Installation and Setup guide

The Checkout module for Prestashop™ 1.7 - 9

July 16, 2025

Overview

This guide describes basic installation and setup instructions for The Checkout module for Prestashop.

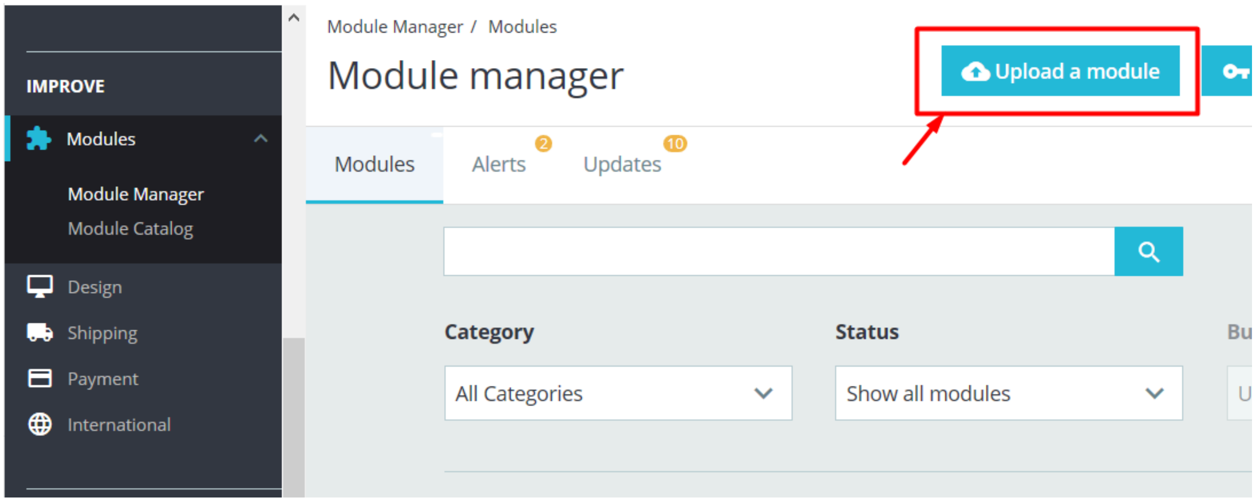
Main goals the module addresses are:

- True One page checkout, with payment methods visible right on page load
- Compact and uniform styles to provide professional checkout form look
- Layout designer - to customize the checkout form look and feel
- Handful of small options to control and fine-tune checkout process

Installation

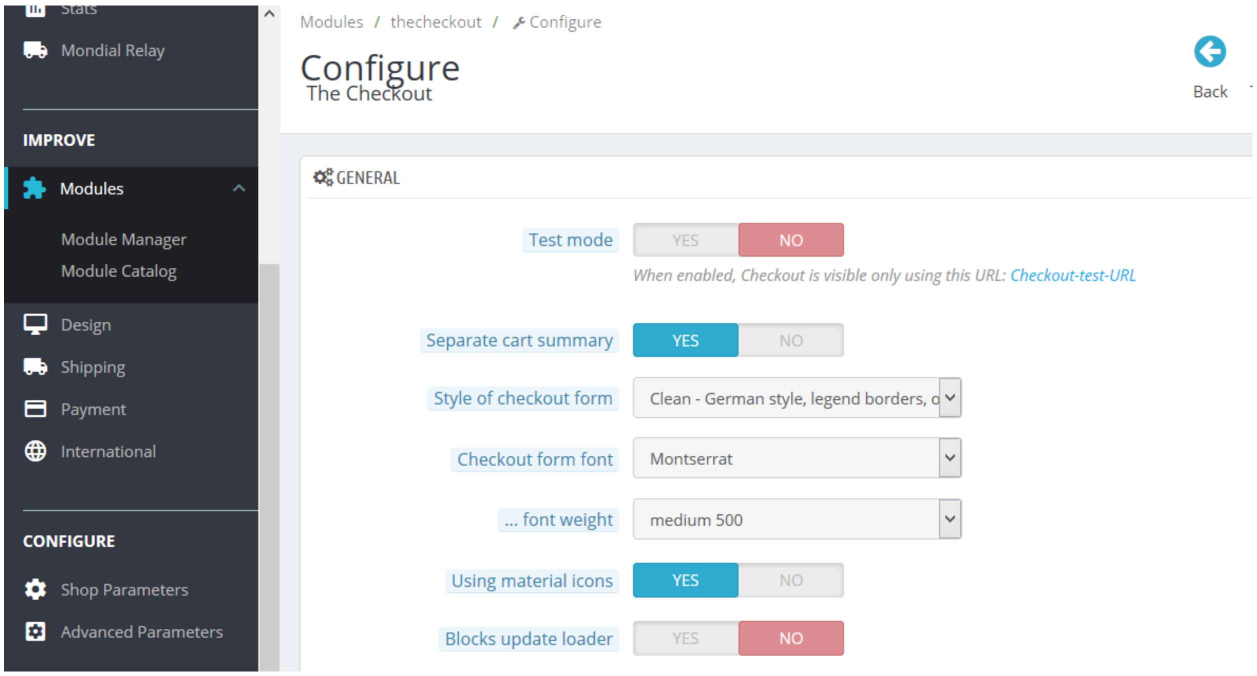
The easiest way to install (any) module is using Prestashop Backoffice only.

