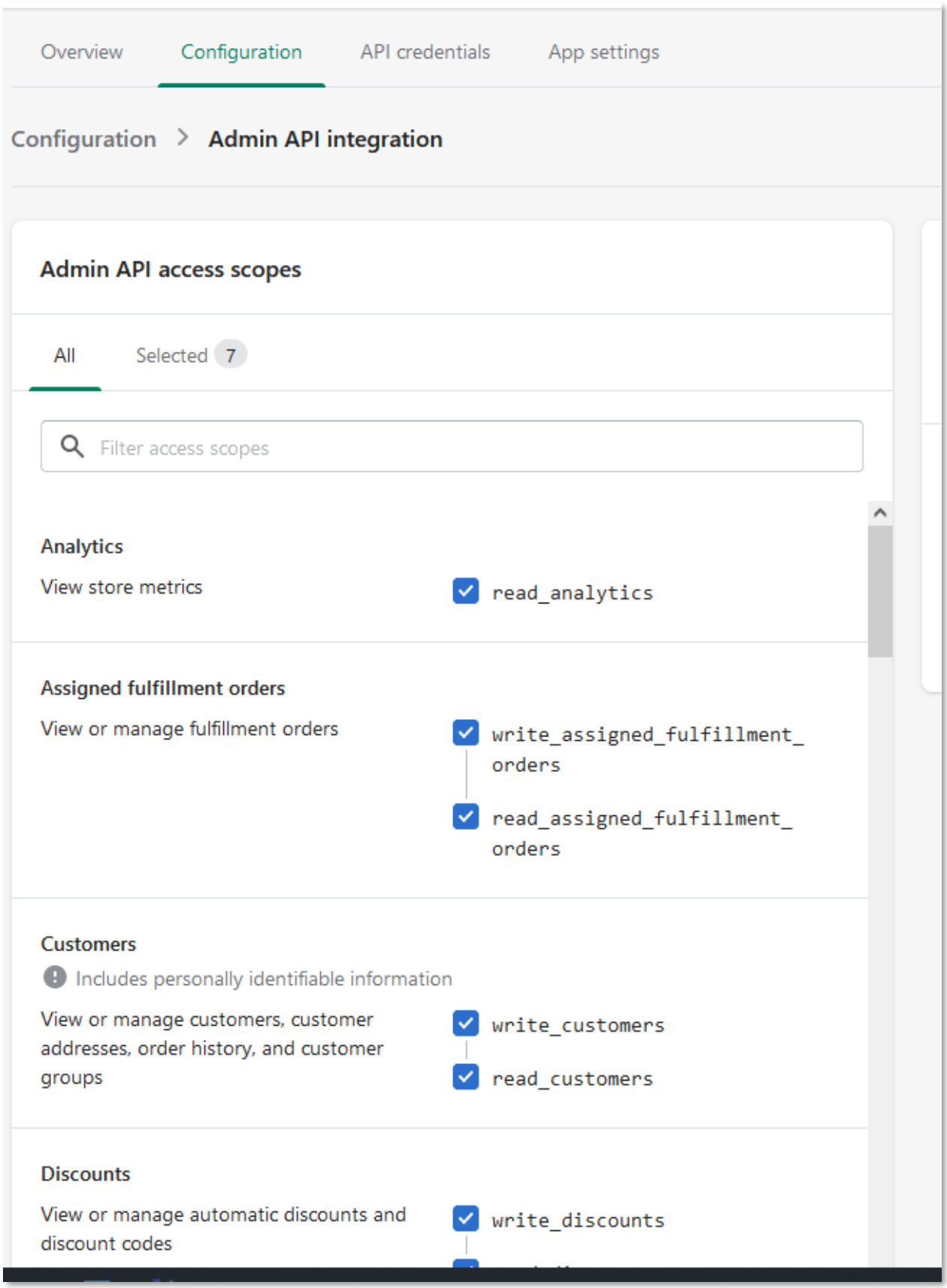
6. Enter an app name, then click Create App.



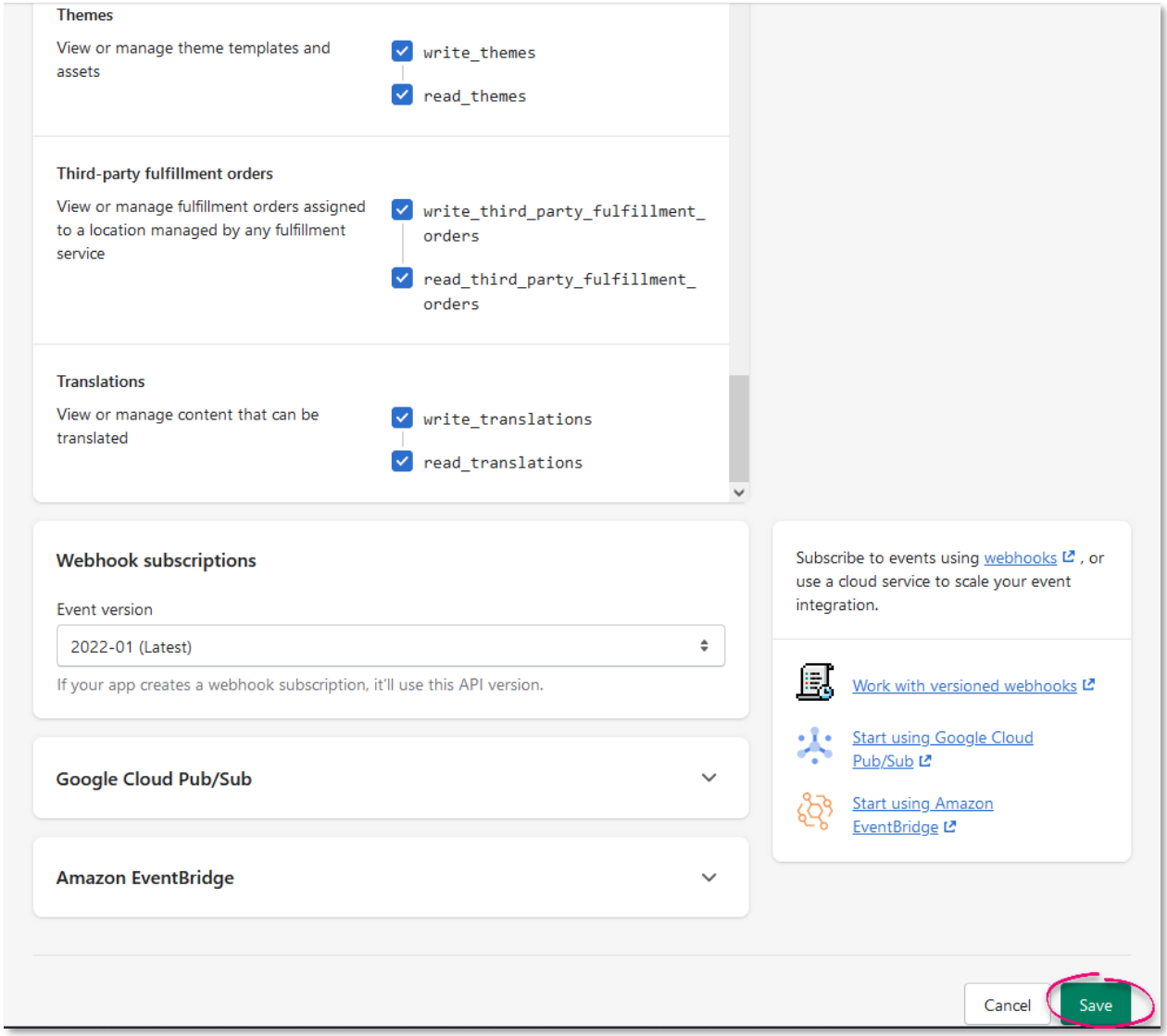
7. In the Overview tab, click Configure Admin API scopes:



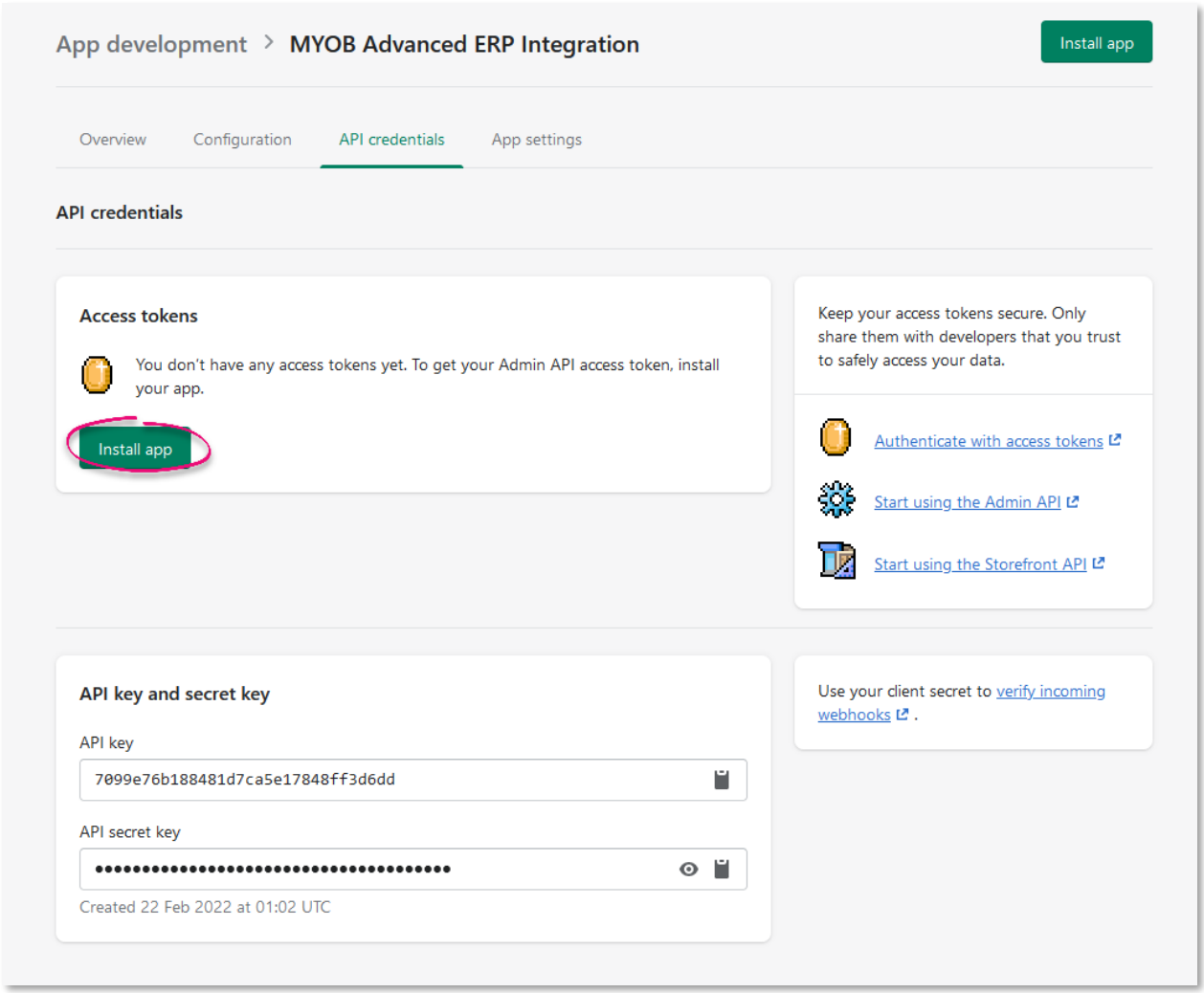
8. In the Admin API Access Scope, grant all access rights to your app. If you don't want to grant all access rights, MYOB Advanced Shopify Integration requires access rights in at least: Customers, Addresses, Orders, Products, Refunds, Fulfillments, Discounts, Files & Images, Gift Cards, Inventory, Locations, Payments, Transactions, Prices, Locales;



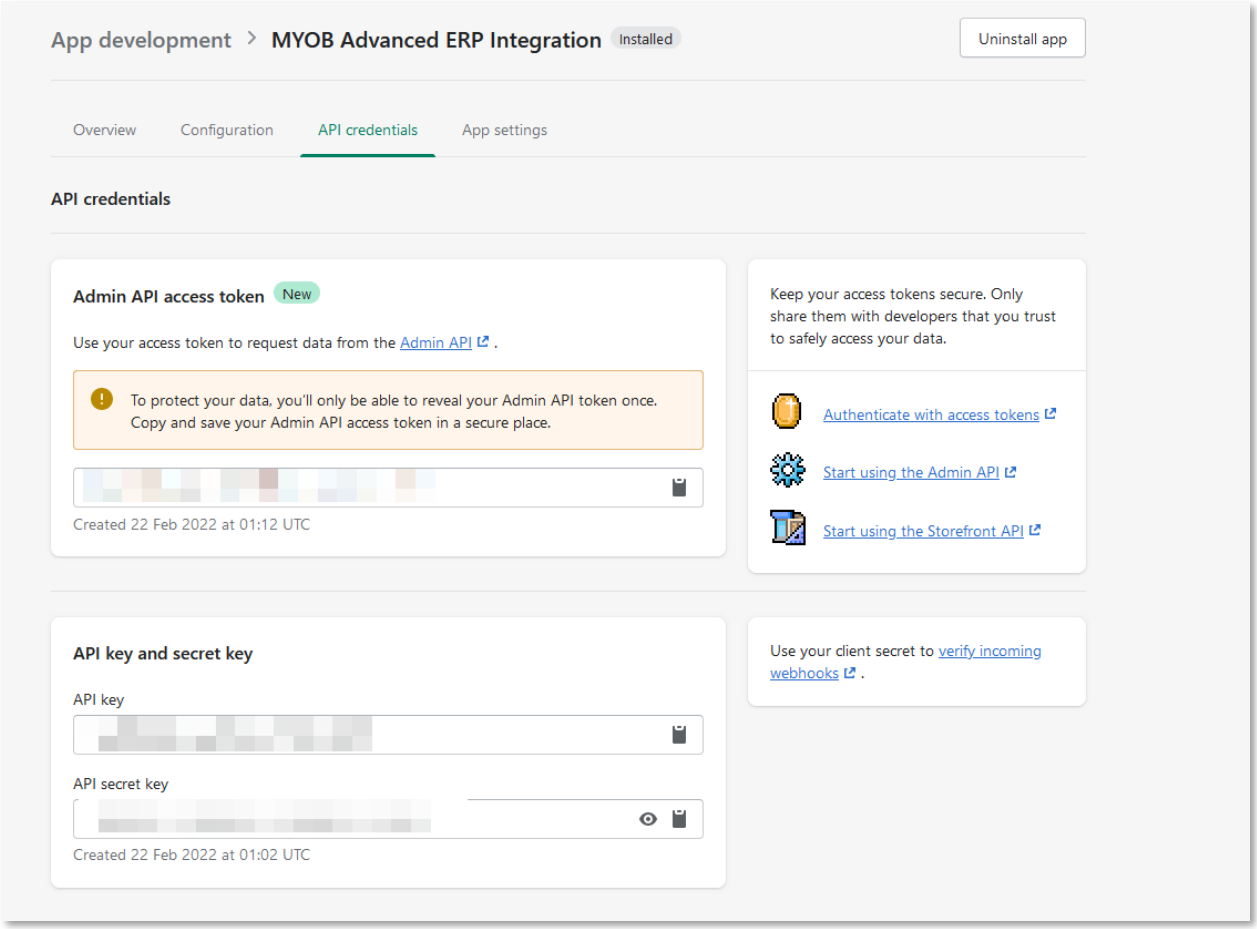
9. In the Webhook Subscription area, leave the default value. Then press Save:



10. In the API Credentials tab, click Install App:

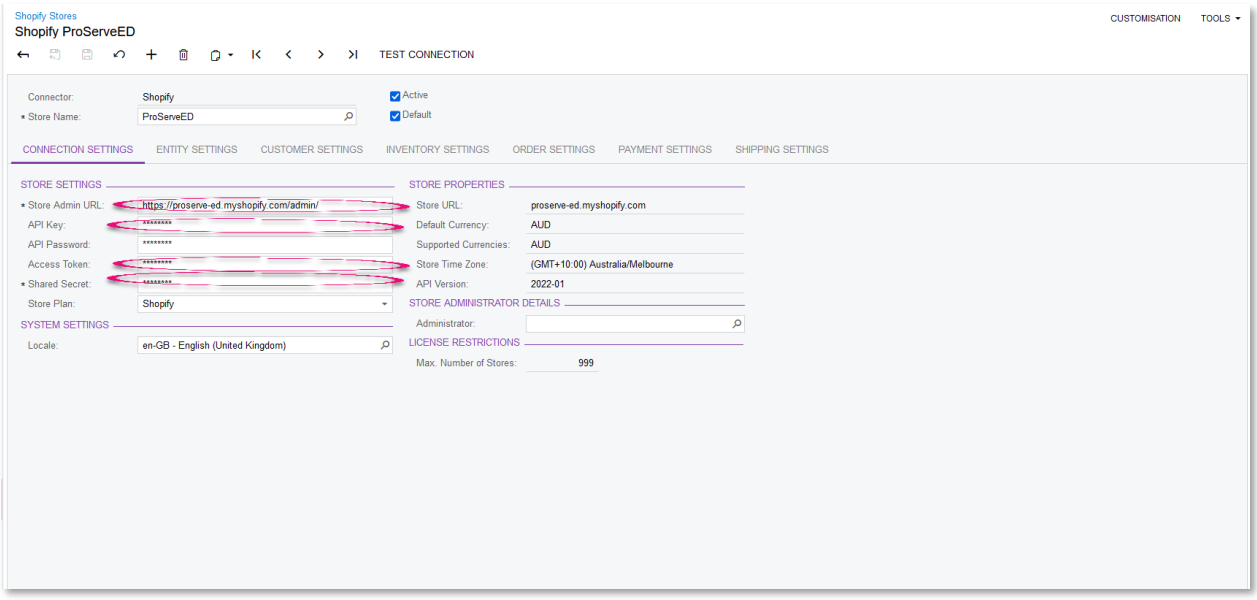


11. Once you've installed the app, an access token is created. Note down the Access Token, API Key and API secret key for the app and store them at a secure place.

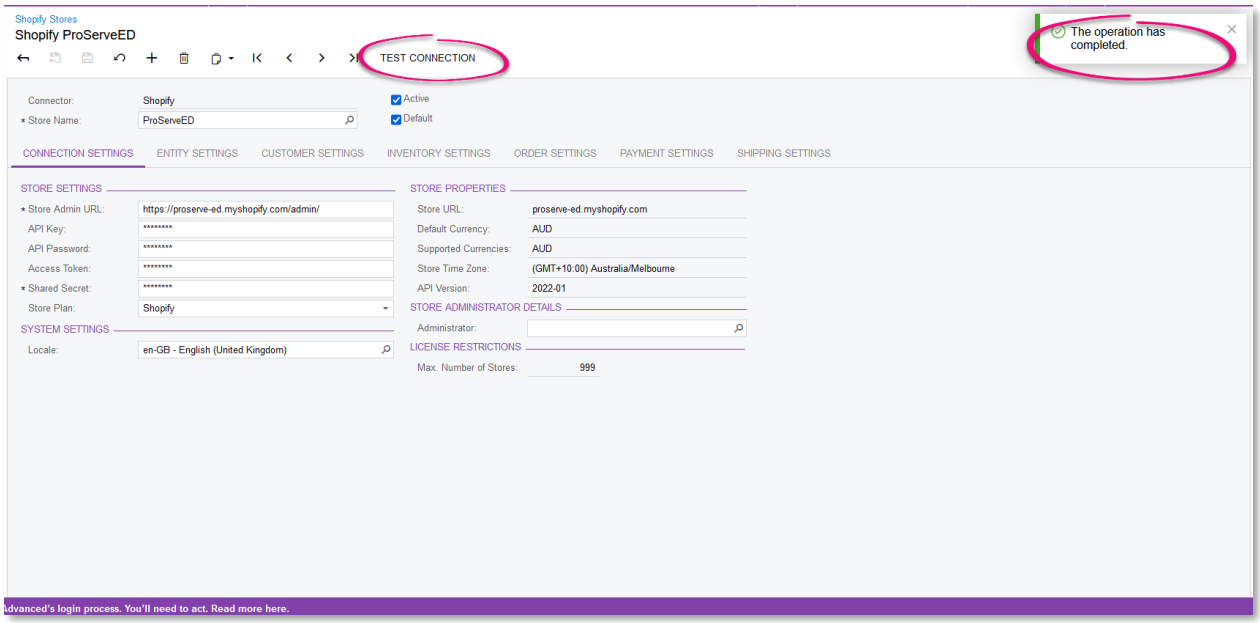


Establish a connection between MYOB Advanced and the Shopify store

1. Go to the Shopify Stores screen (BC201010), create a new Shopify Store. Enter the Store admin URL, API Key, Access Token (Admin API Access Token from Shopify) and Shared Secret (API Secret from Shopify) – notice the API Password field is not touched:



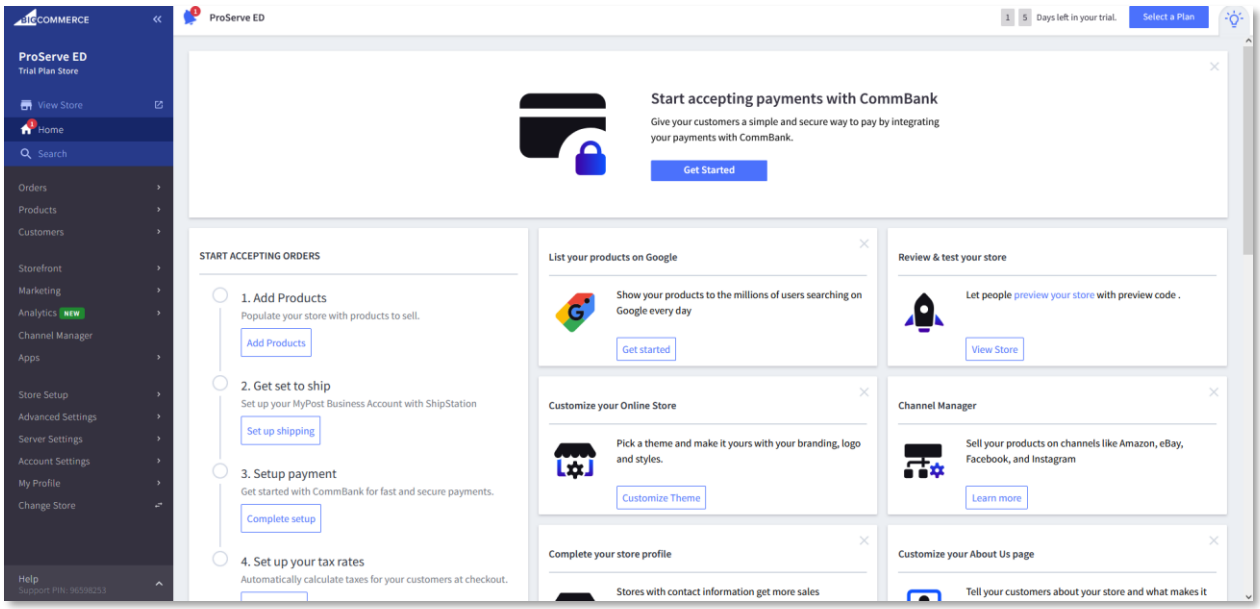
2. Click “Test Connection” – if everything above is done correctly, the Test Connection should be successful:



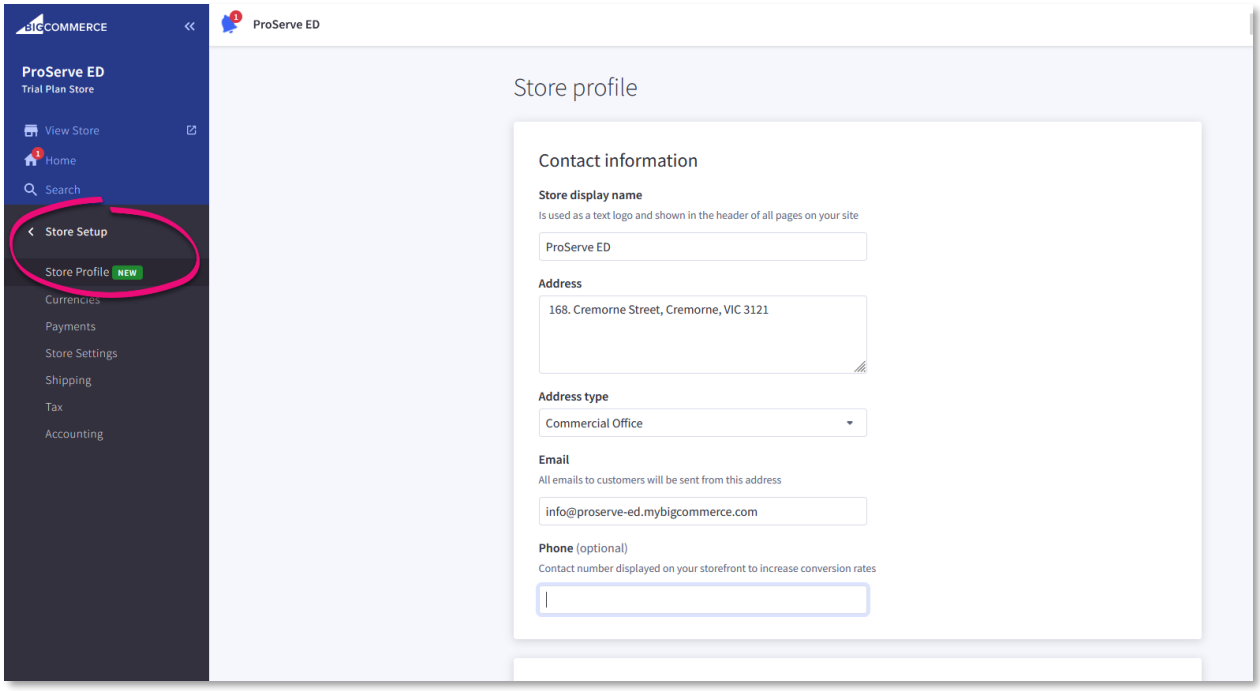
Establish Connection between MYOB Advanced and BigCommerce

Create a BigCommerce store and complete store configuration (optional if you have already had a store)

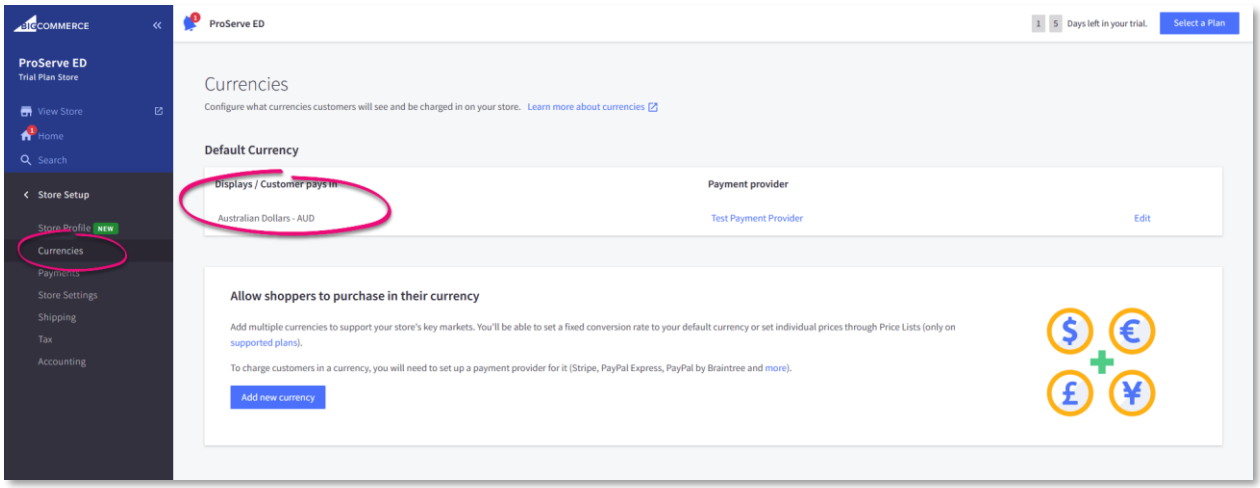
1. Go to BigCommerce website, sign up and set up for a new BigCommerce store. You can follow BigCommerce’s instruction here.
2. Once you have signed up for a store, the store area should look like this:



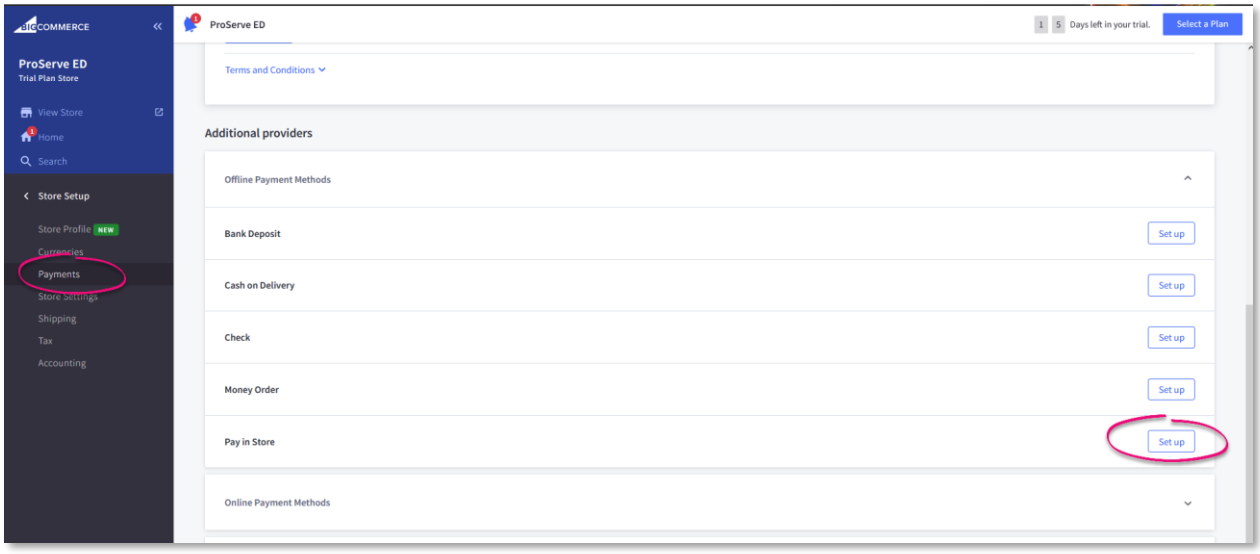
3. On the left navigation pane, select Store Setup > Store Profile. Enter the necessary store information such as address, phone number, country, etc. It's very important to ensure the store's country is correct because it affects a range of shipping and payment methods for your store.



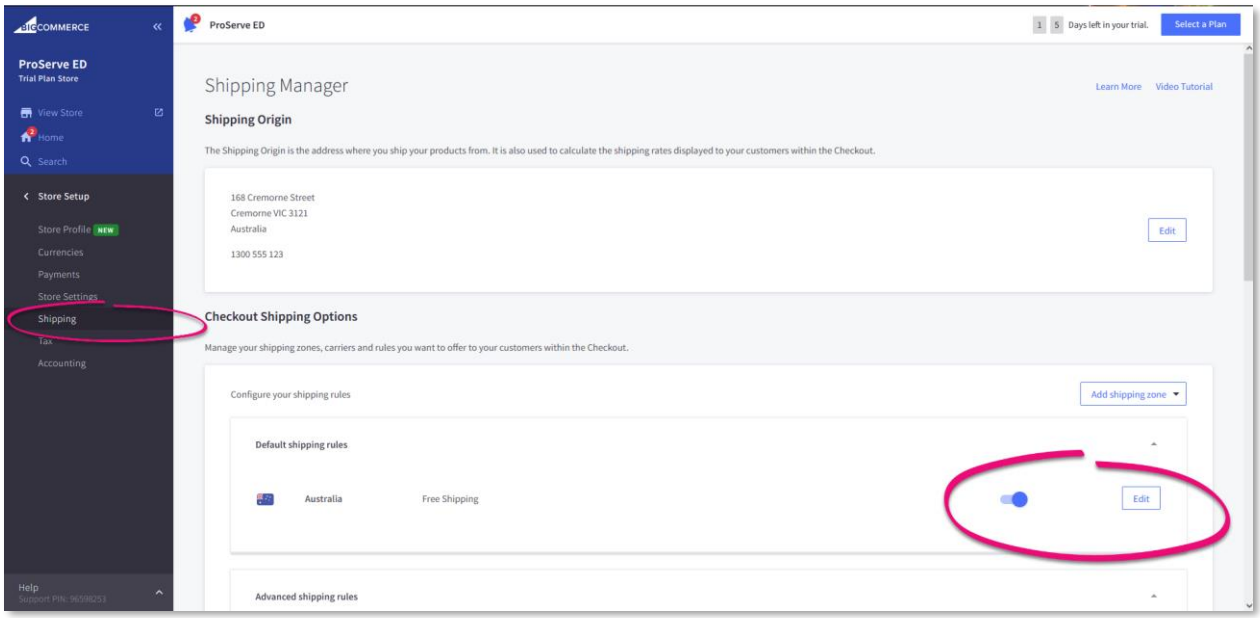
4. In Store Setup > Currencies, make sure that the default currency is the currency that you want to display the price for the store, or wants the customer to pay in that currency. If the default currency here is not the currency you want, then you need to add that currency in, and then set the new currency as the default currency:



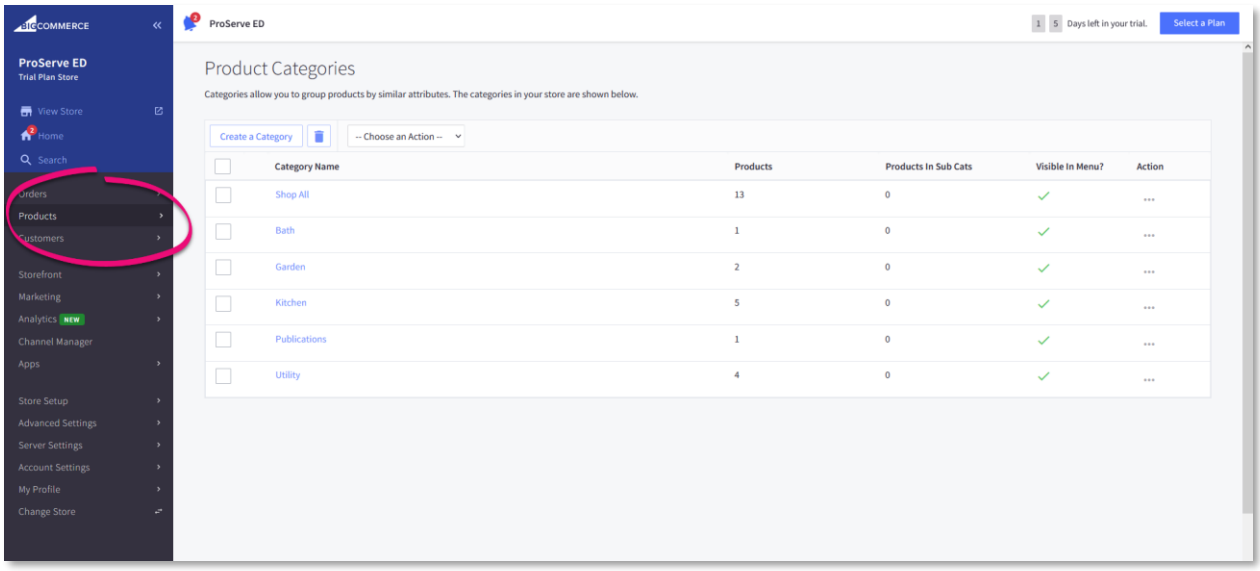
5. In Store Setup > Payments. Review the payment method for your store. If required, you can set up additional payment methods, for example, we'll be adding the Pay In Store payment method here for testing purposes.



6. In Store Setup > Shipping, review the Shipping Zones and Shipping Rules for your store as required. You can add free shipping for a Shipping Zone here as well:

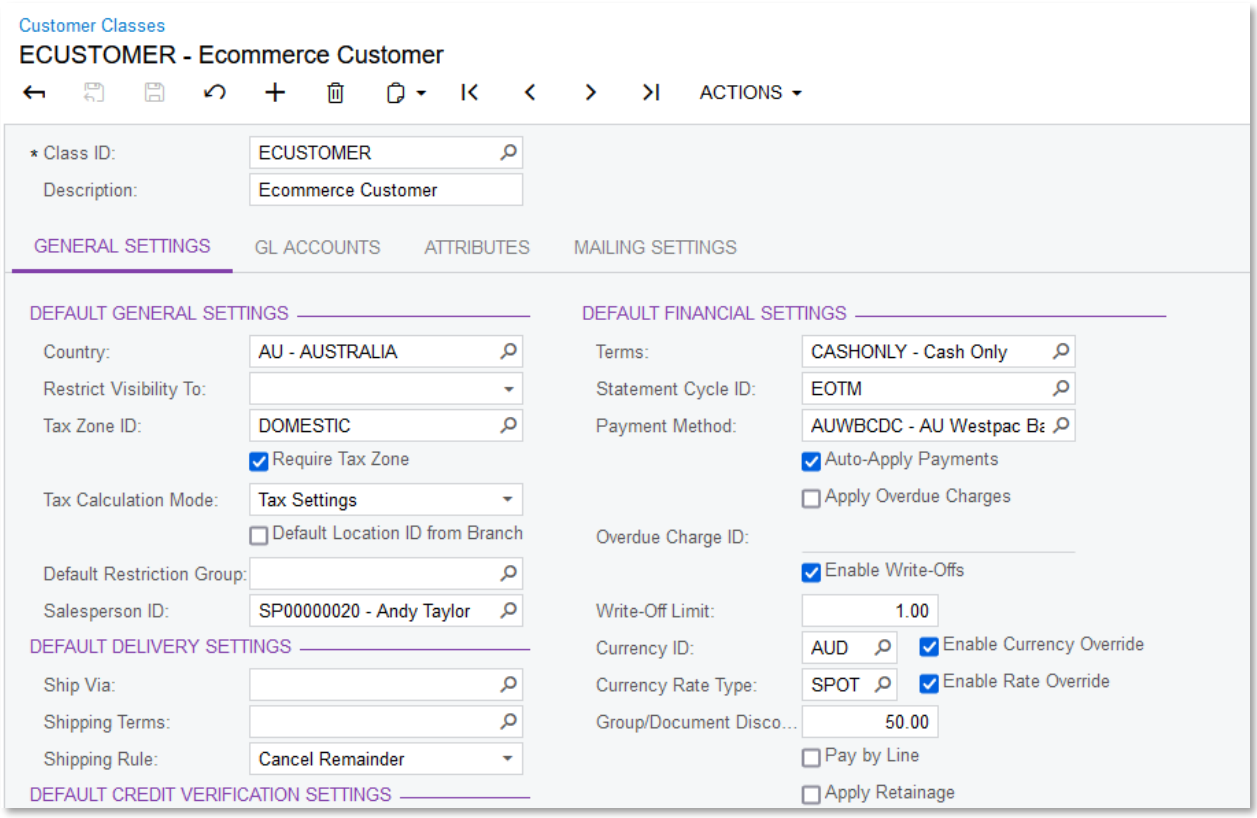


7. If required, delete the existing customers, customer groups, products and product categories, and archive existing sales orders that is part of BigCommerce's demo data in the relevant areas:

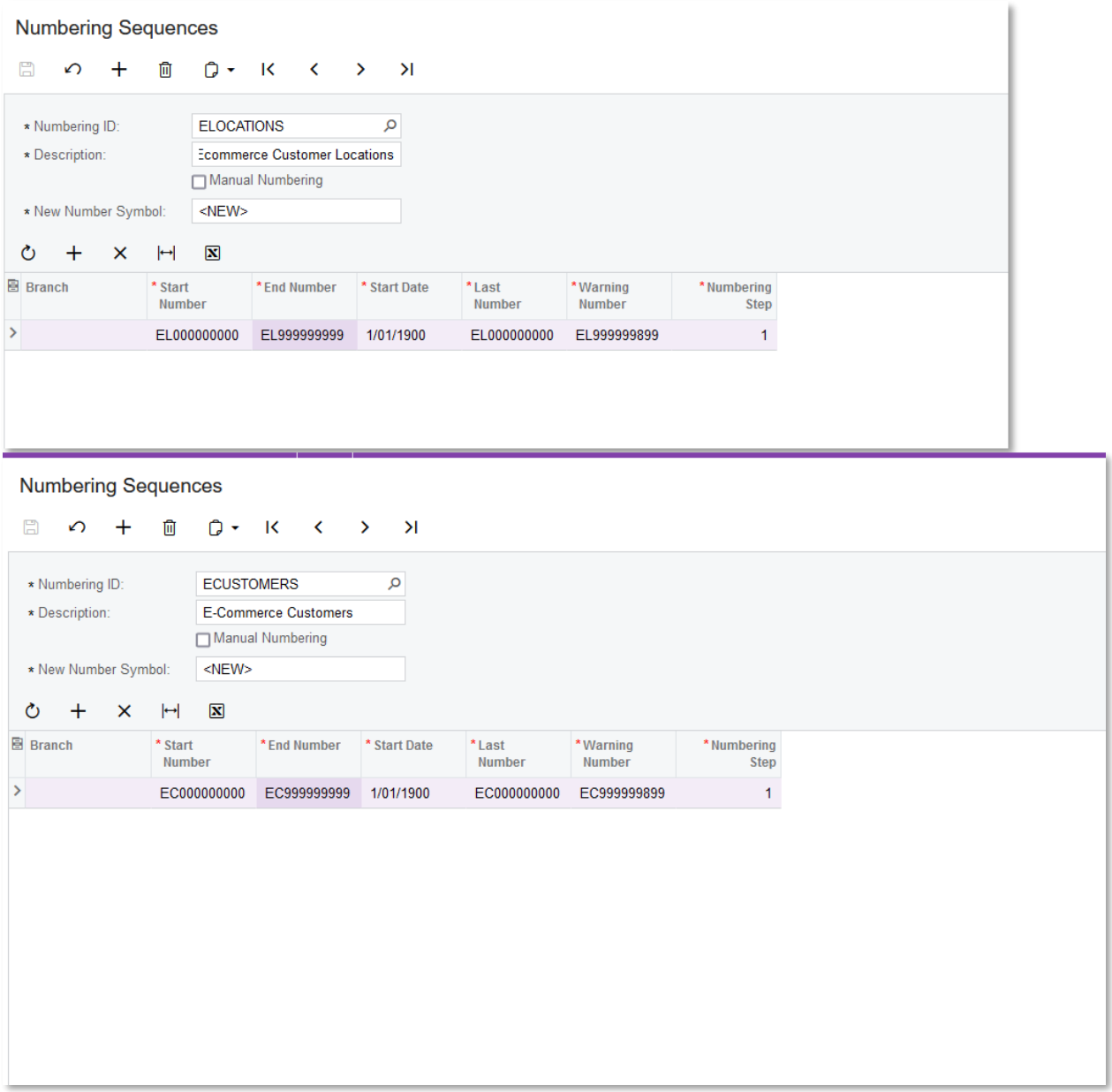


Configure myob advanced erp for integration

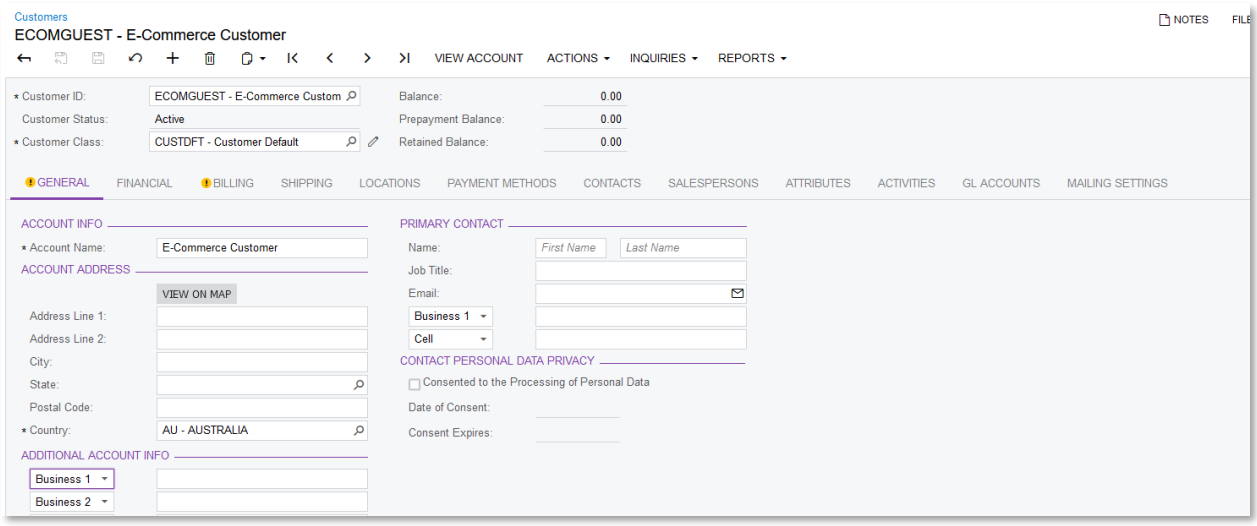
1. Go the screen Enable/Disable features (CS100000), ensure at least the below features are enabled:
    - Inventory
    - Business Account Locations
    - Commerce Integration
    - BigCommerce Connector
2. In the Customer Classes screen (AR201000), create a new Customer Class for BigCommerce Customer (e.g. we'll be calling this customer Class ECUSTOMER):



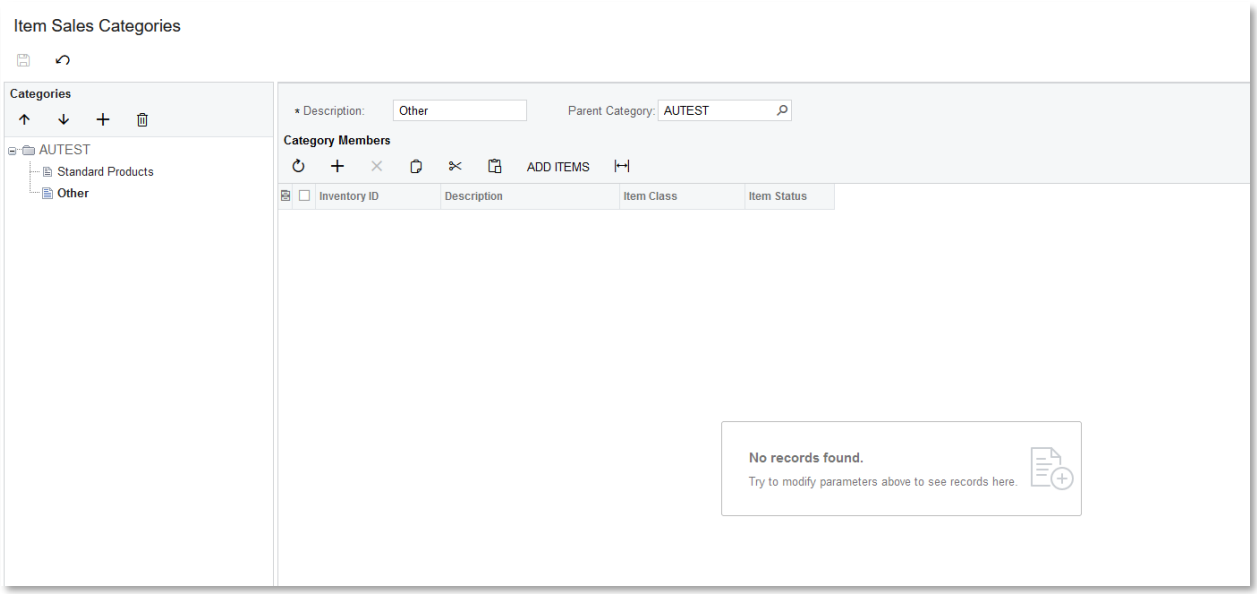
3. In the Numbering Sequences screen (CS201010), create new numbering sequences that will be used for importing new customers and customer locations from BigCommerce:



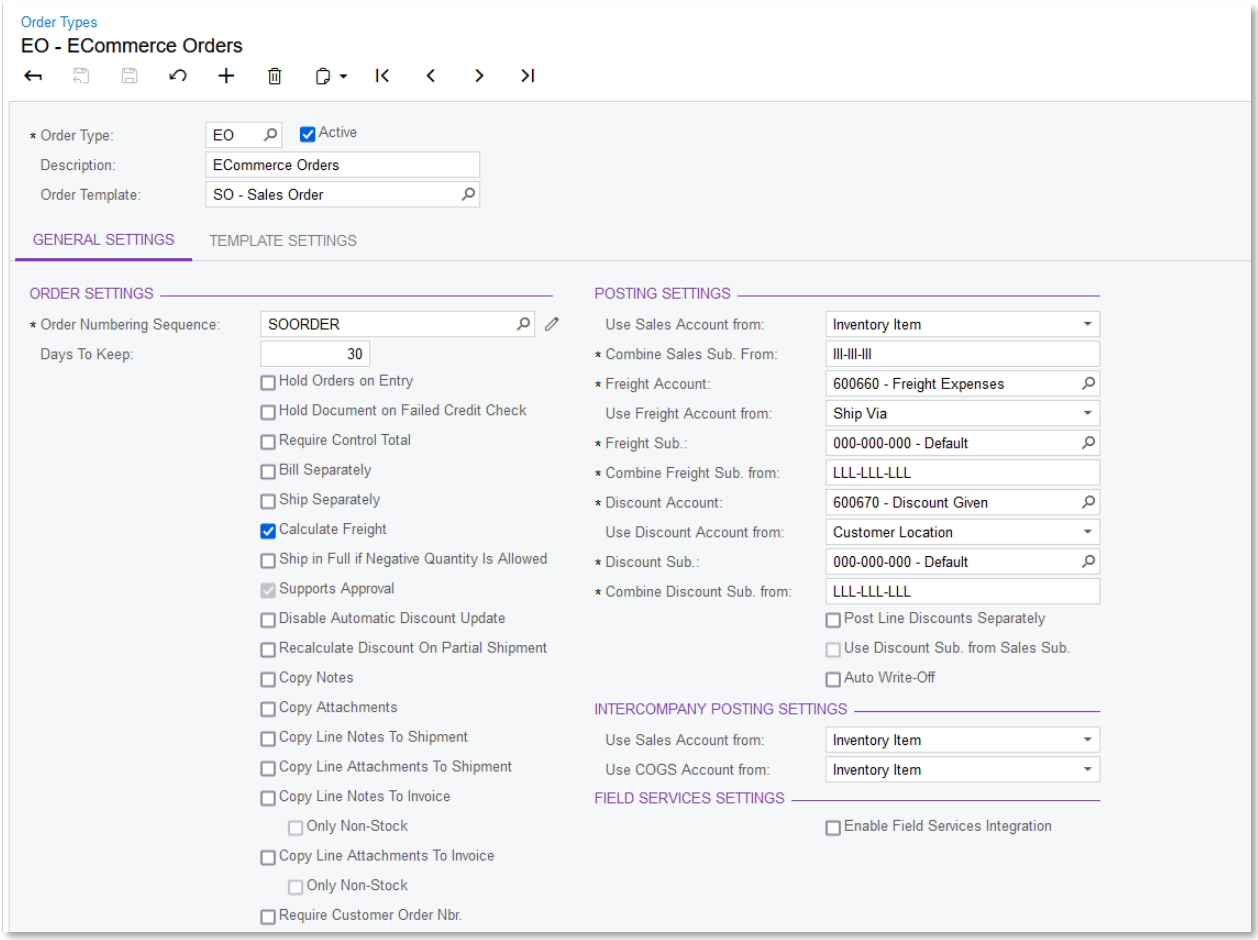
4. Create a customer to be used for guest orders from BigCommerce:



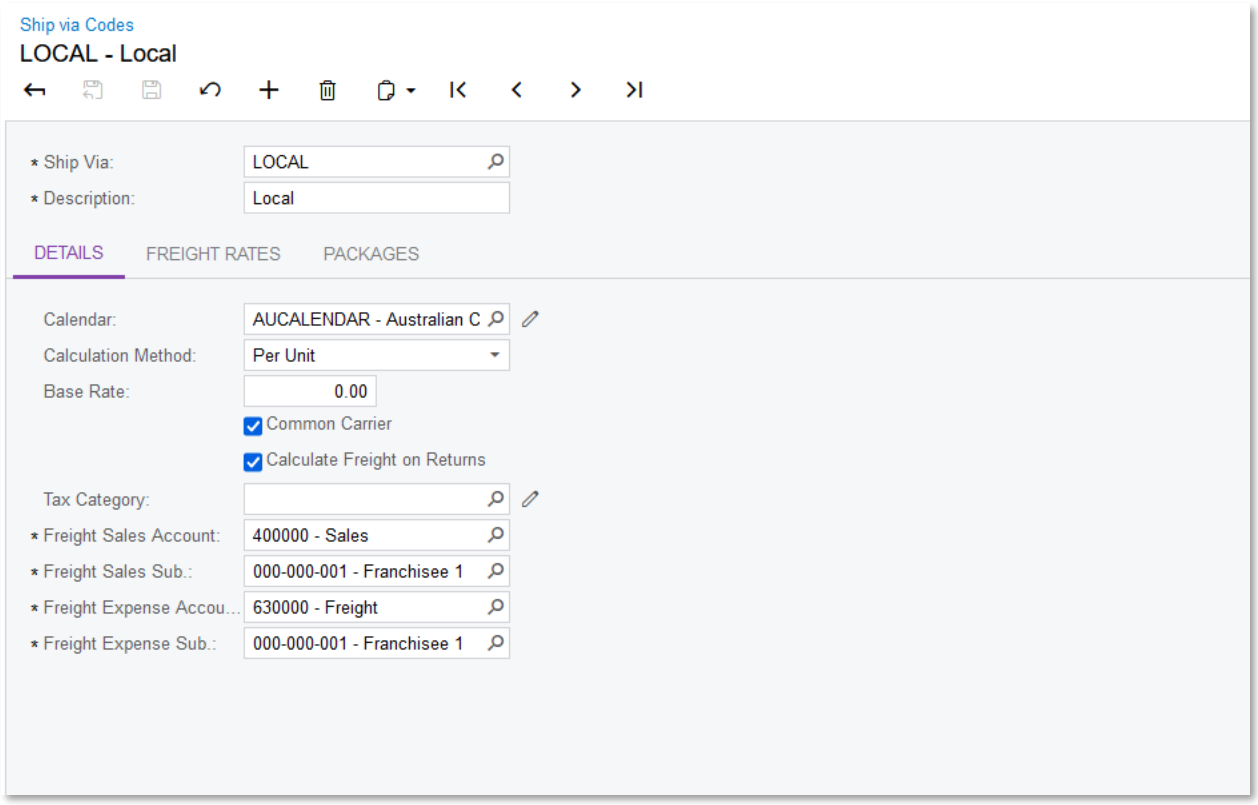
5. In the Item Sales Categories screen, create a new Item Sales Category called “Other” which will be used for Stock and Non-stock items that are imported from BigCommerce that don’t have their own categories



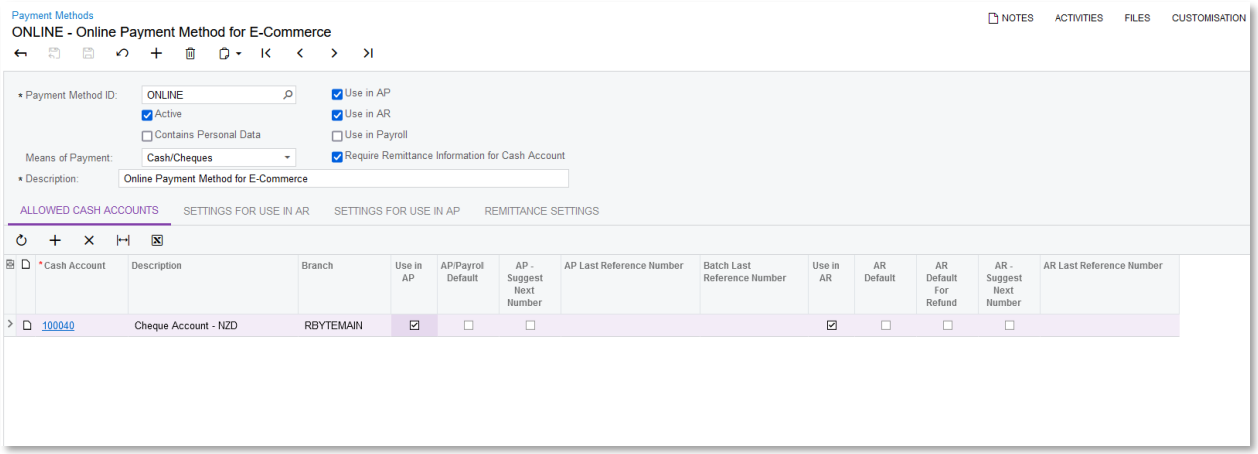
6. In the Order Types screen (SO201000), create two Order Types, one for E-Commerce Sales Orders which is based on the SO Order Type template, the other one is for E-Commerce Return Orders which is based on the RC Order Type template.



