

## 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 w mgnieniu oka

Autor: Phil Ballard

Tłumaczenie: Jacek Lempart, Magdalena Faszczewska

ISBN: 83-246-0662-9

Tytuł oryginału: [Teach Yourself AJAX in 10 Minutes](#)

Format: A5, stron: 256



### Opanuj technologię, która zrewolucjonizowała sposób obsługi aplikacji internetowych

- Poznaj podstawy tworzenia witryn WWW
- Stwórz aplikacje w technologii AJAX
- Wykorzystaj dane w formacie XML

AJAX, będący połączeniem JavaScriptu, języka XML i arkuszy stylów, to jedna z technologii, które wprowadziły ostatnio sporo zamieszania w środowisku projektantów witryn i aplikacji WWW. Jej zastosowanie zdecydowanie upraszcza i przyspiesza korzystanie z programów dostępnych przez przeglądarkę WWW. AJAX sprawia, że zawartość strony nie musi być przeładowywana po każdej akcji użytkownika. Witryna WWW przygotowana z wykorzystaniem technologii AJAX przypomina tradycyjną aplikację, komunikacja z serwerem nie przeszkadza w korzystaniu z niej, a dynamiczne zmiany elementów strony przebiegają znacznie szybciej.

Książka „AJAX w mgnieniu oka” to wprowadzenie do tematyki projektowania witryn i aplikacji WWW w technologii AJAX. Czytając ją, opanujesz podstawy stosowania AJAX-a we własnych projektach. Poznasz technologie przetwarzania danych ze stron WWW po stronie przeglądarki i serwera oraz dowiesz się, czym jest obiektowy model dokumentu. Nauczysz się korzystania z obiektów JavaScriptu, języka XML i protokołu SOAP. Stworzysz własne aplikacje w technologii AJAX i dowiesz się, jak unikać najczęściej popełnianych błędów.

- Podstawy języka HTML
- Zasada działania protokołu HTTP
- JavaScript i PHP
- Obiektowy model dokumentu (DOM)
- Struktura aplikacji AJAX
- Obiekty JavaScriptu
- Zwracanie danych w postaci tekstu i plików XML
- Technologia AJAX
- Usługi sieciowe i protokół SOAP
- Korzystanie z biblioteki Rico

**Zacznij tworzyć nowoczesne witryny**



# Spis treści

<b>0 autorze .....</b>	<b>9</b>
<b>Wprowadzenie .....</b>	<b>11</b>
O książce .....	11
Czym jest Ajax? .....	12
Dla kogo przeznaczona jest ta książka .....	13
Czego potrzebuję, aby skorzystać z tej książki? .....	13
Konwencje użyte w książce .....	14
<b>Część I Powtórka z technologii internetowych .....</b>	<b>15</b>
<b>Rozdział 1. Anatomia witryny internetowej .....</b>	<b>17</b>
Krótką historią sieci Internet .....	17
Działanie sieci Internet .....	18
Podsumowanie .....	24
<b>Rozdział 2. Pisanie stron internetowych w języku HTML .....</b>	<b>25</b>
Wprowadzenie do HTML .....	25
Elementy strony HTML .....	28
Bardziej zaawansowana strona HTML .....	33
Niektóre użyteczne znaczniki HTML .....	35
Kaskadowe Arkusze Stylów (CSS) w dwie minuty .....	36
Podsumowanie .....	37
<b>Rozdział 3. Wysyłanie żądań przy użyciu protokołu HTTP .....</b>	<b>39</b>
Wprowadzenie do HTTP .....	39
Żądanie i odpowiedź HTTP .....	40
Formularze HTML .....	43
Podsumowanie .....	46

<b>Rozdział 4. Programowanie po stronie klienta przy użyciu JavaScript ..</b>	<b>47</b>
O języku JavaScript .....	47
Na samym dnie .....	49
Manipulacja danymi w JavaScript .....	60
Podsumowanie .....	63
<b>Rozdział 5. Programowanie po stronie serwera przy użyciu PHP .....</b>	<b>65</b>
Wprowadzenie do PHP .....	65
Osadzanie PHP w stronach HTML .....	66
Zmienne w PHP .....	68
Kontrola przebiegu programu .....	70
Podsumowanie .....	72
<b>Rozdział 6. Krótkie wprowadzenie do XML .....</b>	<b>73</b>
Wprowadzenie do XML .....	73
Podstawy XML .....	74
JavaScript i XML .....	78
Obiektowy model dokumentu (DOM) .....	79
Podsumowanie .....	81
<b>Część II Wprowadzenie do technologii Ajax .....</b>	<b>83</b>
<b>Rozdział 7. Anatomia aplikacji Ajax .....</b>	<b>85</b>
Technika Ajax jest potrzebna .....	85
Wprowadzenie do Ajax .....	88
Części składowe Ajax .....	90
Złożenie wszystkiego w jedną całość .....	93
Podsumowanie .....	94
<b>Rozdział 8. Obiekt XMLHttpRequest .....</b>	<b>95</b>
Więcej o obiektach JavaScript .....	95
Wprowadzenie do XMLHttpRequest .....	97
Tworzenie obiektu XMLHttpRequest .....	98
Podsumowanie .....	105
<b>Rozdział 9. Komunikacja z serwerem .....</b>	<b>107</b>
Wysyłanie żądań do serwera .....	107
Monitorowanie stanu serwera .....	112
Funkcja zwrotna .....	114
Podsumowanie .....	116
<b>Rozdział 10. Używanie zwróconych danych .....</b>	<b>117</b>
Właściwości responseText oraz responseXML .....	117
Kolejna przydatna właściwość JavaScript DOM .....	121
Parsowanie responseXML .....	122

