

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK

DODAJ DO KOSZYKA

CENNIK I INFORMACJE

ZAMÓW INFORMACJE
O NOWOŚCIACH

ZAMÓW CENNIK

CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

Ajax. Biblia

Autor: Steve Holzner

Tłumaczenie: Anna Trojan

ISBN: 978-83-246-1199-7

Tytuł oryginału: [Ajax Bible](#)

Format: B5, stron: około 670

oprawa twarda



Skuteczność i bezpieczeństwo Twoich aplikacji

- Jak korzystać z DOM?
- Jak tworzyć pływające menu oraz efekty za pomocą CSS?
- Jak szyfrować dane w HTTP przy użyciu JavaScript?

Dynamiczne, efektowne i sprawne strony internetowe to dzisiaj podstawa wizerunku każdego przedsiębiorstwa. Stworzenie takich właśnie witryn umożliwia Ajax – zbiór technologii opartych na języku JavaScript i XML. Dzięki zastosowaniu tej technologii, coraz bardziej popularnej wśród programistów, aplikacje mogą działać w przeglądarce internetowej tak jak te, które są zainstalowane w komputerze użytkownika, a zmiana zawartości strony nie wymaga jej „przeładowania”.

Ajax pozwala na komunikowanie się z serwerem w tle, ułatwia tworzenie wyszukiwania na żywo i umożliwia uaktualnianie danych bez odświeżania strony. Funkcje te powodują, że znajduje zastosowanie tam, gdzie wyświetlany tekst jest stale uaktualniany, na przykład w aplikacjach czatowych, arkuszach kalkulacyjnych lub formularzach zamieszczanych na serwerach oraz w wielu innych sytuacjach.

„Ajax. Biblia” to kompletny przewodnik zawierający praktyczne porady, proste instrukcje i interesujące sztuczki dla webmasterów. Ułatwia opanowanie podstaw języka JavaScript, XML, poznanie dynamicznego HTML i CSS, ale także omawia bardziej zaawansowane zagadnienia, takie jak eliminowanie zapisywania w pamięci podręcznej przeglądarki lub stosowanie ukrytych kontrolki dla sprawdzenia, czy użytkownik widział już daną stronę. Korzystając z tego podręcznika, nauczysz się stosować biblioteki Ajaksa i JavaScriptu oraz kaskadowe arkusze stylów. „Biblia” porusza też wszelkie istotne kwestie dotyczące bezpieczeństwa, w tym także zagadnienia związane z używaniem filtrów w Ajaksie.

- Wykorzystanie zaawansowanych bibliotek Ajaksa i JavaScriptu
- Obsługa XML za pomocą JavaScriptu
- Stylizacja tekstu, kolorów i tła za pomocą CSS
- Praca z PHP
- Interpolacja zmiennych w łańcuchach znaków
- Tworzenie list argumentów o zmiennej długości
- Obsługa kontrolki HTML
- Tworzenie dynamicznych tabel
- Bezpieczeństwo w aplikacjach opartych na Ajaksie
- Ograniczanie dostępu za pomocą filtrów

Twórz strony w Ajaksie, a będą szybkie, sprawne i absolutnie bezpieczne!



Spis treści

O autorze	13
Przedmowa	15
Część I Podstawy Ajaksa	19
Rozdział 1. Wprowadzenie do Ajaksa	21
Co oznacza „Ajax”?	24
Co można zrobić za pomocą Ajaksa?	26
Tworzenie wyszukiwania na żywo opartego na Ajaksie	26
Tworzenie kalkulatora opartego na Ajaksie	27
Rozmawianie za pomocą aplikacji czatowych opartych na Ajaksie	28
Przetwarzanie liczb za pomocą arkuszy kalkulacyjnych	29
Przeglądanie strony Amazon	31
Otrzymywanie odpowiedzi dzięki autouzupełnianiu za pomocą Ajaksa	31
Logowanie się za pomocą Ajaksa	33
Pobieranie obrazków	33
Przeciąganie i upuszczanie za pomocą Ajaksa	35
Gry i zabawy z Ajaksem	39
Modyfikacja stron internetowych w locie	39
Dodanie interaktywności do map	45
Poczta elektroniczna oparta na Ajaksie	47
Podsumowanie	47
Rozdział 2. Zapoznanie się z JavaScriptem	51
Jakie miejsce w Ajaksie zajmuje JavaScript?	51
Krótka historia JavaScriptu	53
Rozpoczęcie pracy z JavaScriptem	55
Naprawianie błędów	57
Komentowanie kodu	59
Umieszczanie kodu w plikach zewnętrznych	61
Odpowiadanie na zdarzenia przeglądarki	61
Tworzenie funkcji w JavaScriptcie	64
Przekazywanie argumentów do funkcji	69
Zwracanie wartości z funkcji	73
Przechowywanie danych w zmiennych	75
Działania na danych wykonywane za pomocą operatorów	78
Podejmowanie decyzji za pomocą instrukcji if	81

Wykonywanie innego kodu w różnych przeglądarkach	83
Praca z pętlami	88
Wykonywanie pętli za pomocą for	88
Wykonywanie pętli za pomocą while	90
Podłączanie JavaScriptu do przycisków HTML	93
Podsumowanie	95
Rozdział 3. Tworzenie aplikacji opartych na Ajaksie	97
Pisanie kodu w Ajaksie	97
Tworzenie aplikacji	99
Pisanie kodu w JavaScriptcie	100
Tworzenie obiektu XMLHttpRequest	101
Otwieranie obiektu XMLHttpRequest	105
Obsługa pobieranych danych	107
Otrzymywanie danych	111
Pobieranie danych	112
Inne sposoby tworzenia obiektów XMLHttpRequest	113
Interakcja z kodem po stronie serwera	115
Przekazywanie danych do skryptów po stronie serwera	117
Przekazywanie danych do serwera za pomocą metody GET	118
Przekazywanie danych do serwera za pomocą metody POST	121
Wykorzystywanie Ajaksa z XML	126
Tworzenie kodu XML	126
Pobieranie XML	128
Podsumowanie	134
Rozdział 4. Rozszerzone techniki programowania w Ajaksie	135
Praca z wieloma jednoczesnymi żądaniami XMLHttpRequest	136
Wykorzystywanie wielu obiektów XMLHttpRequest	138
Przechowywanie obiektów XMLHttpRequest w tablicy	141
Wykorzystywanie funkcji wewnętrznych	144
Obsługiwanie kodu w JavaScriptcie przysłanego z serwera	150
Zwracanie kodu w JavaScriptcie	151
Zwracanie obiektów JavaScriptu	154
Łączenie się z Google Suggest	155
Wywoływanie innej domeny w Ajaksie	163
Przykład logowania	164
Praca z Ajaksem i żądaniami HEAD	166
Otrzymanie wszystkich danych z nagłówek	166
Otrzymywanie określonych danych z nagłówka	168
Eliminowanie zapisywania w pamięci podręcznej przeglądarki	169
Podsumowanie	171
Część II Ajax w szczegółach	173
Rozdział 5. Wprowadzenie do bibliotek Ajaksa	175
Wykorzystywanie Ajax Utility Library	176
Otrzymywanie tekstu z serwera za pomocą funkcji getText	178
Otrzymywanie XML z serwera za pomocą funkcji getXml	184
Przesyłanie danych do serwera i otrzymywanie z powrotem tekstu	189
Przesyłanie danych do serwera i otrzymywanie z powrotem XML	194

Wykorzystywanie libXmlRequest do pobierania XML	202
Wykorzystywanie AJAXLib do pobierania XML	206
Podsumowanie	208
Rozdział 6. Bardziej zaawansowane biblioteki JavaScriptu	209
Wykorzystywanie biblioteki Majax	210
Wykorzystywanie biblioteki Sack	213
Wykorzystywanie biblioteki XHConn	218
Wykorzystywanie biblioteki uniAjax	223
Wykorzystywanie biblioteki AjaxGear	225
Wykorzystywanie biblioteki AjaxRequest	230
Wykorzystywanie biblioteki Http w celu uniknięcia zapisywania danych w pamięci podręcznej	234
Wykorzystywanie biblioteki Sarissa do obsługi XML	238
Wykorzystywanie biblioteki Interactive Website Framework do obsługi XML	242
Podsumowanie	244
Rozdział 7. Wykorzystywanie bibliotek Ajaksa po stronie serwera	245
Praca z bibliotekami opartymi na PHP	246
Wykorzystywanie Sajax oraz PHP	246
Wykorzystywanie biblioteki Xajax oraz PHP	253
Wykorzystywanie biblioteki LibAjax oraz PHP	258
Praca z bibliotekami opartymi na Javie	263
Wykorzystywanie Direct Web Remoting oraz Javy	263
Wykorzystywanie biblioteki Ajax Tags oraz Javy	266
Wykorzystywanie biblioteki SWATO oraz Javy	268
Praca z Ajaksem oraz Ruby on Rails	270
Pobieranie tekstu za pomocą Ajaksa	272
Dostęp do pobranych danych w kodzie	277
Podsumowanie	283
Część III Ajax oraz DOM, XML, CSS i dynamiczny HTML	285
Rozdział 8. DOM oraz obsługa zdarzeń	287
Wprowadzenie do DOM	287
Dodawanie elementów za pomocą DOM	289
Zastępowanie elementów z wykorzystaniem DOM	294
Tworzenie koszyka z zakupami opartego na Ajaksie	299
Tworzenie obiektu zdarzenia myszy działającego w różnych przeglądarkach	303
Obsługa zdarzeń związanych z naciskaniem przycisku myszy	308
Obsługa zdarzeń ruchu myszy	311
Obsługa zdarzeń zwalniania przycisku myszy	312
Pobieranie obrazków w Ajaksie	316
Obsługa przekroczenia czasu odpowiedzi w Ajaksie	319
Podsumowanie	324
Rozdział 9. XML oraz Ajax	325
Tworzenie XML	325
Obsługa XML za pomocą JavaScriptu	330
Pobieranie danych z dokumentu XML	334
Obsługa białych znaków w przeglądarkach internetowych Mozilla, Netscape oraz Firefox	342
Obsługa białych znaków w sposób działający we wszystkich przeglądarkach	345

Bezpośredni dostęp do danych XML	348
Dostęp do wartości atrybutów XML	353
Sprawdzanie poprawności kodu XML	356
Podsumowanie	359
Rozdział 10. Kaskadowe arkusze stylów oraz Ajax	361
Menu oparte na Ajaksie	362
Tworzenie stylów	363
Praca z myszą	370
Pobieranie elementów menu	376
Wyświetlanie menu	377
Obsługa kliknięcia elementu menu	379
Ukrywanie menu	380
Uwydatnianie tekstu w aplikacjach opartych na Ajaksie	381
Przewijający się tekst	385
Stylizacja tekstu za pomocą CSS	387
Stylizacja kolorów oraz tła za pomocą CSS	390
Stylizacja położenia na stronach internetowych	394
Podsumowanie	396
Rozdział 11. Dynamiczny HTML oraz Ajax	397
Tworzenie efektów powiązanych z przesunięciem kursora myszy nad element	397
Wykorzystywanie stylów dynamicznych	401
Korzystanie z document.write	404
Wykorzystywanie metod dynamicznego HTML do uaktualnienia części strony	410
Wykorzystywanie właściwości dynamicznego HTML do uaktualniania części strony	413
Wykorzystywanie zakresów tekstu do uaktualniania części strony	418
Wykorzystywanie createElement do tworzenia nowych elementów w locie	421
Tworzenie dynamicznych tabel	426
Przyciąganie uwagi użytkownika za pomocą animacji	431
Podsumowanie	436
Część IV Zaawansowany Ajax	437
Rozdział 12. Wprowadzenie do Ajaksa oraz PHP	439
Rozpoczęcie pracy z PHP	439
Odsyłanie tekstu do przeglądarki	441
Odsyłanie kodu XML do przeglądarki	442
Komentowanie kodu PHP	445
Praca ze zmiennymi	446
Interpolacja zmiennych w łańcuchach znaków	456
Praca z tablicami	458
Tworzenie tablic	458
Kopiowanie tablic	461
Obsługa danych za pomocą operatorów	462
Dokonywanie wyboru za pomocą instrukcji if	463
Wykorzystywanie instrukcji else	466
Wykorzystywanie instrukcji elseif	467
Wykorzystywanie instrukcji switch	469

Wykorzystywanie pętli	470
Wykorzystywanie pętli for	470
Wykorzystywanie pętli while	472
Wykorzystywanie pętli do...while	474
Wykorzystywanie pętli foreach	474
Podsumowanie	476
Rozdział 13. PHP — funkcje oraz kontrolki HTML	477
Praca z funkcjami	477
Przekazywanie danych do funkcji	480
Podawanie argumentów domyślnych	481
Tworzenie list argumentów o zmiennej długości	483
Zwracanie wartości z funkcji	485
Zwracanie wielu wartości z funkcji	486
Obsługa kontrolki HTML	488
Praca z polami tekstowymi	491
Praca z polami wyboru	494
Praca z przyciskami opcji	497
Praca z listami	499
Praca z kontrolkami z hasłem	504
Praca z mapami obrazu	506
Przesyłanie plików na serwer	508
Podsumowanie	514
Rozdział 14. Obsługa danych od użytkownika w PHP	515
Wyświetlanie wszystkich danych z formularza	516
Spojrzenie na zmienne serwera	520
Umieszczanie danych od użytkownika w tablicach	523
Umieszczanie aplikacji PHP na jednej stronie	526
Wykorzystywanie ukrytych kontrolki do sprawdzania, czy użytkownik widział już daną stronę	530
Sprawdzanie danych wprowadzanych przez użytkownika, liczb oraz tekstu	531
Sprawdzanie danych wprowadzanych przez użytkownika	531
Sprawdzanie poprawności liczb	538
Sprawdzanie poprawności tekstu	543
Obsługa kodu HTML przesłanego przez użytkownika	547
Podsumowanie	550
Rozdział 15. Ajax a bezpieczeństwo	551
Ochrona przed złośliwymi użytkownikami	551
Wstawianie złośliwego kodu w JavaScriptcie	552
Sprawy bezpieczeństwa związane z serwerami proxy	559
Bezpieczeństwo w aplikacjach opartych na Ajaksie	560
Łatwa ochrona hasłem	560
Uwierzytelnienie użytkownika po stronie serwera	563
Ochrona hasła za pomocą kluczy publicznych oraz prywatnych	577
Podsumowanie	587
Rozdział 16. Filtry, MVC oraz Ajax	589
Ograniczanie dostępu za pomocą filtrów	589
Logowanie dostępu użytkowników za pomocą filtrów	598

Ograniczanie dostępu w oparciu o porę dnia	608
Aplikacje typu MVC	612
Kontroler	614
Widok	616
Model	617
Instalacja aplikacji MVC	618
Dostęp do MVC za pomocą Ajaksa	619
Podsumowanie	620
Skorowidz	621

Rozdział 1.