7. In the Ship via Code screen (CS207500), review and create a Ship via Code if necessary for sales orders from BigCommerce:

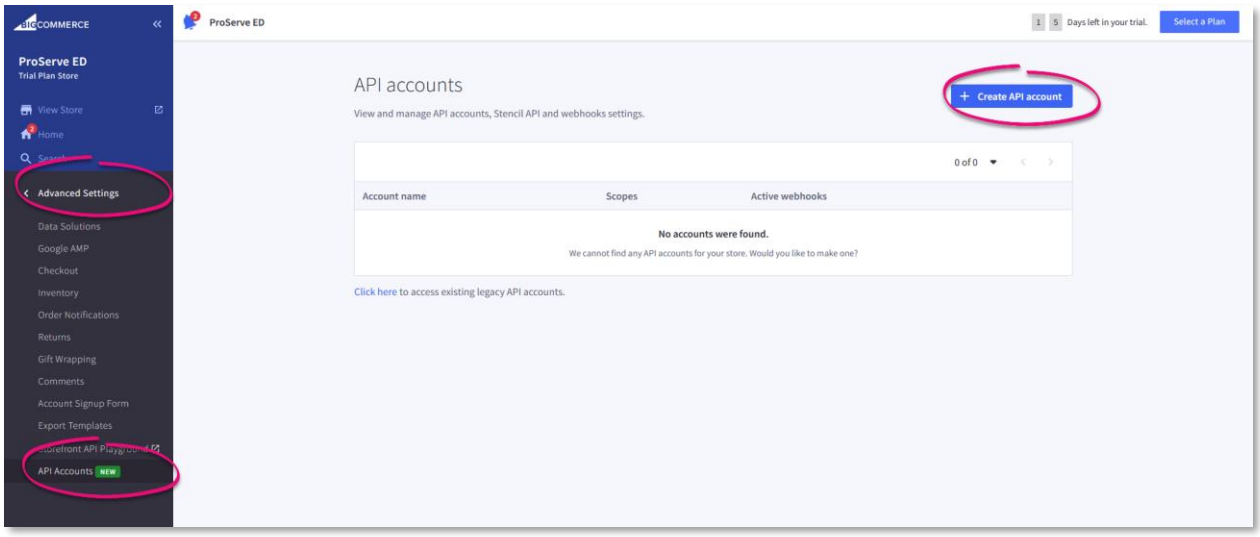


8. Create a new Payment Method for BigCommerce, add a Cash Account into that Payment Method and also, assign that Payment Method as a default payment method to the guest commerce customer we created above (step 4, section B).

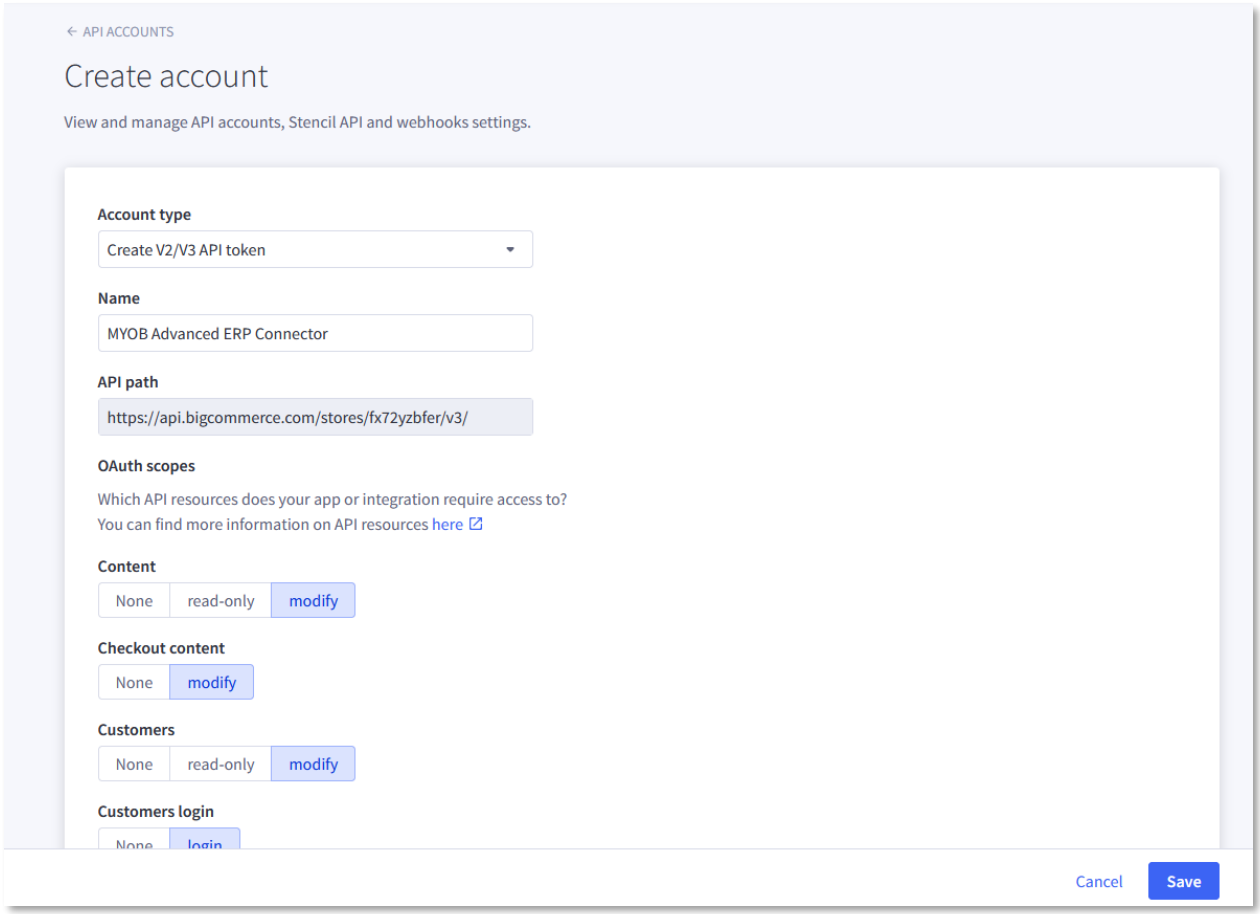


Capture store connection information

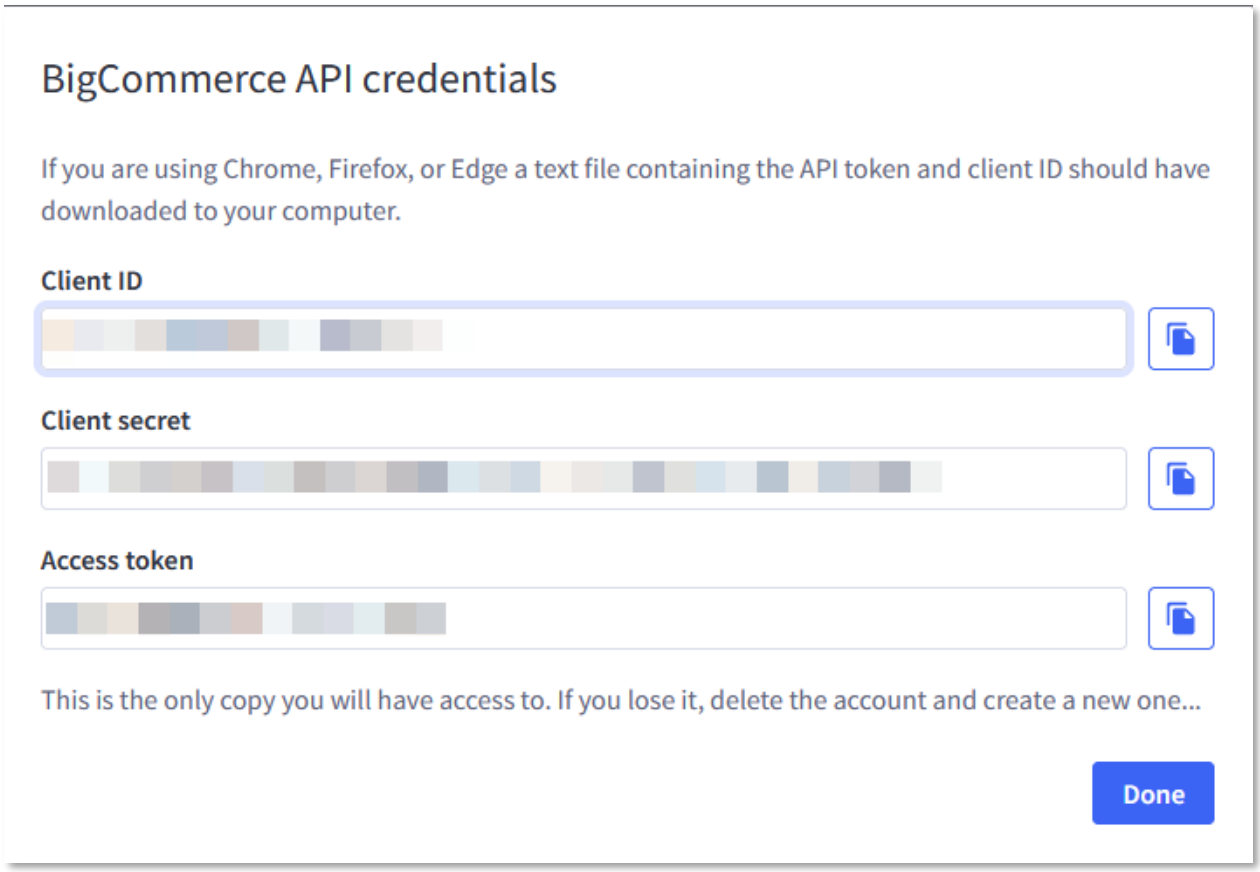
1. Log on to your store admin page by entering https://<yourStoreURL>/manage/ - in the navigation menu on the left, select Advanced Settings > API Accounts and select Create API Account:



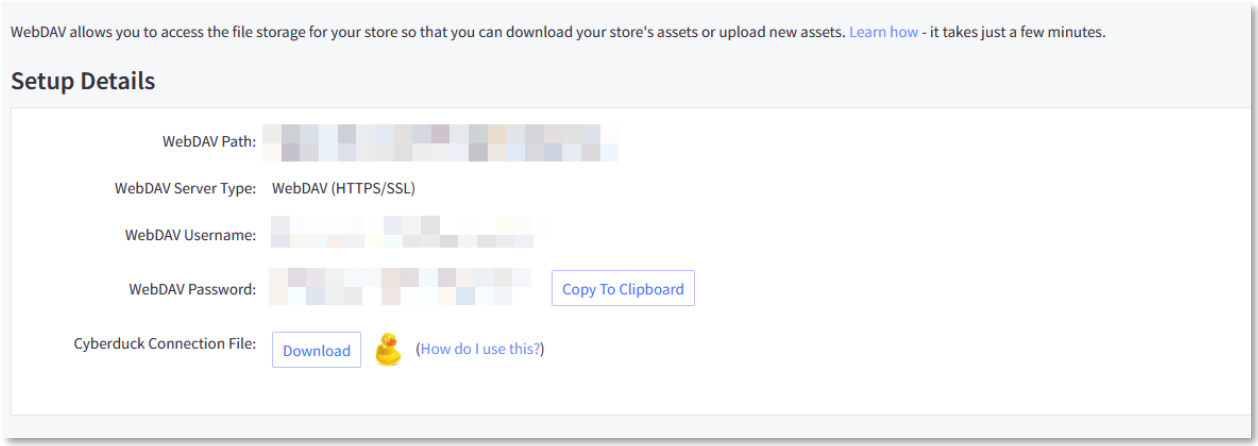
2. Select Account Type = Create V2/V3 API token. Enter a name for the API account. In the OAuth scopes area, set the highest (rightmost) access rights to every setting. Click Save when finished.



3. After you click save, BigCommerce will generate the Client ID, Client Secret, and Access Token. Copy these values and store them in a safe place. In this step, if you're using Chrome or Firefox or Edge, a download for a .txt file will also be automatically triggered – **it's very important to save this text file because it contains your API Path which will be used for establishing the connection between the store and the ERP.**

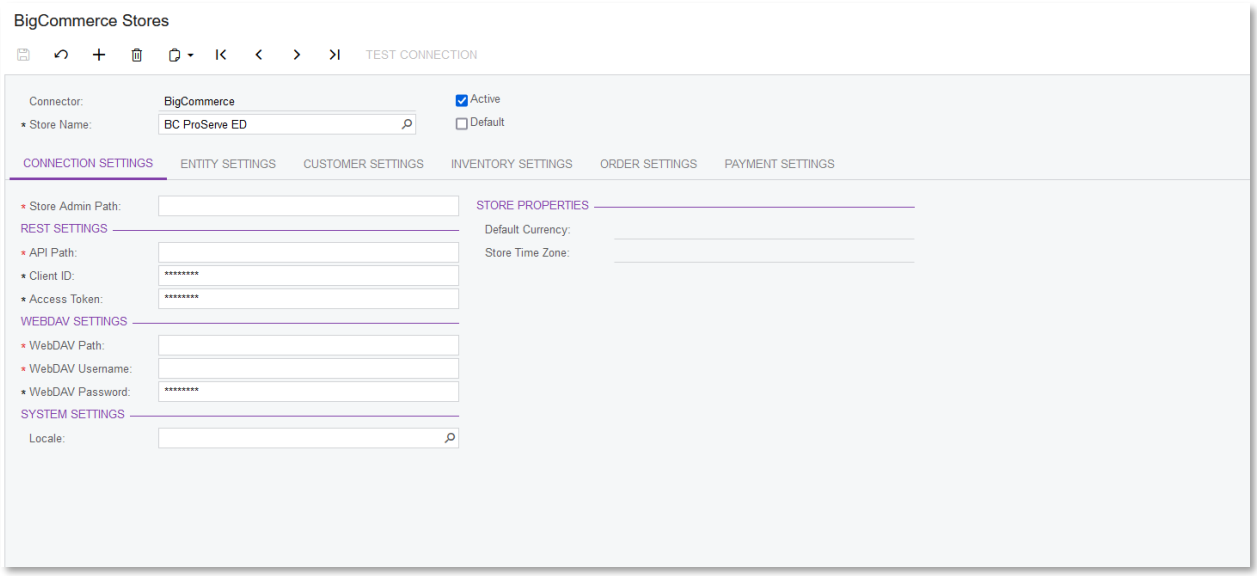


4. Go to Server Settings > File Access (WebDAV) an copy the WebDAV details for the store and save them in a secure place:

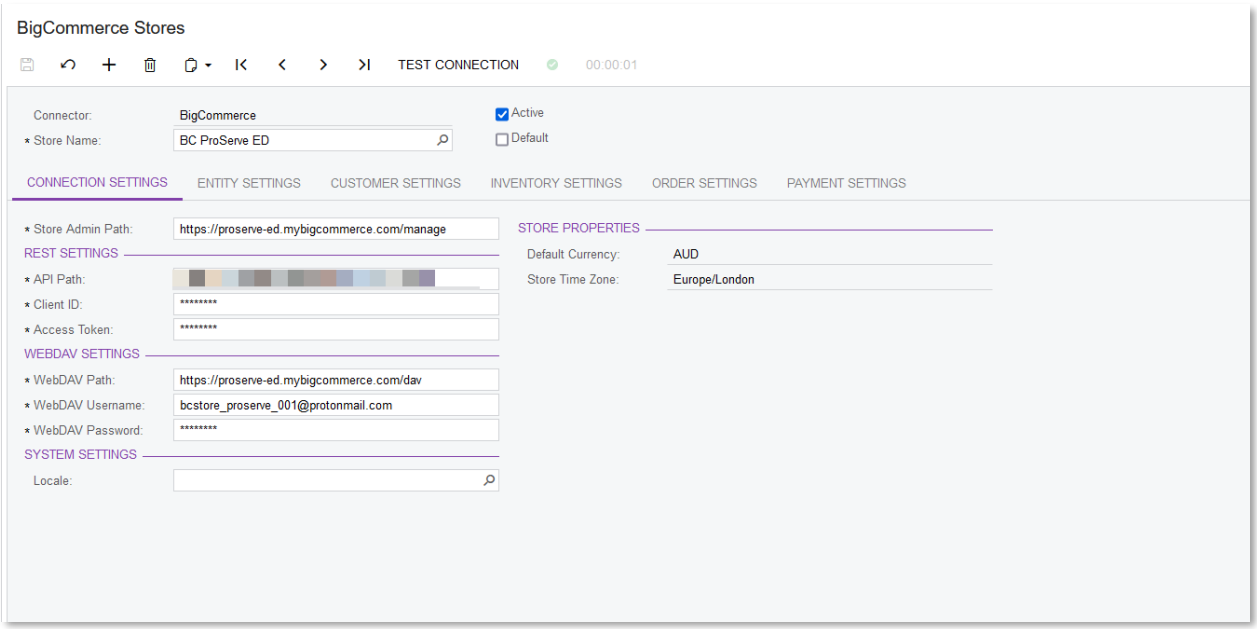


Establish connection between MYOB Advanced and BigCommerce

1. In MYOB Advanced, go to the BigCommerce Store screen (BC201000), create a new store.



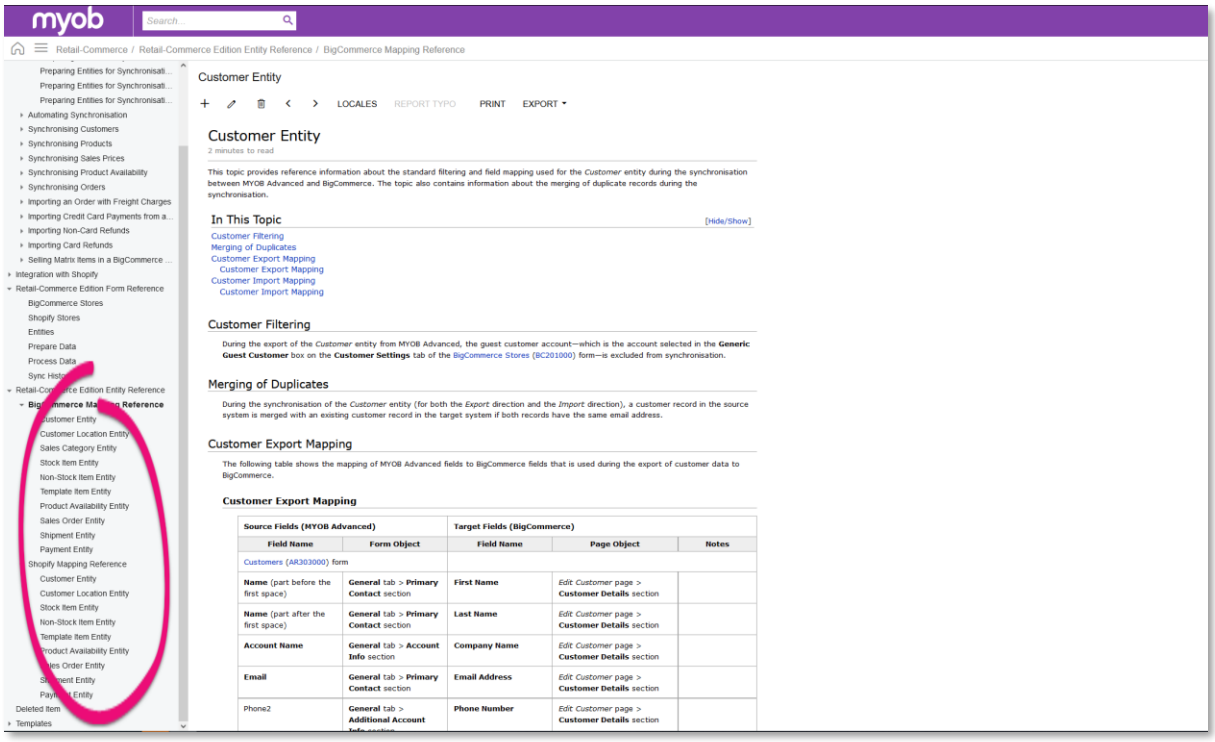
2. Enter the Store Admin Path, which is the Store URL and the /manage suffix, for example: https://<yourStoreURL>/manage, then enter all the information obtained from section C such as API Path, Client ID, Access Token, WebDAV Path, WebDAV username and Password into the store. Refer to the .txt file that was automatically downloaded when you created the API account above.
    - When entering the API Path, you need to enter the API path **without** the version



3. Once all information has been entered, click “Test Connection”. If the Test Connection is successful, that means you have successfully established the connection between BigCommerce and MYOB Advanced.

Standard field mappings

To reference which fields are mapped between BigCommerce/Shopify and MYOB Advanced as part of the synchronization, you can access the following pages in the in-product User Guide:

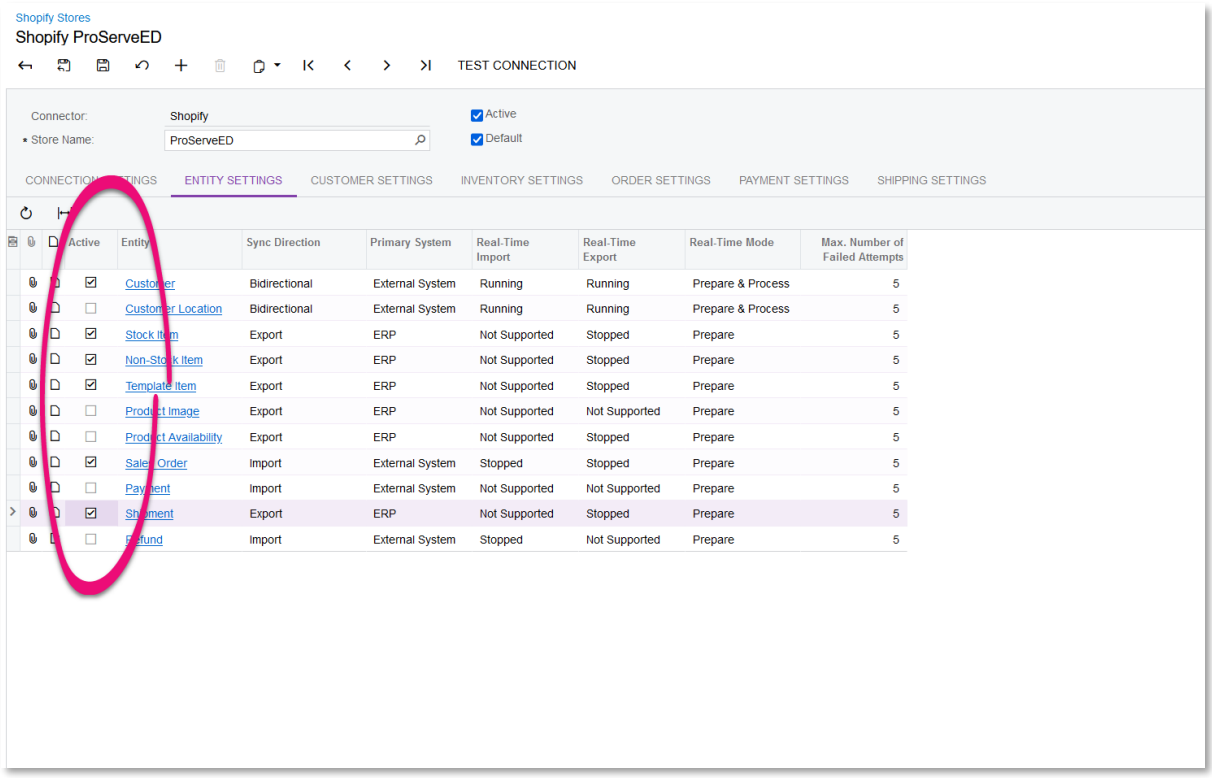


The first synchronisation

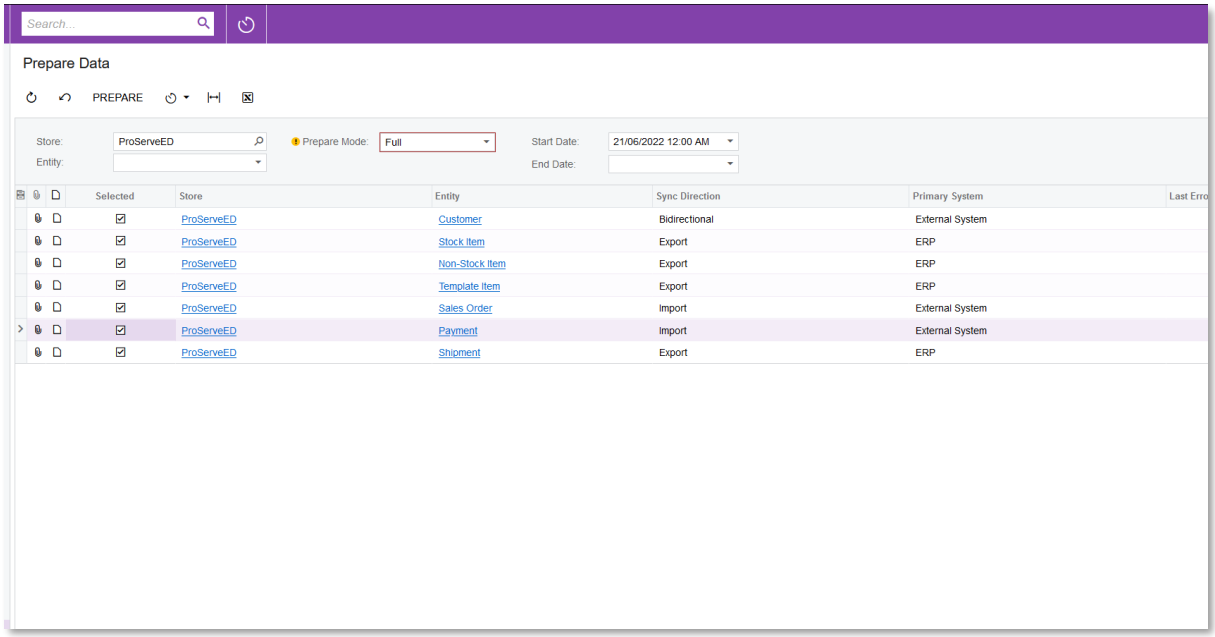
Before you start configurating Automation Schedules to automate the synchronisation process between the commerce platforms and MYOB Advanced, you should first perform a full synchronisation for each entity. Doing this will give you a better understanding of what entities will be synced between the systems, also error messages will surface for you to resolve.

To perform the first sync, go to the Shopify Stores (BC201010) or BigCommerce Stores (BC201000), in the Entity Settings tab, mark all the entities that need to be synced to Active.

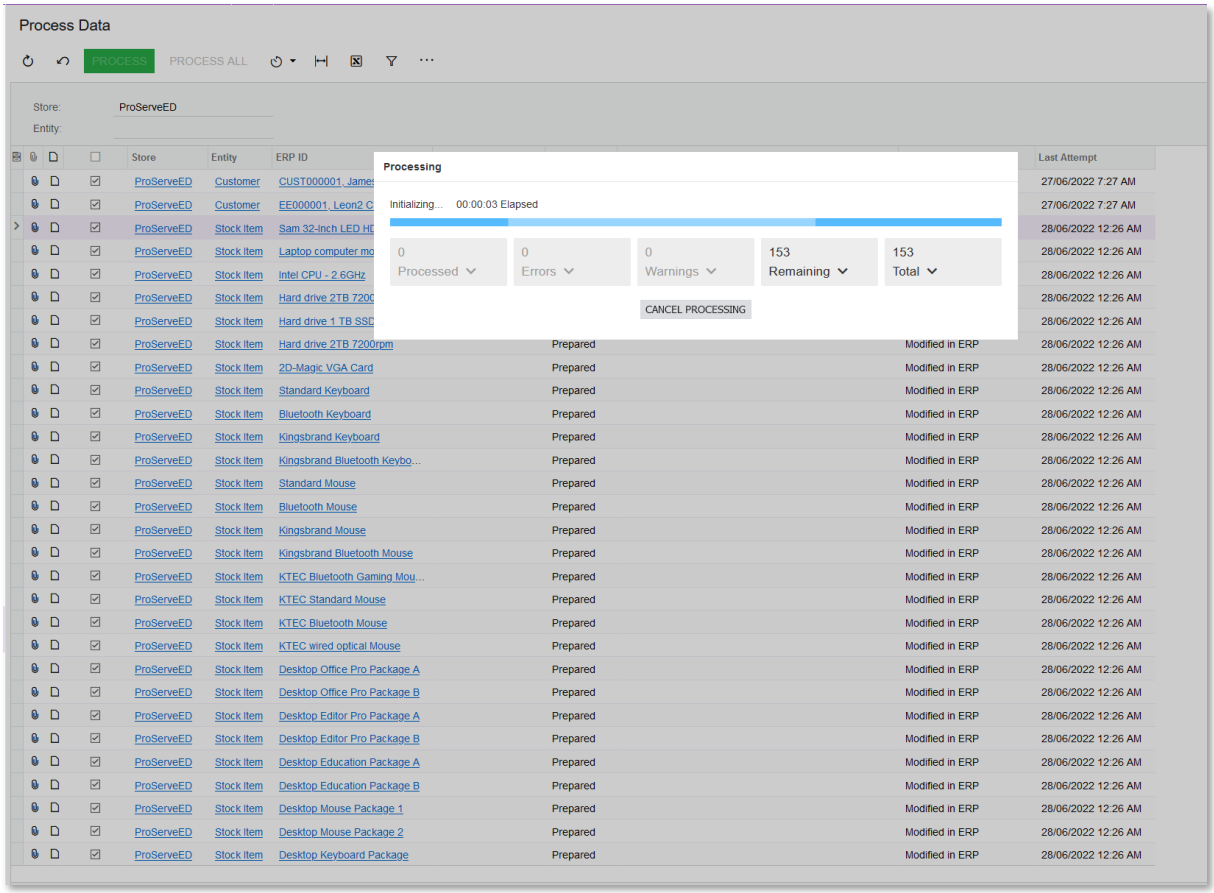
Tip: if you select an entity that is dependent on other entities, all of the related Entities will also be marked as Active. For example, when the Shipment entity is set as Active in the Shopify Stores screen, the Customer, Stock Item, Non-Stock Item, Template Item and Sales Order entities are also marked as Active.



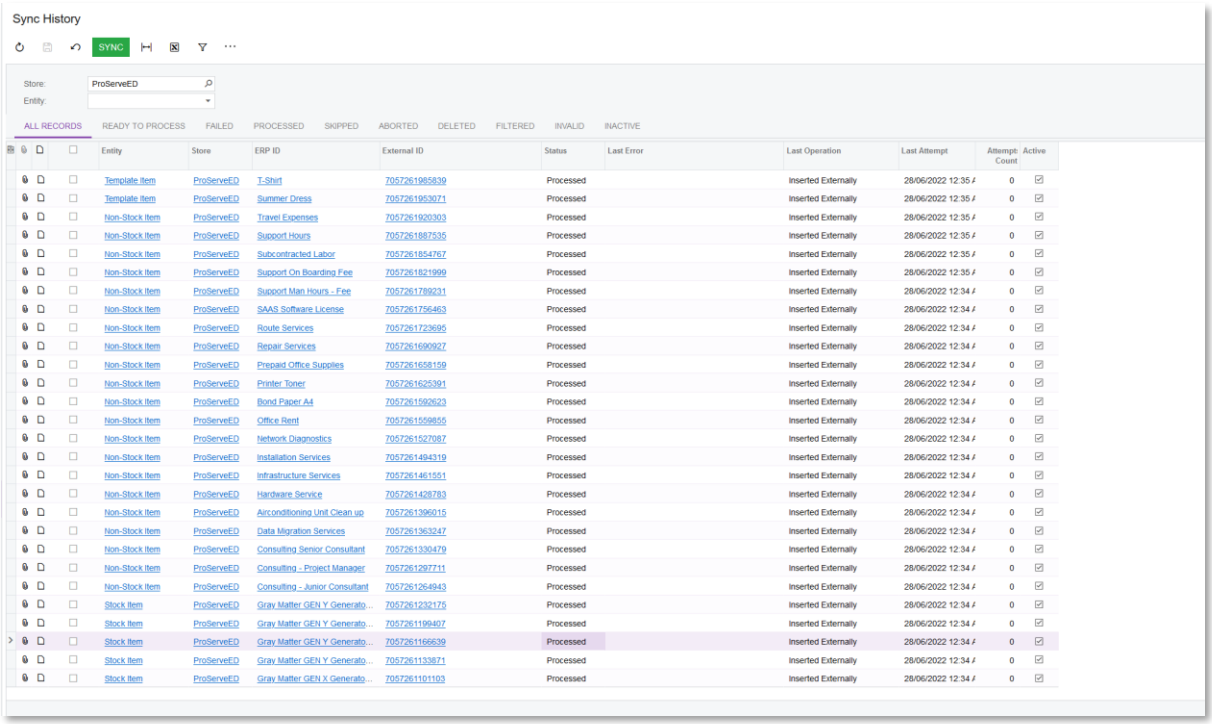
Next, go to the Prepare Data screen, select the entities that you want to sync between the two systems In the Prepare Mode, select Full, then click Prepare.



Then, go to the Process Data screen, and click Process All to start the importing/exporting of records between the two systems. Since this is the first sync and is also a full sync, the process can take a while.

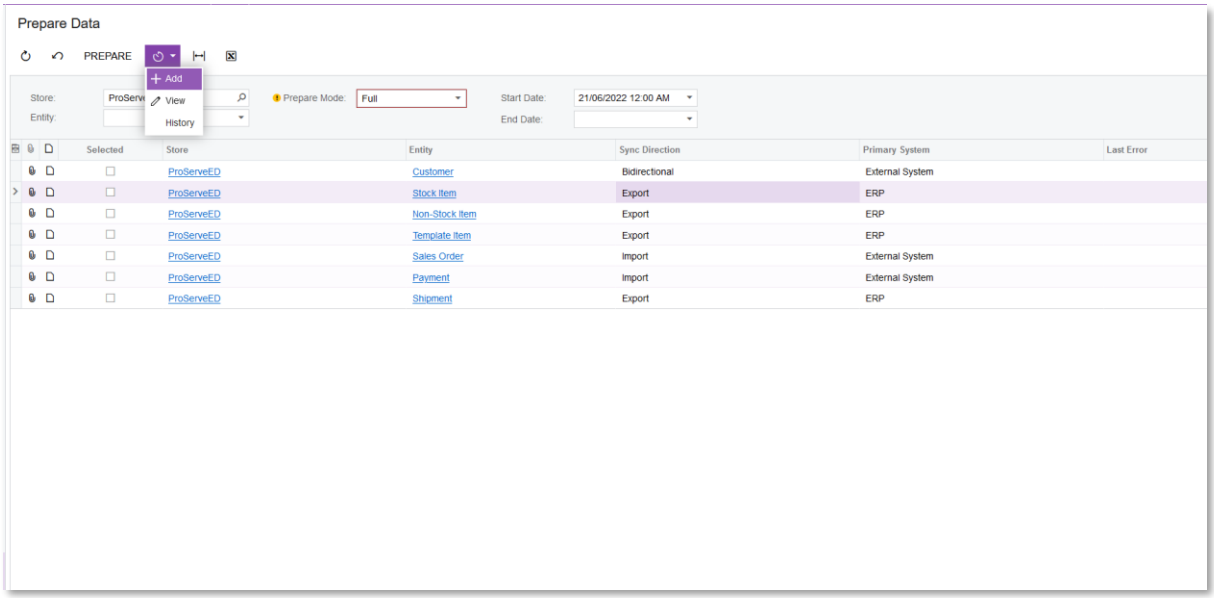


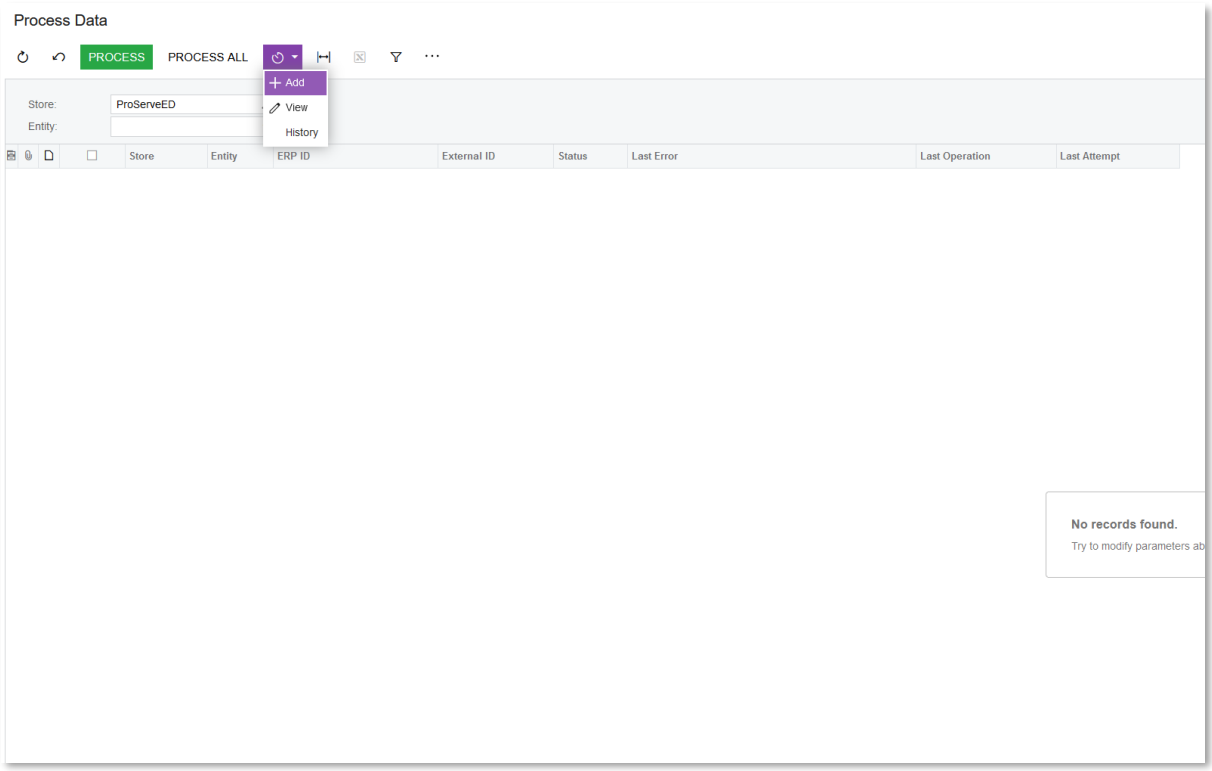
Once the sync is completed, you can go to the Sync History to review the sync result. If there's any error message related to the sync, you can also find them here. If you click on a record and press Sync, the system will do a Prepare Data/Process Data only for that record. This is a short cut for resyncing if you have updated some records in one system and would like the change to go through to the other system.



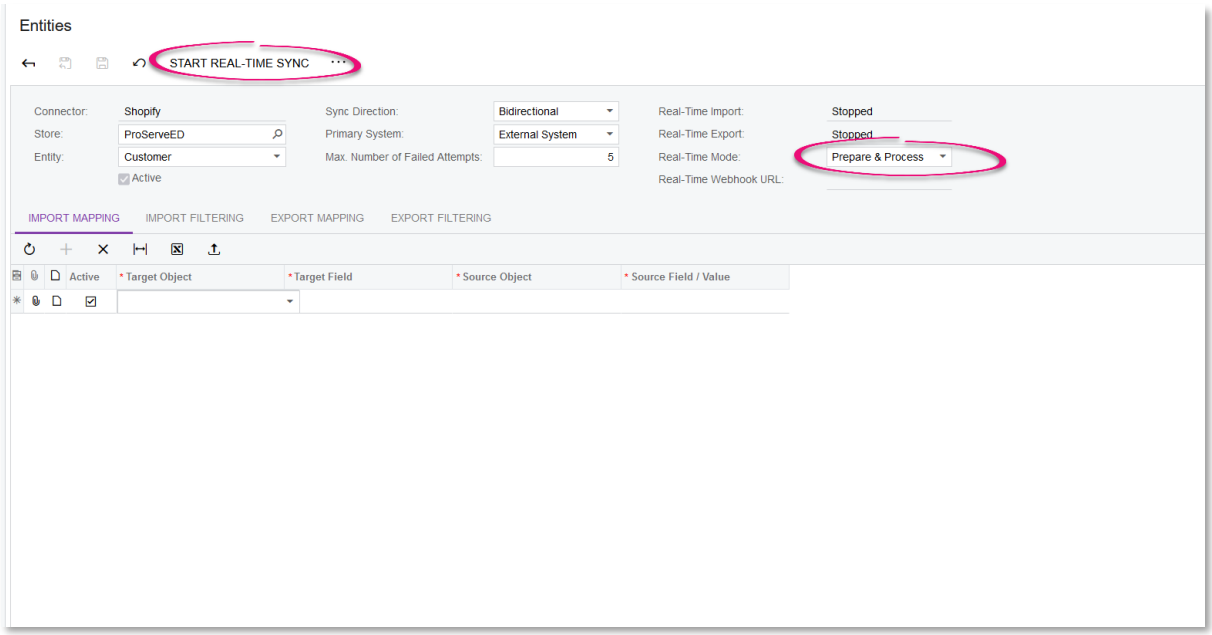
Automated synchronisation and Real-Time sync

The synchronisation between the systems can be automated. In both the Prepare Data screen and Process Data screen, you can add Automation Schedules to automate the syncing.





Alternatively, you can turn on Real-Time Sync for entities that require instant synchronisation whenever there's an update on either system. To start a Real-Time Sync, go to the Entities screen, select the entity you would like to turn on the Real-Time Sync for, select the appropriate Real-Time mode and select Start Real-Time Sync:



During Real-Time Sync, the system will monitor entities in both systems. Whenever there's a change in the record, the system will trigger the Real-Time Mode (Prepare or Prepare and Process). For Real-Time Sync to work, you should first establish a synchronisation of records between the two systems, the records should have already been imported or exported during the

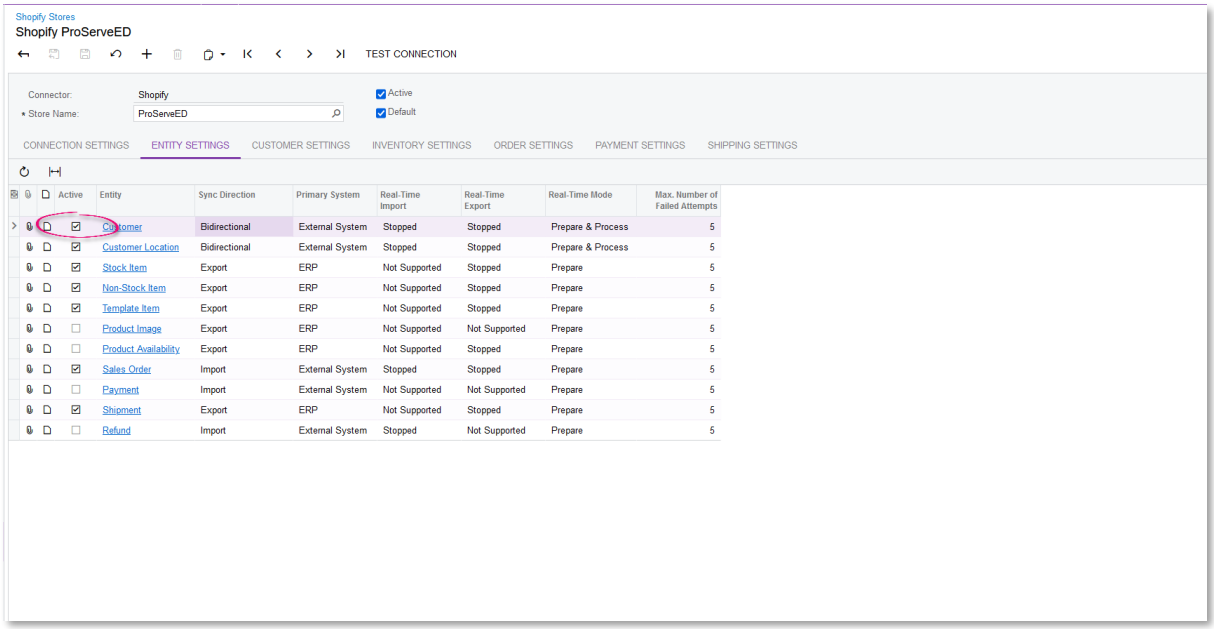
first full synchronisation. Any subsequent changes of records will be tracked by the Real-Time Sync.

