

# Setting up your own Shopify Shop

A quick guide for developers

# By the end of this you should be able to:

- Know how to create a shopify partner account.
- Create a development shop with test data
- Create a simple gui for interacting with the shopify API
- Utilize the shopify api to modify existing items in the shop's database
- Utilize a “dotenv” file to hide important values such as api keys

# First things first:

Navigate to <https://shopify.dev/> and click on the link in the top right to create an account.

Follow the instructions and select "Other" as your main focus.

Provide "Ontario" as your province when prompted.

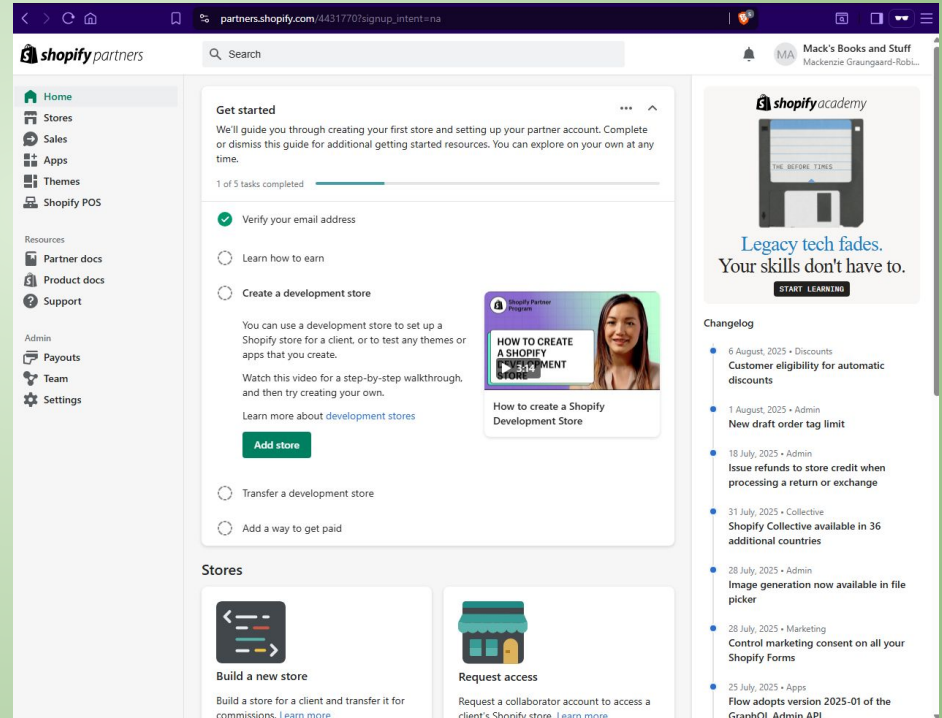
Fill out the following form in order to move onto the next step.

# Create a development store

In order to create a development store for testing purposes you must now be on the homepage for your partner account.

Click on “Create a development store”

Select “Add store”



The screenshot shows the Shopify Partners dashboard for a partner account. The left sidebar contains navigation links: Home, Stores, Sales, Apps, Themes, Shopify POS, Resources (Partner docs, Product docs, Support), and Admin (Payouts, Team, Settings). The main content area is titled 'Get started' and shows a progress bar with 1 of 5 tasks completed. The tasks are: Verify your email address (completed), Learn how to earn, Create a development store (selected), Transfer a development store, and Add a way to get paid. The 'Create a development store' task is expanded, showing a video guide and a button to 'Add store'. Below this, the 'Stores' section is visible, with options to 'Build a new store' and 'Request access'.

shopify partners

Search

Get started

We'll guide you through creating your first store and setting up your partner account. Complete or dismiss this guide for additional getting started resources. You can explore on your own at any time.

1 of 5 tasks completed

- ✓ Verify your email address
- Learn how to earn
- Create a development store
- Transfer a development store
- Add a way to get paid

You can use a development store to set up a Shopify store for a client, or to test any themes or apps that you create.

Watch this video for a step-by-step walkthrough, and then try creating your own.

