

Odbieranie płatności za pomocą PayPal

Zaróbmy trochę pieniędzy! Twoja witryna e-commerce potrzebuje sposobu na otrzymywanie płatności od klientów. Preferowanym rozwiązaniem dla firm o ugruntowanej pozycji jest otwarcie konta sprzedawcy, ale wiele małych firm decyduje się na rozpoczęcie od rozwiązania, które jest prostsze do wdrożenia, w którym nie muszą samodzielnie przetwarzać danych karty kredytowej lub płatności. Wiele firm i witryn internetowych może pomóc osobom fizycznym lub małym firmom, które nie mają zasobów do przetwarzania kart kredytowych i przelewów. Firmy te mogą być wykorzystywane do pośredniczenia w płatnościach między firmami internetowymi a ich klientami. Wiele z tych firm zajmujących się przetwarzaniem płatności jest stosunkowo nowych, a obsługa danych finansowych każdej osoby jest bardzo delikatna. Dodatkowo szybkie wyszukiwanie w Internecie pozwoli wygenerować raporty zarówno od zadowolonych, jak i niezadowolonych klientów dla prawie wszystkich tych firm. Z tych powodów nie polecamy żadnej konkretnej firmy zewnętrznej. Zamiast tego wymieniamy niektóre firmy, które obecnie świadczą te usługi, a następnie przedstawiono niektóre z ich funkcji na przykładzie systemu PayPal. Dowiesz się, jak zintegrować PayPal z TShirtShop w pierwszych dwóch fazach rozwoju. W tej Części:

- Dowiesz się, jak utworzyć nowe konto Standard płatności za pośrednictwem witryny PayPal.
- Dowiesz się, jak zintegrować system PayPal w pierwszej fazie rozwoju, w której potrzebny jest koszyk na zakupy i niestandardowy mechanizm realizacji transakcji.
- Dowiesz się, jak zintegrować system PayPal w fazie II rozwoju, w której będziesz mieć swój własny koszyk, dzięki czemu poprowadzisz użytkownika bezpośrednio do strony płatności.

Biorąc pod uwagę dostawców usług płatności internetowych

Spójrz na tę listę witryn internetowych dostawców usług płatności internetowych. To zróżnicowana grupa, a każda z nich ma swoje zalety. Niektórzy dostawcy przekazują pieniądze od osoby do osoby, gdzie płatności muszą być weryfikowane ręcznie; inne oferują zaawansowaną integrację z Twoją witryną internetową. Niektórzy pracują w dowolnym miejscu na świecie, podczas gdy inni pracują tylko w jednym kraju. Poniższa lista nie jest kompletna. Możesz znaleźć wiele innych takich firm, wyszukując w Google „dostawcy usług płatności internetowych”.

- 2Zamówienie: <http://www.2checkout.com>
- AnyPay: <http://www.anypay.com>
- CCNow: <http://www.ccnw.com>
- Przelew elektroniczny: <http://www.electronictransfer.com>
- Moneybookers: <http://www.moneybookers.com>
- Karty MultiCard: <http://www.multicards.com>
- Płatność przez Internet: <http://www.paybyweb.com>
- Paymate: <https://www.paymate.com.au>
- PayPal: <http://www.paypal.com>
- PaySystems: <http://www.paysystems.com>
- ProPay: <http://www.propay.com>

- QuickPayPro: <http://www.quickpaypro.com>
- WorldPay: <http://www.worldpay.com>

Oprócz popularności, PayPal oferuje usługi, które bardzo dobrze pasują do naszej witryny internetowej w pierwszych dwóch fazach rozwoju. PayPal jest dostępny w wielu krajach - najbardziej aktualną listę można znaleźć na stronie <http://www.paypal.com>. W pierwszej fazie rozwoju (obecnej fazie), w której dostępny jest tylko katalog produktów, który można przeszukiwać, zaledwie kilkoma wierszami kodu HTML, PayPal umożliwia dodanie koszyka na zakupy z funkcją kasy. W drugiej fazie rozwoju, w której zaimplementujesz własny koszyk na zakupy, system PayPal ma funkcję o nazwie Zakupy pojedynczych przedmiotów, której można użyć do wysłania odwiedzającego bezpośrednio na stronę płatności bez pośredniego koszyka na zakupy. Aby zapoznać się z podsumowaniem funkcji oferowanych przez PayPal, przejdź w przeglądarce na stronę <http://www.paypal.com> i kliknij łącze Usługi handlowców. Ta strona zawiera kilka innych przydatnych łączy, które pokażą Ci główne funkcje dostępne w systemie PayPal.