Not all type of entities can have Real-Time Sync on. For example, you cannot turn on Real-Time Sync for Product Images, Payments and Refunds etc. Turning on Real-Time Sync for many entities may cause performance degradation, so you should only turn on Real-Time Sync where necessary. For example, you can turn on Real-Time Sync for Customers, Orders and Shipments, but you can leave the syncing of Stock Item, Non-Stock Item, etc. on an automation schedule if updating these records instantly whenever there's a change is not required.

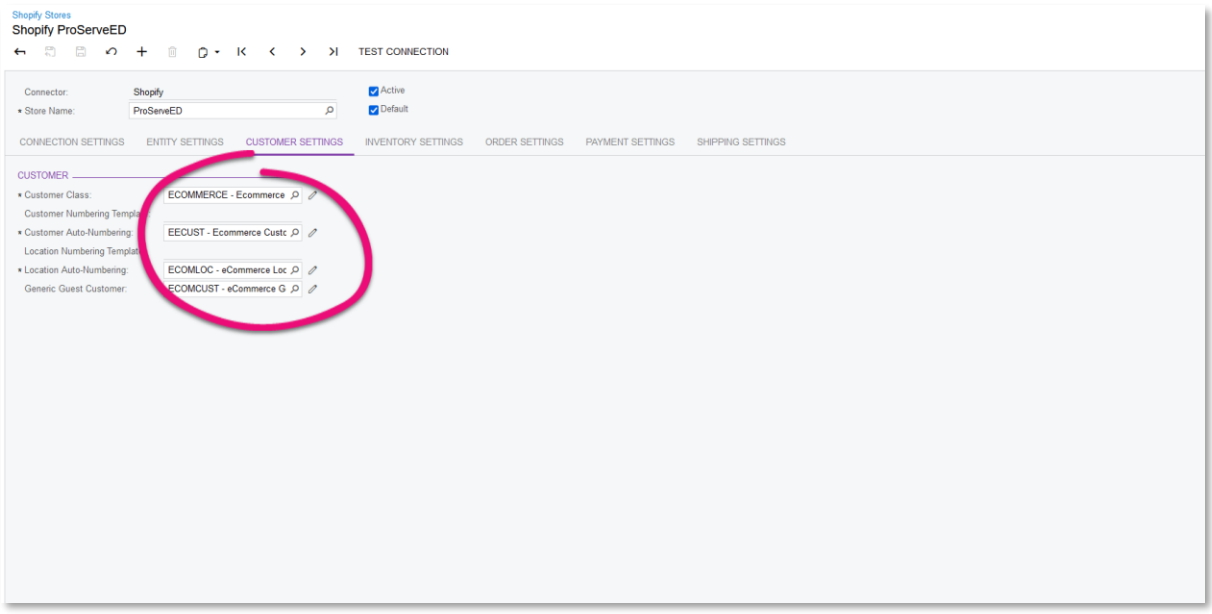
Synchronising Customers and Customer Locations

Synchronising Customers

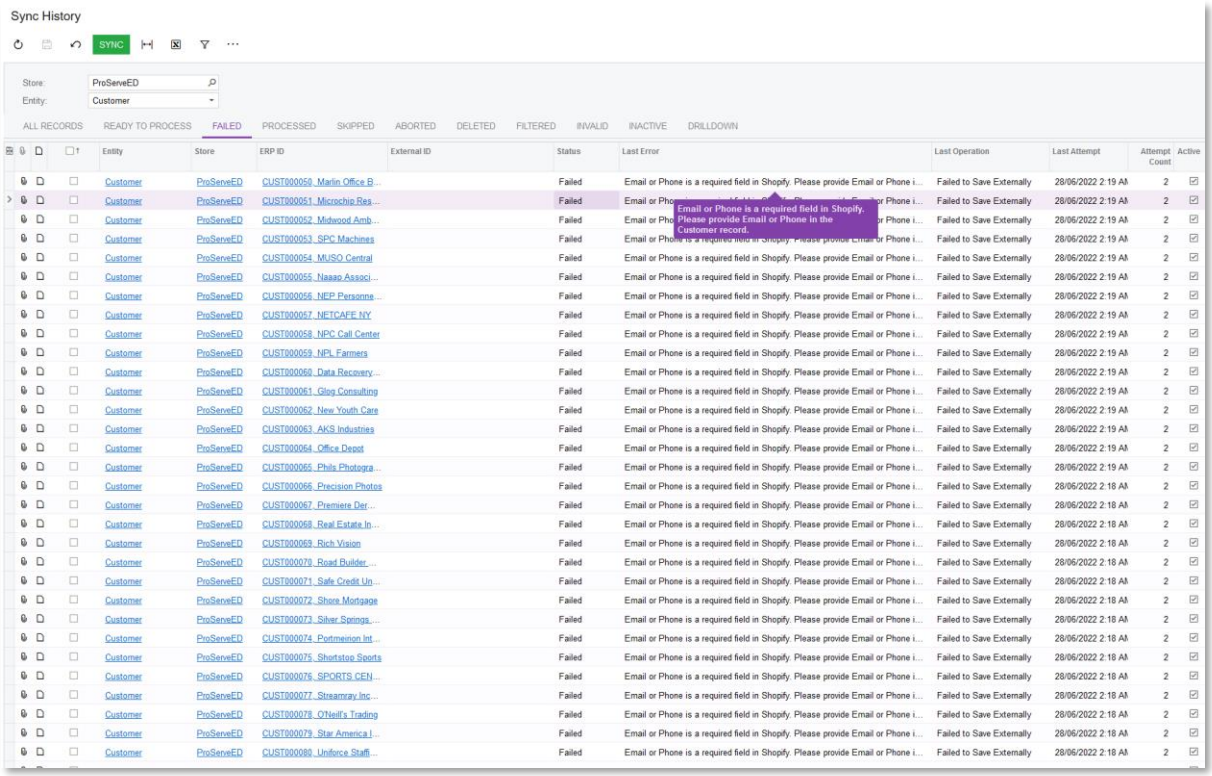
To synchronise Customers and Customer Locations between the two systems, you need to mark the Customer entity as Active in Entity Settings tab of the Shopify Stores screen (BC201010) or BigCommerce Stores screen (BC201000).



When you set the Customer Entity as active, there are a few required items that you need to set up on the Customer Settings tab. This include the Customer Class, the Customer Numbering Sequence and Locations Numbering Sequence for your eCommerce Orders.



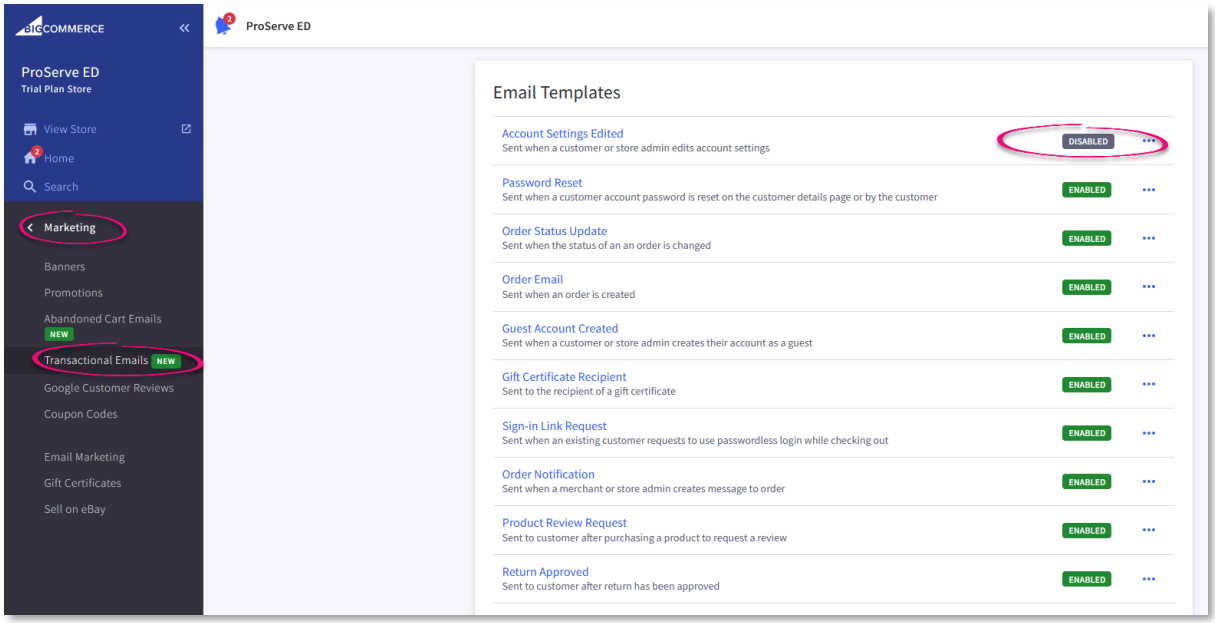
After that, you can perform the sync through the Prepare Data, Process Data screen. To review the sync result, you can go to the Sync History screen. If there is any issue during the sync, the error messages will show here, and you can resolve them accordingly. After you've resolved the errors, you can directly perform a re-sync through this screen.



During the synchronisation of the Customer entity (for both the Export direction and the Import direction), a customer record in the source system is merged with an existing customer record in the target system if both records have the same email address or the same phone number (if there is no email address).

Note: If you are planning to export customer records from MYOB Advanced to the BigCommerce store, we recommend that you turn off sending email notifications about changes to customer accounts in the BigCommerce store. If such notifications are turned on and customers are configured to be exported or to be synchronised in both directions (that is, to be exported and imported), a customer will be sent an email every time the e-commerce connector synchronises the customer data, which may look to the customer as if their store account has been hacked.

You manage sending notifications to customers on the Transactional emails page, which you open by clicking Marketing > Transactional Emails in the left menu of the control panel. To disable notifications about changes in the customer account, you click the button in the rightmost column in the row of the Account Settings Edited template, and in the menu that opens, you click Disable.

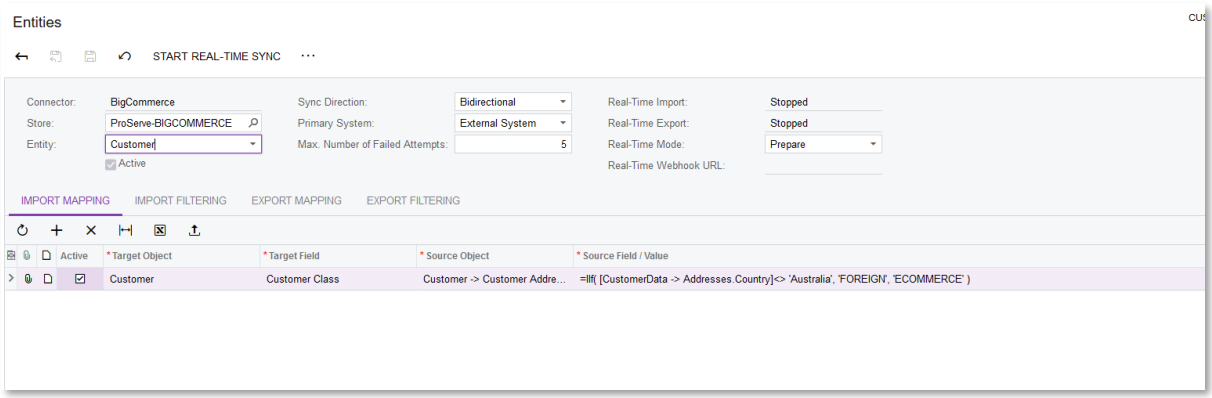


Synchronising Customer Locations

The Customer Location entity is closely related to the Customer entity. Therefore, whenever you perform a sync or resync for the Customer entity, the Customer Location entities are also synced.

Utilising formulas in Field Mappings for Customers

Suppose you have different customer classes, for example, local customers and overseas customers. These customers classes can have different settings (especially Tax Zone, so that the Sales Orders can contain the correct Tax), you can utilize the formulas on the Import Mappings of the Customer Entities like the below example, to pick the correct Customer Class for the customer record when they are imported from the store to MYOB Advanced:



Sync direction, Mapping, Filtering and Primary system

From this section onward, this guide will not mention the Sync Direction, Mapping, Filtering or Primary System of each Entity. You can use the default values, or you can make change if required, depending on the requirements of the customer.

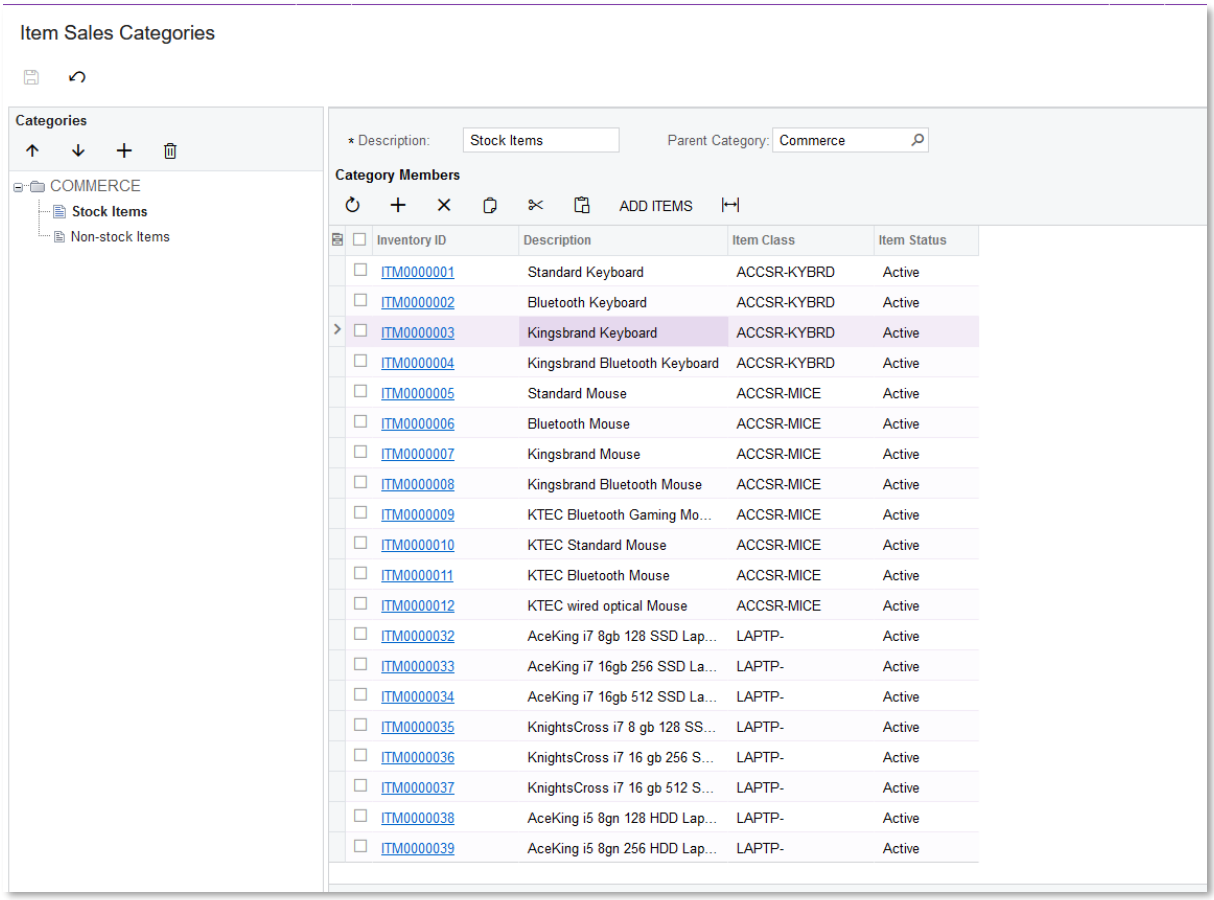
Synchronising Inventory and Product Images

Synchronising Sales Category – Prerequisite for synchronizing Products for BigCommerce

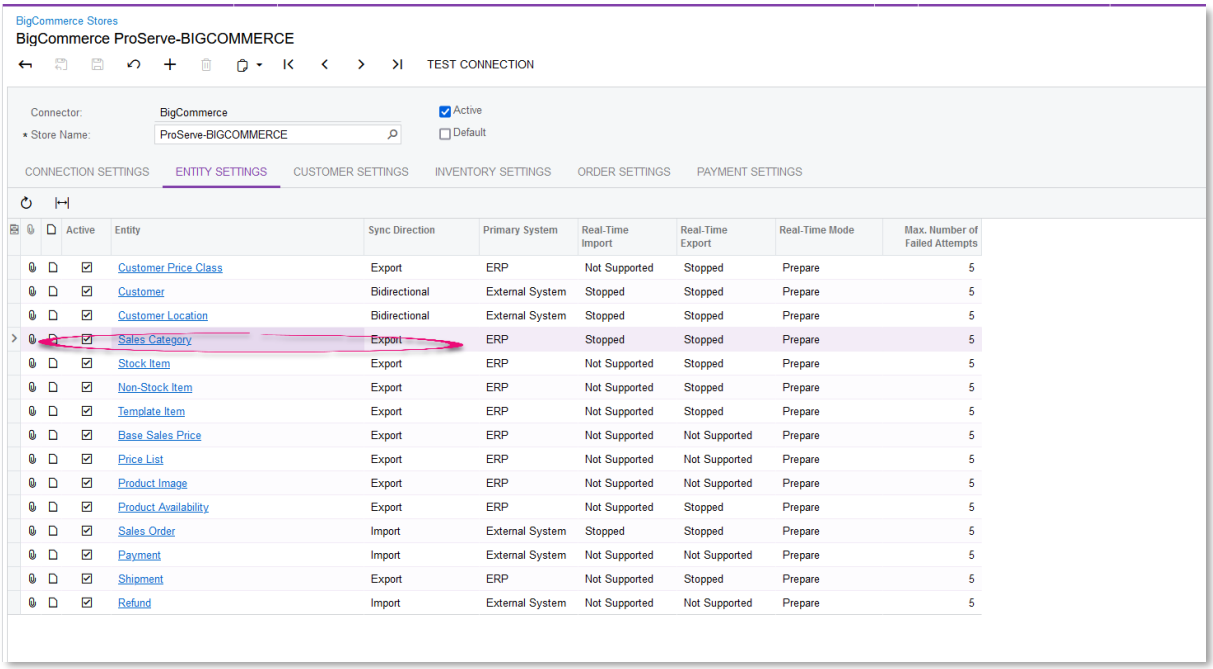
For BigCommerce, before you're able to sync Stock Items, Non-Stock Items, you need to create Sales Categories and add items into Sales Categories. These Sales Categories are pre-requisite and must be created before syncing. If an Inventory ID does not have any Sales Category, they will not sync.

To create Sales Categories, go to the screen Item Sales Categories (IN204060) in MYOB Advanced. Add Sales Categories and items into those categories.

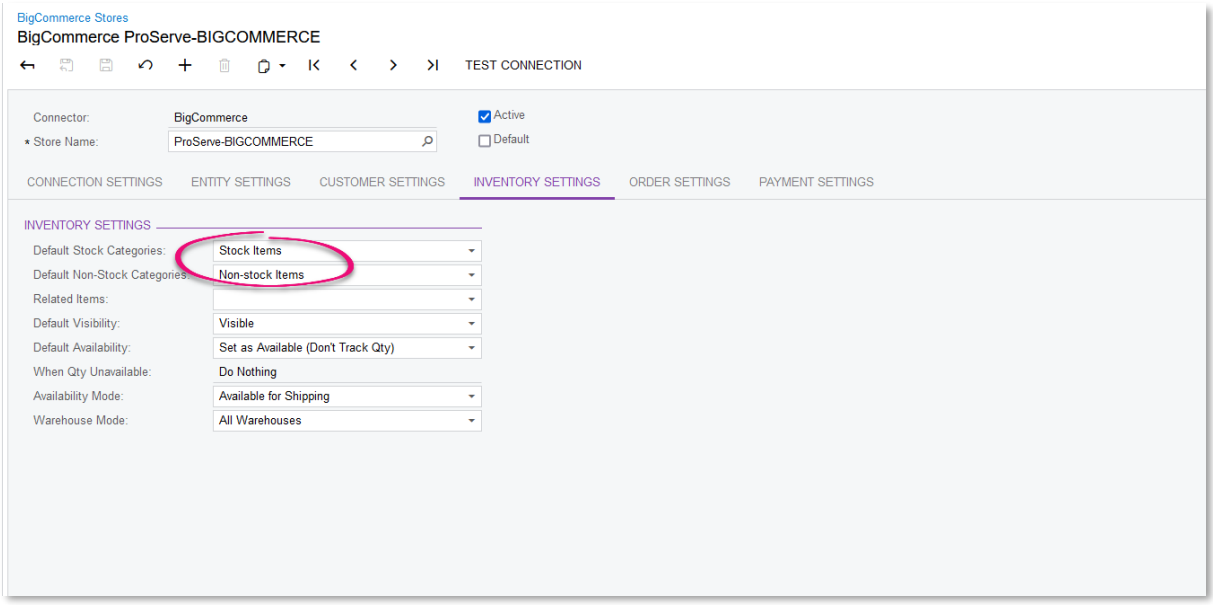
You can use the Add Items button to quickly add all items, or all items from an item class to the screen.



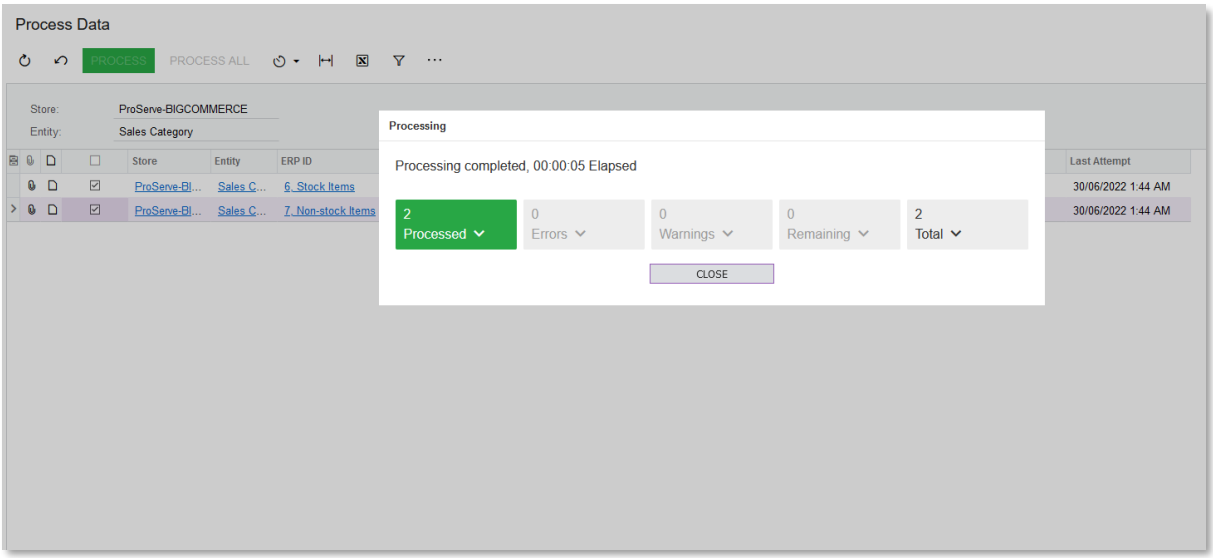
On the BigCommerce Store (BC201000) screen, enable the syncing of the Sales Category entity in the Entities tab



In the Inventory Settings tab, specify the Default Stock Categories and Default Non-Stock Categories:



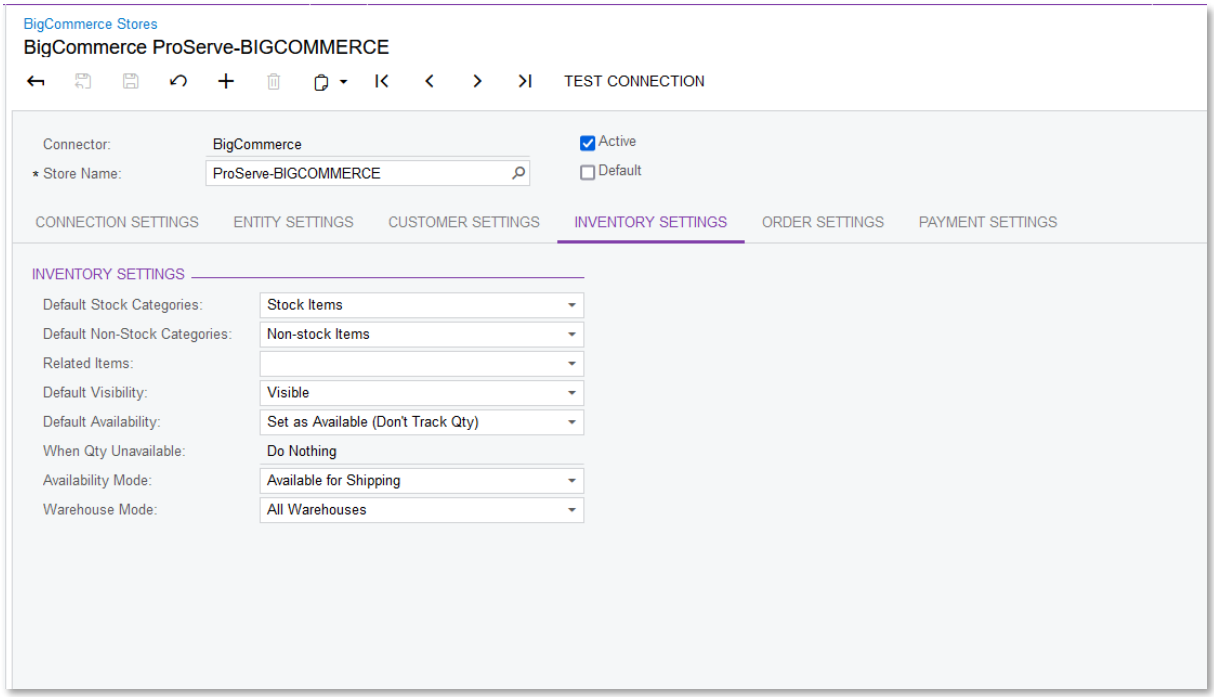
Prepare Data and Process Data for the Sales Category entity:

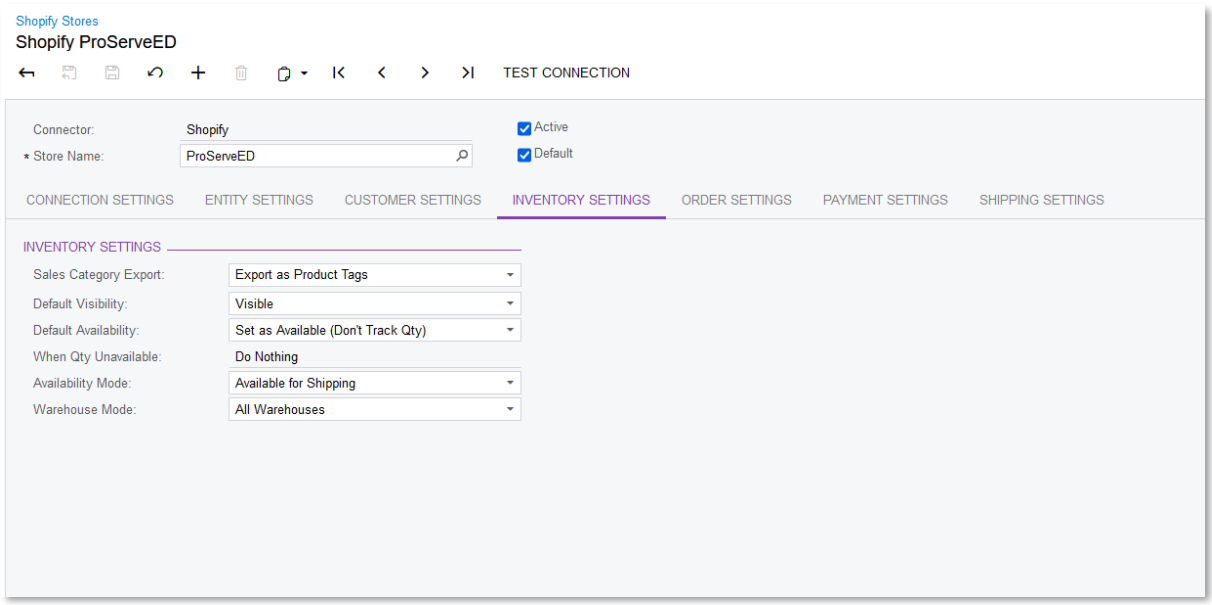


In the BigCommerce Store, the Sales Categories become Product Categories:

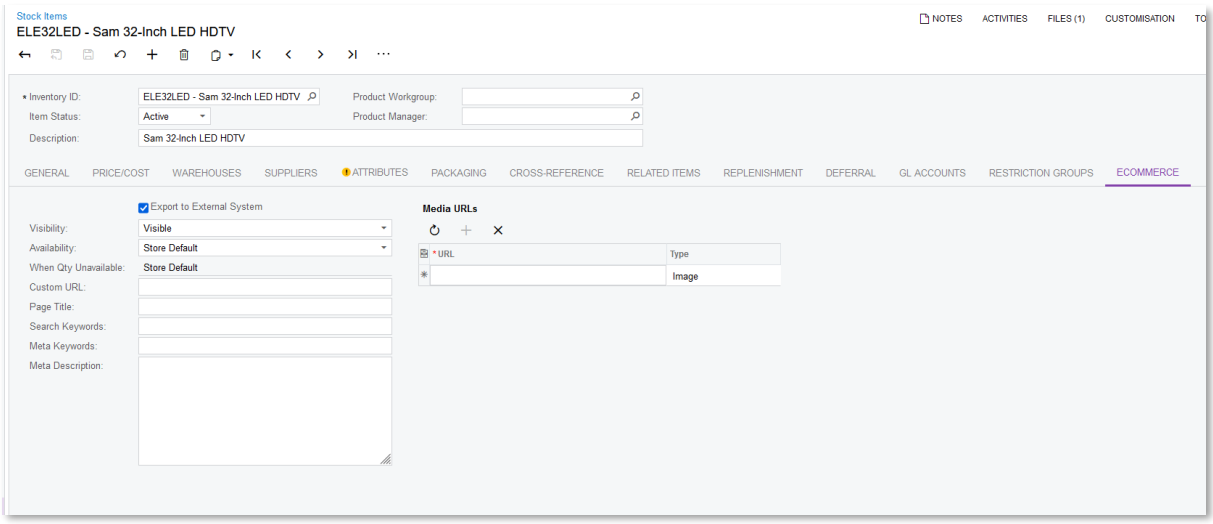
Synchronising Inventory

Before you synchronise Inventory, you need to check the default options of Inventory on the Shopify Stores (BC201010) or BigCommerce Stores (BC201000) screen. You can make change if required.

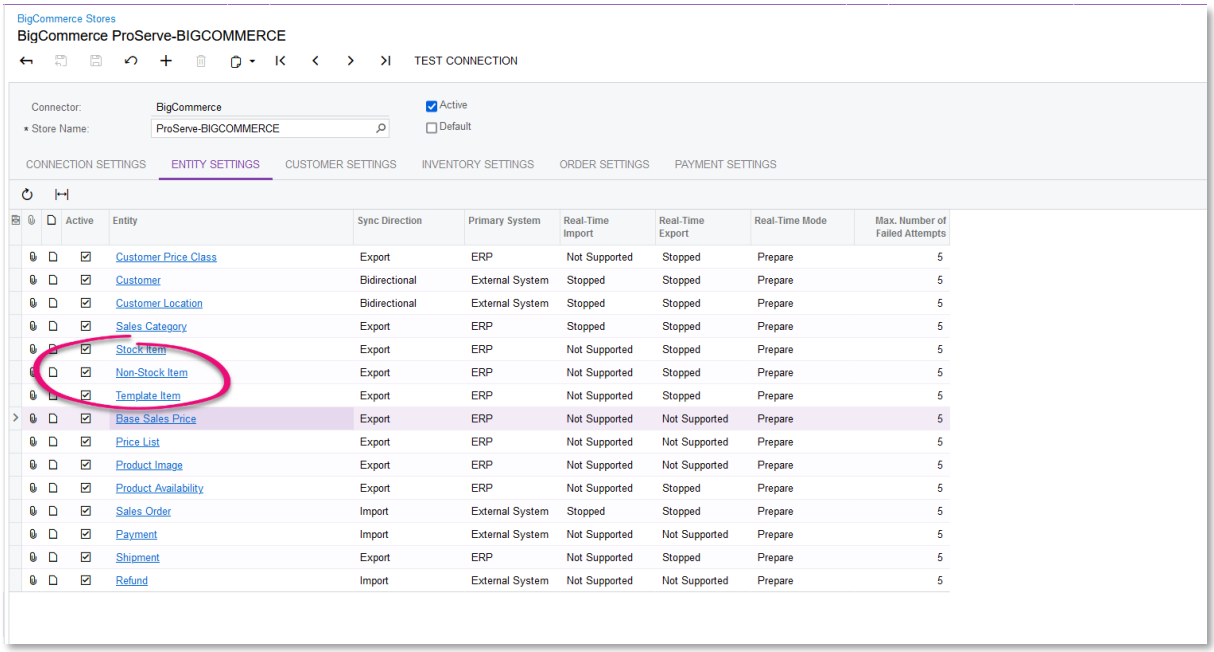


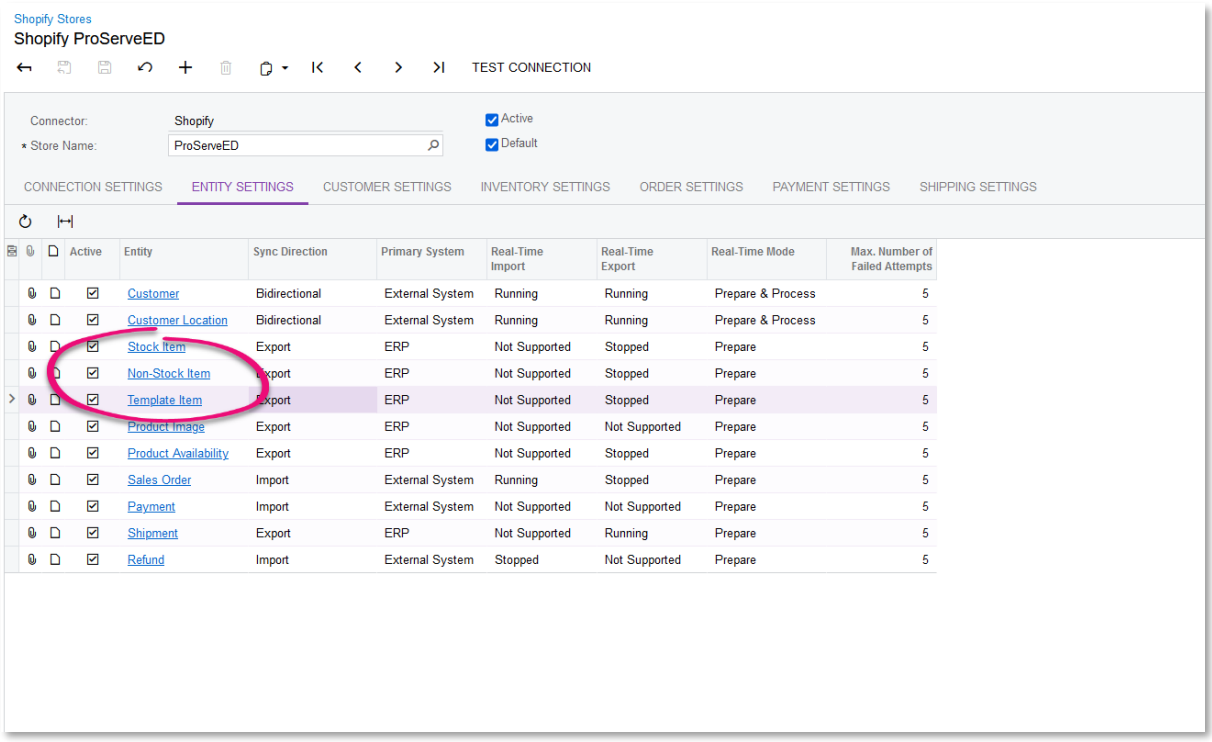


If required, you can override the default values above for a specific Stock Item or Non-Stock Item from the eCommerce tab of the Stock Items screen or Non-stock Items screen:



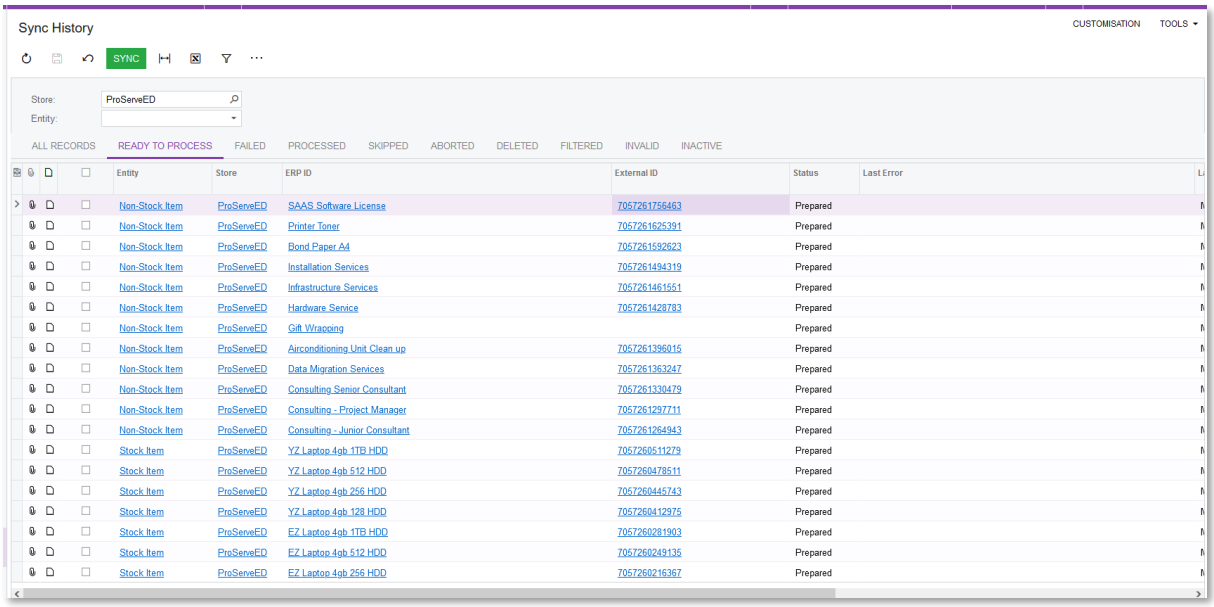
On the Entity Settings tab of each store, set the Stock Items, Non-Stock Items, Template Item to active.





Prepare Data and Process Data to sync Inventory between the systems.

You can use the Sync History screen to review the synchronisation result, and resolve any error if required:

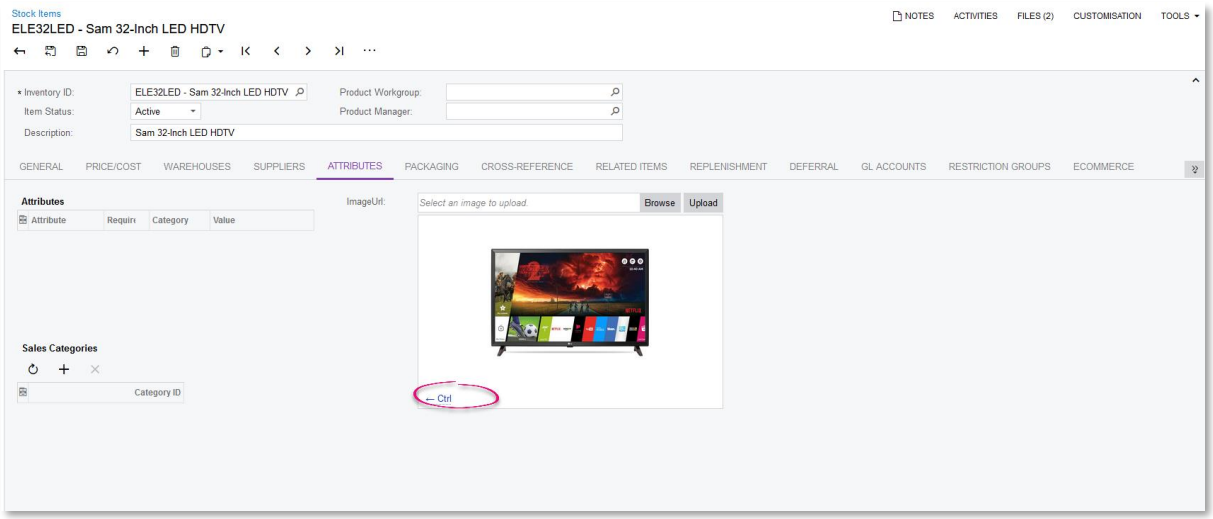


During the export of stock items from MYOB Advanced, a stock item is merged with an existing product in BigCommerce if at least one of the following conditions is met:

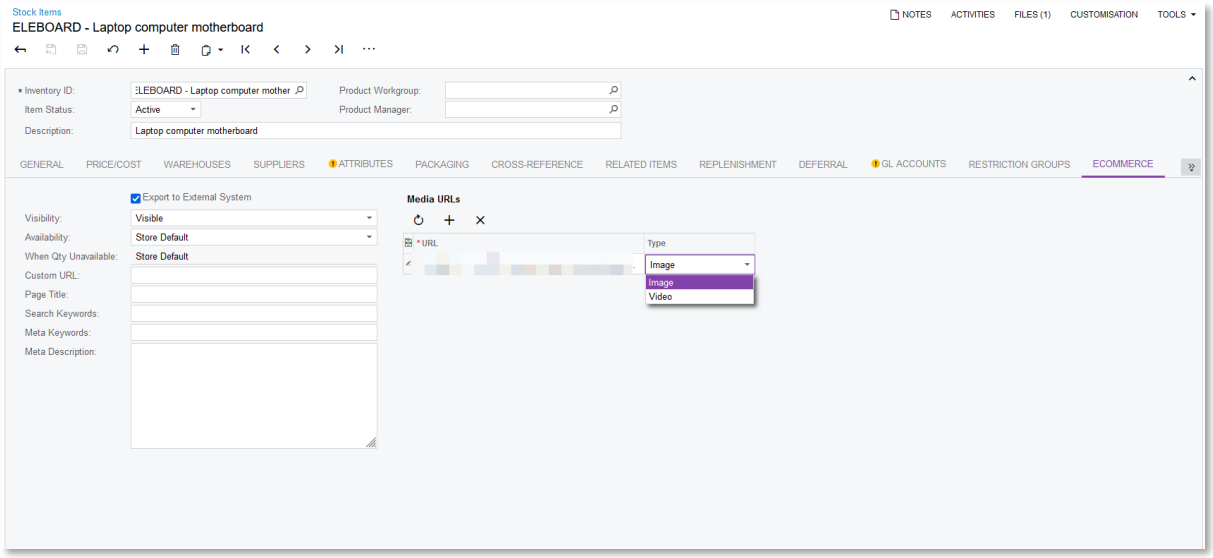
- The stock item's description in MYOB Advanced—that is, the value in the Description box on the Stock Items (IN202500) form—matches the Product Name of the product in BigCommerce.
- The stock item's ID in MYOB Advanced—that is, the value in the Inventory ID box on the Stock Items form—matches the SKU of the product in BigCommerce.

Synchronising Product Images

There are two ways to store and sync Product Images for inventory. The first method, is storing the image on the Stock Items screen or the Non-Stock Items screen as a file, which you can see through the Attributes screen. If you have multiple images for one item, you can attach multiple image files to the stock items screen, and use the **Ctrl** icon to change to which one will be the default image of this item. The default image in MYOB Advanced also becomes the default image or thumbnail image when exported to Shopify or BigCommerce.



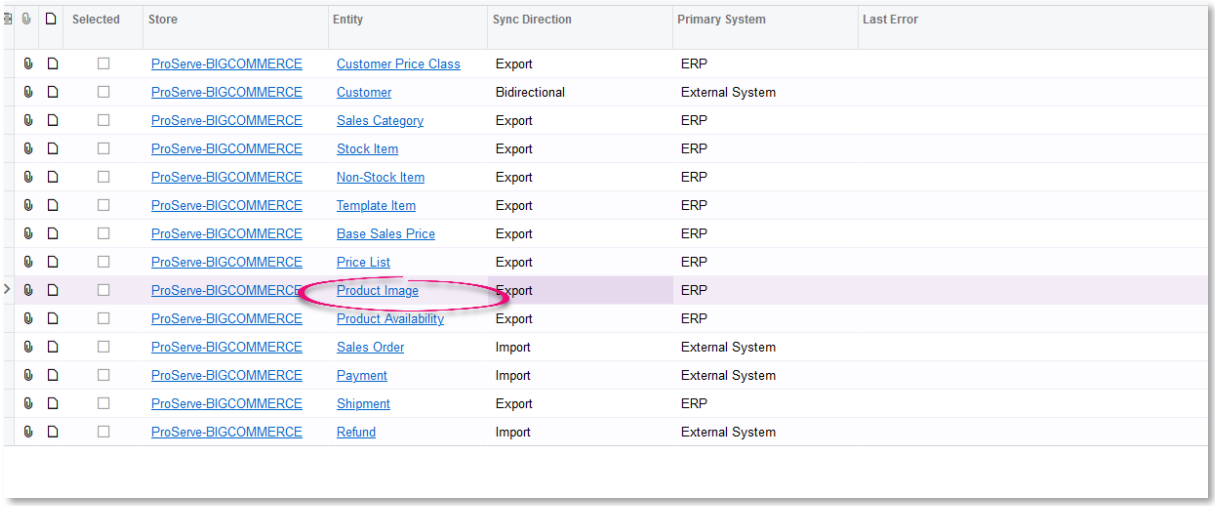
The second method is to store images on a server, like an AmazonS3 bucket, and copy and paste the URL of the image to the Media URL field on the eCommerce tab of the Stock Items or Non-stock Items screen. Make sure the URL is public and accessible. You can specify whether a URL is an Image or a Youtube Video:



Videos are not supported in the Shopify connector. Shopify's REST API does not accept YouTube and Vimeo URLs.