Learn more about [development stores](#)

[Add store](#)

How to create a Shopify Development Store

Build a new store

Build a store for a client and transfer it for commissions. [Learn more](#)

Request access

Request a collaborator account to access a client's Shopify store. [Learn more](#)

shopify academy

Legacy tech fades. Your skills don't have to.

START LEARNING

Changelog

- 6 August, 2025 • Discounts  
Customer eligibility for automatic discounts
- 1 August, 2025 • Admin  
New draft order tag limit
- 18 July, 2025 • Admin  
Issue refunds to store credit when processing a return or exchange
- 31 July, 2025 • Collective  
Shopify Collective available in 36 additional countries
- 28 July, 2025 • Admin  
Image generation now available in file picker
- 28 July, 2025 • Marketing  
Control marketing consent on all your Shopify Forms
- 25 July, 2025 • Apps  
Flow adopts version 2025-01 of the GraphQL Admin API

# Create a development store

Select “Create a store to test and build”

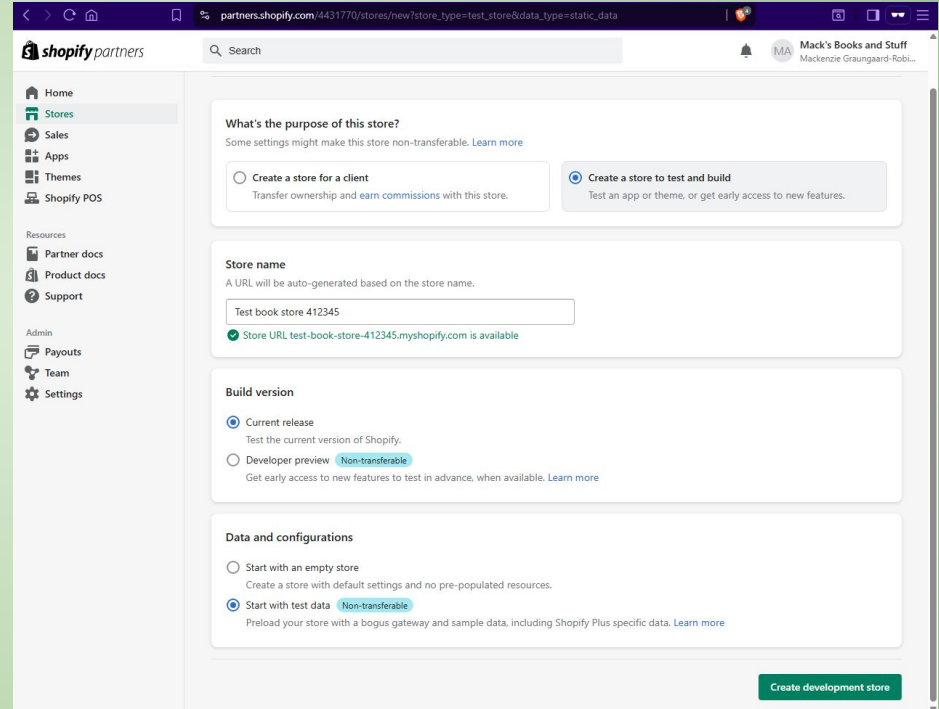
Give a random name to your store

Leave “Build Version” default

Under “Data and configurations”:

-select “Start with test data”

Press the Create button.



