



Własny kod w nagłówku i stopce

INSTRUKCJA OBSŁUGI · Wersja 1.0.2 · PrestaShop 1.7.8 – 9.x

1. Do czego służy moduł

Moduł dodaje do sklepu Twój własny kod bez edycji szablonu: Google Tag Manager, Google Analytics, Meta Pixel, metatagi weryfikacyjne Google Search Console czy Bing, widżety czatu i podobne fragmenty. Wklejasz kod w konfiguracji modułu i wybierasz, gdzie ma trafić: do nagłówka strony (sekcja `<head>`), zaraz po otwarciu `<body>` albo na koniec strony. Osobne pole przyjmuje kod konwersji na stronę potwierdzenia zamówienia, z uzupełnionym numerem, referencją i kwotami zamówienia. Dwa kolejne pola służą do własnego CSS i JavaScriptu.

Kod jest zapisany w bazie danych sklepu, a nie w plikach szablonu, więc zostaje po aktualizacji lub zmianie szablonu, a każdy sklep w multistore może mieć własny kod.

Działa z **PrestaShop 1.7.8, 8.x i 9.x** (PHP 7.2.5 lub nowszy), z szablonami Classic i Hummingbird oraz z trybem multistore.

2. Instalacja

1. Na stronie modułu w sklepie addonspresta.com kliknij **Pobierz za darmo** (konto nie jest potrzebne).
2. W panelu sklepu otwórz **Moduły** → **Menedżer modułów** i kliknij **Załaduj moduł**.
3. Upuść plik `ap_customcode_v1.0.1.zip` i poczekaj na potwierdzenie.
4. Kliknij **Konfiguruj** na karcie modułu.

3. Jak to działa

Każdy blok kodu moduł wstawia przez standardowy hook PrestaShop, który wyświetla szablon:

- **Kod w nagłówku (head) i Własny CSS** — na końcu sekcji `<head>`, po stylach szablonu (hook `displayHeader`);
- **Kod na początku strony (body)** — zaraz po znaczniku `<body>` (hook `displayAfterBodyOpeningTag`);
- **Kod na końcu strony (body) i Własny JavaScript** — przed znacznikiem `</body>`, po skryptach szablonu (hook `displayBeforeBodyClosingTag`);
- **Kod konwersji** — tylko na stronie potwierdzenia zamówienia (hook `displayOrderConfirmation`).

Szablon Classic w PrestaShop 1.7.8, 8 i 9 oraz szablon Hummingbird w PrestaShop 9 wyświetlają te hooki na każdej stronie sklepu. Szablon innego autora może pomijać hooki w `<body>` — jeśli kod na początku lub na końcu strony się nie pojawia, zapytaj autora szablonu. Strona przerwy technicznej i strona „kraj niedostępny” nie mają hooków, więc kod się na nich nie pojawi.

Moduł wstawia kod dokładnie w takiej postaci, w jakiej go wkleisz — niczego nie sprawdza ani nie zmienia. Jedyny wyjątek: jeśli w polu CSS lub JavaScript wkleisz kod razem z otaczającym znacznikiem `<style>` lub `<script>`, moduł go usunie, bo dodaje własny. Blok wyłączony albo pusty nie wstawia na stronę nic.

Kod konwersji uruchamia się przy każdym wyświetleniu strony potwierdzenia. W kodzie zmienne `{order_id}`, `{order_reference}`, `{total_paid}` (z podatkiem), `{total_paid_tax_excl}`, `{total_products}` (produkty bez podatku), `{total_shipping}` (z podatkiem) i `{currency}` (kod ISO waluty, np. `PLN`) zamieniają się na wartości zamówienia pokazanego na stronie. Liczby trafiają do kodu jako liczby zaokrąglone do dwóch miejsc po przecinku, a teksty jako napisy JavaScript w cudzysłowie, więc można ich użyć wprost w skrypcie i nie mogą go zepsuć. Dane osobowe klienta (imię, e-mail, adres) nie są dostępne. Gdy jeden koszyk dzieli się na kilka zamówień (po jednym na przewoźnika), wartości dotyczą zamówienia pokazanego na stronie potwierdzenia.

Sam moduł nie łączy się z `addonspresta.com` ani z żadnym innym serwerem i nie zapisuje plików cookie. Wklejony kod zwykle to robi — odpowiadasz za niego Ty, a skrypty śledzące muszą respektować zgodę odwiedzających na pliki cookie (RODO). Skorzystaj z mechanizmów zgody w dodawanym narzędziu (np. tryb zgody Google w Google Tag Manager) albo z modułu do zarządzania zgodami.

4. Funkcje i ustawienia

Otwórz **Moduły** → **Menedżer modułów** → **Własny kod w nagłówku i stopce** → **Konfiguruj**. Ramka Pomoc na górze prowadzi do tej instrukcji i do strony modułu w `addonspresta.com`, w języku Twojego panelu.

Każdy blok ma przełącznik **Wł. / Wył.** i pole na kod. Wyłączenie bloku zostawia kod w polu, ale przestaje go wstawiać na strony — przydaje się, gdy jakiś skrypt sprawia kłopoty. Każde pole mieści do 48 000 bajtów. Jeden przycisk **Zapisz** zapisuje wszystkie bloki.

Kod na każdej stronie

- **Kod w nagłówku (head)** — Google Tag Manager (część ze znacznikiem `<script>`), Google Analytics, Meta Pixel, metatagi weryfikacyjne, np. `<meta name="google-site-verification" content="...">`.
- **Kod na początku strony (body)** — część `<noscript>` kodu Google Tag Manager.
- **Kod na końcu strony (body)** — widżety czatu i skrypty, które nie muszą ładować się wcześniej, także skrypty zewnętrzne, np. `<script src="https://..." async></script>`.

