



Upgrading to Checkout Extensibility - FAQs

When should I plan my upgrade to Checkout Extensibility?	You can migrate to Checkout Extensibility anytime, from now to August 13, 2024. The sooner you start, the better, as you will start getting access to an improved platform and feature set with Extensibility. We have Branding API, Checkout UI Extensions, Shopify Functions, Web pixel app extensions, and post-purchase extensions available today. If there are more features that you need, keep an eye on our public roadmap for dates when new APIs and features are coming available.
If I make the investment to change to the new system, how can I be sure it's not going to change again in a few years?	At Shopify, we're continually investing in making commerce better for everyone. We've spent years building checkout extensibility to ensure it's the best possible solution for our merchants and can assure you it's the platform of the future for customizing checkout. We'll be continually launching new capabilities and APIs so developers can build apps that solve more and more use cases for Shopify Checkout. Now it is faster for us to support these because they have well defined inputs and outputs.
When will you release One-Page Checkout?	We are actively conducting testing on the one-page checkout, and want to secure clear signals on performance metrics before rolling out widely to the broader merchant base. You can expect more merchants to start getting early access to the one-page checkout in June and a wider number of shops getting access in July.
With Extensibility, can you block people from checking out with a PO Box?	Yes. We do support validation of fields using checkout extensibility and can block checkout based on field content or other signals from the checkout. We have a couple tutorials for doing this using checkout UI extensions which are available right now, and then we also have a function in developer preview specifically for checkout validation function on the server side
How do I connect Google Tag Manager to my checkout?	<p>You can create a custom pixel to send events to Google Tag Manager. We created a tutorial to help you out - Custom Pixel for GTM.</p> <p>As an added bonus, custom pixels will load across your storefront, checkout and thank-you page. Here are the available standard events.</p>
Is there a way to make conditional content in checkout based on customer inputs? For example, adding a Prop 65 warning only in the shipping state of California?	Yes. The standard API provides shipping details that can be used to programmatically display or not display UI for an extension. You could then target a specific line item to render the warning underneath it. Here is a link to our Developer documentation on this.

For my store, a discount is applying on the sale price instead of the total cost of the order. Could you please suggest some apps that show discounts on order price instead of sale price?	<p>It sounds like you're using "compare at" pricing to run sales pricing.</p> <p>One way to prevent discounts on already marked down prices is to create an automated collection where the condition is that the compare-at price is empty. You can then create a product discount that only applies to products in that collection.</p>
Is the 'dev' mode connected to any particular store?	Yes. You get to select the store the first time you run 'dev' for an app. It needs to be a development store associated with your Partners account.
What theme are you using on VSCode?	It's the SynthWave '84 VS Code theme. Here is a link to it . The terminal theme is the Free Ukraine oh-my-posh theme (see here).
Is there typescript support in Shopify Functions?	<p>Yes. When you create a new function, you get the option of TypeScript or 'vanilla' JavaScript. The Shopify CLI will transpile your TypeScript for you.</p> <p>Both TypeScript and JavaScript make use of generated types to make working with Functions easier. In JavaScript, we make use of JSDoc comments to annotate types.</p>
Is there a way to see Shopify Function logs?	<p>Yes, you can see logs for all the runs of your Function on a development store from the Shopify partner portal. For production shops, you can see error logs that you've shared from your stores with partners as well.</p> <p>We're looking to expose those logs to Shopify CLI to make them more accessible. I also want to recognize that logging into 'partners' at a merchant may seem odd, and we are working on the branding of that portal.</p>
As far as I tried, I can create a blank custom app on Shopify Admin. But what's the difference between building an app from CLI and from Shopify Admin?	Both are valid ways of creating a new app, but if you create via the CLI, you get a starter template right away as well. If you create an app from the admin, you can always link your app from the CLI to it later.
Is there planned functionality for placing checkout extensions in different locations on mobile/desktop?	No plans for this right now, although I will mention that one reason we have heard this request is specifically for extensions which might show in the order summary on desktop and might be rolled up inside the collapsed order summary on mobile. We do have plans for creating a checkout setting allowing you to decide whether to expand the order summary by default on mobile.