Log-in to your Backoffice and go to Modules -> Module Manager. At the top, click Upload a module button. Locate the downloaded file (with .zip extension) and drag&drop the file to popup - module will upload and install automatically.



Configuration

Once installation is done, click the Configure button and you shall be redirected to module configuration page:



Configure
The Checkout

GENERAL

Test mode	YES [✓ NO]
	When enabled, Checkout is visible only using this URL: Checkout-test-URL
Separate cart summary	[✓ YES] NO
Style of checkout form	Clean - German style, legend borders, o
Checkout form font	Montserrat
... font weight	medium 500
Using material icons	[✓ YES] NO
Blocks update loader	YES [✓ NO]

GENERAL section

Notice please the very first option - Test mode - which when enabled allows you to see the checkout module, while your original checkout would still be served to your customers. Thus you can make any tests and modifications without affecting your live traffic.

To see the checkout in test mode, click the link Checkout-test-URL which will open in a new tab and will set a cookie in your browser that will instruct Prestashop to use The Checkout module.

We tried to make all options self-explanatory, but some of them still might need a bit of explanation.

“Separate cart summary” - instructs checkout module, whether to show separate, cart-only page (after user clicks on ‘Cart’ link in Prestashop) before going to checkout form. When disabled, cart page is never shown and customer goes to checkout form (without cart only review)

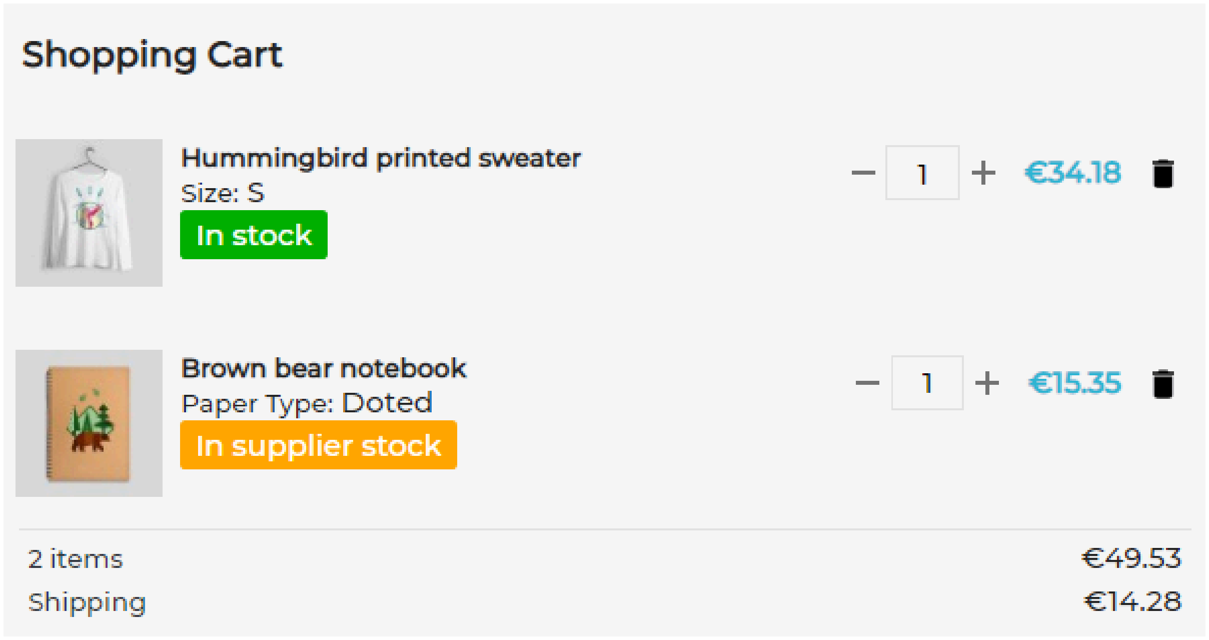
“Style of checkout form” - here you can choose from predefined styles, this is a template for quick start - choose the one that fits best as a starting point for further changes and customization. Name of style here determines which styling file shall be included on checkout. Styling files are located in module folder under views/css/styles.