Potwierdzenie zamówienia

Kod konwersji — konwersje Google Ads, zdarzenie `Purchase` w Meta Pixel, sieci afiliacyjne. Zmienne wpisuj bez cudzysłowów, moduł doda je tam, gdzie trzeba:

```
<script>
  gtag('event', 'purchase', {
    transaction_id: {order_reference},
    value: {total_paid},
    currency: {currency}
  });
</script>
```

Własny CSS i JavaScript

- **Własny CSS** — same reguły CSS, bez znaczników `<style>`. Ładują się po stylach szablonu, więc wygrywają z regułami szablonu o tym samym selektorze.
- **Własny JavaScript** — sam kod JavaScript, bez znaczników `<script>`. Uruchamia się na końcu strony, po skryptach szablonu. Zewnętrzny plik skryptu wklej jako cały znacznik `<script src="...">` w polu **Kod na końcu strony (body)**.

Pole CSS nie może zawierać `</style>`, a pole JavaScript — `</script>` ani `<!--`: taki kod zamknąłby za wcześnie albo w ogóle nie zamknął znacznika dodawanego przez moduł i zepsuł stronę, dlatego zapis kończy się komunikatem o błędzie.

5. Multistore

Kod zapisuje się dla sklepu lub grupy sklepów wybranej w nagłówku panelu; informacja nad formularzem mówi, co właśnie edytujesz. Wybierz **Wszystkie sklepy**, żeby ustawić kod wspólny dla wszystkich sklepów, albo jeden sklep, żeby nadać mu własny kod. Kod zapisany dla jednego sklepu ma pierwszeństwo przed kodem jego grupy i kodem dla wszystkich sklepów. W kontekście wszystkich sklepów albo grupy zapisać może tylko pracownik, który zarządza każdym jej sklepem. Jedna licencja obejmuje wszystkie sklepy instalacji.

6. Odinstalowanie

Menedżer modułów → **Własny kod w nagłówku i stopce** → **Odinstaluj** usuwa cały kod i ustawienia modułu. Jeśli kod może się jeszcze przydać, najpierw go skopiuj albo użyj opcji **Wyłącz**, która wszystko zachowuje.

7. Najczęstsze pytania i problemy

Kodu nie ma w źródle strony. Sprawdź, czy blok jest włączony i zapisany, a potem wyczyść pamięć podręczną w **Zaawansowane** → **Wydajność**. Jeśli przed sklepem działa moduł cache stron albo

CDN, wyczyść także jego pamięć. Jeśli brakuje tylko bloków w `<body>`, szablon może nie wyświetlać tych hooków (zob. „Jak to działa”).

Google Search Console nie znajduje metatagu weryfikacyjnego. Wklej znacznik `<meta ...>` w polu **Kod w nagłówku (head)**, zapisz, wyczyść pamięć podręczną i otwórz źródło strony głównej: znacznik musi być wewnątrz `<head>`. Potem ponów weryfikację.

Po zapisaniu kodu sklep wygląda źle. Wyłącz blok i zapisz: kod zostaje w polu, więc możesz go spokojnie poprawić. Tak samo postępuj ze skryptem, który zgłasza błędy w konsoli przeglądarki.

Konwersja liczy się podwójnie. Strona potwierdzenia uruchamia kod konwersji ponownie, gdy klient ją odświeży. Przekaż `{order_reference}` albo `{order_id}` jako identyfikator transakcji — Google Ads, Meta i większość sieci afiliacyjnych policzą wtedy każde zamówienie raz.

Dlaczego `{total_paid}` wpisuję bez cudzysłowu? Moduł wstawia gotowe wartości JavaScript: `123.45` dla kwot i `"PLN"` w cudzysłowie dla tekstów. Dodatkowy cudzysłów zamieniłby kwotę w tekst.

Czy mogę użyć zmiennych w pikselu obrazkowym albo w linku? Nie bezpośrednio. Zmienne są przeznaczone do JavaScriptu: tekst taki jak `{order_reference}` trafia do kodu razem z cudzysłowem, który przerwałby atrybut HTML, np. ``. Adres piksela zbuduj w skrypcie:

```
<script>
  new Image().src = 'https://example.com/pixel?ref=' +
encodeURIComponent({order_reference});
</script>
```

Jak wyłączyć kod na wybranych stronach? W **Wygląd** → **Pozycje** znajdź hook bloku (np. `displayHeader`), edytuj pozycję modułu Własny kod w nagłówku i stopce i dodaj strony do jego wyjątków.

Czy mogę zmienić sposób wstawiania własnego CSS lub JavaScriptu? Tak. Skopiuj `views/templates/hook/style.tpl` albo `script.tpl` do `themes/<twój-szablon>/modules/ap_customcode/views/templates/hook/` i zmień kopię, np. żeby dodać atrybut wymagany przez narzędzie do zgód na pliki cookie.

8. Aktualizacje i wsparcie

Aktualizacje są bezpłatne: nowe wersje pobierzesz ze strony modułu w addonspresta.com i wgrasz jak nowy moduł. Kod i ustawienia zostają. Każda wersja przed publikacją przechodzi nasz audyt bezpieczeństwa.

Ten bezpłatny moduł ma tę samą licencję co nasze moduły płatne (jeden sklep z dowolną liczbą sklepów multistore, kod możesz zmieniać, ale nie udostępniać) i jest **bez wsparcia technicznego**. Jeśli potrzebujesz pomocy przy instalacji lub konfiguracji, możemy ją zaproponować jako płatną

usługę. Błędy bezpieczeństwa zgłaszaj na **mail@addonspresta.com** — naprawiamy je w pierwszej kolejności.

-->