

**API systemu dropshippingu
MATTERHORN**

**wersja 1.11
8.11.2010**

**Przygotował
Tomasz Jaromin**

Historia zmian

8.11.2010

dodanie metody shipping_options_xml zwracającej listę rodzajów przesyłki w postaci dokumentu XML

dodanie funkcji checkUser sprawdzającej czy użytkownik o podanym adresie email jest już zarejestrowany w systemie.

Dodano informację o adresie powrotnym do sklepu z systemu płatności.pl

Założenia ogólne

Pragniemy przedstawić Państwu innowacyjny produkt skierowany dla obecnych oraz nowych podmiotów na rynku e-commerce odzieży i bielizny damskiej. Matterhorn Dropshipping to kompleksowa usługa dla już działających sklepów odzieżowych w sieci, która pozwala w szybki sposób outsourcować proces magazynowania i realizacji zamówień w swoim sklepie internetowym.

Nasze rozwiązanie daje Państwu możliwość korzystania z dużego magazynu naszej sieci, profesjonalną obsługę Państwa klientów popartą doświadczeniami najlepszego internetowego sklepu odzieżowego wg tygodnika WPROST i portalu Money.pl.

Jak działa Matterhorn Dropshipping?

Usługa polega na oferowaniu w swoim sklepie internetowym bogatej oferty odzieży i bielizny hurtowni Matterhorn. Hurtownia Matterhorn przyjmuje od sklepów zamówienia do zrealizowania i to Matterhorn w imieniu sklepu realizuje w 100% proces wysyłki (włączając w to obsługę płatności, zwrotów, call center oraz obsługę email).

Właściciel sklepu staje się wolny od problemów związanych z logistyką, magazynowaniem, sprawną realizacją zamówień. Może od tej chwili skupić się wyłącznie na działaniach marketingowych i powiększaniem skali sprzedaży.

Co należy zrobić?

Jeżeli posiadacie już Państwo sklep internetowy należy zintegrować ofertę produktową z naszą ofertą. Cała nasza baza produktów jest dostępna w pliku XML opisanym w tej dokumentacji. Należy również zmodyfikować proces zamówienia tak, aby Państwa sklep składał zamówienie w systemie Matterhorn.

Zamawianie produktów

System Matterhorn udostępnia skrypt do automatycznego zamawiania produktów.

Aby wygenerować zamówienie należy wywołać skrypt dostępny pod adresem:
www.matterhorn.pl/api/makepartnerorder.php

i metodą GET lub POST przekazać następujące parametry:

Nazwa zmiennej	Opis	Wymagana
produkty	Tablica z id produktów do zamówienia. Przekazywane ID to identyfikator koloru/rozmiaru konkretnego produktu z pliku XML cennika	TAK
ilosc	Tablica określająca ilość sztuk do zamówienia. Musi zawierać dokładnie tyle samo elementów co tablica <i>produkty</i> Pierwszy element w tablicy <i>ilosc</i> odpowiada pierwszemu elementowi tablicy <i>produkty</i> , drugi drugiemu itd.	TAK
shopId	Identyfikator sklepu nadany przez Matterhorn	TAK
customerId	Identyfikator klienta w systemie Matterhorn. Jeżeli nie zostanie przekazany lub będzie miał wartość 0, wtedy system utworzy nowe konto klientow. Jeżeli przekazemy ten parametr to dane adresowe (fdata, sdata, ulica, nr_domu itd) posłużą do aktualizacji adresu klienta, dlatego muszą być przekazane.	NIE
cena	Tablica z cenami brutto produktów pochodzącymi ze sklepu partnera. W modelu elastycznym partner ma prawo do własnej polityki cenowej i tym parametrem informuje system Matterhorn w jakiej cenie produkt ma zostać sprzedany. Ceny brutto jako separator dziesiętny powinny używać kropki.	NIE (używana wyłącznie w modelu elastycznym)
fdata	Imię klienta końcowego	TAK
sdata	Nazwisko klienta końcowego	TAK
ulica	Ulica adresu wysyłki	TAK
nr_domu	Numer domu adresu wysyłki	TAK
nr_lokalu	Numer mieszkania/lokalu	NIE
miasto	Miasto z adresu wysyłki	TAK
kod	Kod pocztowy z adresu wysyłki	TAK
telefon	Numer telefonu odbiorcy	NIE
email	Adres email odbiorcy	TAK