“Using material icons” - disable this option only when your theme does not use Google’s material icons. Most Prestashop themes use it. You shall recognize that if you see weird looking icons or weird checkboxes and radio buttons on checkout form.

“Blocks update loader” - enables animation loading effect when blocks on checkout form are being updated through ajax background call. Sometimes, it looks better with loader, sometimes without, depends on your theme and design, whether you have more-or-less static pages, without too many dynamic effects, or whether theme is playful with lot of animations.

“Compact cart” - makes cart summary controls smaller, useful if you position cart summary in very narrow column.

“Show product stock info” - display stock information in cart-summary - in stock, on backorder, out of stock, or even available quantities, if customer has 5pcs. in cart, but only 3pcs. are in stock. This is how it looks like in cart summary:



Shopping Cart

Hummingbird printed sweater Size: S In stock	1	€34.18
Brown bear notebook Paper Type: Doted In supplier stock	1	€15.35

2 items €49.53
Shipping €14.28

CUSTOMER & ADDRESS section

"Force email overlay" - with this option enabled, customer enters checkout form and is unable to see the whole form, up until he enters email - which effectively causes guest account creation - which means, you can re-target this customer with abandoned cart reminder. When this option is enabled, it automatically enables also "Silently register guest account" (it's required so that it can work properly)

"Silently register guest account" - this creates listener on email field and whenever customer enters email, The Checkout module immediately creates guest account and assigns it to cart. This is very useful for abandoned cart reminder, where often, we saw an empty cart, but no email attached to the cart, and thus no possibility to send reminder.

[new] "Allow guest checkout for registered" - when enabled, even registered customers are allowed to checkout as guest. In default Prestashop checkout, once customer creates an account, he cannot make guest orders anymore - all orders must be done after logging in. This option removes this PS core restriction.

“Show ‘I am a business’ checkbox” - very useful and often used option, to separate business fields - like company, vat number, company registration, or in Italy - sdi, pec. This way, regular consumers can fill in only their fields, while businesses fill in more details.

“... business fields” and “... business disabled fields” - boxes related to previous option, allows you to set which fields shall be separated as business, and moreover [**new**] which fields shall be hidden for business (and thus effectively allowed only for consumers)

“Offer second address” - shows checkbox for second address (if invoice is primary, it would show checkbox with “delivery to another address”) When disabled, only one address will be shown on checkout form (=simplicity), however, most eshops would want 2 addresses at all times.

SHIPPING & PAYMENT section

“Force customer to choose country” - when enabled, default country is de-selected and customer cannot proceed with order without choosing his country. This is effective to avoid customer errors, where customer would forget to change his country.

“Shipping required fields” - list of fields that are required to be filled in before shipping methods are displayed. This is useful when you have different shipping rates based on postcode, city, country, state, etc. Customer will see shipping rates only after filling-in his address (parts of address you define).

“Payment required fields” - same idea as “Shipping required fields” but applies for payment methods.

“Show "shipping to" in carriers” - displays small text in shipping methods list, which highlights the “shipping to” address (country, state). This is to ensure the customer is choosing the right carrier and has right country selected.

“Show Order Message” - when enabled, textarea box will be displayed (you can change its position using layout designer), that allows customer to enter free text, related to his order - this will be visible in Backend.

“Remove spaces from postcode” - some countries have postcode format in PS core (BO / Localization / Countries -> edit country) defined with inner space, like ‘NNN NN’, e.g. this is valid postcode: 150 01, however, this is not: 15001. But, customers type-in both versions and may get confused, so I recommend - set PS core format to no-spaces and enable this option, so that when customer input 150 01 it will be transformed into 15001 and accepted by PS core.