Wprowadzenie do Ajaksa

W tym rozdziale:

- ◆ Wprowadzenie do Ajaksa
- ◆ Spojrzenie na Ajaksa w działaniu
- ◆ Wykonywanie wyszukiwania na żywo za pomocą Ajaksa
- ◆ Wykorzystywanie czatów opartych na Ajaksie
- ◆ Wykorzystywanie opcji autouzupełniania
- ◆ Przeciąganie i upuszczanie za pomocą Ajaksa
- ◆ Wykorzystywanie koszyków z zakupami opartych na Ajaksie

Witamy w książce *Ajax. Biblia!* Można w niej znaleźć wszystko na temat Ajaksa. Ajax jest nową technologią, która podbija Internet, zdobywając kolejne strony internetowe. Rewolucja Ajaksa nadeszła i odmienia oblicze Internetu. Tak naprawdę Ajax jest podstawą tego, co nazywane jest Web 2.0 — następną wersją World Wide Web (WWW).

O co zatem w tym wszystkim chodzi? Podstawową ideą jest sprawienie, by aplikacje webowe wyglądały i zachowywały się podobnie do aplikacji desktopowych. Warto spojrzeć na przykład na rysunek 1.1, na którym widać znajomą stronę wyszukiwarki Google. Po wpisaniu szukanego pojęcia, takiego jak „Ajax”, należy kliknąć przycisk *Szukaj w Google*.

Ekran miga w czasie, gdy jest uaktualniany nowymi danymi, po czym pokazują się strony pasujące do podanego zapytania, które odnalazła wyszukiwarka Google — jak widać to na rysunku 1.2.

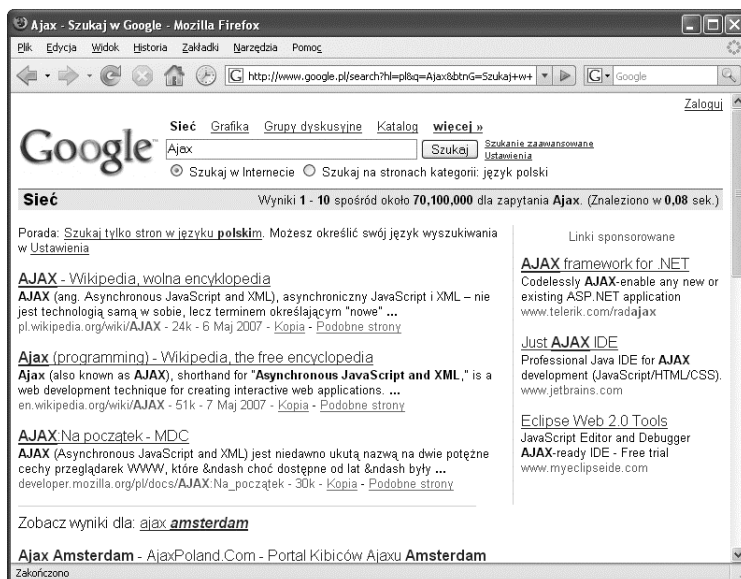
Funkcjonuje to dobrze, jednak nie jest to sposób, w jaki działa Ajax. Dzięki Ajaksowi można działać w tle, łącząc się z serwerem w celu otrzymania danych bez konieczności odświeżania strony w przeglądarce. Warto na przykład przyjrzeć się stronie Google Suggest, dostępnej pod adresem www.google.com/webhp?complete=1&hl=en, która pokazana jest na rysunku 1.3.

W miarę wpisywania kolejnych liter strona w przeglądarce łączy się z serwerem Google i wyszukuje dopasowania dla części szukanego zapytania, która została już wpisana. Kiedy na przykład wpisze się „aj”, można zauważyć, że pojawi się lista rozwijana — widoczna na rysunku 1.4 — z propozycjami Google pasującymi do wpisywanego tekstu.

Rysunek 1.1.
Google wyszukuje strony pasujące do podanego pojęcia

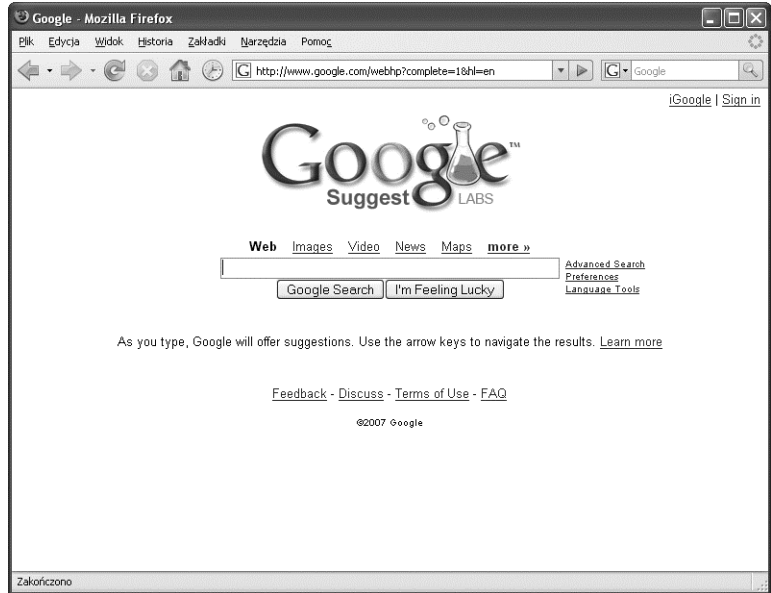


Rysunek 1.2.
Google wyświetla odnalezione wyniki zapytania

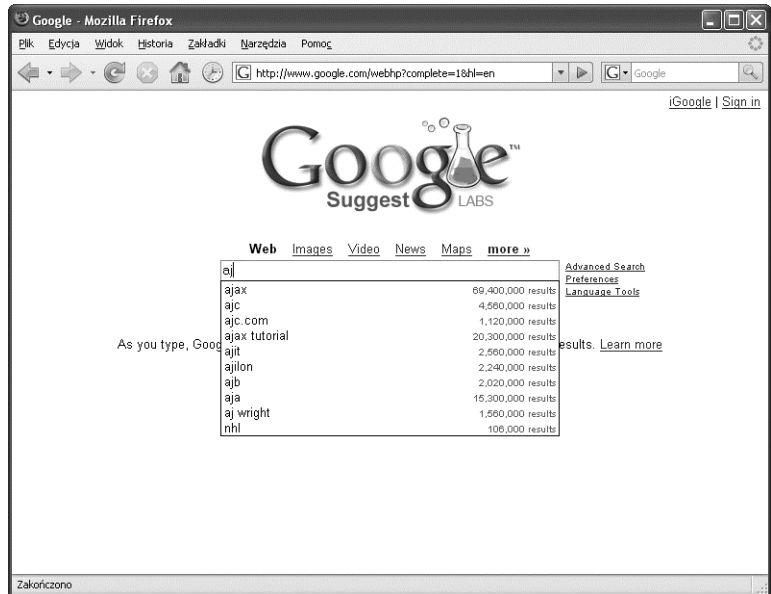


W tle, dzięki wykorzystywaniu technik Ajaksa, strona internetowa łączy się z Google Suggest i wyszukuje propozycje pasujące do wpisywanego zapytania. Następnie wyświetla listę rozwijaną zawierającą odnalezione propozycje, pozwalając na wybranie jednej z nich — a wszystko to bez odświeżania strony. To jest właśnie kluczowa część: odświeżanie strony nie jest konieczne. W dawnych czasach, kiedy chciało się przesłać dane do serwera, należało kliknąć jakiś przycisk, na przykład *Szukaj w Google*. Następnie należało poczekać chwilę, aż strona na ekranie zamigocze i odświeży się. Teraz strona internetowa może przysyłać dane do serwera bez wymuszania odświeżania, tak jak widać to

Rysunek 1.3.
Strona Google
Suggest



Rysunek 1.4.
Google Suggest
szuka pasujących
wyników w miarę
wpisywania
zapytania



na poprzednim przykładzie, w którym szukane zapytanie było automatycznie przesyłane do serwera, a serwer odsyłał z powrotem dane wyświetlane na liście rozwijanej.

Obecnie nie trzeba już wykonywać odświeżania strony przy przesyłaniu danych do serwera lub odczytywaniu ich. Zamiast tego aplikacje webowe mogą działać podobnie do aplikacji desktopowych, przysyłając dane do serwera i pobierając je z powrotem w tle.

Taka właśnie zmiana aplikacji webowych, sprawiająca, że zachowują się one w sposób zbliżony do aplikacji desktopowych, stanowi sedno Web 2.0. Czy podobałoby się nam, gdyby edytor tekstu migał za każdym razem, gdy wpisuje się nowy znak, gdyby cały dokument był ciągle wyświetlany od nowa, a kursor powracał na jego początek? Nie wydaje się to być zbyt atrakcyjną propozycją. Dzięki Ajaksowi można tworzyć internetowe edytory tekstu, których w praktyce prawie nie da się odróżnić od wersji desktopowych — nie ma w nich migania, nie ma zmiany pozycji kursora w trakcie pisania. Takie edytory tekstu działają płynnie, dokładnie tak samo jak desktopowe wersje tych samych aplikacji.

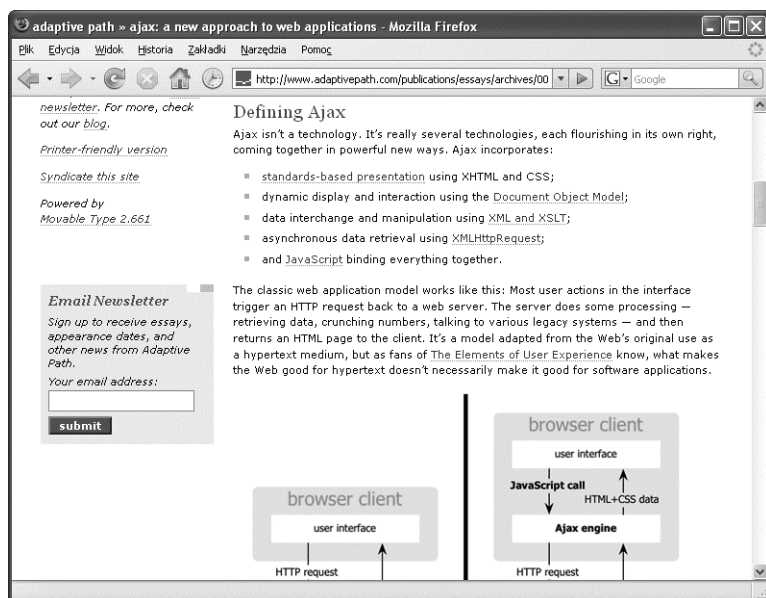
Wiadomo już, dlaczego Ajax wywołuje rewolucję w sposobie działania aplikacji webowych — teraz możliwe jest tworzenie aplikacji internetowych, które wyglądają i działają dokładnie tak samo jak ich desktopowe odpowiedniki.

Niniejszy rozdział jest wprowadzeniem. Prezentuje genezę znaczenia pojęcia „Ajax”, a następnie omawia, w jaki sposób Ajax jest obecnie wykorzystywany. Taki przegląd jest bardzo ważnym elementem niniejszej książki, ponieważ z Ajaksem można spotkać się coraz częściej — czasem dość niespodziewanie — i jeśli jest się zaznajomionym z jego możliwymi zastosowaniami, będzie się wiedziało, w jaki sposób można go zastosować we własnych aplikacjach webowych.

Co oznacza „Ajax”?

Skąd więc wzięła się nazwa „Ajax”? Warto przyjrzeć się rysunkowi 1.5 prezentującemu pierwszy artykuł napisany na temat Ajaksa — artykuł, w którym pierwszy raz pojawiła się ta nazwa i od którego wszystko się zaczęło. Można go znaleźć pod adresem www.adaptivepath.com/publications/essays/archives/000385.php. Tekst ten został napisany przez Jesse Jamesa Garretta z Adaptive Path, który pierwszy nadał tej technologii nazwę „Ajax”.

Rysunek 1.5.
Oryginalny artykuł
na temat Ajaksa



Artykuł rozpoczyna się tak:

„Jeśli cokolwiek w dziedzinie współczesnego projektowania interaktywnego może być nazwane wspaniałym, będzie to tworzenie aplikacji webowych. W końcu kiedy ostatni raz głośno było o jakimkolwiek projekcie interaktywnym, który nie znajdował się w Internecie (no dobrze, poza iPodem)? Wszystkie olśniewające, innowacyjne, nowe projekty znajdują się w Internecie.

Pomimo tego projektanci interaktywni działający w Internecie muszą nieco zazdrościć kolegom, którzy tworzą oprogramowanie desktopowe. Aplikacje desktopowe charakteryzują się bogactwem i interaktywnością, jaka w Internecie wydawała się być niedostępna. Ta sama prostota, która umożliwiła szybki rozwój Internetu, stworzyła również barierę pomiędzy możliwościami, jakie można udostępnić, a tymi istniejącymi w aplikacjach desktopowych.

Bariera ta została właśnie pokonana”.

Bariera ta naprawdę została zniesiona — dzięki Ajaksowi. Co zatem, zgodnie z artykułem, oznacza „Ajax”? Nazwa ta oznacza *Asynchronous JavaScript and XML* (po polsku: asynchroniczny JavaScript i XML). Jak wynika z samej nazwy, a także z lektury artykułu Jesse Jamesa Garretta, Ajax to tak naprawdę zbiór technologii.