info	Dodatkowe informacje jakie odbiorca może przekazać do sklepu (np. proponowane godziny dostaw lub jakakolwiek inna informacja)	NIE
kraj	Dwuliterowy kod kraju adresata. Dla Polski będzie to PL, Niemcy DE itd..	TAK
przesylka_id	Identyfikator określający rodzaj płatności i przesyłki. Wybrany rodzaj wysyłki determinuje również rodzaj płatności (z góry lub za pobraniem). Jeżeli wybrana zostanie płatność z góry, przesyłka zostanie wysłana dopiero po odnotowaniu płatności za zamówienie w naszym systemie. W takim wypadku należy na końcu zamówienia wyświetlić klientowi formularz do płatności.pl Listę rodzajów płatności wraz z informacją o min. wartości zamówienia dla danego typu przesyłki, nazwie kuriera itp. można pobrać w postaci dokumentu XML wywołując funkcję shipping_options_xml	TAK
sessionId	Unikalny identyfikator sesji, może to być np. numer zamówienia w systemie partnera.	TAK
checksum	Suma kontrolna uzyskana z wywołania funkcji MD5 dla ciągu znaków SECRET_WORD+SESSION_ID Secret_word to hasło znany wyłącznie partnerowi i naszej firmie. Dzięki zakodowaniu numeru sesji oraz hasła zabezpieczamy się przeciw nieautoryzowanym transakcjom. Przykład dla języka PHP wygenerowania checksum <? \$checksum = md5('HASLO'.\$sessionId); >	TAK

W wyniku skrypt www.matterhorn.pl/api/makepartnerorder.php powinien zwrócić komunikat

```
<OK orderid='55555' amount='15678' show_payment='true'>Wygenerowano zamówienie nr 5555</OK>
```

gdzie 55555 to numer zamówienia w systemie Matterhorn, amount to wartość zamówienia brutto dla klienta z kosztem wysyłki wyrażona w groszach.

W powyższym przykładzie przedstawiona jest kwota 156,78 zł.

Atrybut show_payment informuje, czy należy wyświetlić formularz płatności.pl.

UWAGA. Jeżeli klient wybrał w sklepie płatność z góry, w tym momencie powinna zostać mu wyświetlony formularz z systemu płatności.pl. Partner może go wygenerować samodzielnie lub skorzystać z gotowego skryptu, który jest opisany w dziale narzędzia.

Opis formatki dla systemu platnosci.pl znajduje się tutaj [LINK!](#)

W przypadku błędu skrypt zwróci odpowiedź:

```
<ERROR errorid='1'>Komunikat błędu</ERROR>
```

Komunikaty błędów

Kod błędu (errorid)	Opis/przyczyna
1	Brak parametru sessionId
2	Brak parametru checksum
3	Suma kontrolna się nie zgadza. Proszę sprawdzić wyliczanie parametru checksum.
4	Parametr produkty jest pusty lub nie jest tablicą.
5	Brak lub błędny identyfikator produktu.
6	Dla identyfikatora produktu nie znaleziono odpowiednika w systemie Matterhorn. Prawdopodobnie należy zaciągnąć na nowo bazę produktów.
7	Inny błąd wygenerowania zamówienia. W takim wypadku prosimy o kontakt z naszym działem technicznym.
8	Brak lub nieprawidłowy parametr shopId

Przykład wywołania skryptu

```
http://www.matterhorn.pl/api/makepartnerorder.php?  
shopId=gustowna.pl&sessionId=12345&checksum=7c6ab5bb234b8667eeb352b96fde1a3&  
produkty[]=57410&ilosc[]=1&produkty[]=23088&ilosc[]=1&fdata=Joanna&sdata=Kowals  
ka&ulica=Jana%20Paw%20B3a  
%20II&nr_domu=1b&nr_lokalu=1&miasto=Warszawa&kod=01-  
001&telefon=048%2055555646&email=joanna@test.pl&info=Prosz%EA%20puka  
%E6,%20domofon%20nie%20dzia%B3a&kraj=pl&przesylka_id=81
```

lub dla modelu elastycznego podajemy dodatkowo ceny w jakich produkty mają być sprzedane:

[http://www.matterhorn.pl/api/makepartnerorder.php?shopId=gustowna.pl&sessionId=12345&checksum=7c6ab5bb234b8667eeb352b96fde1a3&produkty\[\]=57410&iosc\[\]=1&cena\[\]=25.9&produkty\[\]=23088&iosc\[\]=1&cena\[\]=45.88&data=Joanna&sdata=Kowalska&ulica=Jana%20Paw%20B3a%20II&nr_domu=1b&nr_lokalu=1&miasto=Warszawa&kod=01-001&telefon=048%2055555646&email=joanna@test.pl&info=Prosz%EA%20puka%20E6,%20domofon%20nie%20dzia%20B3a&kraj=pl&przesylka_id=81](http://www.matterhorn.pl/api/makepartnerorder.php?shopId=gustowna.pl&sessionId=12345&checksum=7c6ab5bb234b8667eeb352b96fde1a3&produkty[]=57410&iosc[]=1&cena[]=25.9&produkty[]=23088&iosc[]=1&cena[]=45.88&data=Joanna&sdata=Kowalska&ulica=Jana%20Paw%20B3a%20II&nr_domu=1b&nr_lokalu=1&miasto=Warszawa&kod=01-001&telefon=048%2055555646&email=joanna@test.pl&info=Prosz%EA%20puka%20E6,%20domofon%20nie%20dzia%20B3a&kraj=pl&przesylka_id=81)

