interweb

Generated on 21/10/2026 17:27

KNOWLEDGE BASE · PUBLIC DOCUMENT

A CDN in front of a shop or members area: what must never be cached

Speed, Cache & CDN

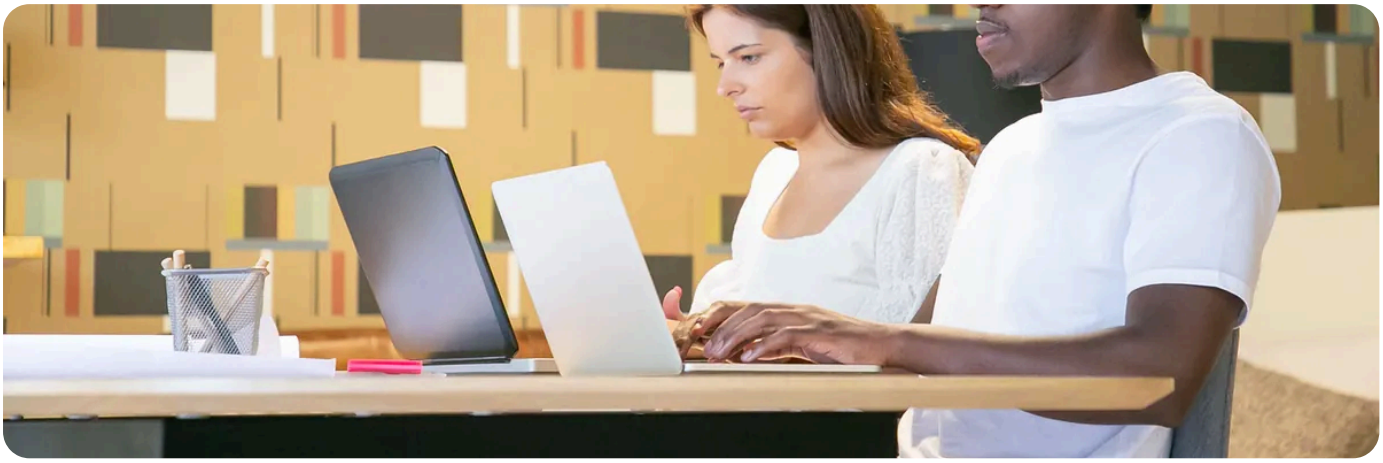
CODE	VERSION	COUNTRY · LANGUAGE
RLX-MZ-KB-2118	v2026.10.16	Mozambique · English

Interweb Mozambique
RLX Holdings, S.A. · Group RLX Holdings
Tax ID (NUIT) 401 191 607 · Legal entity no. (NUEL) 105 045 974 · Economic activity code (CAE) 62022

https://www.meu.interweb.co.mz/knowledgebase/2118

A CDN in front of a shop or members area: what must never be cached

Speed, Cache & CDN Created on **October 16, 2026** 4 min read



The short answer: in a shop, or on any site with accounts, the same page is not the same for everyone: the cart, the total, the name at the top, sometimes the price in a currency. A CDN can only store what is **the same for everyone**. If it stores a personal page, it hands it to the next visitor: swapped carts, someone seeing another person's account, or payments that never get confirmed. The rule is simple: static files, always; public pages, only with careful rules; anything with a session, never. We do not sell a CDN: what follows applies to whichever CDN you choose, on an account of your own, opened directly with that provider.

What always stays out of the CDN cache

What	Why
Cart, checkout, order confirmation page	Different for every customer. Cached, they show someone else's cart or order.
Customer account, members area, admin dashboard (on WordPress, `/wp-admin/` and `wp-login.php`)	They hold personal data and sessions.
Requests carrying a session or cart cookie	On WordPress, `wordpress_logged_in_`; on WooCommerce, `woocommerce_items_in_cart`, `woocommerce_cart_hash` and `wp_woocommerce_session_`. When a request carries one of these, the CDN should pass it straight to the origin.
Requests that change something: form posts (POST), `?add-to-cart=`, AJAX calls (`admin-ajax.php`, `?wc-ajax=`)	They are actions, not reads. A cached action returns the old answer and does nothing.

What	Why
Payment return addresses and webhooks	They must always reach the origin: no cache, no challenge or captcha, no country blocking. Otherwise the customer pays and the order stays pending.
Search, filters, prices by currency or country	They change with the visitor. Either do not cache them, or make sure the cache key includes whatever changes.
Pages with forms (contact, comments, login)	They carry security codes that expire. An old copy gives “the link has expired” or a rejected form. Short lifetime, or out of the cache.

Setting up the rules without breaking the shop

1. **Start with static files only** (images, CSS, JS, fonts). It is what many CDNs do by default, and in a shop with lots of photographs it is where most of the gain is: **product images for a fast shop**.

2. **If you also want to cache HTML,** write the rule for public pages only (home, categories, products), passing straight to the origin when the request carries a session or cart cookie, with explicit exclusions for the paths in the table. How to write these rules differs from one CDN to another: follow your provider’s documentation.

3. **Check the origin’s headers:** personal pages should go out with `Cache-Control: private, no-store` or equivalent. WooCommerce and the cache plugin usually take care of it; confirm with `curl -sI`, as in **cache headers explained**.

4. **Test with two browsers:** in one, put a product in the cart; in the other, in a private window, open the cart: it must be empty. Log in on the first and check the second does not see your name.

5. **Place a full test order** with the real payment method, and check the order turns to paid: **an online shop with local payment methods**.

6. **Repeat these tests** every time you change a CDN rule or the cache plugin.

! **A “cache everything” rule with no exclusions is the worst mistake you can make on a shop.**
If it happened and someone saw another person’s data: switch the rule off, purge the whole CDN, clear the site cache, and only then write the rule again with its exclusions. Similar symptoms without a CDN are covered in **the WooCommerce cart empties by itself or shows the wrong items**.

When in doubt, do not cache HTML at the CDN.

On a shop, the server's page cache does that job and already knows the difference between a visitor with a cart and one without (**LiteSpeed Cache: the first settings**). Leave static files and protection to the CDN.

NEED A HAND?

The shop started misbehaving after you switched the CDN on? Tell us the address, which CDN you use and what rules you created.

Open a support ticket →

SEE ALSO

- Cache rules for a WordPress site, without hiding the dashboard →
- The WooCommerce cart empties by itself or shows the wrong items →
- Checkout errors in WooCommerce: where to look →
- An online shop with local payment methods: WooCommerce and getting paid →
- Cache headers explained: Cache-Control, Expires, ETag and how a CDN reads them →
- LiteSpeed Cache for WordPress: the first settings →

interweb

Interweb Mozambique

RLX Holdings, S.A. · Group RLX Holdings

TAX ID (NUIT)	401 191 607
LEGAL ENTITY NO. (NUEL)	105 045 974
ECONOMIC ACTIVITY CODE (CAE)	62022
REGISTERED OFFICE	Bairro 7 de Setembro Vilankulo 1304 · CEP 0411-02, Inhambane, Mozambique
CONTACT	info@interweb.co.mz · +258 84 635 1111 · +258 86 635 1111 · www.interweb.co.mz
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-MZ-KB-2118 · v2026.10.16 · Generated on 21/10/2026 17:27