How can developers get their app listed in the collection?	The app store review team are periodically reviewing these for usefulness and quality of experience and including a mix of apps with different capabilities. Soon, we'll be moving to just including all public apps in this list and away from the more manual process we have today.
Are there plans to make spinning up new custom apps for our proprietary corporate needs a simpler process for checkout UI/functions? Will we someday be able to install our custom apps on more than one of our corporate sites?	Yes. We have work in-progress to enable 100% serverless Shopify apps which don't require hosting, as well as enabling installation of custom apps across all stores in an organization. More generally, we're looking at a number of opportunities to make creating and deploying apps easier.
How do we add a custom order attribute box?	<p>Custom fields are well supported on checkout extensibility, and there are a number of tutorials on our developer docs, as well a number of public apps which provide this functionality. Custom fields can be saved to order meta fields, order notes, line item properties etc. We also support a bunch of different field types.</p> <p>We also recently released new components for datepicker, which you can explore here.</p>
Can apps be used to create custom dropdowns? We have stores utilizing store address, store number and division number dropdowns to select from in the ship to sections.	Yes. You can configure pick up options (such as store locations) and we have extension points for checkout UI extensions that are coming soon which will allow you to add custom UI for your pick up locations so you can customize based on your needs.
Can you demo a working Cart/Checkout transform function?	The cart transform is a new API that allows partners to write functions that take the contents of the cart and either merge line items into a bundle, or expand a single line item into component parts. It also allows you to configure a price reduction. I don't have a demo set up for today's session, but the API is in the bundles developer preview and will be moving to broader access in the coming weeks.
Can I move the location of the discount entry box?	No, we do not currently provide a capability to move the discount code entry box. We are always accepting feedback and considering how we can make this more flexible.
With the current use of themes, we have theme settings to update things like borders, padding, and customizing input fields. Using extensibility we only seem to have very minimal brand settings. Are there plans to add more to settings?	You can customize the styling of numerous elements beyond what's available in the checkout editor by accessing the branding API. Browse the advanced checkout branding section in our documentation to see what's currently possible. You can change things like border radius and styling, padding, and other form styles. We're continuing to expand branding capabilities as well. Please share any feedback you might have on missing functionality on our public Github discussion board, or via your MSM.

Can we have the access to update the checkout elements to display Shipping amount or tax amount as "0" on frontend?	Being able to alter the appearance of the order summary is something we are considering, but are treating with care since it's important to ensure the checkout maintains cost clarity for buyers. So it's not possible to overwrite the shipping amount to appear as \$0 at the moment.
Can we have the access to write on click events for existing elements like Continue button, Payment radio buttons?	Checkout UI extensions support subscribing to the buyerJourney which calls a function you define as the buyer progresses through checkout. For example, when the buyer continues from the information step to the shipping step, you can execute custom code. You can even prevent the buyer from progressing to the next step and provide a validation message. I encourage you to look at our client side validation tutorial on shopify.dev for more information.
We use Fetchify to validate email, phone number and address. I have contacted the provider and they said that they are not able to build their own app yet, because some of the features from Shopify were not released yet. When is it on the roadmap?	This is one of our highest priority roadmap items. The missing capability right now is the ability to write to the address field. This is something in active development and will be released in the coming weeks. We are working with some of the biggest providers in this space such as Loqate, Fetchify, Roboturk and others to ensure this use case is well supported on extensibility.
Is it possible to add products to the checkout preview while developing an extension?	Yes, typically you will be running your extension using yarn dev and this gives you a real checkout on your partner test shop. You can add products to checkout as long as products exist in your shop. When you are ready to test in a test store, you can deploy your extension and use it. Within the checkout editor preview you should be able to add a product as well, assuming you are running a query against the storefront API to dynamically retrieve products.
Is the app able to identify if a product is on preorder and put a message saying the delivery dates?	Checkout UI extensions can read the items in the cart and whether they have selling plans, and so you could render something conditional on the presence of a selling plan on the product.
How does PHP support appear when customizing the checkout process?	Checkout extensibility does not support customizing your checkout with PHP. Checkout UI extensions support JavaScript including vanilla JS and react.
How should I create an Amazon-like experience in the checkout with shipping options at the item level, not the order level like it is now?	This is something on our roadmap but is not specifically part of checkout extensibility. We call this "split fulfillment" and it's something we intend to make available to all merchants but work is starting on this in the second half of this year.
Is this checkout extension compatible with hydrogen stores?	Yes. Checkout extensibility applies to the same checkout that hydrogen or other headless storefronts use.

How can I install a checkout app for my live store without creating a public app?	Shopify apps don't need to be public apps. We also support single-merchant apps, also known as custom apps. When you create a new app you have the option of choosing a distribution method for it, and the custom app install link is one of these. Learn more here.
For the look and feel of the page, what ways do we have to customize the layout?	We will be offering controls on spacing in the coming months as well as checkout specific settings that work for common business logic (e.g. shipping and billing being the same address). Otherwise, the checkout section and tab order we have today is based on extensive research on what the highest performing and converting checkout can be, and so changing the order of core fields will not be possible.
Is there any way at all to use a buy x get y discount (ie free gift) with first subscription orders? We're on the Recharge checkout integration.	Subscription or 'selling plan' information isn't available to Function APIs today, but I'm happy to report that this is coming soon, which should allow you to apply discounts specific to orders that include a subscription.
What would be the best way to get assistance/give feedback?	<p>Please direct any feedback or questions on UI Extensions specifically to our Github discussion board. We also recently reinstated our Discord AMAs.</p> <p>Please contact Shopify Support for direct assistance.</p>
As a newer dev who was pretty overwhelmed by what was shown in the tutorial, where should I focus my efforts to build apps for checkout extensibility?	This was definitely a lot for a new developer. We were using multiple building blocks, including a UI Extension and a Function, and then the merchant configuration as well. I'd recommend starting with a basic use case with one of these. If you're a front-end developer, maybe start with a UI Extension that proposes some upsell products. If you're more of a back-end developer, maybe start with a Function that offers a simple Volume discount. And what's great is that for both of these, we have tutorials on Shopify.dev that walk you through this step-by-step.
Are there any apps that allow the customer to check out with a purchase order # for Free? Without a discount code?	For B2B customers, you can configure particular company locations to checkout to draft rather than requiring payment at checkout. You could then review the draft order and reduce the prices (down to 0) if appropriate.
Can this access the customer tags to provide a discount or shipping option?	Yes, customer tags are available to these Function APIs.
Is it possible to deploy a single Shopify Function at a time? When I try to deploy from inside a specific Function, it tries to deploy all of them together.	Our app model does require deployment of the app as a whole. We are moving in the direction of your source code being the source of truth for the state of your app. That said, we are looking at options for making your deployment more efficient by only pushing Functions that have changed.