Część „asynchroniczny” oznacza, że przeglądarka nie czeka na powrót danych z serwera, a zamiast tego potrafi obsługiwać te dane, kiedy powrócą. Innymi słowy, transfer danych odbywa się w tle, bez konieczności wykonywania pauzy przez przeglądarkę i oczekiwania, aż coś się zdarzy. Jest to kluczowy element Ajaksa: można obsługiwać dane z serwera, kiedy serwer je odeśle. Nie jest konieczne zatrzymywanie całej aplikacji i oczekiwanie na przybycie danych. Gdyby trzeba było czekać na dane, aplikacja byłaby synchroniczna. W przypadku wolnego połączenia z Internetem jest to problematyczne.

Również JavaScript w nazwie Ajax ma ogromne znaczenie, gdyż właśnie ta technologia wykorzystywana jest do użycia Ajaksa w przeglądarce. Ajax oparty jest na JavaScriptcie z przeglądarki w zakresie łączenia się z serwerem i obsługiwania danych, które serwer odsyła. Wszystkie aplikacje ajaksowe, które będzie się tworzyć w niniejszej książce, wykorzystują JavaScript do łączenia się w tle z serwerem, przesyłania i pobierania danych. A kiedy dane zostaną pobrane, JavaScript w przeglądarce może zostać wykorzystany do ich obsługiwania, wyświetlając dane bądź przetwarzając je tak, jak jest to konieczne.

Co jednak z XML pojawiającym się w rozszerzeniu nazwy Ajax? Jak wiadomo, XML stał się lingua franca Internetu, dostarczając tekstowe sposoby przesyłania danych w Sieci. Powodem tego, że XML stał się tak popularny, jest fakt, iż rzeczywiście jest on oparty na tekście, co oznacza, że można go wykorzystywać w Internecie, ponieważ Internet zaprojektowany został do obsługi dokumentów tekstowych (czyli HTML). Z tej przyczyny aplikacje ajaksowe pisane są często tak, by obsługiwały dane przesyłane z serwera z użyciem XML. Innymi słowy, kiedy aplikacja kontaktuje się z serwerem, odeśle on dane z powrotem jako dokument XML.

Tak naprawdę XML jest tylko jednym ze sposobów obsługi danych przesyłanych z serwera. Można je również wysyłać jako zwykły tekst, a obie te techniki będą wielokrotnie wykorzystywane w niniejszej książce.

Oprócz JavaScriptu i XML Ajax współpracuje również z dynamicznym HTML (DHTML) oraz kaskadowymi arkuszami stylów (CSS). Obie te technologie pozwalają na uaktualnianie danych wyświetlanych na stronie internetowej; ponieważ w przypadku Ajaksa nie przerysowuje się całej strony, a jedynie jej część, DHTML oraz CSS wykorzystywane są dość często. Technologie te pozwalają na uaktualnienie wybranych części strony internetowej. W niniejszej książce dynamiczny HTML oraz CSS pojawiają się w wielu miejscach, ponieważ pozwalają one na odświeżanie jedynie części strony internetowej, co jest kluczowym elementem aplikacji opartych na Ajaksie.

Częścią JavaScriptu, która to umożliwia, jest obiekt XMLHttpRequest. Jest to specjalny obiekt wbudowany w wersje JavaScriptu znajdujące się we wszystkich współczesnych przeglądarkach. Jak zostanie to pokazane, obiekt ten umożliwia połączenie się z serwerem i obsługiwanie danych odesłanych z serwera w tle. Ajax oparty jest nie po prostu na JavaScriptcie, a przede wszystkim na obiekcie XMLHttpRequest będącym częścią JavaScriptu.

Tak to zatem wygląda — Ajax jest zbiorem technologii, a nie pojedynczą technologią. Do połączenia z serwerem wykorzystuje się obiekt XMLHttpRequest wbudowany w JavaScript, następnie obsługuje się XML (lub zwykły tekst), który serwer odsyła z powrotem, za pomocą JavaScriptu. Dynamiczny HTML oraz CSS wykorzystywane są do wyświetlania rezultatów w przeglądarce. Całe szczęście, że te wszystkie części aplikacji opartych na Ajaksie — JavaScript, obiekt XMLHttpRequest, dynamiczny HTML oraz CSS — łączą się ze sobą w taki sposób, bo dzięki temu razem sprawiają, że aplikacje internetowe wyglądają tak jak aplikacje desktopowe.

Tak naprawdę technologie stojące za Ajaksem istnieją mniej więcej od 1998 roku i były już używane przez wiele aplikacji takich jak Outlook Web Access firmy Microsoft. Nie przyjęły się jednak aż do początku 2005 roku, kiedy znane aplikacje, takie jak Google Suggest, zaczęły je wykorzystywać, a Jesse James Garrett napisał artykuł tworzący nazwę Ajax, co pozwoliło skupić ten zbiór technologii pod jednym dachem.

Od tego czasu wszystko potoczyło się szybko, a programiści uświadomili sobie, że oprogramowanie webowe może w końcu zacząć się zachowywać i działać tak jak oprogramowanie desktopowe. Co zatem można zrobić za pomocą Ajaksa? Kwestii tej poświęcona jest dalsza część niniejszego rozdziału.

Co można zrobić za pomocą Ajaksa?

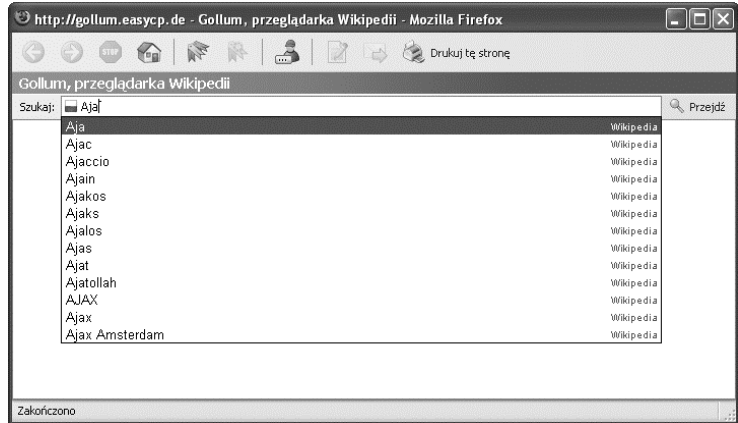
Za pomocą Ajaksa można zrobić bardzo wiele, a na kolejnych stronach omówiona zostanie ta skarbnica możliwości. Zamieszczono tam szeroką prezentację tego, do czego obecnie wykorzystywany jest Ajax.

Tworzenie wyszukiwania na żywo opartego na Ajaksie

Jednym z najpopularniejszych zastosowań Ajaksa jest tworzenie **wyszukiwania na żywo** (ang. *live search*), którego przykładem był Google Suggest omówiony na początku niniejszego rozdziału. Dzięki wyszukiwaniu na żywo użytkownik może wpisać część szukanego pojęcia, a aplikacja webowa połączy się z serwerem za pomocą Ajaksa i odnajdzie propozycje, które pasują do podanego niepełnego zapytania.

W Internecie dostępnych jest wiele narzędzi służących do wyszukiwania na żywo, nie tylko Google Suggest. Można na przykład przyrzeć się aplikacji o nazwie Gollum znajdującej się pod adresem <http://gollum.easycp.de/en/>, która na żywo przeszukuje Wikipedię, wolną encyklopedię internetową, dostępną pod adresem www.wikipedia.org. Aplikacja Gollum zaprezentowana została na rysunku 1.6.

Rysunek 1.6.
Gollum na żywo
przeszukuje
Wikipedię

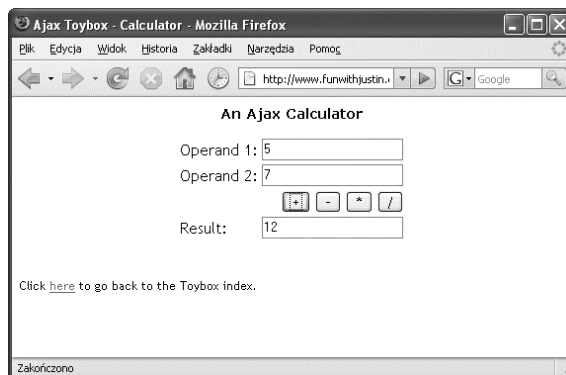


Po wybraniu języka działania aplikacji oraz odpowiedniej wersji językowej Wikipedii wystarczy podać część szukanego pojęcia — jak na przykład „Aja” dla Ajaksa — by zobaczyć rezultat widoczny na rysunku 1.6, gdzie Gollum połączył się w tle z polskim wydaniem Wikipedii i odnalazł propozycje pasujące do podanej części szukanego pojęcia. Jak to zwykle bywa w przypadku wyszukiwania na żywo, propozycje te wyświetlane są na liście rozwijanej i można z nich wybrać pasującą użytkownikowi. Po zrobieniu tego w przeglądarce otwierany jest odpowiedni artykuł z Wikipedii.

Tworzenie kalkulatora opartego na Ajaksie

Każda sytuacja, w której trzeba przesłać dane do serwera, a następnie obsłużyć dane odebrane z powrotem w tle, jest wręcz stworzona dla Ajaksa. Dlaczego więc nie utworzyć kalkulatora opartego na Ajaksie? Przykład takiej aplikacji znajduje się na stronie www.funwithjustin.com/ajax-toybox-demo/calculator.html i został zaprezentowany na rysunku 1.7.

Rysunek 1.7.
Kalkulator oparty
na Ajaksie



By wykorzystywać kalkulator, wystarczy wprowadzić dwa argumenty, na których będzie się działać, takie jak 5 i 7 z rysunku 1.7. Następnie należy wybrać działanie, które chce się wykonać — dodawanie, odejmowanie, mnożenie lub dzielenie. Dzięki Ajaksovi strona internetowa przesyła argumenty do znajdującego się na serwerze programu, który odpowiednio dodaje, odejmuje, mnoży bądź dzieli otrzymane liczby i odsyła wynik z powrotem.

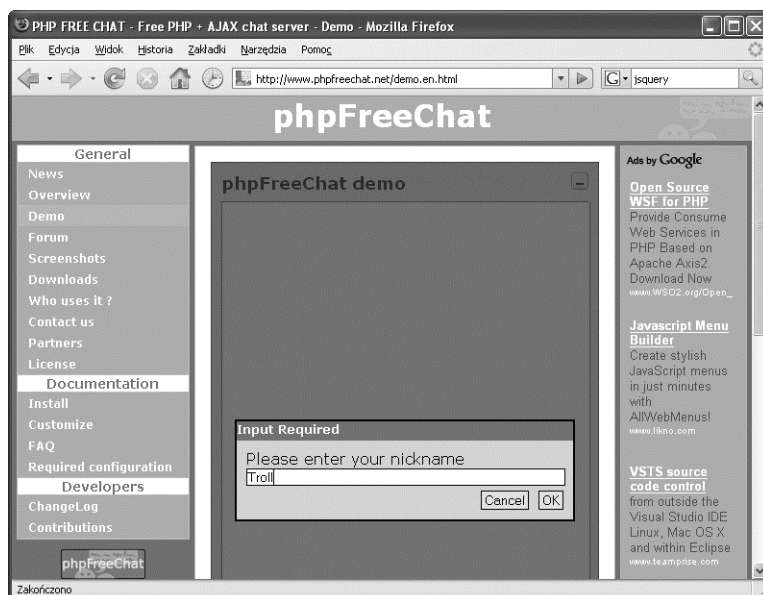
Wynik pojawia się następnie w dolnym polu tekstowym, jak widać na rysunku 1.7, na którym dodawane są do siebie liczby 5 oraz 7. Wszystko to odbywa się bez odświeżania strony — strona nie miga. Aplikacja ta, tak jak wiele innych programów opartych na Ajaksie, wygląda tak, jakby była aplikacją desktopową.

Rozmawianie za pomocą aplikacji czatowych opartych na Ajaksie

Ajax sprawdza się doskonale wszędzie tam, gdzie wymagane jest częste uaktualnianie danych, na przykład w aplikacjach czatowych, w których dowolna liczba użytkowników może wpisywać tekst, jaki następnie automatycznie powinien być widoczny dla wszystkich zalogowanych osób. Ajax jest tutaj dobrym wyborem, ponieważ wyświetlany tekst jest stale uaktualniany, a obserwowanie jego migania w czasie uaktualniania strony byłoby bardzo denerwujące.

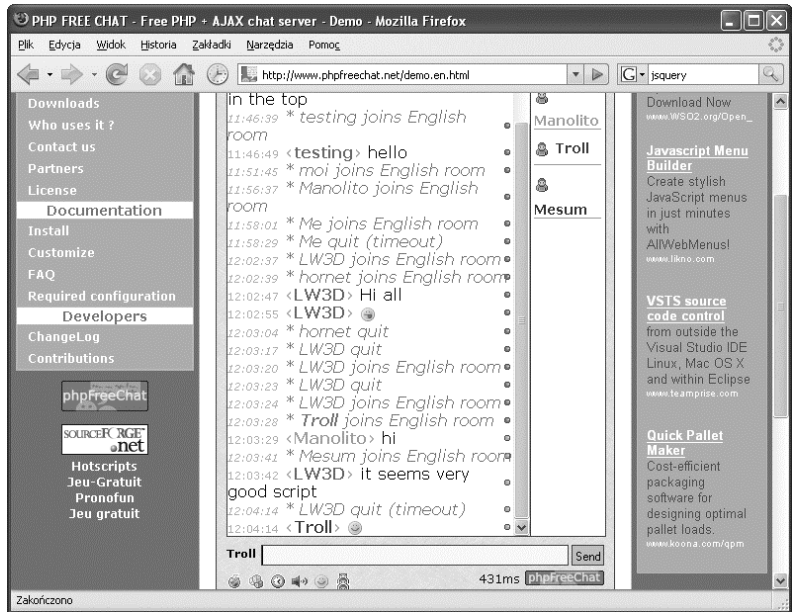
Za pomocą Ajaksa można jednak uaktualniać tekst znajdujący się w dowolnym miejscu na stronie bez konieczności odświeżenia jej. Można na przykład przyrzeć się stronie www.phpfreechat.net/demo.en.php, na której znajduje się PHP Free Chat. Strona ta łączy się ze skryptem PHP znajdującym się na serwerze w celu obsługiwania aplikacji czatowej. Kiedy wejdzie się na stronę PHP Free Chat, najpierw należy podać nazwę użytkownika, jak na rysunku 1.8.