Zapewnienie obsługi użytkownika .....	123
Podsumowanie .....	125
<b>Rozdział 11. Nasza pierwsza aplikacja Ajax .....</b>	<b>127</b>
Budowanie aplikacji Ajax .....	127
Dokument HTML .....	128
Dodanie kodu JavaScript .....	129
Złożenie wszystkiego w jedną całość .....	133
Podsumowanie .....	136
<b>Część III Bardziej zaawansowane technologie Ajax ....</b>	<b>137</b>
<b>Rozdział 12. Zwracanie danych w postaci tekstu .....</b>	<b>139</b>
Więcej korzyści z właściwości responseText .....	139
Podsumowanie .....	146
<b>Rozdział 13. AHAH — Asynchroniczny HTML i HTTP .....</b>	<b>147</b>
Wprowadzenie do AHAH .....	147
Utworzenie małej biblioteki dla AHAH .....	149
Wykorzystanie myAHAHlib.js .....	151
Podsumowanie .....	156
<b>Rozdział 14. Zwracanie danych w postaci XML .....</b>	<b>157</b>
„x” w Ajax .....	157
Własność responseXML .....	158
Projekt — czytnik nagłówek RSS .....	162
Podsumowanie .....	170
<b>Rozdział 15. Usługi sieciowe i protokół REST .....</b>	<b>171</b>
Wprowadzenie do usług sieciowych .....	171
Protokół REST .....	172
Wykorzystanie protokołu REST w praktyce .....	174
REST i AJAX .....	178
Podsumowanie .....	179
<b>Rozdział 16. Usługi sieciowe korzystające z protokołu SOAP .....</b>	<b>181</b>
Poznajemy protokół SOAP (ang. Simple Object Access Protocol) .....	181
Protokół SOAP .....	183
Wykorzystywanie modelu Ajax i protokołu SOAP .....	186
Ponowne spojrzenie na protokoły SOAP i REST .....	187
Podsumowanie .....	188

<b>Rozdział 17. Biblioteka języka JavaScript dla modelu Ajax .....</b>	<b>189</b>
Biblioteka modelu Ajax .....	189
Przegląd myAHAHlib.js .....	190
Implementacja naszej biblioteki .....	191
Korzystanie z biblioteki .....	195
Poszerzanie biblioteki .....	198
Podsumowanie .....	198
<b>Rozdział 18. Pułapki Ajax .....</b>	<b>199</b>
Typowe błędy Ajax .....	199
Przycisk Wstecz .....	199
Ulubione i hiperłącza .....	200
Informowanie użytkownika o zdarzeniu .....	201
Radzenie sobie z brakiem obsługi Ajax .....	201
Roboty indeksujące w wyszukiwarkach .....	202
Aktywne elementy strony .....	203
Nie używaj technologii Ajax, gdy jest to nieodpowiednie .....	204
Bezpieczeństwo .....	204
Testowy kod na różnych platformach .....	205
Ajax nie naprawi złego projektu .....	205
Niektóre programistyczne pułapki .....	206
Podsumowanie .....	207
<b>Część IV Komercyjne oraz otwarte zasoby technologii Ajax .....</b>	<b>209</b>
<b>Rozdział 19. Zestaw narzędzi prototype.js .....</b>	<b>211</b>
Wprowadzenie do prototype.js .....	211
Obudowywanie obiektu XMLHttpRequest .....	214
Przykładowy projekt — czytnik cen akcji .....	217
Podsumowanie .....	219
<b>Rozdział 20. Zastosowanie Rico .....</b>	<b>221</b>
Wstęp do Rico .....	221
Inne narzędzia Rico do tworzenia interfejsu .....	226
Podsumowanie .....	231
<b>Rozdział 21. Wykorzystanie XOAD .....</b>	<b>233</b>
Wprowadzenie do XOAD .....	233
XOAD HTML .....	237
Zaawansowane programowanie w XOAD .....	239
Podsumowanie .....	240
<b>Skorowidz .....</b>	<b>241</b>

## Rozdział 7.

# **Anatomia aplikacji Ajax**

W tym rozdziale zawarte są informacje o budowaniu poszczególnych bloków Ajax oraz o tym, jak one ze sobą współpracują tworząc architekturę aplikacji Ajax. W dalszych rozdziałach części II, „Wprowadzenie do Ajax”, zostaną dokładniej przeanalizowane te komponenty, które ostatecznie składają się na działającą aplikację Ajax.

