

Korzystanie z usług internetowych Amazon.com

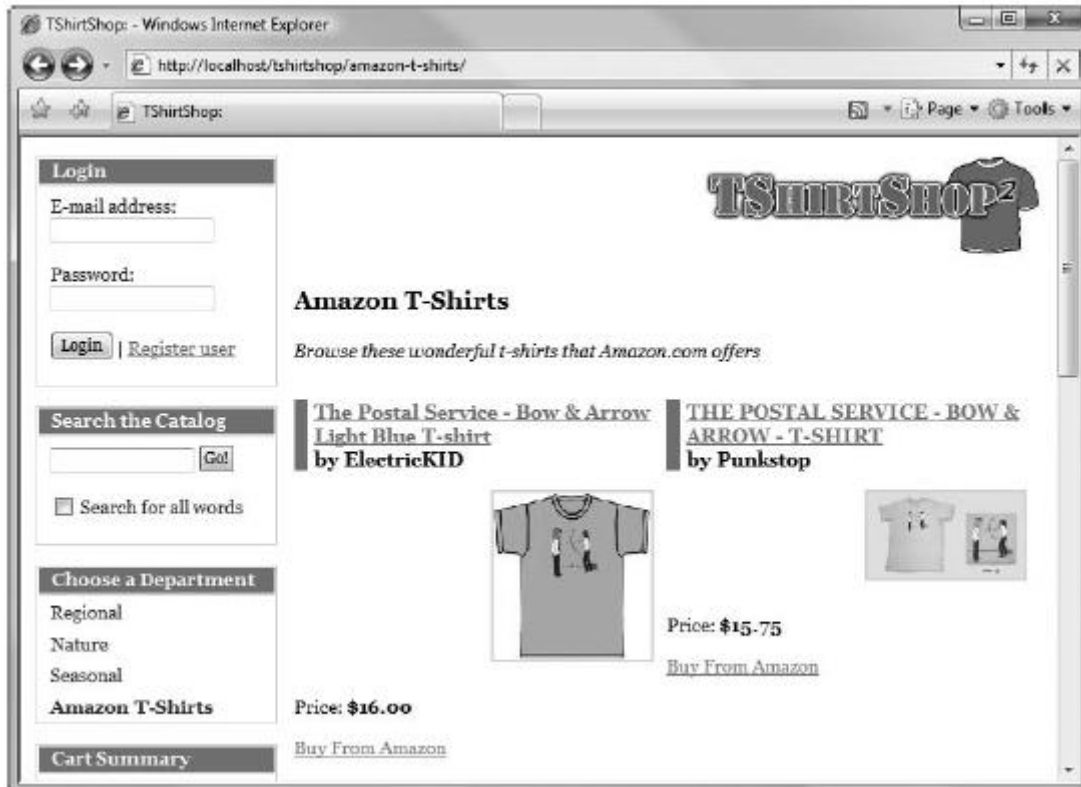
Do tej pory dowiedziałeś się, jak zintegrować zewnętrzne funkcje dostarczane przez PayPal, DataCash i Authorize.net w celu przetwarzania płatności od klientów. Tu poznasz nowe możliwości integracji funkcji ze źródeł zewnętrznych poprzez usługi sieciowe. Umiejętność interakcji z usługami internetowymi innych firm może zapewnić Ci istotną przewagę nad konkurencją. Dokładniej

- Dowiesz się, czym są usługi internetowe
- Dowiesz się, jak połączyć się z usługą Amazon E-Commerce
- Skorzystasz z usługi Amazon E-Commerce Service, aby sprzedawać koszulki Amazon za pośrednictwem TShirtShop

Aby uzyskać więcej informacji na temat dostępu do usług internetowych za pomocą PHP, zalecamy zapoznanie się z Pro PHP XML i usługami sieciowymi, które zawierają przykłady dostępu do usług internetowych Amazon.com, Google, eBay i Yahoo

Przedstawiamy usługi sieciowe

Usługa sieciowa to element funkcji udostępniany za pośrednictwem interfejsu internetowego przy użyciu standardowych protokołów internetowych, takich jak HTTP. Komunikaty wymieniane przez klienta i serwer są kodowane przy użyciu protokołu opartego na XML o nazwie Simple Object Access Protocol (SOAP) lub przy użyciu Representational State Transfer (REST) i wysyłane do serwera za pośrednictwem protokołu HTTP. REST używa starannie spreparowanych adresów URL z określonymi parami nazwa-wartość do wywoływania określonych metod na serwerach. REST jest uważany za najłatwiejszy sposób komunikowania się z usługami sieciowymi, które udostępniają ten interfejs. Korzystając z REST w celu uzyskania dostępu do usługi internetowej, po prostu wysyłasz żądanie HTTP GET, a otrzymasz odpowiedź w formacie XML. SOAP to oparty na XML standard kodowania informacji przesyłanych w żądaniu lub odpowiedzi usługi sieciowej. SOAP jest wspierany przez wiele organizacji, w tym potężne firmy, takie jak Microsoft, IBM i Sun. Piękno korzystania z usług internetowych polega na tym, że klient i serwer mogą korzystać z dowolnej technologii, dowolnego języka i dowolnej platformy. Dopóki wymieniają informacje za pomocą standardowego protokołu, takiego jak SOAP przez HTTP, nie ma problemu, jeśli klient jest telefonem komórkowym, a serwerem jest aplikacja Java działająca na przykład w systemie Solaris. Możliwości są ekscytujące i zalecamy zakup książki specjalizującej się w usługach internetowych, aby dowiedzieć się więcej o ich świecie. Zapoznaj się z listą publicznych usług internetowych pod adresem <http://www.xmlmethods.net/>, aby dowiedzieć się, jakie rodzaje zewnętrznych funkcji można zintegrować ze swoją aplikacją. W tym rozdziale dowiesz się, jak zintegrować Amazon Web Services (AWS) w celu interakcji z Amazon i sprzedawania produktów Amazon.com za pośrednictwem witryny internetowej TShirtShop. Masz już witrynę e-commerce, która sprzedaje koszulki swoim klientom. Możesz pójść dalej i zarobić trochę więcej pieniędzy na ich pasji do koszulek, umieszczając na swojej stronie powiązane prezenty z Amazon.com. Czy robisz to za darmo? O nie, na swojej stronie wyświetlisz dane Amazon.com, ale ostateczna płatność zostanie przetworzona przez Amazon.com, a Amazon.com dostarczy na Twoje konto bankowe niewielką prowizję za zakupy dokonane na Twojej stronie internetowej. Brzmi jak łatwe pieniądze, prawda? W tym rozdziale dowiesz się, jak używać AWS, aby dodać specjalny dział o nazwie Amazon T-Shirts do Twojego sklepu internetowego, który możesz zobaczyć na rysunku.



Będzie to dział „specjalny”, ponieważ będzie obsługiwany inaczej niż inne - na przykład płatność jest obsługiwana bezpośrednio przez Amazon.com, gdy odwiedzający chce kupić produkt. Ten rozdział bada tylko mały podzbiór możliwości AWS, więc jeśli naprawdę chcesz zarobić fortunę na tej usłudze, zajrzyj głębiej w dodatkowe zasoby, aby znaleźć więcej treści.

Pozostała część podzielona jest na dwie części. W pierwszej części dowiesz się, jak uzyskać dostęp do AWS; w drugiej części zintegrujesz AWS ze stroną internetową TShirtShop.