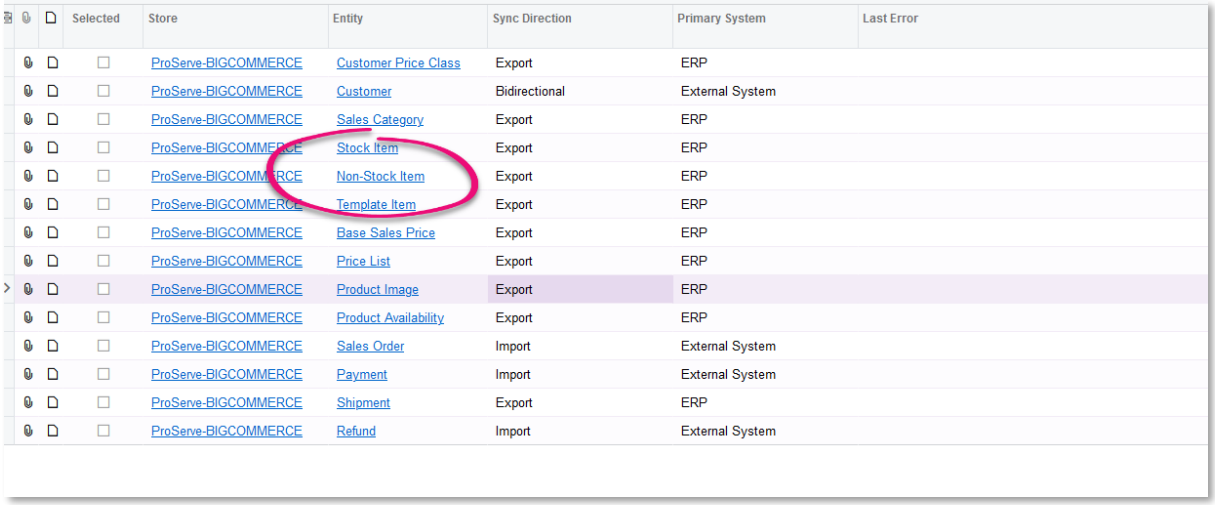
There is a difference when choosing which Entity to sync Product Images:

- If you use the first method to store images, which is to drag and drop image files directly on the Stock Items screen, then these images can only be synched through the “Product Image” entity:



Selected	Store	Entity	Sync Direction	Primary System	Last Error
☐	ProServe-BIGCOMMERCE	Customer Price Class	Export	ERP	
☐	ProServe-BIGCOMMERCE	Customer	Bidirectional	External System	
☐	ProServe-BIGCOMMERCE	Sales Category	Export	ERP	
☐	ProServe-BIGCOMMERCE	Stock Item	Export	ERP	
☐	ProServe-BIGCOMMERCE	Non-Stock Item	Export	ERP	
☐	ProServe-BIGCOMMERCE	Template Item	Export	ERP	
☐	ProServe-BIGCOMMERCE	Base Sales Price	Export	ERP	
☐	ProServe-BIGCOMMERCE	Price List	Export	ERP	
☐	ProServe-BIGCOMMERCE	Product Image	Export	ERP	
☐	ProServe-BIGCOMMERCE	Product Availability	Export	ERP	
☐	ProServe-BIGCOMMERCE	Sales Order	Import	External System	
☐	ProServe-BIGCOMMERCE	Payment	Import	External System	
☐	ProServe-BIGCOMMERCE	Shipment	Export	ERP	
☐	ProServe-BIGCOMMERCE	Refund	Import	External System	

- If you use the second method, which is to store images as Media URLs, then these images can only be synched through the Stock Item/Non-Stock Item/Template Item entity:

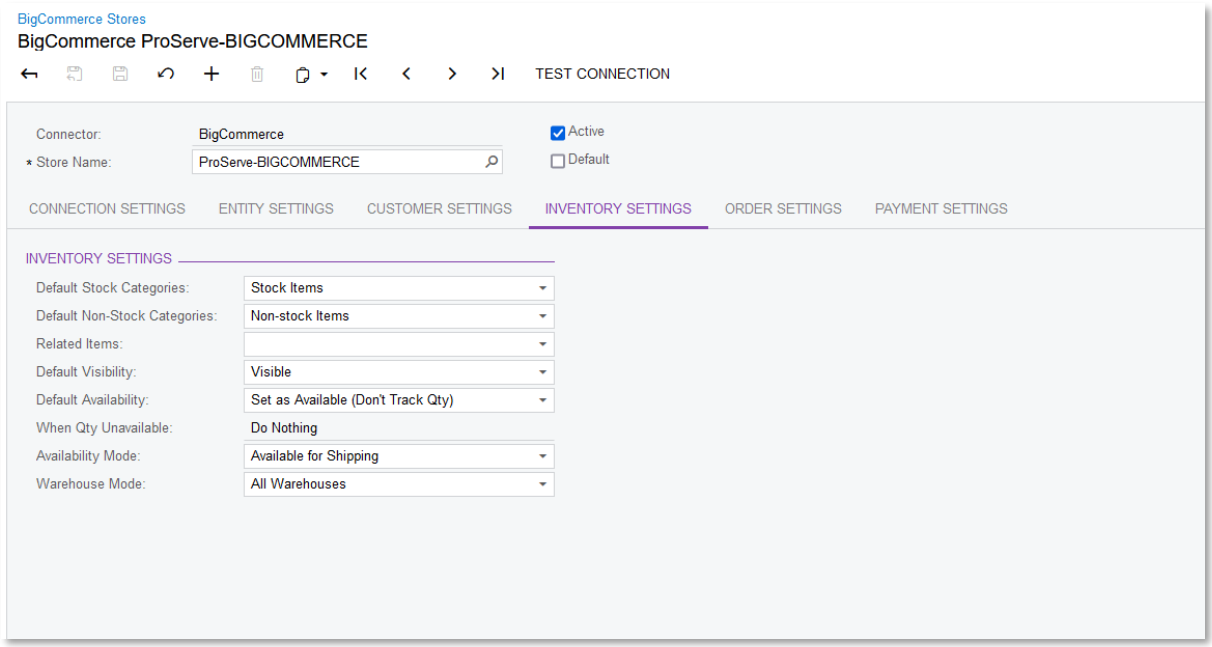


Selected	Store	Entity	Sync Direction	Primary System	Last Error
☐	ProServe-BIGCOMMERCE	Customer Price Class	Export	ERP	
☐	ProServe-BIGCOMMERCE	Customer	Bidirectional	External System	
☐	ProServe-BIGCOMMERCE	Sales Category	Export	ERP	
☐	ProServe-BIGCOMMERCE	Stock Item	Export	ERP	
☐	ProServe-BIGCOMMERCE	Non-Stock Item	Export	ERP	
☐	ProServe-BIGCOMMERCE	Template Item	Export	ERP	
☐	ProServe-BIGCOMMERCE	Base Sales Price	Export	ERP	
☐	ProServe-BIGCOMMERCE	Price List	Export	ERP	
☐	ProServe-BIGCOMMERCE	Product Image	Export	ERP	
☐	ProServe-BIGCOMMERCE	Product Availability	Export	ERP	
☐	ProServe-BIGCOMMERCE	Sales Order	Import	External System	
☐	ProServe-BIGCOMMERCE	Payment	Import	External System	
☐	ProServe-BIGCOMMERCE	Shipment	Export	ERP	
☐	ProServe-BIGCOMMERCE	Refund	Import	External System	

Prepare and Process Data for the relevant entities to finish the synchornisation of Product Images.

Inventory Settings

On the BigCommerce Stores screen (BC201000) and in the Shopify Stores screen (BC201010), you can specify default Inventory Settings on the Inventory Settings tab.

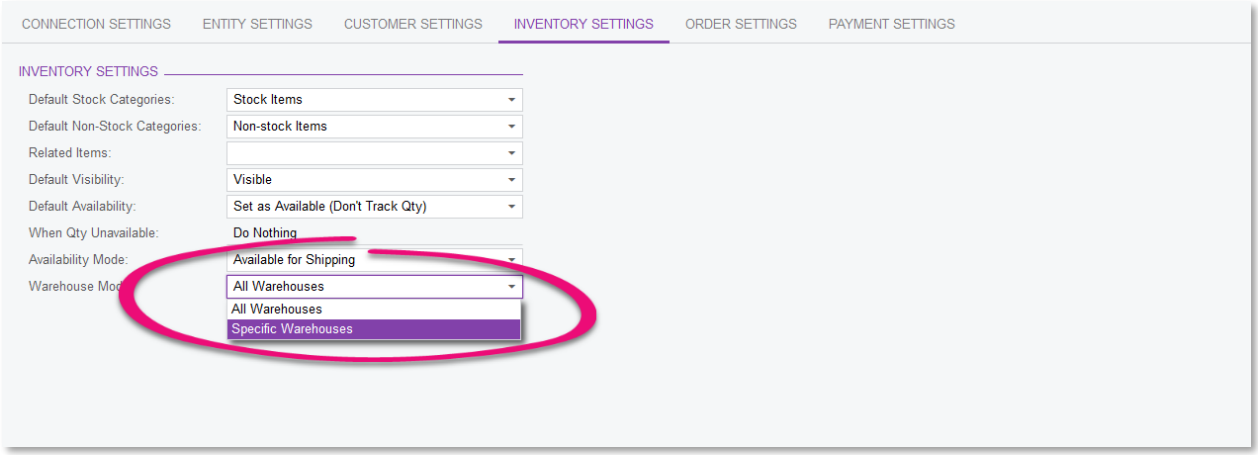


These settings define the behavior of the stock items or non-stock items after they are exported to the storefront. For example, when Default Visibility is set to Visible, when an item is exported, it'll be visible on the storefront. When set to Featured, the stock item becomes a featured product on the front page of the store. You can also set the Availability Mode, which will export the qty available of the product based on the Availability Calculation Rule; there are three options: Available, Available for Shipping and On Hand, etc.e

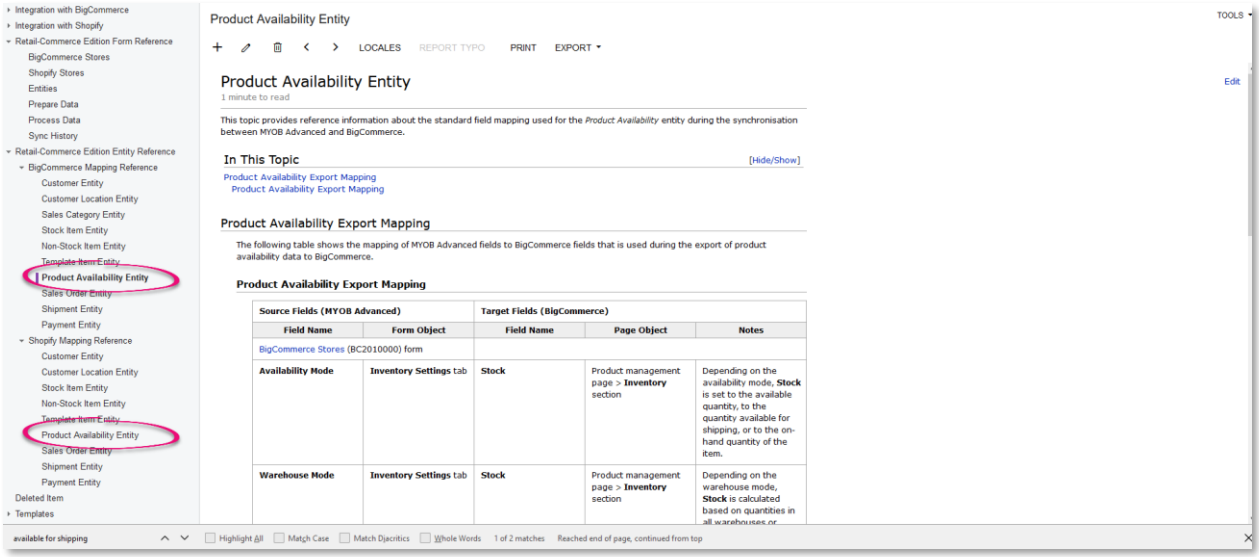
Refer to the in-product User Guide for full explanation of each setting.

Synchronising product availability

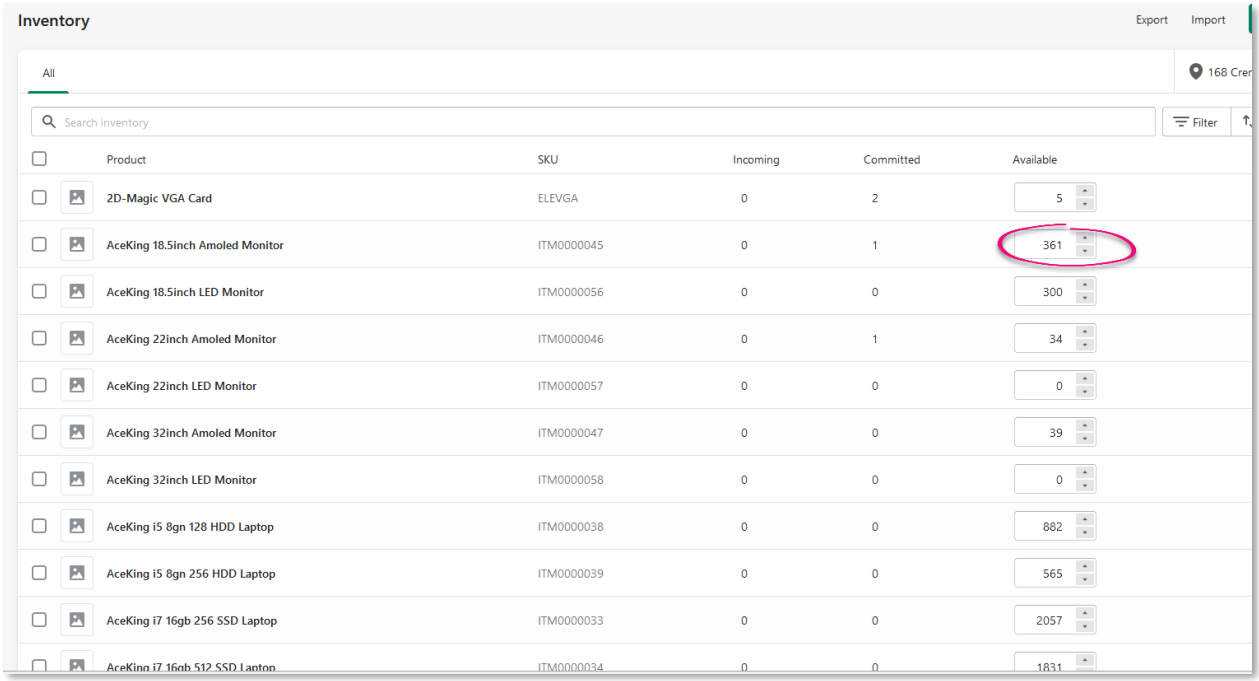
Product Availability can only be exported from MYOB Advanced to the stores, and you can configure Real-Time Sync for product availability. Product Availability obeys the Availability Calculation Rule specified for the Item Class, and you can specify whether MYOB Advanced should export the Available qty, Available for Shipping qty, or On Hand qty through the BigCommerce Store screen or through the Shopify Store screen, in the Inventory Settings tab. You can also specify whether the Available qty should be taken for all Warehouse, or only for a specific Warehouse and Warehouse Location.

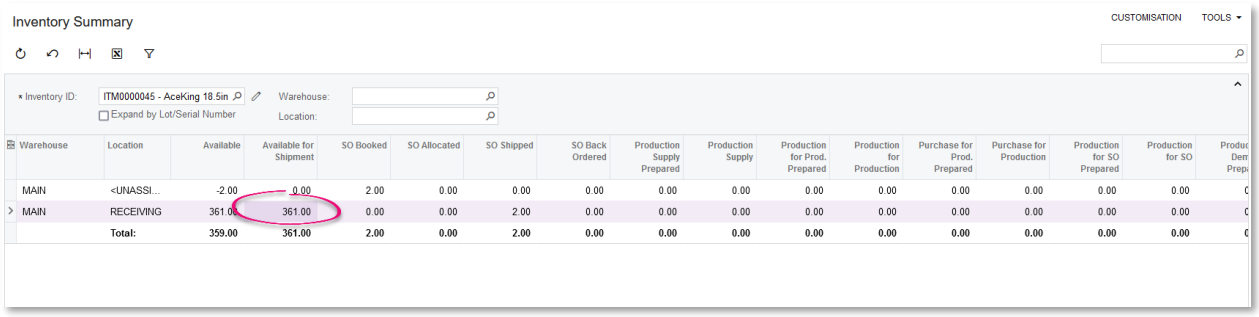


Depending the value of the Default Availability field on the BigCommerce Store and Shopify Store, or the value of the Availability field in the eCommerce store on the Stock Items tab, the behavior of the products in the stores will be different. For further information on this, refer to the in-product user guide, section Product Availability Entity.



The Product Availability once synced to the store will be shown as the available quantity for the Stock Item:



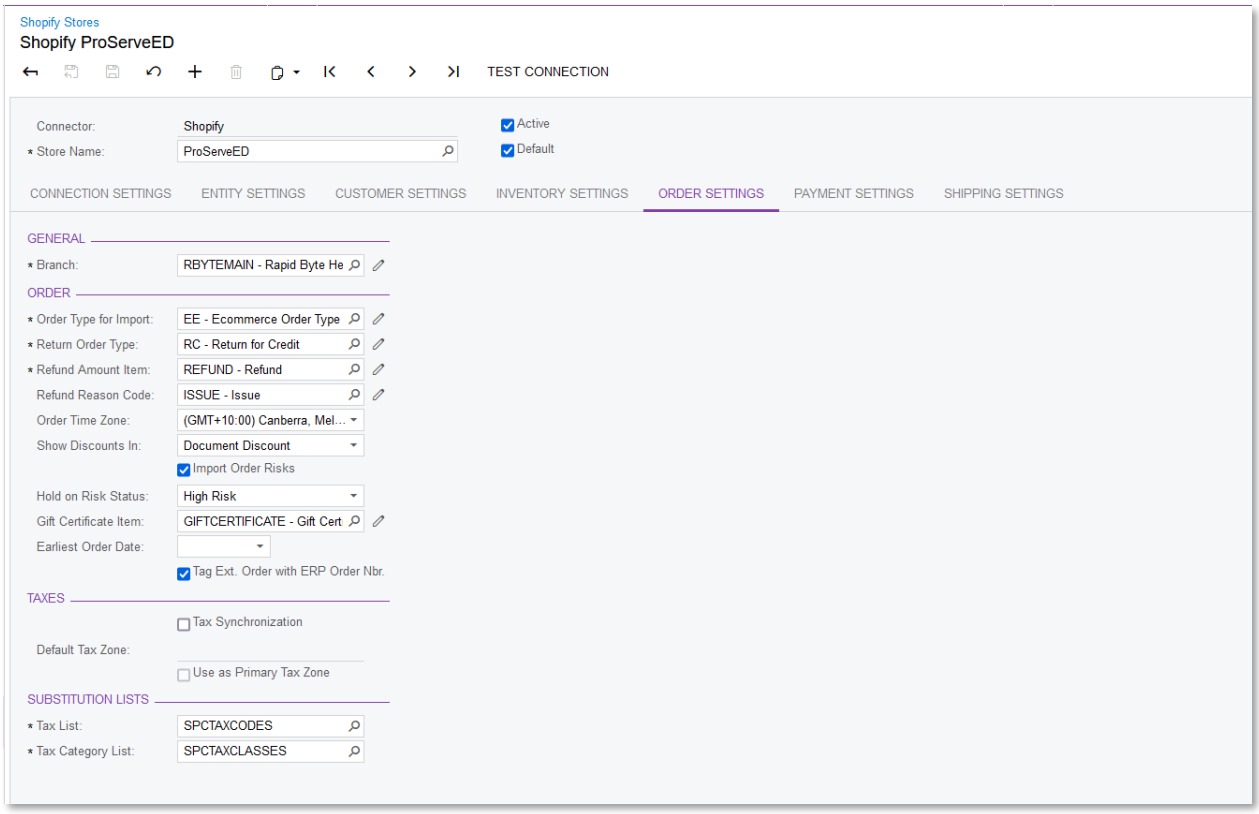


Synchronising Sales Orders

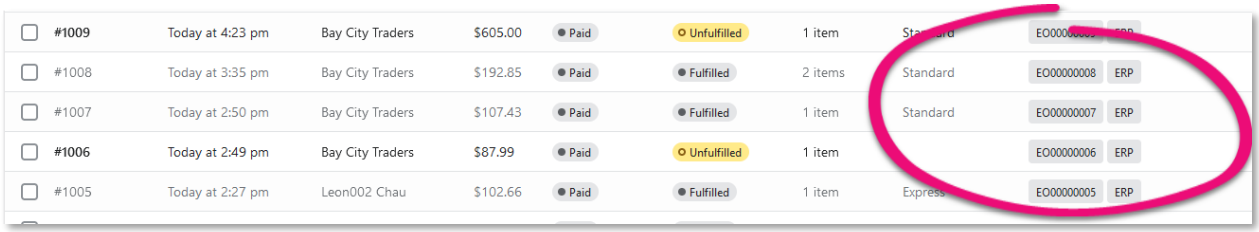
Order Settings

Before synchronizing orders, we need to do the necessary settings for the Orders Entities. On the Shopify Stores screen or the BigCommerce Stores screen, set up the default values to support the synchronisation. Below are some important settings for each type of store:

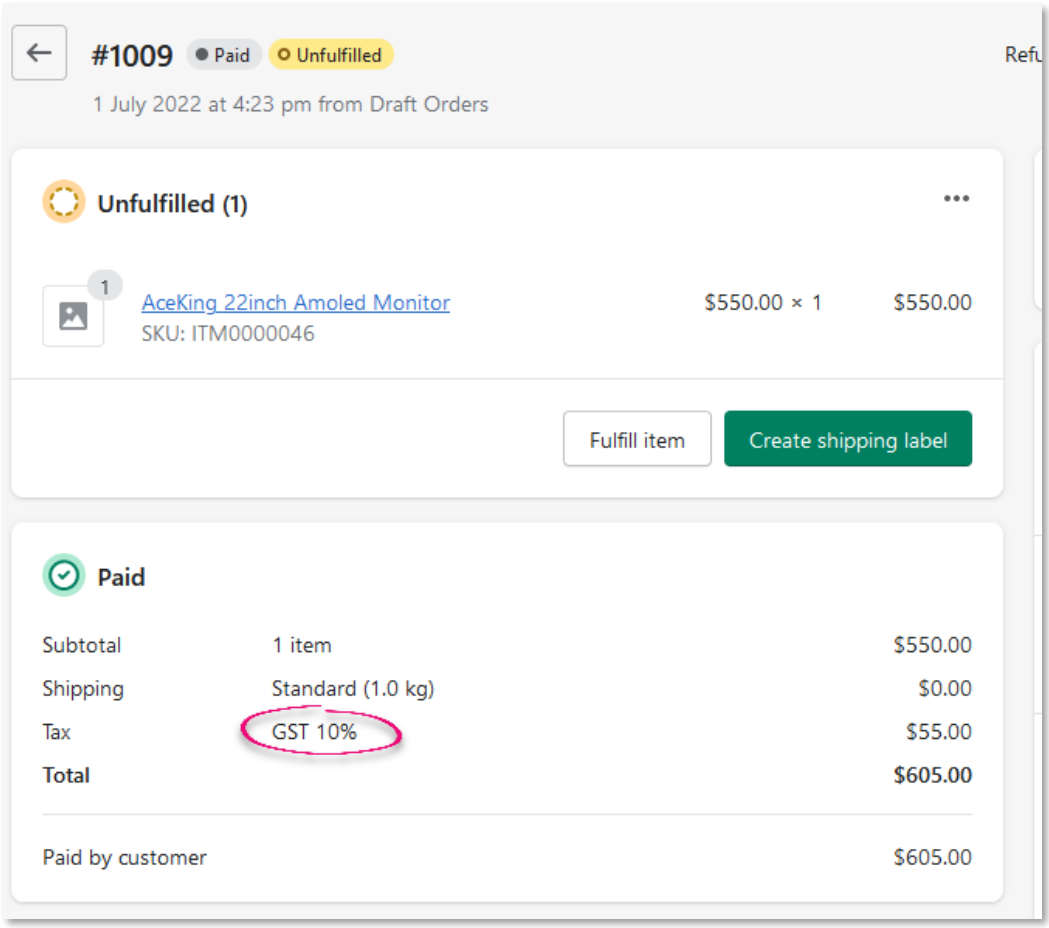
For Shopify Stores:



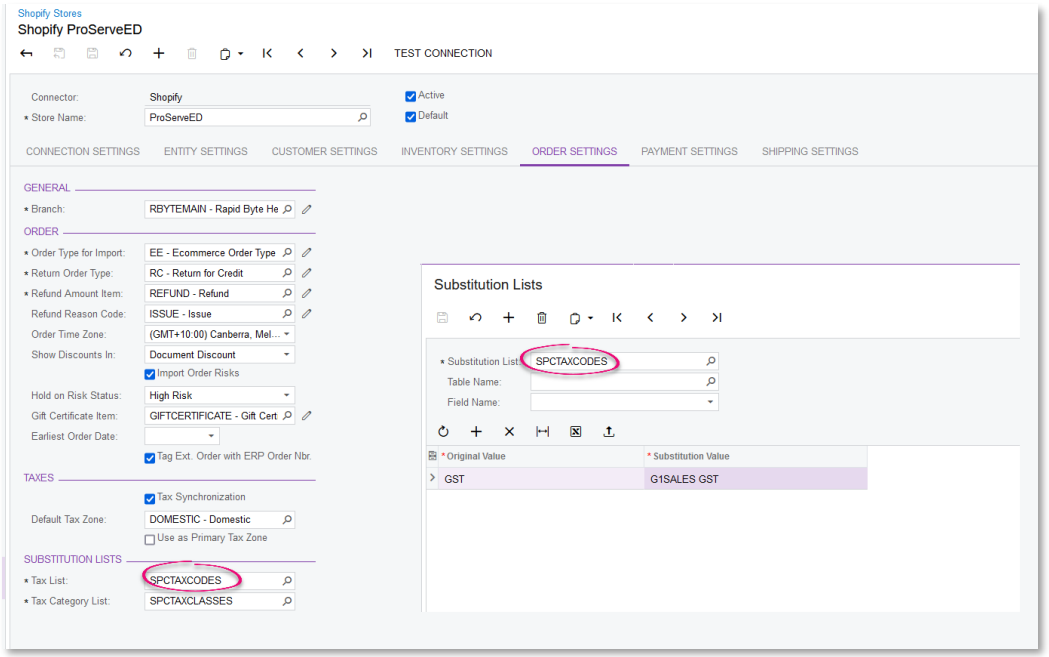
- Branch: The default branch for the Sales Orders when they're imported from Shopify
- Order Type for import: The Order Type to be used when the orders are imported from Shopify
- Return Order Type: The Sales Order Type that will be used for customer returns
- Refund Amount Item: A non-stock item that represents the amount to refund to the customer.
- Gift Certificate Item: a non-stock item to be used as a Gift Certificate/Gift Card. Customer can purchase this item on Shopify. They use the Gift Card as a payment method.
- Tag Ext. Order with ERP Order Nbr – when ticked: the Sales Order number in MYOB Advanced will be tagged to the Shopify Order, like this:



- Tax Synchronisation: When ticked: The Tax ID from Shopify will be synced to myob Advanced. The default Sales Tax ID in Shopify for Australia is "GST":

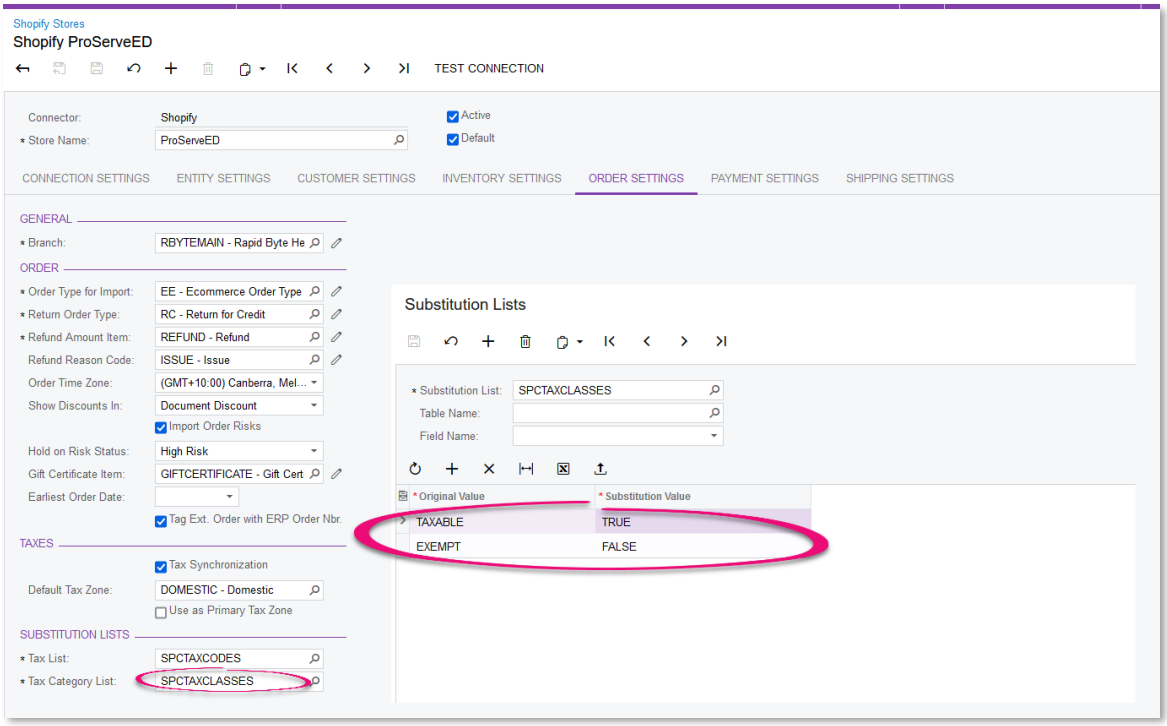


- You should ensure that the GST Tax ID also exists in MYOB Advanced. Alternatively, in the Substitution List section, in the Tax List substitution list, you should create a mapping for the GST Tax ID to the Tax ID in MYOB Advanced, like this:

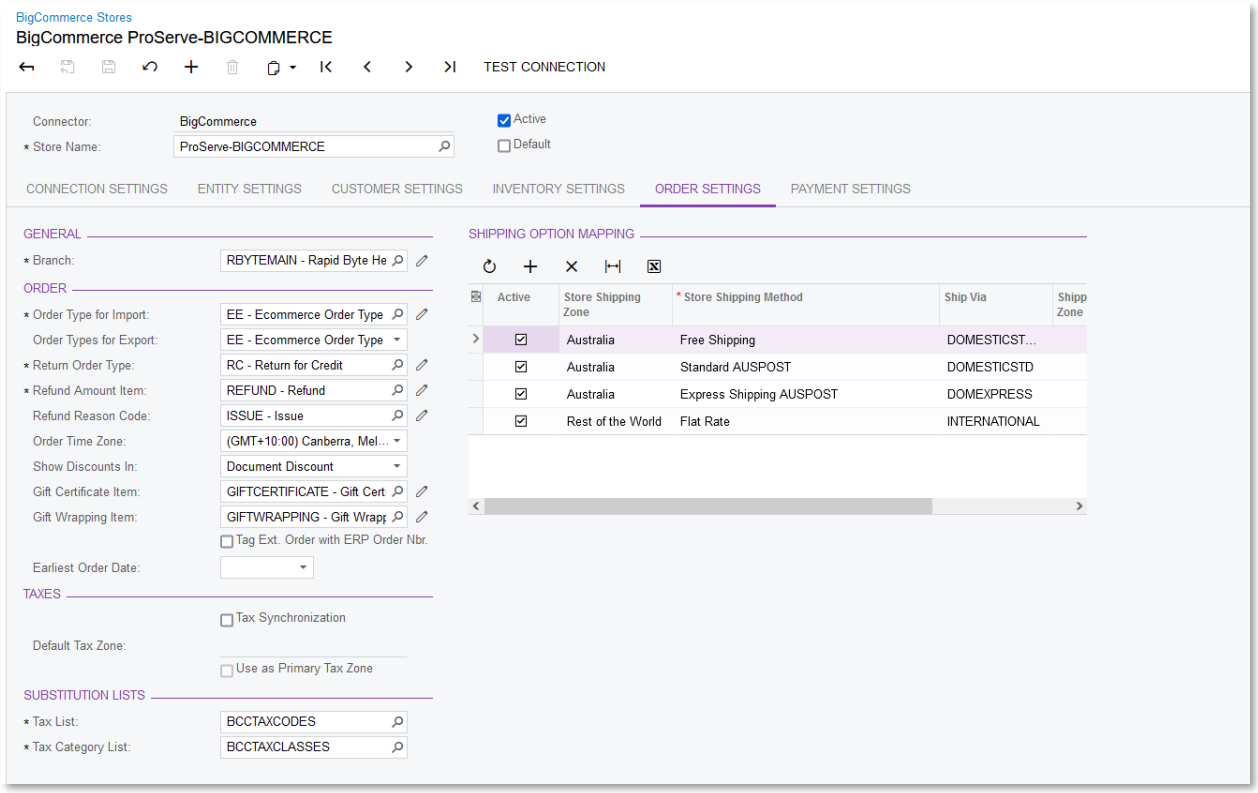


- With the above setup, when the Sales Orders are imported, they'll have the G1SALES GST tax code.

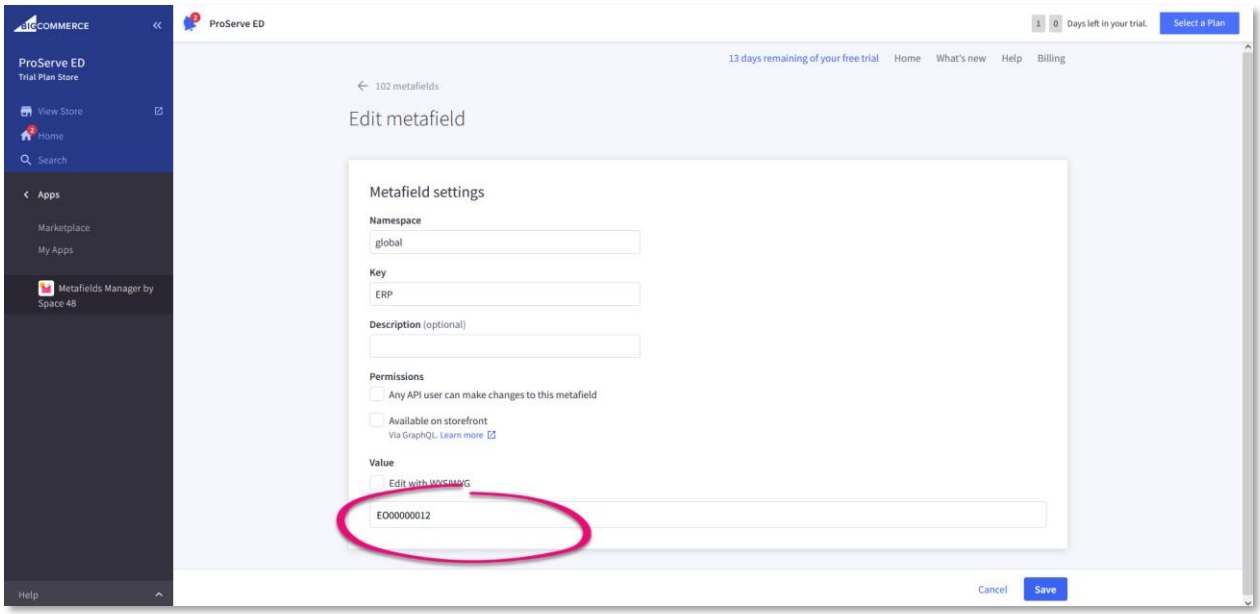
- The SPECTAXCLASSES Substitution List is used in exporting of Inventory from MYOB Advanced to Shopify. If the Stock Item's Tax Category is "TAXABLE", when exported to Shopify, the field "Charge Tax on this Product" is set to "True". If the Stock Item's Tax Category is EXMEMPT, when exported to Shopify, the field "Charge Tax on This Product" is set to "False". You change this mapping if you have different Tax Categories set up in MYOB Advanced.



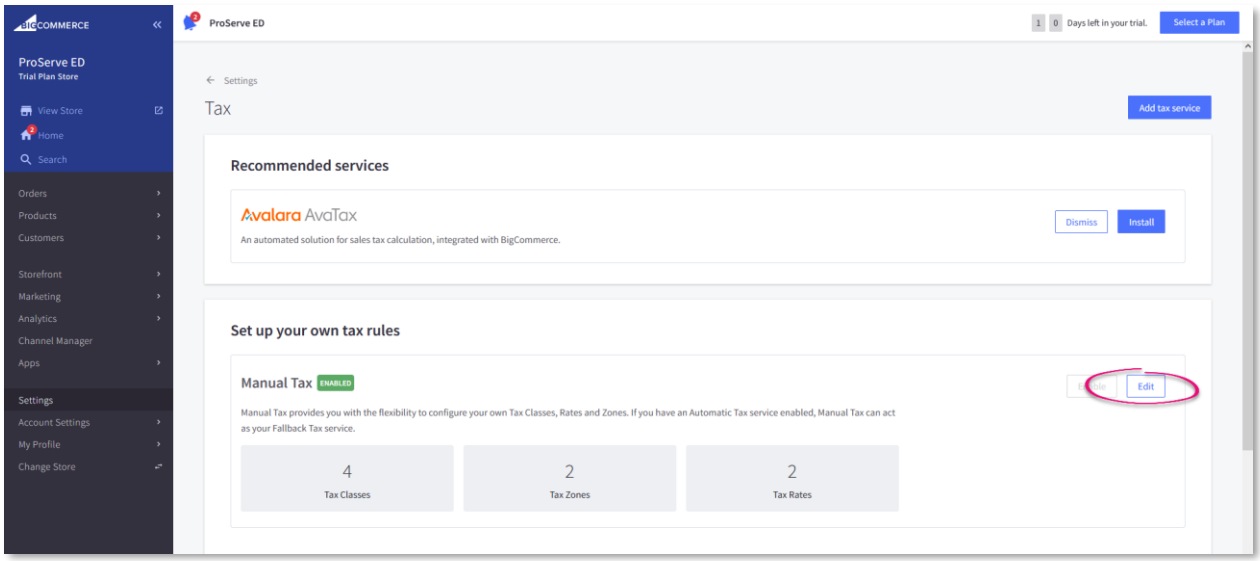
For BigCommerce Stores:



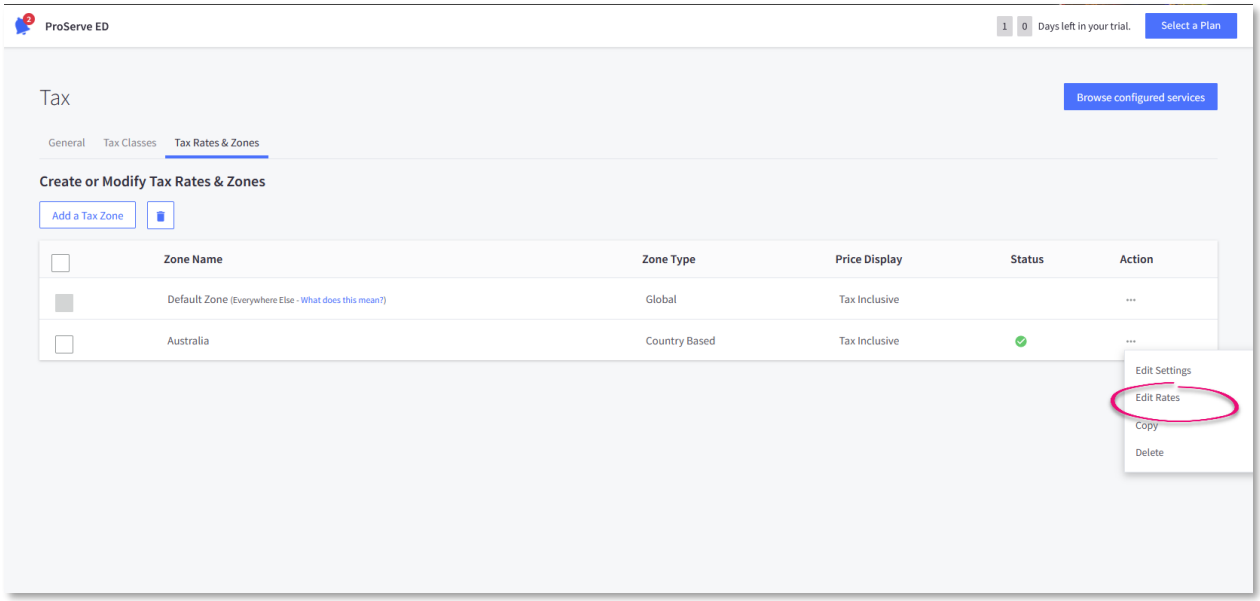
- Branch: The default branch for the Sales Orders when they're imported from BigCommerce
- Order Type for import: The Order Type to be used when the orders are imported from BigCommerce
- Order Types for Export: The Order Types that you would like to export to BigCommerce – for example, customers might call to raise a Sales Orders. You then enter this order into MYOB Advanced, then fulfill it. You can then export this order back to BigCommerce so that when customer logs in to BigCommerce they can see their order.
- Return Order Type: The Sales Order Type that will be used for customer returns.
- Refund Amount Item: A non-stock item that represents the amount to refund to the customer.
- Gift Certificate Item: a non-stock item to be used as a Gift Certificate/Gift Card. Customer can purchase this item on BigCommerce. They use the Gift Card as a payment method.
- Gift Wrapping Item: a non-stock item to be used to represent the Gift Wrapping service fee.
- Tag Ext. Order with ERP Order Nbr – when ticked: the Sales Order number in MYOB Advanced will be tagged to the BigCommerce Order as a metafield, which you can use for development work. Or if you have an App on BigCommerce like Metafield Manager, you'll be able to see the value of the metafield, like this:



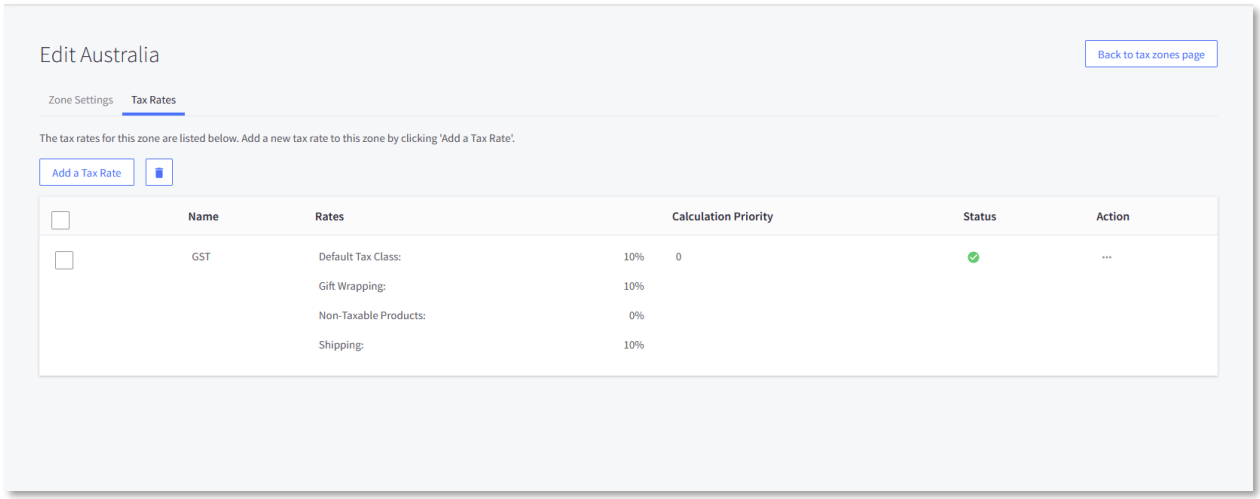
- Tax Synchronisation:
  - When ticked: The Tax ID from BigCommerce will be synced to myob Advanced. In BigCommerce, for Australia, the default Sales Tax ID is “GST”. To view the Tax ID, you can go to Settings > Tax > select Edit in the Manual Tax section:



In the Tax Rate and Zones tab, select Australia and select Edit Rates

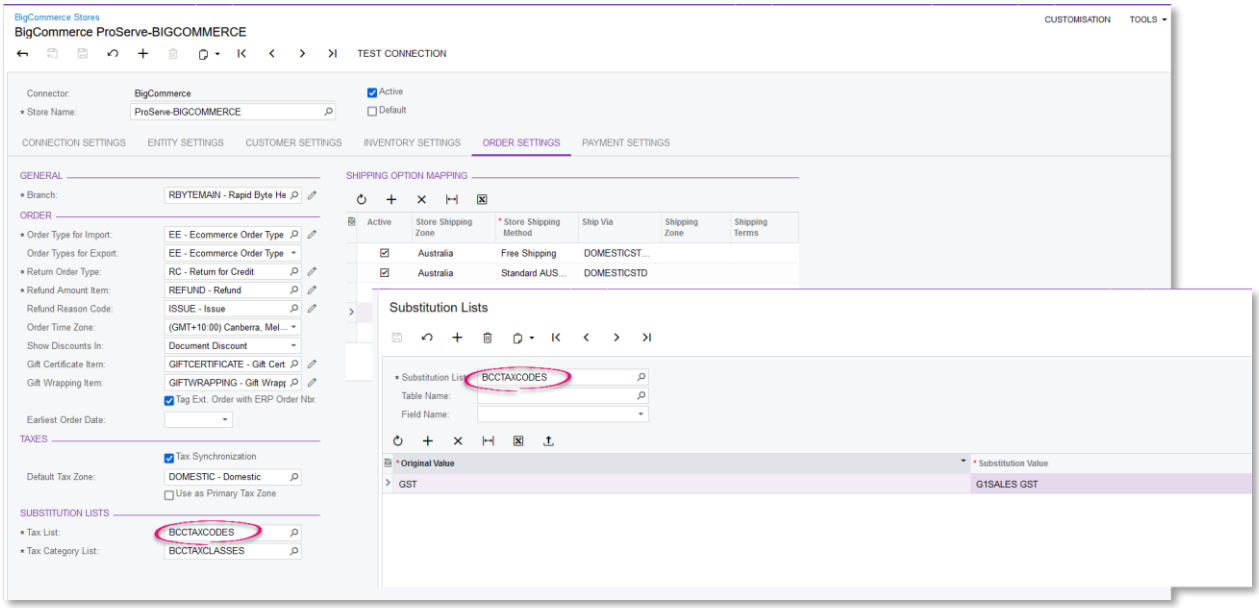


You'll be able to see the tax rate and the Tax ID and the rate applicable to each Tax Class:



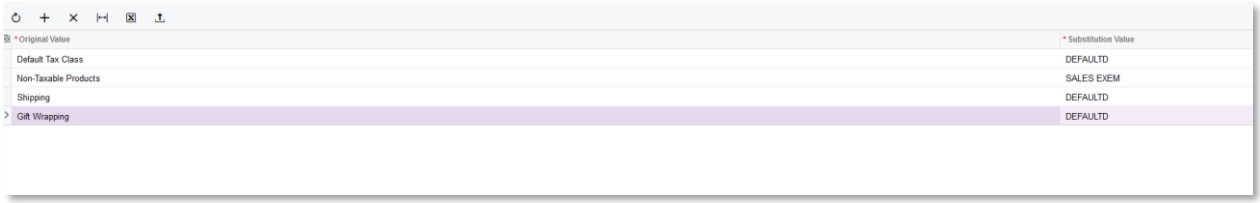
Tip: Tax Classes are equivalent to Tax Categories for inventory items in MYOB Advanced.

- You you should ensure that the GST Tax ID also exists in MYOB Advanced. Alternatively, in the Substitution List section, in the Tax List substitution list, you should create a mapping for the GST Tax ID to the Tax ID in MYOB Advanced, like this:

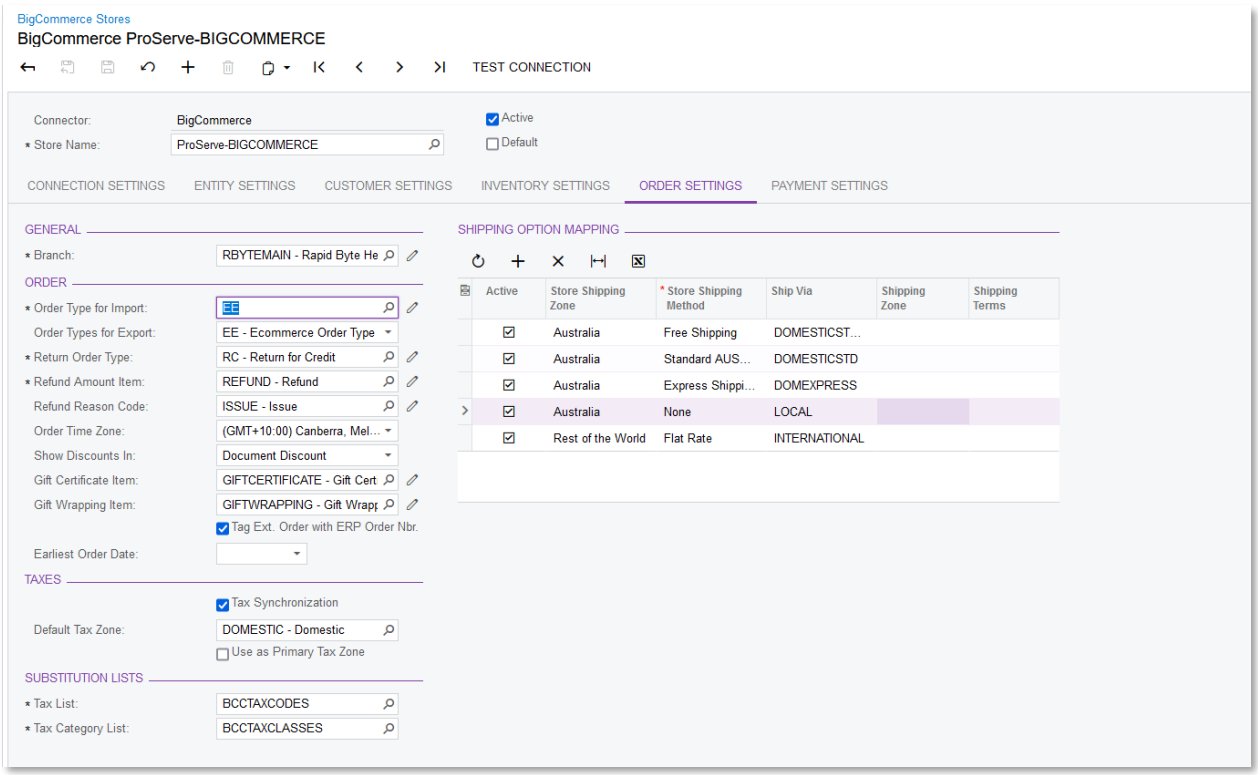


- With the above setup, when the Sales Orders are imported, they'll have the G1SALES GST tax code.

- The BCCTAXCLASSES Substitution List is used for mapping BigCommerce Tax Classes with MYOB Advanced Tax Categories – below is an example:

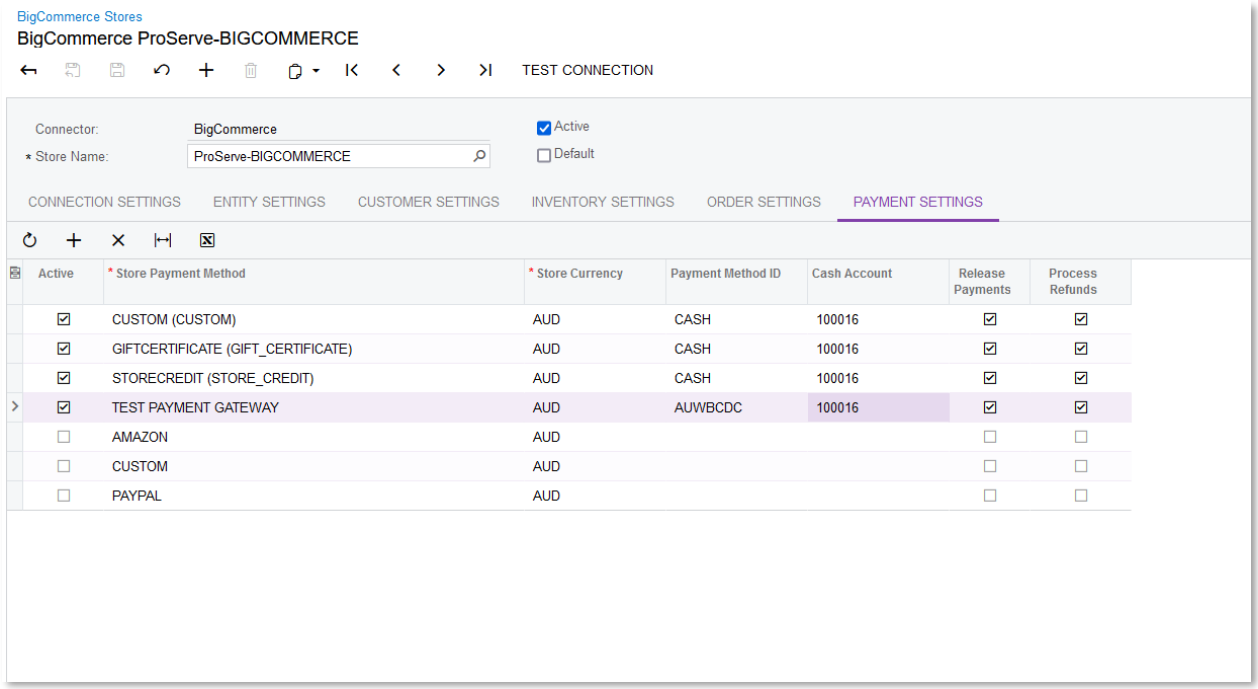


- Shipping Options Mapping: you can map the Store Shipping Zone and Store Shipping Method with MYOB Advanced's Ship Via Codes (and Shipping Zone and Shipping Terms if required):



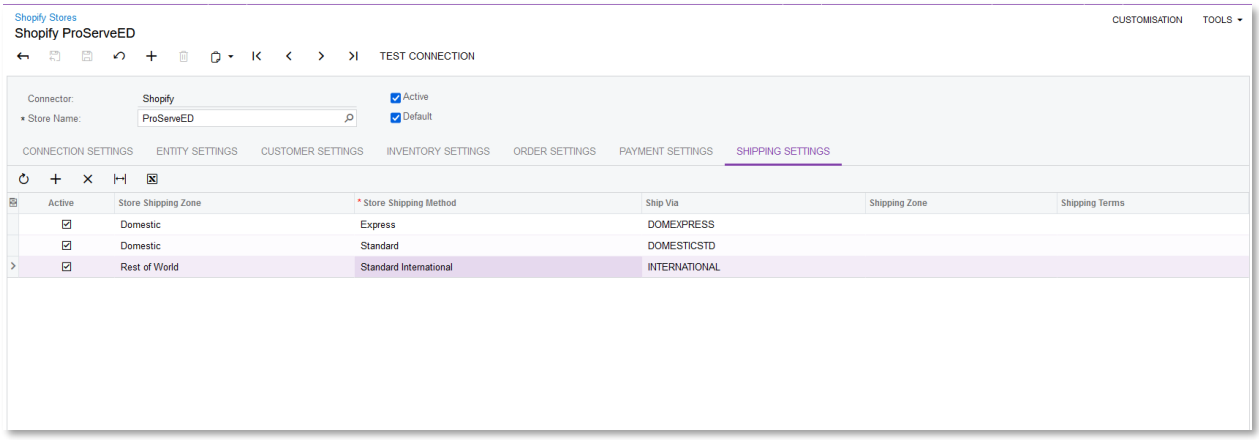
Payment Settings

In the Payment Settings tab (both Shopify Stores and BigCommerce Stores), map the store's Payment Methods with MYOB Advanced's Payment Methods and Cash Accounts.



