



Leveraging Shopify for upsells & revenue boost



Leveraging Shopify for upsells & revenue boost



By implementing Shopify with Solidgate billing, you can easily offer upsells to your customers after their first purchase on your website. It's as simple as creating a cart permalink that redirects users to a page showcasing their purchased items, along with instructions.

This way, you can provide additional physical goods, such as t-shirts or books, as special offers to interested customers. It's a great way to enhance their shopping experience, offer them even more value, and maximize your revenue.

Process breakdown

01

User makes a purchase on your website via Solidgate using their card data

03

The user agrees to the upsell, and you perform the purchase using a Solidgate token

05

The user places an order

07

You make a request to mark the order as paid

02

After the purchase, you offer an upsell of your predefined physical goods (e.g., t-shirts, books, etc.)

04

After the successful payment, the user will be redirected to a permalink you created earlier (that will be passed within the success URL). The permalink will display the predefined physical goods and instructions for the user

06

You receive a webhook notification from Shopify

Implementation flow: Setting up manual payment method

01.A

Access your Shopify account and go to *Settings*

The screenshot shows the Shopify admin interface. On the left is a navigation sidebar with options like Home, Orders, Products, Customers, Content, Analytics, Marketing, Discounts, Sales channels (Online Store, Point of Sale), and Apps. The main content area features a promotional banner for POS Pro with a price of \$89 USD/mo and a list of features: Staff and store management, Checkout & multi-channel selling, Inventory management and reporting, and Clienteling and customer loyalty. Below this, it lists included features for all POS plans, such as All POS Lite features and compatibility with retail accessories. At the bottom, there are two more promotional banners: 'SHOPIFY POINT OF SALE' and 'Unlimited POS Staff PRO Manage your team'.

Implementation flow: Setting up manual payment method

01. B

Click on *Payments*

The screenshot displays the Shopify Settings interface. On the left, a sidebar lists various settings categories, with 'Payments' highlighted in blue. The main content area is titled 'Payments' and contains several sections: 'Payment providers' with a 'Choose a provider' button; 'Supported payment methods' with an 'Add payment methods' button; 'Manual payment methods' with an 'Add manual payment method' button; and 'Payment customizations' with a link to the 'Shopify App Store'. On the right, the 'Payment capture' section is set to 'Automatic' and includes a 'Manage' button.

Implementation flow: Setting up manual payment method

01. C

In the *Manual payment methods* section, select *Add manual payment method*

01. D

Click on *Create custom payment method*

The screenshot shows the Shopify Settings interface. On the left is a navigation menu with 'Payments' highlighted. The main content area is titled 'Payments' and contains three sections: 'Payment providers', 'Supported payment methods', and 'Manual payment methods'. The 'Manual payment methods' section has a dropdown menu open, showing options like 'Create custom payment method', 'Bank Deposit', 'Money Order', 'Cash on Delivery (COD)', and 'I have already paid'. To the right of the main content is a 'Payment capture' section with an 'Automatic' radio button selected and a 'Manage' button.

Implementation flow: Setting up manual payment method

01. E

Set the name for the new method (for example, name it *"I have already paid"* if your payments are processed through Solidgate)

The screenshot shows the Shopify Settings interface. On the left is a navigation menu with 'Payments' highlighted. The main content area is titled 'Payments' and contains three sections: 'Payment providers' with a 'Choose a provider' button; 'Supported payment methods' with an 'Add payment methods' button; and 'Manual payment methods' which includes a text entry field containing 'I have already paid' and a 'Manage' link, followed by an 'Add manual payment method' button with a dropdown arrow. To the right of the 'Manual payment methods' section is a 'Payment capture' section with an 'Automatic' radio button selected, a descriptive text, and a 'Manage' button.

Implementation flow: Creating cart permalink

02. Cart permalink

Generate a cart permalink that redirects users to a page displaying the items they paid for along with instructions. You can find detailed information by clicking the button below

Check instructions

Example of cart permalink <https://5a1509.myshopify.com/cart/45548668518708:1> and where it leads

5a1509.myshopify.com/checkouts/c/dcf343b4d1436e7c905c33c25ede303f/information

My Store

[Cart](#) > [Information](#) > [Shipping](#) > [Payment](#)

Contact

Email me with news and offers

Shipping address

Country/Region
Ukraine

Save this information for next time

[Return to cart](#)

Continue to shipping



T-Shirt
M

€10.00

Subtotal

€10.00

Shipping

Calculated at next step

Total

UAH **€10.00**

Including €1.67 in taxes

Implementation flow: Adding a webhook endpoint

03. Webhook

Set up a webhook endpoint to receive notifications when an order is created. Follow the guide below

[Check instructions](#)

The screenshot shows the Shopify Settings interface. On the left, a sidebar menu lists various settings categories, with 'Notifications' highlighted in green. The main content area is titled 'Settings' and is divided into two columns. The left column contains 'My Store' information and a list of settings categories: Store details, Plan, Billing, Users and permissions, Payments, Checkout, Customer accounts, Shipping and delivery, Taxes and duties, Locations, Gift cards, Markets, Apps and sales channels, Domains, Customer events, Brand, Notifications (highlighted), Custom data, Languages, and Policies. The right column is titled 'Notifications' and contains sections for 'Customer notifications', 'Customize email templates', 'Sender email', and 'Orders'. A green callout box is overlaid on the top right, titled 'Webhooks', containing the text: 'You can subscribe to events for your products and orders by creating web hooks that will push XML or JSON notifications to a given URL.' Below this text is a 'Create webhook' button and a message: 'You haven't created any webhooks yet'. The 'Sender email' section shows the email 'timetoplay322@proton.me' with an 'Unverified' status and a 'Resend verification' link. The 'Orders' section lists three notification types: 'Order confirmation' (sent automatically), 'Order edited' (sent after editing), and 'Order invoice' (sent when there is an outstanding balance).