Rysunek 1.8.
Logowanie się do PHP Free Chat



Po zalogowaniu się można wpisywać tekst do pola tekstowego znajdującego się na dole rysunku 1.9. Po naciśnięciu klawisza *Enter* bądź kliknięciu *Send* w aplikacji tekst jest za pomocą Ajaksa przesyłany do serwera, który dodaje go do tekstów wpisywanych przez innych użytkowników, a rezultaty wyświetlane są w oknie czatu, jak widać to na rysunku 1.9.

Rysunek 1.9.
PHP Free Chat
pozwala wpisywać tekst, który widzą inni użytkownicy



Istnieje wiele aplikacji czatowych opartych na Ajaksie. Można na przykład zajrzeć na stronę www.plasticshore.com/projects/chat/, pokazaną na rysunku 1.10. By korzystać z tej aplikacji czatowej, wystarczy podać swoje imię (lub przyjąć domyślnie zaproponowaną nazwę użytkownika) i wpisać wiadomość, po czym kliknąć przycisk *Submit*. Tekst pojawi się następnie w oknie czatu innych użytkowników.



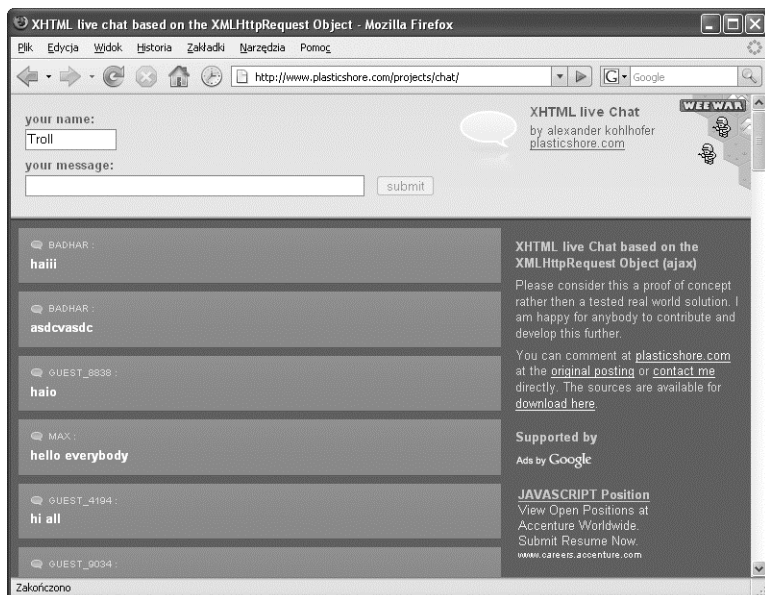
Istnieje o wiele więcej aplikacji czatowych opartych na Ajaksie. Kolejną dobrą propozycją jest na przykład aplikacja dostępna pod adresem <http://treehouse.ofb.net/chat/?lang=en>.

Przetwarzanie liczb za pomocą arkuszy kalkulacyjnych

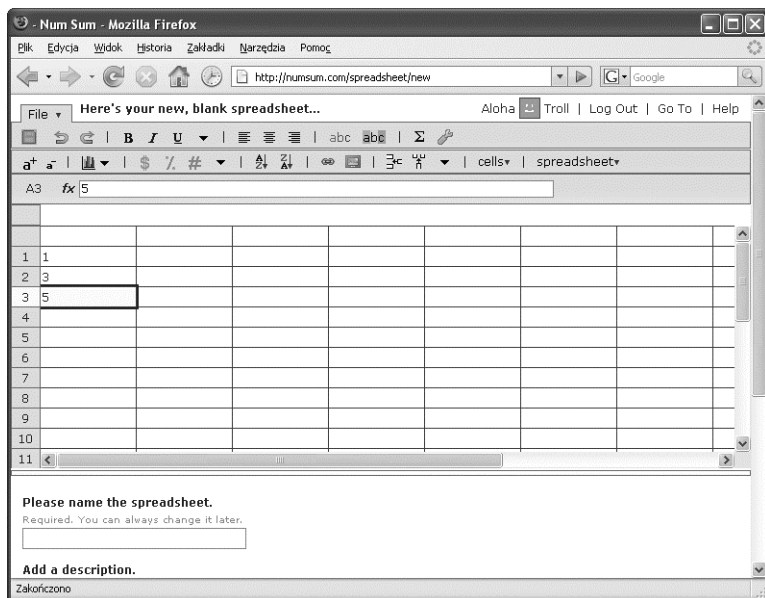
Coraz więcej aplikacji desktopowych przenosi się do Internetu dzięki Ajaksowi. Jednym z ostatnich rodzajów takich aplikacji są arkusze kalkulacyjne, które można obecnie znaleźć w wielu miejscach.

Można na przykład przyrzeć się Num Sum, darmowemu internetowemu arkuszowi kalkulacyjnemu (<http://numsum.com/spreadsheet/new>), który działa tak, jak mogłaby działać jego wersja desktopowa. Num Sum w działaniu można zobaczyć na rysunku 1.11.

Rysunek 1.10.
Darmowa aplikacja
czatowa oparta
na Ajaksie



Rysunek 1.11.
Num Sum jest
internetową
aplikacją
z arkuszami
kalkulacyjnymi



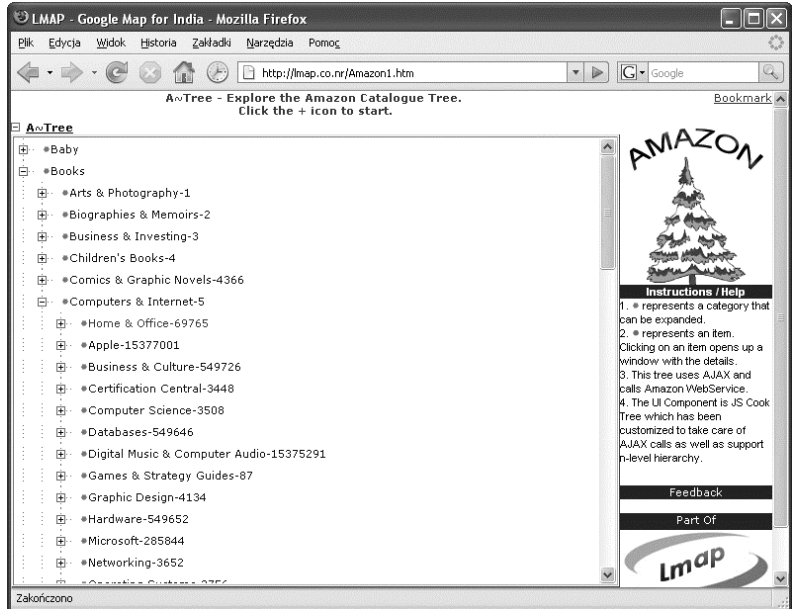
Dzięki aplikacji Num Sum można tworzyć prawdziwe arkusze kalkulacyjne, łącznie z obsługą formuł, a następnie zapisać dane na serwerze Num Sum. Wykorzystywanie tej aplikacji jest praktycznie nie do odróżnienia od wersji desktopowej — wpisywane dane są od razu uaktualniane bez konieczności odświeżania strony¹.

¹ Darmowe arkusze kalkulacyjne można również tworzyć z użyciem Google Docs & Spreadsheets (<http://docs.google.com>). Aplikacja ta ma interfejs dostępny w języku polskim — *przyj. thum*.

Przeglądanie strony Amazon

Oto ciekawa aplikacja: pozwala ona przeglądać produkty będące w ofercie sklepu internetowego Amazon.com. Strona ta pozwala na wyświetlanie oferty Amazon.com dzięki użyciu struktury hierarchicznej drzewa, uaktualnianej za pomocą Ajaksa. Wystarczy przejść na stronę <http://lmap.co.nr/Amazon1.htm> i kliknąć węzeł drzewa, by go otworzyć i zobaczyć produkty Amazon, jak widać na rysunku 1.12.

Rysunek 1.12.
Przeglądanie strony
Amazon.com



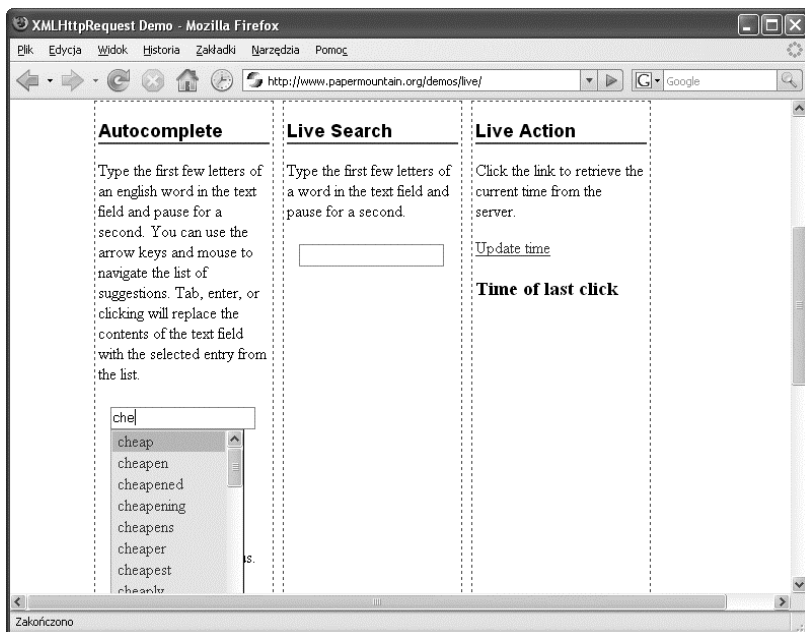
Otrzymywanie odpowiedzi dzięki autouzupełnianiu za pomocą Ajaksa

Aplikacje wykonujące autouzupełnianie w dużym stopniu przypominają wyszukiwanie na żywo. W przypadku wyszukiwania na żywo do pola tekstowego wpisuje się część szukanego pojęcia. W przypadku autouzupełniania w grę nie wchodzi żadne szukanie; autouzupełnianie po prostu oferuje podpowiedzi, które kończą wpisywane pojęcie (każda osoba wpisująca wiadomości SMS na telefonach komórkowych zapewne zna ten pomysł pod postacią słowników T9, które oferują propozycje słów pasujące do wpisywanych znaków).

Przykład autouzupełniania widoczny jest na stronie www.papermountain.org/demos/live/, która została przedstawiona na rysunku 1.13.

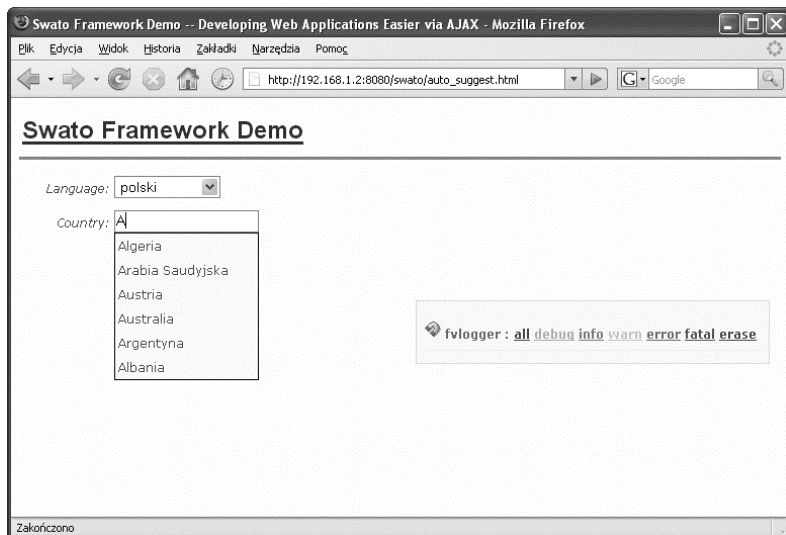
Wystarczy wpisać początek angielskiego słowa w polu tekstowym widocznym na rysunku 1.13, a aplikacja prześle tę część słowa do serwera, który znajduje propozycje pasujących wyrazów i odsyła je z powrotem. Propozycje te wyświetlane są tak jak na rysunku i można spośród nich wybrać właściwą. Kiedy się to zrobi, wybrane pojęcie zastąpi początek słowa, który wpisano wcześniej.

Rysunek 1.13.
Wykorzystywanie
autouzupełniania



Innym przykładem autouzupełniania, udostępnionym do pobrania, jest SWATO — zestaw narzędzi ajaksowych dostępny pod adresem <https://swato.dev.java.net/>. Przykład działania tej aplikacji widoczny jest na rysunku 1.14.

Rysunek 1.14.
Wykorzystywanie
SWATO do
autouzupełniania



Przykład ten dopasowuje nazwy krajów, jak widać na rysunku 1.14. Wystarczy wpisać jakąś literę, na przykład „A”, by pokazały się możliwe dopełnienia, takie jak Arabia Saudyjska, Austria, Australia i tak dalej.

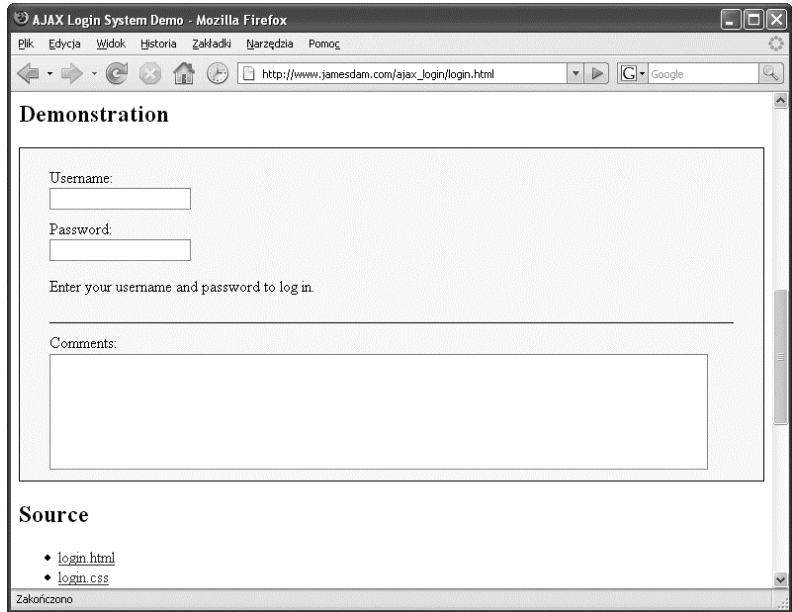
Logowanie się za pomocą Ajaksa

Kolejny dobry przykład: zamiast prosić użytkowników o przejście przez kilka stron w celu zalogowania się na stronie internetowej, można wykorzystać Ajaksa do uproszczenia tego procesu, sprawdzając wpisane identyfikatory użytkowników oraz hasła w tle.