Shipping Settings (Shopify Stores only)

Map the Store Shipping Zone and Store Shipping Method with MYOB Advanced's Ship Via Codes (and Shipping Zone and Shipping Terms if required):



Sales Order Processing – Shopify

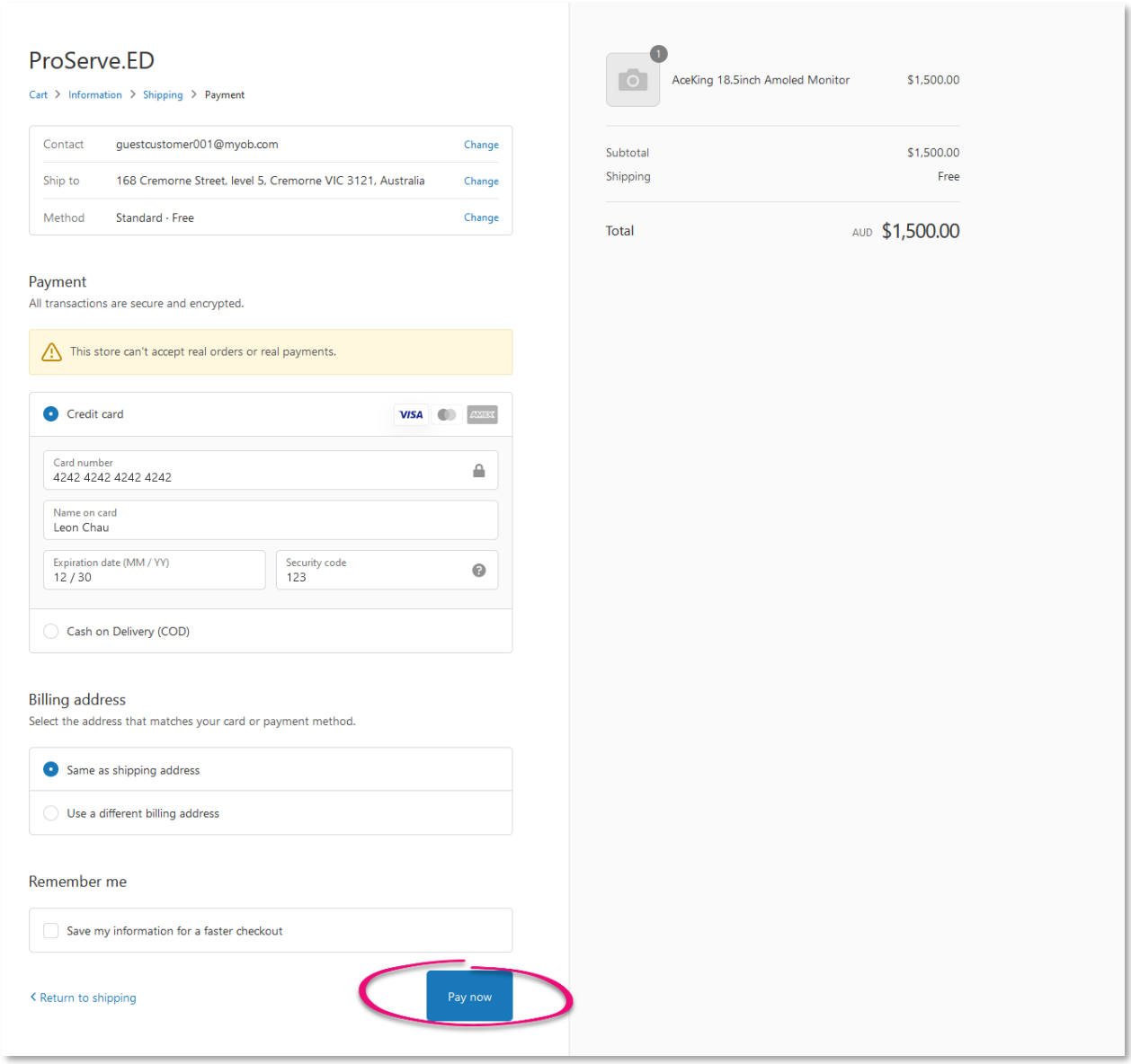
In the example below, synchronisation has been turned on for the following entities:

- Customer
- Customer Location
- Sales Order
- Stock Item
- Shipment
- Payment

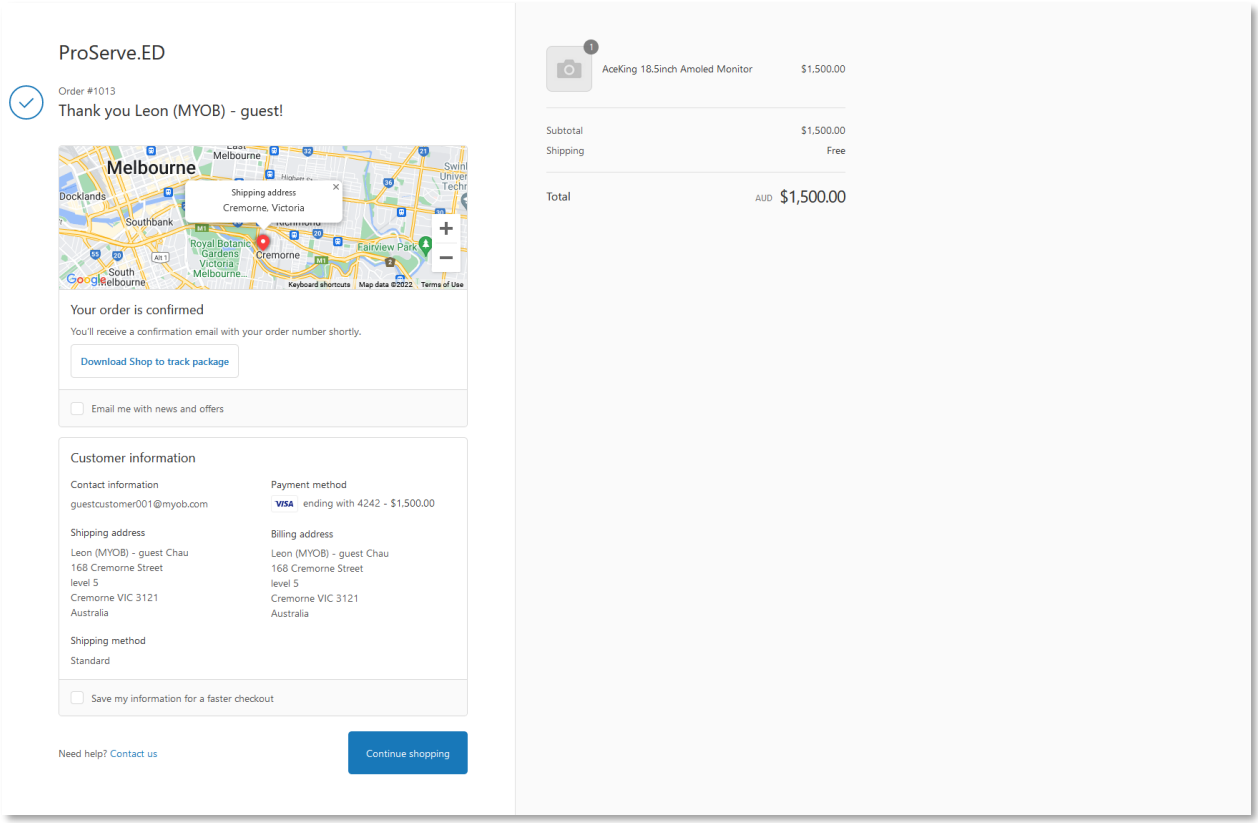
Therefore, every time an entity is created/updated in either system, a corresponding entity is also created/updated in the other system.

Raising a Sales Order:

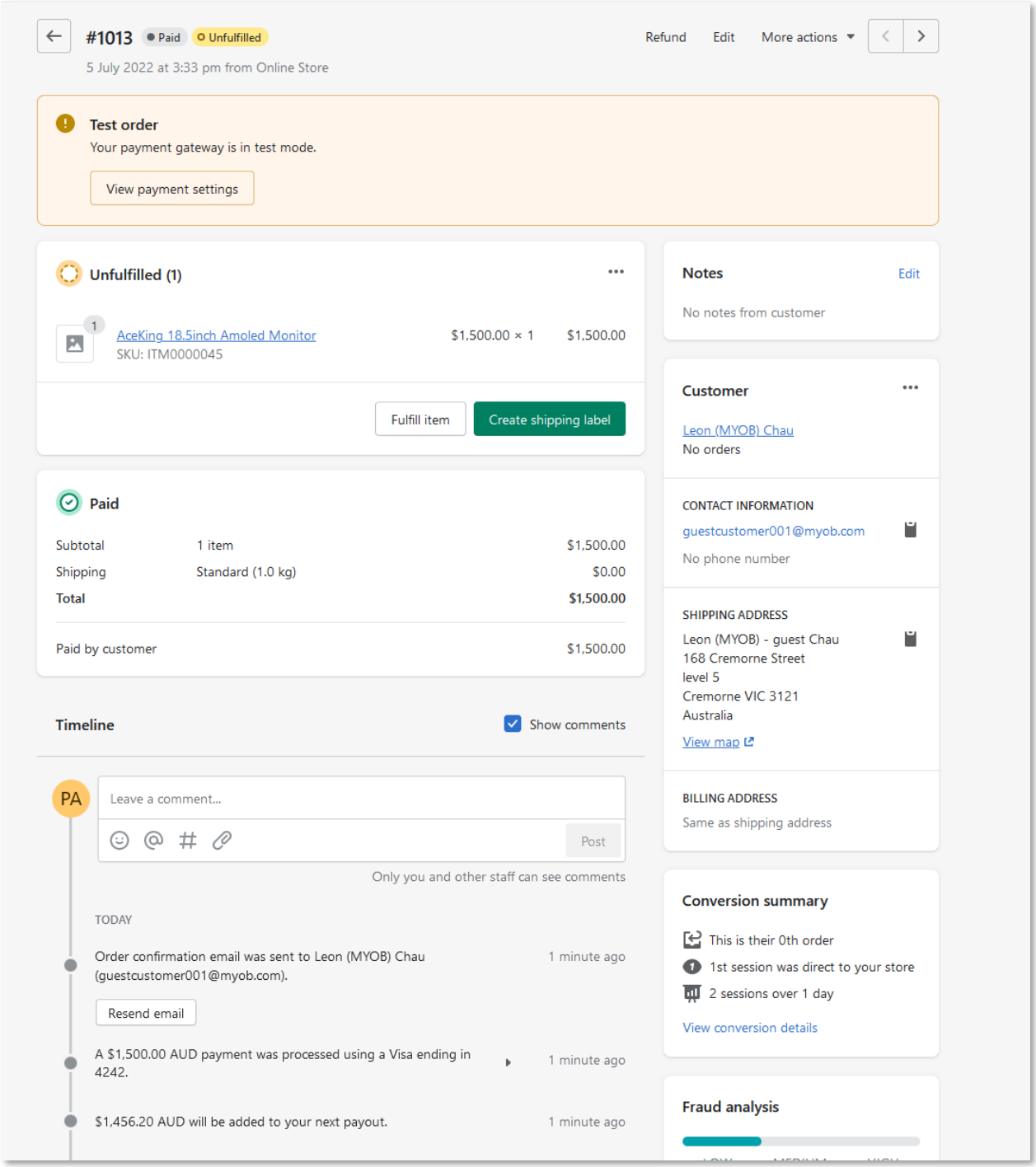
On the store front, the customers can browse through the product and add the product to Cart. After that, they would review the Cart, check out, enter the Contact Email Address, pick shipping method, enter credit card details, and pay:



Once a payment is completed, a Sales Order is generated on Shopify:



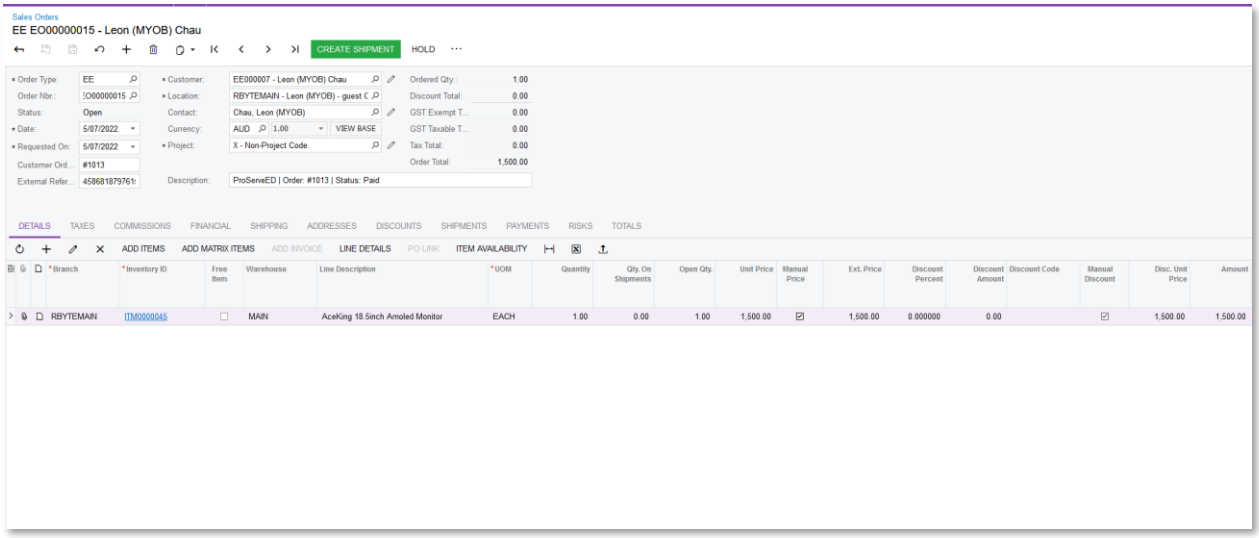
Here we're doing a checkout as a Guest (notice on the screen above, the field "Save my information for a faster checkout was left blank). In the backend, in the Orders list, you can see the Sales Order like below – notice the status of the Sales Order is "Paid", "Unfulfilled"



Syncing the Sales Order:

If the synchronisation was set up correctly, this Sales Order will then be synced to MYOB Advanced (if you have Real-Time Sync turned on and Real-Time Mode is Prepare & Process, the Sales Orders will be imported immediately; If you don't have real time sync turn on, you need to Prepare Data and Process Data for the Sales Order to be imported).

The Sales Order imported to MYOB Advanced looks like below:

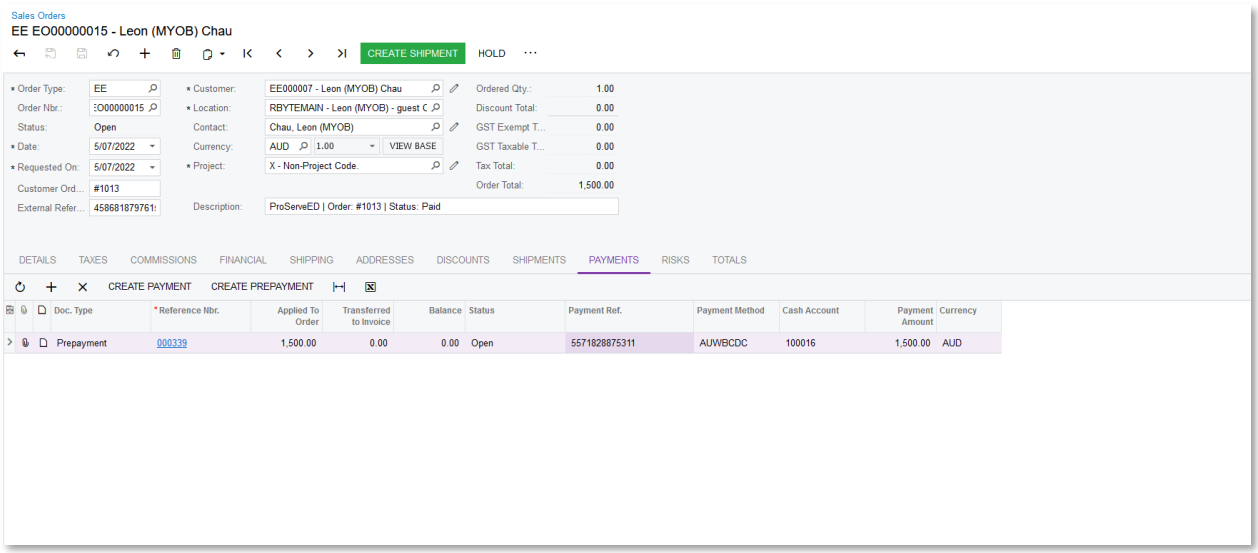


Notice that even when the customer checks out as Guest, the system still creates a customer using the Numbering Sequence configured in the Order Settings tab of the Shopify Stores screen (the numbering sequence used in this example starts with "EE"). This is because the legacy checkout as guest option is not available for Shopify. Instead, Shopify has streamlined their Customer Accounts process where each customer can register and have ShopPay account, instead of being a . Therefore, the system doesn't treat a guest customer as a guest customer, but instead, an actual customer, which will be then imported to MYOB Advanced.

If the customer checks out as an existing customer, the Sales Order will have the Customer ID of that existing customer. Also on this Sales Orders screen, the following fields are populated:

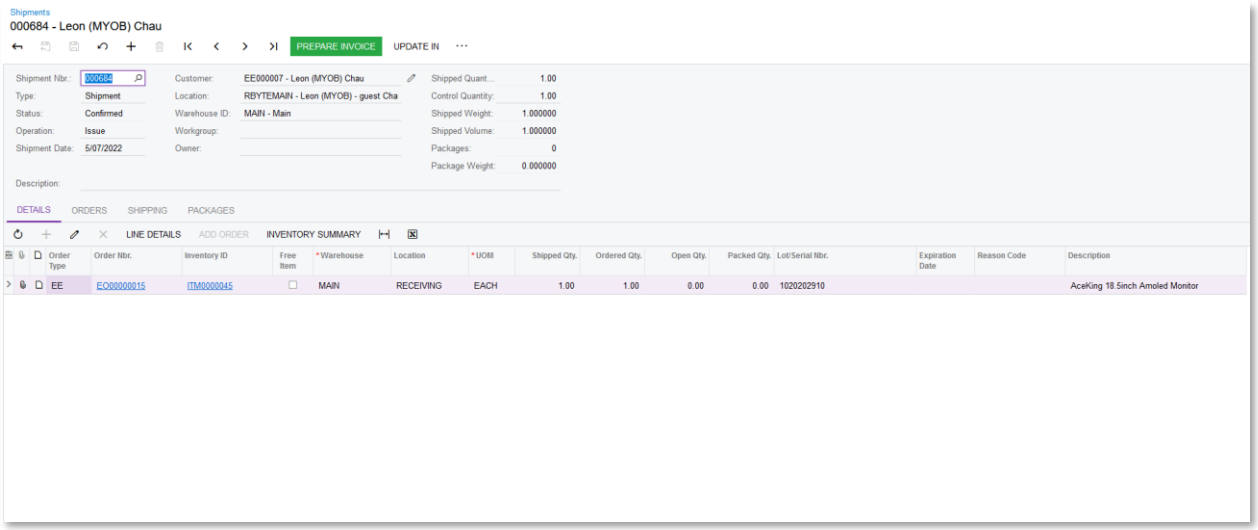
- Customer Order: populated with Shopify Order number
- External Reference: is populated with the External ID of the Sales Order Entity.
- Description: shows Store name, Order Number, and payment status.

In the Payments tab, the system automatically creates a Prepayment – in this example, the status of the Prepayment is "Open" because the field "Release Payment" of the Payment Method was set to true in the Payment Settings tab in the Shopify Store.

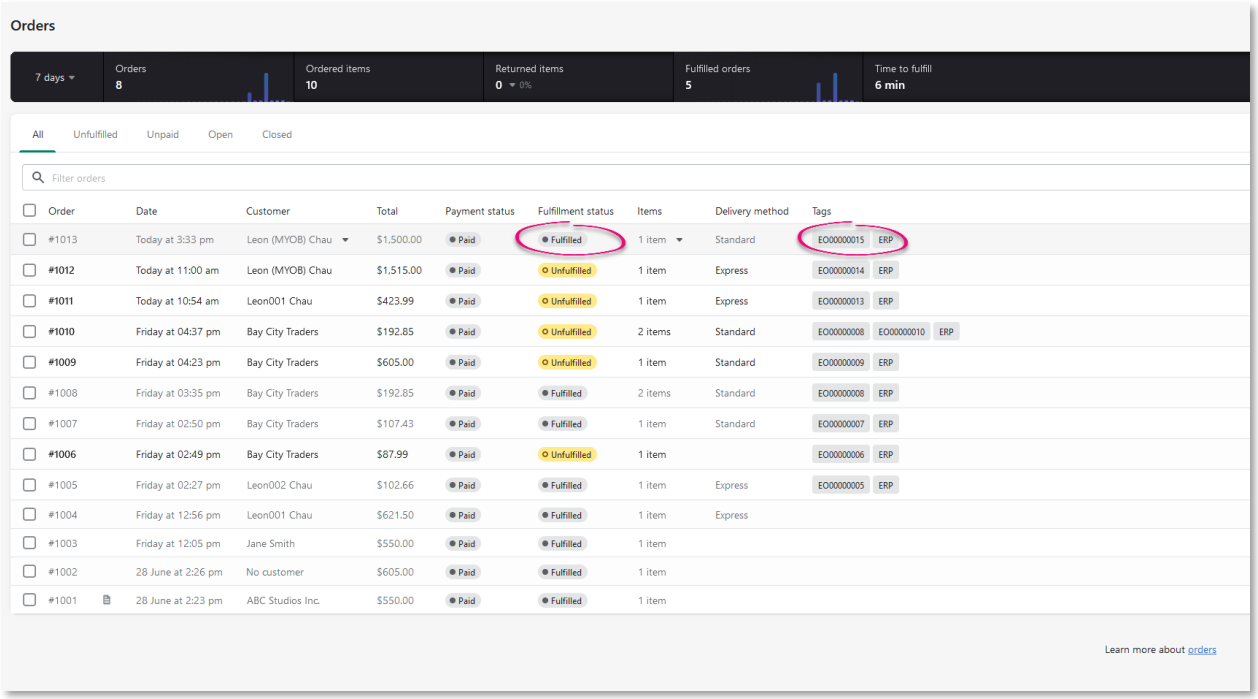


Fulfilling the Sales Order:

In MYOB Advanced, the warehouse employee will create a Shipment and Confirm the Shipment as usual:When the Shipment is Confirmed, and synched to Shopify (through the Shipment Entity), the Status of the Sales Order in Shopify is changed to Shipped.



When the Shipment is Confirmed, and synched to Shopify (through the Shipment Entity), the Status of the Sales Order in Shopify is changed to “Fulfilled”:

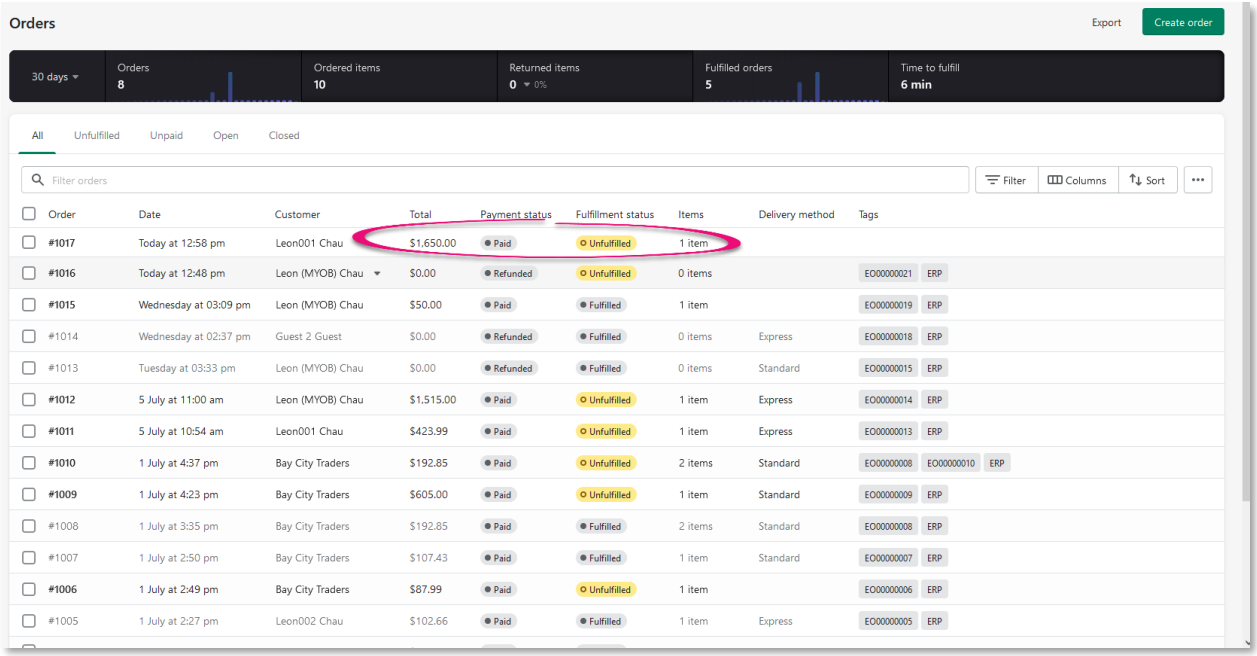


Preparing Invoice in MYOB Advanced:

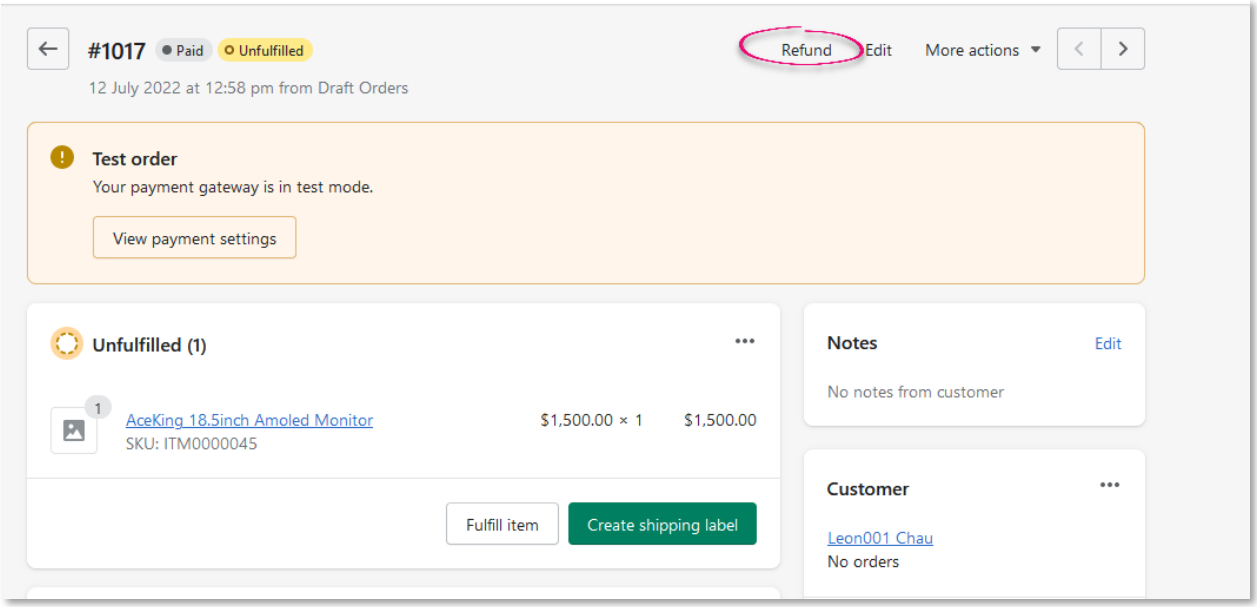
In MYOB Advanced, if the Prepayment created from Shopify was released, then when the users Prepare Invoice and Release Invoice for a Sales Order after the Shipment has been confirmed, the Status of the SO Invoice becomes “Closed”. However, if the Prepayment was not released, then the status of the SO Invoice becomes “Open”. The users will have to manually apply the Prepayment on the SO Invoice to close it.

Processing Customer Refund (for unfulfilled orders)

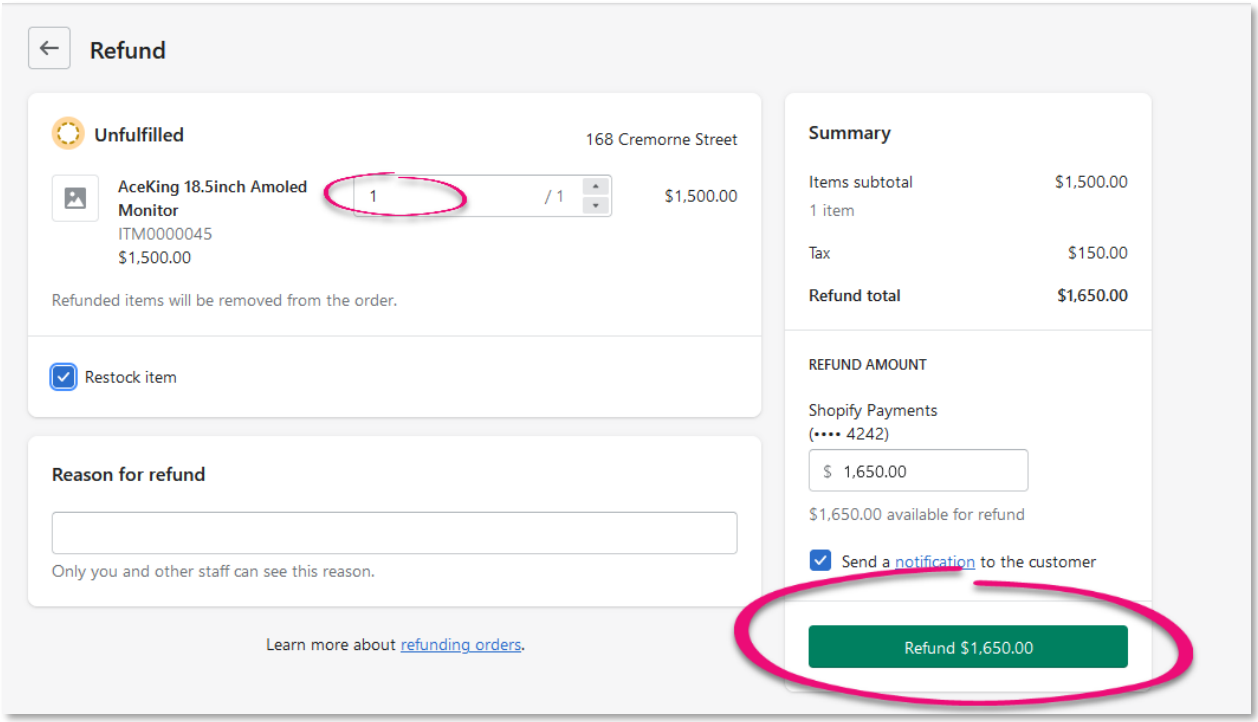
To process a Customer Refund for unfulfilled orders, firstly, in Shopify, you need to open a Sales Order that has been synced paid and synced to MYOB Advanced:



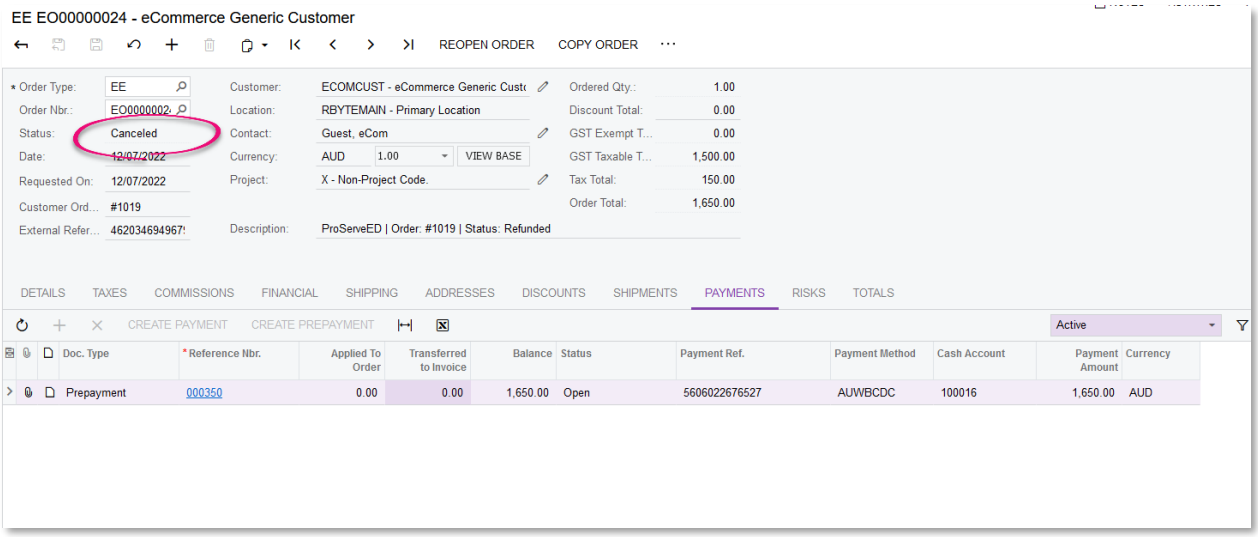
Click Refund on the Sales Order:



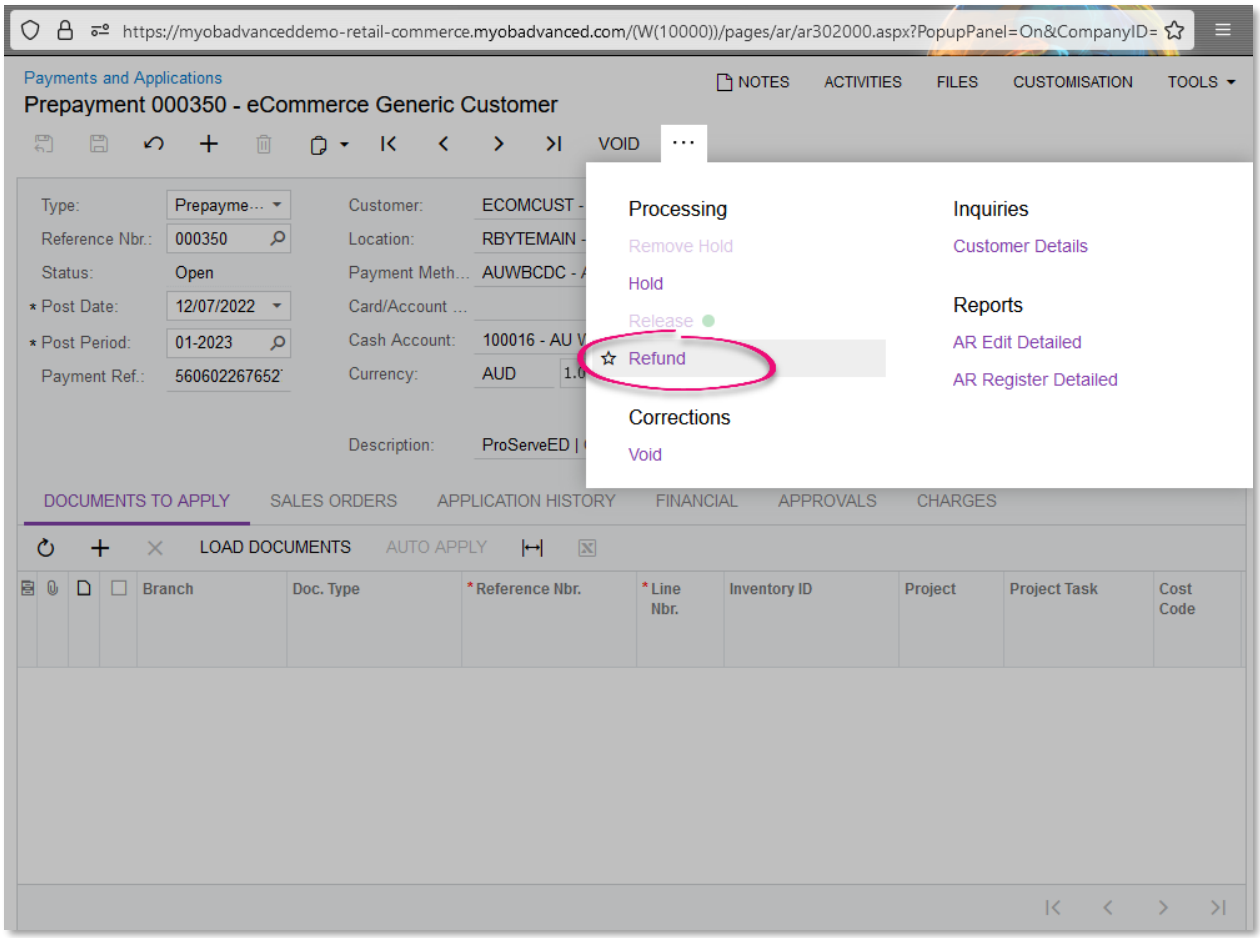
Enter the qty to be refunded, then click Refund:



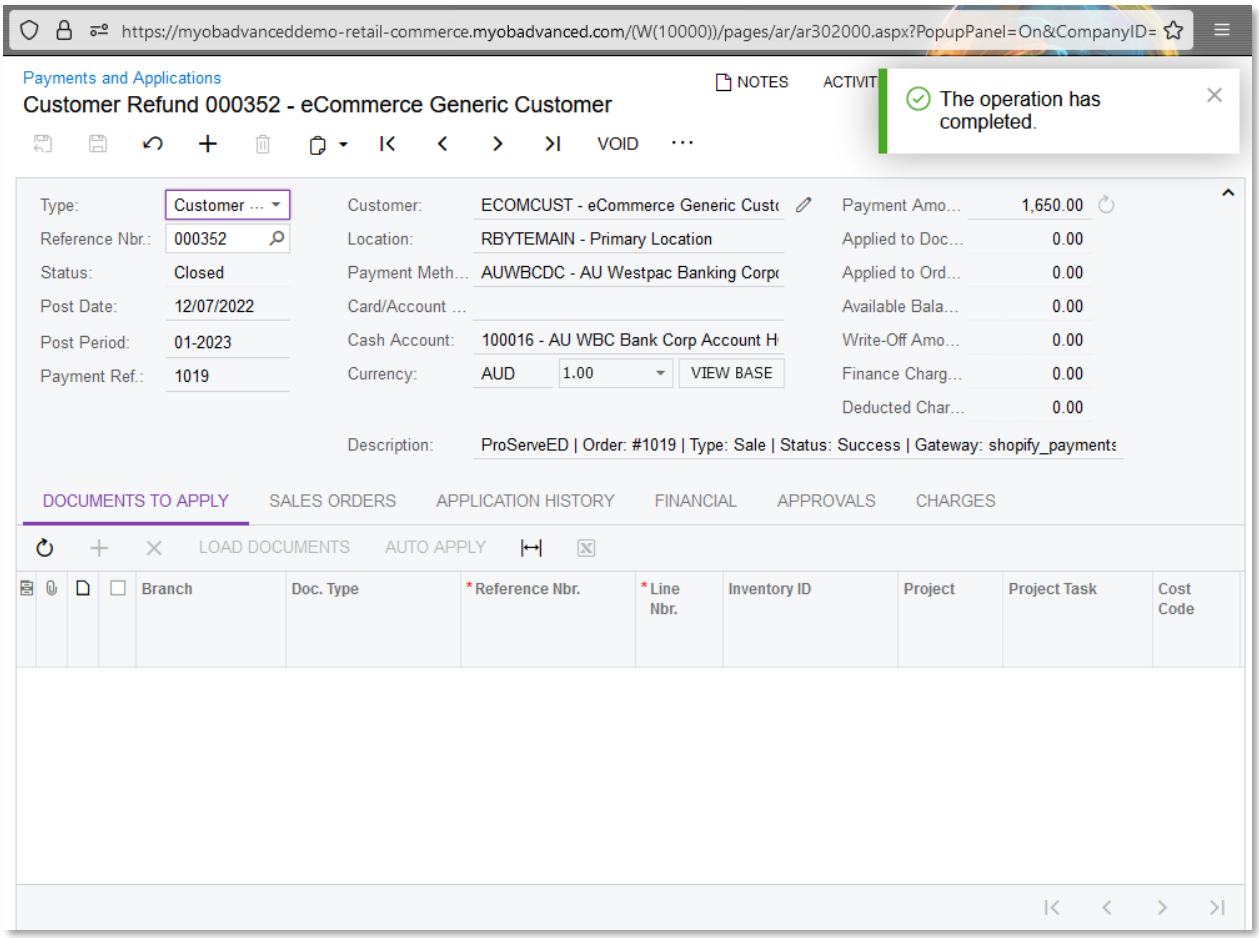
Go back to the Sales Order, notice the Sales Order status has been cancelled:



Open the Prepayment attached to the Sales Order, and select Actions > Refund to create a Customer Refund:

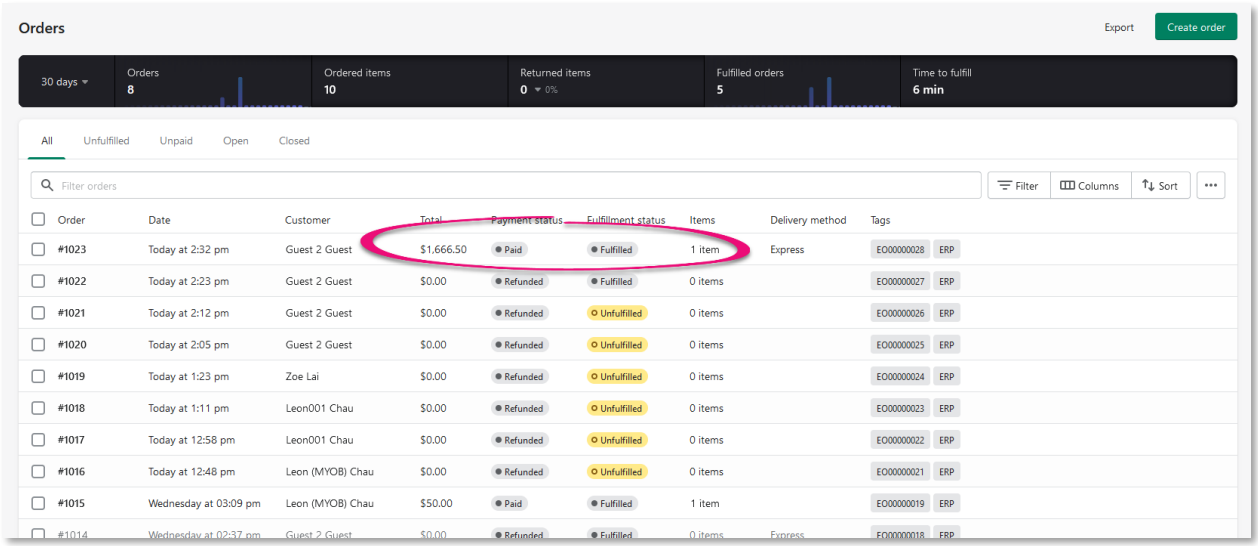


Release the Customer Refund to complete the Refund process:

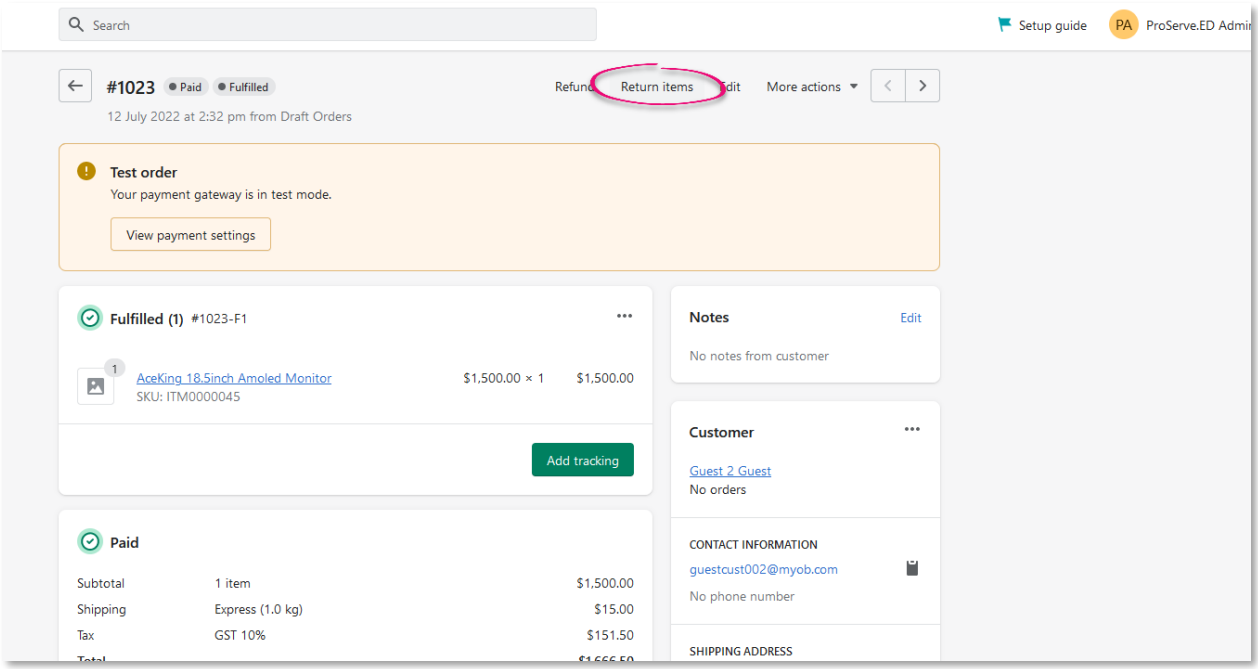


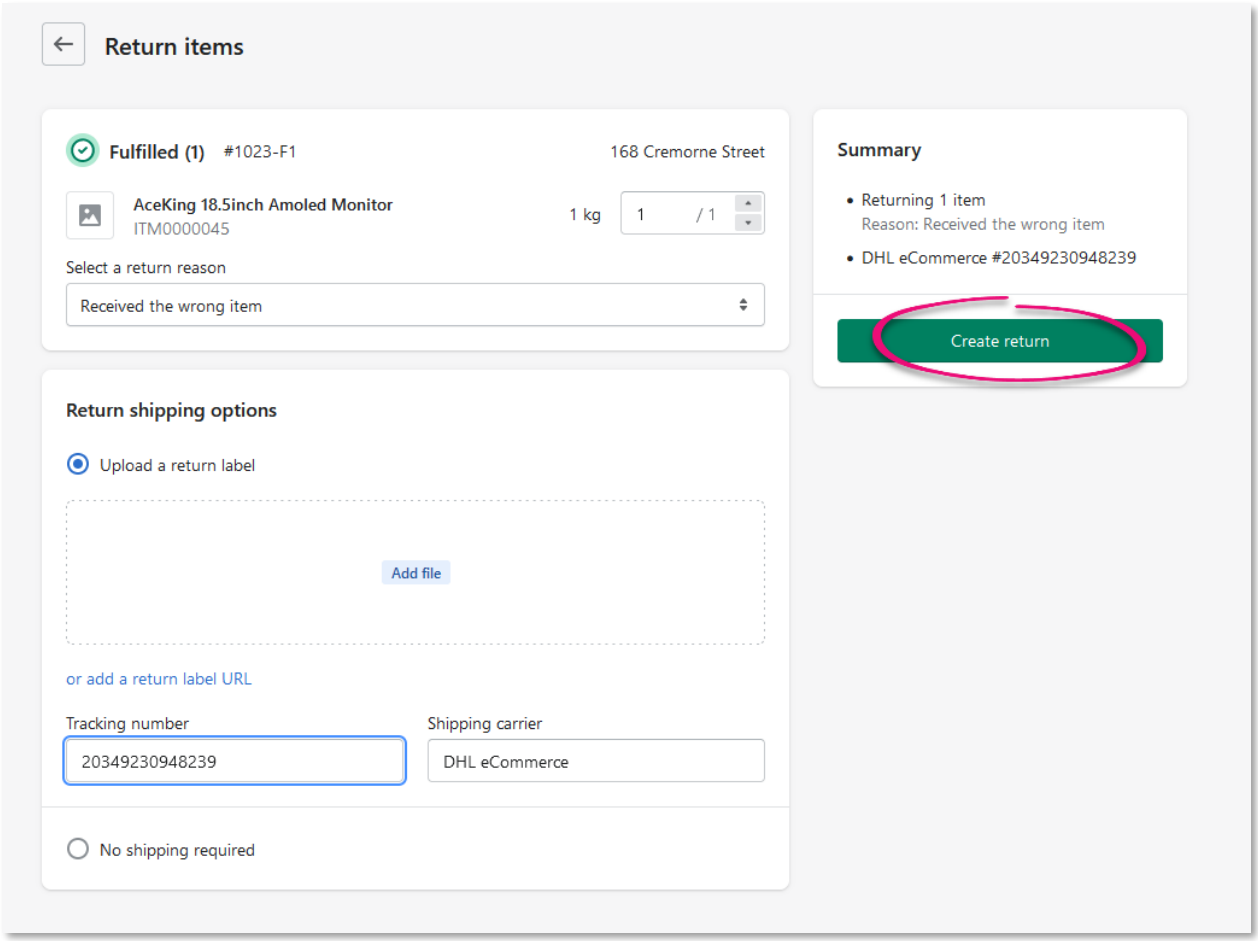
Processing Customer Refund (for fulfilled orders)