Wskazówka Kod jest niezależny od reszty witryny, więc wszystko, czego potrzebujesz, aby rozpocząć integrację funkcjonalności Amazon.com, to kod z pierwszych czterech rozdziałów (a więc masz działający katalog produktów). Oczywiście, po drobnych poprawkach, możesz również dostosować ten kod do własnych, osobistych rozwiązań.

Dostęp do usług internetowych Amazon

Większość dostawców usług (w tym Amazon.com) używa SOAP lub REST (lub obu), aby udostępnić usługi internetowe programom klienta internetowego, a obie opcje uzyskują takie same wyniki. W tym rozdziale dowiesz się, jak uzyskać dostęp do AWS za pomocą zarówno REST, jak i SOAP. AWS to ogólna nazwa, której Amazon.com używa dla swojej kolekcji usług internetowych. Konkretną usługą, którą jesteśmy zainteresowani, jest Amazon Associates Web Service (A2S, wcześniej znana jako Amazon E-Commerce Service lub ECS). Ta usługa internetowa pozwala nam przeszukiwać katalog Amazon, co pozwala nam programowo uzyskiwać informacje o produktach. Wykorzystamy tę usługę internetową, aby pobrać dane dotyczące koszulek z katalogu Amazon, abyśmy mogli wykorzystać te dane do utworzenia naszego działu Amazon T-Shirts. Podczas uzyskiwania dostępu do dowolnej usługi internetowej Amazon.com, w tym A2S, możesz wysłać żądanie za pomocą REST lub wysyłając wiadomość SOAP. Usługa sieciowa zwróci odpowiedź XML z żądanymi danymi.

Tu dotkniemy tylko części funkcjonalności zapewnianej przez Amazon AWS, a nawet w szczególności przez A2S. Poważna dyskusja na ten temat wymagałaby osobnej książki, ale to, co zobaczysz w tej Części, wystarczy, aby naprowadzić Cię na właściwą ścieżkę. Należy również pamiętać, że integrujemy funkcje Amazon US, ale korzystając z tego samego konta AWS, można uzyskać dostęp do usług Amazon.fr, Amazon.ca, Amazon.de, Amazon.co.jp i Amazon.co. i innych.

Tworzenie konta usług internetowych Amazon.com

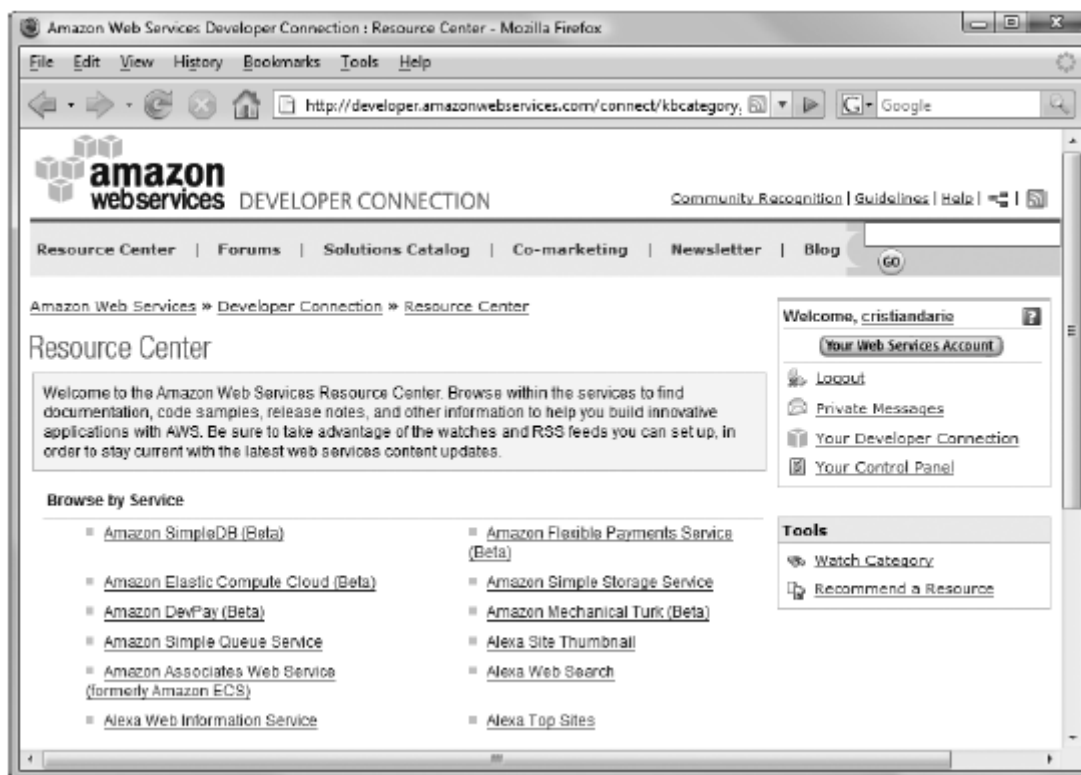
Oficjalna strona internetowa AWS znajduje się pod adresem <http://www.amazon.com/webservices>. Możesz znaleźć najnowszą wersję dokumentacji pod adresem :

<http://developer.amazonwebservices.com/connect/> - pamiętaj, aby dodać ten adres URL do zakładek, ponieważ uznasz go za bardzo przydatny.

Zanim przejdziesz dalej, musisz utworzyć konto w AWS. Aby uzyskać dostęp do AWS, potrzebujesz identyfikatora klucza dostępu, który identyfikuje Twoje konto w systemie AWS. Jeśli jeszcze go nie masz, złóż wniosek teraz na <http://www.amazon.com/gp/aws/registration/registration-form.html>. Identyfikator klucza dostępu to 20-znakowy ciąg alfanumeryczny.

Uwaga Przed 11 października 2005 r. Amazon.com dostarczał coś, co nazywało się identyfikatorem subskrypcji zamiast identyfikatora klucza dostępu. Cel jest podobny i jeśli masz już identyfikator subskrypcji, możesz nadal z niego korzystać. W przypadku nowych aplikacji Amazon.com zachęca do używania identyfikatora klucza dostępu.

Identyfikator klucza dostępu umożliwia dostęp do większej liczby usług internetowych Amazon i usług internetowych Alexa (Alexa jest usługą należącą do Amazon.com), jak widać na rysunku .



Klucz dostępu nie jest informacją publiczną (nie należy go nikomu udostępniać), ale nie jest to również bardzo tajna informacja, ponieważ każdy może go uzyskać za darmo. W przypadku płatnych usług internetowych lub usług, do których należy uzyskać dostęp w bezpieczny sposób, Amazon.com używa

innego klucza, zwanego tajnym kluczem dostępu, który również otrzymujesz po rejestracji. Jednak w tym rozdziale nie będziemy używać tajnego klucza dostępu. Aby uzyskać dostęp do A2S, potrzebujesz tylko klucza dostępu.

Uzyskiwanie identyfikatora partnera Amazon.com

Utworzony wcześniej identyfikator klucza dostępu jest Twoim kluczem do pobierania danych za pośrednictwem Amazon AWS. Te dane pozwalają skomponować dział Amazon T-Shirts, który widziałeś na rysunku 1. Jeśli chcesz zarabiać prowizje za produkty Amazon.com, które sprzedajesz za pośrednictwem swojej witryny, potrzebujesz również identyfikatora stowarzyszonego. Identyfikator stowarzyszony jest używany w linkach Kup od Amazon, które wyświetlasz w swoim specjalnym dziale Amazon.com, i jest to klucz, którego Amazon.com używa do identyfikacji Ciebie jako źródła tej sprzedaży. Więc zanim przejdziesz dalej, jeśli chcesz zarabiać na swoim dziale T-shirtów Amazon, zdobądź swój identyfikator współpracownika ze strony :

<http://associates.amazon.com/gp/associates/apply/main.html>.