Można na przykład spojrzeć na stronę www.jamesdam.com/ajax_login/login.html, pokazaną na rysunku 1.15. Strona ta pozwala na automatyczne logowanie się za pomocą Ajaksa bez konieczności odświeżenia strony.

Rysunek 1.15.

Strona
z logowaniem
oparta na Ajaksie



Jeśli poda się niepoprawny identyfikator użytkownika i hasło, na przykład Troll i idzie-troll, i kliknie gdziekolwiek na stronie, otrzyma się komunikat informujący o błędzie („Invalid username and password combination”), jak na rysunku 1.16.

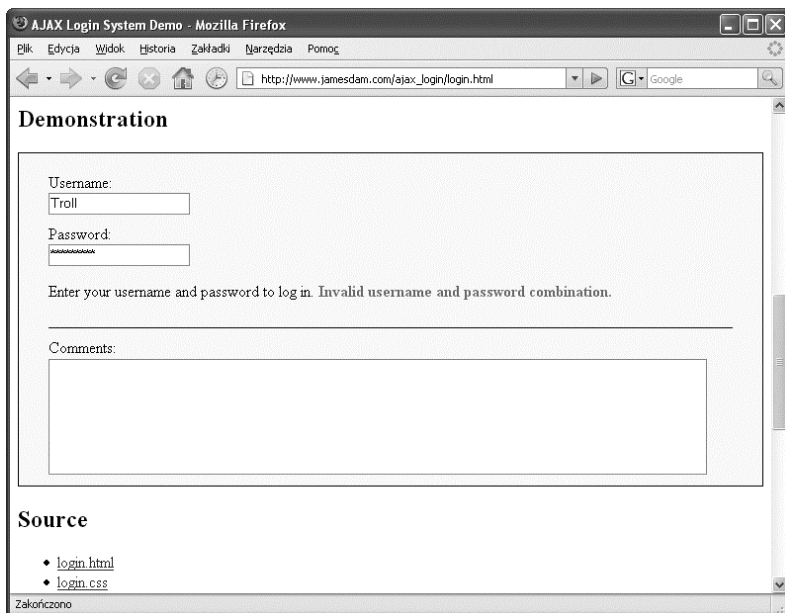
Z drugiej strony, kiedy poda się poprawne dane — na przykład identyfikator user1 i hasło pass1 — i kliknie gdzieś na stronie, będzie widać, że logowanie się powiodło, jak pokazano to na rysunku 1.17.

Pobieranie obrazków

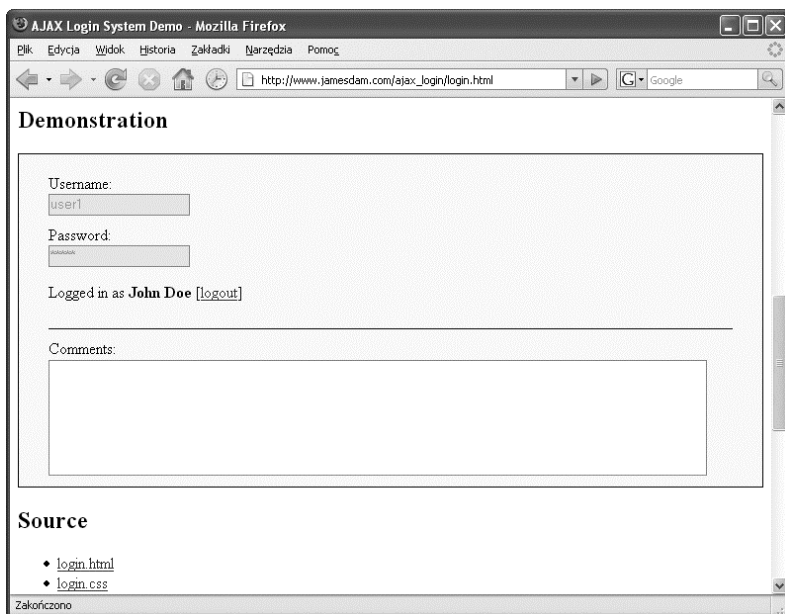
Za pomocą Ajaksa można pobierać tylko tekst i XML. Czy aby na pewno? W jednym z przykładów zaprezentowanych w książce zastosowano Ajaksa wraz z dynamicznym HTML do pobierania obrazków. Przykład ten zaprezentowany jest na rysunku 1.18.

Kiedy kliknie się przycisk, przykład pobiera i wyświetla obrazek, jak widać na rysunku 1.19. Czy Ajax naprawdę pobiera ten obrazek? Czy technologia ta nie miała być ograniczona do pobierania tekstu i XML?

Rysunek 1.16.
Logowanie się
jest zablokowane

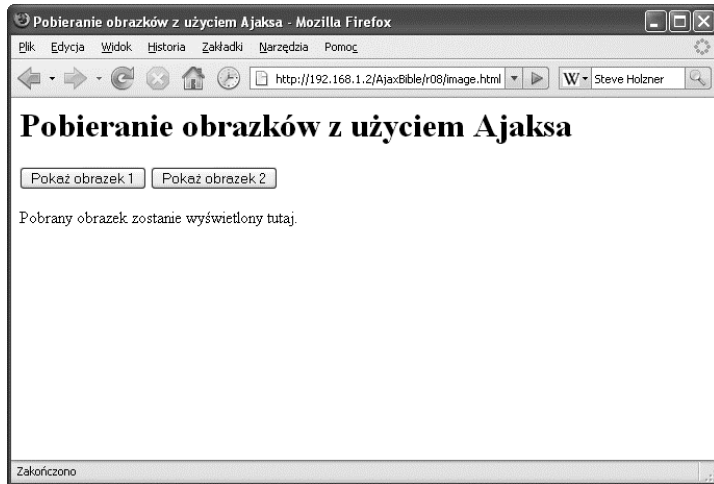


Rysunek 1.17.
Logowanie
za pomocą Ajaksa
się powiodło

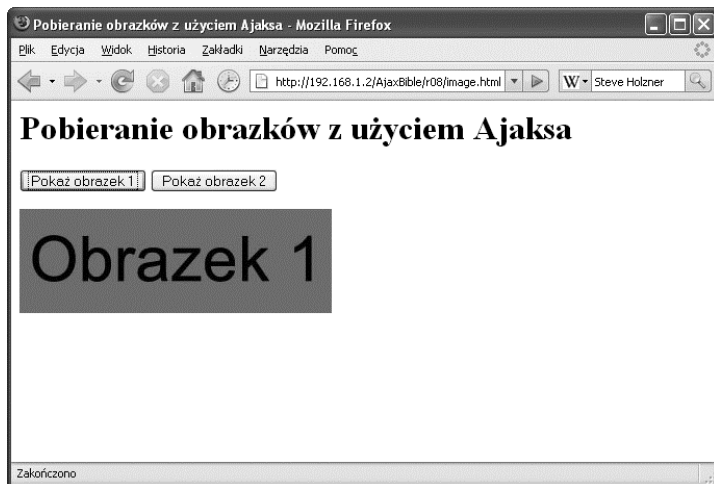


Tak naprawdę Ajax pobiera nazwę nowego obrazka, który ma zostać wyświetlony. Następnie w przykładzie wykorzystywany jest JavaScript, który nadpisuje element HTML `` w stronie internetowej, wykorzystując do tego nazwę pliku do pobrania. Kiedy przeglądarka widzi, że element `` został nadpisany, pobiera obrazek, na który wskazuje ten element, dzięki magii dynamicznego HTML.

Rysunek 1.18.
W tym przykładzie
można pobierać
obrazki



Rysunek 1.19.
W tym przykładzie
można pobrać
nowy obrazek
bez odświeżania
strony



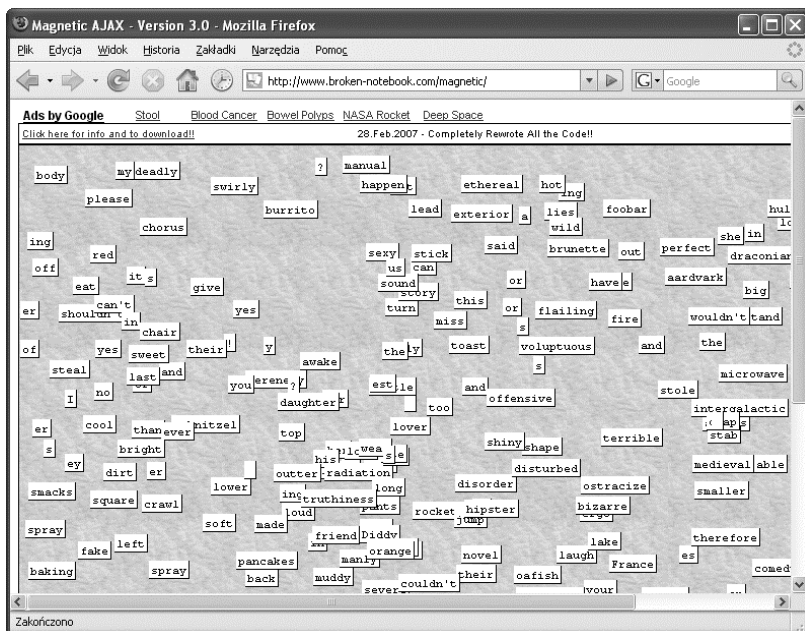
Rezultat jest taki, że po kliknięciu przycisku pojawia się nowy obrazek i odświeżenie strony nie jest do tego potrzebne. W ten sposób działa kombinacja Ajaksa i dynamicznego HTML, co wskazuje na to, że naprawdę będzie można pobierać dane binarne za pomocą Ajaksa.

Przeciąganie i upuszczanie za pomocą Ajaksa

W miarę jak aplikacje webowe coraz bardziej upodabniają się do aplikacji desktopowych, coraz więcej elementów znanych z tych ostatnich pojawia w aplikacjach internetowych. Operacje przeciągania i upuszczania czynią na przykład życie użytkownika o wiele prostszym i teraz, gdy przeciąga się i upuszcza elementy na stronie internetowej, serwer może zostać powiadomiony o tym fakcie w tle. Oznacza to, że następnym razem, gdy spojrzysz się na stronę, element, który zostaje przeciągnięty i upuszczony, pojawi się w nowym miejscu, w którym go umieszczono.

Można na przykład przyjrzeć się zabawie w magnesy ze słowami, dostępnej pod adresem www.broken-notebook.com/magnetic/ i zaprezentowanej na rysunku 1.20.

Rysunek 1.20.
Słowa z magnesów
pozostają
w miejscu, w które
się je przesunie,
dopóki ktoś inny
ich nie przemieści



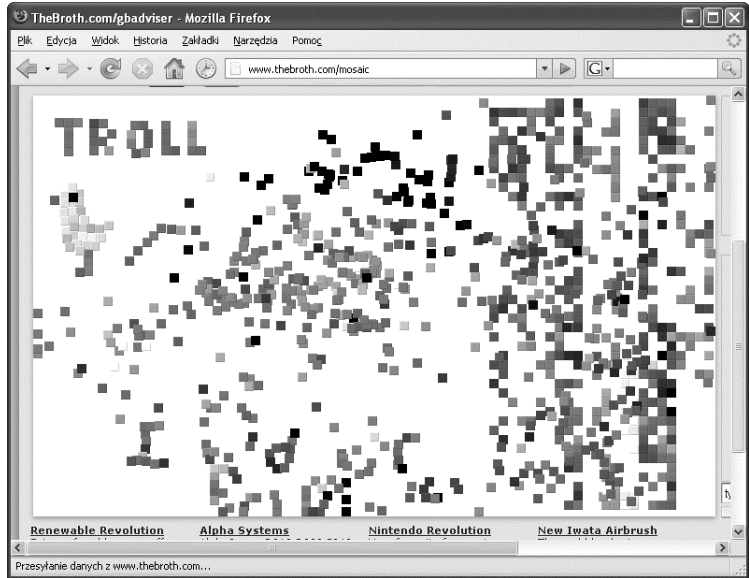
Kiedy przeciągnie się „magnes” w nowe miejsce, jego pozycja przesyłana jest do serwera za pomocą Ajaksa, a dane dotyczące położenia magnesu zostają zachowane. Inne osoby odwiedzające stronę widzą magnesy w pozycjach, w jakich się je ustawiło, a kiedy wraca się na stronę, magnesy nadal będą tam, gdzie się je pozostawiło — o ile ktoś ich nie przestawił.

Poniżej znajduje się inny przykład przeciągania i upuszczania — mozaika ze strony *thebroth.com*. Pomysł opiera się na tym, by użytkownicy przeciągali elementy tak, aby razem stworzyć wspólne dzieło, wykorzystując do tego Ajaksa. Kiedy przeciąga się element, jego nowe położenie przesyłane jest do serwera za pomocą Ajaksa, a nowa pozycja elementu uaktualniana jest wszędzie, w przeglądarce każdego z użytkowników. Mozaikę tę można znaleźć pod adresem www.thebroth.com/mosaic i jest ona również zaprezentowana na rysunku 1.21.