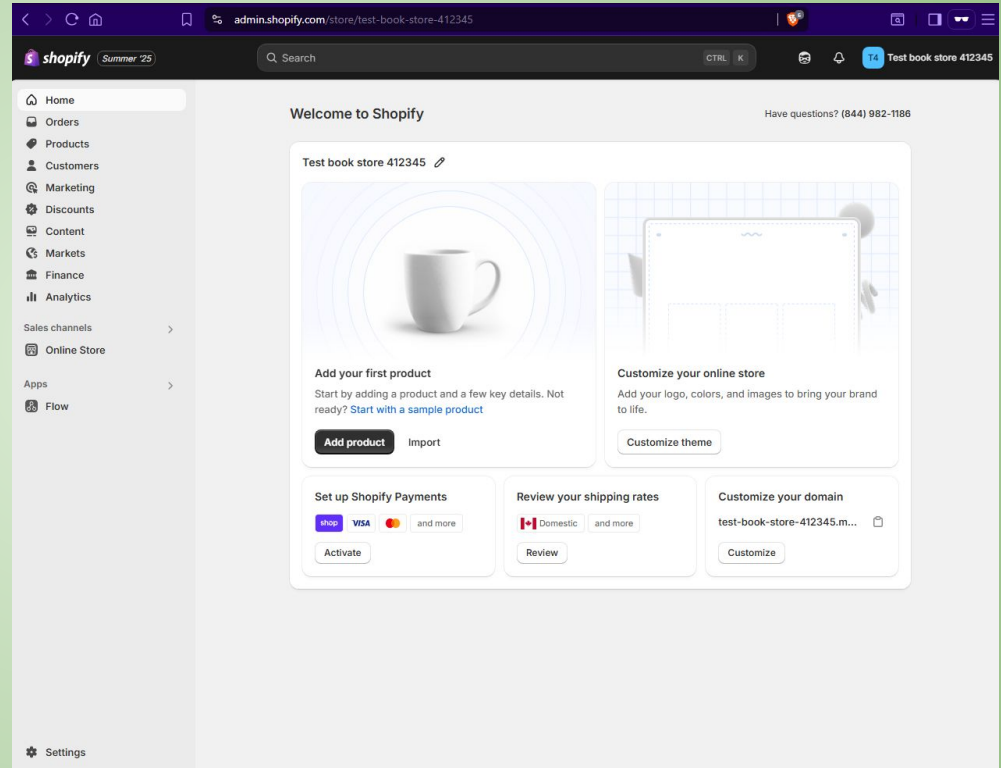
The screenshot shows the 'Create a development store' form in the Shopify Partners interface. The form is titled 'What's the purpose of this store?' and includes a sidebar with navigation links: Home, Stores, Sales, Apps, Themes, Shopify POS, Resources, Partner docs, Product docs, Support, Admin, Payouts, Team, and Settings. The form contains the following sections:

- What's the purpose of this store?**: Two radio button options. 'Create a store to test and build' is selected, with a description: 'Test an app or theme, or get early access to new features.'
- Store name**: A text input field containing 'Test book store 412345'. Below it, a green checkmark indicates 'Store URL test-book-store-412345.myshopify.com is available'.
- Build version**: Two radio button options. 'Current release' is selected, with a description: 'Test the current version of Shopify.' The 'Developer preview' option is marked as 'Non-transferable'.
- Data and configurations**: Two radio button options. 'Start with test data' is selected, with a description: 'Preload your store with a bogus gateway and sample data, including Shopify Plus specific data.' This option is also marked as 'Non-transferable'.

A green 'Create development store' button is located at the bottom right of the form.

