shopifyplus

Checkout extensibility upgrade kit

Your guide to planning your upgrade from checkout.liquid to checkout extensibility.

Contents

Introduction

This kit is designed to support you in planning your upgrade from checkout.liquid to checkout extensibility. In this kit, you'll find steps for upgrading, useful templates for assessing the current state of your checkout customizations, links to helpful resources, and more. While this kit includes resources to help plan your upgrade, you can find technical instructions for performing the upgrade in the Shopify help center and dev docs.

What is checkout extensibility?

Checkout extensibility makes it easier for Shopify Plus merchants to customize their checkout in a way that's app-based, upgrade-safe, higher-converting, and integrated with Shop Pay. It includes a suite of powerful platform features that let merchants make code-free customizations to their checkout and a collection of developer tools for making more advanced customizations via apps.

Platform features that let merchants make code-free customizations include:

- Checkout editor: An intuitive, drag-and-drop editor where merchants can customize the look of their checkout by customizing colors, fonts, and logos, and add new functionality by installing apps
- Shopify pixels manager: A new section in the Shopify admin where merchants can securely add and manage pixels that track customer events
- Checkout apps from the Shopify App Store: A collection of apps that merchants can install to add functionality to their checkout, built using native Shopify UI components and native APIs

Developer tools for making more advanced customizations include:

- Checkout UI extensions
- Web pixel app extension
- Branding API
- Shopify Functions
- Post-purchase extensions

Checkout extensibility replaces the need for checkout.liquid. As a result, any merchants that currently customize their checkout with checkout.liquid need to upgrade to checkout extensibility. Upgrading requires replicating any required checkout.liquid customizations in checkout extensibility using the platform features and developer tools listed above.

What are the benefits of checkout extensibility?

Code-free customization

For the first time ever, you can customize your checkout without touching code. With the new checkout editor, you can edit the look of your checkout, including colors, fonts, and logos. And you can quickly add new functionality to your checkout by simply installing apps from the Shopify App Store and configuring them in the checkout editor. The best part? Apps added in the checkout editor automatically inherit branding settings, making it even faster to add new functionality.

You can also install apps that add tracking pixels, post-purchase pages, and custom logic to your checkout.

Increased speed and conversion

Checkout extensibility is built with the latest platform technology, which boosts conversion by delivering a faster checkout experience for your customers.

Integration with Shop Pay

Add any or all of your checkout customizations to Shop Pay with a few clicks of a button in the checkout editor. This allows you to deliver a consistent experience across your guest and express checkouts, while benefiting from Shop Pay's 91% higher conversion rates on mobile and 4X faster checkout experience.

Upgrade safety and better security

Unlike using checkout.liquid, checkout extensibility doesn't require periodic upgrades—which means you can gain access to new checkout features as soon as they're available. You also benefit from better security because the components and APIs that power checkout extensibility run in a sandboxed environment.

Powerful new checkout features

The last upgrade for checkout.liquid took place in January 2020. Since then, Shopify launched powerful new features for Shopify Checkout, including:

New purchase options:

- Pre-orders

New payment option:

- Shop Pay Installments

New shipping and delivery functionality:

- Shop Promise
- Estimated delivery dates
- Shipping to pickup points
- Subscription shipping rate names

New discount functionality:

- Discount combinations
- Shipping discounts

New marketing features:

- SMS marketing opt-in
- Email marketing opt-in

New B2B functionality:

- Vaulted credit cards
- Flexible shipping addresses
- Customizable checkout experience

New checkout functionality:

- [Coming soon] One-page checkout
- Tooltip for estimated tax

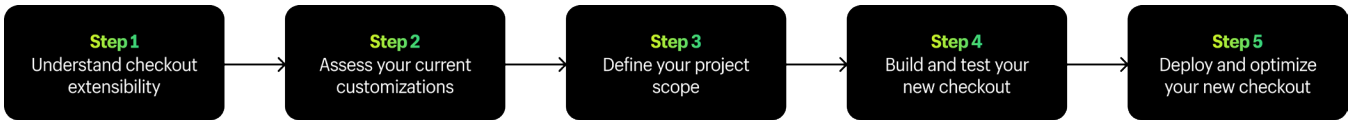
When you upgrade to checkout extensibility, you instantly gain access to these new features, plus any new checkout features as soon as they're available.