Przykładowy skrypt w języku PHP wywołujący zamówienie:

```
<?
    $shopId = 'gustowna.pl'; // identyfikator sklepu nadawany przy rejestracji
    $sessionId = 12345; // numer sesji lub id zamówienia w sklepie
    $secret_word = 'fsdfs'; // hasło
    $checksum = MD5($secret_word.$sessionId); // generowanie sumy kontrolnej
    $imie = 'Joanna'; // imię adresata
    $nazwisko = 'Kowalska'; // nazwisko adresata
    $ulica = 'Jana Pawła II'; // ulica adresata
    $nr_domu = '1b'; // numer domu
    $nr_lokalu = '1'; // numer lokalu
    $miasto = 'Warszawa'; // miasto adresata
    $kod = '01-001'; // kod pocztowy wysyłki
    $telefon = '048 55555646'; // numer telefonu klienta
    $email = 'joanna@test.pl'; // adres email klienta
    $info = 'Proszę pukać, domofon nie działa'; // dodatkowe informacje od klienta

    $produkty = '&produkty[]=23088&iosc[]=1&produkty[]=57410&iosc[]=1';

    $parameters = 'shopId='.$shopId.'&sessionId='.$sessionId.'&checksum='.$checksum.
    '&fdata='.$imie.'&sdata='.$nazwisko.'&ulica='.$ulica.'&nr_domu='.$nr_domu.'&nr_lokalu='
    .$nr_lokalu.'&miasto='.$miasto.'&kod='.$kod.'&telefon='.$telefon.'&email='
    .$email.'&info='.$info.$produkty;

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, "http://www.matterhorn.pl/api/makepartnerorder.php");
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
    curl_setopt($ch, CURLOPT_HEADER, 0);
    curl_setopt($ch, CURLOPT_TIMEOUT, 20);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $parameters);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    $api_response = curl_exec($ch);
    curl_close($ch);

// zmiany w wersji dokumentacji 1.1
if (eregi("<OK orderid=(.*) amount=(.*) show_payment=(.*)>(.*?)</OK>", trim($api_response), $parts)) {
    // zamówienie wygenerowane pomyslnie
    $orderid = $parts[1];
    $amount = $parts[2];
    $show_payment = $parts[3]; //czy wyswietlic formatke dla platnosci TRUE/FALSE
    $message = $parts[4];
    echo "Zamowienie zlozone: $orderid $show_payment $message";
}

if (eregi("<ERROR errorid=(.*)>(.*?)</ERROR>", $api_response, $parts)) {
    // wysapil blad w zamowieniu
    $error_id = $parts[1];
    $error_message = $parts[2];
    echo "BLAD: $error_id $error_message ";
}
}
```

?>

Tryb sprzedaży z własną polityką cenową.

Jeżeli chcemy sprzedawać produkty podając własne ceny, należy zmodyfikować linię:

```
$produkty = '&produkty[]=23088&ilosc[]=1&produkty[]=57410&ilosc[]=1';
```

dodając własne ceny dla każdego produktu:

```
$produkty = '&produkty[]=23088&ilosc[]=1&cena[]=24.99&produkty[]=57410&ilosc[]=1&cena[]=58.70';
```

Proszę pamiętać, że ta funkcjonalność jest możliwa wyłącznie, kiedy Państwa sklep jest przestawiony w tryb elastycznej polityki cenowej. W tym celu prosimy o kontakt z naszym działem dropshippingu.

Dobra praktyka:

Schemat procesu zakupowego powinien wyglądać następująco:

1. klient dodaje produkty do koszyka
2. w podsumowaniu koszyka lub stronie zamówienia powinien mieć możliwość wpisania danych adresowych lub zalogowania się.
3. Jeżeli klient zalogował się do systemu (do tego celu używamy funkcji getUserId) należy pobrać jego dane adresowe z systemu Matterhorn (funkcja getUserData) i wstawić je do formularza zamówienia. W tym momencie klient może je ew. poprawić.
4. Klient wybiera przycisk „Zamawiam”. W tym momencie system sklepowy powinien złożyć zamówienie w systemie Matterhorn. Jeżeli klient był zalogowany, należy przekazać również parametr customerId będący identyfikatorem klienta. Jeżeli klient nie logował się wcześniej tylko po prostu wypełnił formularz zamówienia, wtedy nasz system założy nowe konto dla klienta.

Praktyczny przykład można obejrzeć na stronie sklepu Intymnie.com dokonując testowego zamówienia.

Struktura pliku XML z bazą produktów

Aby sprawnie zarządzać ofertą konieczna jest integracja Państwa sklepu z naszą bazą produktów. W tym celu system Matterhorn udostępnia dokument XML zawierający wszystkie niezbędne dane produktów oraz informacje o aktualnych stanach magazynowych. Informacje te są aktualizowane co 3 godziny.