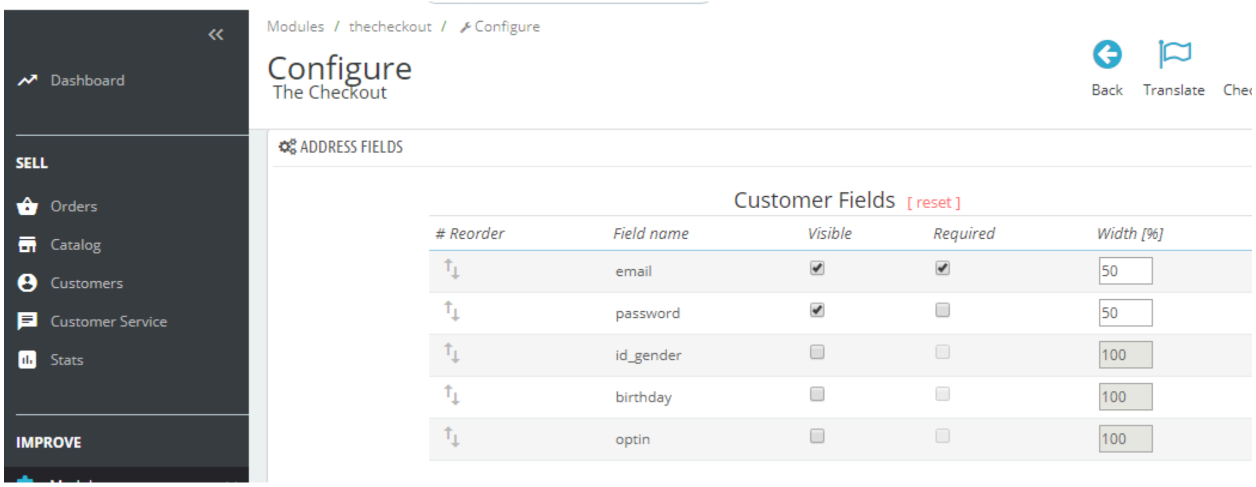
ADDRESS FIELDS section

“Customer Fields”, “Invoice Address Fields”, “Delivery Address Fields” - you can reorder here the fields, set them visible, required, and their width. Please note that width is set as flex-base, so it will apply only when there’s enough room in particular block - when blocks get too narrow, these percentage will fall-back to 100%

“Live” option in Address fields - enable this for fields that affect shipping methods (methods selection or shipping cost). When ‘live’ field is changed by customer, shipping methods are reloaded with newly input information. Typically used with shipping modules that use Postcode or City to show delivery methods or shipping cost - in that case, set ‘live’ for postcode and city fields.

“managed automatically” in fields definition means that this field typically cannot be controlled on fixed basis and field visibility depends on country selection. This is used for state and postcode fields - states are defined for every country, some countries have them, some not, and the same is valid for postcode - some countries have it required, some not.

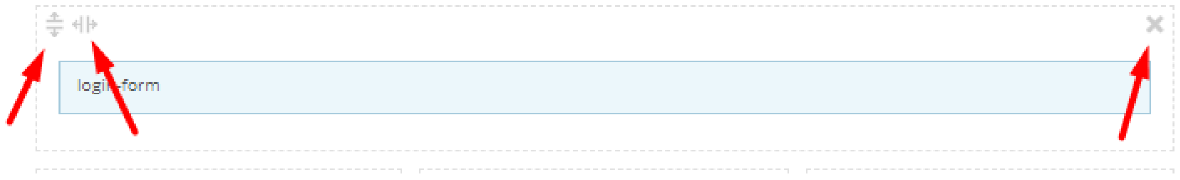
Notice also the [reset] link - this will restore default fields configuration, in case it's messed up. [**new**] In case you have customization or module that updates core Address::$definition, it will be shown here and custom fields will appear at the bottom of Invoice and Delivery address section. After they're visible, reposition them and click Save.



# Reorder	Field name	Visible	Required	Width [%]
⇅	email	✓	✓	50
⇅	password	✓	☐	50
⇅	id_gender	☐	☐	100
⇅	birthday	☐	☐	100
⇅	optin	☐	☐	100