Pierwsze kroki z PayPal

Prawdopodobnie najlepszy opis tej usługi znajduje się na jej stronie internetowej: „PayPal to system oparty na koncie, który pozwala każdemu, kto ma adres e-mail, bezpiecznie wysłać i odbierać płatności online za pomocą karty kredytowej lub konta bankowego”. Zamiast płacić bezpośrednio sprzedawcy, odwiedzający płaci PayPal za pomocą karty kredytowej lub konta bankowego. Następnie firma handlowa używa swojego konta PayPal, aby uzyskać pieniądze otrzymane od klientów. W chwili pisania tego tekstu utworzenie nowego konta PayPal nie wiąże się z żadnymi kosztami, a usługa jest bezpłatna dla kupującego. Opłaty związane z otrzymywaniem pieniędzy są pokazane na stronie http://www.paypal.com/cgi-bin/webscr?cmd=_display-fees-outside.

LINKI I ZASOBY PAYPAL

Sprawdź te zasoby, jeśli potrzebujesz więcej informacji niż ten krótki rozdział:

- Podręcznik integracji Płatności za pośrednictwem witryny internetowej zawiera informacje zawarte wcześniej w oddzielnych podręcznikach, takich jak podręcznik Koszyk i Podręcznik natychmiastowego powiadomienia o płatnościach. Pobierz go na :

https://www.paypal.com/en_US/pdf/PP_WebsitePaymentsStandard_IntegrationGuide.pdf.

- Sieć programistów PayPal jest oficjalnym źródłem informacji dla programistów PayPal, do którego można uzyskać dostęp pod adresem <https://www.paypal.com/pdn>.
- PayPalDev, według witryny, jest niezależnym forum dla programistów PayPal. Uzyskaj do niego dostęp pod adresem <http://www.paypaldev.org/>. Znajdziesz tu również liczne łącza do różnych zasobów PayPal.

W poniższym ćwiczeniu utworzysz nowe konto PayPal i zintegrujesz je z TShirtShop (kroki tworzenia konta PayPal są również opisane bardziej szczegółowo we wspomnianych wcześniej podręcznikach PayPal).

Ćwiczenie: Tworzenie konta PayPal

1. Przejdź do <http://www.paypal.com> za pomocą swojej ulubionej przeglądarki internetowej.
2. Kliknij łącze Zarejestruj się.

3. PayPal obsługuje trzy typy kont: Osobiste, Premier i Biznes. Aby otrzymywać płatności kartą kredytową, musisz otworzyć konto Premier lub Business. Wybierz swój kraj z pola kombi i kliknij Kontynuuj.

4. Podaj wszystkie wymagane informacje, a otrzymasz wiadomość e-mail z prośbą o ponowne odwiedzenie witryny PayPal w celu potwierdzenia wprowadzonych danych.

Jak to działa: konto PayPal

Po skonfigurowaniu konta PayPal podany adres e-mail będzie Twoim identyfikatorem PayPal. W ramach usługi PayPal dostępnych jest wiele funkcji - ponieważ strona jest łatwa w obsłudze, a wiele funkcji nie wymaga wyjaśnień, nie będziemy tutaj wszystkiego opisywać. Pamiętaj, że witryny takie jak PayPal są przeznaczone dla Twojej firmy, więc z przyjemnością odpowiedzą na wszelkie Twoje pytania. Zobaczmy teraz, jak faktycznie możesz korzystać z nowego konta w witrynie internetowej.

Integracja koszyka na zakupy PayPal i realizacji transakcji

Pierwszym etapem jest integracja funkcji koszyka i kasy z PayPal. W drugiej fazie rozwoju, po utworzeniu własnego koszyka na zakupy, będziesz musiał polegać wyłącznie na mechanizmie realizacji transakcji PayPal. Aby akceptować płatności, musisz dodać dwa ważne elementy do części interfejsu użytkownika witryny: przyciski Dodaj do koszyka dla każdego produktu oraz przycisk Wyświetl koszyk gdzieś na stronie. PayPal sprawia, że dodanie tych przycisków to bułka z masłem. Funkcjonalność tych przycisków jest realizowana za pomocą bezpiecznych łączy do witryny internetowej PayPal, które można zaimplementować jako formularze lub specjalnie spreparowane adresy URL. Przyjrzyjmy się obu metodom. Poniżej znajduje się przykład użycia formularzy reprezentujących przycisk Dodaj do koszyka dla produktu o nazwie Koszulka Jemioła, który kosztuje 17,99 USD:

```
<form target="paypal" action="https://www.paypal.com/cgi-bin/webscr"
method="post">
```

```
<input type="hidden" name="cmd" value="_cart" />
```

```
<input type="hidden" name="business" value="youremail@example.com" />
```

```
<input type="hidden" name="item_name" value="Mistletoe T-Shirt" />
```

```
<input type="hidden" name="on0" value="Size" />
```

```
<select name="os0">
```

```
<option value="S">S</option>
```

```
<option value="M">M</option>
```

```
<option value="L">L</option>
```

```
<option value="XL">XL</option>
```

```
<option value="XXL">XXL</option>
```

```
</select>
```

```
<input type="hidden" name="on1" value="Color" />
```