Plik XML zawiera informacje o wszystkich produktach dostępnych w systemie Matterhorn, ich nazwie, marce, kategorii, obrazkach oraz dostępności.

Głównym elementem pliku XML jest node <products> zawierający elementy <product> zawierające informację o każdym produkcie.

UWAGA: Node „product” zawiera parametr „ID” oznaczający identyfikator produktu. Nie jest to identyfikator używany w procesie składania zamówienia gdyż określa on jedynie model odzieży a nie konkretny rozmiar czy kolor.

W skrypcie do zamawiania produktów (kod we wcześniejszym rozdziale) podawane są kody określające konkretną opcję produktu. W pliku XML możemy je znaleźć jako parametr noda product->options o nazwie externalID.

Założmy, że klient chce kupić 1 parę stringów Parysa z poniższego pliku XML w kolorze białym i rozmiarze XML. W takim wypadku id tej opcji to 145.

W kodzie zamawiania produktów będzie to wyglądało zatem tak:

```
$produkty = '&produkty[]=145&ilosc=1';
```

Przykładowy plik XML

```
<products>
  <product id="25">
    <name>Parysa</name>
    <brand>Róża</brand>
    <category>Stringi</category>
    <images>
      <image_url>http://www.matterhorn.pl/bigimg.php?id=301</image_url>
    </images>
    <price>33.9</price>
    <description>
      Delikatna elastyczna, transparentna tkanina w połączeniu z pięknym, grubym haftem tworzy duet niezwykle zmysłowy i
      niewinny. Do kompletu dobiez finezyjny wiązany z przodu biustonosz typu pup, który zaznaczy i podkreśli kobiece wdzięki.
    </description>
    <options>
      <offer externalID="146">
        <availability>1</availability>
        <onStock>1</onStock>
        <detail_name> Białe Rozmiar L</detail_name>
      </offer>
    </options>
  </product>
</products>
```

```
</offer>
```

```
<offer externalID="145">  
<availability>1</availability>  
<onStock>1</onStock>  
<detail_name> Biały Rozmiar XL</detail_name>  
</offer>
```

```
<offer externalID="147">  
<availability>1</availability>  
<onStock>0</onStock>  
<detail_name> Biały Rozmiar M</detail_name>  
</offer>
```

```
<offer externalID="148">  
<availability>1</availability>  
<onStock>1</onStock>  
<detail_name> Biały Rozmiar S</detail_name>  
</offer>
```

```
<offer externalID="18593">  
<availability>1</availability>  
<onStock>1</onStock>  
<detail_name> Czarny Rozmiar L</detail_name>  
</offer>
```

```
<offer externalID="18594">  
<availability>1</availability>  
<onStock>2</onStock>  
<detail_name> Czarny Rozmiar XL</detail_name>  
</offer>
```

```
<offer externalID="18592">  
<availability>1</availability>  
<onStock>0</onStock>  
<detail_name> Czarny Rozmiar M</detail_name>  
</offer>
```

```
<offer externalID="144">  
<availability>1</availability>  
<onStock>1</onStock>  
<detail_name> Czarny Rozmiar S</detail_name>  
</offer>
```

```
</options>
```

```
</product>
```

```
</products>
```

Szczegółowe wyjaśnienie pliku XML zawiera poniższa tabela

Opis pól pliku XML

Nazwa pola	Opis	Typ danych
name	Nazwa modelu / produktu	string
brand	Marka produktu	string
category	Kategoria w jakiej znajduje się produkt. Listę możliwych kategorii można znaleźć na końcu tego dokumentu.	string
images	Node zawierający nody <code>image url</code> . Każdy z nich zawiera adres URL do największej możliwej wersji zdjęcia produktu w naszym systemie.	NODE
price	Cena brutto dla klienta końcowego	float
description	Opis produktu w języku polskim.	string
options	Node zawierający elementy z informacjami o opcjach produktu oraz ich dostępności.	NODE
offer	Node grupujący informacje o konkretnej opcji w jakiej występuje produkt. Parametr <code>externalID</code> identyfikuje jednoznacznie konkretną opcję konkretnego modelu (jest unikalny w skali całej oferty). Ten parametr służy do złożenia zamówienia.	
availability	Czas dostępności konkretnego towaru w dniach. Jeżeli dostępność wynosi 1 dzień, wtedy node <code>onStock</code> jest większy od 0 czyli mamy towar fizycznie na własnym magazynie. Każda inna wartość oznacza, że produkt musimy sprowadzić od naszego dostawcy.	number
onStock	Liczba sztuk w magazynie. Jeżeli wartość = 0 oznacza to, że produkt należy sprowadzić.	Number
detail_name	Nazwa opcji. Najczęściej jest to rozmiar lub rozmiar i kolor.	string