Are there plans to make spinning up new custom apps for our proprietary corporate needs a simpler process for checkout ui/functions? Will we someday be able to install our custom apps on more than one of our corporate sites?	Yes. We have work in-progress to enable 100% serverless Shopify apps which don't require hosting, as well as enabling installation of custom apps across all stores in an organization. More generally, we're looking at a number of opportunities to make creating and deploying apps easier.
Is there any planned solution for the ability to manage more complex multi-discounting from shopify functions running at the same time that will show the appropriate net discount for each line item based on which function rule was used?	Yes, this is on our roadmap. For this feature, we'll likely add an "all" option to the discount application strategy and leverage the "message" field on the discount output to show which rule was applied. On order discount functions You can leverage the excludedVariantIds on OrderMinimumSubtotal for items that won't qualify towards a minimum spend. You can also use excludedVariantIds on the OrderSubtotalTarget to prevent the discount from applying to a particular variant.
Interested to understand the differences of discount combination and shipping discounts on current version and with new checkout extensibility	Assuming "current version" means "Scripts" here. There are currently some gaps in functionality between Scripts which we document in the migration guide. For example you cannot reject discount codes via functions and we don't currently have the ability to stack discounts. Shipping discount functions are still in developer preview so the only existing Shipping Discount is the built in Free Shipping Discount Code ability.
Can you do checkout field validation client-side without having to press the "continue" button? Like after typing something on an address field, checking for some input and showing the error promptly.	Yes, Checkout UI extensions can render an error as the buyer is entering data. They can then use the buyer journey intercept API to block the buyer from advancing before fixing the error. If you're interested in address validation / autocomplete, we have an API for that coming in the next month. There's also a client side validation API in developer preview that lets you target a native field.
Are these discounts available for all customers of the store? Is it possible to make automatic discounts different depending on the current customer?	We currently cannot restrict automatic discounts based on customer segments. However, in automatic discount functions you have access to BuyerIdentity so functions can apply some custom logic to restrict customers.
Do the custom checkout fields map to cart/note attributes? So if values are passed to checkout with existing cart/note attributes those values can be edited in checkout? For example, gift messages where the customer wants to make an edit.	Yes, you can apply changes to the cart's note and attributes as well as react to any changes to them using Checkout UI Extensions.

What happens if the variant you selected is out of stock?	For simplicity, this example doesn't necessarily handle all edge cases – you may need additional logic in place to hide the offer based on stock, or other business rules.
Any comment on the performance difference between rust and JS? My understanding is that JS is quite a bit slower.	JavaScript functions can be 2-3x slower. For most simple use cases, this means 600 microseconds instead of 200. But there may be some use cases, for example cryptography algorithms, which aren't possible today in Functions within our performance constraints. We're working on some various options to improve this, but if you are comfortable in Rust, or want to learn Rust, it will likely always be faster to some degree.
How do I get the function's logs?	Logs including both the input graph, output of the function, and any errors if encountered can be found for dev shops within your Partners Dashboard. Production shops will currently only log errors.
Is it possible for the custom input field to do the validation via an external api?	The answer to this depends on what kind of validation you are doing. Checkout UI extensions can make calls to an external API to perform validation. Cart and checkout validation functions, which run on the server side of Shopify, do not currently support the ability to make external network calls as part of the validation. Regardless of which method you are using, it will generally be simpler and more efficient to perform validation against local data such as metafields where possible.
How do we leverage language localization when building checkout UI extensions? Would this still utilize the checkout language settings?	The checkout extension platform supports translation of extension content. Whether those particular translations are available for your markets will depend on what apps you are using. If you are building a custom app, it will certainly be possible to provide translations of app content for those markets. UI extensions are provided access to the language settings of the current checkout session.
Are there plans to create more checkout UI components?	Yes there are. We released datepicker and disclosure recently and will be releasing datefield shortly. For more, you can checkout our public roadmap on shopify.dev for any upcoming releases. Or send your requests over to our github discussion board
Is it possible for the UI extension to customize the options shown inside a default dropdown inputs i.e. Regions	<p>Checkout UI extensions can't be used to change native dropdowns, but custom field drop downs can be added with custom options.</p> <p>Delivery and payment customizations are not dropdowns, but using either of these extensibility features allows you to add/hide options that are displayed to the user.</p>
What about email validation?	<p>Both Checkout UI extensions and validation functions have access to the customer email address, so they can certainly perform a validation against that value.</p> <p>Note that extensions cannot write to the customer email address field, but they can suggest changes and optionally block progress until they are addressed.</p>