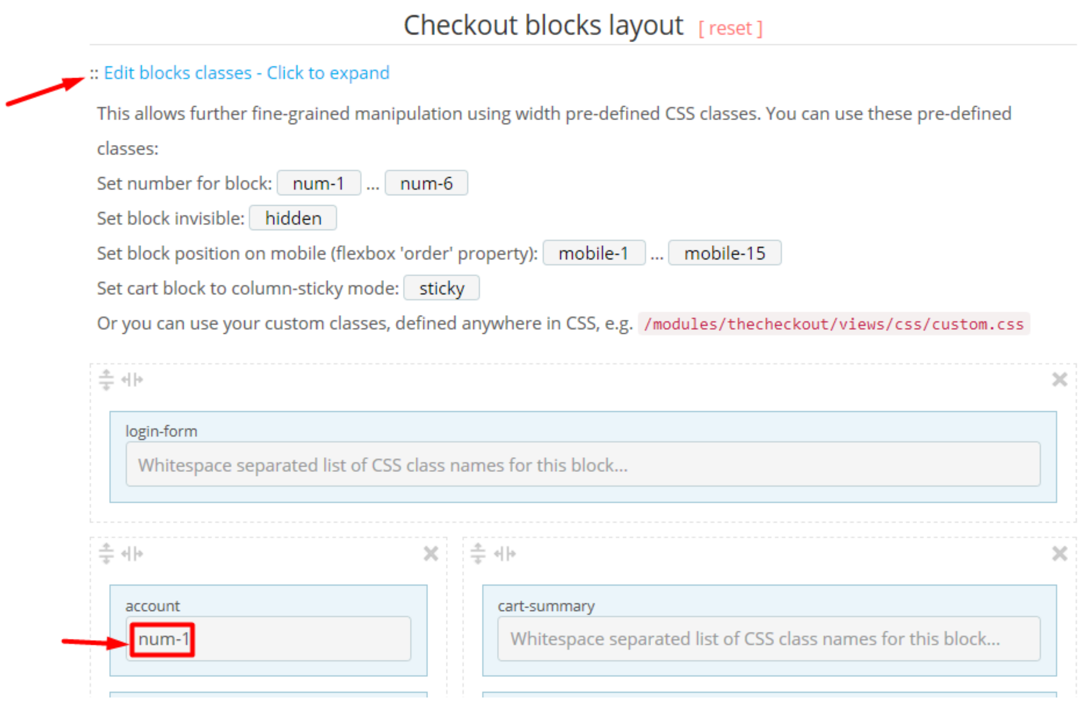
LAYOUT section

"Checkout blocks layout" - this is the main Layout designer for module, here you can leverage unlimited design possibilities by creating new sections. Every section has 3 controls to split it vertically, horizontally or delete section:



With this approach, you can create almost any layout.

Advanced usage: For further customization of blocks look, you can add custom classes to every block, to identify it better and target it with CSS. You can see added classes by clicking on Edit blocks classes - Click to expand:


Checkout blocks layout [reset]

:: Edit blocks classes - Click to expand

This allows further fine-grained manipulation using width pre-defined CSS classes. You can use these pre-defined classes:

Set number for block: num-1 ... num-6

Set block invisible: hidden

Set block position on mobile (flexbox 'order' property): mobile-1 ... mobile-15

Set cart block to column-sticky mode: sticky

Or you can use your custom classes, defined anywhere in CSS, e.g. `/modules/thecheckout/views/css/custom.css`

login-form: Whitespace separated list of CSS class names for this block...

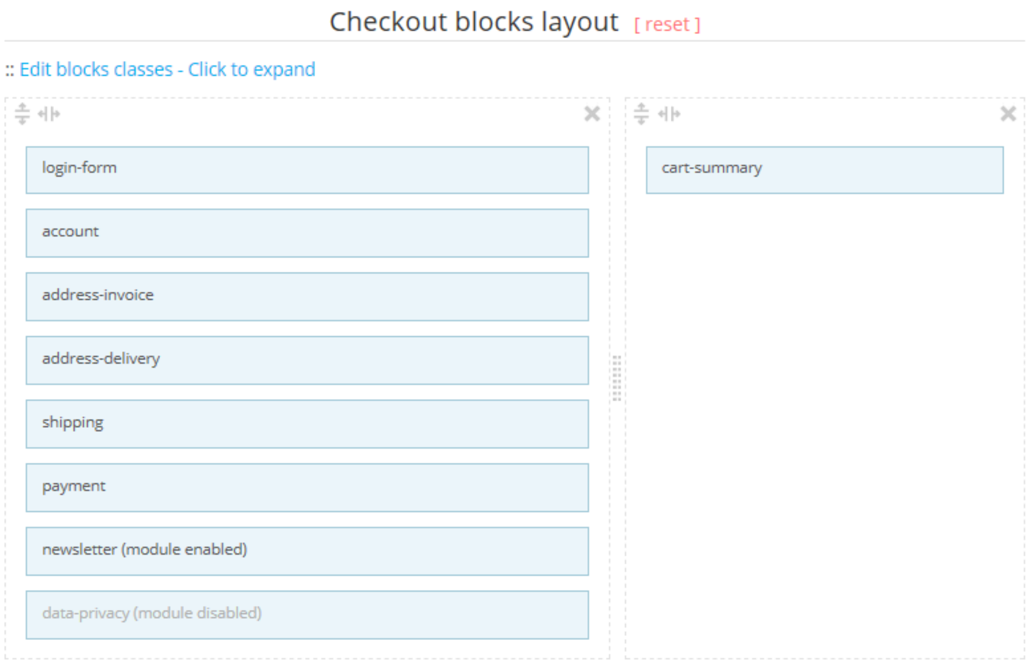
account: num-1

cart-summary: Whitespace separated list of CSS class names for this block...