Plik XML jest generowany przez nasz system 2 godziny.
Istnieje również możliwość pobierania danych na żywo dla pojedynczego produktu.

http://www.matterhorn.pl/api/cennik_partner.php?shopId=gustowna.pl&productId=1001

Podany przykład w wyniku zwróci dokument XML dla produktu o ID 1001.

```
<products>
  <product id="1001">
    <name>Edyta</name>
    <brand>Violana</brand>
    <category>Podkoszulki/Topy</category>
    <images>
      <image url=http://www.matterhorn.pl/bigimg.php?id=1176</image_url>
    </images>
    <price>14.9</price>
    <description>
      Koszulka damska bez rękawków, na szerokich ramiączkach. Pod szyją okrągłe wycięcie.
      Wykonana ze 100 % bawełny.
    </description>
    <options>
      <offer externalID="12743">
        <availability>7</availability>
        <onStock>0</onStock>
        <detail_name> Biały Rozmiar L</detail_name>
      </offer>
      <offer externalID="12744">
        <availability>7</availability>
        <onStock>0</onStock>
        <detail_name> Biały Rozmiar XL</detail_name>
      </offer>
      <offer externalID="12742">
        <availability>7</availability>
        <onStock>0</onStock>
        <detail_name> Biały Rozmiar M</detail_name>
      </offer>
      <offer externalID="12741">
        <availability>7</availability>
        <onStock>0</onStock>
        <detail_name> Biały Rozmiar S</detail_name>
      </offer>
      <offer externalID="12747">
        <availability>7</availability>
        <onStock>0</onStock>
        <detail_name> Czarny Rozmiar L</detail_name>
      </offer>
      <offer externalID="12748">
        <availability>7</availability>
        <onStock>0</onStock>
        <detail_name> Czarny Rozmiar XL</detail_name>
      </offer>
      <offer externalID="12746">
        <availability>1</availability>
        <onStock>1</onStock>
        <detail_name> Czarny Rozmiar M</detail_name>
      </offer>
      <offer externalID="12745">
        <availability>7</availability>
        <onStock>0</onStock>
        <detail_name> Czarny Rozmiar S</detail_name>
      </offer>
    </options>
  </product>
</products>
```

Skrypty pomocnicze do osadzenia na stronie partnera.

Aby ułatwić integrację i zapewnić komunikację online system Matterhorn udostępnia kilka narzędzi pomocniczych dla partnerów, które mogą być pobierane automatycznie z serwera Matterhorn kiedy klient porusza się po sklepie partnera.

Wśród funkcji pomocniczych znaleźć można takie funkcjonalności jak:

- lista opcji produktu z informacją o czasie dostępności produktu
- wygenerowanie formularza do systemu platnosci.pl dla konkretnego zamówienia
- lista rodzajów przesyłek

Przy każdym wywołaniu funkcji pomocniczej trzeba przekazywać parametry shopId oraz action – czyli nazwę funkcji. Lista funkcji jest dostępna poniżej.

Przykład użycia funkcji pomocniczej:

Pobieranie listy opcji produktu z informacją o czasie ich dostępności:

http://www.matterhorn.pl/api/ajaxFunctions.php?&shopId=gustowna.pl&action=product_options&id_art=1001

gdzie:

shopId – identyfikator sklepu

action – nazwa funkcji

id_art – parametry wymagany dla funkcji product_options określający ID produktu.

Wywołanie w skrypcie PHP

```
<html>
  <body>
    <!-- formularz dodawania produktu do koszyka
    <form action='koszyk.php'>
      <select name='opcja'>
        <?=  

        $url = 'http://www.matterhorn.pl/api/ajaxFunctions.php?  

        &shopId=gustowna.pl&action=product_options&id_art=1001';  

        echo file_get_contents($url);  

        ?>
      </select>
    </form>
```

</body>

Nazwa funkcji	Opis	Wymagane parametry dodatkowe
product_options	<p>Zwraca listę opcji w jakich występuje produkt z informacją o czasie jego dostępności.</p> <p>Przykładowy wynik:</p> <pre data-bbox="467 857 1145 1003"><option value='12743'> Biały Rozmiar L - 7 dni</option> <option value='12744'> Biały Rozmiar XL - 7 dni</option> <option value='12742'> Biały Rozmiar M - 7 dni</option> <option value='12741'> Biały Rozmiar S - 7 dni</option> <option value='12747'> Czarny Rozmiar L - 7 dni</option> <option value='12748'> Czarny Rozmiar XL - 7 dni</option> <option value='12746'> Czarny Rozmiar M - wysyłka dzisiaj (ostatnie 1 szt)</option> <option value='12745'> Czarny Rozmiar S - 7 dni</option></pre> <p>Każdy element option w parametrze VALUE zawiera identyfikator UID konkretnej opcji i koloru, który ma zostać użyty w funkcjo zamawiania produktów.</p> <p>Przykładowe wywołanie:</p> <p>http://www.matterhorn.pl/api/ajaxFunctions.php?&shopId=gustowna.pl&action=product_options&id_art=1001</p>	id_art – identyfikator produktu
payment_form	<p>Funkcja zwraca formularz HTML odsyłający do systemu platnosci.pl dla konkretnego zamówienia.</p> <p>Wynik tej funkcji powinien zostać wywołany, kiedy po złożeniu zamówienia parametr show_payment będzie miał wartość TRUE (klient wybrał płatność z góry na konto).</p> <p>W formularzu zawarte są dane osobowe klienta dlatego wymagane jest podobnie jak w przy składaniu zamówienia parametru z sumą kontrolną.</p> <p>Przykład wywołania funkcji dla zamówienia nr 52531:</p> <p>http://www.matterhorn.pl/api/ajaxFunctions.php?&shopId=gustowna.pl&action=payment_form&orderid=52513&checksum=cb75ffd504c4c74cc9552450597766de</p>	orderid – numer zamówienia, którego formularz ma dotyczyć checksum = wynik funkcji MD5 dla ciągu znaków: SECRET_WORD+orderid