Flexibility to build the experience you want

Make advanced customizations and build the exact checkout experience you want by working with a developer. You can make advanced customizations to the look of your checkout with the branding API and build bespoke customizations for your checkout by developing custom apps with our collection of components and APIs.

Upgrading to checkout extensibility

In this section, you'll find a high level overview of the upgrade process from start to finish. Your upgrade journey may vary depending on your store's specific customization requirements.



Understand checkout extensibility

Set yourself up for success by learning about checkout extensibility and the upgrade process. This includes:

1. Reviewing available resources to learn more about checkout extensibility and the upgrade process, including:
   - Shopify help center
   - Shopify blog
   - Shopify dev docs
2. If necessary, engaging with Shopify or a Shopify Partner for guidance, high level upgrade timelines, and estimated level of effort.

Assess your current customizations

Review your existing checkout.liquid customizations so you can understand what you need to build with checkout extensibility. This includes:

1. Conducting an assessment of your checkout.liquid customizations, making sure to consider the value of each customization.
2. Mapping your checkout.liquid customizations to checkout extensibility's suite of features and develop tools.

Refer to our tips for assessing your checkout customizations on page 8 for detailed guidance.

Define your project scope

Outline the scope of work to upgrade from checkout.liquid to checkout extensibility and align all necessary stakeholders. This includes:

1. Identifying project stakeholders and any necessary development resources. For example, you may choose to work with a Shopify Partner to assist with your upgrade.
2. Documenting a project outline for your upgrade including timelines, sprint plans, feature and workstream rollouts, and priorities.

Build and test your new checkout

Build and test the next version of your checkout using checkout extensibility. This includes:

1. Creating a draft checkout in the checkout editor or setting up development and staging environments.
2. Installing any necessary apps or conducting custom development work.
3. Completing end-to-end testing for quality assurance, user acceptance, integrations, and more.

> Refer to our tips for customizing Shopify Checkout and upgrade instructions for detailed guidance.

Deploy and optimize your new checkout

Complete your upgrade from checkout.liquid to checkout extensibility. This includes:

1. Publishing your new checkout on your live store.
2. Conducting a live checkout test order to ensure everything is working as intended.
3. Monitoring your key performance metrics and activities, like order volume and upsells, to ensure everything is working as expected.

> Refer to our testing best practices and upgrade guide for detailed guidance.

Assessing your checkout customizations

Identify your current customizations

There are two ways to assess your current checkout.liquid customizations: a visual review and a code review. Make a copy of this Google Sheet to keep track of any customizations you identify throughout your assessment.

Visual review

First, you should visually assess your checkout. Start a checkout flow in the same way that your buyers would. If you have multiple types of buyers, make sure to go through the checkout flow in each way your buyers would. Proceed through each step of checkout and take note of any visible customizations. This could include:

- Trust badges
- Banners/custom messages
- Address input validation (preventing PO Boxes, limiting characters, etc.)
- Agree to Terms and Conditions Checkbox
- Extra input fields
- Hidden/removed input fields

You can compare your live checkout to an uncustomized version of Shopify Checkout to identify visual customizations more easily. To see what an uncustomized version of Shopify Checkout looks like, create a draft checkout and preview it.

Code review

Some customizations aren't visible unless a specific interaction takes place. Others are entirely invisible to the buyer, like saving certain information to order fields. The only way to obtain a full list of customizations and what they're doing is to perform a code-level review of your checkout.liquid theme file.