Will Global-E Native still function as normal after upgrading to Checkout Extensibility? Without requiring additional customizations?	Yes, we are working with the Global-E team very closely to make sure that there is compatibility with Checkout extensibility. There is dev work already in works on both Shopify and Global E side. You can expect to see availability for this, starting Q3 of this year.
With the new extensibility options, is it possible to run a promotion where a customer enters a coupon code and a specific product is automatically added to cart?	<p>Yes. You have the freedom to add, remove, or update anything within the cart depending on any business criteria you can dream up.</p> <p>Your UI extension can react to any changes within checkout, added discount codes are just one kind of change.</p>
Can you remove products from the "cart" in the checkout?	Yes. The cart lines API allows you to add, remove and update cart lines during checkout. The example app that Nick walked us through today used this API to add the upsell product and the gift card.
How can discounts be aware of each other when they are generated by different apps/methods?	<p>On discounts UI, we can see discounts that were created via native or functions apps.</p> <p>Other than that, a function discount is treated the same as a native discount.</p>
Would love to see Shopify build an official version of this for Shopify Functions: https://jgodson.github.io/shopify-script-creator/	Jason's tool is a great one for Scripts. The development model is a bit different for Functions: You can have multiple Functions active, they're app based, etc. So it'd be difficult to create a 1-1 tool. That said, if you don't want to create an app yourself, there are some great public apps which provide a rule-like interface for configuring payment, shipping, and discount functions.
Is it possible to access line item stock and hide shipping method or pickup method base on that logic	The recommended way to hide delivery methods is using Delivery Customization Functions.
How does the express/react part actually get deployed for production? Is there documentation on how to do that?	<p>Checkout UI extensions are packaged and delivered via apps. This means they are added to a shop by installing either a custom or public app on the shop. If you are building a custom app, deployment is performed using the Shopify CLI tools.</p> <p>The embedded app in today's example would need to be externally hosted, and our docs include tutorials for doing this on Fly.io and Heroku. We are working on removing this requirement for your function UI.</p>
Are we able to build gifts with purchase based on threshold or SKU with the new checkout extensibility?	Yes, that would be a variation of what we built today. You could change the conditions upon which the gift is displayed and the discount is applied based on your business rules. You can add products to the cart automatically as well, but be careful about taking actions the buyer doesn't expect.

Is it possible to create logic based on customer location?	Yes. The shipping address is available to checkout UI extensions. An example of using this might be presenting a banner based on the shipping address.
We have pickup up on the checkout, are we able to change the customer's shipping address if they choose store pickup from our custom popup?	Our recommended approach in this case is to use Shopify's built in Pickup in Store feature. When this is enabled and the buyer selects it, they will not be required to provide a shipping address. While Shopify's Pickup feature has some limitations today, such as not being able to provide your own pickup locations, this is an area we intend to offer further customization options in the future.
Follow up on discounts: If multiple discounts are running, is there an order or hierarchy in which they run?	<p>We try to find the "best possible discount". But we do run the product discounts before order discounts to ensure the order total still meets the qualifying spend.</p> <p>But if the order discount gives a better discount and it's not set to combine it will reject the product discounts.</p>
Can you please elaborate on the Discount combinations? Can you stack multiple promos (automatic promo and manual with promo code)?	<p>First, a bit of clarity on Combination vs Stacking. Combination refers to which discounts can be added to the cart and stacking refers to having multiple discounts on the same line item.</p> <p>The product + order stacking that's planned for Summer editions will allow adding an order discount on top of any product level discounts that were already applied to the order and distributed at the line level. This can be either automatic or code based discounts and native or function discounts. Order + order is more likely to be in an early access beta but it will allow stacking of many order discounts on top of each other.</p> <p>On order discount functions You can leverage the excludedVariantIds on OrderMinimumSubtotal for items that won't qualify towards a minimum spend. You can also use excludedVariantIds on the OrderSubtotalTarget to prevent the discount from applying to a particular variant.</p>
Interested to understand the differences of discount combination and shipping discounts on current version and with new checkout extensibility.	<p>[Assuming "current version" means "Scripts" here]. There are currently some gaps in functionality between Scripts which we document in the migration guide. For example you cannot reject discount codes via functions and we don't currently have the ability to stack discounts.</p> <p>Shipping discount functions are still in developer preview so the only existing Shipping Discount is the built in Free Shipping Discount Code ability.</p>
Do you need to be on Shopify 2.0 for the new checkout?	Checkout extensibility works equally well for all kinds of online stores. You can be using original Online store themes, Online Store 2.0, or even completely custom storefronts using the Hydrogen framework.