## **Technika Ajax jest potrzebna**

W części I, „Przypomnienie technologii internetowych”, dokonaliśmy przeglądu głównych technologii, które umożliwiają budowę komponentów aplikacji Ajax. Posiadasz zatem przynajmniej podstawową wiedzę na temat JavaScript, PHP oraz XML, czyli wszystkiego, z czego będziemy korzystać w części II.

Zanim przystąpimy do omawiania poszczególnych komponentów, przyjrzyjmy się dokładniej temu, czego oczekujemy od naszej aplikacji Ajax.

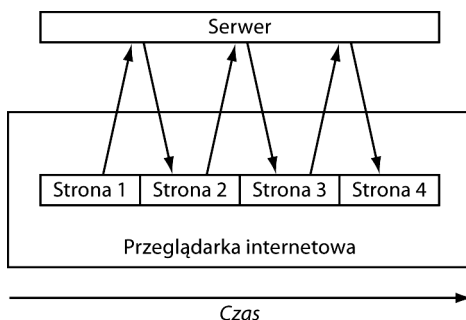
## Interakcje tradycyjne kontra Ajax klient-serwer

W rozdziale 1., „Anatomia witryny internetowej”, został omówiony tradycyjny model interfejsu witryny internetowej. Kiedy pracujesz z taką witryną, poszczególne strony zawierające tekst, obrazki, formularze do wprowadzania danych itd. prezentowane są raz w danej chwili. Każda strona musi zostać obsłużona z osobna przed przejściem do kolejnej.

Przykładowo, możesz wypełnić formularz z danymi, wprowadzając je i poprawiając tyle razy, ile zechcesz. Jednocześnie wiesz, że dane nie powędrują do serwera, dopóki formularz nie zostanie ostatecznie zatwierdzony.

Rysunek 7.1 ilustruje ten rodzaj interakcji.

**Rysunek 7.1.**  
*Tradycyjne interakcje  
klient-serwer*



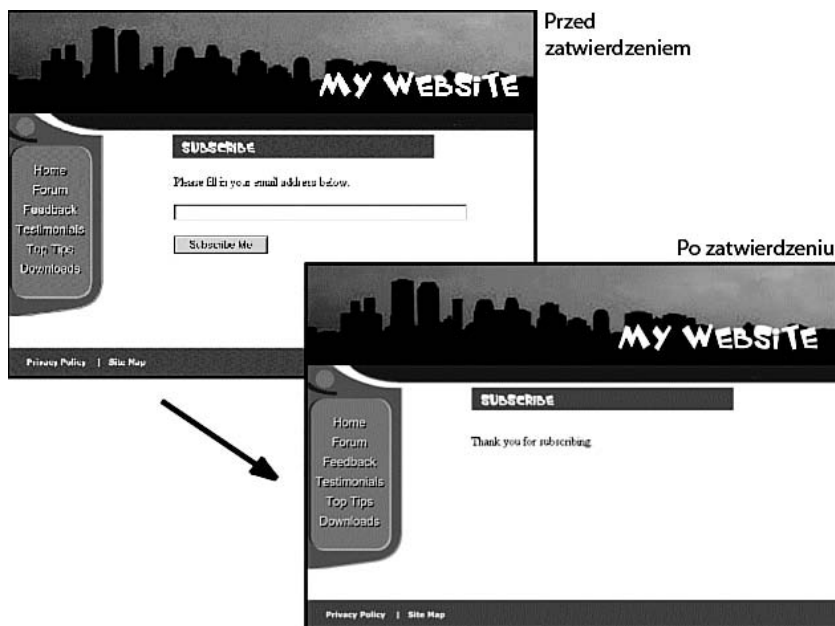
Po zatwierdzeniu formularza lub kliknięciu łącza nawigacyjnego musisz czekać, dopóki ekran przeglądarki jest odświeżany, by wyświetlić nową lub skorygowaną stronę, która została wygenerowana przez serwer.

W miarę jak Twoje doświadczenie jako użytkownika Internetu się wzbogaca, korzystanie z takiego interfejsu staje się niemal drugą naturą. Uczysz się pewnych praktycznych zachowań, które pozwalają unikać kłopotów, jak na przykład „nie naciskaj ponownie przycisku zatwierdzającego formularz” czy „nie naciskaj przycisku *Wstecz* po zatwierdzeniu formularza”.