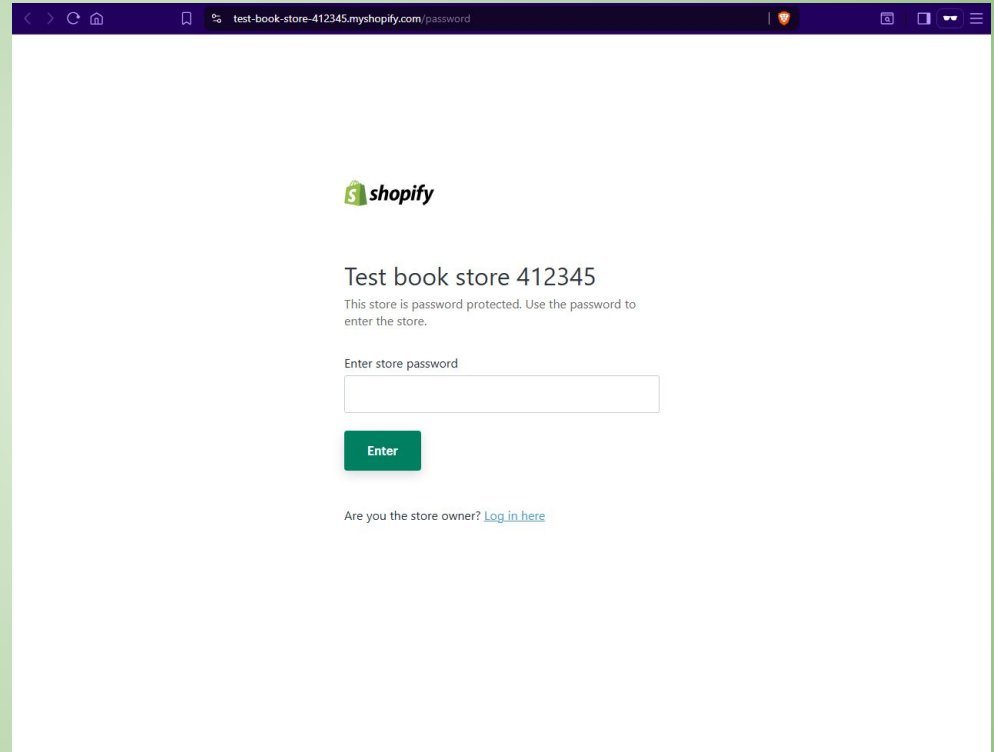
# Create a development store

We now have a shopify store with a custom URL!



# Accessing your store

So we have our new store, but how can we access it?



# Accessing your store

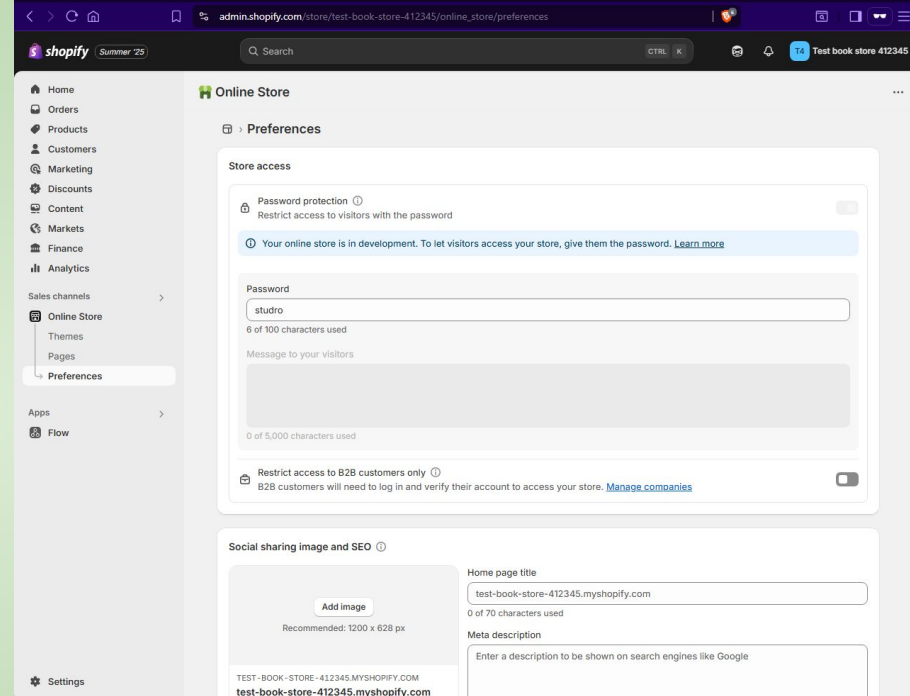
Navigate back to tab that is the store's homepage.

On the left side panel, select “Online Store”

Select “Preferences” from the dropdown.

The password that is display is the password used to log in and access the public version of your store.

In my case the password is “studro”.