Will Shipping Discount API turn into prod?	This is currently in Dev Preview. Our priority is to roll out discount stacking first as that is the top priority for brands. After we've rolled out discount stacking, we will focus on making shipping discounts generally available.
Are there plans to make spinning up new custom apps for our proprietary corporate needs a simpler process for checkout ui/functions? Will we someday be able to install our custom apps on more than one of our corporate sites?	Yes. We have work in-progress to enable 100% serverless Shopify apps which don't require hosting, as well as enabling installation of custom apps across all stores in an organization. More generally, we're looking at a number of opportunities to make creating and deploying apps easier.
Will we have a dedicated session for discounts combination and shipping discounts?	We do intend to have another discord AMA on Functions in June where you'll be able to learn more about discount Functions.
How can we migrate our discount scripts?	<p>It's recommended to start migrating to Shopify Functions with delivery or payment customization Shopify Functions-based apps, and using them along with line-item discount scripts. When you're comfortable using delivery and payment Shopify Functions, you can then progress to using discount Shopify Functions.</p> <p>Please note that Shopify Scripts will execute after the Function. As a result you should test your Scripts and Functions combination before implementing it in a live environment.</p>
Is there any chance to apply volume discount or tiered discount on Checkout extensibility? Currently we have to use an app or develop a custom app for ourselves	<p>Currently we don't have a native way to configure volume / tiered discounts. And as of now, that feature is not in our roadmap.</p> <p>As you noted, volume discounts can be created with Discount functions. You can find public apps in the App Store today, or your dev can create a custom app with tiered or volume discounts.</p> <p>Please use the developers/API docs for reference.</p>
Does the Checkout Extensibility mean I have to ALSO upgrade to Shopify Functions as currently some of the API Calls needed are not existing in Shopify Functions.	<p>No you do not have to migrate to Functions at the same time. Shopify Scripts will continue to work alongside Checkout Extensibility.</p> <p>It's recommended to start migrating to Shopify Functions with delivery or payment customization Shopify Functions-based apps, and using them along with line-item discount scripts. When you're comfortable using delivery and payment Shopify Functions, you can then progress to using discount Shopify Functions and plan a full migration.</p>

Is there a way to access the cart item's weight? I have a validation to check a certain total cart weight.	<p>Yes! The weight of items in the cart is available to the input query that is passed into the validation function. In the backend validation function:</p> <p>Starting on the Cart object:</p> <pre>`lines.merchandise.weight` and `lines.merchandise.weightUnit`</pre> <p>If you want to do front-end validation ("buyerJourney" extension API + intercept), you can fetch the weight using the Storefront API from the extension.</p>
Is it possible to add a different quantity of the gift item depending on the quantity of eligible items in the cart? Also could it be added automatically or only by the user clicking the add gift button?	Yes. Your extension can add any number of line items to the cart, and the extensions' business logic can use other information about the cart to decide what to add. An extension can add items without a specific buyer action, although it is important to preserve buyer trust and not make unexpected changes for the buyer (for example adding an item that will increase the total cost).
Do other product discounts have to be set as combinable for the free gift discount to combine with them? I would want the free gift to combine but not for other discounts to combine with each other.	Yes, they would have to be set as combinable. We currently don't have fine-grained control over exactly which product discounts can combine with each other but this is something we're looking into.
Do you have any recommendations to migrate a CI/CD pipeline that used to use themekit deploy from the old checkout.liquid file?	All customizations with checkout extensibility happen through apps. If you are building custom extensions for your store, whether they are functions or UI extensions, you will be deploying them through a custom app on your store. The Shopify CLI can be used in a CI pipeline to automate deployment of new versions of your custom app to your store.
Are these discounts available for all customers of the store? Is it possible to make automatic discounts different depending on the current customer?	We currently cannot restrict automatic discounts based on customer segments. However, in automatic discount functions you have access to BuyerIdentity so functions can apply some custom logic to restrict customers.
Is there something special that needs to be done for Cart Transformation (bundles) functions to work?	This function is currently in developer preview but will be made generally available soon.
Can I bypass the shopify app store verification process if it's a custom app for one store, for checkout extensibility apps?	Custom apps are not submitted to the app store, and can be deployed directly to your shop without any approval process.

