

GD: Wholesale Pro - How to use

Overview:

GD Wholesale Pro enables a scalable B2B portal that works within your existing store. Capture qualified buyers through custom registration forms and approve B2B customers with a single click. Apply exclusive pricing for approved accounts while keeping retail traffic fully isolated. Manage everything from one place - edit B2B prices directly in the app or use our powerful Excel Import/Export engine to update thousands of variant prices instantly. Grow your wholesale B2B channel without writing lines of code.

Navigation Overview:

The application currently includes the following navigation sections:

- Custom Forms
- B2B Product Price
- Import Product Prices
- Export Product Prices
- How To Use

So now, let me explain how this navigation routes will help you to create a B2B environment.

Application Usage Flow & How It Works:

First Step: Integration of B2B Custom Form Block

To create a B2B environment in your store, start by creating a custom form from the Custom Forms section. This form collects customer data, which you can review to approve or reject customers as B2B. Only approved customers will be able to view and access B2B pricing.

First, create a B2B sign-up form of your choice, selecting the customer details you want to collect, as shown in the screenshot below.

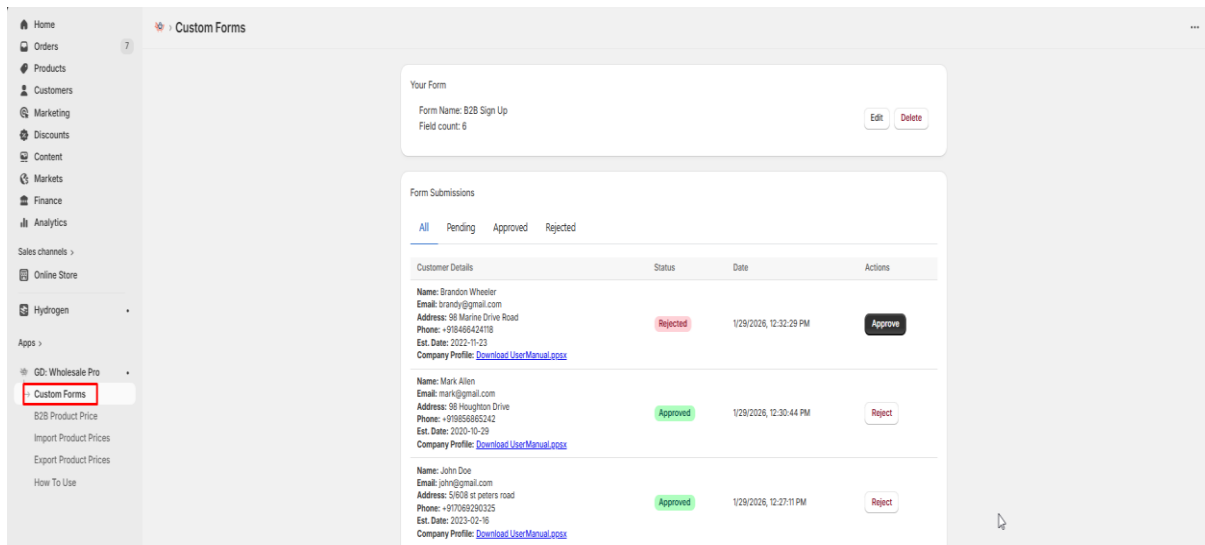
The screenshot displays the 'Edit Form' interface for GD Wholesale Pro. On the left, a sidebar lists navigation options, with 'GD: Wholesale Pro' expanded and 'Custom Forms' selected. The main content area is titled 'Edit Form' and includes a 'Form Integration' section with a text box for a form ID. Below this is the 'Form Settings' section, which contains fields for 'Form Title' (set to 'B2B Sign Up'), 'Content' (set to 'Complete the form to sign up.'), 'Submit Button Text' (set to 'Submit'), 'Success Message' (set to 'Thank you for contacting us!'), and a 'Consent Disclaimer' text area. To the right of the settings is a 'Toolbox' containing various form elements like 'Single Line Text', 'Paragraph Text', 'Email', 'Phone', 'Number', 'Date / Birthday', 'Dropdown', 'Radio Buttons', 'Checkbox', and 'File Upload'. At the bottom, there is a 'Style Customization' section with a 'Show' button.

To integrate the form, copy the Form Integration ID displayed above the Form Settings in Custom Forms navigation. Then, open your theme editor, add the app block named **B2B Custom Form**, and paste the copied Form ID. Once added, your B2B form will appear as expected.

Once the form is added, any customer submission will appear in the Custom Forms route in app. From there, you can review the submitted data and approve or reject the customer.

If a customer is approved, they will see the special B2B pricing on the site. If the customer is rejected, only regular pricing will be shown.

When a customer is approved, the **B2B_approved** tag is automatically added to their profile. If rejected, the **B2B_rejected** tag is added. These tags help administrators track B2B applications. Please do not remove these tags, as doing so may affect the app's functionality.