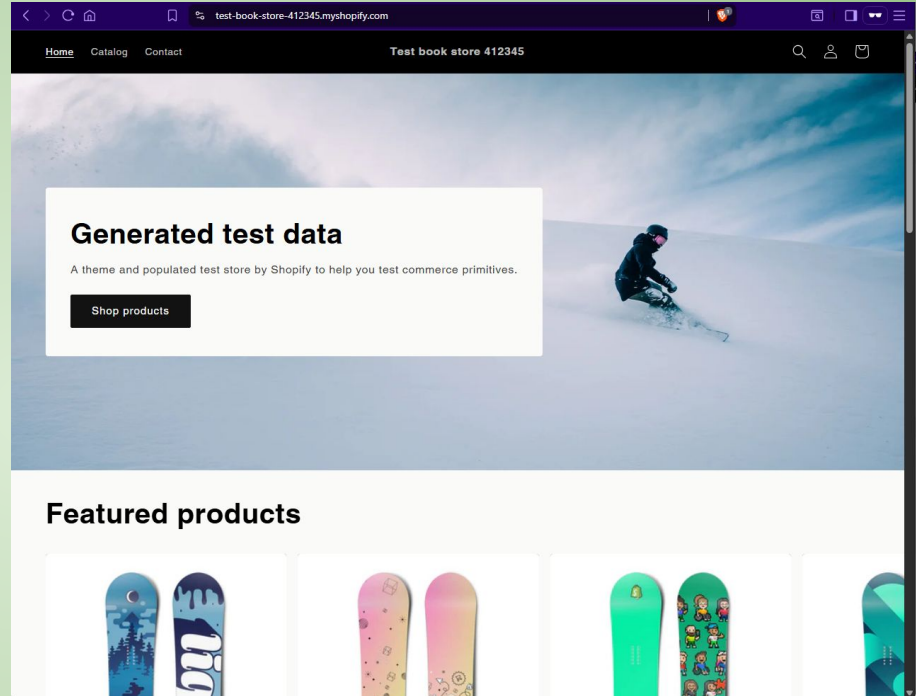




# Accessing your store

Use the password you find to access your store.

When logged in, the store should display something similar to this →



# Editing products via api

Now that the store is set up we will move onto using the shopify api to edit some of these test products.

To do so, clone the following repo for our starter code:

<https://github.com/Mackintac/shopify-quickstart-exercise>

The provided code is a simple GUI app that utilizes axios, express, and dotenv.

# Editing products via api

Since the code is now downloaded, lets install the required packages.

Run this command in the project directory's terminal without quotes:

```
“npm install axios dotenv express”
```

Each of these are required for the following reasons:

Axios - Simplifies the process of sending HTTP requests.

Express.js - Our web framework for handling web requests

Dotenv - A package that allows us to use .env files to hold important and private values such as api keys, without exposing them in our code.

# Editing products via api

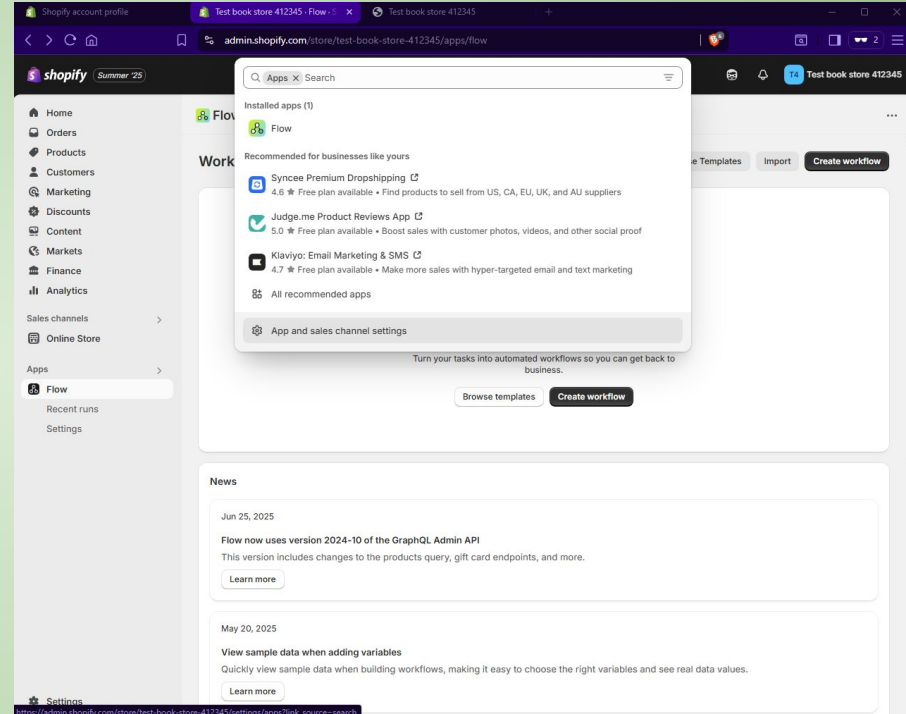
Before we are able to communicate with our store, we need to create a Shopify “App”.