Can we track purchase events that occur via Shop Pay?	Yes, both App pixel and Custom pixels capture Shop Pay events.
How do loyalty point redemptions work with discounts are they treated as payment or applying a discount code to replace other product/order level discounts?	There's several of them out there so I can't speak to the details of all of them, but most loyalty point apps I've seen are treated as a discount code that are restricted to a one-time use. When stacking is released this means it could be applied on top of other product discounts if desired.
Will Extensibility work with the Script Editor? Script Editor is also sunseting our custom vendor rules so we are plotting the attack of B2B and Extensibility.	You can continue using Script Editor during your upgrade to Extensibility, up until its removal in August 2024.
Can you confirm that AdditionalCheckout Scripts will continue working, so the only thing we need to change is custom code in checkout.liquid?	<p>In the long term, customization on these pages will use the same checkout extension technology as the rest of checkout. We released a developer preview of extensibility on the order status page earlier this year. This will be available on production shops starting in July of this year.</p> <p>Meanwhile, Additional scripts on the TYP and Order Status Page will continue to work.</p>
Currently we do client side validation by rendering in an extensionsPoint. Is it possible to validate without rendering anything? not to mention, return null	Yes, while client side validations are typically used in combination with rendering custom elements, this is not a hard requirement. You could add a UI extension that does not render and only performs validation. However, I'd encourage you to consider cart and checkout validation functions for this use case, since it is designed for exactly this scenario.
Some apps use draft orders when checking out. Are functions able to execute on these draft orders as well?	Discounts are locked so they don't apply on checkouts coming from Draft Orders. I don't believe there's the same restriction for other APIs but it's best to confirm with the other areas.
Are there any plans or options to have separate delivery options based on inventory availability.	So this is something we refer to as "split cart" meaning the ability to have different delivery methods for different items in the cart. This is something that is on the checkout roadmap generally but is not available just yet.

What is the enhanced security?	<p>Checkout UI extensions run in a sandbox, which is an iframe and a Web Worker. This means no extension can intercept payment details or other sensitive data from the checkout (they can't access HTML). If the extension wants to access customer data, they have to submit a request and go through an approval process. It's not possible for one extension in the checkout to intercept data from any other extension.</p> <p>Checkout UI extensions run sandboxed in checkout, meaning they are isolated from other customizations. This means that any compromise of a given customization or partner cannot affect others and can be easily isolated for action.</p>
Interested in knowing more about Pre-orders.	<p>Pre-orders allow buyers to place an order for products or variants that haven't been released yet. You can learn more about selling plans which power pre-orders in the Shopify Dev Docs. One interesting use of checkout UI extensions we recently saw was an extension which offered a pre-order option in checkout for a product which was sold out. This would not have been possible to achieve previously with checkout.liquid.</p>
Will checkout extensibility be fully embedded with Global-e?	<p>Yes, we are working with the Global-E team very closely to make sure that there is compatibility with Checkout extensibility. There is dev work already in works on both Shopify and Global E side. You can expect to see availability for this, starting Q3 of this year.</p>
That Pixel implementation is no small feat, and choosing the correct setup is very difficult. Shops that use GTM will go through a very strange debugging process.	<p>We acknowledge implementing a custom pixel can be challenging. As such, we are making improvements to make the implementation of a custom pixel more user friendly like auto-complete in the code editor. We made a tutorial to implement Google Tag Manager with custom pixel.</p> <p>In the near-future, we will be publishing pixel templates for popular integrations like GTM.</p>
Is it possible to have the old checkout.liquid and the new checkout draft in parallel in one test shop?	<p>Yes. When you begin the migration to Checkout Extensibility, you will have a draft version of your new checkout. You can edit and preview this new checkout using the "Customize" link on that draft in the checkout settings page. Meanwhile, your store will continue to use the checkout.liquid customizations until your new draft checkout is published.</p>
One of our merchants wants to hide the billing address as this is not currently possible with their ERP system. With checkout.liquid we do this with css. This is not possible yet with extensions?	<p>I can confirm that this is unfortunately not yet possible with checkout extensions. Hiding built in checkout fields is certainly a use case we are aware of, and we are exploring options for this. Follow our public roadmap for any developments in this space.</p>

Does checkout extensibility work with Shopify Scripts?	Yes, you can continue to use Shopify Scripts with Checkout Extensibility. However, Shopify Scripts are also deprecated, and the functionality they provided is now available through Shopify Functions. We recommend upgrading to Shopify Functions for these use cases.
Please go more in depth for tracking customer behavior (installing app pixel)?	App pixels are pixels that are installed automatically when you install the app from the Shopify app store. Custom pixels allow for more customization if required for your set up. We have a tutorial to install google tag manager with a custom pixel
Is it possible to add order notes to the checkout?	Yes, checkout extensions have access to an API that can be used to add or edit order notes.
Is there any documentation for how headless storefronts should handle this?	Headless storefronts allow you to customize product pages and the cart but as soon as the buyer checks out, they'll enter the Shopify hosted checkout. A headless storefront doesn't need to do anything special. The merchant can use checkout extensibility just like theme based shops.
If we don't have checkout.liquid currently, do we need to do the pixel thing that was just shown?	If you don't have any custom code in checkout.liquid, you do not need to modify code as shown in the intro. You can install an app pixel by installing an app to your store and/or using a custom pixel.
Do carrier calculated services shipping apps need to be installed through checkout extensibility?	No - Carrier Services API requires no changes to work with checkout extensibility.
Pixel Manager is very difficult to test if your store is using GTM. The Preview mode is broken due to the "sandboxing" of the Customer Events.	We acknowledge that testing pixels is challenging. The new google debugging tools do not work with a sandbox as you mentioned. However, you can use their legacy chrome extension that works. It is also mentioned in these docs at the bottom under Testing and debugging .
Do we need to use GTM in checkout extensibility or is the new GA4 tracking via the Google app sufficient?	We recommend using app pixel, like from the GA4 app because it will give you maximum data and it can be upgraded automatically by google. Whereas, a custom pixel requires your maintenance.
We have 6 languages / markets. Will all the custom fields in check-out be translated into these languages?	The checkout extension platform supports translation of extension content. Whether those particular translations are available for your markets will depend on what apps you are using. If you are building a custom app, it will certainly be possible to provide translations of app content for those markets.