Open a Sales Order in Shopify that has been paid and fulfilled:

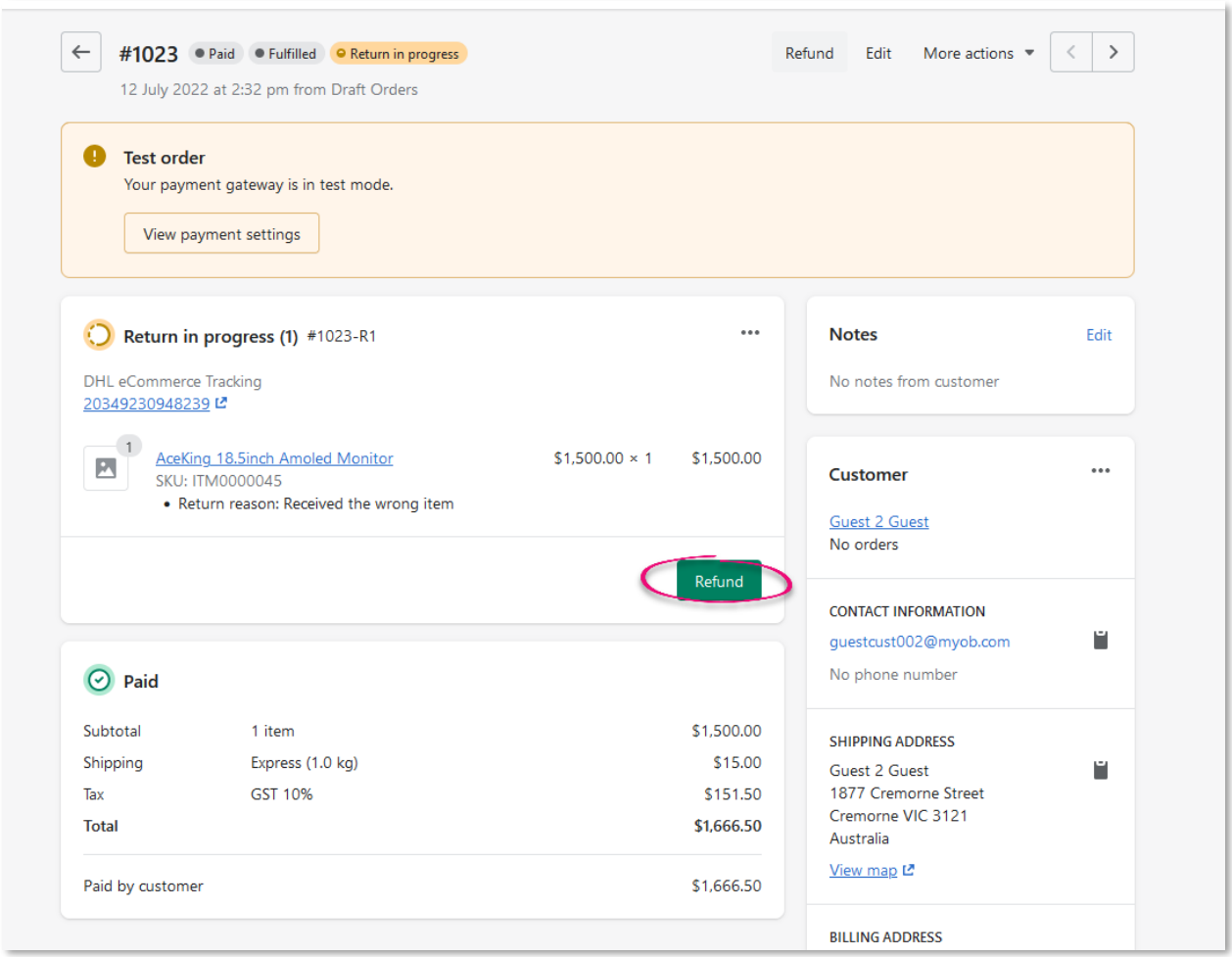


OPTIONAL: if you use Shopify to create a Return label, then click Return Item to generate and send a Return label to the customer:

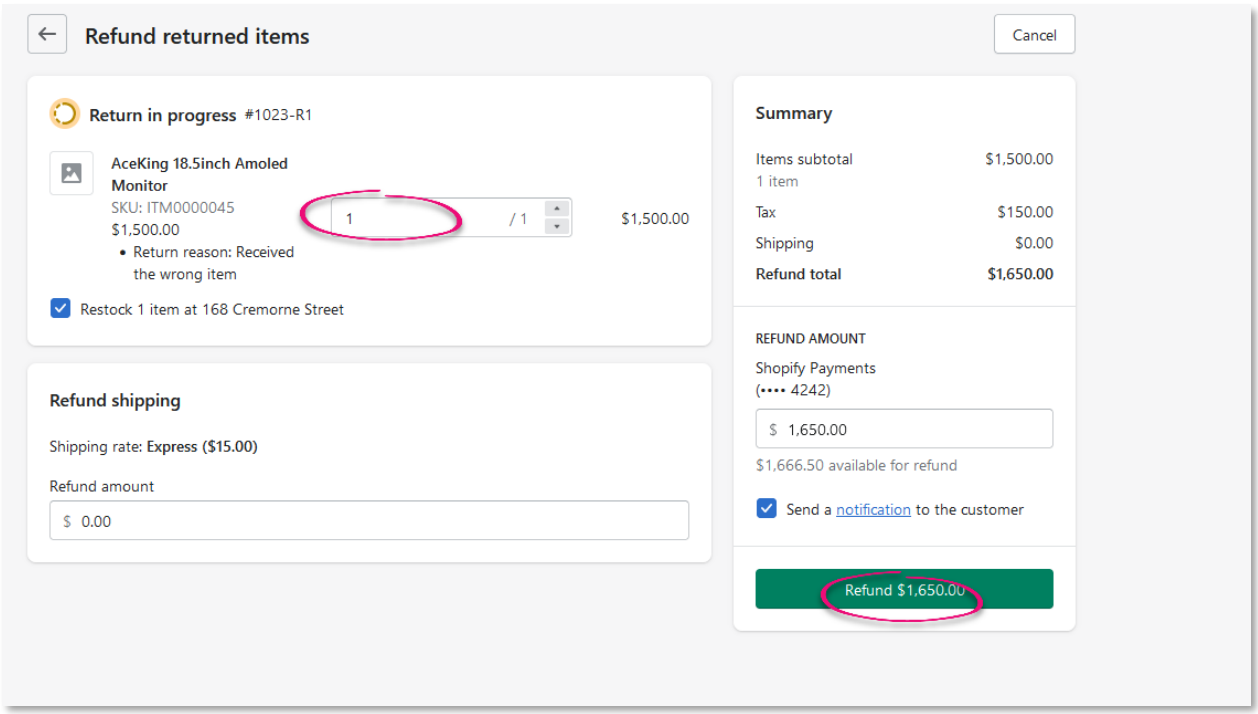




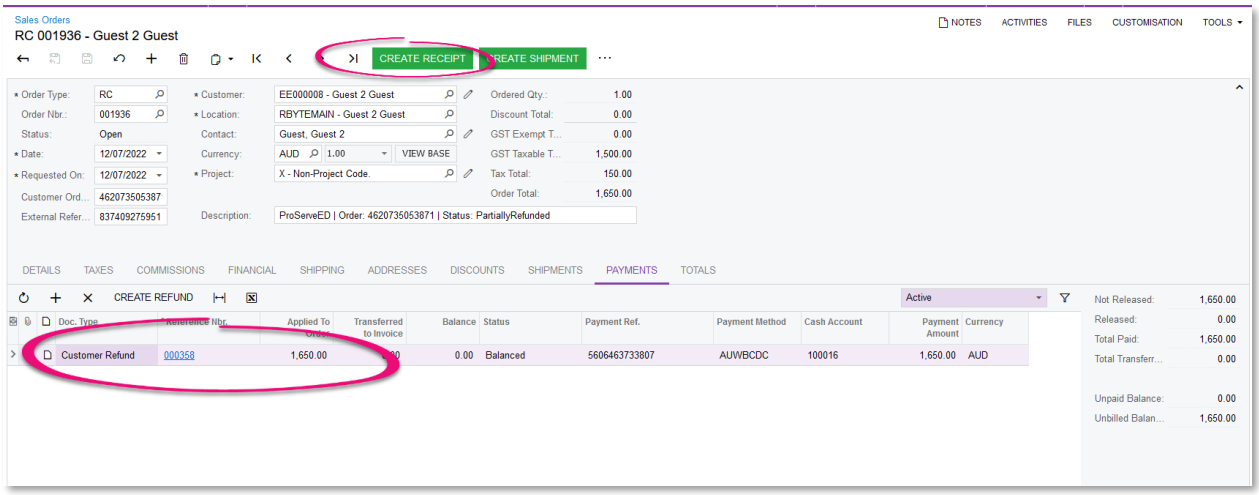
Click Refund to process a Refund:



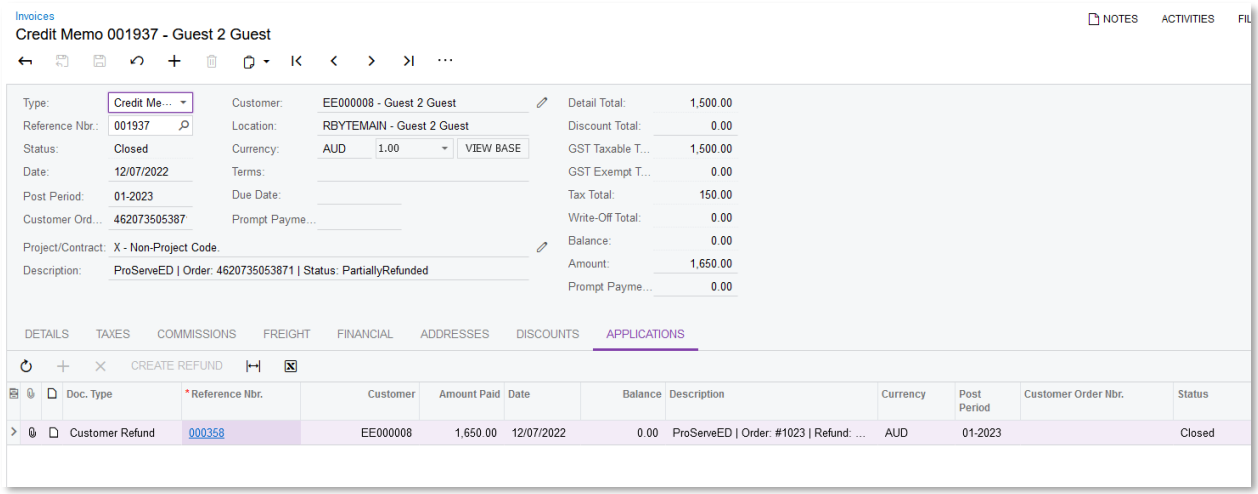
Enter the amount to be Refunded and proceed with the refund (OPTIONAL: if you want to refund the Shipping fee to the customer as well, you can enter the Refund Shipping amount:



In MYOB Advanced, the system automatically generates a Return order type, with a customer refund automatically applied against it. Select Create Receipt to bring the stock back into inventory:

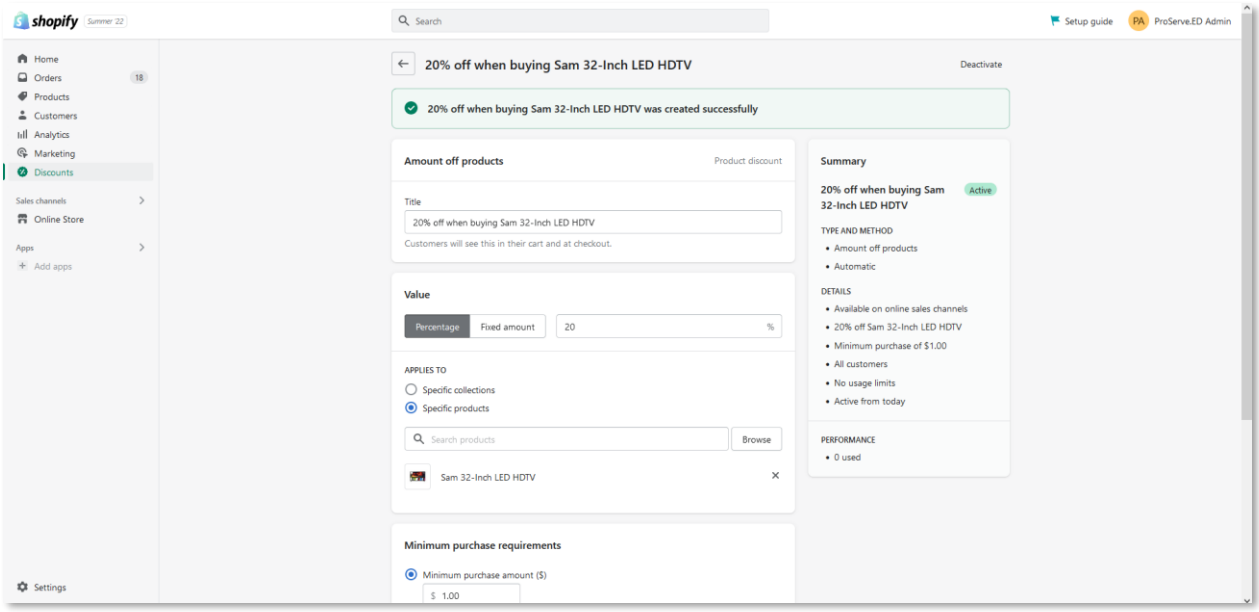


Once you Confirm Shipment and Prepare Invoice, MYOB Advanced generates a Credit Memo with the Customer Refund applied on it:

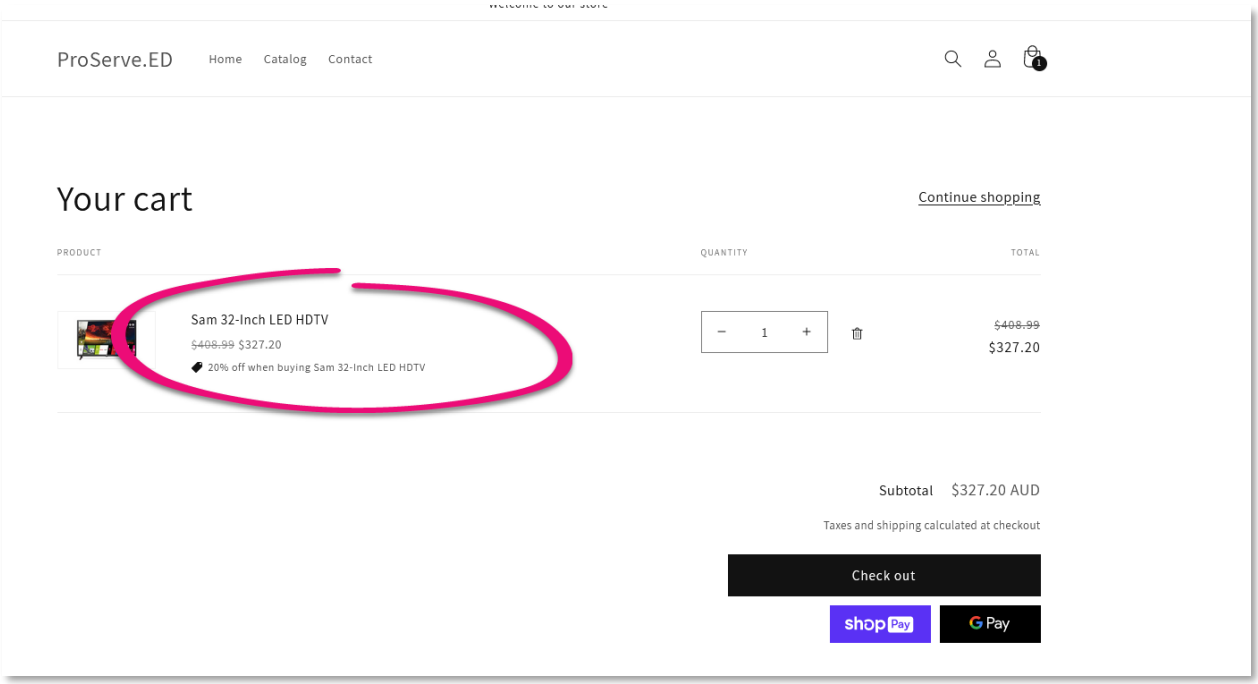


Discount

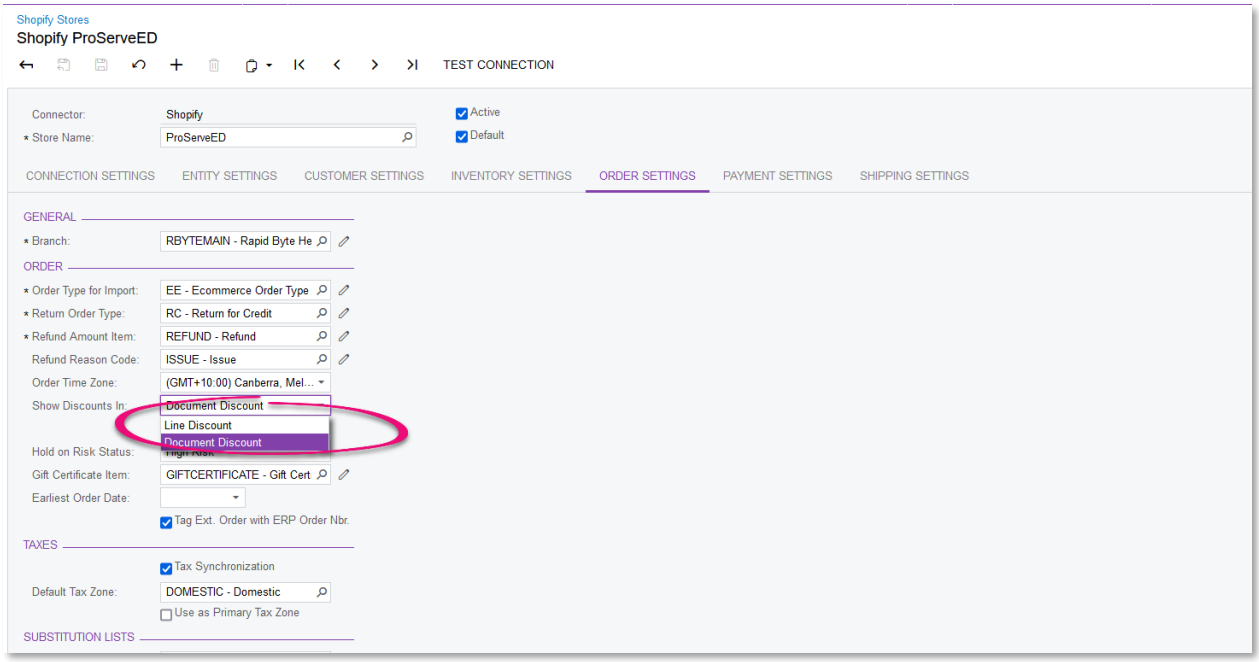
In Shopify, Discount is configured in the Discounts page – you can configure different types of Discounts in Shopify, such as discount off products, discount off orders, discount off shipping, or buy X get Y:



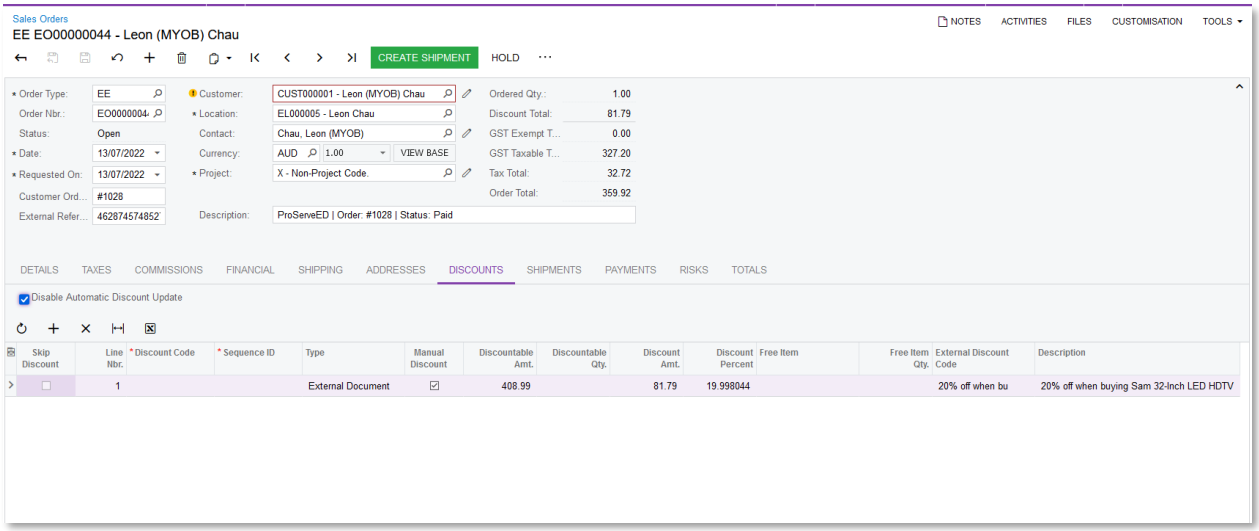
When the customer meets the condition for discounts, they will see the discount during checkout:



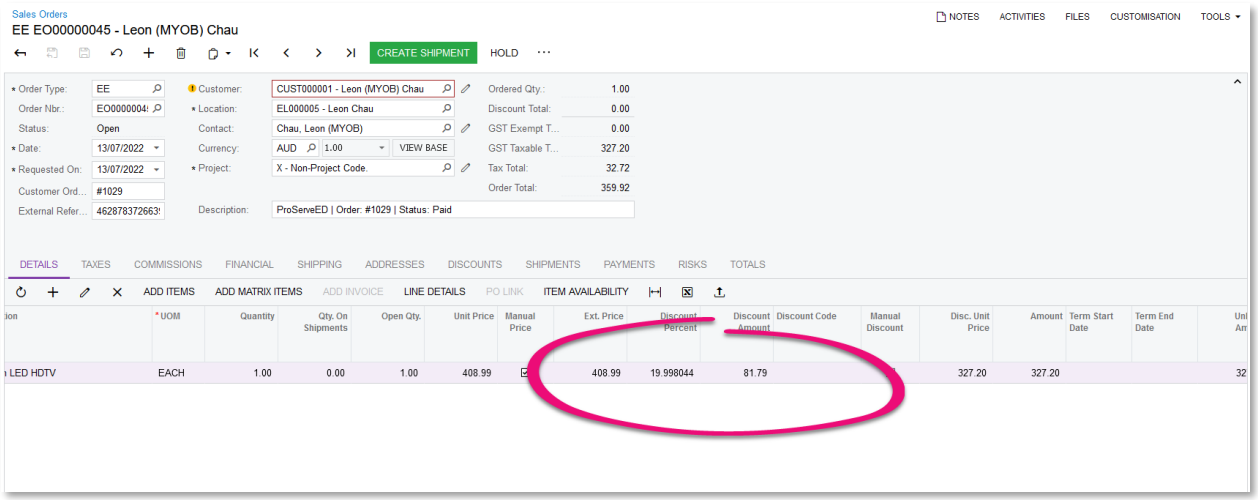
On the Shopify Stores, you have the option to show discount as a Line Discount or a Document Discount:



If you select Document Discount, when the Sales Order is synced to MYOB Advanced, the discount will be shown in the Discount tab of the Sales Order screen:



If you select Line Discount, when the Sales Order is imported into MYOB Advanced, the discount will be shown in the Details of the Sales Order:



Sales Order Processing – BigCommerce

In the example below, synchronisation has been turned on for the following entities:

- Customer
- Customer Location
- Sales Order
- Stock Item
- Shipment
- Payment
- Sales Category
- Stock Item
- Non-stock Items

Therefore, every time an entity is created/updated in either system, a corresponding entity is also created/updated in the other system.

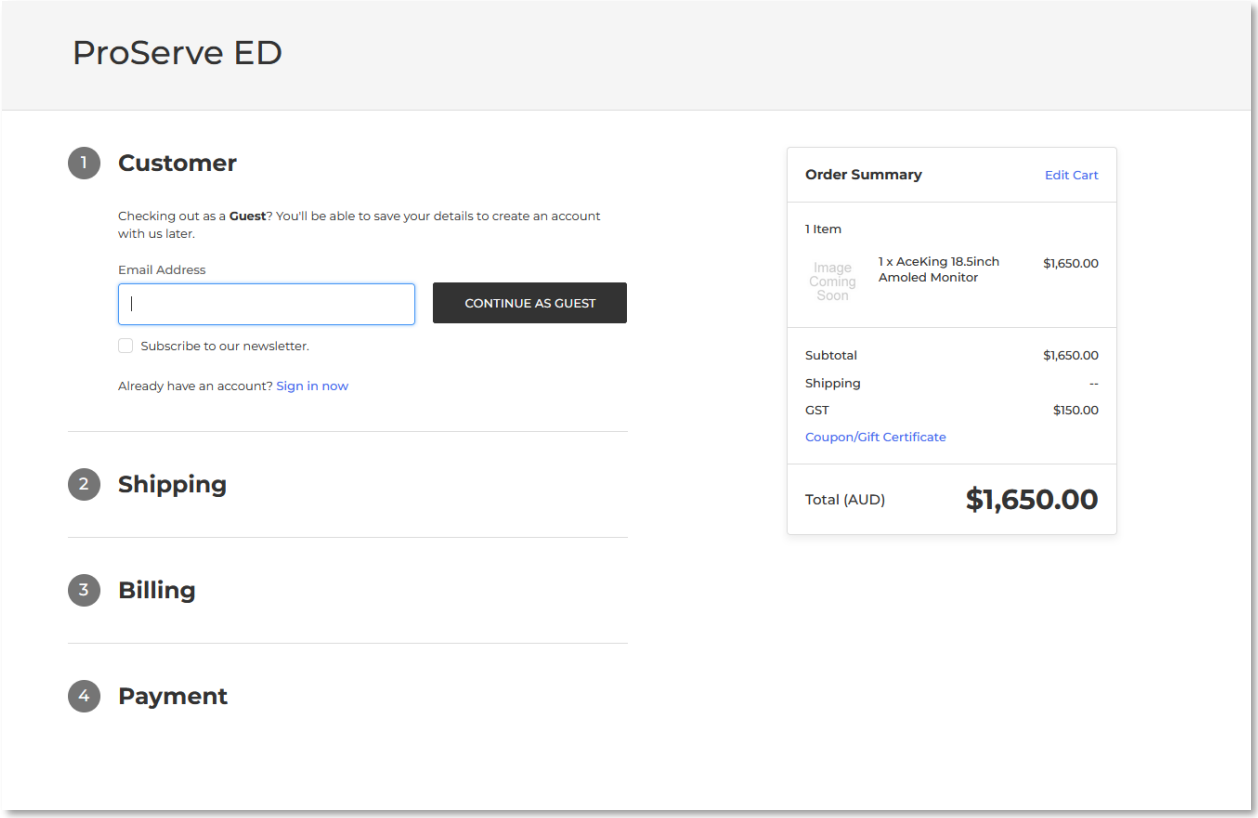
Raising a Sales Order:

On the store front, the customers can browse through the product and add the product to Cart. After that, they would review the cart and checkout.

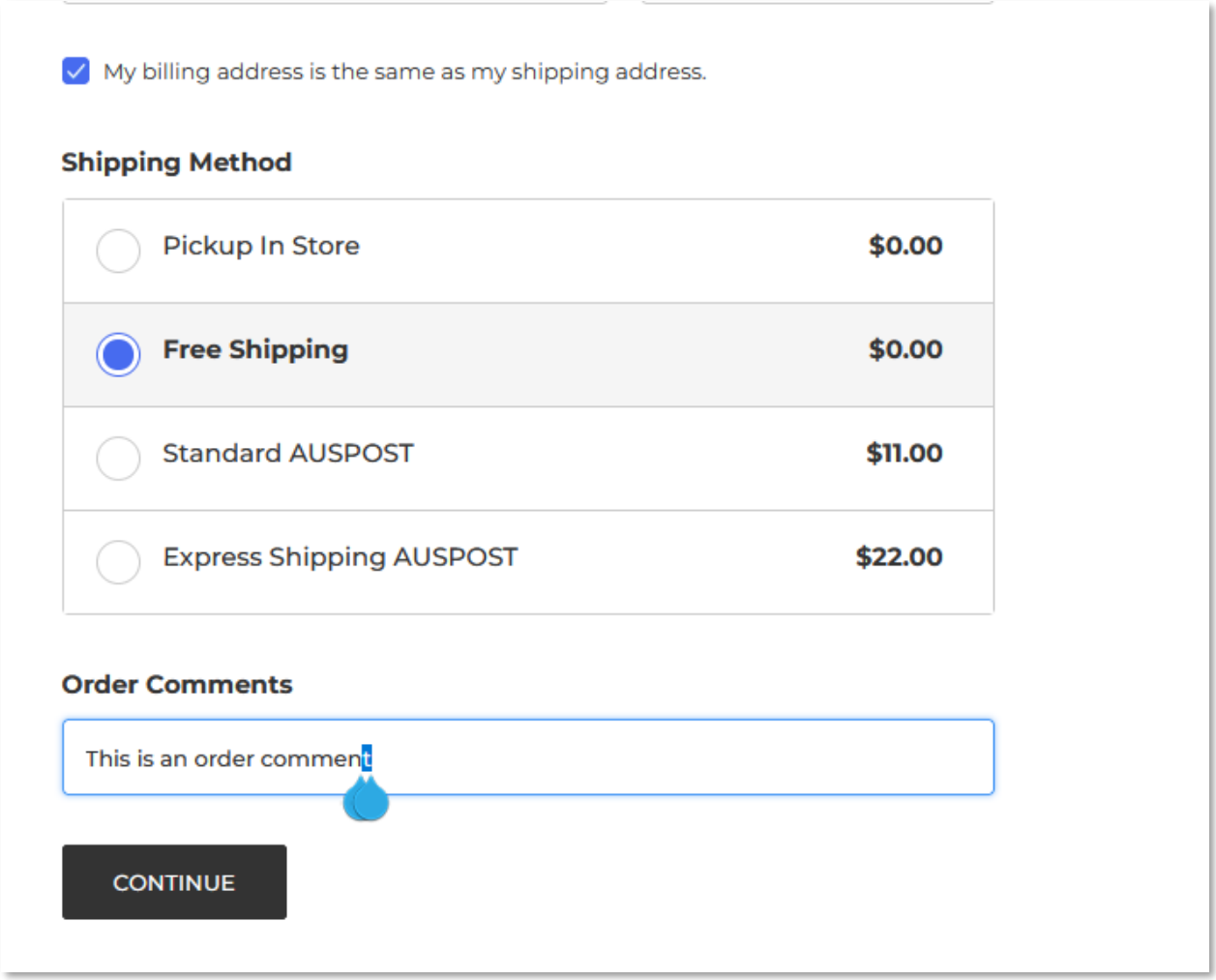
During checkout for BigCommerce, the customers can have an option to Check out as a Guest, or if they are an existing customer with the store, they can sign in and check out as an existing customer.

The difference between checking out as a guest and checking out as a customer is, if the end user checks out as a guest, when a Sales Order from BigCommerce is imported into MYOB Advanced, the system will insert the generic customer specified on the Order Settings tab of the BigCommerce store as the Customer ID on the Sales Order. If the end user checks out as a customer, and this customer exists in MYOB Advanced, the system will use the existing Customer ID on the Sales Order. If this is an existing customer in BigCommerce, but does not exist in MYOB Advanced, the system will create a Customer in MYOB Advanced using the Customer Class and Customer Numbering sequence specified in the Order Settings tab of the BigCommerce store.

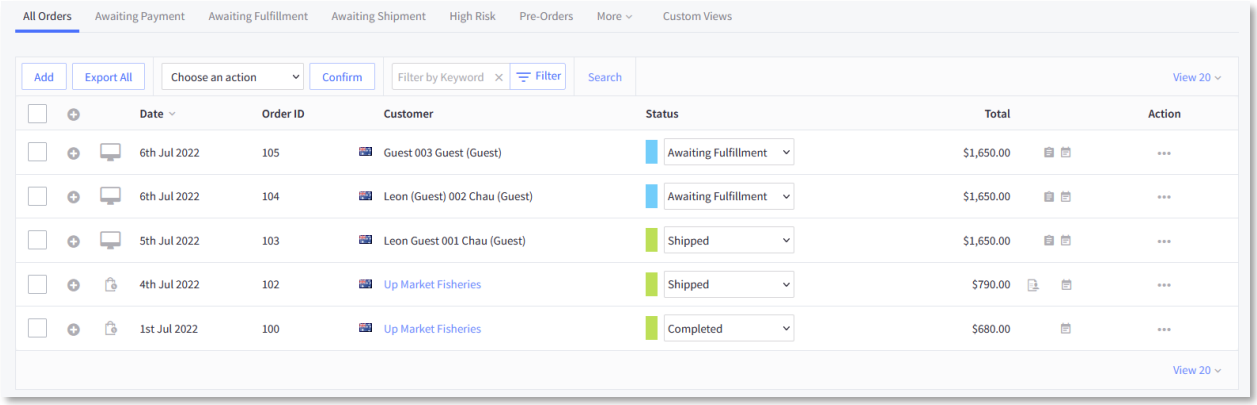
In the below example, we will be checking out as a guest.



During checkout, you can add an Order Comment. This Order Comment when synced to MYOB Advanced will be come a Note on the Sales Order.



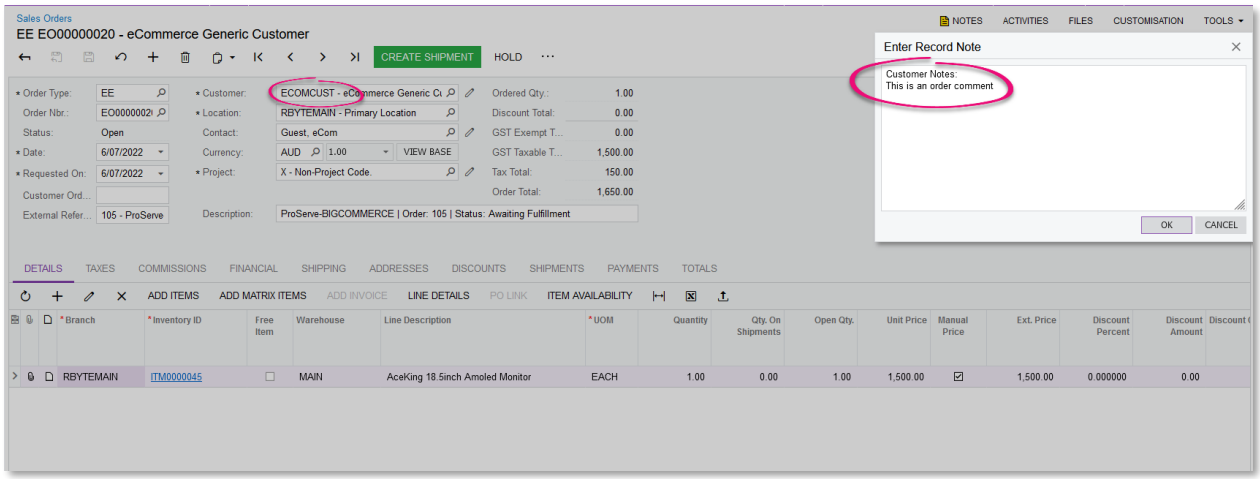
After the payment is captured, then a Sales Order is created in BigComerce. You can view the Sales Order in the backend of BigCommerce with the Status “Awaiting Fulfillment”:



Syncing the Sales Order:

If the synchronisation was set up correctly, this Sales Order will then be synced to MYOB Advanced (if you have Real-Time Sync turned on and Real-Time Mode is Prepare & Process, the Sales Orders will be imported immediately; If you don’t have real time sync turn on, you need to Prepare Data and Process Data for the Sales Order to be imported).

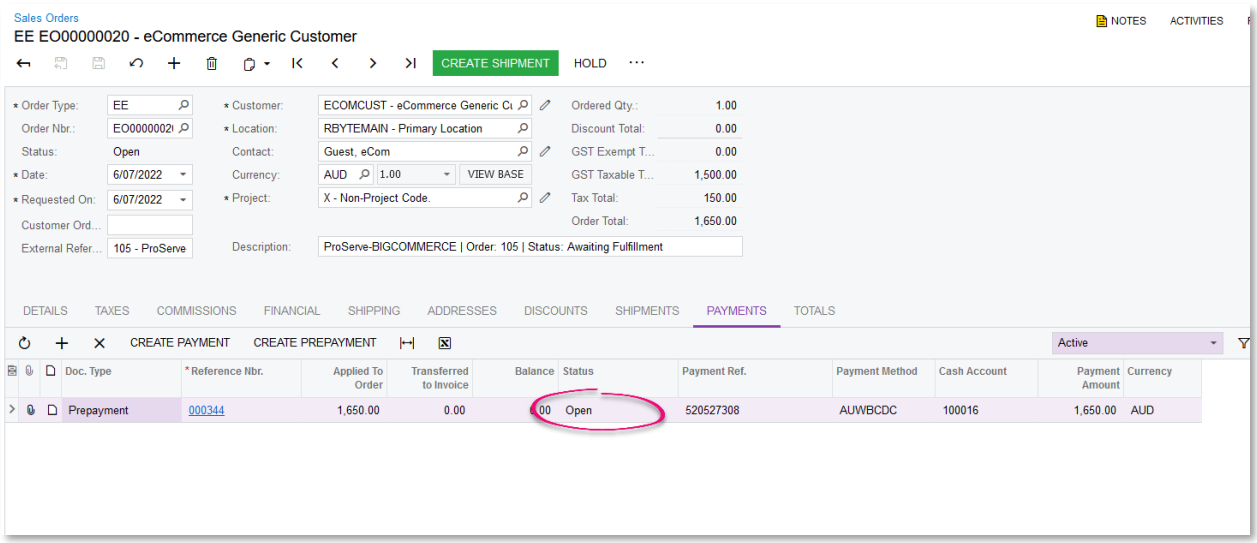
The Sales Order imported to MYOB Advanced looks like below:



Because we're checking out as a Guest, when imported to MYOB Advanced, the system uses the generic guest customer that we specified on the Orders Settings tab of the BigCommerce store. Also on this Sales Orders screen, the following fields are populated:

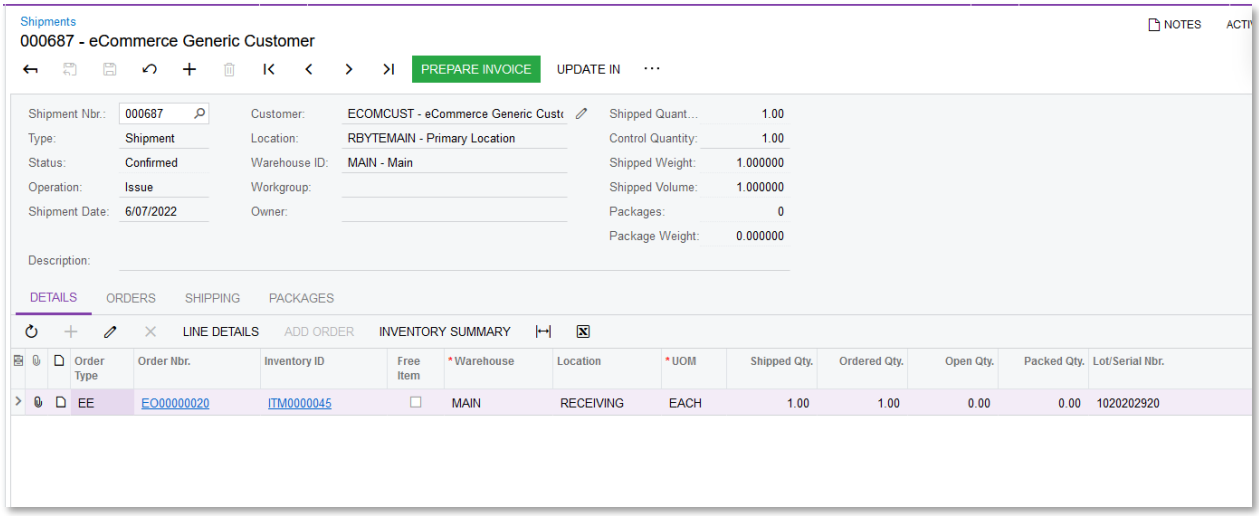
- Customer Order: populated with Shopify Order number
- External Reference: is populated with the External ID of the Sales Order Entity.
- Description: shows Store name, Order Number, and payment status.
- Order Comment is synced through as a Sales Order note.

In the Payments tab, the system automatically creates a Prepayment – in this example, the status of the Prepayment is "Open" because the field "Release Payment" of the Payment Method was set to true in the Payment Settings tab in the BigCommerce Store.

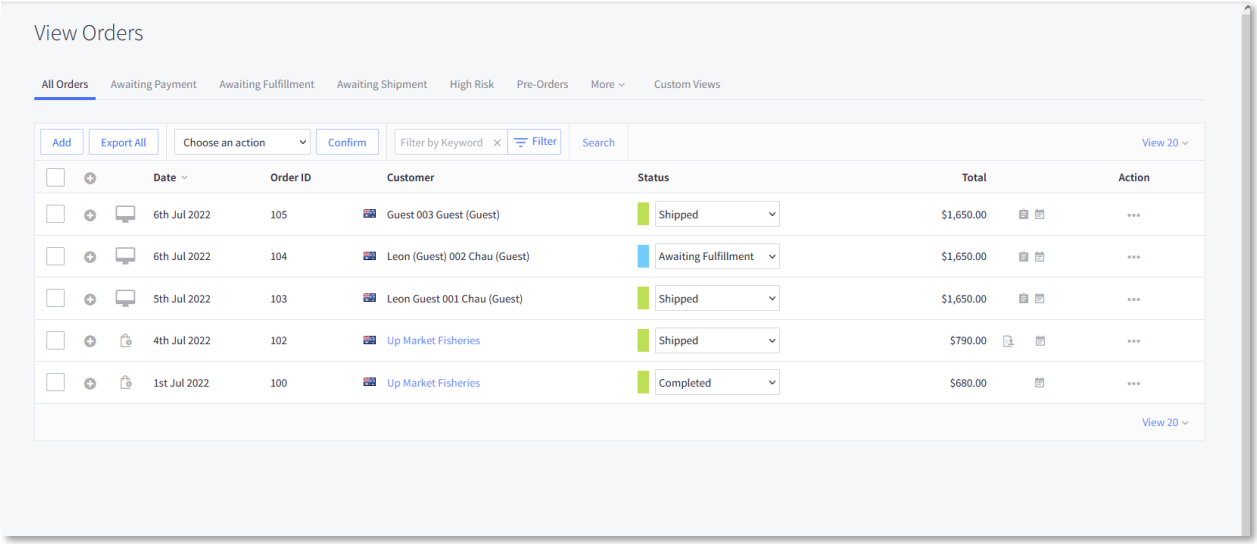


Fulfilling the Sales Order:

In MYOB Advanced, the warehouse employee will create a Shipment and Confirm the Shipment as usual:



When the Shipment is Confirmed, and synched to BigCommerce (through the Shipment Entity), the Status of the Sales Order in BigCommerce is changed to “Shipped”:

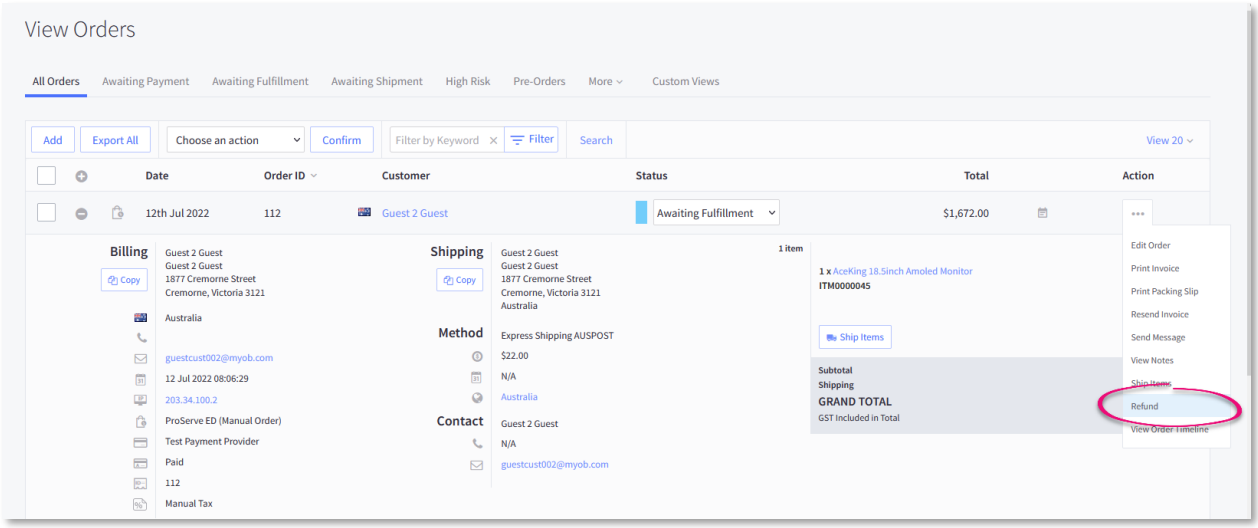


Preparing Invoice in MYOB Advanced:

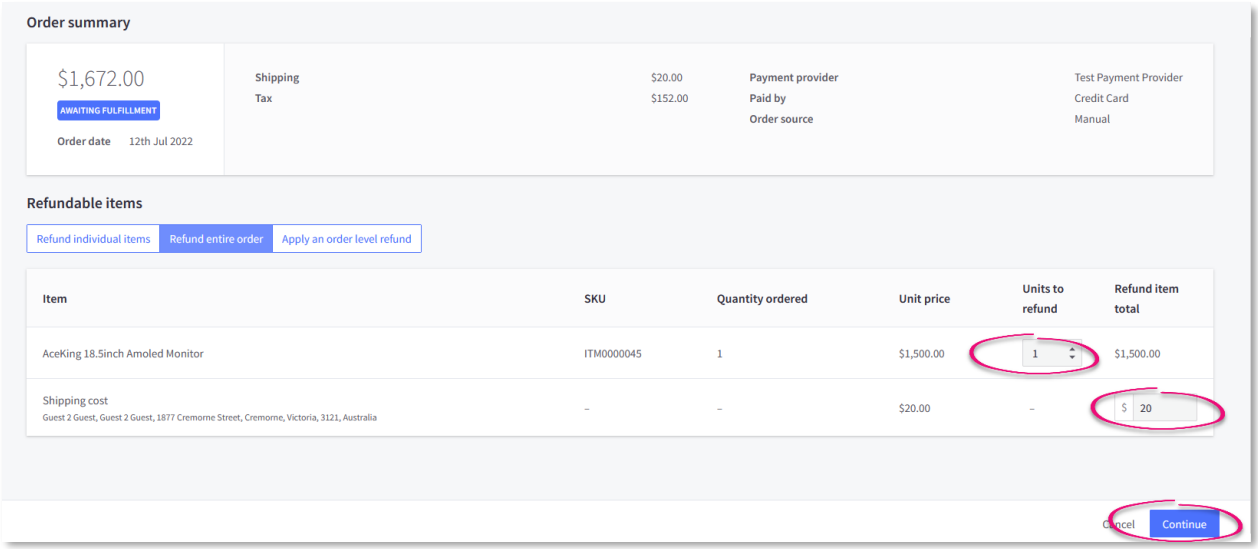
In MYOB Advanced, if the Prepayment created from Shopify was released, then when the users Prepare Invoice and Release Invoice for a Sales Order after the Shipment has been confirmed, the Status of the SO Invoice becomes “Closed”. However, if the Prepayment was not released, then the status of the SO Invoice becomes “Open”. The users will have to manually apply the Prepayment on the SO Invoice to close it.