On the left side panel: select Apps.

On the search bar that pops up: select “App and sales channel settings”

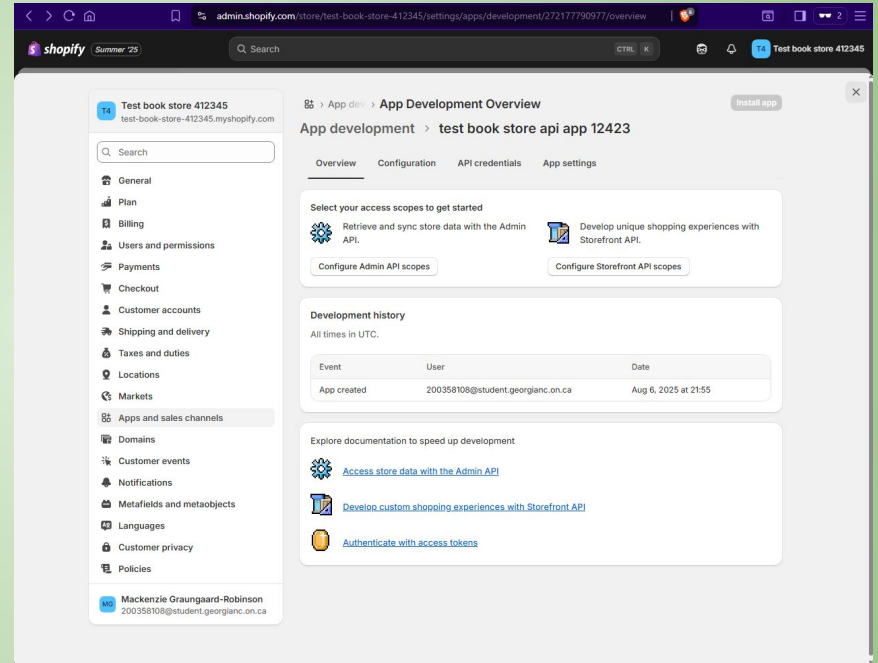
On the following screen click on “Develop Apps” → allow custom app development → Create an app.

Give your app a name.



# Editing products via api

On the App Development Overview screen select: “Configure Admin API scopes”.