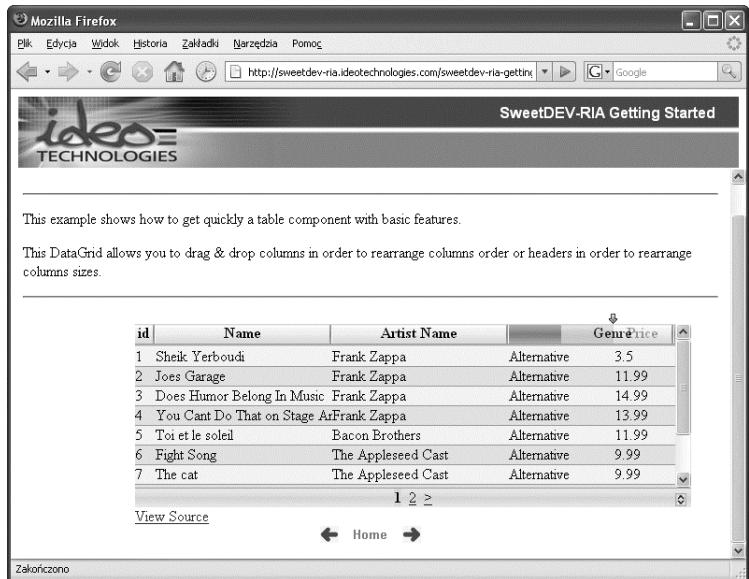
Przeciąganie i upuszczanie nie zawsze musi dotyczyć pojedynczych elementów. Można na przykład spojrzeć na siatkę danych Ideo Technologies z rysunku 1.22, którą można znaleźć pod adresem <http://sweetdev-ria.ideotechnologies.com/sweetdev-ria-gettingStarted-2.0/>. Kontrolki siatki danych pozwalają na zmianę kolejności kolumn poprzez przeciąganie ich, jak widać to na rysunku.

Kiedy przeciąga się kolumnę, nowe ustawienie kolumn przesyłane jest do serwera i przechowywane na nim, co oznacza, że można przechodzić na inne strony tej samej siatki danych (wykorzystując odnośniki liczbowe znajdujące się pod danymi), a ustawienie to zostanie zachowane.

Rysunek 1.21.
Tworzenie wspólnej
mozaiki



Rysunek 1.22.
Przeciąganie
kolumny w siatce
danych firmy Ideo
Technologies

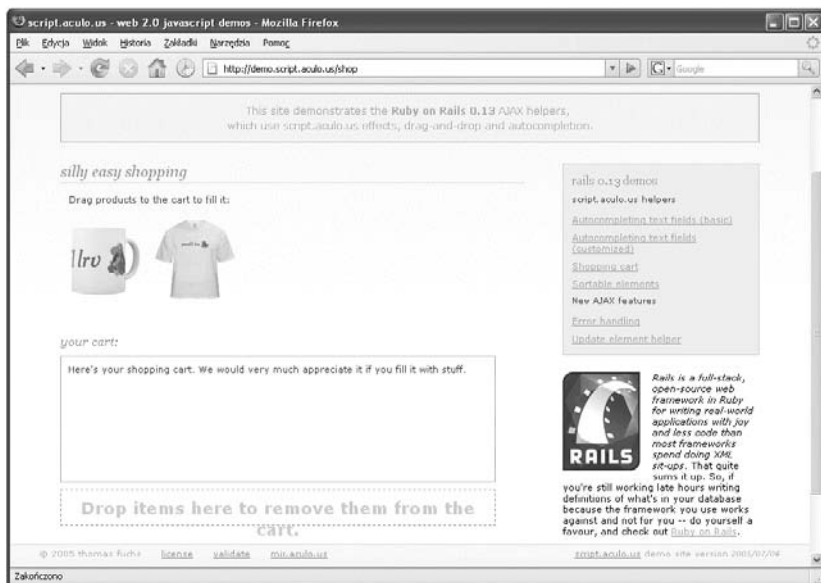


Jednym z najczęstszych zastosowań przeciągania i upuszczania za pomocą Ajaksa jest implementowanie koszyków z zakupami w sklepach internetowych. Normalnie, kiedy chce się dodać element do internetowego koszyka, trzeba przejść przez kilka stron. Kiedy kliknie się przycisk *Dodaj do koszyka*, pojawia się nowa strona odpowiadająca samemu koszykowi, a następnie trzeba wracać do stron z ofertami, by móc kontynuować zakupy.

Czy nie byłoby o wiele prościej, gdyby nie trzeba było w ogóle opuszczać strony z zakupami? Gdyby po prostu można było przeciągać przedmioty do koszyka, a serwer był o tym informowany w tle naszych zakupów? Nie ma problemu!

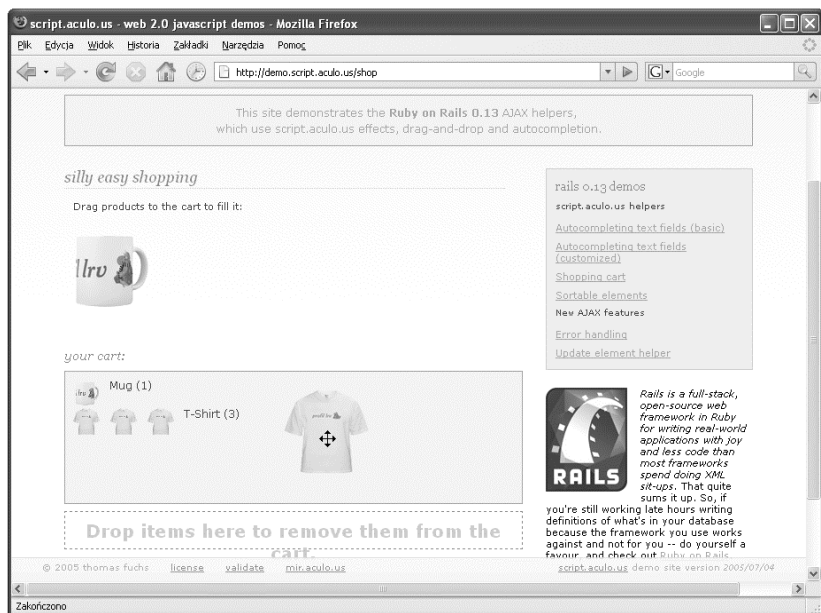
Właśnie tak działa przykładowy skrypt na stronie <http://demo.script.aculo.us/shop>, zaprezentowany na rysunku 1.23.

Rysunek 1.23.
Przeciąganie przedmiotu do koszyka z zakupami opartego na Ajaksie



Kiedy przeciąga się przedmiot bezpośrednio na stronie, a następnie upuszcza go do ajaxowego koszyka z zakupami, serwer jest powiadamiany o zakupie, a bieżące produkty są wyświetlane w koszyku, jak widać na rysunku 1.24.

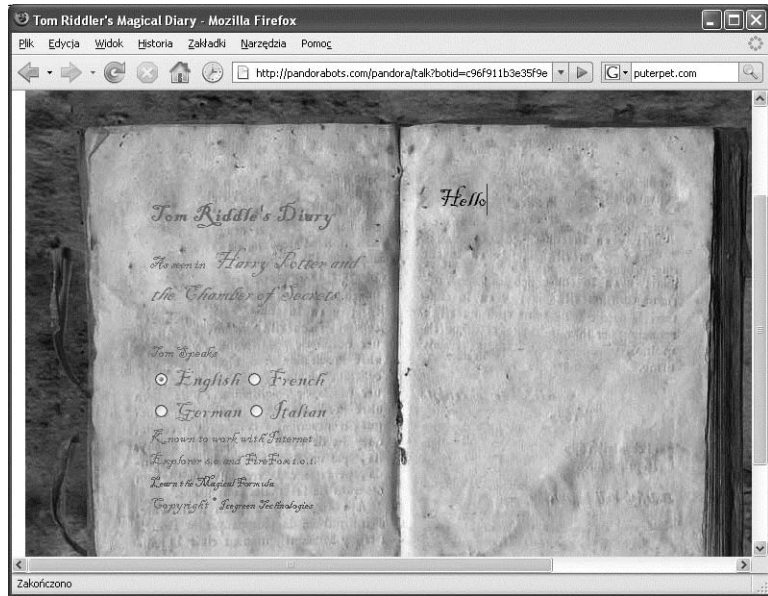
Rysunek 1.24.
Dodawanie przedmiotu do koszyka z zakupami opartego na Ajaksie



Gry i zabawy z Ajaksem

Poniżej znajduje się zabawne zastosowanie Ajaksa — „dziennik” w stylu Harry’ego Pottera, który odpowiada na to, co się do niego pisze po angielsku, niemiecku, francusku oraz włosku za pomocą Ajaksa. Można go znaleźć pod adresem <http://pandorabots.com/pandora/talk?botid=c96f911b3e35f9e1>; aplikacja ta zaprezentowana została na rysunku 1.25.

Rysunek 1.25.
Wykorzystywanie
dziennika w stylu
Harry’ego Pottera



Można spróbować napisać „Hello” w dzienniku. Słowo to od razu pojawia się w dzienniku, potem znika, by za chwilę pojawiła się odpowiedź — pobrana przez Ajaksa — widoczna na rysunku 1.26.

Dziennikowi można zadać całkiem sporo pytań, jak widać na rysunku 1.27, na którym pytany jest on o miejsce, gdzie się znajduje.

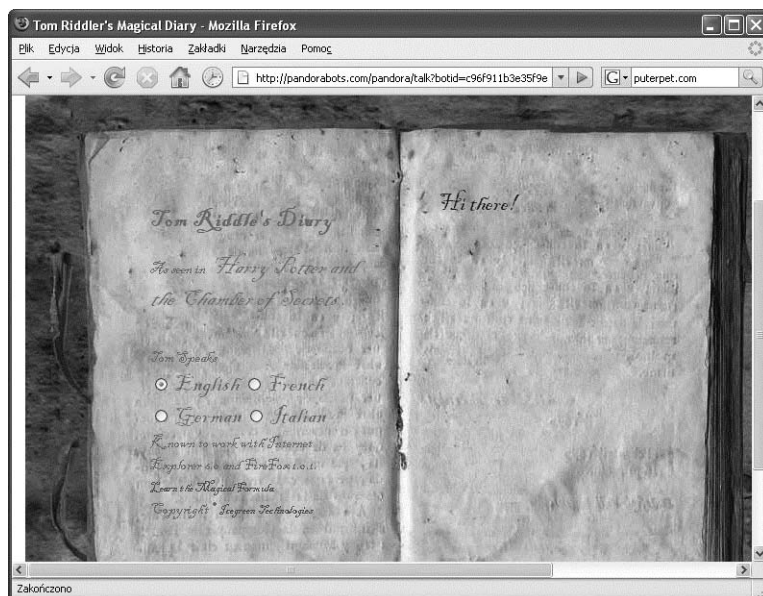
Odpowiedź dziennika widoczna jest na rysunku 1.28.

A może partyjka szachów opartych na Ajaksie? Wystarczy zajrzeć na stronę www.Jesper-Olsen.Net/PChess/ zaprezentowaną na rysunku 1.29. By przesunąć pionek, wystarczy go kliknąć, a następnie kliknąć jego nową pozycję — pionek automatycznie przesunie się w nowe położenie. Gra przesyła wszystkie potrzebne dane do serwera za pomocą Ajaksa, w tle, i otrzymuje potrzebne dane z powrotem. Następnie wykorzystuje CSS do odpowiedniego przesunięcia pionka.

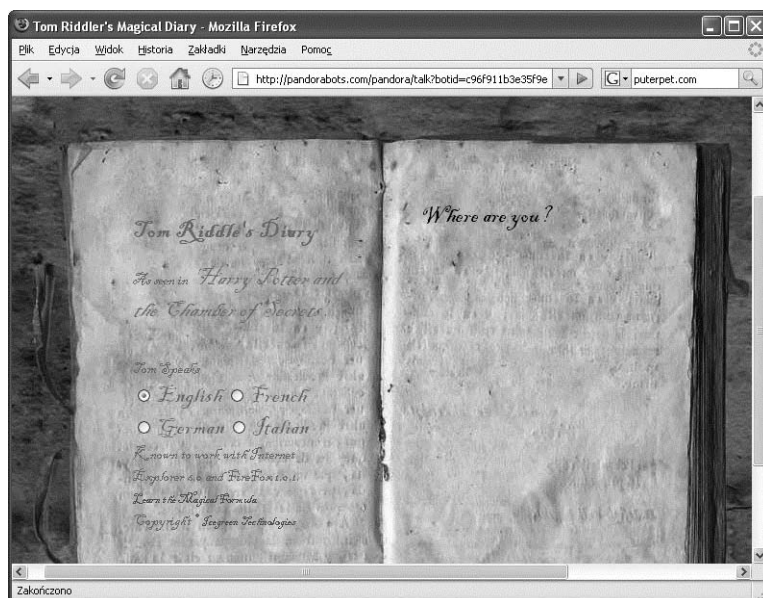
Modyfikacja stron internetowych w locie

Jednym z zadań, do których świetnie nadaje się Ajax, jest uaktualnianie lub modyfikowanie stron internetowych oglądanych przez użytkownika. Ponieważ aplikacje ajaksowe

Rysunek 1.26.
Dziennik odpowiada



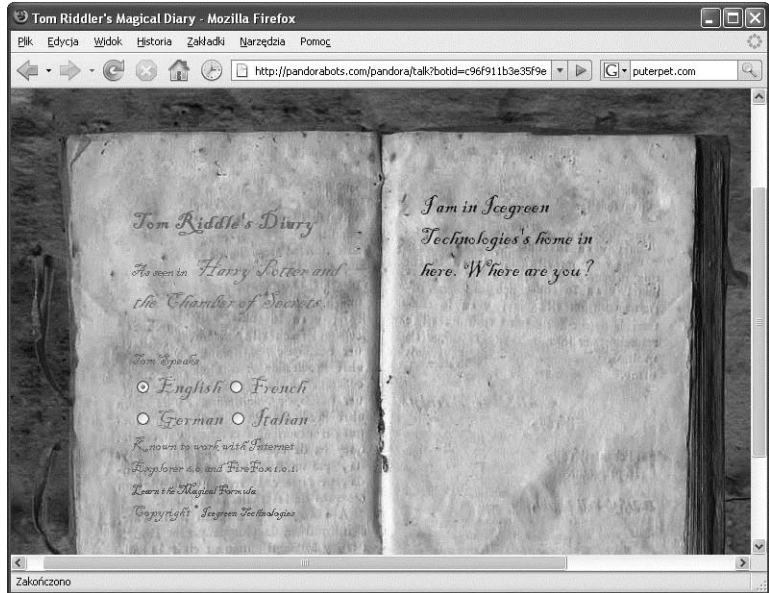
Rysunek 1.27.
Pytanie dziennika,
gdzie się znajduje