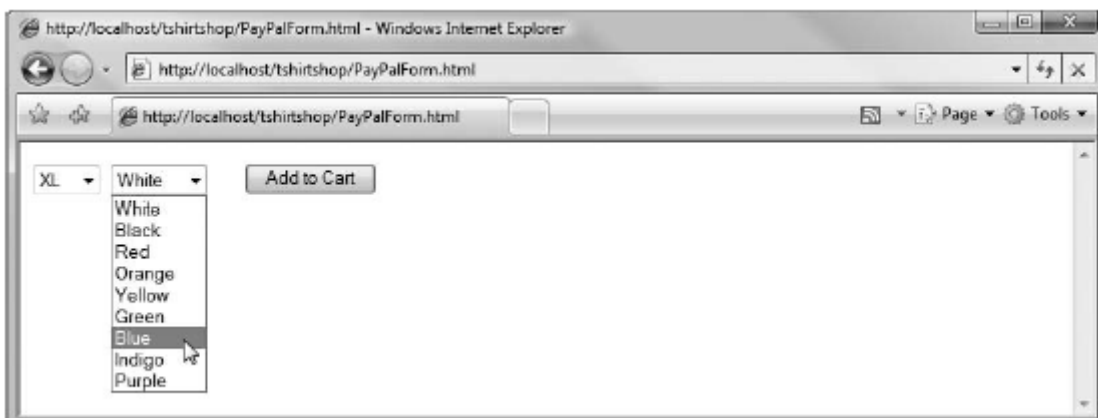
```
<select name="os1">
```

```

<option value=" White">White</option>
<option value=" Black">Black</option>
<option value="Red">Red</option>
<option value="Orange">Orange</option>
<option value="Yellow">Yellow</option>
<option value="Green">Green</option>
<option value="Blue">Blue</option>
<option value="Indigo">Indigo</option>
<option value="Purple">Purple</option>
</select>
<input type="hidden" name="amount" value="17.99" />
<input type="hidden" name="currency_code" value="USD" />
<input type="hidden" name="add" value="1" />
<input type="hidden" name="shopping_url"
value="http://www.example.com/mistletoe-t-shirt-p84/" />
<input type="hidden" name="return" value="http://www.example.com" />
<input type="hidden" name="cancel_return" value="http://www.example.com" />
<input type="submit" name="submit" value="Add to Cart" />
</form>

```

Jeśli załadujesz ten formularz w przeglądarce internetowej, otrzymasz formularz podobny do tego pokazanego na rysunku



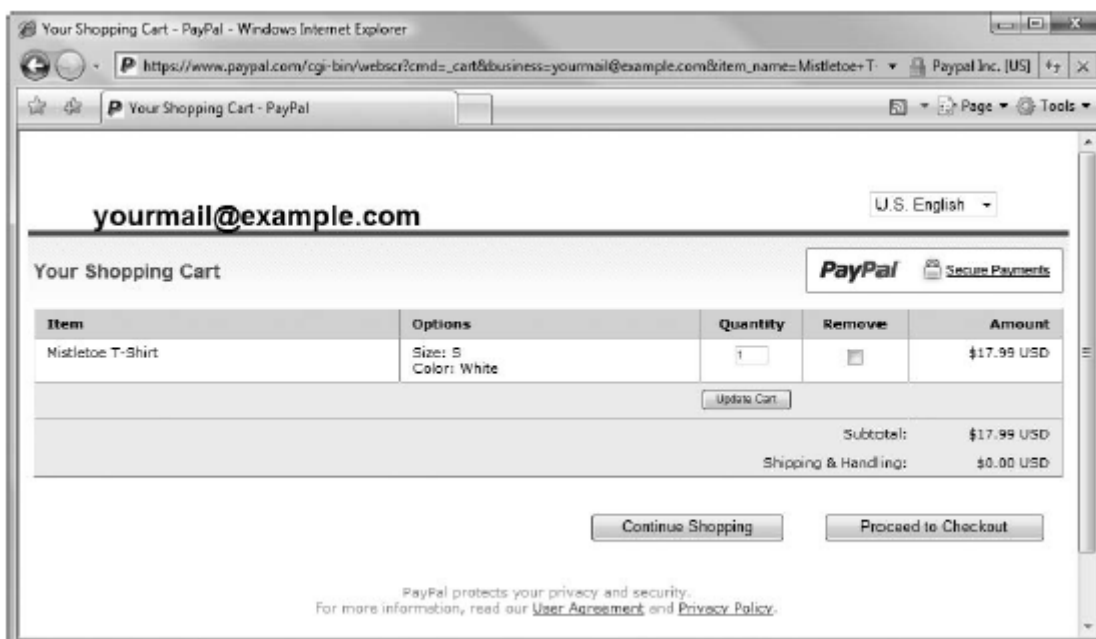
Wskazówka : chociaż nie będziemy ich używać w naszej witrynie, dobrze jest wiedzieć, że PayPal udostępnia generatory przycisków oparte na określonych przez Ciebie danych (takich jak nazwa produktu i cena produktu), dając blok kodu HTML podobny do pokazanego wcześniej.