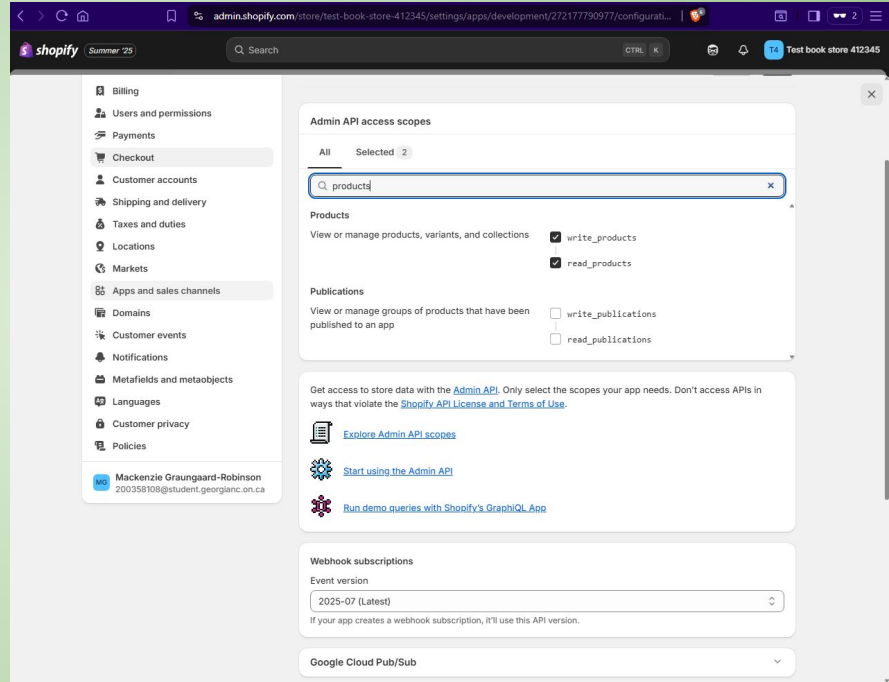


# Editing products via api

Enable the following permissions:

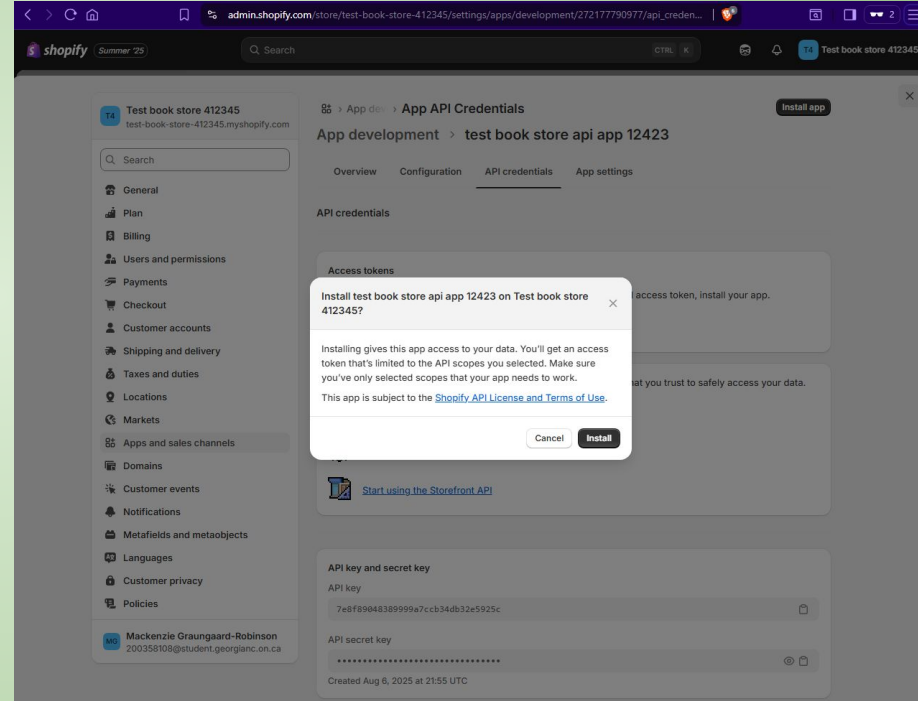
- read\_products
- write\_products

Click save at the bottom of the page.



# Editing products via api

Add the app to your store by clicking the “Install app” button near the top of the page.