To view the file, under your currently published Theme "Actions" navigate to "Edit code" > "Layout" > "checkout.liquid". You can compare your checkout.liquid file to an uncustomized checkout.liquid file to identify customizations more easily.

When reviewing your checkout.liquid file, you may see different types of content that make up a customization. They will typically fall into one of four buckets: comments, liquid references, JavaScript, or styles/CSS.

Comments

There may be useful comments throughout checkout.liquid which indicate what a customization does. Comments can come in a few different forms depending on its context, so look for text similar to the following:

```
<!-- Trust badges -->
/* Text styling */
// Function to validate address
{% comment %}Terms and Conditions{% endcomment %}
```

Liquid references

When customizations have several elements to it, they are often contained in a single liquid snippet. This snippet is referenced in the checkout.liquid file like this:

```
{% render 'plus-required-checkbox' %}
{% include 'font-face' %}
{% include 'translation' %}
{% include 'checkout-tax-id-field' %}
```

The names of these snippets are usually descriptive enough for you to get an idea of what the customization does. If not, you may need to look into the referenced file within the "snippets" folder of the theme code and read the comments.

Javascript

Some customizations like apps or tracking pixels often require referencing custom JavaScript. These customizations are often contained within a “src” reference, and end in “.js” like this:

```
src="https://rebuyengine.com/js/rebuy?shop=store-name.myshopify.com"
src="https://xxxxxxxxxx.cloudfront.net/pobox_checker.js"
src="https://cdn-widgetsrepository.yotpo.com/v1/loader/xxxxxxx"
src="{{ 'checkout.js' | asset_url }}"
```

Based on the domain name of the URL being referenced, you can get an idea of where the JavaScript file originated from. For example, if it’s built by an app, the name of the app will likely be referenced in the domain. Along with possible organization names in code comments, this domain can help you understand which app partner, if any, was involved in the development of this customization. Once you know who this is you can approach them to confirm that their app/solution is compatible with checkout extensibility, helping simplify the upgrade process.

Styles / CSS

Custom styling may also be applied to checkout. These styles are used to alter the typography, colors, spacing, visibility, and positioning of elements in the checkout. If you have custom styles present, they might look similar to the following examples:

```
style="background-color: yellow;"
style="margin-bottom: 30px;"
<style>...</style>
{{ 'checkout.css' | asset_url | stylesheet_tag }}
```

While it may not be obvious what elements on the page are being targeted, the style values themselves can give you a sense of what is being altered, for example “background-color” and “margin-bottom.”

> If you need assistance with identifying the customizations in your checkout, reach out to the original developer of the customization or a Shopify Partner.

Evaluate the customization value

Over time, customizations can be added and forgotten in your checkout. Some of these customizations bring value to your business and should be replicated with checkout extensibility, while others may no longer be needed. In order to avoid unnecessary work while performing this upgrade, it's a good idea to evaluate the value of each customization you identified in your assessment.

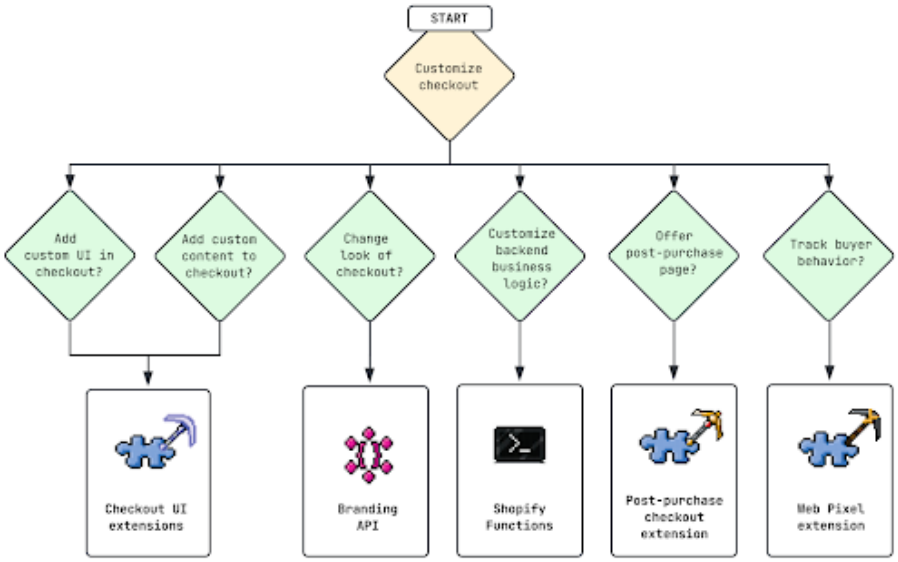
Understanding the value of each of these customizations will require a discussion with the appropriate stakeholders in your organization. Some questions to ask as part of this discussion include:

- What business value does this customization bring?
- Does the value of this customization justify the cost of replicating it in checkout extensibility?
- Is there another way to achieve this value? For example, if the current customization improves conversion rate, is there another way to improve conversion rate using public checkout apps or a storefront customization?

Map your customizations to checkout extensibility

At a high level, your customizations may fall into one of the following buckets: adding custom UI and content to checkout, changing the look of checkout, customizing the backend business logic of checkout, offering a post-purchase page, or tracking buyer behavior.

Note: It might not be possible to replicate all of your checkout.liquid customizations using checkout extensibility at this time. Refer to our product roadmap to see if your customization will be supported by checkout extensibility in the future. You can also work with a Shopify Partner for help mapping your customizations to checkout extensibility.



Adding custom UI or content to checkout

If your checkout includes custom UI or content such as upsells, additional fields, informational banners, or loyalty programs, you can replicate these using checkout UI extensions. Checkout UI extensions are deployed via an app and allow app developers to build custom functionality and UI that can be installed at defined points in the checkout flow. Checkout UI extensions are additive only and cannot be used to remove or hide elements.

You can either install an app from our growing collection on the Shopify app store and configure it in the checkout editor, or develop a custom app by following our tutorials.

If a customization in your checkout.liquid file is being generated by an app and you're unsure if it will work when you upgrade to checkout extensibility, reach out to the app developer.

Changing the look of checkout

If your checkout has custom styling, including logos, colors and fonts, you can likely replicate most of these customizations in the branding section of the checkout editor.

However, if you need finer control over the style of your checkout—like customizing the corner radius on buttons, for example—you can use GraphQL Admin API.

Customizing backend business logic

If your checkout has custom backend business logic, like custom discounts, or custom payment and delivery options—which are likely deployed using Shopify Scripts—you can replicate it in checkout extensibility using Shopify Functions. Shopify Functions are deployed via an app and allow developers to customize the backend logic that powers parts of Shopify.

You can either install an app from our growing collection on the Shopify app store, or develop a custom app by following our tutorials.

Note: Shopify Scripts are compatible with checkout extensibility, so you can continue using them to add custom logic throughout your checkout for now. However, Shopify Scripts are being deprecated and will no longer work starting August 13, 2024, so you'll eventually need to migrate your scripts to Shopify Functions.

Offering a post-purchase page

If your checkout has a post-purchase page—for loyalty programs or upsells, for example—you can replicate it in checkout extensibility using post-purchase extensions. Post-purchase extensions are deployed via an app and allow you to add a post-purchase page directly in Shopify Checkout. The post-purchase page appears after the order is confirmed, but before the thank you page.

You can either install an app from our growing collection on the Shopify app store, or develop a custom app by following our tutorials.

Tracking buyer behavior

If you have tracking pixels present in your checkout, either installed directly, or through a tag manager like Google Tag Manager, you need to migrate them to our new pixel manager. This includes, but is not limited to, pixels like Meta and Twitter tracking scripts. In the new pixel manager, pixels are added via apps built with the web pixel app extension.