Can we connect the checkout theme to Github or another code repository?	This is not possible in the same way that you might export and import theme files. However, all the customizations through checkout extensibility are either delivered through apps, or through use of the Branding API. Apps can be installed on multiple shops, and the branding API can be used to apply the same change on multiple shops.
How can we add third-party chat to checkout extensibility?	Third party chat is currently on our roadmap for Q4 of this year and we will add them to the public roadmap as we get closer to working on implementing the APIs to support third party chat.
How do I know if I already have Checkout Extensibility? I've recently upgraded my themes and do not have checkout.liquid.	You can navigate to the settings section of your admin, under the "Checkout" section. If you have access to checkout extensibility, you will see a "Checkout customization" card and should be able to customize your live checkout. You can have a look at the documentation here . (If you don't see this card, you may be using a feature that is not compatible.)
Can you please let us know how we can write custom javascript code?	Checkout UI extensions would be written in JavaScript but since they run in a WebWorker they can't query the DOM or make direct changes to HTML or CSS. The extension can use standard API methods to make changes to the checkout. Merchants can use the branding API to customize the checkout look and feel
Is it still possible to have customer "logic" based on customer "tags", like free shipping for certain customers?	This can be handled via built-in discounts using customer segments. Create a customer segment in the customers section of Admin and then assigning it to the discount either via the Admin or via GraphQL DiscountCustomerSelectionInput
We currently use Shopify Scripts to block certain shipping options based on address. Can we still do that with extensions and if not is it on the roadmap to conditional shipping option display?	Yes. Once released, you'll be able to add logic for shipping discount applicability based on address.
Will we be able to write custom javascript using checkout extensibility? Will we be able to write custom javascript using checkout extensibility? (continued)	<p>Checkout UI extensions are the way that you can write custom javascript for your checkout. Kumar mentioned in the security answer that Extensions are run inside a sad box and are not able to manipulate the DOM. You won't have access to the full set of browser APIs or the window object but do have access to most javascript features from your extension such as <code>fetch()</code>.</p> <p>You can write custom javascript to run in your checkout using UI extensions and interact with the checkout by using any of our extensibility APIs.</p>

Do I need to upgrade my online store to Online Store 2.0 to use checkout extensibility?	No you do not. You can access checkout extensibility regardless of whether you have online store 2.0 or not. Checkout extensibility works equally well for all kinds of online stores. You can be using original Online store themes, Online Store 2.0, or even completely custom storefronts using the Hydrogen framework.
How do I know if I am already on Checkout Extensibility?	If you go to the admin and your checkout settings, you should see your live checkout profile and can click “customize” to see the checkout editor where you can make branding changes and add apps to checkout.
How can we implement our custom javascript?	<p>Checkout UI extensions are written in JavaScript but since they run in a WebWorker they can't query the DOM or make direct changes to HTML or CSS.</p> <p>This means no extension can intercept payment details or other sensitive data from the checkout as they can't access any HTML elements. If an extension wants to access customer data, they have to submit a request and go through an approval process and it's not possible for one extension in the checkout to intercept data from any other extension.</p> <p>The extension uses a standard Checkout UI extension API methods to make changes to the checkout.</p>
Can I create a custom pixel for Google Tag Manager?	Yes. We have a tutorial that will help, but all available customer events can be pushed to Google Tag Manager with a custom pixel. We are adding more event data each week
What is the roadmap for the branding API? At the moment not everything is possible that we currently do for our merchants.	<p>We just released support for custom fonts and are actively working on finer applications of color for the API, as well as customizations for the header and footer sections..</p> <p>For the roadmap for Extensibility, including the new features planned for Branding API can be found on Shopify Dev Docs. If you have any feature requests, please refer to our Github discussion board.</p>
How can we add third-party chat to checkout extensibility?	Third party chat is currently on our roadmap for Q4 of this year and we will add them to the public roadmap as we get closer to working on implementing the APIs to support third party chat.
It's not possible to enable checkout extensibility features if a store is currently using checkout.liquid. What is the recommended procedure to upgrade and minimize downtime?	Using checkout extensibility, you can create a draft checkout that you can view in the checkout editor, and start customizing this using the checkout extensibility features. You can also click on the “preview” button to see an end to end preview of your extensibility powered checkout. Checkout extensibility will only “replace” your checkout.liquid checkout once you hit publish. If you currently have checkout.liquid you can also revert back to checkout.liquid if you run into any issues.