Tak, wytworzenie linku Dodaj do koszyka jest tak proste! Pola formularza oprócz on0, os0, on1 i os1 mają predefiniowane znaczenie, a ich nazwy nie wymagają wyjaśnień. Elementy on0 i os0 oraz on1 i os1 mogą służyć do definiowania niestandardowych atrybutów produktu; w naszym przypadku użyliśmy ich do zdefiniowania rozmiaru i koloru. Pamiętaj, że PayPal akceptuje maksymalnie dwa atrybuty, więc nie możesz dodać ich więcej. Najważniejszym parametrem jest biznes, którym musi być adres e-mail podany podczas rejestracji konta PayPal (adres e-mail, na który zostaną przesłane pieniądze). Aby uzyskać więcej informacji, zapoznaj się z Przewodnikiem integracji standardowych płatności za pośrednictwem witryny PayPal. O tym, jak pokonać ten limit atrybutów, omówimy później. Alternatywnie możesz utworzyć adresy URL Dodaj do koszyka bez korzystania z formularzy. Jest to szczególnie przydatne, gdy przycisk Dodaj do koszyka jest już zawarty w formularzu, ponieważ HTML nie obsługuje formularzy zagnieżdżonych. Używając adresów URL Dodaj do koszyka zamiast formularzy, musisz określić wartości atrybutów jako parametry ciągu zapytania. Oto przykład:

```
https://www.paypal.com/cgi-bin/webscr?cmd=_cart&business=yourmail@example.com&item_name=Mistletoe+T-Shirt&on0=Size&os0=S&on1=Color&os1=White&amount=17.99&currency_code=USD&add=1&cancel_return=http://www.example.com&shopping_url=http://www.example.com/mistletoe-t-shirt-p84/&return=http://www.example.com
```

Uwaga : używając ciągu zapytania adresu URL do przekazywania informacji do serwera, należy starannie zakodować wartości parametrów. Na przykład, aby przekazać ciąg „Koszulka Jemioła” jako parametr adresu URL, należy zakodować spację, która nie jest dozwolonym znakiem adresu URL. Spacja może być zakodowana jako %20 lub +. PHP wygodnie udostępnia funkcję urlencode(), której możemy użyć do kodowania ciągów w celu włączenia adresu URL.

Wczytanie tego adresu URL w przeglądarce internetowej spowoduje przejście do strony koszyka PayPal, takiej jak ta pokazana na rysunku



Uwaga : Otrzymując płatność należy dokładnie sprawdzić, czy otrzymana kwota odpowiada sumie zamówionych produktów. Ponieważ wszystkie dane koszyka na zakupy są przekazywane za pośrednictwem formularzy lub adresów URL, każdemu bardzo łatwo jest dodać fałszywy produkt do koszyka lub istniejący produkt ze zmodyfikowaną ceną. Można to zrobić po prostu, tworząc jeden z

tych łączy PayPal Dodaj do koszyka i przechodząc do niego. Możesz przeczytać szczegółowy artykuł na temat tego problemu pod adresem <http://www.alphabetware.com/pptamper.asp>.

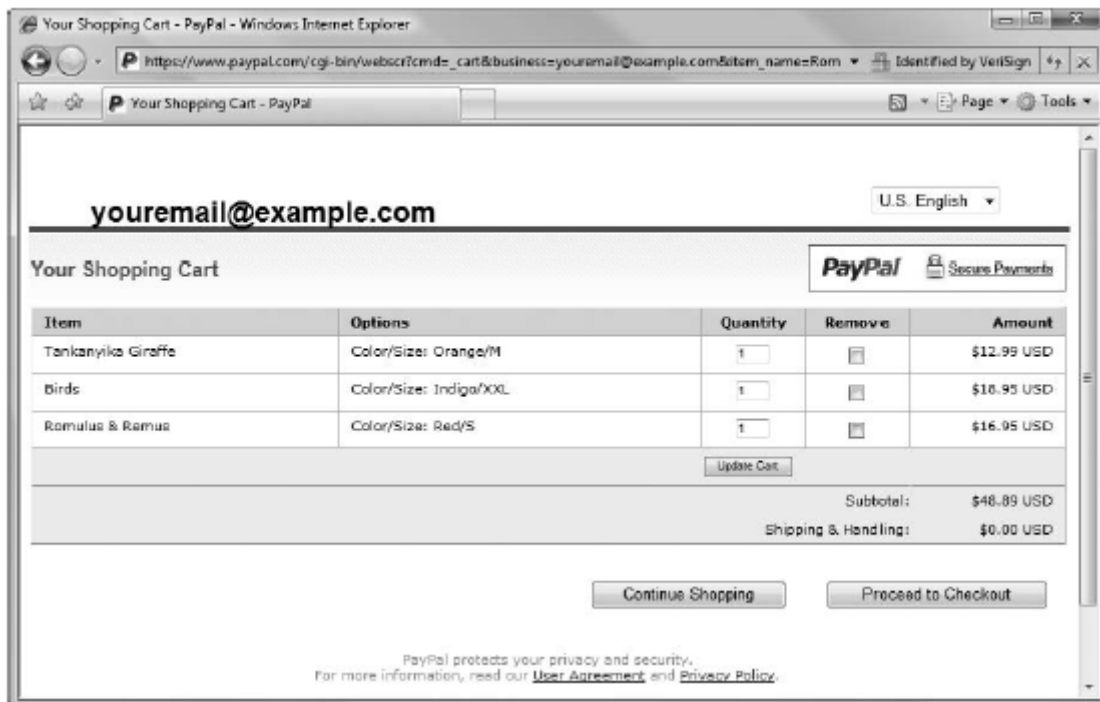
Korzystając z adresów URL koszyka na zakupy, takich jak PayPal, musisz kontrolować długość nazw i atrybutów produktów, chociaż jest mało prawdopodobne, że otrzymasz adresy URL, które są tak długie, że nie mogą być obsługiwane przez przeglądarki internetowe. Strona pod adresem <http://www.boutell.com/newfaq/misc/urllength.html> przedstawia interesującą analizę maksymalnych długości adresów URL akceptowanych przez najpopularniejsze przeglądarki internetowe. W przypadku TShirtShop użyjemy formularzy do utworzenia przycisku Wyświetl koszyk i adresów URL, aby utworzyć przyciski Dodaj do koszyka. Możesz zobaczyć te przyciski na rysunku