Przykładowy wynik:

```
<form action="https://www.platnosci.pl/paygw/ISO/NewPayment" method="POST"
      name="payform">
<input type="hidden" name="pos_id" value="14281">
<input type="hidden" name="pos_auth_key" value="26nIGTw">
<input type="hidden" name="session_id" value="47649">
<input type="hidden" name="amount" value="1100">
<input type="hidden" name="desc" value="Gustowna.pl. Zamówienie w nr 52513">
<input type="hidden" name="client_ip" value="79.163.199.5">
<input type="hidden" name="js" value="0">
<input type="hidden" name="order_id" value="">
<input type="hidden" name="first_name" value="Jan">
<input type="hidden" name="last_name" value="Kowalski">
<input type="hidden" name="street" value="Jagiellońska">
<input type="hidden" name="street_hn" value="82">
<input type="hidden" name="street_an" value="2">
<input type="hidden" name="city" value="Warszawa">
<input type="hidden" name="post_code" value="03-301">
<input type="hidden" name="country" value="pl">
<input type="hidden" name="email" value="tjaromin@gmail.com">
<input type="hidden" name="phone" value="0222100303">
<input type="hidden" name="language" value="pl">
<input type="submit" value="OK">
</form>
<script language="JavaScript" type="text/javascript">
<!--
//document.forms["
payform"].js.value=1;
-->
</script>
```

WAŻNA INFORMACJA!

Klient po dokonaniu płatności zostanie przekierowany na adres sklepu.

W przypadku udanej płatności klient powróci na stronę:

http://www.TWOJSKLEP.pl/?payment=ok&orderid=NUMER_ZAMOWIENIA

a przypadku nieudanej płatności zostanie przekierowany na stronę

http://www.TWOJSKLEP.pl/?payment=error&orderid=NUMER_ZAMOWIENIA

System sam odnotowuje informacje z platnosci.pl dlatego adres powrotu użytkownika powinien służyć wyłącznie do zdecydowania jaką informację wyświetlić finalnie użytkownikowi (podziękowanie za zapłatę lub poinformowanie o nieudanej płatności i ew. wyświetlenie ponownie formularza).

shipping_options

Funkcja zwraca listę opcji płatności dla konkretnego kraju.

Przykład wywołania:

Kraj – dwuliterowy kod kraju którego dotyczy wysyłka.

	<p>http://www.matterhorn.pl/api/ajaxFunctions.php?&shopId=gustowna.pl&action=shipping_options&kraj=pl</p> <p>Wynik:</p> <pre><option value='78'>POCZTA POLSKA - płatne z góry (przelew lub karta) 9,80,- </option> <option value='91'>Kurier GLS - Płatne z góry (przelew lub karta) 13,00,- </option> <option value='95'>POCZTA POLSKA- płatne przy odbiorze 12,00,-</option> <option value='105'>Kurier GLS - płatne przy odbiorze 19,00,-</option></pre>	
shipping_options_xml	<p>Funkcja zwraca listę opcji płatności dla konkretnego kraju.</p> <p>Przykład wywołania:</p> <p>http://www.matterhorn.pl/api/ajaxFunctions.php?&shopId=gustowna.pl&action=shipping_options_xml&kraj=pl</p> <p>Wynik:</p> <pre><shipping_options> <option id="78" price="9.8" min_order_value="0" payment_in_advance="N">POCZTA POLSKA - płatne z góry (przelew lub karta)</option> - <option id="91" price="13" min_order_value="0" payment_in_advance="N"> Przesyłka kurierska- płatne z góry (przelew lub karta) </option> <option id="95" price="12" min_order_value="0" payment_in_advance="Y">POCZTA POLSKA- płatne przy odbiorze</option> <option id="105" price="15" min_order_value="0" payment_in_advance="Y">Przesyłka kurierska - płatne przy odbiorze</option> </shipping_options></pre> <p>każdy tag option zawiera parametry:</p> <p>id – identyfikator przesyłki wysyłany do zamówieniach price – cena przesyłki brutto min_order_value – min wartość zamówienia od której można użyć danego rodzaju przesyłki payment_in_advance – wartość Y oznacza płatność z góry, wartość N oznacza płatność pobraniową.</p>	kraj – dwuliterowy kod kraju którego dotyczy wysyłka.