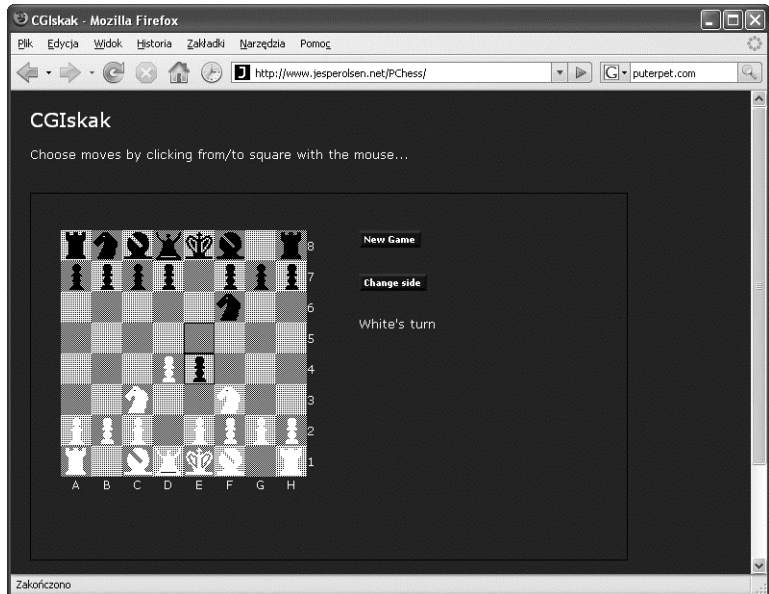
unikają całkowitego odświeżania strony wraz z towarzyszącym mu miganiem i błyskaniem, należy uaktualnić w przeglądarce jedynie wybrane części strony. Istnieją tysiące aplikacji opartych na Ajaksie, które działają w ten sposób, i w niniejszej książce pokazane zostanie tworzenie kilku z nich.

Poniżej znajduje się przykład, który już jest w Internecie — http://openrico.org/demos/inner_ajax_HTML, przedstawiony na rysunku 1.30.

Rysunek 1.28.
Dziennik
odpowiada,
gdzie się znajduje



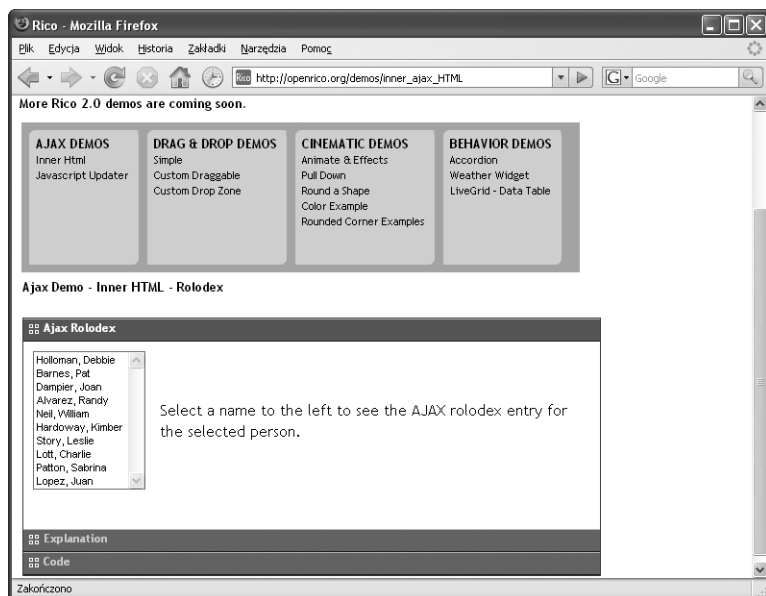
Rysunek 1.29.
Gra w szachy
oparta na Ajaksie



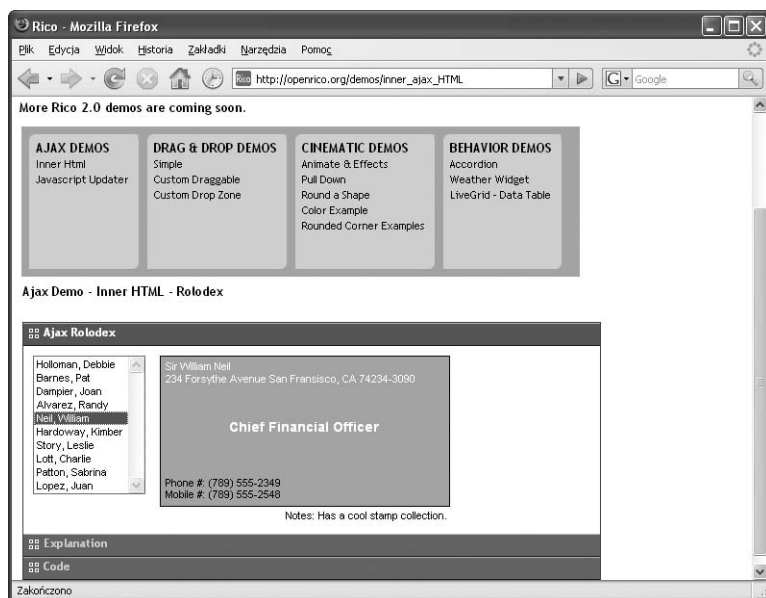
Przykład ten to oparta na Ajaksie kartoteka o nazwie Rolodex. Wystarczy kliknąć nazwisko osoby, a pojawi się „kartka” z informacjami o niej, jak widać to na rysunku 1.31.

Kartoteka Rolodex do wyświetlania danych wykorzystuje CSS. Za każdym razem, gdy kliknie się nazwisko osoby, aplikacja pobiera dane tej osoby z serwera, łączy je w jedną kartkę, a następnie wyświetla tę kartkę za pomocą CSS, modyfikując stronę internetową w locie.

Rysunek 1.30.
Kartoteka oparta
na Ajaksie



Rysunek 1.31.
Wykorzystywanie
kartoteki opartej
na Ajaksie

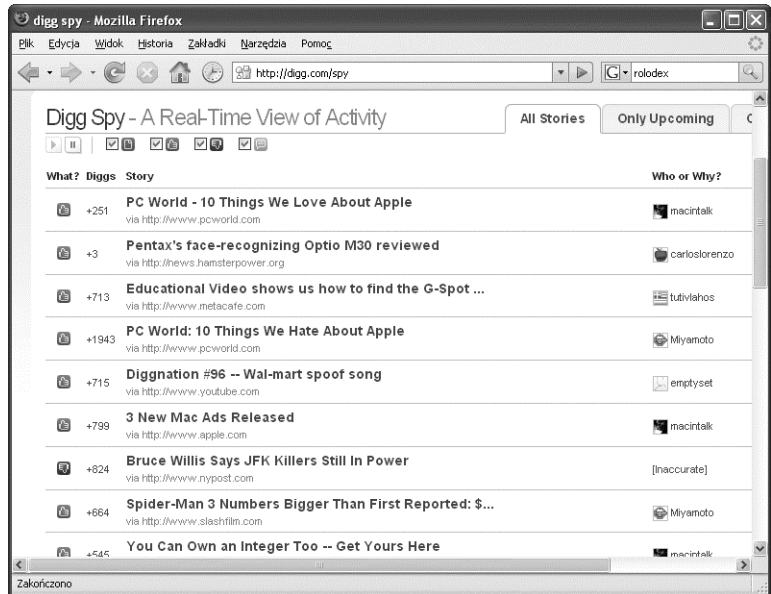


Właśnie tak to działa — wyniki jakiegoś działania wyświetlane są na stronie internetowej w momencie ich tworzenia. Nie ma konieczności wymuszania odświeżania strony, by były one widoczne.

Inny przykład modyfikowania strony internetowej znajduje się na stronie <http://digg.com/spy> pokazanej na rysunku 1.32. Strona ta prezentuje w czasie rzeczywistym artykuły, na które głoszą użytkownicy, i stale się uaktualnia dzięki połączeniu CSS oraz dynamicznego HTML. Lista artykułów, którą widać, zmienia się w czasie gdy użytkownicy na nie

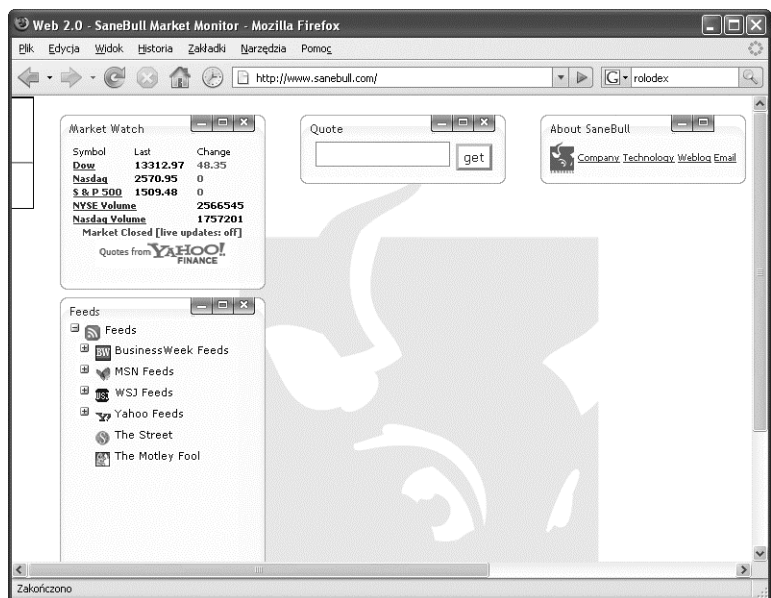
głosują, a każdy artykuł otrzymuje ikonę z kciukiem skierowanym do góry bądź na dół. Do pobierania danych z serwera wykorzystywany jest Ajax, dzięki czemu strona nie musi być odświeżana.

Rysunek 1.32.
Aplikacja Digg Spy
ciągle się
uaktualnia



Innym przykładem aplikacji wykorzystującej Ajaksa do ciągłego uaktualniania jest SaneBull Market Monitor. Można ją znaleźć na stronie www.sanebull.com; została również zaprezentowana na rysunku 1.33.

Rysunek 1.33.
SaneBull ciągle
uaktualnia
indeksy giełdowe
za pomocą Ajaksa

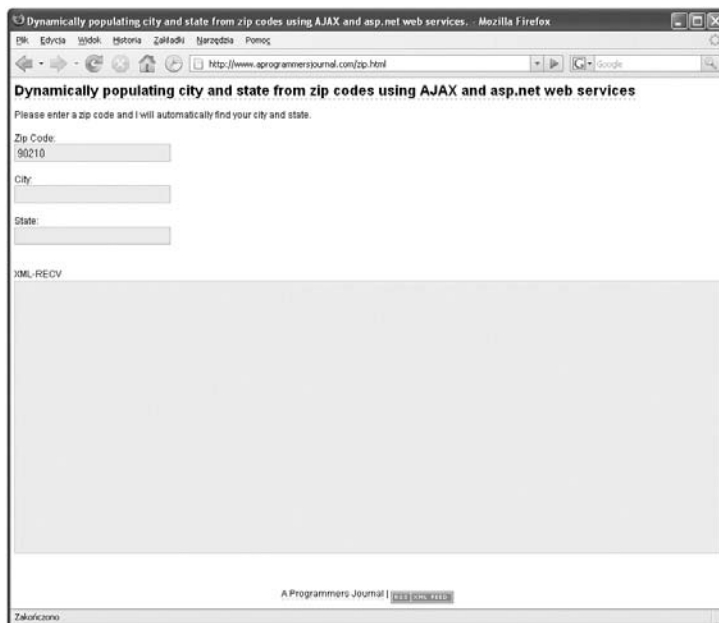


Program ten uaktualnia informacje z sekcji Market Watch znajdującej się po lewej stronie w czasie rzeczywistym, podając aktualne indeksy giełdowe dzięki modyfikowaniu strony internetowej.

Aplikacje oparte na Ajaksie czasami działają poprzez uaktualnianie kontrolek HTML znajdujących się na stronie internetowej, a nie tylko zmianę samego wyświetlania strony. Przykład takiego zachowania można znaleźć na stronie <http://www.a programmersjournal.com/zip.html>, na której zamieszczono aplikację odnajdującą lokalizację dla danego kodu pocztowego. Wystarczy wpisać kod pocztowy, a aplikacja podaje miasto i stan, w jakim się ono znajduje; dodatkowo wyświetlona będzie odpowiedź w formacie XML. Program ten można zobaczyć na rysunku 1.34.

Rysunek 1.34.

Znalezienie lokalizacji dla podanego kodu pocztowego



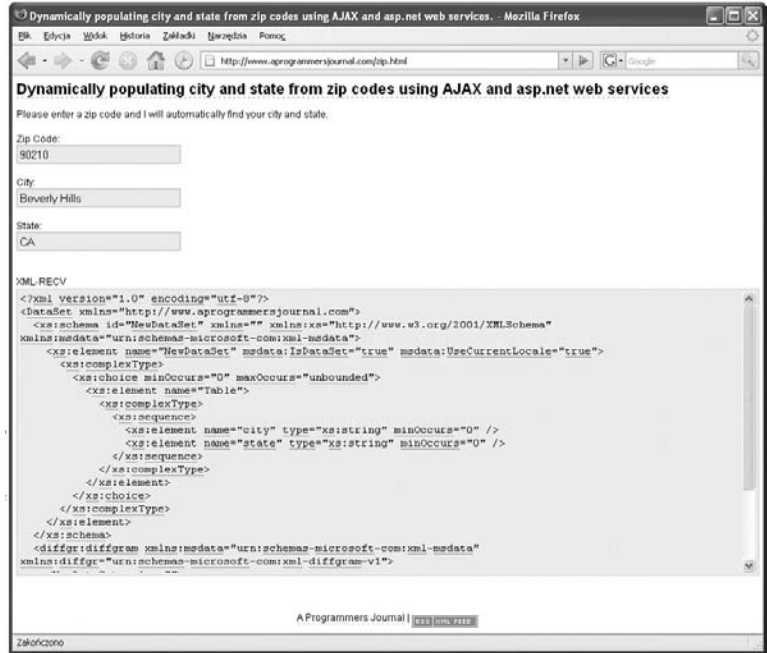
Po wpisaniu kodu pocztowego i kliknięciu myszą na stronie miasto oraz stan odpowiadające podanemu kodowi pocztowemu pojawiają się w polach tekstowych strony (jak widać na rysunku 1.35) — a wszystko to dzięki magii Ajaksa i dynamicznego HTML.