Aby przezwyciężyć ograniczenie PayPal dotyczące dwóch atrybutów na produkt, wszystkie wartości atrybutów produktów umieszczamy w jednym atrybucie PayPal. Wartością tego atrybutu PayPal byłby tekst zawierający atrybuty wybranego produktu, więc odwiedzający zobaczy te dane w koszyku PayPal. Spójrz w przyszłość na Rysunek, aby zobaczyć, jak to działa. Aby zaimplementować tę funkcję, ponieważ nie możemy użyć formularza PayPal do dodawania produktów, system działa w następujący sposób:

1. Nasze przyciski Dodaj do koszyka przesyłają formularz do naszej własnej aplikacji, do linku takiego jak <http://localhost/tshirtshop/add-product-65/>.
2. Adres URL jest przepisany przez mod_rewrite na adres URL taki jak `index.php?AddProduct=65`.
3. `index.php` tworzy atrybut PayPal, którego wartość sumuje atrybuty wybrane przez odwiedzającego dla produktu. Nowa wartość atrybutu będzie wyglądać jak `Rozmiar/Kolor: S/Biały`, ale zostanie automatycznie wygenerowana, aby pomieścić dowolną liczbę atrybutów.
4. Adres URL Dodaj do koszyka PayPal jest tworzony przy użyciu nowo utworzonego atrybutu.
5. Na koniec odwiedzający jest przekierowywany za pomocą kodu statusu 302 do nowego adresu URL.

Podsumowując te zasady: po kliknięciu przycisku Dodaj do koszyka przeglądarka odwiedzającego otrzyma przekierowanie 302 do adresu URL Dodaj do koszyka PayPal. Następnie załaduje ten adres URL, który przeniesie użytkownika do koszyka PayPal, jak pokazano na rysunku



Teraz, gdy masz już podstawową wiedzę na temat etapów, zaimplementujmy ten kod w ćwiczeniu.

Ćwiczenie: integracja koszyka PayPal i niestandardowej realizacji transakcji

1. Otwórz config.php z folderu include i dodaj następujące deklaracje stałych:

```
// PayPal configuration
define('PAYPAL_URL', 'https://www.paypal.com/cgi-bin/webscr');
define('PAYPAL_EMAIL', 'youremail@example.com');
define('PAYPAL_CURRENCY_CODE', 'USD');
define('PAYPAL_RETURN_URL', 'http://www.example.com');
define('PAYPAL_CANCEL_RETURN_URL', 'http://www.example.com');
```

Przeostroga : Upewnij się, że zastąpiłeś swójemail@example.com adresem e-mail podanym podczas tworzenia konta PayPal! Musisz podać poprawny adres e-mail, jeśli chcesz, aby pieniądze trafiły na Twoje konto! Zastąp też oba wystąpienia www.example.com adresem swojego sklepu e-commerce.