<p>order_status</p>	<p>Funkcja zwraca informacje o stanie zamówienia w postaci dokumentu XML.</p> <p>Przykład wywołania: http://www.matterhorn.pl/api/ajaxFunctions.php?&shopId=gustowna.pl&action=order_status&orderid=52513&checksum=cb75ffd504c4c74cc9552450597766de</p> <p>Przykładowy wynik:</p> <pre><order_status orderid='53660'> <shipping_number></shipping_number> <shipping_date>0000-00-00</shipping_date> <amount>52.9</amount> <deadline>2010-06-03</deadline> <status>4</status> <status_description>Zamowienie opłacone lub za pobraniem, jeszcze nie wyslane.</status_description> <order_positions> <order_position> <position_id>162607</position_id> <product_name>Model 25660</product_name> <product_option> Biały 75 D</product_option> <quantity>1</quantity> <total>52.9</total> <status>W realizacji</status> <status_id>0</status_id> </order_position> </order_positions> <user_id>123456</user_id> </order_status></pre> <p>Znaczenie pól:</p> <ol style="list-style-type: none"> shipping_number – numer przesyłki shipping_date – data wysłania paczki amount – suma zamówienia w zł deadline – max. termin realizacji zamówienia status – id statusu w jakim znajduje się zamówienie <ul style="list-style-type: none"> 1 – zamówienie zrealizowane (wysłane i opłacone) 2 – zamówienie potwierdzone (status ten nie jest konieczny do zrealizowania zamówienia) 3 – zamówienie anulowane 4 – zamówienie zapłacone lub wybrana została płatność za pobraniem ale jeszcze nie wysłane 5 – zamówienie wysłane za pobraniem status_description – status opisowy order_positions – node zawierający informacje o pozycjach zamówienia i ich stanie user_id – identyfikator klienta w systemie Matterhorn 	<p>orderid – numer zamówienia, którego formularz ma dotyczyć</p> <p>checksum = wynik funkcji MD5 dla ciągu znaków: SECRET_WORD+orderid</p>
<p>customer_history</p>	<p>Funkcja zwraca informacje o złożonych zamówieniach, wpłatach i bilansie klienta w sklepie internetowym.</p> <p>Funkcję powinno się zaimplementować na stronie panelu dla klienta aby mógł on zapoznać się ze swoim bilansem finansowym, statusem swoich zamówień oraz z historią wpłat.</p>	<p>customerId – identyfikator klienta w systemie Matterhorn. Można go uzyskać np funkcją GetUserId</p>

	<p>Przykładowe wywołanie:</p> <p>http://www.matterhorn.pl/api/ajaxFunctions.php? &shopId=gustowna.pl&action=customer_history&customerId=125175&c checksum=cb8e76090q6f0sddw98f394d19702f087</p> <p>Przykładowy wynik:</p> <pre><customer_history> <payments> <payment> <date>2010-07-19</date> <amount>103.62</amount> </payment> </payments> <orders> <order> <id>57725</id> <date>2010-07-18 00:41:56</date> <amount>103.62</amount> </order> </orders> <balance> <sum_orders>103.62</sum_orders> <sum_payments>103.62</sum_payments> <balance>0</balance> </balance> </customer_history></pre>	<p>checksum = wynik funkcji MD5 dla ciągu znaków: SECRET_WORD+customerId</p>
<p>getUserId</p>	<p>Funkcja zwraca identyfikator klienta z systemu Matterhorn na podstawie loginu i hasła. Pozwala na implementację funkcji logowania na stronie partnera.</p> <p>Przykładowe wywołanie:</p> <p>http://matterhorn.pl/api/ajaxFunctions.php? &shopId=gustowna.pl&action=getUserId&customerLogin=joanna.kowals ka@poczta.pl&customerPassword=34df2sw</p> <p>Przykładowa odpowiedź:</p> <pre><customer_id>125175</customer_id></pre>	<p>customerLogin – adres email lub nazwisko użytkownika.</p> <p>customerPassword – hasło użytkownika zdefiniowane w systemie Matterhorn. Jeżeli klient nie zdefiniował hasła zakładając konto, system wygenerował automatycznie i poinformował klienta w mailu powitalnym.</p>
<p>checkUser</p>	<p>Funkcja pozwalająca sprawdzić, czy użytkownik istnieje w systemie na podstawie podanego adresu email.</p> <p>System Matterhorn daje dowolność twórcy aplikacji, czy chce zablokować tworzenie wielu kont dla tej samej osoby dla czystości bazy danych, czy też ułatwić zakupy i nie blokować zakładania wielu kont pod tym samym adresem email.</p> <p>Jeżeli chcemy sprawdzić na podstawie adresu email, czy klient już istnieje w naszym systemie, powinniśmy wywołać w pierw funkcję checkUser i na podstawie zwróconej informacji zdecydować, czy wymagane będzie zalogowanie użytkownika, czy też jego rejestracja.</p>	<p>Email – adres email klienta.</p>