You can either install an app pixel from our growing collection on the Shopify app store, or add a custom pixel. You can also add Google Tag Manager in checkout extensibility via a custom pixel.

Building and testing your new checkout

There are two ways to build and test your new checkout with checkout extensibility:

1. Creating a draft checkout in the checkout editor
2. Setting up a development store

During the assessment of your checkout customizations, you should have determined whether you're able to replicate your customizations in checkout extensibility using existing tools and apps or if you need to build custom apps. If you're able to achieve your customizations with public apps and editor tools, you can create a draft checkout in the checkout editor. If you need to build custom apps, then you need to set up a development store.

Creating a draft checkout

1. Navigate to Settings > Checkout in the admin
2. Either create a new draft checkout or select an existing checkout duplicate an existing draft and click "Customize"
3. In the branding section of the checkout editor, customize the colors, fonts, and logo in your checkout to match your branding
4. Install any apps required to customize your checkout, including any app pixels which are configured in the pixels manager
5. Configure your apps in the customizations section of the checkout editor
6. Preview your draft checkout
7. Once you're happy with the way your checkout looks, you can publish your draft checkout. This will replace your live checkout that's customized with checkout.liquid

Setting up a development store

1. Visit the Shopify Partners site and create an account if you don't already have one
2. Create a development store with access to the checkout UI extensions developer preview
3. Install Shopify CLI. In there, you can create an app following the CLI app tutorial or the specific checkout extensions custom field tutorial
4. The app and store setup addresses beta flags and default scopes. For additional scopes, configure from the app listing in the Partners site.

> If you have any questions, please reach out to your Shopify contact or Shopify Plus Support.

Testing best practices

Whether you're using a draft checkout or development store to test your customizations, it's important to replicate your customers' experience as accurately as possible. If you're using a development store, it needs to be set up to mirror your production store as closely as possible. This includes replicating:

- Theme (identical to production)
- Apps installed in theme
- Products (descriptions, variants, images, etc)
- Collections
- Metafields (on all applicable objects)
- Discounts (automatic and codes)
- Functions
- Shipping profiles
- Tax settings
- Payment gateways

It's also important to test each checkout journey for each customer group that exists for your store, making sure that specific aspects of the checkout that apply only to select groups of customers work only for those customers.

If you're using a draft checkout in the checkout editor of your main store to perform testing, the items in the list above should be available when previewing the draft profile. The checkout editor has preview functionality, allowing you to configure and test your new checkout experience end-to-end before publishing the upgrade. We recommend thorough testing to ensure everything is set up correctly beforehand.

When testing bespoke checkout customizations, each customization should initially be tested individually (where it's possible to test individually) on the development store to confirm it's performing as expected. Once each checkout customization has been tested and works as expected on its own, they can be tested altogether.

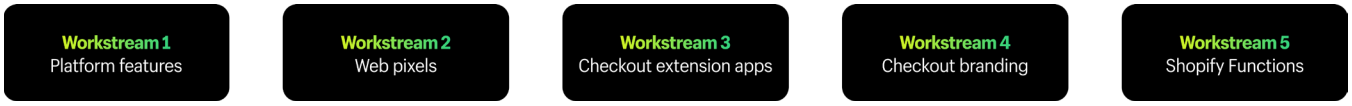
Currently, there's no tool to bulk apply the changes made on the development store to the production store. As such, each change (theme, apps, extensions, shipping profiles, etc) that is made on the development store (e.g. deploying and installing apps, configuring branding in the checkout editor, etc) needs to also be made manually on the production store when you're ready to move these changes to production and complete your upgrade.