2. Otwórz Presentation/products_list.php i dodaj podświetlony kod na końcu metody init(), jak pokazano:

```
// Build links for product details pages
for ($i = 0; $i < count($this->mProducts); $i++)
{
```

```

$this->mProducts[$i]['link_to_product'] =
Link::ToProduct($this->mProducts[$i]['product_id']);
if ($this->mProducts[$i]['thumbnail'])
$this->mProducts[$i]['thumbnail'] =
Link::Build('product_images/' . $this->mProducts[$i]['thumbnail']);
// Create the Add to Cart link
$this->mProducts[$i]['link_to_add_product'] =
Link::ToAddProduct($this->mProducts[$i]['product_id']);
$this->mProducts[$i]['attributes'] =
Catalog::GetProductAttributes($this->mProducts[$i]['product_id']);
}
}

```

3. Otwórz products_list.tpl z folderu prezentacji/szablonów i zmodyfikuj go, jak pokazano:

```

<p class="section">
Price:
{if $obj->mProducts[k].discounted_price != 0}
<span class="old-price">{$obj->mProducts[k].price}</span>
<span class="price">{$obj->mProducts[k].discounted_price}</span>
{else}
<span class="price">{$obj->mProducts[k].price}</span>
{/if}
</p>
{* The Add to Cart form *}
<form class="add-product-form" target="_self" method="post"
action="{ $obj->mProducts[k].link_to_add_product }">
{* Generate the list of attribute values *}
<p class="attributes">
{* Parse the list of attributes and attribute values *}
{section name=l loop=$obj->mProducts[k].attributes}
{* Generate a new select tag? *}
{if $smarty.section.l.first ||

```



```

$Obj->mProducts[k].attributes[l].attribute_name !==
$Obj->mProducts[k].attributes[l.index_prev].attribute_name}
{$Obj->mProducts[k].attributes[l].attribute_name}:
<select name="attr_{$Obj->mProducts[k].attributes[l].attribute_name}">
{/if}
{* Generate a new option tag *}
<option value="{ $Obj->mProducts[k].attributes[l].attribute_value}">
{$Obj->mProducts[k].attributes[l].attribute_value}
</option>
{* Close the select tag? *}
{if $smarty.section.l.last ||
$Obj->mProducts[k].attributes[l].attribute_name !==
$Obj->mProducts[k].attributes[l.index_next].attribute_name}
</select>
{/if}
{/section}
</p>
{* Add the submit button and close the form *}
<p>
<input type="submit" name="submit" value="Add to Cart" />
</p>
</form>
</td>
{if $smarty.section.k.index % 2 != 0 && !$smarty.section.k.first ||
$smarty.section.k.last}
</tr>
{/if}

```

4. W Presentation/link.php zmodyfikuj metodę CheckRequest() zaraz na początku, jak pokazano:

```

// Redirects to proper URL if not already there
public static function CheckRequest()
{

```

```

$proper_url = '';
if (isset($_GET['Search']) || isset($_GET['SearchResults']) ||
isset($_GET['AddProduct']))
{
return ;
}

```

5. Ponadto w pliku link.php dodaj następującą metodę na końcu klasy Link:

```

// Create an Add to Cart link
public static function ToAddProduct($productId)
{
return self::Build('index.php?AddProduct=' . $productId);
}

```

6. Teraz otwórz product.php z folderu prezentacji i dodaj podświetlony kod do metody init() klasy Product:

```

$this->mProduct['attributes'] =
Catalog::GetProductAttributes($this->mProduct['product_id']);
$this->mLocations = Catalog::GetProductLocations($this->_mProductId);
// Create the Add to Cart link
$this->mProduct['link_to_add_product'] =
Link::ToAddProduct($this->_mProductId);
// Build links for product departments and categories pages
for ($i = 0; $i < count($this->mLocations); $i++)
{
$this->mLocations[$i]['link_to_department'] =
Link::ToDepartment($this->mLocations[$i]['department_id']);
}

```

7. W product.tpl z folderu prezentacja/szablony dodaj podświetlony kod, który dodaje przycisk Dodaj do koszyka na stronie produktu:

```

<p class="section">
Price:
{if $obj->mProduct.discounted_price != 0}
<span class="old-price">{$obj->mProduct.price}</span>
<span class="price">{$obj->mProduct.discounted_price}</span>

```

```

{else}
<span class="price">{$obj->mProduct.price}</span>
{/if}
</p>
{* The Add to Cart form *}
<form class="add-product-form" target="_self" method="post"
action="{$obj->mProduct.link_to_add_product}">
{* Generate the list of attribute values *}
<p class="attributes">
{* Parse the list of attributes and attribute values *}
{section name=k loop=$obj->mProduct.attributes}
{* Generate a new select tag? *}
{if $smarty.section.k.first ||
$obj->mProduct.attributes[k].attribute_name !==
$obj->mProduct.attributes[k.index_prev].attribute_name}
{$obj->mProduct.attributes[k].attribute_name}:
<select name="attr_{$obj->mProduct.attributes[k].attribute_name}">
{/if}
{* Generate a new option tag *}
<option value="{$obj->mProduct.attributes[k].attribute_value}">
{$obj->mProduct.attributes[k].attribute_value}
</option>
{* Close the select tag? *}
{if $smarty.section.k.last ||
$obj->mProduct.attributes[k].attribute_name !==
$obj->mProduct.attributes[k.index_next].attribute_name}
</select>
{/if}
{/section}
</p>
{* Add the submit button and close the form *}