Implementation flow: API configuration

04. Creating a transaction for an order

To update your order from *Pending* to *Paid* via API, you need to send a request to create a transaction for such an order. Please refer to the documentation and step 5

[Documentation](#)

05. Receiving access token

A. Click on Settings in your Shopify account

- Choose *Apps and sales channels*;
- Go to *Develop Apps*;
- Click on *Allow custom app development*;
- Select *Create an app*;
- Enter the app's name and click *Create app*.

- Customer accounts
- Shipping and delivery
- Taxes and duties
- Locations
- Gift cards
- Markets
- Apps and sales channels**
- Domains
- Customer events
- Brand
- Notifications
- Custom data
- Languages



Start developing apps on your store

Develop your own apps to integrate your store's data with external services or build custom storefronts.

App development should only be done by trusted developers on your team. Apps can give API access to your data, and that data can be accessed outside of Shopify.

All apps are subject to the [Shopify API License and Terms of Use](#).

[Allow custom app development](#)



Create a custom app

Integrate your store's data with external services or build a custom storefront. Apps are subject to the [Shopify API License and Terms of Use](#).

[Create an app](#)

Implementation flow: API configuration

05. Receiving access token

B. Configure admin API scopes

- Choose *Order editing* scope. You can also choose any other scopes you want to access via API;
- Click *Save* to apply the selected scopes.

The screenshot shows the Shopify Admin API configuration interface. On the left, a sidebar lists various store settings, with 'Apps and sales channels' highlighted. The main content area is titled 'App development > Test' and shows the 'Configuration > Admin API integration' page. The 'Admin API access scopes' section is active, showing a list of scopes under the 'Order editing' category. The 'write_order_edits' and 'read_order_edits' scopes are selected, indicated by blue checkmarks. Other categories like 'Orders' and 'Packing slip management' have unselected scopes. A 'Save' button is visible in the top right corner of the configuration area. A sidebar on the right provides additional information and links, including 'Explore Admin API scopes', 'Start using the Admin API', and 'Run demo queries with Shopify's GraphQL App'.

Implementation flow: API configuration

05. Receiving access token

C. Receive API credentials

- Navigate to the *API credentials* tab;
- Select *Install app* to get an access token;
- Make sure to save the token, as it will be shown only once.

06. Order status

Now, you'll be able to send an API request to change the order status from *Pending* to *Paid*.

The screenshot shows the Shopify Admin interface. On the left is a navigation menu with 'Apps and sales channels' selected. The main content area is titled 'App development > Test' and has an 'Install app' button in the top right. Below the title are tabs for 'Overview', 'Configuration', 'API credentials' (which is active), and 'App settings'. The 'API credentials' section contains two main cards. The first card, 'Access tokens', states 'You don't have any access tokens yet. To get your Admin API access token, install your app.' and includes an 'Install app' button. The second card, 'API key and secret key', displays an 'API key' field with the value '1e6728aa9c37e12b4a1a3dabfcfb5c9' and an 'API secret key' field with masked characters. Below these fields, it says 'Created Jul 20, 2023 at 10:37 UTC'. To the right of these cards are two informational boxes: one about keeping access tokens secure with links to 'Authenticate with access tokens', 'Start using the Admin API', and 'Start using the Storefront API'; and another about using the client secret to verify incoming webhooks.

How it can look like: Upsell flow

Main purchase

Select Payment Method

2.99 USD

Beginner's Step-by-Step Coding Course – eBook

Apple Pay

Google Pay

PayPal

or pay with card

Card Number

1234 1234 1234 1234

VISA

MasterCard

AM EX

UnionPay

Expiration Date

MM / YY

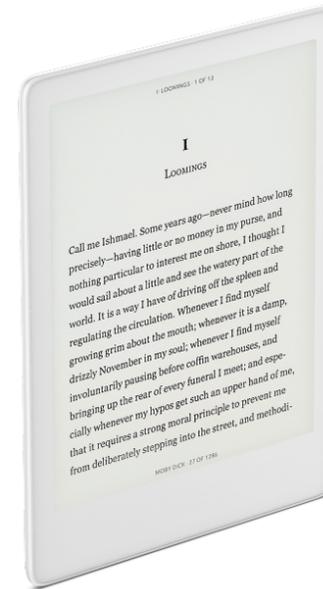
CVC 

Pay 2.99 USD

Upsell

eBook Reader 3

129.99 USD



The lightest and most compact eBook, now with a 6" 300 ppi high-resolution display

Buy now 129.99 USD

Order confirmation

Order details

Beginner's Step-by-Step Co...

2.99 USD

eBook Reader 3

129.99 USD

Total paid

132.98 USD

Shipping information

First Name

Last Name

Country

City

Address line

Postal code

Phone

XXXXXX

+380

Ship my item



Following these steps and processing payments with Solidgate allow you to effectively implement Shopify as an option to make upsells and potentially increase your revenue by offering additional physical goods to customers who agree to the upsell offer