Zauważ, że identyfikator stowarzyszony i identyfikator klucza dostępu to dwa niezależne klucze, które masz na Amazon.com, z których każdy ma swój własny cel. Associate ID to identyfikator, który możesz umieścić w linkach Amazon.com na swojej stronie internetowej, aby Amazon.com wiedział, że użytkownicy, którzy klikają te linki, pochodzą od Ciebie. Identyfikator stowarzyszony nie jest tajny, ponieważ każdy, kto przegląda Twoją witrynę, może zobaczyć identyfikator w linkach Amazon.com w Twojej witrynie. Wszystko, czego potrzebujesz, aby sprzedawać produkty Amazon.com za pośrednictwem swojej witryny internetowej, to identyfikator stowarzyszony. Nie potrzebujesz identyfikatora klucza dostępu, który jest wymagany tylko podczas łączenia się z usługami internetowymi Amazon.com. W TShirtShop łączymy się z usługami internetowymi Amazon.com (i używamy identyfikatora klucza dostępu), aby przeszukiwać katalog Amazon.com i uzyskiwać produkty, które umieszczamy w dziale Amazon T-Shirts.

Dostęp do usługi e-commerce Amazon.com za pomocą REST

Dostęp do usług internetowych REST można uzyskać, żądając prawidłowo utworzonego adresu URL. Wypróbuj poniższy link w przeglądarce (nie zapomnij zastąpić ciągu [Twój identyfikator klucza dostępu] prawdziwym identyfikatorem klucza dostępu, który uzyskałeś wcześniej):

<http://webservices.amazon.com/onca/xml?Service=AWSECommerceService>

&AWSAccessKeyId=[Your Access Key ID]

&Operation=ItemLookup

&IdType=ASIN

&ItemId=1590598644

Wskazówka Upewnij się, że wpisujesz cały adres URL w jednym wierszu; podzieliśmy go tutaj na poszczególne elementy, aby ułatwić czytanie. Twoja przeglądarka wyświetli strukturę XML z informacjami o książce, którą właśnie czytasz. Rysunek pokazuje tę strukturę XML w Firefoksie, która ładnie wyświetla drzewo dokumentów XML.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <ItemLookupResponse xmlns="http://webservices.amazon.com/AWSECommerceService/2005-10-05">
- <OperationRequest>
- <HTTPHeaders>
  <Header Name="UserAgent" Value="Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; SLCC1;
  .NET CLR 2.0.50727; .NET CLR 3.0.04506)" />
</HTTPHeaders>
<RequestId>05NZG0XHB9MX6WJ0F3Z1</RequestId>
- <Arguments>
  <Argument Name="ItemId" Value="1590598644" />
  <Argument Name="Service" Value="AWSECommerceService" />
  <Argument Name="Operation" Value="ItemLookup" />
  <Argument Name="IdType" Value="ASIN" />
  <Argument Name="AWSAccessKeyId" Value="129VHA1RCM5KFA12Q282" />
</Arguments>
<RequestProcessingTime>0.0144410133361816</RequestProcessingTime>
</OperationRequest>
- <Items>
- <Request>
  <IsValid>True</IsValid>
  - <ItemLookupRequest>
    <IdType>ASIN</IdType>
    <ItemId>1590598644</ItemId>
  </ItemLookupRequest>
</Request>
- <Item>
  <ASIN>1590598644</ASIN>
  <DetailPageURL>http://www.amazon.com/gp/redirect.html%3FASIN=1590598644%
  26tag=ws%26code=xml%26id=2025%26ccmID=165953%
  26location=/o/ASIN/1590598644%
  253FSubscriptionId=129VHA1RCM5KFA12Q282</DetailPageURL>
  - <ItemAttributes>
    <Author>Emilian Balanescu</Author>
    <Author>Cristian Darie</Author>
    <Manufacturer>Apress</Manufacturer>
    <ProductGroup>Book</ProductGroup>
    <Title>Beginning PHP and MySQL E-Commerce: From Novice to Professional, Second
    Edition (Beginning, from Novice to Professional)</Title>
  </ItemAttributes>
</Item>
</Items>
</ItemLookupResponse>
```

(Omówimy wyświetlanie produktów, wizualnie, w TShirtShop później. Na razie jesteśmy zainteresowani zobaczeniem danych, które są zwracane z żądania.)

Całkiem fajnie, co? Właśnie widziałeś REST w akcji. Każdy produkt w bazie danych Amazon.com ma unikalny identyfikator zwany standardowym numerem pozycji Amazon.com (ASIN). W przypadku książek ASIN jest numerem ISBN książki (ta książka ma numer ASIN 1590598644). Zgłoszone przed chwilą żądanie usługi sieciowej zawiera dla AWS następujące informacje: Mam identyfikator klucza dostępu (AWSAccessKeyId=[Identyfikator Twojego klucza dostępu]) i chcę wykonać operację wyszukiwania elementu (&Operation=ItemLookup), aby dowiedzieć się więcej o produkcie za pomocą 1590598644 ASIN (&IdType=ASIN&ItemId=1590598644). W tym przykładzie nie uzyskałeś zbyt wielu informacji o tej książce — żadnych informacji o cenie lub dostępności, ani linków do zdjęcia na okładkę lub recenzji klientów. Możesz dostosować dane, które chcesz otrzymywać, korzystając z grup odpowiedzi (grupa odpowiedzi to zestaw informacji o produkcie).

Poprośmy więc o więcej danych, korzystając z grup odpowiedzi. Na końcu utworzonego wcześniej łącza dodaj następujący ciąg, aby uzyskać bardziej szczegółowe informacje o książce: &ResponseGroup=Request,SalesRank,Small,Images,VariationSummary. Kompletny link powinien wyglądać tak:

<http://webservices.amazon.com/onca/xml?Service=AWSECommerceService>

&AWSAccessKeyId=[Your Access Key ID]

&Operation=ItemLookup

&IdType=ASIN

&ItemId=1590598644

&ResponseGroup=Request,SalesRank,Small,Images,VariationSummary

Nowa odpowiedź XML z Amazon.com zawiera więcej szczegółów na temat elementu Amazon.com, jak pokazano na rysunku.