```

```

<p>
<input type="submit" name="submit" value="Add to Cart" />
</p>
</form>
{if $obj->mLinkToContinueShopping}
<a href="{ $obj->mLinkToContinueShopping}">Continue Shopping</a>
{/if}

```

8. W store_front.php z folderu prezentacji dodaj podświetlony kod do klasy StoreFront, jak pokazano:

```

// Page title
public $mPageTitle;
// PayPal continue shopping link
public $mPayPalContinueShoppingLink;
// Class constructor
public function __construct()
{
$this->mLinkToIndex = Link::ToIndex();
}
public function init()
{
// Create "Continue Shopping" link for the PayPal shopping cart
if (!isset($_GET['AddProduct']))
{
/* Store the current request needed for the paypal
continue shopping functionality */
$_SESSION['paypal_continue_shopping'] =
Link::Build(str_replace(VIRTUAL_LOCATION, "",
$_SERVER['REQUEST_URI']));
$this->mPayPalContinueShoppingLink =
$_SESSION['paypal_continue_shopping'];
}
// If Add to Cart was clicked, prepare PayPal variables

```

```

else
{
// Clean output buffer
ob_clean();
$product_id = 0;
// Get the product ID to be added to cart
if (isset ($_GET['AddProduct']))
$product_id = (int)$_GET['AddProduct'];
else
trigger_error('AddProduct not set');
$selected_attribute_groups = array ();
$selected_attribute_values = array ();
// Get selected product attributes if any
foreach ($_POST as $key => $value)
{
// If there are fields starting with "attr_" in the POST array
if (substr($key, 0, 5) == 'attr_')
{
// Get the selected attribute name and value
$selected_attribute_groups[] = substr($key, strlen('attr_'));
$selected_attribute_values[] = $_POST[$key];
}
}
// Get product info
$product = Catalog::GetProductDetails($product_id);
// Build the PayPal url to add the product to cart
$paypal_url = PAYPAL_URL . '?cmd=_cart&business=' . PAYPAL_EMAIL .
'&item_name=' . rawurlencode($product['name']);
if (count($selected_attribute_groups) > 0)
$paypal_url .= '&on0=' . implode('/', $selected_attribute_groups) .
'&os0=' . implode('/', $selected_attribute_values);

```

```

$paypal_url .=
'&amount=' . ($product['discounted_price'] == 0 ?
$product['price'] : $product['discounted_price']) .
'&currency_code=' . PAYPAL_CURRENCY_CODE . '&add=1' .
'&shopping_url=' .
rawurlencode($_SESSION['paypal_continue_shopping']) .
'&return=' . rawurlencode(PAYPAL_RETURN_URL) .
'&cancel_return=' . rawurlencode(PAYPAL_CANCEL_RETURN_URL);
// Redirect to the PayPal cart page
header('HTTP/1.1 302 Found');
header('Location: ' . $paypal_url);
// Clear the output buffer and stop execution
flush();
ob_flush();
ob_end_clean();
exit();
}
// Load department details if visiting a department
if (isset($_GET['DepartmentId']))

```

```
{
```

9. Otwórz store_front.tpl z folderu prezentacji/szablonów i dodaj kod dla przycisku Wyświetl koszyk, jak pokazano:

```

<div class="yui-b">
{include file="search_box.tpl"}
{include file="departments_list.tpl"}
{include file=$obj->mCategoriesCell}
<div class="view-cart">
<form target="_self" method="post"
action="{$_smarty.const.PAYPAL_URL}">
<input type="hidden" name="cmd" value="_cart" />
<input type="hidden" name="business"

```

```

value="{\$smarty.const.PAYPAL_EMAIL}" />
<input type="hidden" name="display" value="1" />
<input type="hidden" name="shopping_url"
value="{\$obj->mPayPalContinueShoppingLink}" />
<input type="hidden" name="return"
value="{\$smarty.const.PAYPAL_RETURN_URL}" />
<input type="hidden" name="cancel_return"
value="{\$smarty.const.PAYPAL_CANCEL_RETURN_URL}" />
<input type="submit" name="view_cart" value="View Cart" />
</form>
</div>
</div>

```

10. Dodaj następujące style do pliku styles/tshirtshop.css:

```
div.yui-b div form.add-product-form
```

```
{
margin: 0;
padding: 0;
}
```

```
.add-product-form p
```

```
{
margin: 0;
padding: 0 0 10px 0;
}
```

```
.view-cart
```

```
{
padding: 10px;
text-align: center;
}
```

11. Załaduj TShirtShop w przeglądarce i kliknij jeden z przycisków Dodaj do koszyka lub kliknij przycisk Wyświetl koszyk. Powinieneś otrzymać koszyk PayPal, który wygląda jak na rysunku powyżej. Poeksperymentuj z koszykiem PayPal, aby sprawdzić, czy działa zgodnie z reklamą.