You can use predefined classes, like num-1, num-2, … which will create block number on checkout form, or you can add arbitrary class name you have in your CSS, or you create with Custom CSS block.

Or, you can set custom order of blocks for mobile devices, with mobile-1, mobile-2, … classes - e.g. set cart-summary block to have mobile-1 class, account mobile-2, ... and summary will be shown first on mobile screen, than account, etc., while keeping their standard position on desktop browsers.

There's also a special class to make block 'sticky' (CSS position=sticky), which typically will be applied to cart-summary block and would cause cart-summary to slide up-down in it's parent block. To use this, please make sure to have only cart-summary in whole section, side-by-side with another section - only then the CSS sticky parameter works as desired, see an example of such configuration:


Checkout blocks layout [reset]

:: Edit blocks classes - Click to expand

login-form

account

address-invoice

address-delivery

shipping

payment

newsletter (module enabled)

data-privacy (module disabled)

cart-summary

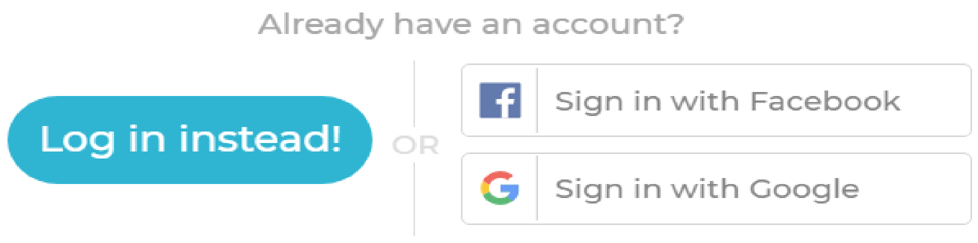
Notice please 3 special blocks - newsletter, data-privacy and psgdpr - content of these blocks is generated by third party modules, so to enable or disable them, please go to Modules and services there locate these modules and enable / disable them there. Only when they are enabled, The Checkout module can control their position on checkout form.

Further down there are 4 HTML blocks, you can use as you wish - adding headers, side columns, footer, etc. As an example, The Checkout module comes with one predefined block with quick security and delivery policies.

[new] Required checkboxes - 2 HTML blocks, which when filled would appear as checkbox that customer needs to tick. This is useful as a replacement for various privacy / GDPR / consents modules.

SOCIAL LOGIN section

Social Login configuration - this is fairly straightforward, enter your Facebook or Google API details and when configured properly, login buttons shall appear for visitor (=not logged in customer, for logged in, buttons will not appear), it looks like this on checkout page:



ADVANCED section

“Clean checkout session” - after the customer makes an order, his actions, like ticked agreement with TOS are remembered, so that when he makes another order one hour later, that checkbox will be pre-ticked. If you enable this option, all session related checkbox status will be erased after order is confirmed.

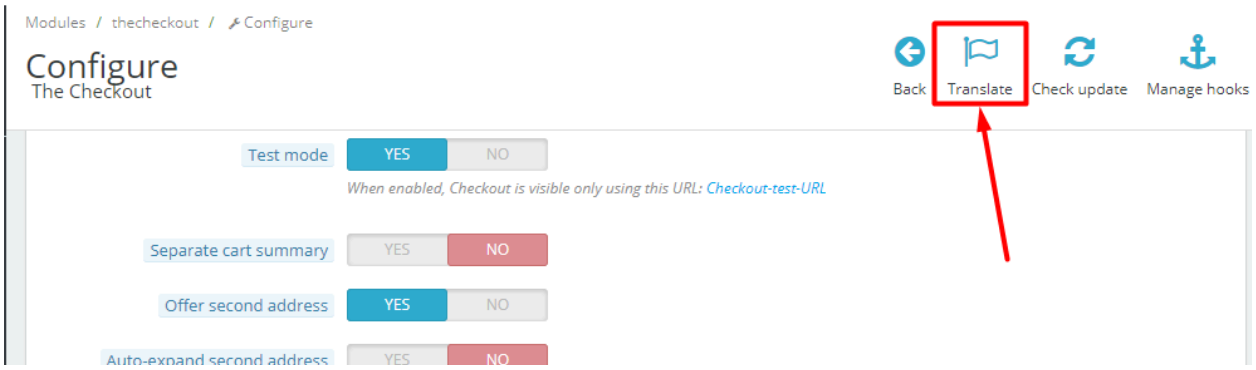
“PS CSS / JS Cache versions” - this is actual version of caching files inside of Prestashop. Normally you do not need to change this manually, but sometimes, it happens that Prestashop does not refresh caching file (that is CCC compressed file for styling - CSS and scripts - JS). Typically, you can see it when you update CSS or JS file, and compression is disabled - it works. But when you enable compression, it’s broken (old version is still used). In this case, increase these numbers +1, and click Save. That shall force Prestashop core to re-cache these CCC files.