```
<?xml version="1.0" encoding="UTF-8" ?>
- <ItemLookupResponse xmlns="http://webservices.amazon.com/AWSECommerceService/2005-10-05">
+ <OperationRequest>
- <Items>
- <Request>
- <IsValid>True</IsValid>
- <ItemLookupRequest>
  <IdType>ASIN</IdType>
  <ItemId>1590598644</ItemId>
  <ResponseGroup>SalesRank</ResponseGroup>
  <ResponseGroup>VariationSummary</ResponseGroup>
  <ResponseGroup>Small</ResponseGroup>
  <ResponseGroup>Request</ResponseGroup>
  <ResponseGroup>Images</ResponseGroup>
</ItemLookupRequest>
</Request>
- <Item>
  <ASIN>1590598644</ASIN>
  <DetailPageURL>http://www.amazon.com/gp/redirect.html%3FASIN=1590598644%
  26tag=ws%26lcode=xm2%26cID=2025%26ccmID=165953%
  26location=/o/ASIN/1590598644%
  253FSubscriptionId=129VHA1RCM5KFA12Q282</DetailPageURL>
  <SalesRank>1274582</SalesRank>
  - <SmallImage>
    <URL>http://ecx.images-amazon.com/images/I/11HLNxDHmSL.jpg</URL>
    <Height Units="pixels">75</Height>
    <Width Units="pixels">57</Width>
  </SmallImage>
```

Właśnie zmieszaliśmy pięć grup odpowiedzi: Request, SalesRank, Small, Images i OfferSummary. Aby dowiedzieć się więcej o grupach odpowiedzi, przejdź na stronę :

<http://developer.amazonwebservices.com/connect/kbcategory.jspx?categoryID=5>.

i kliknij Najnowsza technologia. przycisk Dokumenty. Możesz też kliknąć łącze Dokumentacja techniczna, a następnie łącze do najnowszej wersji dokumentacji.

W dokumentacji AWS szczegóły grup odpowiedzi znajdziesz w sekcji „API Reference, Response Groups”. Oto opis pięciu grup odpowiedzi użytych w poprzednim przykładzie:

- Grupa odpowiedzi na żądanie jest domyślną grupą odpowiedzi w każdym rodzaju operacji i zwraca listę par nazwa-wartość użytych do utworzenia żądania.
- Grupa odpowiedzi Sales Rank zwraca dane o aktualnej pozycji sprzedaży produktu w Amazon.com.

- Grupa Mała odpowiedź zwraca ogólne dane elementu (ASIN, nazwę elementu, adres URL itd.) dotyczące elementów zawartych w odpowiedzi. Jest to domyślna grupa odpowiedzi dla operacji ItemLookup (tak jak w tym przykładzie).
- Grupa odpowiedzi Obrazy zawiera adresy trzech obrazów (małego, średniego i dużego) dla każdego elementu odpowiedzi.
- Grupa odpowiedzi VariationSummary zawiera szczegółowe informacje o cenie (najniższa cena, najwyższa cena, najniższa cena sprzedaży i najwyższa cena sprzedaży) dla każdego elementu w odpowiedzi.

Kontynuujmy, ucząc się, jak wykonać żądanie REST z PHP. Aby zapełnić przyszły dział Amazon T-Shirts, przeszukasz dział odzieżowy Amazon.com pod kątem słów kluczowych związanych z koszulkami pocztowymi. Jednym z trywialnych sposobów jest użycie funkcji PHP `file_get_contents()`, tak jak można zobaczyć w poniższym skrypcie. Aby przetestować dostęp do usług internetowych za pomocą REST, utwórz nowy plik o nazwie `test_rest.php` w swoim katalogu `tshirtshop` i napisz w nim następujący kod:

```
<?php
// Tell the browser it is going to receive an XML document.
header('Content-type: text/xml');

/* DON'T FORGET to replace the string '[Your Access Key ID]' with your
Access Key ID in the following line */
$url = 'http://webservices.amazon.com/onca/xml?Service=AWSECommerceService' .
'&AWSAccessKeyId=[Your Access Key ID]' .
'&Operation=ItemSearch' .
'&Keywords=postal+t-shirt' .
'&SearchIndex=Apparel' .
'&ResponseGroup=Request,Medium,VariationSummary';
echo file_get_contents($url);
?>
```

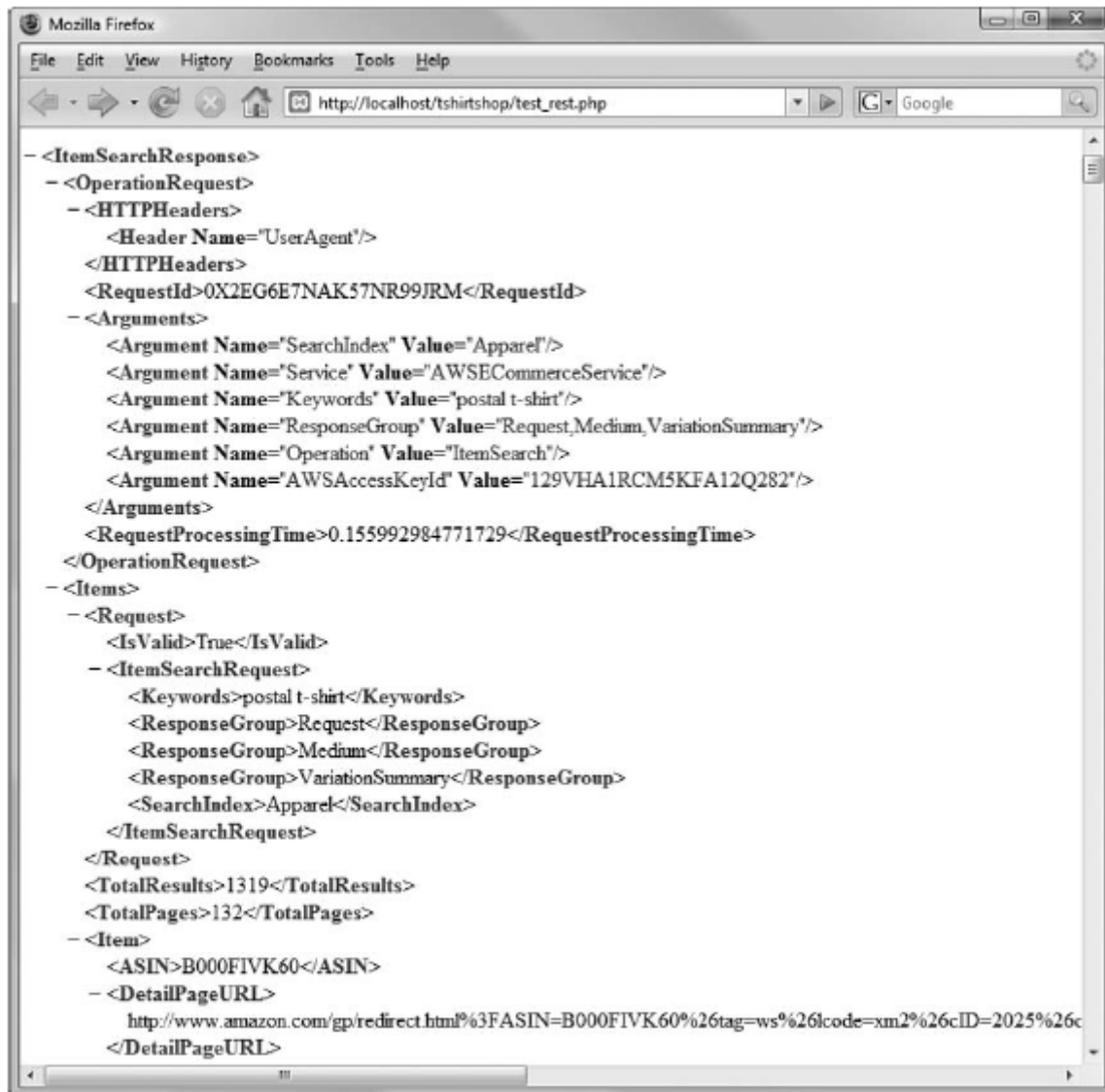
Uwaga Niektóre instalacje PHP i dostawcy usług hostingowych mogą nie zezwalać na domyślne uruchamianie tego kodu. W takim przypadku możesz zmienić to ustawienie w `php.ini`:

```
allow_url_fopen = On.
```

Alternatywnie możesz dodać następującą linię do `include/config.php`. To drugie rozwiązanie jest preferowane, ponieważ dotyczy tylko Twojej aplikacji i pozostaje ustawione, jeśli musisz przenieść aplikację na inny serwer.