	<p>Przykładowe wywołanie:</p> <p>http://matterhorn.pl/api/ajaxFunctions.php?&shopId=gustowna.pl&action=checkUser&email=test@test.pl</p> <p>Przykładowa odpowiedź:</p> <p><IS_REGISTERED>TRUE</IS_REGISTERED> w przypadku, jeżeli klient jest już zarejestrowany w sklepie lub <IS_REGISTERED>FALSE</IS_REGISTERED> gdy użytkownik nie ma jeszcze konta.</p>	
getUserData	<p>Funkcja zwraca dane adresowe klienta.</p> <p>Przykładowe wywołanie:</p> <p>http://www.matterhorn.pl.com/api/ajaxFunctions.php?&shopId=gustowna.pl&action=getUserData&customerId=125175&checksum=cb8e76090d6f052c98f3922349702f087</p> <p>Przykładowa odpowiedź:</p> <pre><customer_data> <id>125175</id> <firstname>Joanna</firstname> <surname>Kowalska</surname> <email>asia.kowalska@o22.pl</email> <street>Mickiewicza</street> <house_nr>10</house_nr> <local_nr>1</local_nr> <city>Gdańsk</city> <postcode>62-095</postcode> <country>PL</country> <phone>555333222</phone> </customer_data></pre>	<p>customerId – identyfikator klienta w systemie Matterhorn. Można go uzyskać np funkcją GetUserId</p> <p>checksum = wynik funkcji MD5 dla ciągu znaków: SECRET_WORD+customerId</p>
changeUserPassowrd	<p>Funkcja pozwala na zmianę hasła klienta. Można jej użyć do implementacji na stronie panelu klienta do zmiany hasła. Rekomendujemy w tym celu weryfikację hasła aby uniknąć pomyłek.</p> <p>Przykładowe wywołanie:</p> <p>http://www.matterhorn.pl/api/ajaxFunctions.php?&shopId=gustowna.pl&action=changeUserPassowrd&customerId=125175&checksum=cb2176090d6f052c98f394d19702f087&oldPassword=234234324&newPassword=ALA</p> <p>Przykładowa odpowiedź:</p> <p><OK>Hasło zostało zmienione.</OK></p> <p>lub w przypadku podania złego hasła:</p> <p><ERROR>Obecne hasło jest inne.</ERROR></p>	<p>customerId – identyfikator klienta w systemie Matterhorn. Można go uzyskać np funkcją GetUserId</p> <p>checksum - wynik funkcji MD5 dla ciągu znaków: SECRET_WORD+customerId</p> <p>oldPassword – obecne hasło użytkownika.</p> <p>newPassword – nowe hasło użytkownika.</p>
registerUser	<p>Funkcja pozwala na rejestrację nowego użytkownika w sklepie, używana kiedy chcemy założyć konto użytkownika bez składana zamówienia.</p> <p>Przykładowe wywołanie:</p>	<p>fdata – Imię użytkownika</p> <p>sdata – nazwisko</p>

	<p>http://www.matterhorn.pl/api/ajaxFunctions.php?&shopId=gustowna.pl&action=registerUser&checksum=8450dc20f33eb6b4ce5bd06dc7b14a23&fdata=Joanna&sdata=Kowalska&ulica=Mickiewicza&nr_domu=4&miasto=Warszawa&kod=00-012&email=joanna.kowalska@poczta.com&kraj=PL&password=h4s10</p> <p>Przykładowa odpowiedź:</p> <p><OK>123456</OK></p> <p>Numer w odpowiedzi to identyfikator klienta (customerId), który można użyć np. w procesie zamówienia lub funkcjach panelu klienta.</p> <p>lub w przypadku błędu:</p> <p><ERROR>Wystąpił problem przy tworzeniu konta</ERROR></p>	<p>użytkownika</p> <p>ulica – ulica, plac aleja bez numeru domu i lokalu</p> <p>nr_domu – numer budynku adresu zamieszkania klienta</p> <p>nr_lokalu – nie obowiązkowo</p> <p>miasto</p> <p>kod – kod pocztowy</p> <p>kraj – dwuliterowy kod kraju, np. PL</p> <p>email – adres email klienta</p> <p>password – hasło ustanowione przez klienta w procesie rejestracji.</p> <p>checksum - wynik funkcji MD5 dla ciągu znaków: SECRET_WORD+fdata+sdata</p>
--	---	---