Jedną z najlepszych modyfikacji stron internetowych wszech czasów, a przy tym aplikacją, jakiej żaden przegląd rozwiązań opartych na Ajaksie nie może pominąć, jest Writely, dostępna pod adresem <http://docs.google.com>. Writely jest całkowicie internetowym edytorem tekstu, opartym na Ajaksie, a próbkę jego działania można zobaczyć na rysunku 1.36.

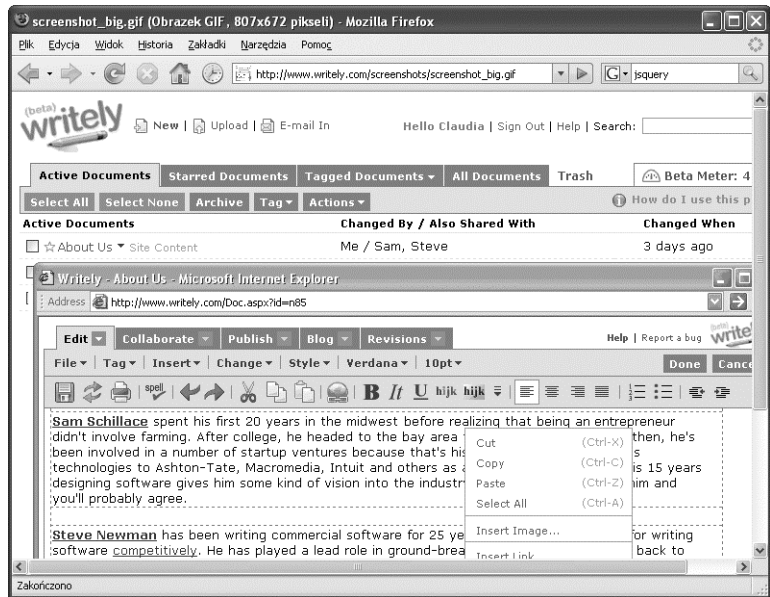
Writely pozwala na tworzenie i edytowanie dokumentów w zupełnie taki sam sposób, w jaki robi się to w przypadku aplikacji desktopowych. W celu uniknięcia odświeżania strony w czasie pracy użytkownika aplikacja ta oparta jest na Ajaksie. Writely jest jedną z aplikacji, które przenoszone są z komputerów do Internetu. Wcześniej pokazano już internetowe arkusze kalkulacyjne; teraz zaprezentowany został internetowy edytor tekstu — w obu przypadkach były to programy desktopowe, które teraz przeniesiono do Internetu.

Rysunek 1.35.

Wykorzystywanie narzędzia do odnajdywania lokalizacji dla podanego kodu pocztowego

**Rysunek 1.36.**

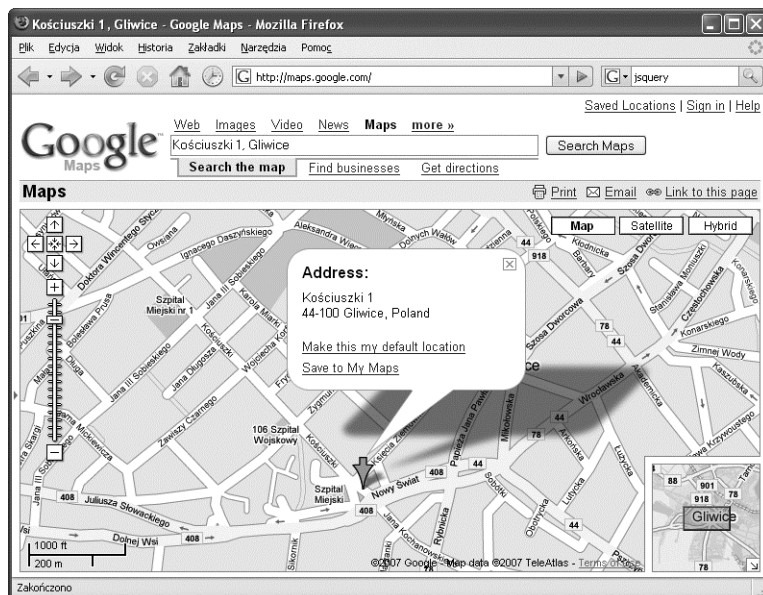
Przykładowa strona Writely



Dodanie interaktywności do map

Jedną z pierwszych aplikacji opartych na Ajaksie, od których wszystko się rozpoczęło, jest Google Maps, dostępna pod adresem <http://maps.google.com>. Na rysunku 1.37 pokazuje ona lokalizację siedziby Wydawnictwa Helion przy ulicy Kościuszki 1 w Gliwicach.

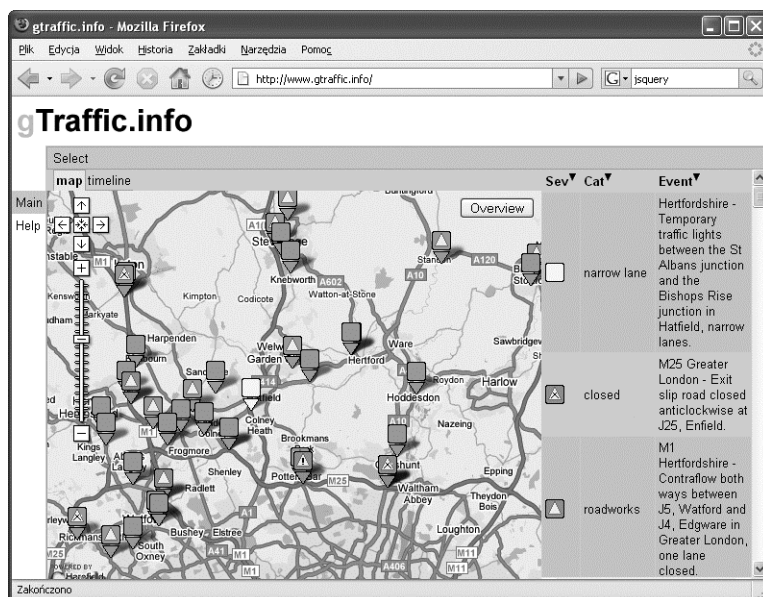
Rysunek 1.37.
Wykorzystywanie
Google Maps



Czy widać wyskakujące okienko, a także strzałkę wskazującą ul. Kościuski 1? Dane dla tych elementów zostały pobrane w tle za pomocą Ajaxa. Super!

Po wprowadzeniu Google Maps pojawiło się wiele innych stron z mapami. Można na przykład zajrzeć na stronę www.gtraffic.info, na której podaje się informacje o warunkach na drogach w Wielkiej Brytanii. Strzałki widoczne na mapie z rysunku 1.38 oparte są na Ajaxie.

Rysunek 1.38.
Sprawdzanie
warunków
na drogach
w Wielkiej Brytanii
za pomocą Ajaxa

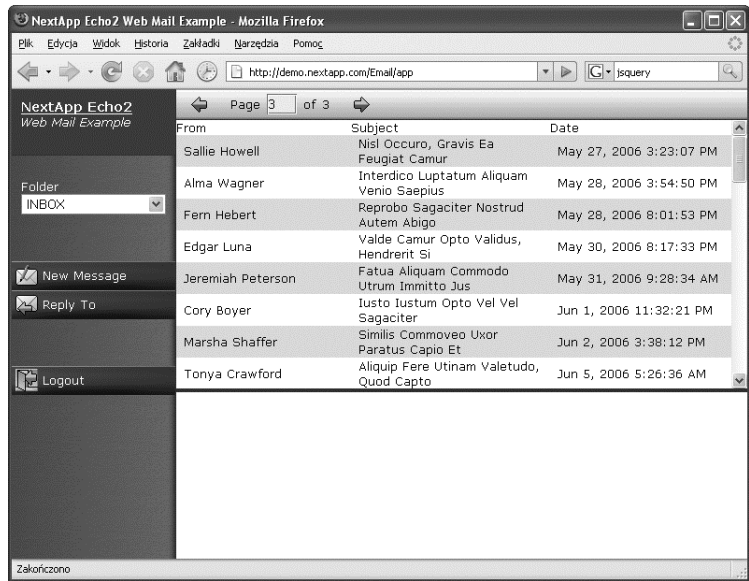


Poczta elektroniczna oparta na Ajaksie

Innym powszechnym zastosowaniem Ajaksa jest poczta elektroniczna oparta na przeglądarce, której przykład można zobaczyć na stronie <http://demo.nextapp.com/Email/app> pokazanej na rysunku 1.39.

Rysunek 1.39.

Poczta elektroniczna obsługiwana w przeglądarce



Dotychczas problem z pocztą elektroniczną obsługiwaną w przeglądarce był taki, że lista wysłanych e-maili znajdowała się na jednej stronie, ale za każdym razem, gdy kliknięto jakiś e-mail, było się przenoszonym na zupełnie nową stronę. Później trzeba było wracać na stronę z listą i znów przewijać ją na dół, żeby sprawdzić kolejne wiadomości.

Obsługa poczty elektronicznej oparta na Ajaksie jest inna. Na rysunku 1.39 widać listę e-maili; kliknięcie któregoś z nich powoduje automatyczne pobranie treści wiadomości i wyświetlenie jej — bez konieczności odświeżania przeglądarki — tak jak widać to na rysunku 1.40.

Innymi słowy, raz jeszcze Ajax z powodzeniem zdołał przemienić aplikację desktopową w aplikację internetową, rozwiązując problemy, przed którymi stały wcześniej aplikacje webowe.

Podsumowanie

W niniejszym rozdziale przedstawiono wprowadzenie do Ajaksa oraz przegląd współczesnych zastosowań tej technologii. Jak widać, nacisk położono na sprawienie, by aplikacje internetowe wyglądały i zachowywały się podobnie jak aplikacje desktopowe — Ajax jest w tym niezastąpiony. Dużą część takiego działania ma coś wspólnego z wykorzystywaniem JavaScriptu w przeglądarce, dlatego kwestii tej poświęcony zostanie kolejny rozdział, którego celem jest przybliżenie JavaScriptu — podstawy Ajaksa.

Rysunek 1.40.
Wykorzystywanie
poczty
elektronicznej
przy użyciu Ajaksa

The screenshot shows a web browser window titled "NextApp Echo2 Web Mail Example - Mozilla Firefox". The address bar displays "http://demo.nextapp.com/Email/app". The page content is divided into several sections:

- Navigation:** "Page 3 of 3" with navigation arrows.
- Folder:** A dropdown menu showing "INBOX".
- Actions:** "New Message", "Reply To", and "Logout" buttons.
- Email List:** A table with columns "From", "Subject", and "Date".
- Email Details:** A form showing "To:", "Cc:", "Bcc:", and "Subject:" fields with their respective values.
- Body:** A block of Latin text starting with "Verto feugiat neque brevitatis ad, mactio duis, quidem tamen, facilisi neque, praemitto. Olim nimis suscipere ullamcorper, utinam ut odio,".
- Footer:** "Zakończono" (Finished).

From	Subject	Date
Sallie Howell	Nisi Occuro, Gravis Ea Feugiat Camur	May 27, 2006 3:23:07 PM
Alma Wagner	Interdico Luptatum Aliquam Venio Saepius	May 28, 2006 3:54:50 PM
Fern Hebert	Reprobo Sagaciter Nostrud Autem Abigo	May 28, 2006 8:01:53 PM
Edgar Luna	Valde Camur Opto Validus, Hendrerit Si	May 30, 2006 8:17:33 PM
Jeremiah Peterson	Fatua Aliquam Commodo Utrum Immitto Jus	May 31, 2006 9:28:34 AM
Cory Boyer	Iusto Iustum Opto Vel Vel Sagaciter	Jun 1, 2006 11:32:21 PM
Marsha Shaffer	Similis Commoveo Uxor Paratus Capio Et	Jun 2, 2006 3:38:12 PM
Tonya Crawford	Aliquip Fere Utinam Valetudo, Quod Capto	Jun 5, 2006 5:26:36 AM

To: Martha Williams <Martha.Williams@nextapp.com>, Roman Zamora <Roman.Zamora@nextapp.com>

Cc: Terrell Hill <Terrell.Hill@nextapp.com>, Kayla York <Kayla.York@nextapp.com>, Lorene Carpenter <Lorene.Carpenter@nextapp.com>

Bcc: Joe Smith <joe.smith@nextapp.com>

Subject: Nisi Occuro, Gravis Ea Feugiat Camur

Verto feugiat neque brevitatis ad, mactio duis, quidem tamen, facilisi neque, praemitto. Olim nimis suscipere ullamcorper, utinam ut odio, mactio duis, quidem tamen, facilisi neque, praemitto.

Okenka wyskakujące oparte na Ajaksie

Internetowe aplikacje z mapami nie są jedynymi, które wykorzystują wyskakujące okienka oparte na Ajaksie — okienka typu pop-up można znaleźć wszędzie. Można na przykład spojrzeć na Netflix Top 100 — listę stu najlepszych filmów, dostępną pod adresem www.netflix.com/Top100 i zaprezentowaną na poniższym rysunku.



Pobieranie informacji o 100 najlepszych filmach według Netflix

Żeby otrzymać więcej informacji o filmie, wystarczy pozostawić kursor myszy nad nazwą filmu, a aplikacja automatycznie pobierze dane z serwera za pomocą Ajaksa i wyświetli je w wyskakującym okienku, jak widać to na rysunku. Właśnie tak działa Ajax — odświeżanie strony nie jest konieczne, nie trzeba klikać przycisku *Wstecz* w przeglądarce, żeby powrócić do początkowej strony.