```
ini_set('allow_url_fopen', 'On');
```

Wczytanie `http://localhost/tshirtshop/test_rest.php` pokaże dane XML dotyczące koszulek Amazon.com



Aby ćwiczyć i budować więcej linków XML, po prostu przestuduj przykłady w sekcji „API Reference” dokumentacji AWS. Materiał pokaże Ci, jak wykonywać różne operacje Amazon.com za pomocą REST. Dostęp do usługi e-commerce Amazon.com za pomocą SOAP

Używając SOAP, korzystasz ze złożonego API, aby uzyskać dostęp do potrzebnej funkcjonalności Amazon.com. Poniższy kod, który wykonuje tę samą operację wyszukiwania koszulek, co wcześniej z REST, używa obiektów AWSECommerceService, ItemSearch i ItemSearchRequest z interfejsu API Amazon.com do wykonania operacji.

Uwaga Aby uzyskać dostęp do serwera Amazon.com za pomocą SOAP, używamy rozszerzenia PHP SOAP. Upewnij się, że odwołanie php_soap.dll w php.ini nie jest komentowane. Dokumentację funkcji PHP SOAP można znaleźć pod adresem <http://www.php.net/soap/>.

Aby przetestować dostęp do usług internetowych za pomocą SOAP, utwórz nowy plik o nazwie test_soap.php w swoim katalogu tshirtshop i napisz w nim następujący kod:

```
<?php
```

```
try
```



```

{
// Initialize SOAP client object
$client = new SoapClient(
'http://webservices.amazon.com/AWSECommerceService/AWSECommerceService.wsdl');
/* DON'T FORGET to replace the string '[Your Access Key ID]' with your
subscription ID in the following line */
$request = array ('Service' => 'AWSECommerceService',
'AWSAccessKeyId' => '[Your Access Key ID]',
'Request' => array ('Operation' => 'ItemSearchRequest',
'Keywords' => 'postal+t-shirt',
'SearchIndex' => 'Apparel',
'ResponseGroup' =>
array ('Request',
'Medium',
'VariationSummary')));
$result = $client->ItemSearch($request);
echo '<pre>';
print_r($result);
echo '</pre>';
}
catch (SoapFault $fault)
{
trigger_error('SOAP Fault: (faultcode: ' . $fault->faultcode . ', ' .
'faultstring: ' . $fault->faultstring . '), E_USER_ERROR);
}
?>

```

Cały kod żądania SOAP jest zawarty w bloku try. Jeśli żądanie SOAP nie powiedzie się, zgłasza wyjątek typu SoapFault, który przekształcamy w błąd za pomocą funkcji trigger_error(). Przeczytaj więcej na temat wyjątku SOAP na <http://www.php.net/manual/en/function.is-soapfault.php>.

Wynikiem żądania SOAP jest obiekt zawierający żądane dane. Jeśli załadujesz test_soap.php w swojej przeglądarce (nie zapomnij umieścić w niej swojego identyfikatora klucza dostępu), powinna ona wyświetlać dane w formacie tekstowym, który nie jest łatwy do odczytania dla ludzkiego oka. Kod zaczyna się od utworzenia obiektu klienta SOAP w serwisie internetowym SOAP Amazon.com:

```
// Initialize SOAP client object
```

```
$client = new SoapClient(
```

```
'http://webservices.amazon.com/AWSECommerceService/AWSECommerceService.wsdl');
```

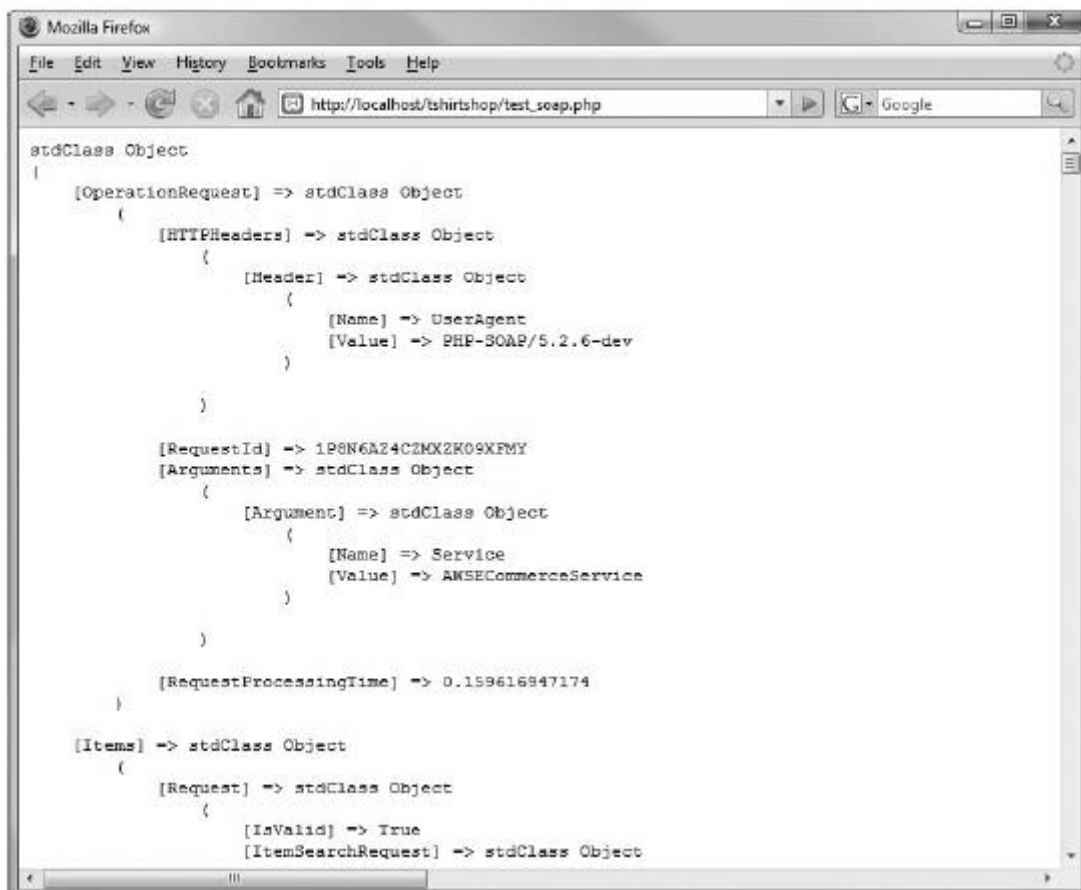
Wspomniany plik Web Services Definition Language (WSDL) opisuje wszystkie funkcje i typy ich parametrów, które rozumie serwer SOAP Amazon.com. Wcześniej utworzony obiekt klienta SOAP Amazon.com wie o wszystkich tych funkcjach i możesz je teraz wywołać, używając czegoś takiego:

```
$result = $client->ItemSearch($request);
```

Alternatywnie możesz wykonać to samo wywołanie i niejawnie uzyskać te same wyniki, używając funkcji `__soapCall()` (<http://www.php.net/manual/en/function.soap-soapclient-soapcall.php>), tak:

```
$result->__soapCall('ItemSearch', array ($request));
```

Żądanie usługi sieci Web wykonuje operację `ItemSearch` na słowa kluczowe `poczta+koszulka` w sklepie „Odzież”. Całe żądanie jest umieszczane w bloku `try-catch`, który przechwytuje wszelkie potencjalne wyjątki i generuje błąd. Przeczytaj więcej o klasie wyjątków `SoapFault`, która zawiera szczegóły błędu SOAP, pod adresem <http://www.php.net/manual/en/function.is-soap-fault.php>. Załadowanie `test_soap.php` generuje wynik pokazany na rysunku.