Tips for testing pixels

- For the best integration experience, always check if there's a compatible app pixel available before implementing a custom pixel.
- The best way to confirm your web pixels are properly configured, firing, and captured is to use tools provided by the app or third party that pixel data is being sent to. Navigate to your shop as a buyer, perform the actions needed to trigger the relevant events, and then confirm in the third party's tooling that the data is received. Some third parties have purpose-built browser extensions for testing (examples include Facebook Pixel Helper, Twitter Pixel Helper, and TikTok Pixel Helper) . Others may require you to create a new test pixel and then review reporting or other tools in their own interfaces. Contact the third party for support in testing the capture and display of this data in their tools.
- Most browsers have dev tools that capture network calls while websites are loading and operating. For example, Google Chrome's dev tools has a Network tab that can be used to confirm web pixel events are firing. You may need to search and filter for your specific third party domain to find the details.
- When working with custom pixels, the console.log() javascript method can be used to log event details in the browser's console in dev tools. This allows you to confirm that events are firing and inspect the data being passed along.

Upgrade instructions

For a more seamless upgrade journey, we recommend sorting your existing customizations into workstreams including platform features, web pixels, checkout extension apps, checkout branding, and Shopify Functions. Note that not all workstreams may be applicable to you, and the order in which you approach each workstream mayl vary depending on your customizations, resourcing, dependencies, and other variables.



Platform features

When assessing your checkout.liquid customizations, keep in mind that some of your customizations may now be built right into Shopify (i.e. you don't need to replicate them with checkout extensibility).

Common use cases that are natively supported in Shopify Checkout include, but are not limited to, the following:

- SMS marketing consent flags
- Increased locations count
- Inventory states
- Bundles
- Extensible discounts
- Checkout editor and checkout apps
- Shopify discounts
- Shopify Functions
- Shopify metafields
- Shopify Scripts
- Pre-orders
- Shipping discounts
- Estimated delivery dates
- Shipping to pickup points
- Tooltip for estimated tax
- Subscription shipping rate names
- B2B selling
  - Vaulted credit cards
  - Flexible shipping addresses
  - Customizable checkout experience
- [Coming soon] One-page checkout

Web Pixels

All pixel and event tracking customizations in checkout.liquid must be migrated to the Shopify pixels manager, where you can securely add and manage third-party pixels that track customer events. Customer events are actions that take place in the customer's browser, for example, clicking a link or adding a product to a cart.

First, you need to determine what pixels you use today. Review the code across your store, from checkout.liquid to additional scripts, for code added by you or a partner that receives and subscribes to customer events. Identify and create a running list of all apps and pixels.

Once you have your list of pixels, you can start adding them in the new pixels manager. The fastest way to add pixels is by installing an app pixel. These apps are built with the web pixel app extension. Go to the app store and review the collection of apps with embedded pixels, or check with your app partner point of contact. Once you install one of these apps, the pixel will be added for you to subscribe to customer events. In cases where there is no app to support your needs, you can add a custom pixel. For the best integration possible, always look for an app pixel before adding a custom pixel.

Refer to our guide on migrating pixels and video tutorial on adding and modifying custom pixels for detailed guidance.

Note: Any pixels added to the Order Status Page/Thank You Page, such as those added in Additional Scripts, do not need to be migrated to the Shopify pixels manager before upgrading to checkout extensibility. However, we encourage you to explore centralizing all of your pixels in the pixels manager to make it easier for you to manage. In addition, web pixels are able to run on all surfaces of the Shopify storefront and checkout.

Checkout extension apps

Checkout extensibility empowers you to add functionality to your checkout by installing apps built with checkout UI extensions. You can either install apps from the Shopify App Store and configure them in the checkout editor, work with an in-house developer to build a custom app, or hire a Shopify Partner to build a custom app.

Some checkout apps can only be added only to specific areas of your checkout, like your Shipping Page, for example. Below are some resources that we've compiled regarding checkout extension apps.

Refer to these videos on the checkout editor and checkout UI extensions to get started.

Checkout branding

With checkout extensibility, you can customize the look of your checkout in the branding section of the checkout editor. There, you can add your brand's logo, add a background image, change the colors, and select your font. For advanced checkout branding customizations, you can use branding API.

> Refer to this video on the checkout editor to get started.