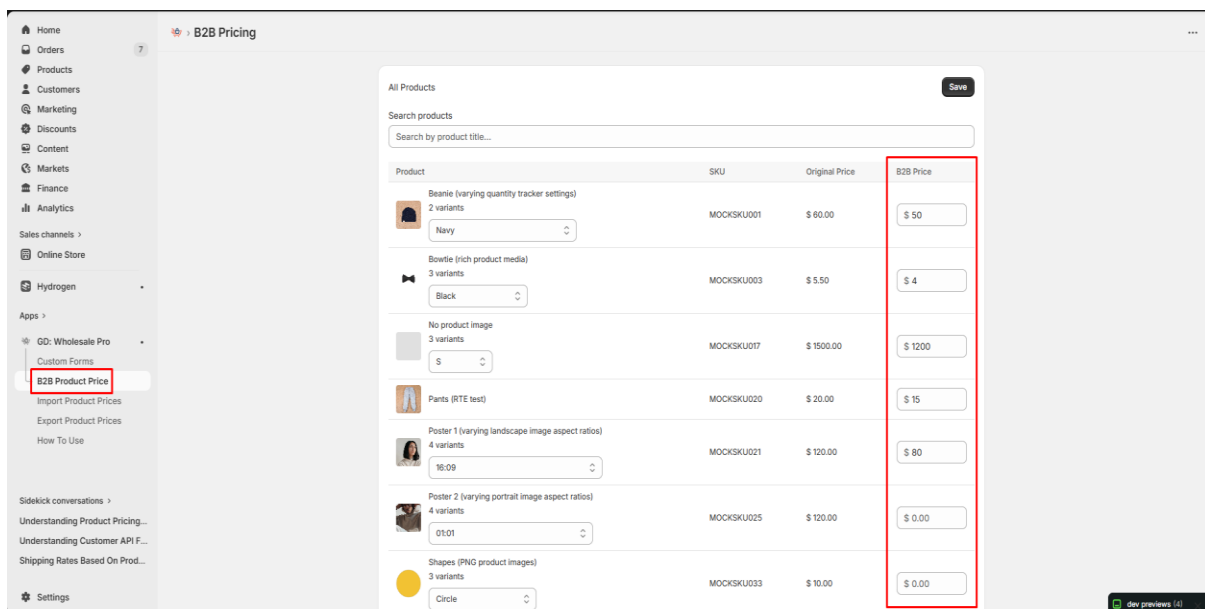


Second Step: Integration of B2B Product Price Block

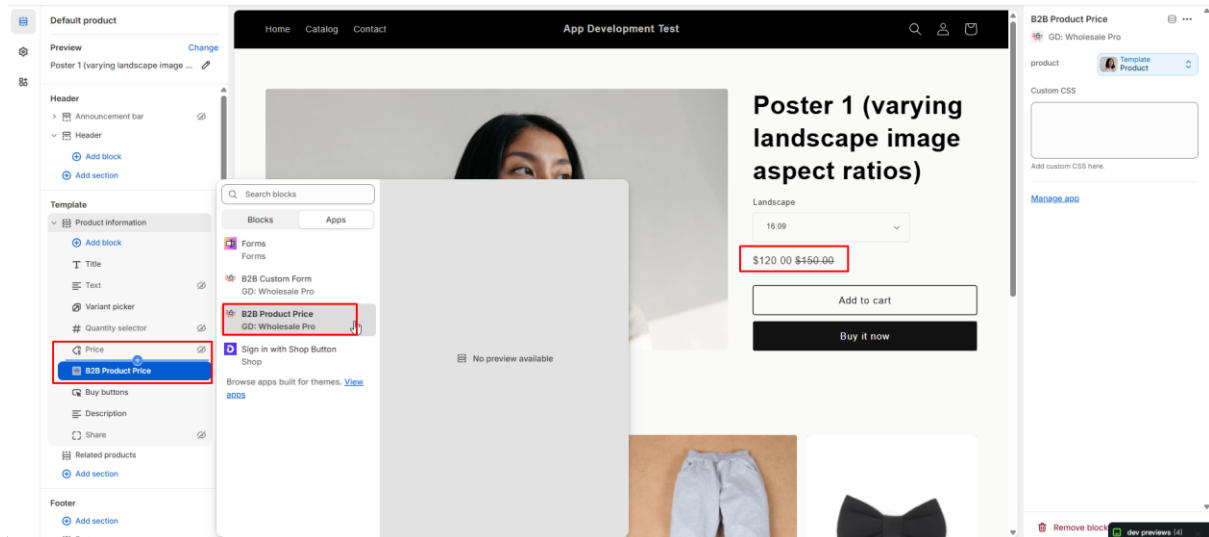
Next, set the special B2B pricing that will be shown to approved B2B customers. This can be done in two ways: manually or through bulk editing via Excel.

For manual updates, navigate to the B2B Products section and enter the desired price in the **B2B Price** column for each product, as shown in the image below.