“Custom CSS” - this is your CSS code, used directly in HTML of checkout page.

“Custom JS” - your custom JS code, used directly in HTML. Notice please that code is executed “in place”, so if you have any deferred JS include (like move JS to the end), write your jQuery or other library related code in .ready handler.

Translations

The Checkout module reuses Prestashop’s default translations as much as possible, but there are still few labels that needs modification. You can do that easily following Translate link in checkout module configuration:



Test mode: YES [✓] NO
When enabled, Checkout is visible only using this URL: Checkout-test-URL

Separate cart summary: YES NO [✓]

Offer second address: YES [✓] NO

Auto-expand second address: YES NO [✓]

Use cases - how to...

Flower shop

For flower shop it's typical that invoice address is different from the shipping address, so ideally we would want both addresses to be expanded ("Auto-expand second address") and invoice address will be shown first (position invoice address before shipping address in layout designer - first address in designer = primary address).

Also, we may want to enable gift message and gift wrapping - you can do this in default Prestashop settings, new options will appear on checkout automatically. And additionally, we may want to add another message ("Show Order Message").

Virtual goods - licenses, digital products

If you're selling non-tangible goods, like music, software, licenses, access codes, etc., typically you don't need client's shipping address.

In Delivery address fields, you can disable all fields you like, to make form a lot more compact. Or, you can leave fields as they are, and place all the fields (a list) into "... business fields", which would cause all the fields to be hidden by default and visible only for business customers, ticking special checkbox on checkout form.

GDPR

Confirmed by lawyers, if you do collect customer details only and only for the purpose of order fulfillment, you don't need to ask for GDPR consent.

However, if you provide data to a third party for data analytics, or you do marketing using customer details, you need to ask. There are plenty of GDPR modules out there, even Prestashop has own one (psgdpr). Actually, to ask for consent is as easy as letting client to tick a checkbox, so I've also a built-in option (required checkbox no.1 and no.2), where you can enter your consent text and that would be shown to customer, and would be necessary for customer to tick it.

Captcha to prevent fake registration or fake order creation

So far, to date in 10 year's history, there wasn't a single case automated targeting / spamming checkout form of my module. Due to various validations and wide variety of options, it's simply unreasonable for anyone to write one, as it would have very low success rate.

I suggest to customers, always disable captcha for checkout. It only makes things complicated for customer and we want the checkout process as smooth as possible. And once you observe any fake orders or accounts, only then we can look at the problem, there's no need to do anything upfront.

Basic troubleshooting

As a first troubleshooting step, please try same functionality that's not working well in our Checkout module, in default Prestashop checkout.

Typically that is missing shipping or payment method, missing image on product in cart, wrong calculation of shipping or cart totals, etc. Most of the time, the issue is that those respective modules are not configured well.

Once it's working well in default checkout and using our The Checkout module it's not, please reach out to us and we'll be happy to help you.

There are few issues that we know a solution for, but it's outside of our control, so let's mention and recap them.

Zone shipping calculation modules

E.g. zipcodezone module (but also other may have this issue) uses zone calculation from address, which exhibits a problem in PS core, where zone for calculation can be assigned only from customer's address, and not "any" address. To overcome this limitation, please edit classes/Cart.php, inside of getPackageShippingCost method, there's condition:

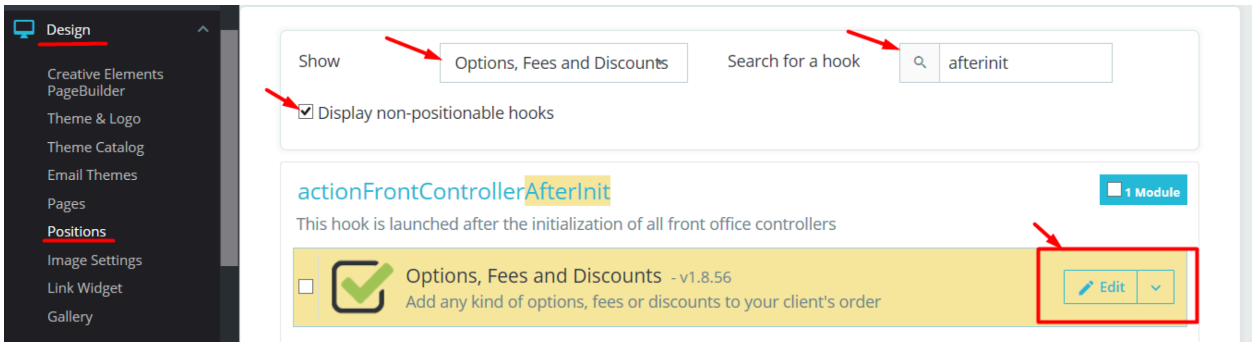
```
if (!isset($id_zone)) {
    if (!$this->isMultiAddressDelivery()
      && isset($this->id_address_delivery)
      && $this->id_address_delivery
      && Customer::customerHasAddress($this->id_customer, $this->id_address_delivery)
```