Shopify Functions

Shopify Functions is a powerful new way to extend and customize Shopify's backend logic to meet your unique business needs. With Functions, developers can build powerful customizations, including for order and product discounts, delivery and payment customizations, and shipping discounts. Shopify Functions customizations execute in under 5ms, and easily scale up for the biggest sales events on the planet.

Migrating from Shopify Scripts

1. Assess the Shopify Scripts currently running on your store. Refer to the discount, delivery, and payment customization comparison tables to see which use-cases are currently possible using Functions.
2. For the use-cases that can be achieved using Functions, check out the Shopify App Store to see if a publicly-available app meets your requirements. If not, you may want to consider developing your own custom Functions-based app (only available to Shopify Plus merchants). There are tutorials available for discounting, payment customizations and delivery customizations to get your development team started.
    - Please note that custom Functions-based apps must be created via the Partner Dashboard and then connected to your store. This is because Functions apps require Shopify App Bridge. Custom apps created in the admin do not currently support App Bridge.
3. Once you've identified a Functions-based app that meets your needs, you can install the app on your store and configure the settings directly in your Shopify Admin. After the app has been configured and tested, and you're confident that it meets your requirements, you can disable the relevant Shopify Script.

We recommend that you start with delivery or payment customization Functions and run those alongside your line-item discount scripts. Once you feel comfortable with Functions, progress to updating your Discounts.

If you're using Scripts to complete use-cases that are not yet supported with Functions, continue to use your Scripts until the Functions APIs support your use-case. In 2023 we will be releasing additional Functions APIs to cover the prior Scripts use-cases, and more. If you're waiting for select use-cases to be supported in Functions, rest assured that Scripts and Functions can co-exist on a store.

Check out the guidance below on how line item, shipping, and payment Scripts execute alongside Functions.

- Line item Scripts execution: Execute after Shopify's Discount logic has completed (which includes discounts powered by Functions) and can see the amounts discounted.
- Line item Scripts limitations:
    - When using reject (discount code) it will only operate on the first discount code applied to checkout.
    - You can only access the first discount code applied to checkout, and don't have visibility into the other discount codes that may have also been used.
- Shipping Scripts execution: Executes after Delivery customization Function APIs. Does not operate on the modified delivery customization rename results, but does on move and hide.
- Payment scripts execution: Executes after Payment customization Function APIs. Does not operate on the modified payment customization rename results, but does on move and hide.

Refer to our blog post on Shopify Functions and video on Shopify Functions to get started.

Resources

Below is a compilation of all resources that have been linked throughout the checkout extensibility upgrade kit.

Shopify help center docs:

- Checkout extensibility upgrade guide
- Customizing and editing your checkout
- Pixels and customer events
- Creating development stores

Shopify dev docs:

- Customizing Shopify checkout
- Checkout extensibility product roadmap
- Create an app
- Install Shopify CLI
- checkout.liquid

Shopify App Store:

- Create a custom checkout with apps
- Pixels app collection

Videos:

- Shopify Checkout with Checkout Extensibility
- Find out more about the all-new drag-and-drop checkout editor
- Extend checkout in more ways with the latest from Shopify Functions and checkout UI extensions
- A Brand New Type Of Shopify Apps: Checkout Extensions
- Shopify Functions - a new way to customize Shopify
- How to add or modify custom pixels in your Shopify Store?

Blog posts:

- 5 Ways to Customize Shopify Checkout
- Checkout Extensibility Opens New Ways to Customize Checkouts on Shopify
- 10 Ways to Customize Checkout with Checkout Extensibility
- Shopify Functions: The New Way to Extend and Customize Shopify

Other:

- Partner Directory | Checkout Upgrade Service Partners
- Discord | Shopify Developer Discord
- PDF | Migrating to Pixels
- Google Sheet | Shopify Checkout Upgrade Helper

If you have any questions, please reach out to your Shopify contact or Shopify Plus Support.

Is this upgrade kit helpful? Submit your feedback.