Bulk price updates using an Excel sheet is explained later on in this documentation.



To display B2B pricing on your live site, open the theme editor and remove or hide the existing Price block. Then, add the app block named **B2B Product Price** everywhere in the theme where price block is present.



Along with adding the block, you also need to add the suggested CSS classes at the specified locations, as mentioned.

- **js-gd-ext-variant-picker:** Add this class to the variant picker on the product page. Variants can be displayed in two ways—buttons or a dropdown.
 - If variants are displayed as buttons, add this class to all radio `<input>` elements of the variant buttons.
 - If variants are displayed as a dropdown, add this class to the `<select>` element of the variant picker.
- **js-gd-ext-selected-variant-id:** Add this class to the hidden input field (`<input type="hidden" name="id" ...>`) inside the product `<form>` element. This input's value should update when different variants are selected on the product page.
- **js-gd-ext-pdp-info-section:** Add this class to the container of the product information section on the product page.
- **js-gd-ext-product-info-container:** Add this class to the Product Details container within the Product Information section in the product page. The Product Information section is typically divided into two parts: media and product details. Apply this class to the container that holds the product details not media.

The classes mentioned above must be added to your theme to ensure that B2B pricing displays correctly wherever the B2B Product Price block is used.

In **newer themes** like Horizon that use a **block-based structure**, product cards also include a **Price block**. In this case, simply replace the existing **Price block** with the **B2B Product Price block**, and it will work as expected.

However, in **older themes** like Dawn where product cards do not use Price blocks and instead rely on the **price snippet**, our block cannot be added directly. For such themes, a different approach is required, which is explained in the later following sections of this document.

Some themes include a **Sticky Add to Cart section** on the product page, which does not use the Price block. In such cases, the price is rendered using the price snippet. To display B2B pricing correctly, locate where the price snippet is rendered, comment it out, and add the following snippet in its place:

```
<b2b-sticky-cart-price class="b2b-sticky-cart-price-container">
  <div class="js-gd-ext-sticky-add-to-cart-price">
    </div>
  </b2b-sticky-cart-price>
```

Now, for older themes that do not follow a block-based structure (unlike newer themes such as Dawn), updating the price on the product card requires a different approach. In these themes, the price is rendered using the price snippet.

To display B2B pricing correctly, locate where the price snippet is rendered in the product card, comment it out, and add the snippet provided below in its place.

```
{% comment %} App block {% endcomment %}

{% assign has_b2b_tag = false %}
{% for tag in customer.tags %}
  {% assign tag_downcase = tag | downcase %}
  {% if tag_downcase == 'b2b_approved' %}
    {% assign has_b2b_tag = true %}
  {% endif %}
{% endfor %}

<b2b-card-price class="b2b-card-price-container">
  <div class="js-gd-ext-card-price"></div>

<script type="application/json" data-prod-card-price>
  {
    "isB2B": {{ has_b2b_tag | json }},
    "moneyFormat": {{ shop.money_format | json }},
    "variantsData": {
      {% for variant in card_product.variants %}
        "{{ variant.id }}": {
          "price": {{ variant.price }},
          "compare_at_price": {% if variant.compare_at_price %}{{ variant.compare_at_price }}{% else %}null{% endif %},
          "b2b_price": {% if variant.metafields.app.original_price.value %}
            {{ variant.metafields.app.original_price.value | times: 100 }}
          {% else %}
            null
          {% endif %}
        }{% unless forloop.last %},{% endunless %}
      {% endfor %}
    }
  }
```

```
    }  
  }  
</script>  
</b2b-card-price>
```

{% comment %} End App block {% endcomment %}

In the code snippet above, note the for loop ({% for variant in card_product.variants %}). In most themes, the product object in the product card file is named **card_product** only. However, in some themes this may differ.

If the product card file uses **product** as product object instead of **card_product** as the product object, you must update the loop accordingly to {% for variant in product.variants %}.

Please ensure this is checked and updated where necessary, as it is required for the code to work correctly.

You're now all set to use the B2B features.

Updating Product Price using Import/Export Module:

This module allows you to update variant prices, compare-at prices, and B2B prices for all products in bulk within minutes. It is primarily designed to simplify B2B pricing updates while also giving you the flexibility to adjust regular and compare-at prices at the same time.

To update prices, simply import an Excel file that includes the SKU column along with any pricing columns you want to update, such as Price, Compare-at Price, or B2B Price. The SKU column is mandatory for accurate updates. For a faster workflow, you can use the Export feature to download all variant pricing data with a single click. After modifying the required values, re-import the file to quickly apply the changes, making the entire pricing update process smooth, fast, and efficient.

Contact Us for Support:

Please follow this document to complete the full setup of our app and ensure everything is functioning correctly. If you experience any difficulties during integration, feel free to contact us at dev@growthdigital.com.au. We're just one message away and happy to assist you at any time.