Would it be possible to integrate the one-page checkout in Native Apps?	Yes - checkout UI extensions and their extension points were designed with the upcoming one page checkout in mind. Apps you have added to your checkout will automatically work with the upcoming one page checkout design.
Does this validate the address to block PO Boxes or is it just a message?	<p>Blocking checkout with validations are 100% supported using UI extensions. You can block checkout based on a field input, or any other condition you come up with.</p> <p>We have also recently released a new Shopify Function for cart and checkout validation which is currently in developer preview and we expect to release to production stores in July.</p>
Can I add text wherever I want?	Extensions are placed at extension points, predefined locations in the checkout ui. You can find documentation for the supported extension points in our Dev Docs . There are also public apps available on the app store to place text, banners, trust badges, custom fields and more using UI extensions.
If I upgrade to checkout extensibility, will this automatically opt me out of any Shopify Scripts I may have running currently?	While the deprecation of Shopify Scripts has also been announced, and will be replaced by Shopify Functions, you don't need to move to Shopify Functions to use Checkout Extensibility. Shopify Scripts will continue to work on Checkout extensibility through August 2024.
How can Google Tag Manager and a custom data layer that uses liquid variables be added to checkout with Checkout Extensibility?	We recommend adding a custom pixel that sends events to Google Tag Manager's data layer. We have a basic tutorial published here . Checkout UI Extensions can also publish their own customer events to the customer events stream.
Are there any impacts to load times or checkout speed using extensibility vs checkout.liquid?	Checkout extensibility is built on the latest platform technology, making checkout interactions and page transitions 2X faster (50% reduced latency), increasing conversion by over 1% on average.
Is there or will there be a template for a checkout extension app available to work off?	<p>Yes. When you create a new extension it's a working version of an extension. We also have a number of tutorials on shopify.dev that delve deeper into some common customizations. There are tutorials available for:</p> <ol style="list-style-type: none"> 1. Custom fields 2. Custom messages/banners/warnings 3. Validation 4. Product offers like gift with purchase or upsell and more!

Does subscriptions upsells mean I can upsell subscription products on the Upsell page?	With a checkout UI Extension you could show an upsell from a one-time product purchase to an upsell, offer that product as part of a subscription, or another thing we have seen possible is offering a buyer the ability to swap an out of stock item for a preorder as Mike mentioned earlier.
Will there be custom CSS support for the checkout UI extension?	Checkout branding is designed to work across all checkout contexts including shop pay, onepage checkout and any app with an extension installed. In order to do this, and have everyone take advantage to new innovations we will not be able to support custom CSS for UI extensions
Do you recommend building a feature like a checkout bundle via custom functions? Or will the new checkout support more apps without slowing down the site?	We recently announced a new function called the Cart Transform Function which allows you to combine multiple items into a single line item, or expand a kit to show its child items in the order summary. This function is currently in developer preview. We are able to pre-load extensions and do much more optimization with checkout extensions.
If we can't make changes to the DOM, can we implement logic based on the entered details? e.g validating a date of birth, or changing shipping options based on address.	<p>Yes, this can be done a few ways: UI Extensions and Cart and Checkout validation functions. Native fields are available to validate with any criteria you need.</p> <p>Future efforts on delivery method extensibility will be able to address many custom logic needs.</p>
Will there be native support for GA4 coming including tracking that's required on the checkout page?	Yes, if you install the Google Channel App, it includes an App pixel that will be loaded on all shopify surfaces (including Shop Pay) and will be available in your GA4 account.
We use custom javascript to validate customers' validity in checkout.liquid based on the input of the email field. Will we be able to achieve the same in checkout extensibility?	Yes, and we'll be able to extend that functionality with Cart and Checkout Validation functions.
What is the best way to test for issues in checkout extensibility?	For testing customizations, it is recommended to use the draft and live checkout features in the checkout editor. You can apply changes in a draft checkout and preview the experience with test data. As soon as you're ready, publishing a checkout makes it live for that particular store
If Checkout Extensibility is published, is there a way to return to checkout.liquid?	Yes, you can work on your draft extensible checkout while having a published checkout.liquid and vice versa
Will apps which control Shipping Methods	Yes. Specifically apps which use the carrier service API still work with checkout extensibility with no

work with Checkout Extensibility?	issues.
Will every Shopify Plus store that I work with need to build their own custom app for handling their unique use cases that were previously run through Shopify Scripts? Or have to pay for a third party app to get what was part of Plus before?	Custom apps are supported in checkout extensibility and should be able to handle unique use cases that do not apply to a broader shop base.
Will the Scripts running in the 'additional scripts' section of Shopify still run?	Yes. Additional scripts will still run on the order status page.