Processing Customer Refund (for unfulfilled orders)

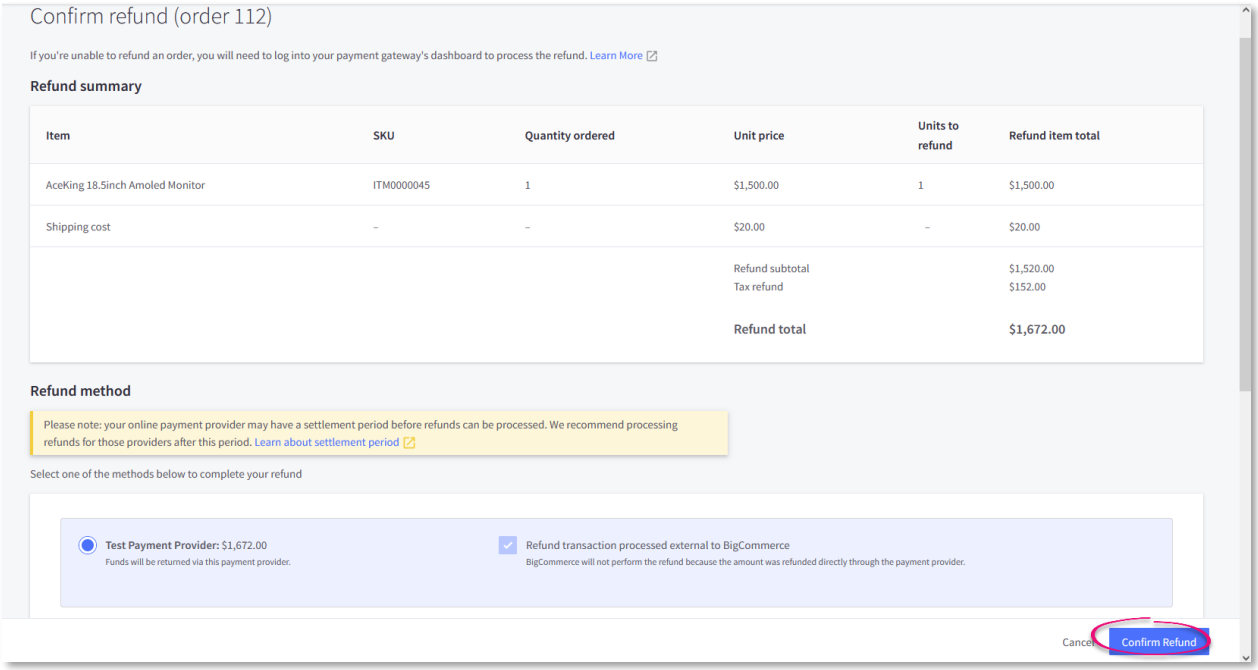
To refund a Sales Order that not been fulfilled, go to the Orders list, select an order with the Status “Awaiting Fulfillment” – this order must have already been synced into MYOB Advanced. Select Refund:



Specify the qty to refund, as well as the Shipping Cost, since you have not shipped this order yet:

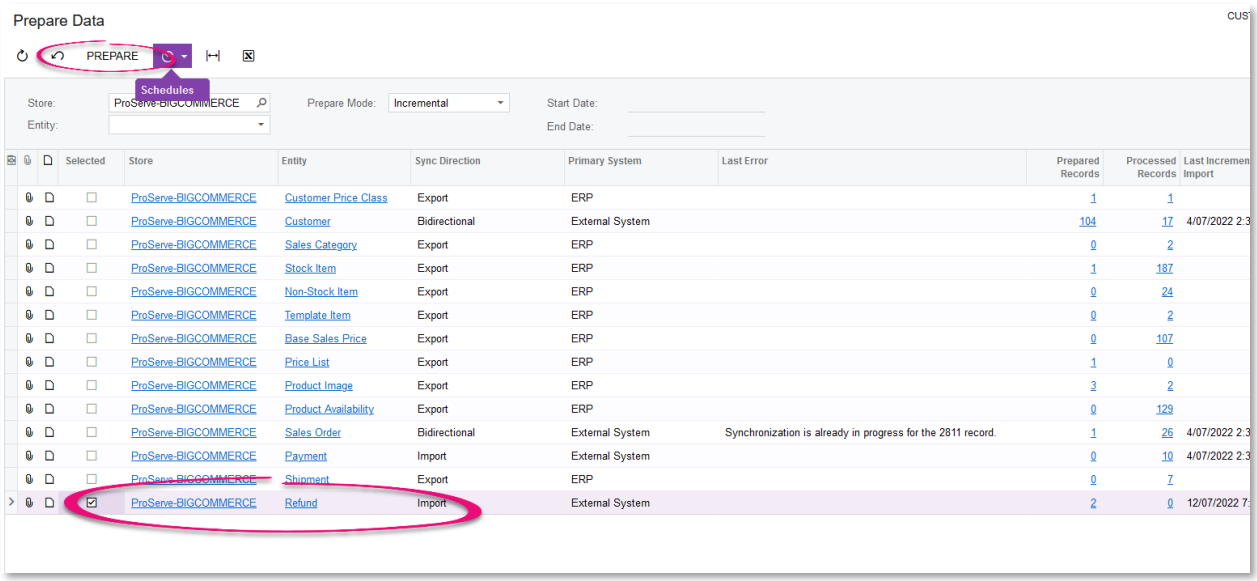


Select the Payment Method to refund and confirm:

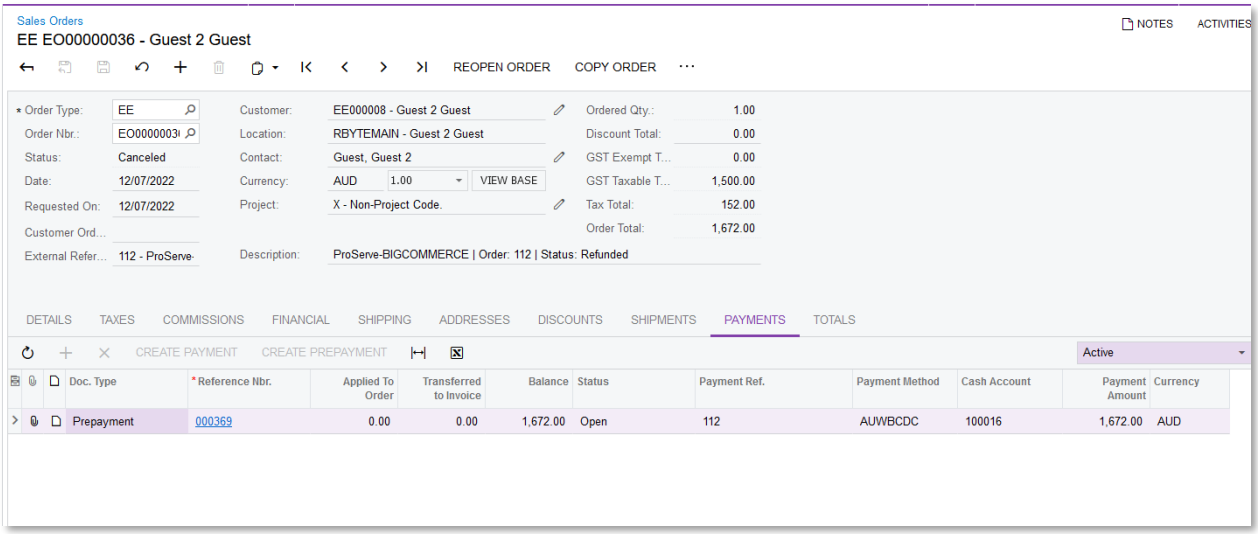


BigCommerce will Refund the money to the original payment method of the customer.

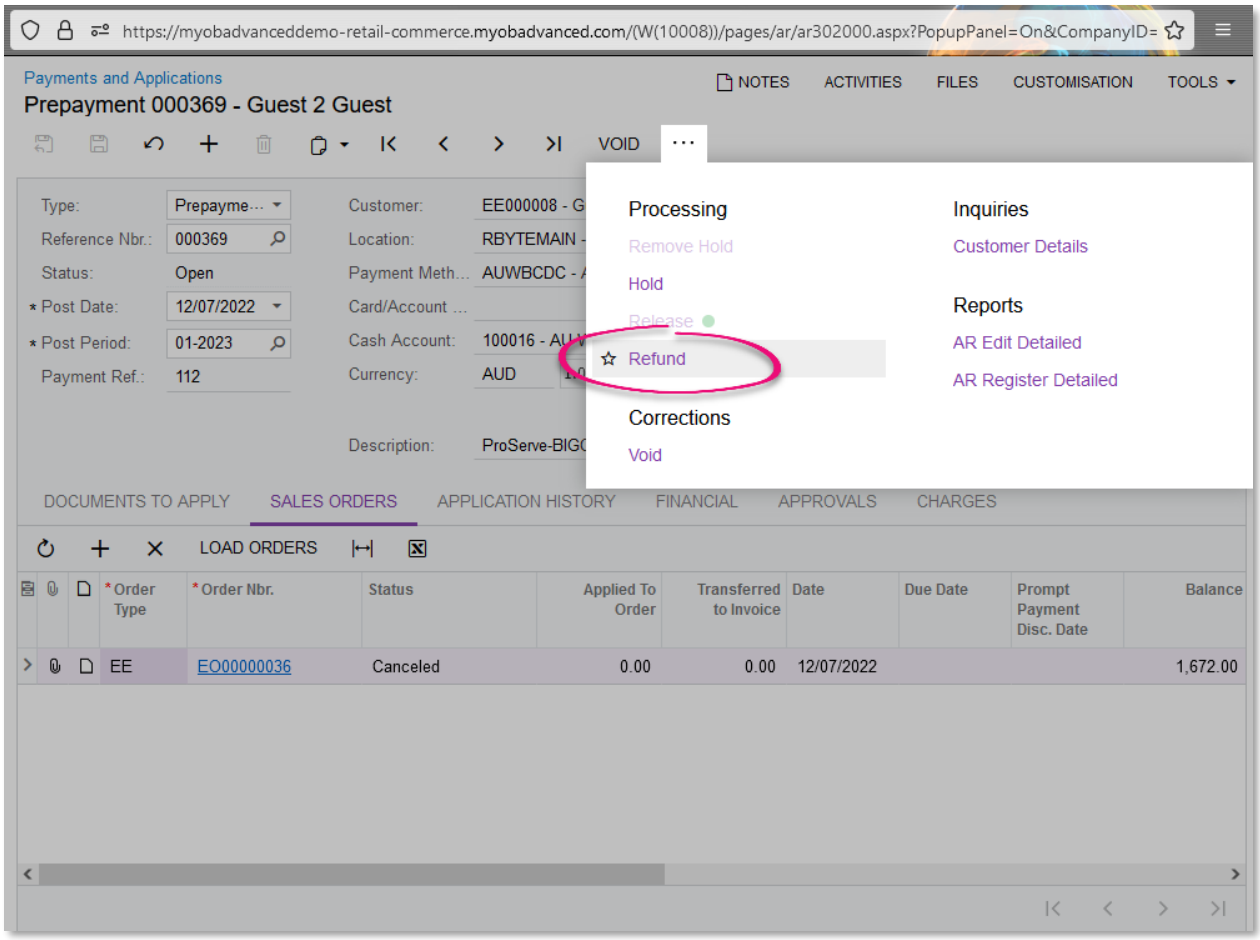
In MYOB Advanced, the Refund Entity of the BigCommerce store does not support Real-Time sync, therefore you need tor Prepare Data/Process Data for the Refund entity, either manually or through an Automation Schedule, to import the Refund from BigCommerce back to MYOB Advanced:



After the Refund has been imported, go back to the Sales Order, notice the Status of the Sales Order is now Cancelled:

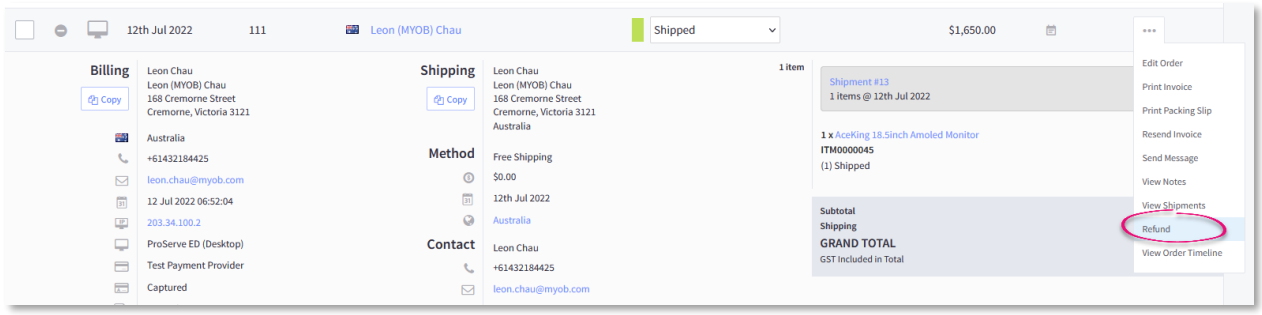


Open the related Prepayment, select Actions > Refund to create a Customer Refund AR Document. Release this Customer Refund to complete the Customer Refund:

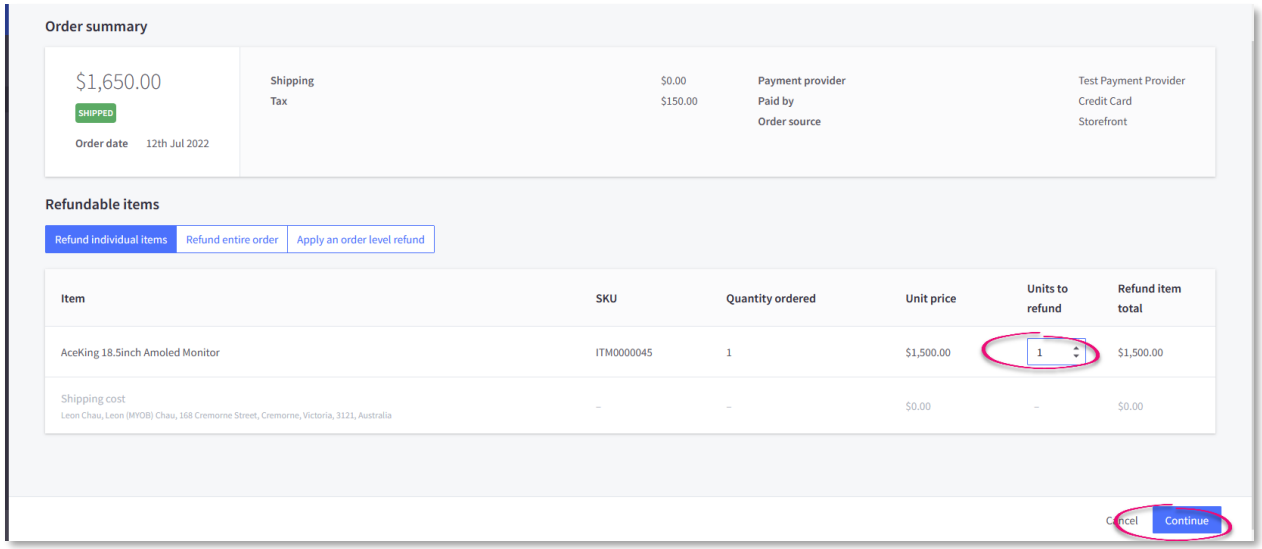


Processing Customer Refund (for fulfilled orders)

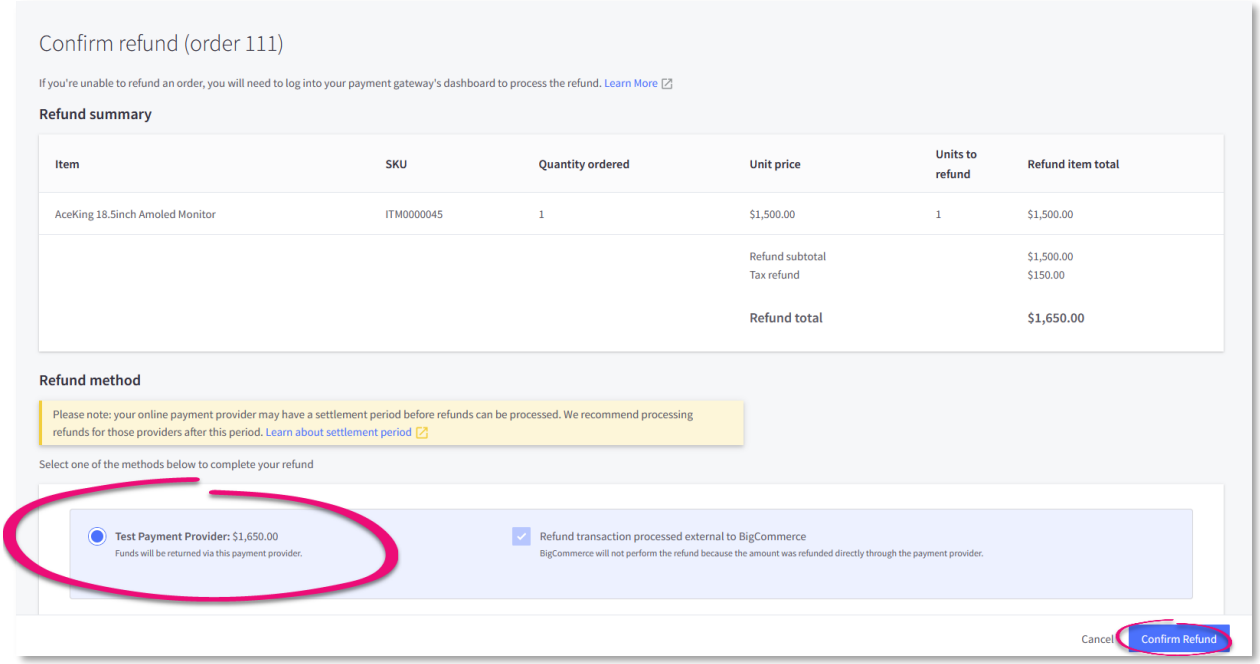
To create a refund for fulfilled order, select the order that has been shipped, and select Refund:



Specify the quantity to be refunded:

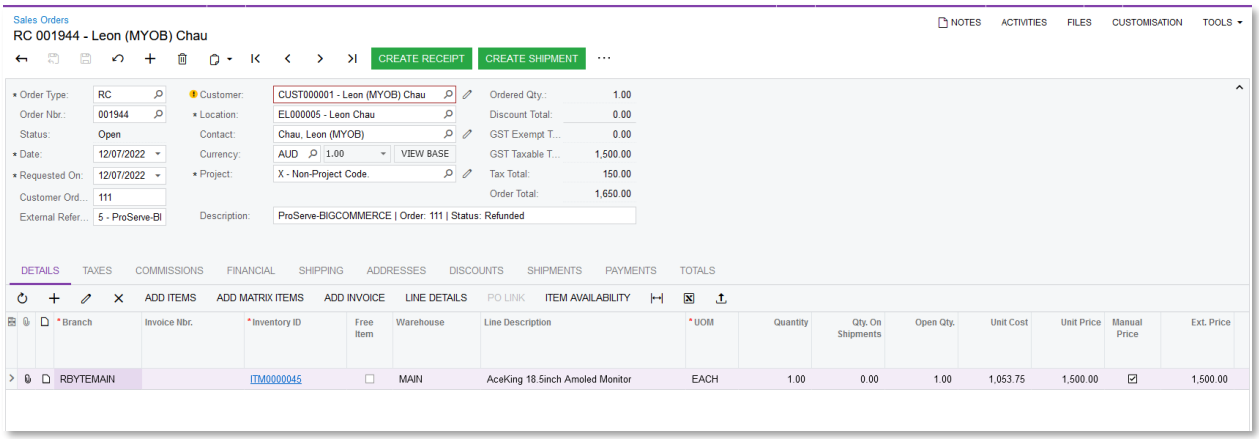


Specify the Payment Method to refund:

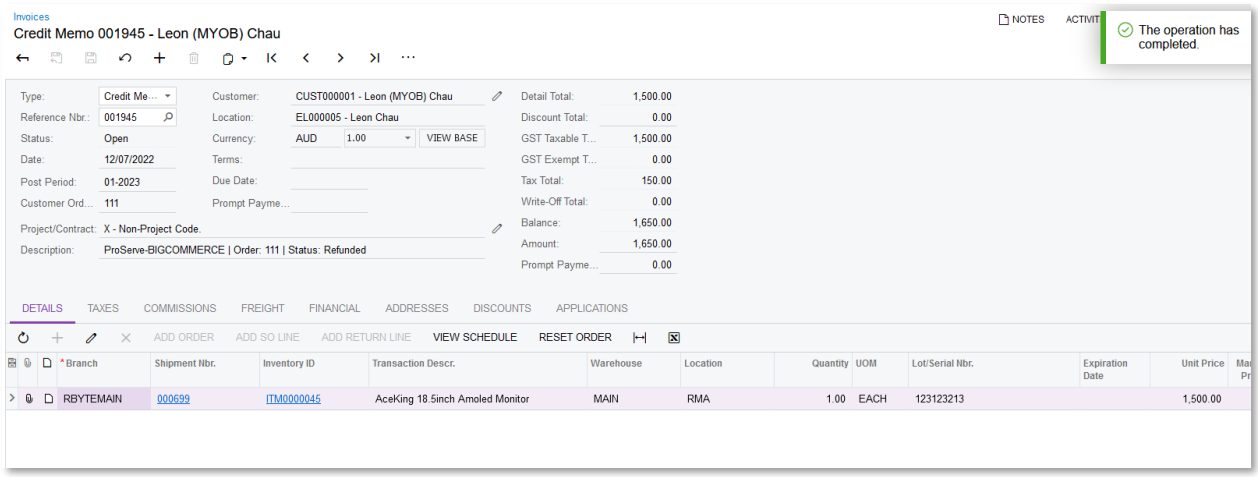


Prepare and Process Data for the Refund entity in MYOB Advanced, either manually or through an automation schedule. Once the Refund entity has been processed, go to the Sales Order list,

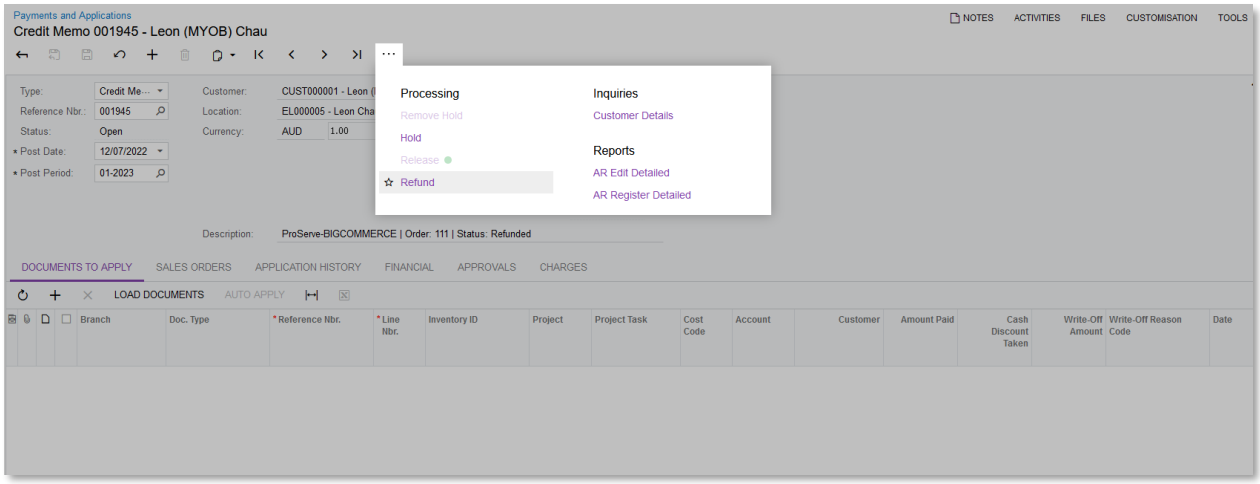
you'll notice a new RC order type has been created to process the refund for the customer – from here, select Create Receipt and confirm the resultant Shipment to bring the good back into Inventory:

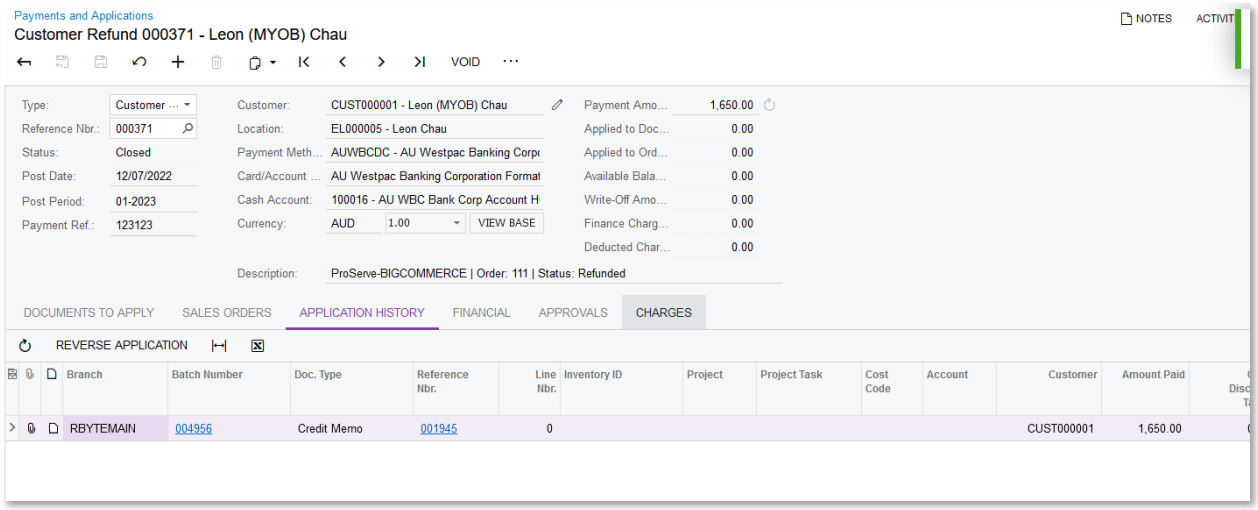


When you Prepare Invoice for the Shipment, the system generates a Credit Memo for the Refund amount:



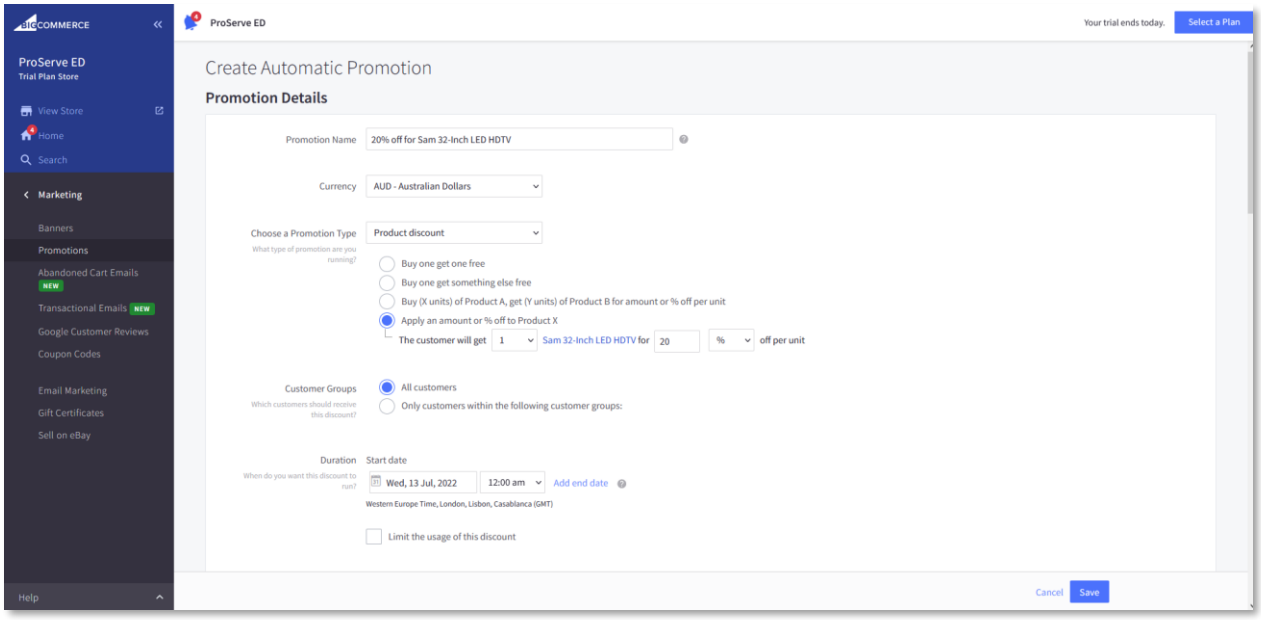
In the Payments and Applications screen AR302000, you can select the Credit Memo, and select Actions > Refund. Release the resultant Customer Refund to complete the Refund process:



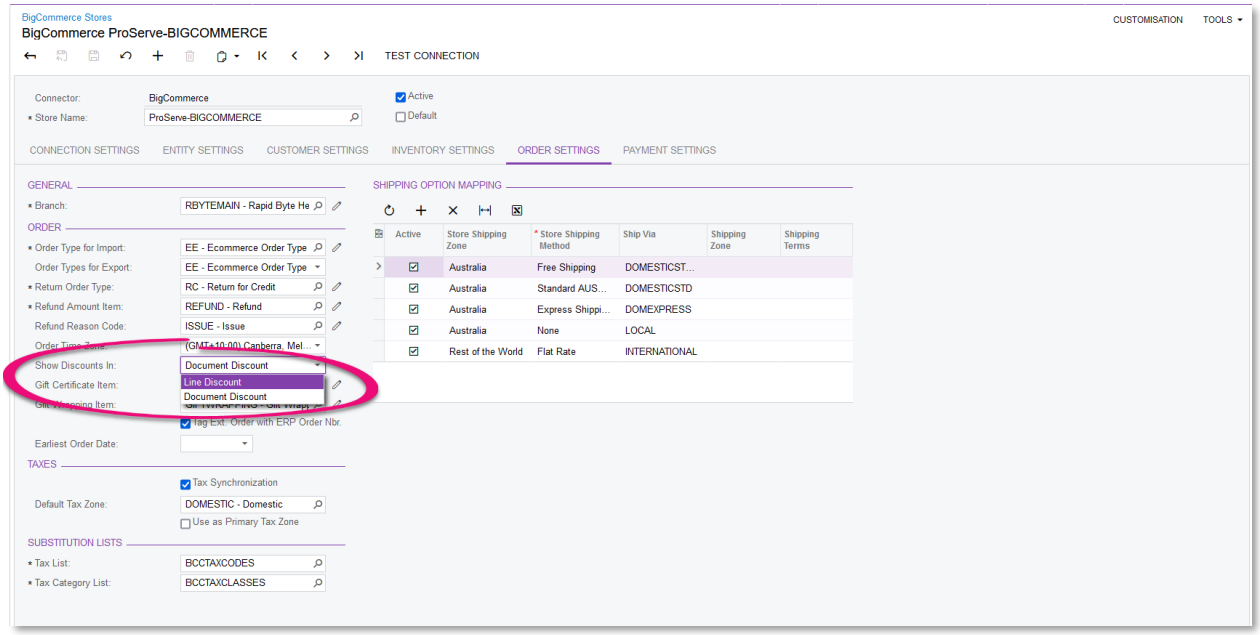


Discounts

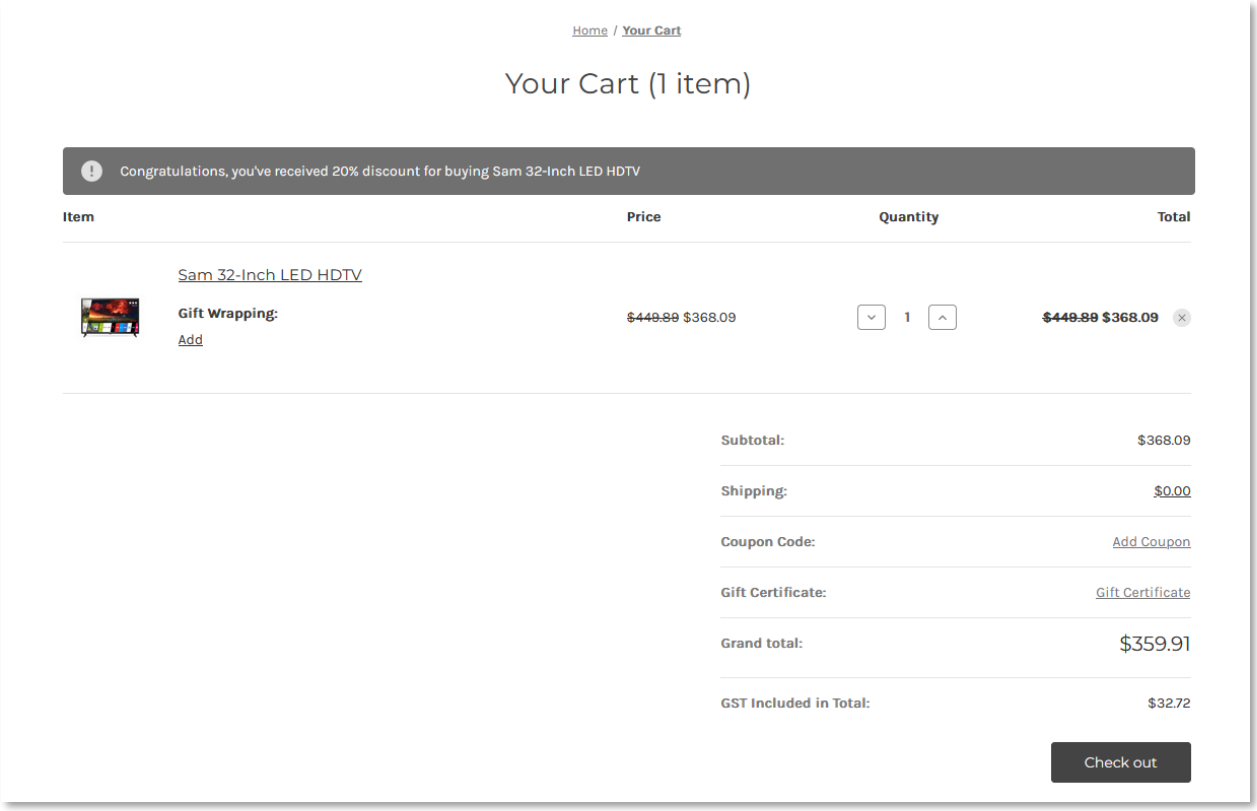
In BigCommerce, you can define different type of discounts, such as Shipping Discounts, Product Discounts, Order Discounts, etc. These discounts are set up in Marketing > Promotions.



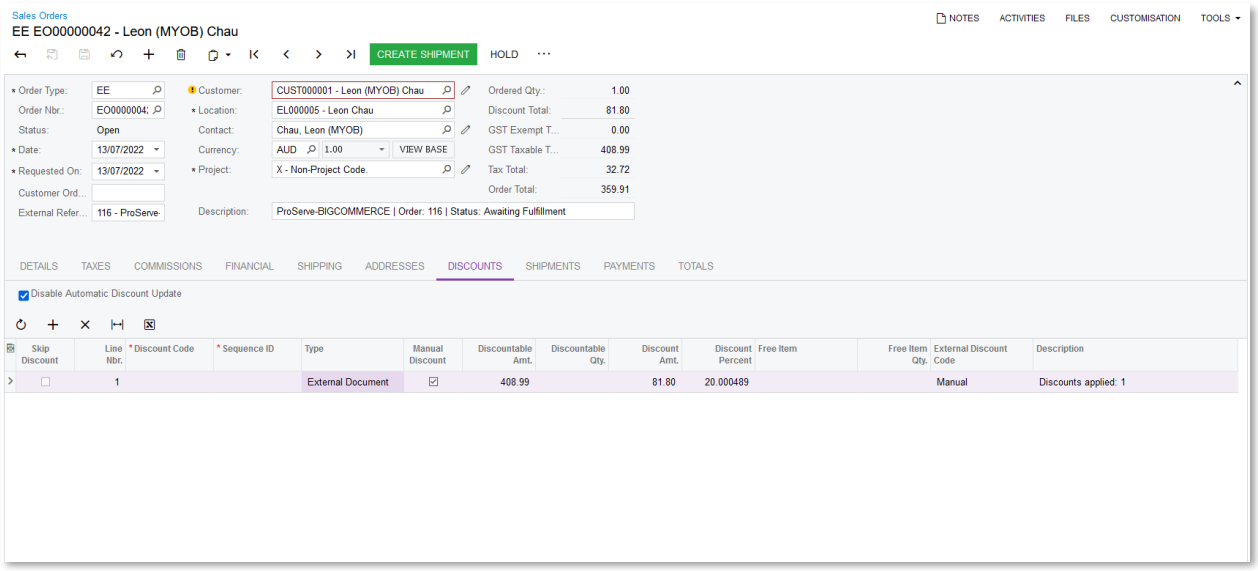
In the BigCommerce Stores screen, you can specify whether you would like the discount to show as a Line Discount or as Document Discount:



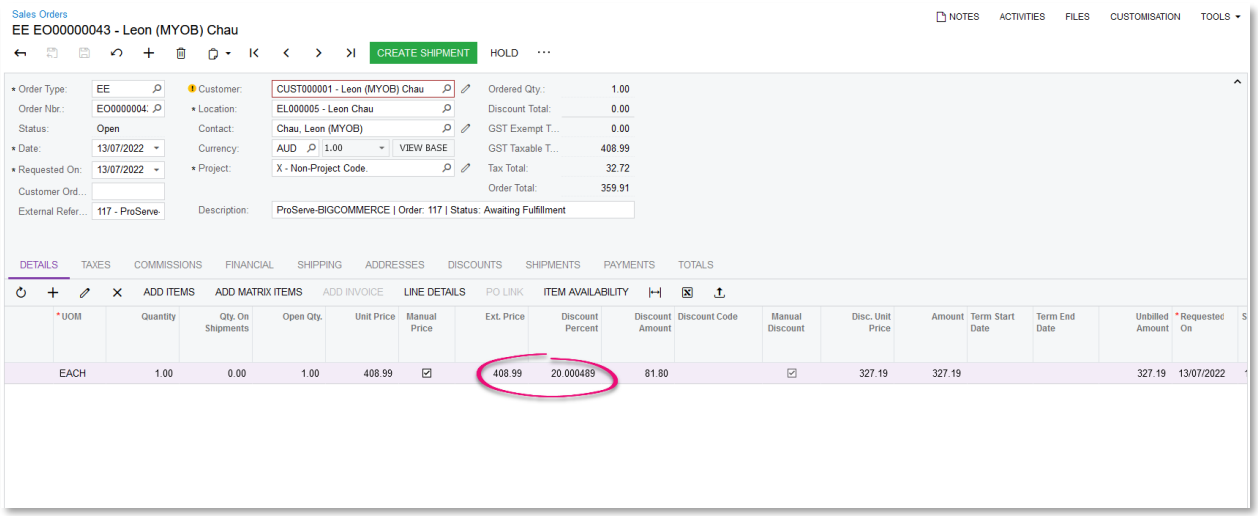
After the discount is configured, when the user checks out, they can see the discounted price for the item that they have added to cart:



If you select Document Discount on the BigCommerce store screen, when a Sales Order is synced to MYOB Advanced, the discount is shown in the Discount tab of the Sales Order:



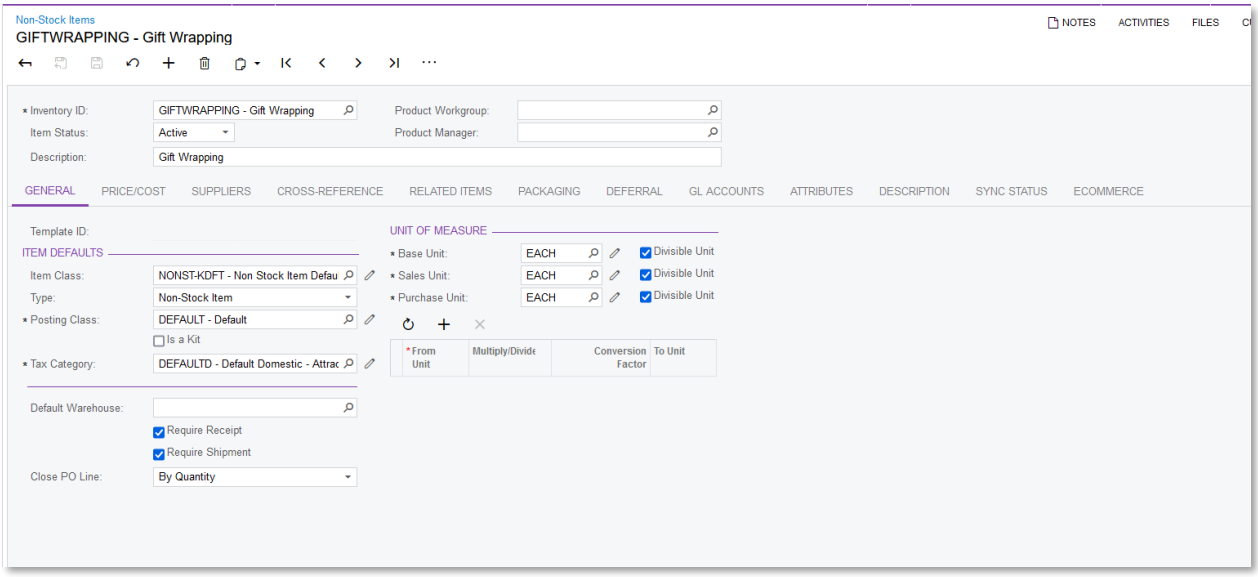
If you select Line Discount on the BigCommerce store screen, when a Sales Order is synced to MYOB Advanced, the discount is shown in the Details Line:



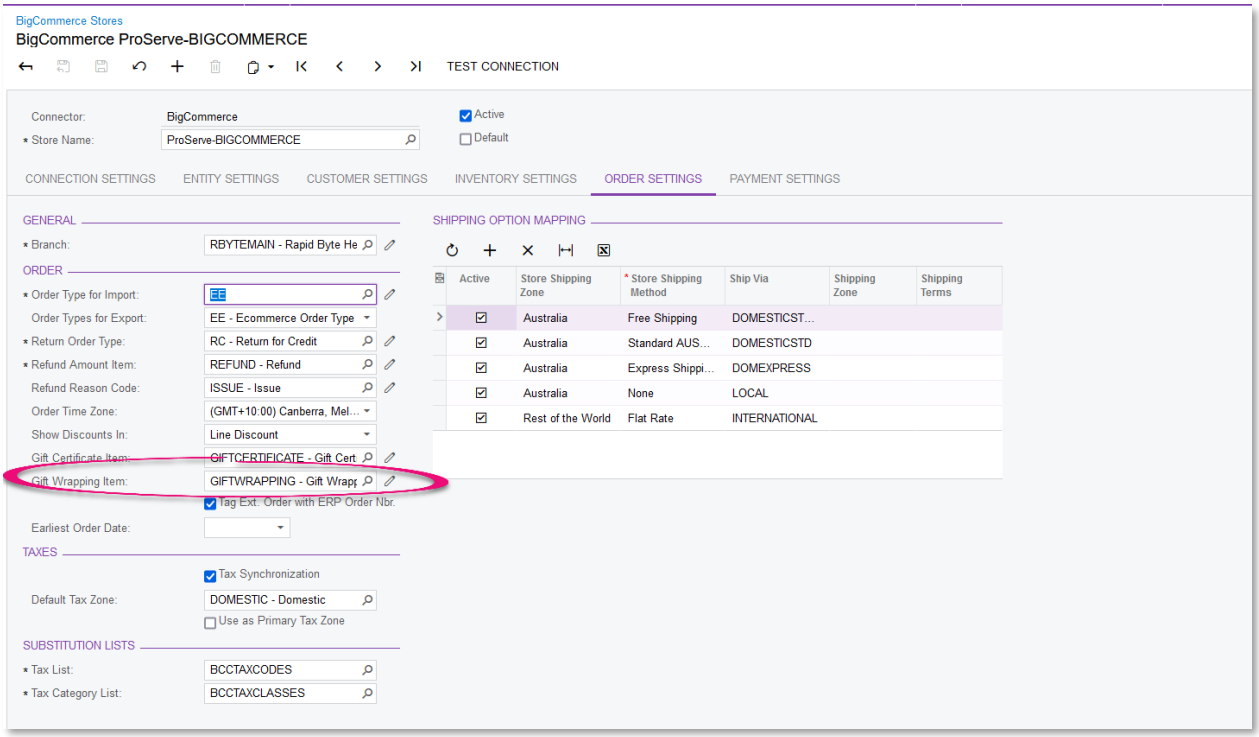
Gift Wrapping and Gift Message

Gift Wrapping and Gift Message are BigCommerce specific features. During the checkout, customers can decide whether they would like to wrap the purchased item as a gift and add personalized message. The gift wrapping service can be charge to the customer for an additional fee.