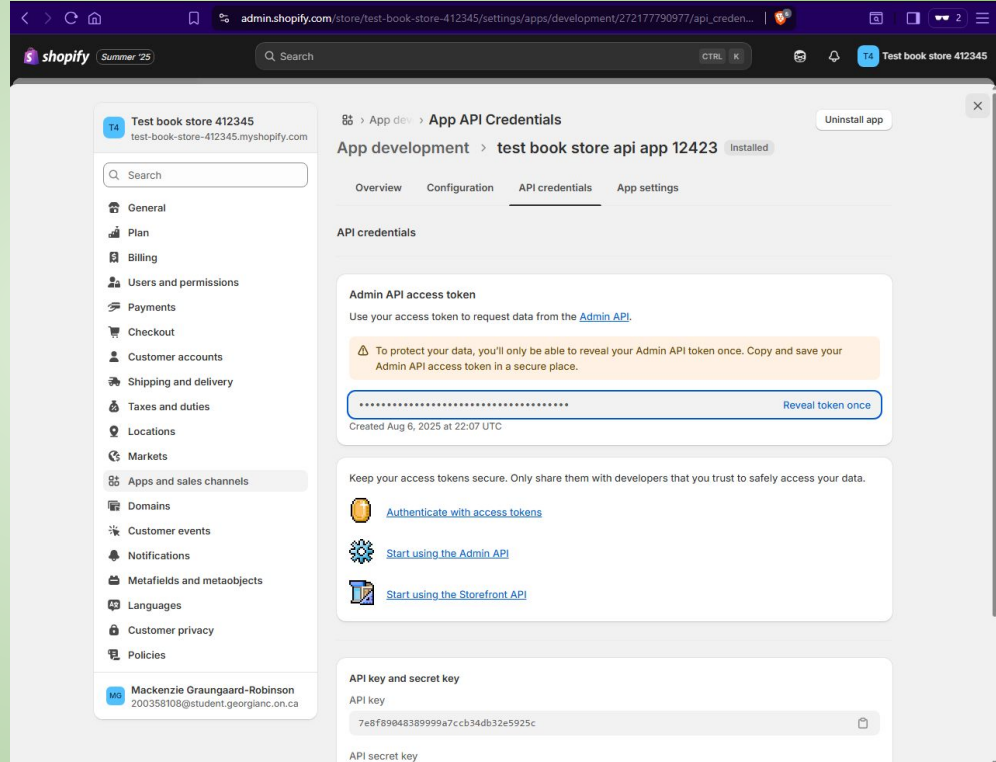




# Editing products via api

Now that the app is installed we are provided with an Admin API Access token.

Reveal the token and immediately copy it and save it to a text file or wait until the next step.

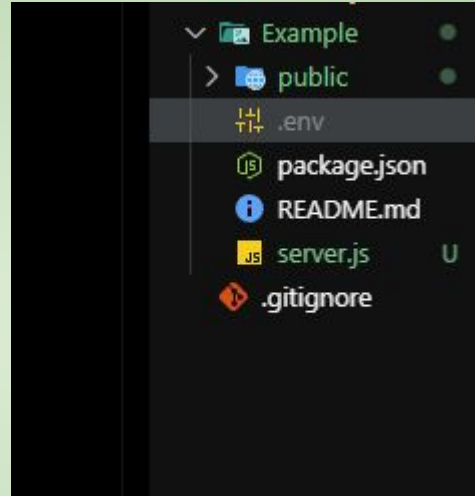




# Editing products via api

Back in your code editor create a new file in the root directory of the project named “.env”.

This will hold sensitive information that we will use in the app.




# Editing products via api

Create two variables named:

- ADMIN\_TOKEN
- STORE\_NAME

Follow the variable assignment syntax and paste your Admin API access token for “ADMIN\_TOKEN”

And the name of your store for ‘STORE\_NAME’.

```
Example >  .env  
1 ADMIN_TOKEN=shpat_adjspoijsasopijdsajdsaijdoas  
2 STORE_NAME=Test book store 412345
```

# Editing products via api

In your terminal run the command:

- Npm start

```
DESKTOP-656SR0S ~/d/j/Demo (main)> npm start
```

```
> node server.js
```

```
[dotenv@17.2.1] injecting env (4) from .env -- tip: 🛡️ prevent building .env in docker: https://dotenvx.com/prebuild  
Server running on http://localhost:3000
```

Our app is now running on port 3000  
of our machine (localhost)

Open up that link in a browser tab.

# Editing products via api

Now we have a GUI app that allows us to modify the existing items in our shop!

Just select the item you want to modify in the dropdown menu, give it a new name & price, and finally press the Update Product button to update the product's information!

