URL Coupons: Developer Hooks and Filters Reference

Developers · https://wpfactory.com/docs/url-coupons-developer-hooks-filters/

Coupons & Add to Cart by URL Links for WooCommerce exposes a set of actions and filters so you can change how a URL coupon is detected, validated, applied, and where the customer is sent afterward, all in code and without editing plugin files. Every hook name below is taken directly from the plugin source (Pro v1.8.2); the shared prefix is `alg_wc_url_coupons_`.

The options these hooks interact with live under **WPFactory > Coupons by URL** (also reachable at **WooCommerce > Settings > URL Coupons**).

How a URL coupon flows through the hooks

When a visitor opens a URL that carries the coupon key (for example `?alg_apply_coupon=SAVE10`), the plugin runs this sequence. Knowing the order tells you which hook to use.

1. `alg_wc_url_coupons_apply_url_coupon_validation` is checked. If it returns `false`, nothing happens.
2. `alg_wc_url_coupons_before_coupon_applied` fires.
3. The coupon is either applied immediately or delayed (the Delay coupon feature).
4. `alg_wc_url_coupons_coupon_applied` fires after the coupon is added to the cart.
5. `alg_wc_url_coupons_after_coupon_applied` fires last, which is where the optional redirect happens.

Actions

Action	Fires when	Arguments
`alg_wc_url_coupons_before_coupon_applied`	Just before the coupon is applied or delayed	`$coupon_code, $key`
`alg_wc_url_coupons_coupon_applied`	Right after the coupon is added to the cart	`$coupon_code, $key, $result`

Action	Fires when	Arguments
`alg_wc_url_coupons_after_coupon_applied`	After application finishes (redirect runs here)	`$coupon_code, $key, $result`
`alg_wc_url_coupons_coupon_delayed`	When a coupon is queued by the Delay coupon feature	`$coupon_code, $key, $result`
`alg_wc_url_coupons_after_coupon_delayed`	After the delay handler finishes	`$coupon_code, $key, $result`

In every case `$coupon_code` is the code taken from the URL, `$key` is the URL key in use (default `alg_apply_coupon`), and `$result` is the value returned by WooCommerce when the coupon was added, which is truthy on success.

Run your own code when a URL coupon is applied

```
add_action( 'alg_wc_url_coupons_coupon_applied', function ( $coupon_code, $key, $result ) {
    if ( $result ) {
        // e.g. log the campaign hit, tag the session, fire an analytics event.
        error_log( 'URL coupon applied: ' . $coupon_code );
    }
}, 10, 3 );
```

Filters

Filter	Filtered value	Arguments
`alg_wc_url_coupons_apply_url_coupon_validation`	Whether the coupon should be applied at all (seeded `true`)	`$validation, $args`
`alg_wc_url_coupons_redirect_url`	The URL the customer is redirected to after a successful apply	`$url, $coupon_code, $key`
`alg_wc_url_coupons_keys_to_remove_on_redirect`	The query-arg keys stripped from the URL on redirect	`$keys_to_remove`

Filter	Filtered value	Arguments
`alg_wc_url_coupons_data_storage_type`	Where the plugin stores its data, `session` or `cookie`	`$type`
`alg_wc_url_coupons_javascript_reloaded_param`	The query-arg name used by the Javascript reload feature (default `reloaded`)	`$param`
`alg_wc_url_coupons_get_notices`	The list of custom notices to show (Pro notice feature)	`$notices`

Only apply a URL coupon for logged-in users

```
add_filter( 'alg_wc_url_coupons_apply_url_coupon_validation', function ( $validation, $args ) {
    return is_user_logged_in() ? $validation : false;
}, 10, 2 );
```

Send the customer to a thank-you page after any URL coupon is applied

```
add_filter( 'alg_wc_url_coupons_redirect_url', function ( $url, $coupon_code, $key ) {
    return home_url( '/coupon-applied/' );
}, 10, 3 );
```

Note that the built-in **Redirect URL** option (Pro) already uses this filter internally, so a per-coupon redirect or a global redirect set in the admin will take priority unless you run at a later priority.

The internal settings gate filter

`alg_wc_url_coupons_settings` is how the plugin decides whether a Pro-only field is disabled in the admin. In the free version the filter is left untouched so it returns the value it is passed (the `array( 'disabled' => 'disabled' )` attribute or the "available in Pro" marketing string). In Pro, `Alg_WC_URL_Coupons_Pro::settings()` hooks it to return an empty string, which removes the `disabled` attribute and enables the field. It is an internal gate and is not intended for store customization.

Data storage helpers

The plugin reads and writes its own data (queued coupons, saved empty-cart coupons, delayed notices) through two helper functions that respect the **Data storage type** option, so the same call works whether the store is set to session or cookie.

- `alg_wc_url_coupons_data_storage_get( $key, $default_value = '' )`
- `alg_wc_url_coupons_data_storage_set( $key, $value )`

You can reach the core object through `alg_wc_url_coupons()->core`, and the list of hooks the plugin can attach to for detection is returned by `alg_wc_url_coupons()->core->get_possible_main_hooks()` (currently `wp_loaded` and `init`).

Per-coupon meta keys (Pro)

The Pro per-coupon options store their values as post meta on the coupon. If you build coupons programmatically, set these keys:

- `_alg_wc_url_coupons_redirect_url` - the per-coupon redirect URL (used when **Redirect URL per coupon** is enabled).
- `_alg_wc_url_coupons_notice` - the per-coupon notice text (used when **Notice per coupon** is enabled).

Shortcodes

The plugin registers two shortcodes. For the full parameter list see the URL Coupons shortcodes guide.

- `[alg_wc_url_coupons_translate]` - shows or hides content based on the active WPML or Polylang language. Registered in the free core.
- `[alg_wc_url_coupons_print_notices]` - prints the queued WooCommerce notices. Registered in the Pro version.

Need help?

If you are extending URL Coupons and a hook does not behave as expected, contact our team on the URL Coupons support forum with the hook name and a short code sample, and we will help you find

the right extension point.



URL COUPONS

See every feature and the pricing on the URL Coupons plugin page.

View plugin →

Related articles

- URL Coupons Settings and Option IDs Reference
- Make URL Coupons Work for Guests and Cached Sites
- How to Translate URL Coupon Text with WPML and Polylang
- How to Set a Redirect URL Per Coupon
- How to Delay a URL Coupon Until a Product Is Added