Jak to działa: integracja z PayPal

Napisałeś sporo kodu, ale teraz wszyscy Twoi goście to potencjalni klienci! Mogą korzystać z koszyka PayPal, aby kupować Twoje produkty! Aby wdrożyć tę funkcję, najpierw utworzyłeś przyciski Dodaj do koszyka na listach produktów. Zmodyfikowałeś prezentację/

products_list.php, aby dodać adresy URL Dodaj do koszyka do każdego produktu:

```
$this->mProducts[$i]['link_to_add_product'] =
```

```
Link::ToAddProduct($this->mProducts[$i]['product_id']);
```

Then you modified the template file by adding the buttons:

```
<form class="add-product-form" target="_self" method="post"
```

```
action="{ $obj->mProducts[k].link_to_add_product}">
```

```
...
```

```
<p>
```

```
<input type="submit" name="submit" value="Add to Cart" />
```

```
</p>
```

```
</form>
```

Wdrożyłeś podobne zmiany do product.php i product.tpl, aby utworzyć przyciski Dodaj do koszyka na stronach szczegółów produktu. Aby zobaczyć, co się dzieje po kliknięciu jednego z tych przycisków, użyliśmy wtyczki Firefox LiveHTTPHeaders. Na rysunku 9-5 podświetliliśmy adres URL ładowany po kliknięciu przycisku oraz adres URL, na który zostało przekierowane 302 żądanie.

Po dokonaniu przez klienta płatności w serwisie internetowym, na adres e-mail zarejestrowany w systemie PayPal oraz do klienta wysłane jest powiadomienie e-mail. Twoje konto PayPal będzie odzwierciedlać płatność, a informacje o transakcji możesz wyświetlić w historii konta lub jako część dziennika transakcji historii. Po potwierdzeniu płatności przez PayPal możesz wysłać produkty do klienta. Jeśli zdecydujesz się korzystać z systemu PayPal we własnej witrynie internetowej, zapoznaj się ze wszystkimi jego funkcjami. Na przykład możesz nauczyć PayPal automatycznego obliczania kosztów wysyłki i podatku dla każdego zamówienia.

Korzystanie z funkcji zakupów pojedynczych przedmiotów w systemie PayPal

Zakupy pojedynczych przedmiotów to funkcja systemu PayPal, która umożliwia kierowanie użytkownika bezpośrednio do strony płatności zamiast do koszyka PayPal. Koszyk PayPal stanie się bezużyteczny w rozdziale 12, w którym utworzysz własny koszyk. W rozdziale 14 zaimplementujesz przycisk Złóż zamówienie w koszyku, który zapisuje zamówienie w bazie danych i przekazuje je na stronę płatności PayPal. Aby wywołać stronę płatności PayPal (z pominięciem koszyka PayPal), przekierowujesz do następującego łącza:

```
https://www.paypal.com/cgi-
```

```
bin/webscr?cmd=_xclick&business=youremail@example.com&item_name=Order#123&item_number=123&amount=123&currency_code=USD&return=www.example.com&cancel_return=www.example.com
```

Wskazówka: w trzeciej fazie rozwoju stworzysz własny, kompletny system obsługi zamówień, w którym będziesz przetwarzać transakcje kartą kredytową.

Podczas implementacji Zakupów pojedynczych przedmiotów w systemie PayPal w rozdziale 14 będziesz używać kodu, który wygląda jak poniższy fragment kodu, aby utworzyć adres URL strony Zakupy pojedynczych przedmiotów w systemie PayPal:

```
// Calculate the total amount for the shopping cart
$this->mTotalAmount = ShoppingCart::GetTotalAmount();
// If the Place Order button was clicked ...
if(isset($_POST['place_order']))
{
// Create the order and get the order ID
$order_id = ShoppingCart::CreateOrder();
// This will contain the PayPal link
$redirect =
PAYPAL_URL . '&item_name=TShirtShop Order ' . urlencode('#') . $order_id .
'&item_number=' . $order_id .
'&amount=' . $this->mTotalAmount .
'&currency_code=' . PAYPAL_CURRENCY_CODE .
'&return=' . PAYPAL_RETURN_URL .
'&cancel_return=' . PAYPAL_CANCEL_RETURN_URL;
// Redirection to the payment page
header('Location: ' . $redirect);
exit();
}
```

Podsumowanie

Zobaczyłeś, jak zintegrować system PayPal z witryną handlu elektronicznego - proste rozwiązanie płatnicze, które wybiera wiele małych firm, aby nie musiały samodzielnie przetwarzać danych karty kredytowej lub płatności. Najpierw wymieniliśmy niektóre alternatywy dla PayPal, zanim przeprowadzimy Cię przez proces tworzenia nowego konta PayPal. Następnie omówiliśmy, jak zintegrować PayPal w fazie I i II rozwoju, najpierw omawiając koszyk na zakupy, niestandardowy mechanizm realizacji transakcji.