Here, comment out the last line, to remove above mentioned constraint.

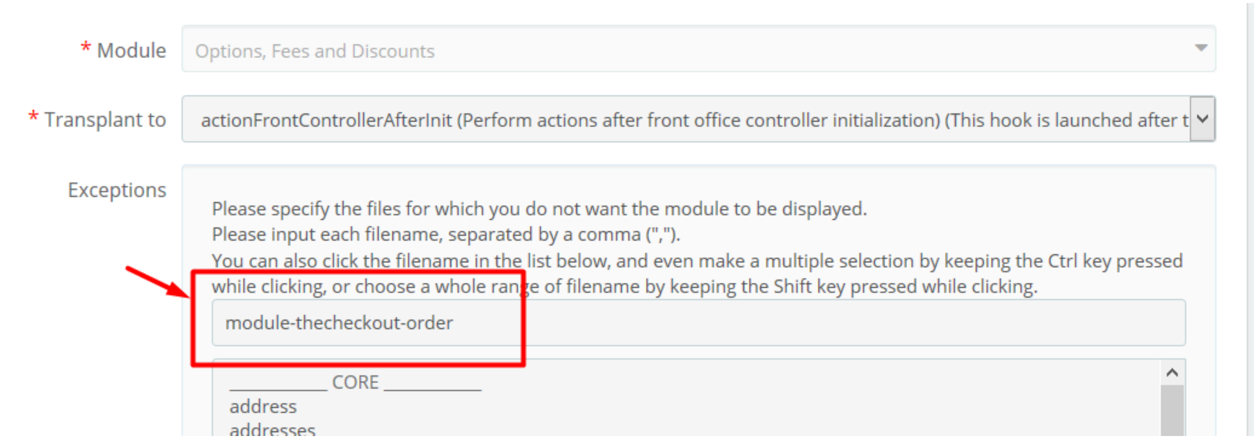
Orderfees module (Options, Fees and Discounts by motionSeed)

This module uses a hook actionFrontControllerAfterInit in which it sets and caches address before it's actually updated with data from checkout form. That causes troubles with shipping fee calculation modules.

Solution: Go to Prestashop BO, choose Design / Positions and there find afterInit hook, like this:


Design
Creative Elements PageBuilder
Theme & Logo
Theme Catalog
Email Themes
Pages
Positions
Image Settings
Link Widget
Gallery
Show
Options, Fees and Discounts
Search for a hook
afterinit
Display non-positionable hooks
actionFrontControllerAfterInit
1 Module
This hook is launched after the initialization of all front office controllers
Options, Fees and Discounts - v1.8.56
Add any kind of options, fees or discounts to your client's order
Edit

Click on Edit and on following page, add exception for checkout controller: module-thecheckout-order, like this, and hit Save.



actionFrontControllerAfterInit (Perform actions after front office controller initialization) (This hook is launched after t

Please specify the files for which you do not want the module to be displayed.
Please input each filename, separated by a comma (",").
You can also click the filename in the list below, and even make a multiple selection by keeping the Ctrl key pressed while clicking, or choose a whole range of filename by keeping the Shift key pressed while clicking.

module-thecheckout-order

__________ CORE __________
address
addresses

Darique gift module

Uses JS initialization code to display gifts in the cart summary. To support that, please add following Custom JS code in Thecheckout module configuration:

```
prestashop.on('thecheckout_updateCart', function() {
  if ("undefined" !== typeof dariqueModule) { dariqueModule?.updateSelectedPresents(); }
})
```

Optionally, add also Custom CSS to align the layout a bit better:

```
.card-block.cart-darique {
   border: none;
   & > .card-block, & > .separator {   display: none;   }
   & ul.dariqueWrapper {   margin-bottom: 0;   }
}
```

Support

Do not feel comfortable installing module yourself? Or do you face any kind of issue not listed above? Do you have any suggestions for improvement?

Our support is happy to help, just contact us using ‘Contact developer’ link on Addons market.