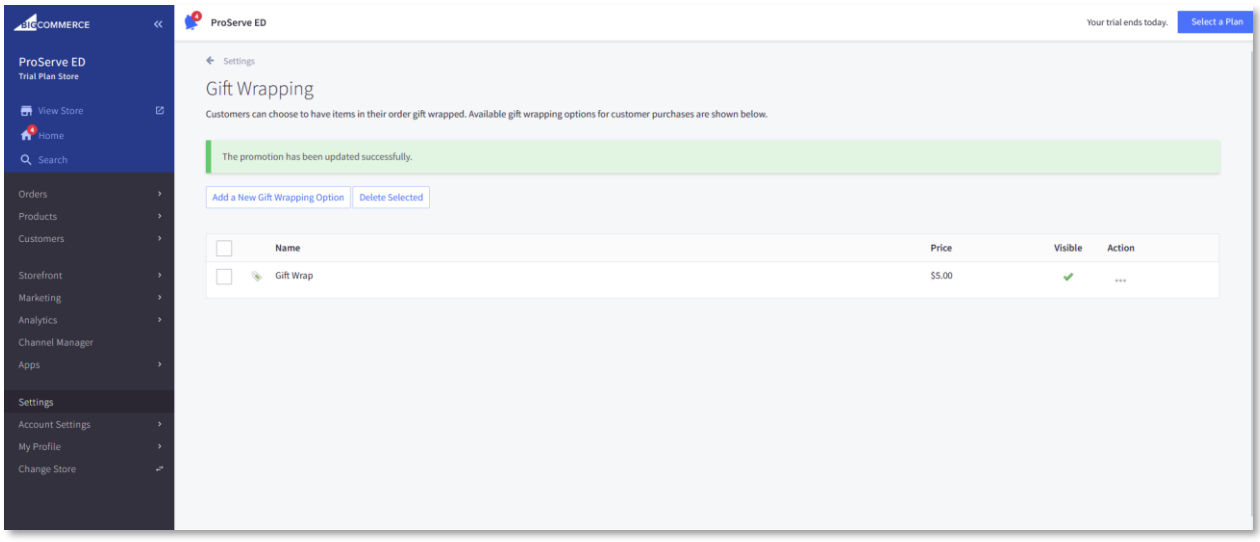
To configure Gift Wrapping, you need to first create a Non-Stock Item that represents the Gift Wrapping service. Make sure to tick Require Receipt/Require Shipment:



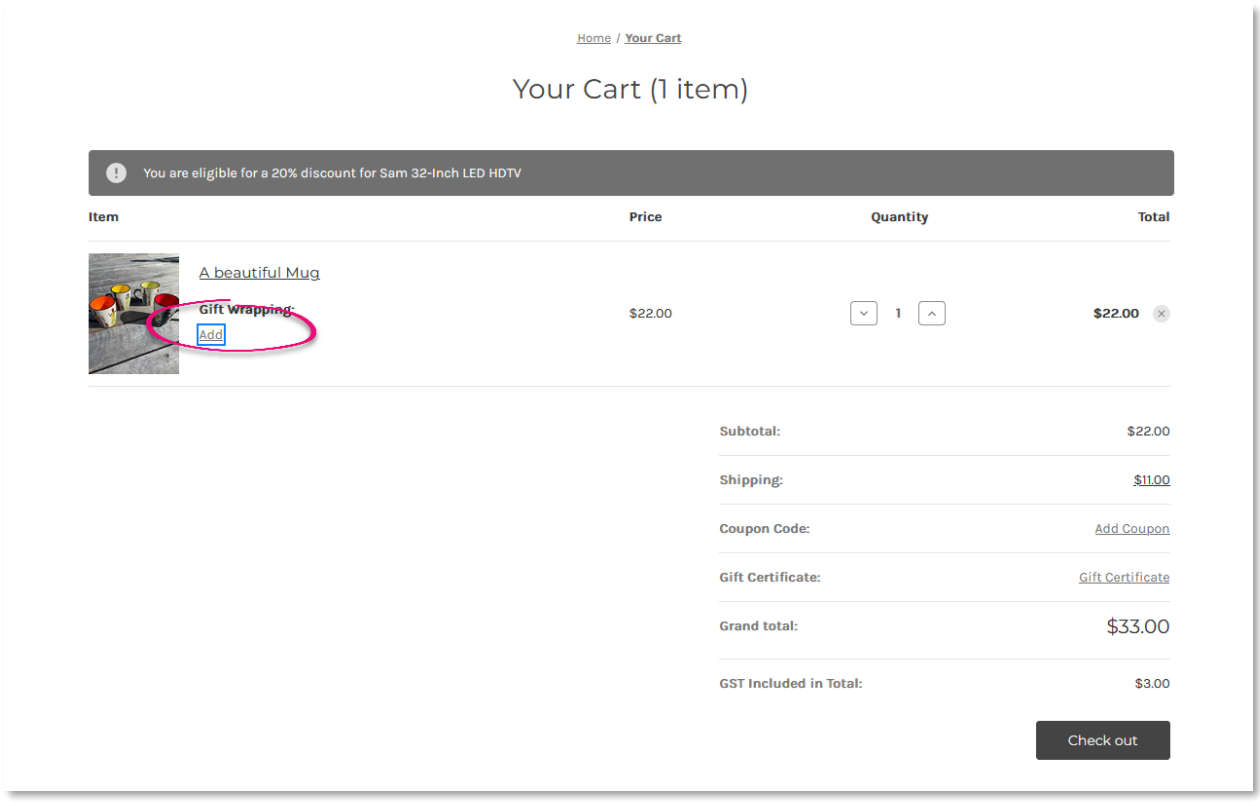
Then, you need to specify the Gift Wrapping non-stock item on the Order Settings tab of the BigComerce store:



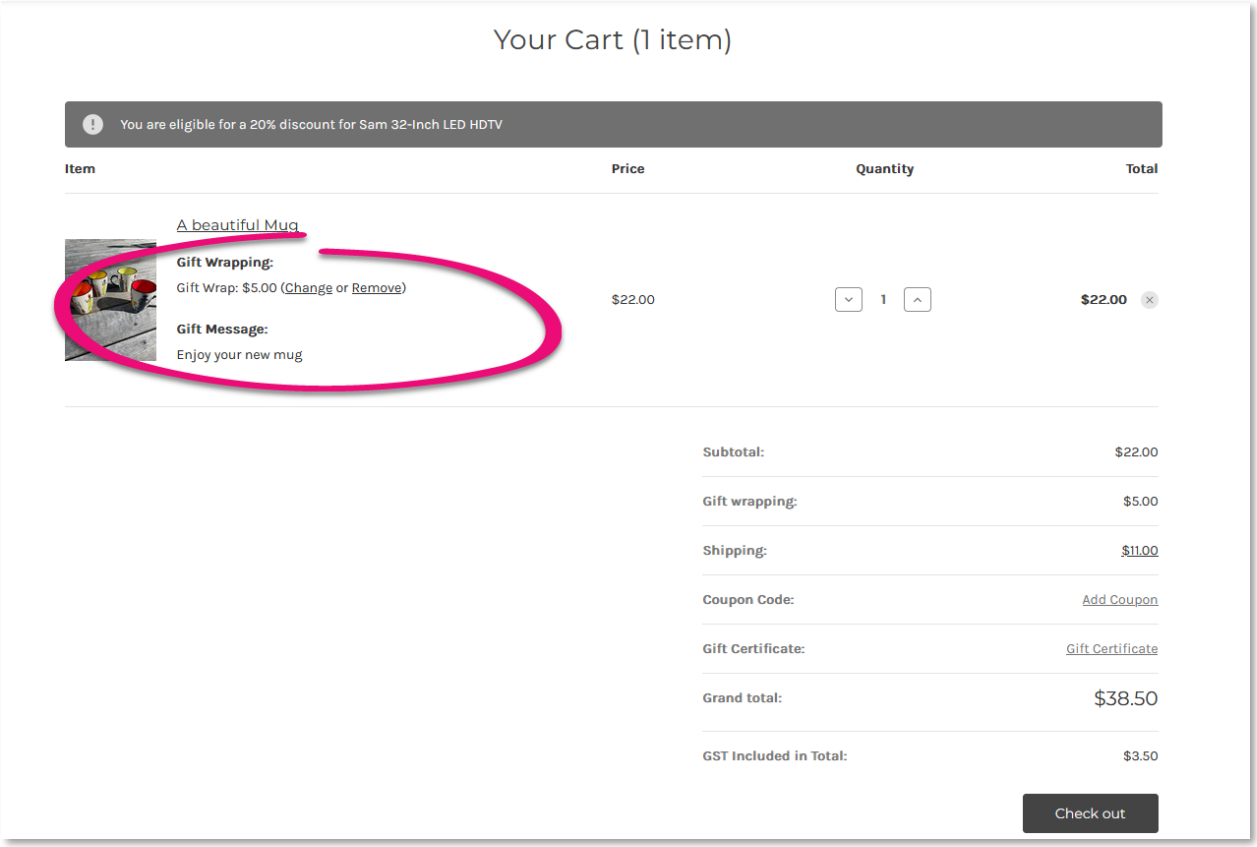
In BigCommerce, set up Gift Wrapping under Settings > Gift Wrapping – here you can specify the amount to be charged for the giftwrapping service:



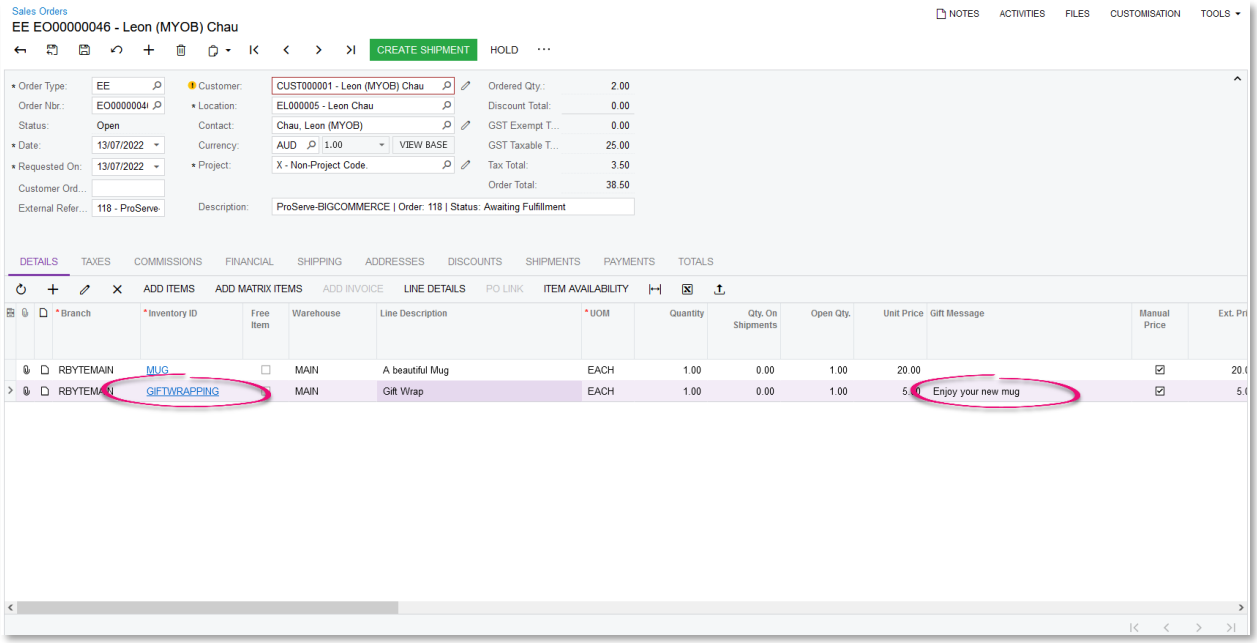
During checkout, customer has the option to Add giftwrapping service:



They can also add a personalized message if required:



When the Sales Order is synced to MYOB Advanced, the system inserts the Gift Wrap non-stock item into the Sales Order, and the Gift Wrap message is shown in the Gift Wrap message field:



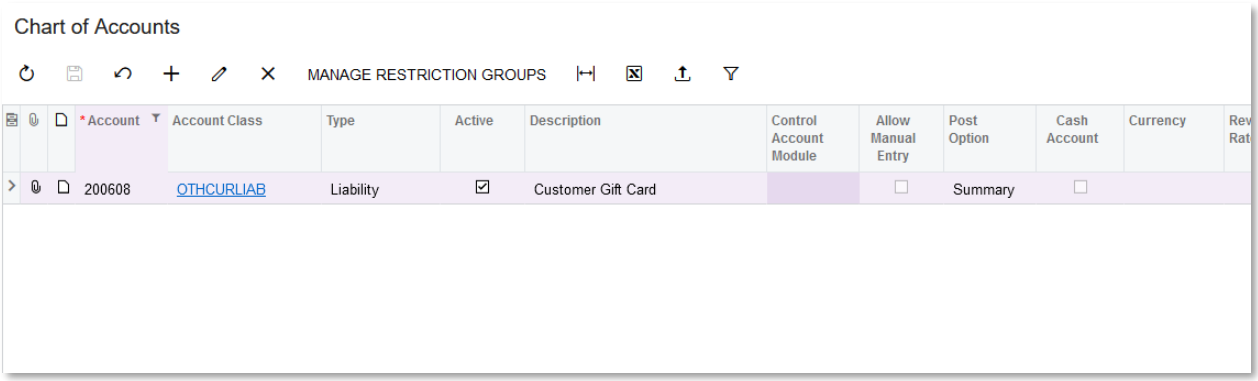
Gift Certificate/Gift Card

The Commerce Connector supports Gift Certificate both in BigCommerce and Shopify. In the storefront, customers can buy and pay for the Gift Certificate, and later use the Gift Certificate as

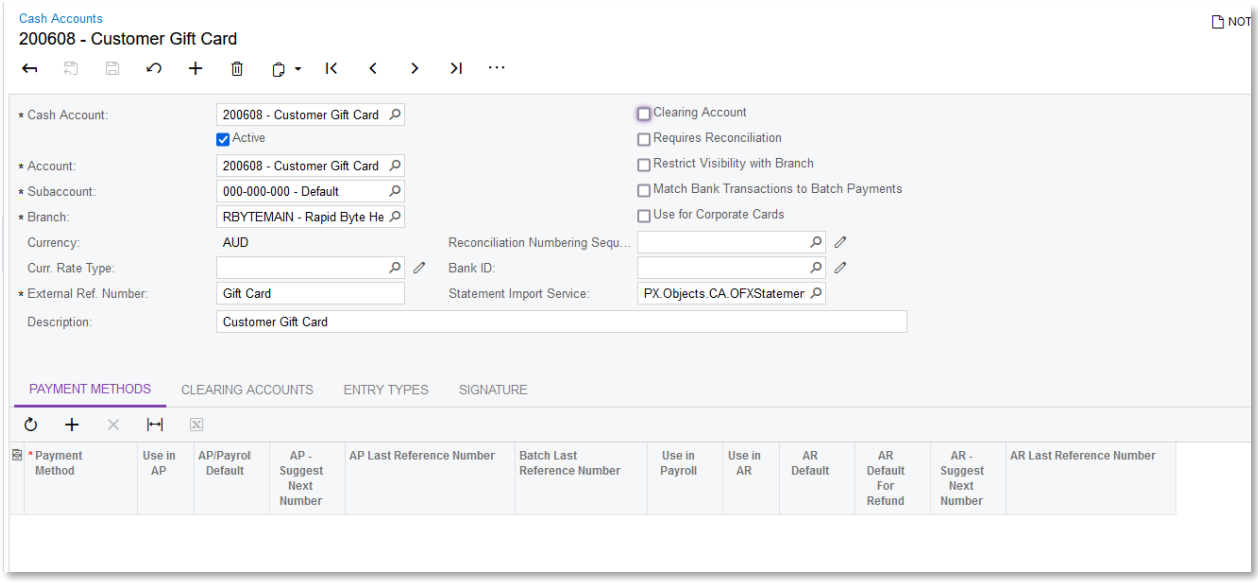
a payment method for their future purchases. The Gift Certificate is a configured as a non-stock item, linked to a Liability GL account to hold the customer's Gift Certificate balance, and also linked to a Cash Account and a Payment Method, that can be used to pay the Sales Order.

Configure Gift Certificate in MYOB Advanced

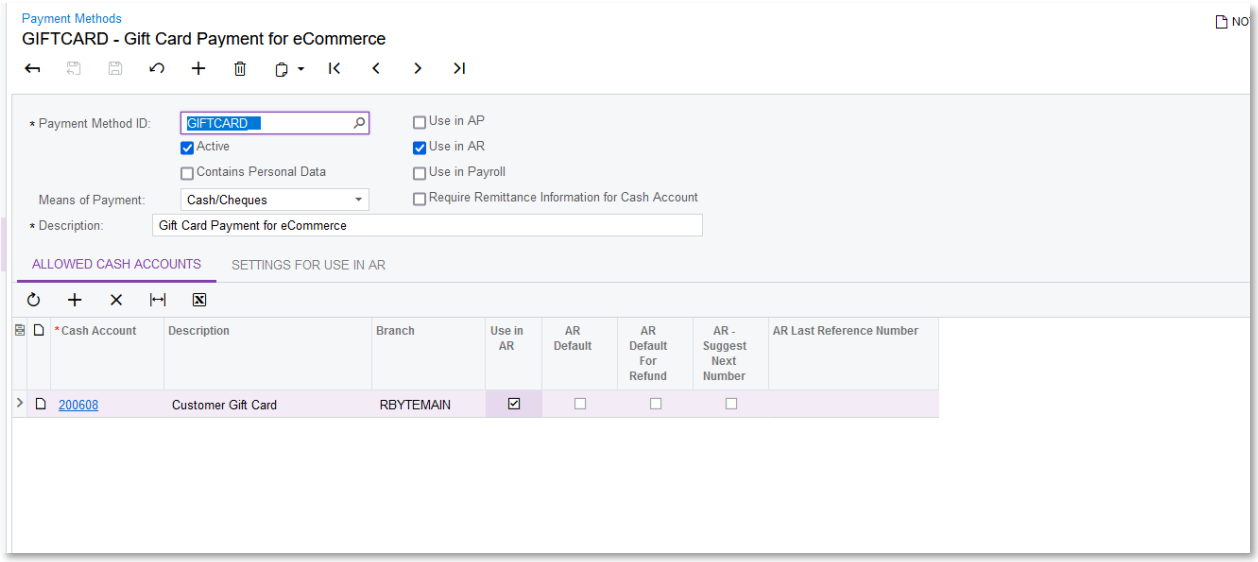
Firstly, we need to create a GL Account of type Liability to accrue the revenue of sold Gift Certificates:



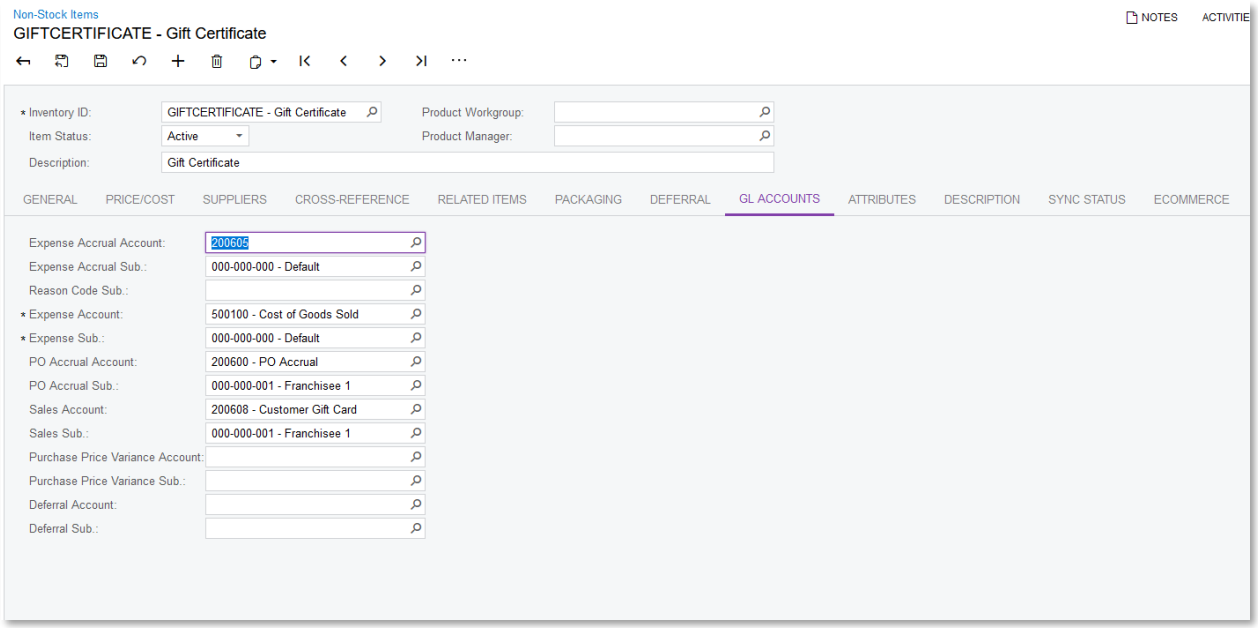
Then create a Cash Account, link to the GL Account that we just created to represent payment by Gift Certificate:



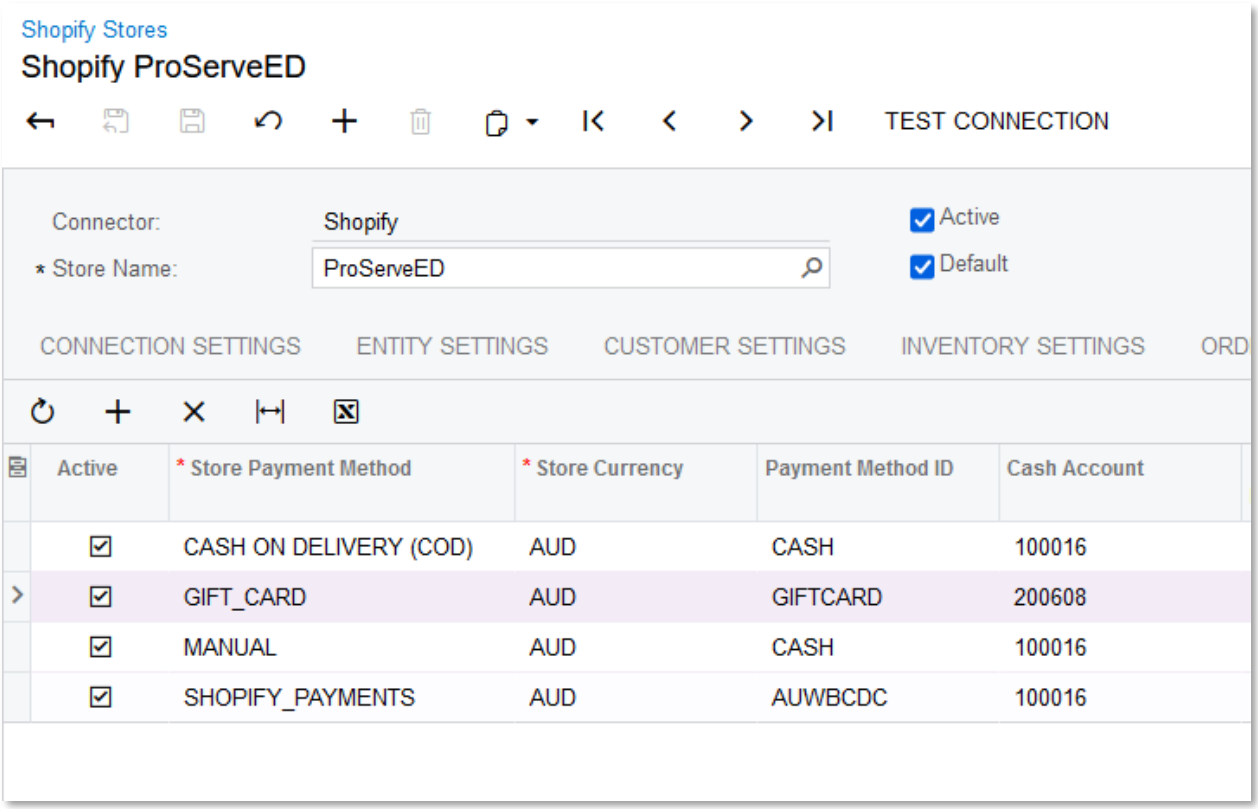
Next, we need to create a Payment Method linked to the above Cash Account:



We also need to create a Non-Stock Item to represent the Gift Certificate purchased through the storefront. When the customer purchase the gift card through Shopify or BigCommerce, a Sales Order will be created, this Non-Stock Item Gift Certificate will be added to the Sales Order. Make sure that Sales GL Account of this non-stock item is set to the liability account that we have configured above, so that the customer gift card balance is charged to that liability account.

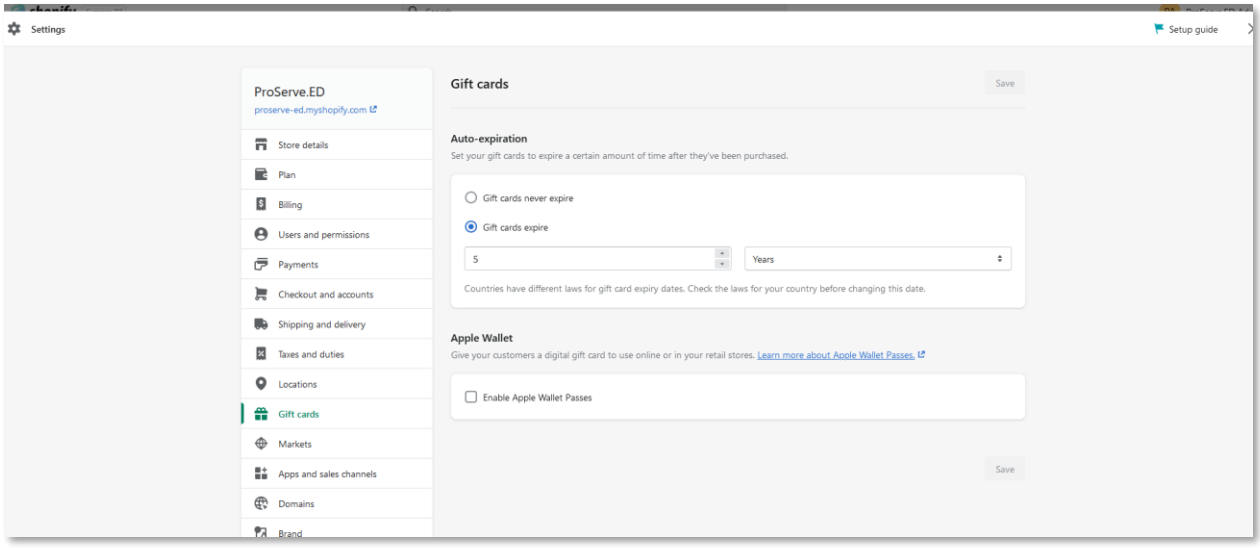


Finally, on the Shopify Store/BigCommerce Store, link the store's gift card payment method to the above Payment Method and Cash Account:

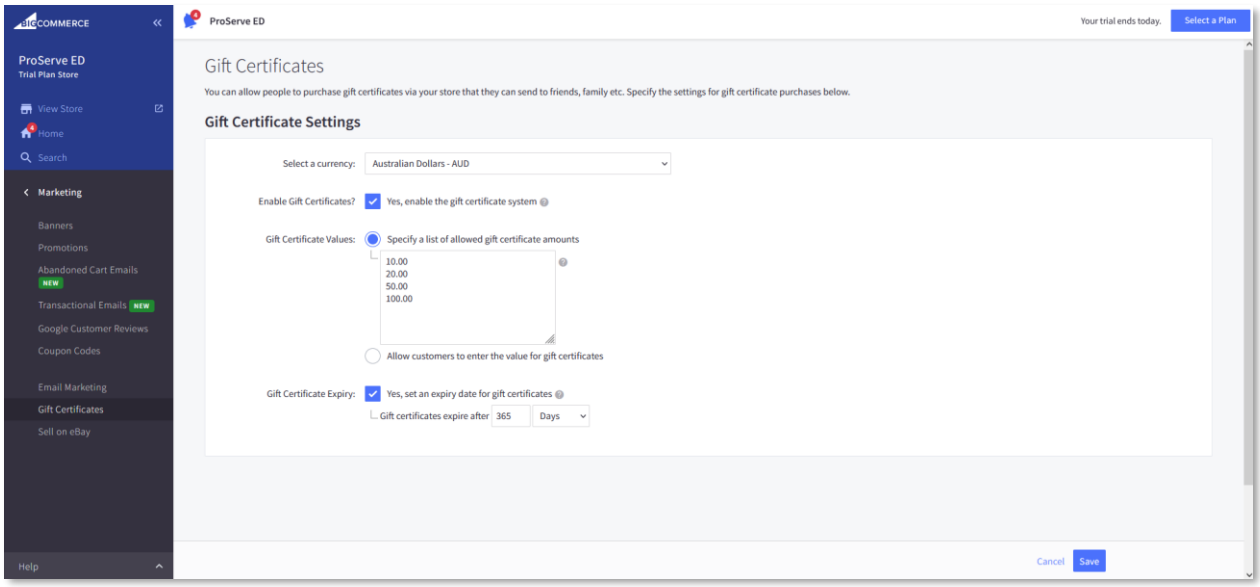


Enable Gift Certificate/Gift Card feature in the stores

Before you can allow customer to purchase and use gift card, you need to enable the Gift Card feature in the stores. In Shopify, this is done through Store Settings > Gift Card:

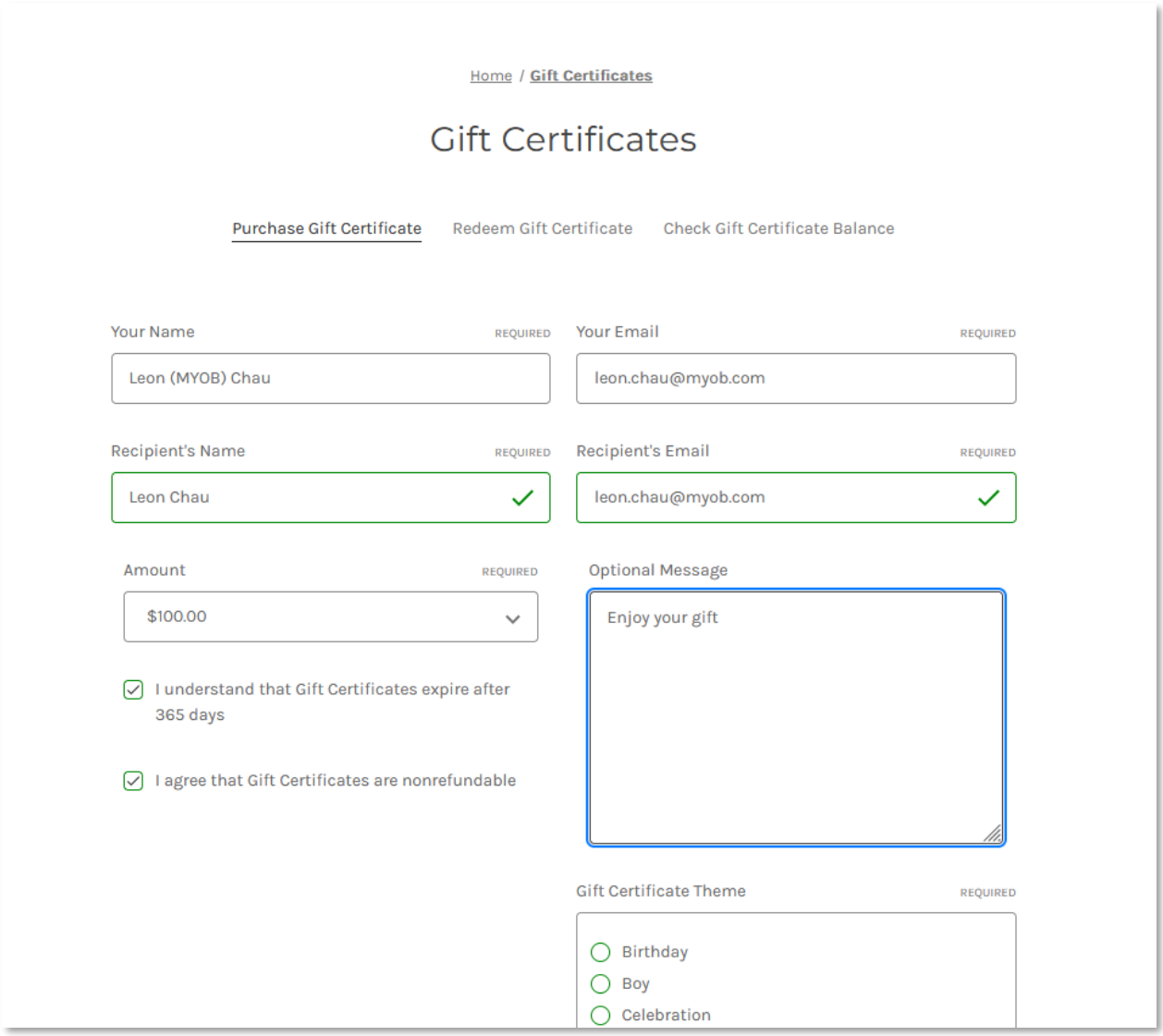


In BigCommerce, this is done through Marketing > Gift Certificates:

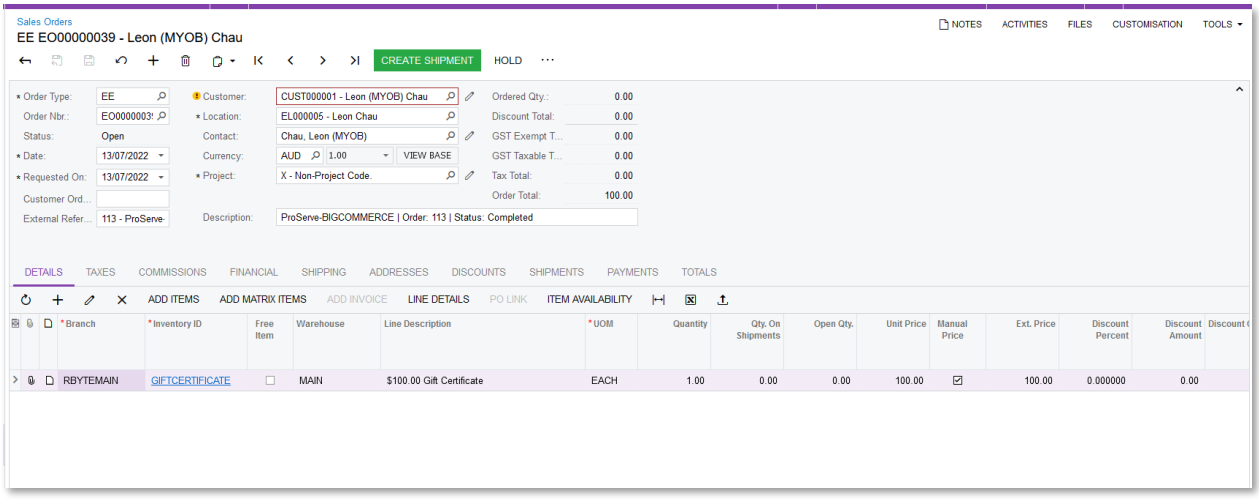


Buying a Gift Certificate/Gift Card

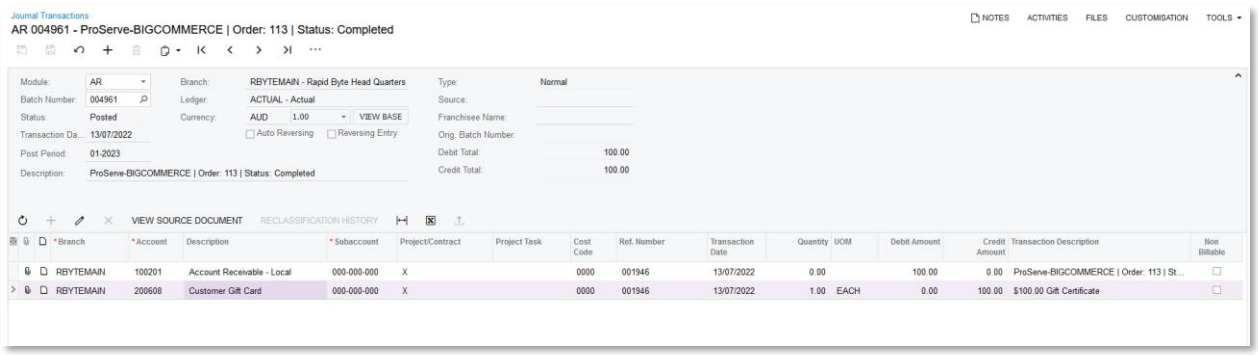
From the storefront, the customer can purchase a Gift Certificate just like any other products:



The Gift Certificate is synced to MYOB Advanced as a Sales Order, containing the Gift Certificate non-stock item, with the unit price equals to the value of the gift card:

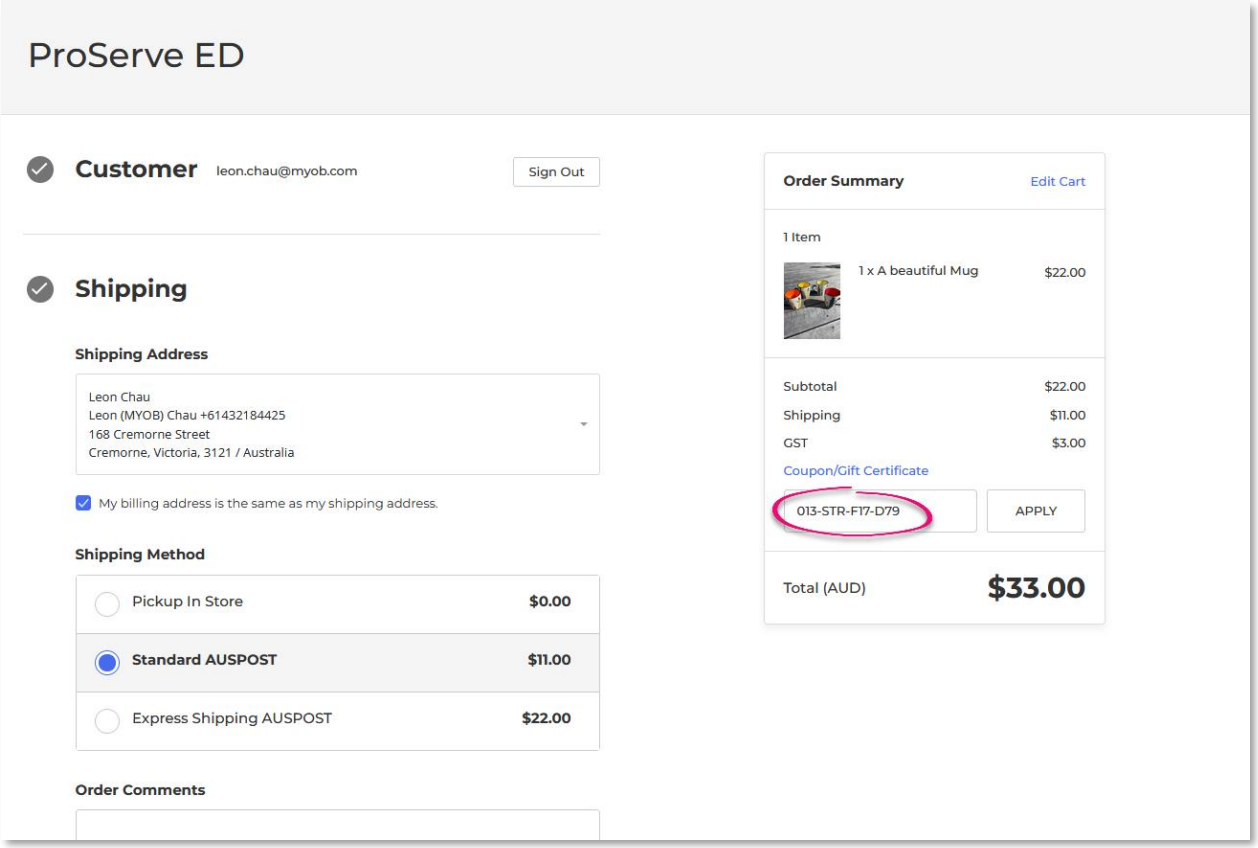


When the AR Invoice is released, the Balance of the Gift Card is charged to the Liability account:

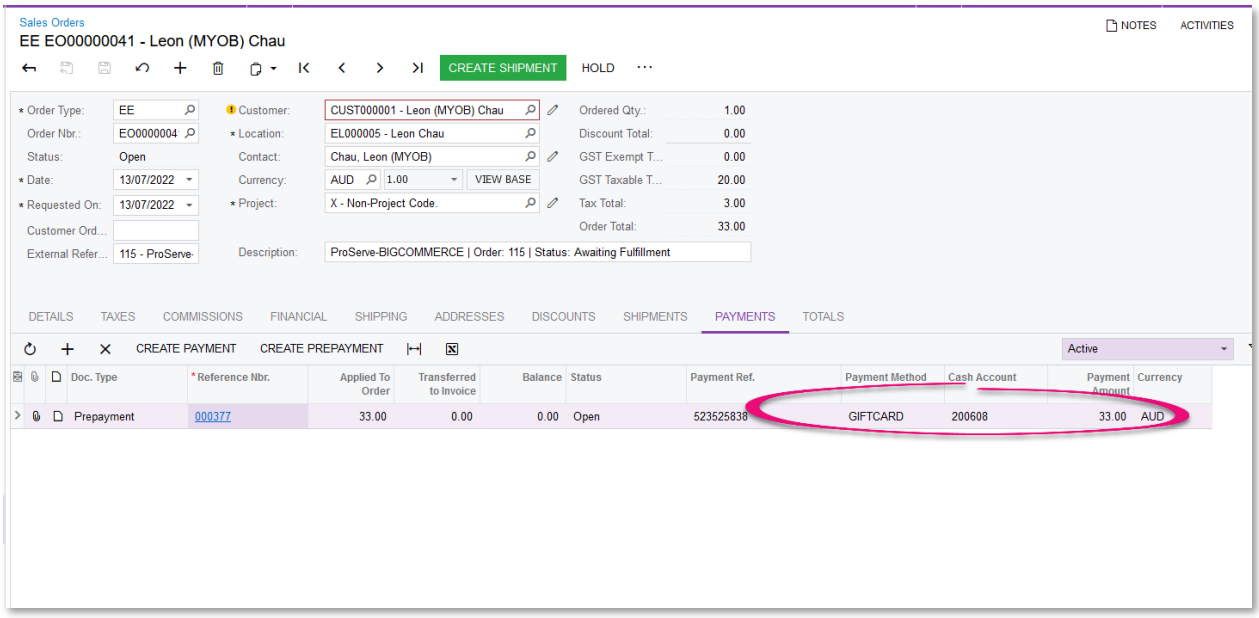


Redeeming a Gift Certificate/Gift Card

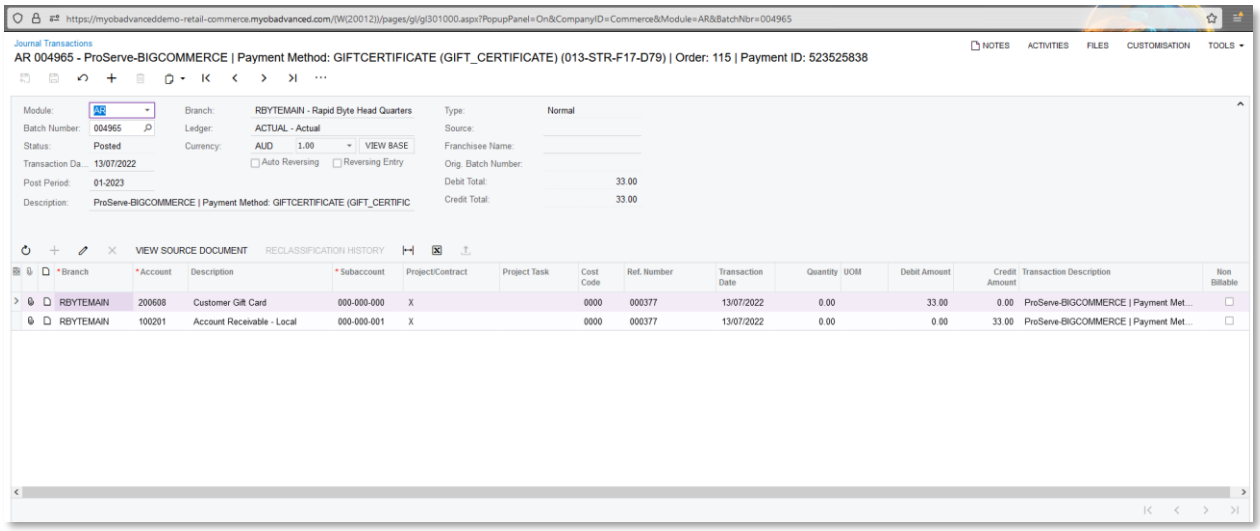
During checkout, the customer can apply gift card to redeem it:



When the Sales Order is synced to MYOB Advanced, you can see that the system picks up the Gift Card Payment Method and use that for the Prepayment:



This reduces the Gift Card liability in the GL Account that we use to hold the Gift Card balance:



Customer Pricing, Bulk Pricing, and Pricing Limitation (BigCommerce Only)

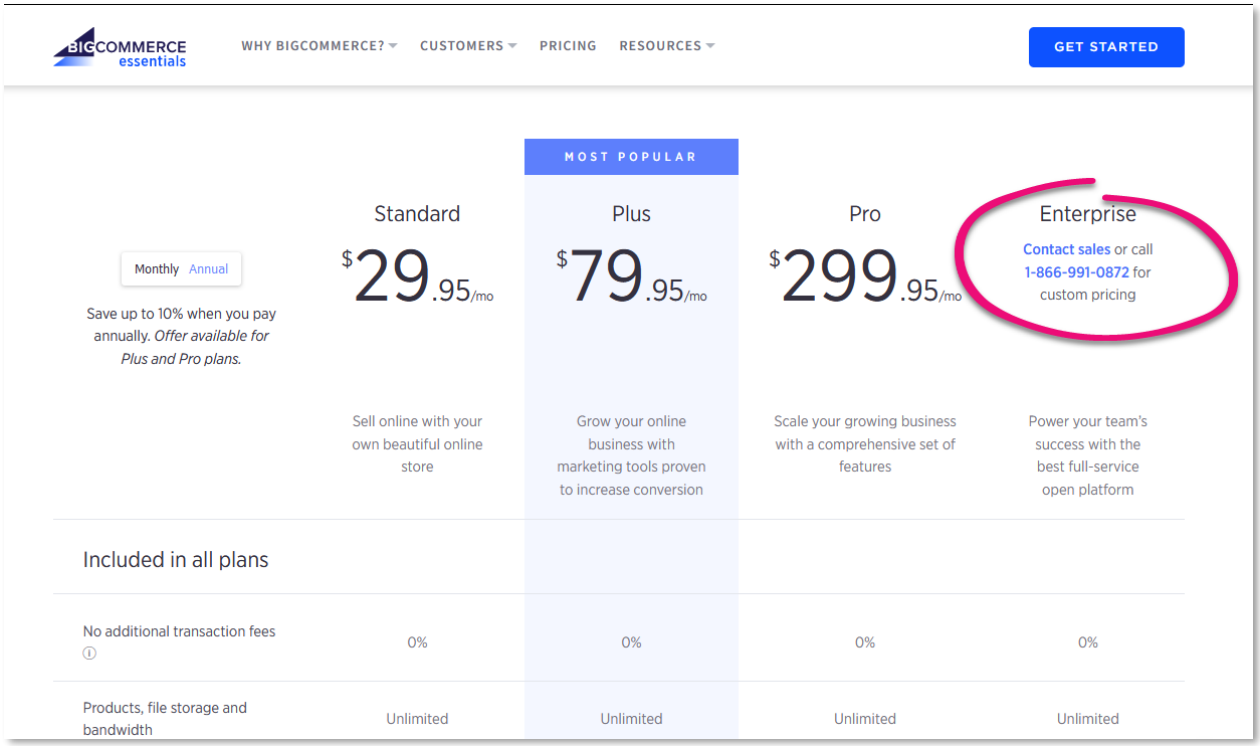
Customer Pricing

BigCommerce allows users to create different customer groups, and therefore support customer specific pricing.

In order for the store's customer to receive customer pricing, the following condition must be met:

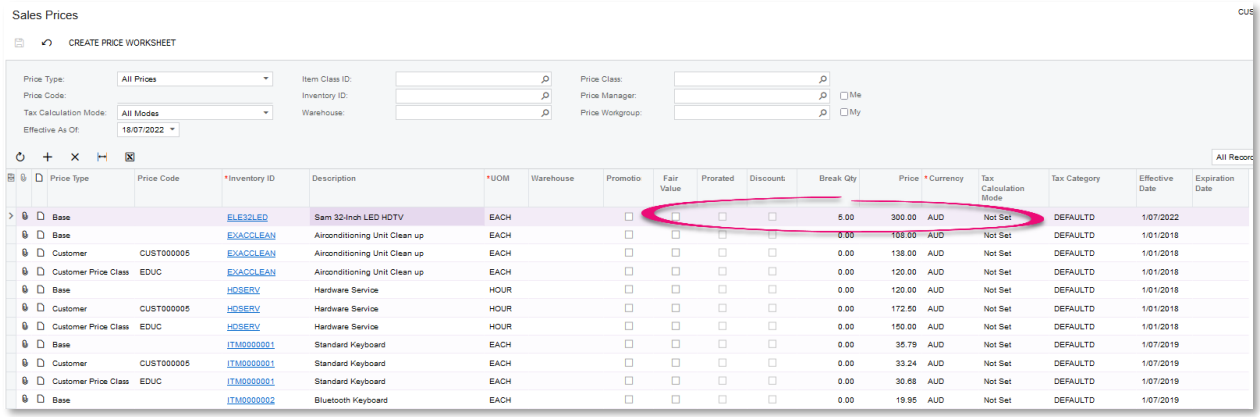
- The customer record must have been synced in MYOB Advanced
- The customer must have a customer price group in MYOB Advanced
- Customer specific pricing must have been set up in MYOB Advanced (through the Sales Price screen or through the Sales Price worksheet)
- The customer must log in at the storefront to see the customer specific price.

- The storefront must be on the BigCommerce's Enterprise subscription plan.

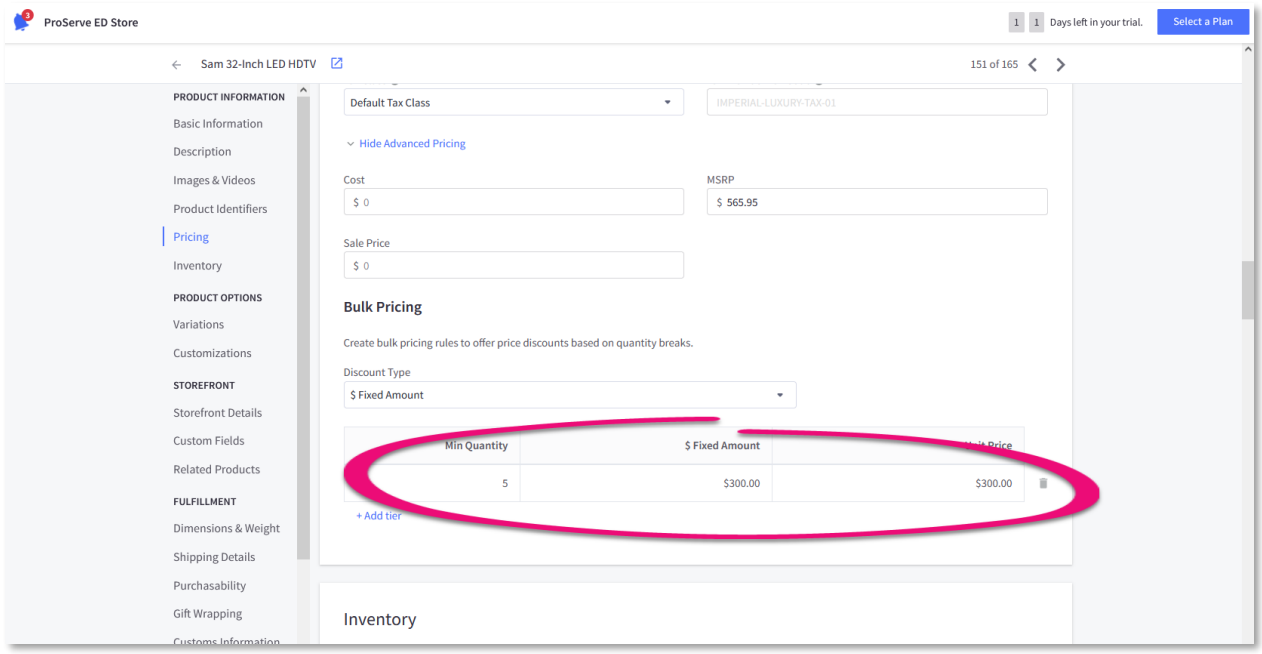


Bulk Pricing

If the site is using the Volume Pricing in MYOB Advanced to configure and use Bulk Pricing, these prices can also be exported to BigCommerce. For example, if you have a Sales Price with a Break quantity defined on the Sales Price screen (AR202000),



When the Base Sales Price entity is synced, the system will insert a bulk pricing line for that particular product.



Pricing Limitation

The following price types maintained in MYOB Advanced cannot be exported to a BigCommerce store:

- Customer-specific prices: Prices defined in MYOB Advanced for a particular customer—that is, prices that have the Customer type—cannot be exported to the BigCommerce store. If you want to provide a particular customer with the ability to purchase products at individual prices, create a separate customer price class for this customer. This way, the prices that you define for this customer price class will be available for only the customer that you add to this customer price class.
- UOM-specific prices: In MYOB Advanced, if the Multiple Units of Measure feature is enabled on the Enable/Disable Features (CS100000) form, you can define a sales price of an item for each of the units of measure in which your company sells this item. However, only prices defined for the base UOM can be exported to BigCommerce.
- Warehouse-specific prices: If the Multiple Warehouses feature is enabled on the Enable/Disable Features form, prices that are defined to apply only when products are sold from a particular warehouse—that is, when a specific warehouse ID is selected in the Warehouse column for a particular price—are not exported to the BigCommerce store.

Only prices that are effective on the date of the synchronisation are exported; historical prices and prices with an effective date in the future are not synchronised. If an item has multiple prices of the same type that are effective on the date of the synchronisation, the price of the highest priority is imported. For example, if a regular base price of $5 and a promotional base price of $4.50 for the same item are effective on January 31, 2021, which is the date when you synchronise the Base Sales Price entity, the promotional price of $4.50 will be exported to the BigCommerce store.

Template items, matrix items and product variants

Both Shopify and BigCommerce support the selling of Products with different variants. Because the Template Item entity can only be exported from MYOB Advanced to the stores, you must

configure the appropriate attributes, item classes, template items and matrix items in MYOB Advanced, and then export these to the stores.

The Template Item entity is separated from the Stock Item entity, therefore you should activate the Template Item entity and configure syncing or real-time sync as required. When you sync the Template Item entity, the Template Item itself will be a product of the store, while its Matrix Items become the product variants.

When implementing the Template Items and Matrix Items feature for a store, if the site has a large number of product variant images, this can lengthen the implementation timeline, because users will need to manually drag and drop product images on the Stock Items screen, Attributes tab for each variant. If images are added for a matrix item as links in the Media URLs table on the eCommerce tab of the Stock Items (if the matrix item is a stock item) or Non-Stock Items form (if the matrix item is a non-stock item), these images are not exported to BigCommerce or Shopify.

Implementation Efforts Estimation

The information below provides an estimate on the time and effort to implement the eCommerce Connector. The connector itself requires little configuration to be up and running, however, most of the solution relies heavily on other modules of MYOB Advanced, and their implementation efforts need to be considered separately.

Item	Effort Estimate	Note
Implementation of Finance: GL, AP, AR, reports, etc.	N/A	Needs to be estimated separately, varies based on requirements
Implementation of Inventory: Item Class, Stock Items, Non-Stock Items, Kits, Template Item, Matrix Items, Item Sales Categories	N/A	Needs to be estimated separately, varies based on requirements
Implementation of Orders: SO, PO, Transfers, etc.	N/A	Needs to be estimated separately, varies based on requirements
Basic eCommerce store setup: Payment Methods, Shipping, Taxes, Store Info, etc.	3 hours per store	
Configure BigCommerce/Shopify Store screen in MYOB Advanced	1.5 hour per store	
Configure Field Mappings to support the synchronisation	From 0 to minimum 2 hours per entity	Field Mappings are not required – by default, the Commerce Connector exports/imports data into a set of standard fields. If the customer has a requirement to import/export into custom fields, change the mapping of standard field, or require

		writing of formulas to accommodate specific scenarios, then the effort will change and need to be re-estimated accordingly
Establish Store Connections	1 hour per store	
Perform first synchronisation and resolve error messages	Minimum 3 hours	
Automate Synchronisation and configure Real-Time Sync	2 hours per store	
Configure Gift Certificate/Gift Cards	2 hours per store	
Configure Customer Pricing, Bulk Pricing	2 hours per store	