```
stdClass Object
(
    [OperationRequest] => stdClass Object
        (
            [HTTPHeaders] => stdClass Object
                (
                    [Header] => stdClass Object
                        (
                            [Name] => UserAgent
                            [Value] => PHP-SOAP/5.2.6-dev
                        )
                )
            [RequestId] => 1P8N6A24CZMX2K09XPMY
            [Arguments] => stdClass Object
                (
                    [Argument] => stdClass Object
                        (
                            [Name] => Service
                            [Value] => AWSECommerceService
                        )
                )
            [RequestProcessingTime] => 0.159616947174
        )
    [Items] => stdClass Object
        (
            [Request] => stdClass Object
                (
                    [Invalid] => True
                    [ItemSearchRequest] => stdClass Object
                )
        )
)
```

Integracja A2S z TShirtShop

Celem jest przyniesienie do Twojego sklepu kilku książek związanych ze słowami „postal t-shirt” z Amazon.com. Zbudujesz specjalny dział bez kategorii, który będzie wyświetlał niektóre informacje o

książce (zdjęcie na okładce, tytuł, autorzy i cena). Każda książka będzie miała link Kup od Amazon, który pozwala odwiedzającemu kupić książkę na Amazon.com. Jeśli złożysz wniosek o konto ID współpracownika Amazon.com, otrzymasz z tego niewielką prowizję. Po wykonaniu ćwiczeń zaimplementujesz integrację Amazon.com, jak pokazano wcześniej na rysunku 1. Poniższy link uruchamia wyszukiwanie REST dla książek Amazon.com na podstawie słów kluczowych związanych z koszulkami pocztowymi i zwraca pierwszych dziesięć danych produktów posortowanych według ich pozycji sprzedaży:

```
http://webservices.amazon.com/onca/xml?Service=AWSECommerceService
```

```
&Operation=ItemSearch
```

```
&AWSAccessKeyId=[Your Access Key ID]
```

```
&Keywords=postal+t-shirt
```

```
&SearchIndex=Apparel
```

```
&ResponseGroup=Request%2CMedium%2CVariationSummary&Sort=salesrank
```

Spośród wszystkich produktów, które zostaną zwrócone na naszą prośbę, wyświetlimy tylko te, które są dostępne do zakupu i ze zdjęciami na okładce.

Wdrażanie poziomu biznesowego

W warstwie biznesowej dodasz kod, który uzyskuje dostęp do systemu A2S.

Ćwiczenie: Dodawanie kodu komunikacyjnego A2S do warstwy biznesowej

1. Dodaj następujący kod do pliku include/config.php:

```
// Amazon E-Commerce Service
// define('AMAZON_METHOD', 'REST');
define('AMAZON_METHOD', 'SOAP');
define('AMAZON_WSDL',
'http://webservices.amazon.com/AWSECommerceService/AWSECommerceService.wsdl');
define('AMAZON_REST_BASE_URL',
'http://webservices.amazon.com/onca/xml?Service=AWSECommerceService');
// Set Amazon Access Key ID
define('AMAZON_ACCESS_KEY_ID', '[Your Access Key ID]');
// Set Amazon Associates ID
define('AMAZON_ASSOCIATES_ID', '[Your Amazon associate ID]');
// Set Amazon request options
define('AMAZON_SEARCH_KEYWORDS', 'postal t-shirt');
define('AMAZON_SEARCH_NODE', 'Apparel');
define('AMAZON_RESPONSE_GROUPS', 'Request,Medium,VariationSummary');
```

2. Utwórz nowy plik o nazwie amazon.php w folderze biznesowym i dodaj do niego następujący kod. Pojedyncza metoda publiczna, która zostanie wywołana z wyższych warstw, to GetProducts(), podczas gdy pozostałe są prywatnymi metodami do użytku wewnętrznego, które obsługują funkcjonalność GetProducts().

```
<?php

// Class for accessing A2S

class Amazon
{
    public function Amazon()
    {
    }

    // Retrieves Amazon products for sending to presentation tier

    public function GetProducts()
    {
        // Use SOAP to get data
        if (AMAZON_METHOD == 'SOAP')
            $result = $this->_GetDataWithSoap();

        // Use REST to get data
        else
            $result = $this->_GetDataWithRest();

        // Initializes Array object
        $results = array ();

        // Format results
        $results = $this->_DataFormat($result);

        // Returns results
        return $results;
    }

    // Call A2S using REST

    private function _GetDataWithRest()
    {
        $params = array ('Operation' => 'ItemSearch',
            'SubscriptionId' => AMAZON_ACCESS_KEY_ID,
```

```

'Keywords' => AMAZON_SEARCH_KEYWORDS,
'SearchIndex' => AMAZON_SEARCH_NODE,
'ResponseGroup' => AMAZON_RESPONSE_GROUPS,
'Sort' => 'salesrank');
$query_string = '&';
foreach ($params as $key => $value)
$query_string .= $key . '=' . urlencode($value) . '&';
$amazon_url = AMAZON_REST_BASE_URL . $query_string;
// Get the XML response using REST
$amazon_xml = file_get_contents($amazon_url);
// Unserialize the XML and return
return simplexml_load_string($amazon_xml);
}
// Call A2S using SOAP
private function _GetDataWithSoap()
{
try
{
$client = new SoapClient(AMAZON_WSDL);
/* Set up an array containing input parameters to be
passed to the remote procedure */
$request = array ('SubscriptionId' => AMAZON_ACCESS_KEY_ID,
'Request' => array ('Operation' =>
'ItemSearchRequest',
'Keywords' =>
AMAZON_SEARCH_KEYWORDS,
'SearchIndex' =>
AMAZON_SEARCH_NODE,
'ResponseGroup' =>
AMAZON_RESPONSE_GROUPS,
'Sort' => 'salesrank'));

```

```

// Invoke the method
$result = $client->ItemSearch($request);
return $result;
}
catch (SoapFault $fault)
{
trigger_error('SOAP Fault: (faultcode: ' . $fault->faultcode . ', ' .
'faultstring: ' . $fault->faultstring . ')',
E_USER_ERROR);
}
}

/* Places an "image not available" picture for products with no image,
and saves the results in an array with a simple structure for easier
handling at the upper levels */
private function _DataFormat($result)
{
/* Variable k is the index of the $new_result array, which will
contain the Amazon products to be displayed in TShirtShop */
$k = 0;
$new_result = array ();
/* Analyze all products retrieved from A2S
and save them into the $new_result array */
for ($i = 0; $i < count($result->Items->Item); $i++)
{
// Make a temporary copy for product data
$temp = $result->Items->Item[$i];
/* Set product's image to images/not_available.jpg,
if image url is empty */
if (property_exists($temp, 'MediumImage') &&
((string) $temp->MediumImage->URL) != "")
$new_result[$k]['image'] = (string) $temp->MediumImage->URL;

```

```

else
$new_result[$k]['image'] = 'images/not_available.jpg';
// Save asin, brand, name, and price into the $new_result array
$new_result[$k]['asin'] = (string) $temp->ASIN;
$new_result[$k]['brand'] = (string) $temp->ItemAttributes->Brand;
$new_result[$k]['item_name'] = (string) $temp->ItemAttributes->Title;
if (property_exists($temp, 'VariationSummary') &&
property_exists($temp->VariationSummary, 'LowestPrice'))
{
$new_result[$k]['price'] =
(string) $temp->VariationSummary->LowestPrice->FormattedPrice;
$highest_price = $new_result[$k]['price'];
if (property_exists($temp->VariationSummary, 'HighestPrice'))
$highest_price =
(string) $temp->VariationSummary->HighestPrice->FormattedPrice;
if ($highest_price !== $new_result[$k]['price'])
$new_result[$k]['price'] .= ' - ' . $highest_price;
}
else
$new_result[$k]['price'] = "";
$k++;
}
return $new_result;
}
}
?>

```

Jak to działa: komunikacja z AWS

Jedyną publiczną metodą warstwy biznesowej Amazon.com jest GetProducts(), która dba o pobieranie danych. Jego funkcjonalność jest dość jasna, ponieważ używa wielu metod pomocniczych do wykonania pracy. Najpierw decyduje, czy powinien używać SOAP czy REST w zależności od ustawienia konfiguracji dodanego do pliku include/config.php:

```
define('AMAZON_METHOD', 'SOAP');
```

Stała AMAZON_METHOD zdefiniowana w pliku include/config.php informuje o tym, czy z AWS będzie się kontaktować przez REST czy SOAP. Wartość tej stałej (która powinna być REST lub SOAP) decyduje o tym, czy _GetDataWithRest() lub _GetDataWithSoap() będą używane do kontaktu z Amazon. Niezależnie od wybranej metody wyniki powinny być takie same:

```
// Retrieves Amazon products for sending to presentation tier
```

```
public function GetProducts()
```

```
{
```

```
// Use SOAP to get data
```

```
if (AMAZON_METHOD == 'SOAP')
```

```
$result = $this->_GetDataWithSoap();
```

```
// Use REST to get data
```

```
else
```

```
$result = $this->_GetDataWithRest();
```

_GetDataWithSoap() i _GetDataWithRest() zwracają listę produktów jako obiekt. Następnie używamy metody _DataFormat(), aby przeanalizować dane z tego obiektu i zwrócić je w postaci tablicy asocjacyjnej. Metoda _DataFormat() umieszcza również obraz „obraz niedostępny” dla produktów Amazon.com, które nie mają obrazu produktu.

```
// Initializes Array object
```

```
$results = array ();
```

```
// Format results
```

```
$results = $this->_DataFormat($result);
```

```
// Returns results
```

```
return $results;
```

```
}
```

Przyjrzyjmy się teraz _GetDataWithRest() i _GetDataWithSoap(), czyli metodom, które realizują komunikację z AWS. GetDataWithRest() pobiera dane usługi sieciowej za pomocą REST. Rozpoczyna się od skonstruowania wymaganego ciągu zapytania, łącząc poszczególne parametry, które chcesz wysłać do Amazon.com:

```
// Call AWS using REST
```

```
private function _GetDataWithRest()
```

```
{
```

```
$params = array ('Operation' => 'ItemSearch',
```

```
'SubscriptionId' => AMAZON_ACCESS_KEY_ID,
```

```
'Keywords' => AMAZON_SEARCH_KEYWORDS,
```



```
'SearchIndex' => AMAZON_SEARCH_NODE,
'ResponseGroup' => AMAZON_RESPONSE_GROUPS,
'Sort' => 'salesrank');
$query_string = '&';
foreach ($params as $key => $value)
$query_string .= $key . '=' . urlencode($value) . '&';
```

Pełny adres URL Amazon.com, który należy wywołać, składa się z podstawowego adresu URL (zapisanego jako stała w pliku include/config.php), do którego dołącza się właśnie utworzony ciąg zapytania:

```
$amazon_url = AMAZON_REST_BASE_URL . $query_string;
```

Używając funkcji `file_get_contents()`, wykonujesz proste żądanie HTTP GET do Amazon. To tak jak wpisywanie adresu w przeglądarce:

```
// Uzyskaj odpowiedź XML za pomocą REST
```

```
$amazon_xml = file_get_contents($amazon_url);
```

Zmienna `$amazon_xml` będzie zawierać łańcuch ze zwróconymi danymi XML. Aby go dalej przetworzyć, używamy funkcji `simplexml_load_string()`, która analizuje tekst XML i zwraca obiekt `SimpleXMLElement` reprezentujący dokument XML.

```
// Unserialize the XML and return
```

```
return simplexml_load_string($amazon_xml);
```

```
}
```

Metoda `_GetDataWithSoap()` ma podobną funkcjonalność jak `_GetDataWithRest()`, ale wykonuje operację `ItemSearch` przy użyciu protokołu SOAP. Logika, której używa ta metoda do kontaktu z AWS, jest taka sama jak na stronie, którą napisałeś wcześniej w tym rozdziale.

Wdrażanie poziomej prezentacji

Stwórzmy szablon skomponentowany, który będzie wyświetlał koszulki, a następnie zmodyfikujmy szablon skomponentowany `lista_działów`, aby uwzględnić ten nowy dział.

Ćwiczenie: Wyświetlanie produktów Amazon.com w TShirtShop

1. Dodaj nowy plik o nazwie `amazon_products_list.tpl` w folderze prezentacji/templates swojego projektu i dodaj w nim następujący kod:

```
{* amazon_products_list.tpl *}
{load_presentation_object filename="amazon_products_list" assign="obj"}
<h1>{$obj->mDepartmentName}</h1>
<p class="description">{$obj->mDepartmentDescription}</p>
<table class="product-list">
```

```

<tbody>
{section name=k loop=$obj->mProducts}
{if $smarty.section.k.index % 2 == 0}
<tr>
{/if}
<td valign="top">
<h3 class="product-title">
<a href="{ $obj->mProducts[k].link_to_product}">
{ $obj->mProducts[k].item_name}
</a>
<br />
by { $obj->mProducts[k].brand}
</h3>
<p>
{if $obj->mProducts[k].image neq ""}
<a href="{ $obj->mProducts[k].link_to_product}">
mProducts[k].item_name}" width="120" />
</a>
{/if}
</p>
<p class="attributes">
{if $obj->mProducts[k].price neq ""}
Price: <font class="price">{ $obj->mProducts[k].price}</font>
{/if}
</p>
<p class="section">
<a target="_blank" href="{ $obj->mProducts[k].link_to_product}">
Buy From Amazon
</a>
</p>

```

```

</td>
{if $smarty.section.k.index % 2 != 0 && !$smarty.section.k.first ||
$smarty.section.k.last}
</tr>
{/if}
{/section}
</tbody>
</table>

```

2. Utwórz nowy plik o nazwie amazon_products_list.php w folderze prezentacji i dodaj w nim następujący kod:

```

<?php
// Class that handles receiving AWS data
class AmazonProductsList
{
// Public variables available in smarty template
public $mProducts;
public $mDepartmentName;
public $mDepartmentDescription;
// Constructor
public function __construct()
{
$this->mDepartmentName = AMAZON_DEPARTMENT_TITLE;
$this->mDepartmentDescription = AMAZON_DEPARTMENT_DESCRIPTION;
}
public function init()
{
$amazon = new Amazon();
$this->mProducts = $amazon->GetProducts();
for ($i = 0;$i < count($this->mProducts); $i++)
$this->mProducts[$i]['link_to_product'] =
'http://www.amazon.com/exec/obidos/ASIN/' .

```

```
$this->mProducts[$i]['asin'] . '/ref=nosim/' . AMAZON_ASSOCIATES_ID;
}
}
?>
```

3. Dodaj następujące dwie linie konfiguracyjne na końcu pliku include/config.php:

```
// Amazon.com department configuration options
define('AMAZON_DEPARTMENT_TITLE', 'Amazon T-Shirts');
define('AMAZON_DEPARTMENT_DESCRIPTION',
'Browse these wonderful t-shirts that Amazon.com offers');
```

4. Zmodyfikuj plik szablonu Presentation/templates/departments_list.tpl, aby dodać dział Amazon T-Shirts. Dodaj wyróżniony kod pokazany tutaj:

```
<li>
{* Generate a link for a new department in the list *}
<a {$selected} href="{$_obj->mDepartments[i].link_to_department}">
{$_obj->mDepartments[i].name}
</a>
</li>
{/section}
{assign var=selected value=""}
{if $_obj->mAmazonSelected}
{assign var=selected value="class=\"selected\""}
{/if}
<li>
<a {$selected} href="{$_obj->mLinkToAmazonDepartment}">
{$_obj->mAmazonDepartmentName}
</a>
</li>
</ul>
</div>
{* End departments list *}
```

5. Zaktualizuj prezentację/departments_list.php, jak zaznaczono w tym fragmencie kodu:

```

<?php
// Manages the departments list
class DepartmentsList
{
/* Public variables available in departments_list.tpl Smarty template */
public $mSelectedDepartment = 0;
public $mDepartments;
public $mAmazonSelected = false;
public $mAmazonDepartmentName;
public $mLinkToAmazonDepartment;
// Constructor reads query string parameter
public function __construct()
{
/* If DepartmentId exists in the query string, we're visiting a
department */
if (isset($_GET['DepartmentId']))
$this->mSelectedDepartment = (int)$_GET['DepartmentId'];
elseif (isset($_GET['ProductId']) &&
isset($_SESSION['link_to_continue_shopping']))
{
$continue_shopping =
Link::QueryStringToArray($_SESSION['link_to_continue_shopping']);
if (array_key_exists('DepartmentId', $continue_shopping))
$this->mSelectedDepartment =
(int)$continue_shopping['DepartmentId'];
}
// Set Amazon department name and build the link for department
$this->mAmazonDepartmentName = AMAZON_DEPARTMENT_TITLE;
$this->mLinkToAmazonDepartment = Link::ToAmazonDepartment();
// Check whether the Amazon department is selected
if ((isset($_GET['DepartmentId'])) &&

```

```
((string) $_GET['DepartmentId'] == 'Amazon'))
```

```
$this->mAmazonSelected = true;
```

```
}
```

...

6. Dodaj następującą metodę do klasy Link w pliku Presentation/link.php:

```
// Creates a link to the Amazon T-Shirts department
```

```
public static function ToAmazonDepartment()
```

```
{
```

```
return self::Build('amazon-t-shirts/');
```

```
}
```

7. W tej samej klasie zmodyfikuj podświetloną metodę CheckRequest(), aby wyjść z funkcji, jeśli zażądamy strony działu Amazon T-Shirts:

...

```
// Obtain proper URL for department pages
```

```
elseif (isset ($_GET['DepartmentId']))
```

```
{
```

```
if ((string) $_GET['DepartmentId'] == 'Amazon') return;
```

```
if (isset ($_GET['Page']))
```

```
$proper_url = self::ToDepartment($_GET['DepartmentId'],
```

```
$_GET['Page']);
```

```
else
```

```
$proper_url = self::ToDepartment($_GET['DepartmentId']);
```

```
}
```

...

8. Otwórz plik .htaccess z folderu głównego projektu i dodaj następującą regułę przepisywania definicji dla strony działu Amazon T-Shirts:

...

```
# Rewrite order error pages
```

```
RewriteRule ^order-error/?$ index.php?OrderError [L]
```

```
# Rewrite Amazon t-shirts department pages
```

```
RewriteRule ^amazon-t-shirts/?$ index.php?DepartmentId=Amazon [L]
```

```
</IfModule>
```

```
# Set the default 500 page for Apache errors
```

```
ErrorDocument 500 /tshirtshop/500.php
```

```
# Set the default 404 page
```

```
ErrorDocument 404 /tshirtshop/404.php
```

9. Zaktualizuj plik include/index.php, aby odwoływał się do nowej klasy warstwy biznesowej, dodając następujący kod w górnej części wiersza sprawdzającego żądania stron:

```
require_once BUSINESS_DIR . 'amazon.php';
```

```
// URL correction
```

```
Link::CheckRequest();
```

```
// Load Smarty template file
```

```
$application = new Application();
```

10. Otwórz plik Presentation/store_front.php i zmodyfikuj metodę init() z klasy StoreFront, aby na żądanie załadować dział Amazon T-Shirts:

```
...
```

```
// Load department details if visiting a department
```

```
if (isset($_GET['DepartmentId']))
```

```
{
```

```
if ((string) $_GET['DepartmentId'] == 'Amazon')
```

```
$this->mContentsCell = 'amazon_products_list.tpl';
```

```
else
```

```
{
```

```
$this->mContentsCell = 'department.tpl';
```

```
$this->mCategoriesCell = 'categories_list.tpl';
```

```
}
```

```
}
```

```
...
```

11. Załaduj TShirtShop w przeglądarce, a następnie kliknij nowo utworzony dział Amazon T-Shirts. Powinieneś zobaczyć nowy dział

Jak to działa: wyświetlanie produktów Amazon.com w TShirtShop

W tym ćwiczeniu po prostu zaktualizowałeś TShirtShop, aby wyświetlał produkty Amazon.com, stosując techniki, które poznałeś w pierwszej części rozdziału. Nowa funkcjonalność nie jest szczególnie złożona, ale możliwości są ekscytujące. Aby zmienić metodę dostępu, zmodyfikuj następujące elementy w pliku include/config.php:

```
// Usługa e-commerce Amazon
```

```
define('AMAZON_METHOD', 'REST');
```

```
//define('AMAZON_METHOD', 'MYDŁO');
```

Po kliknięciu linku Kup od Amazon, Amazon.com kojarzy tego klienta i zakupione przedmioty z Twoim identyfikatorem partnera (który jest wymieniony w linkach). W metodzie `init()` z klasy `AmazonProductsList` wywoływana jest metoda `GetProducts()` z klasy `Amazon` w celu pobrania danych do wypełnienia listy produktów. Te dane są odczytywane w celu zbudowania linków Amazon.com do pobranych produktów:

```
public function init()
{
    $amazon = new Amazon();

    $this->mProducts = $amazon->GetProducts();

    for ($i = 0;$i < count($this->mProducts); $i++)
        $this->mProducts[$i]['link'] =
            'http://www.amazon.com/exec/obidos/ASIN/' .
            $this->mProducts[$i]['asin'] .
            '/ref=nosim/' . AMAZON_ASSOCIATES_ID;
}
```

Musisz jednak wiedzieć, że Amazon.com oferuje wiele sposobów, dzięki którym możesz pozwolić swoim odwiedzającym na kupowanie jego produktów. Jeśli zalogujesz się na stronie Associates, zobaczysz wiele typów linków, które możesz zbudować i zintegrować ze swoją witryną internetową. Być może najbardziej interesująca i potężna jest możliwość tworzenia i zarządzania koszykami zakupów Amazon.com z Twojego kodu PHP za pomocą API Amazon.com. Jeśli naprawdę chcesz zintegrować Amazon.com ze swoją stroną internetową, powinieneś dokładnie przestudiować dokumentację AWS i jak najlepiej ją wykorzystać.

Podsumowanie

Dowiedziałeś się, jak uzyskać dostęp do usługi handlu elektronicznego Amazon.com za pomocą REST i SOAP. Będziesz mógł korzystać z tych samych technik podczas uzyskiwania dostępu do dowolnej zewnętrznej funkcjonalności udostępnianej za pośrednictwem tych protokołów. Gratulacje, właśnie zakończyłeś swoją podróż do nauki tworzenia witryn e-commerce w PHP i MySQL. Masz wiedzę, aby tworzyć własne, spersonalizowane rozwiązania, być może nawet bardziej interesujące i potężne niż to, co Ci pokazaliśmy.