Niestety, interfejsy użytkownika zbudowane w oparciu o ten model posiadają kilka wad. Pierwsza to taka, że istnieje znaczące opóźnienie podczas czytania nowej lub skorygowanej strony. Następuje przerwanie tego, co użytkownicy postrzegają jako tak zwany **przebieg** aplikacji.

Co więcej, **cała** strona musi zostać wczytana przy każdej okazji, nawet jeśli większość jej zawartości jest identyczna z poprzednią stroną. Elementy wspólne dla wielu stron danej witryny internetowej, takie jak nagłówki, stopki czy sekcje nawigacyjne, mogą stanowić znaczną proporcję danych zawartych w stronie.

Rysunek 7.2 ilustruje witrynę wyświetlającą strony przed zatwierdzeniem formularza i po nim, ukazując, jak wiele identycznej zawartości zostało wczytane ponownie oraz jak relatywnie mało zmieniła się zawartość strony.



**Rysunek 7.2.** *Wiele elementów strony jest niepotrzebnie ponownie wczytywana*

To niepotrzebne pobieranie danych marnuje pasmo i zwiększa opóźnienie we wczytywaniu każdej nowej strony.



**Pasmo** odnosi się do pojemności kanałów komunikacyjnych przesyłających informację. W Internecie pasmo jest zwykle mierzone w *bps* (ang. *bits per second* — *bity na sekundę*) lub wyższych mnożnikach, takich jak *Mbps* (ang. *million bits per second* — *million bitów na sekundę*).



## Bogate doznania użytkownika

Połączenie skutków problemów opisanych powyżej powoduje, że użytkownik posiada dużo gorszy interfejs użytkownika w sensie jego interaktywności w porównaniu do interfejsów oferowanych przez ogromną większość aplikacji biurkowych.

W aplikacjach stacjonarnych oczekujesz, że wyświetlana zawartość programu pozostaje widoczna, a elementy interfejsu reagują na wydawane polecenia, podczas gdy proces przetwarzania wykonuje się po cichu w tle. Gdy na przykład piszę ten rozdział wykorzystując procesor tekstu, mogę zapisać dokument na dysku, przewinąć stronę w górę lub dół, zmienić krój czcionki i jej rozmiar bez potrzeby czekania przy każdej okazji, aż cały ekran zostanie odświeżony.

Ajax pozwala dodać do interfejsu Twojej aplikacji internetowej funkcjonalność, która dotychczas bardziej powszechnie spotykana była w aplikacjach biurkowych. Wiąże się to często z określeniem **bogatego doznania użytkownika** (ang. *rich user experience*).

## Wprowadzenie do Ajax

By wzbogacić doznania użytkownika, musisz dodać pewne możliwości do tradycyjnego podejścia do interfejsu aplikacji internetowych. Chcesz, aby strona Twojego użytkownika była interaktywna, odpowiadała na akcje użytkownika zmieniając zawartość i by była aktualizowana bez jakichkolwiek przerw na wczytywanie strony lub odświeżanie ekranu.

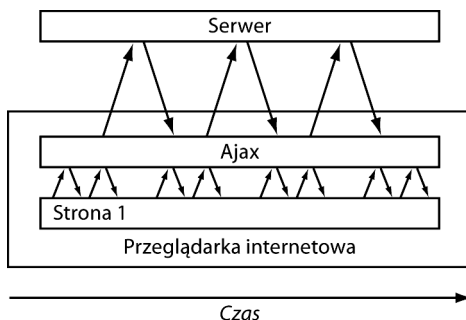
Aby to osiągnąć, Ajax buduje dodatkową warstwę przetwarzania, pomiędzy stroną internetową a serwerem.

Warstwa ta często określana jest mianem **Modułu Ajax** lub **Osnowy Ajax**. Przechwytuje żądania od użytkownika, a w tle obsługuje komunikację z serwerem — po cichu, dyskretnie i **asynchronicznie**. Rozumiemy przez to, że żądania i odpowiedzi serwera nie muszą być zbieżne z określoną akcją użytkownika, ale mogą mieć miejsce w dowolnym, wygodnym dla użytkownika czasie, by poprawić działanie aplikacji. Przeglądarka internetowa nie zamraża się teraz i nie oczekuje ukończenia przez serwer ostatniego żądania. Zamiast tego, pozwala użytkownikowi na przewijanie ekranu strony, klikanie czy pisanie na bieżącej stronie.

Aktualizowanie elementów strony, tak by odpowiadały skorygowanym danym otrzymanym z serwera, jest również wykonywane przez Ajax. Dzieje się to dynamicznie podczas ciągłego korzystania ze strony.

Rysunek 7.3 prezentuje, jak wyglądają te interakcje.

**Rysunek 7.3.**  
*Interakcje Ajax  
klient-serwer*



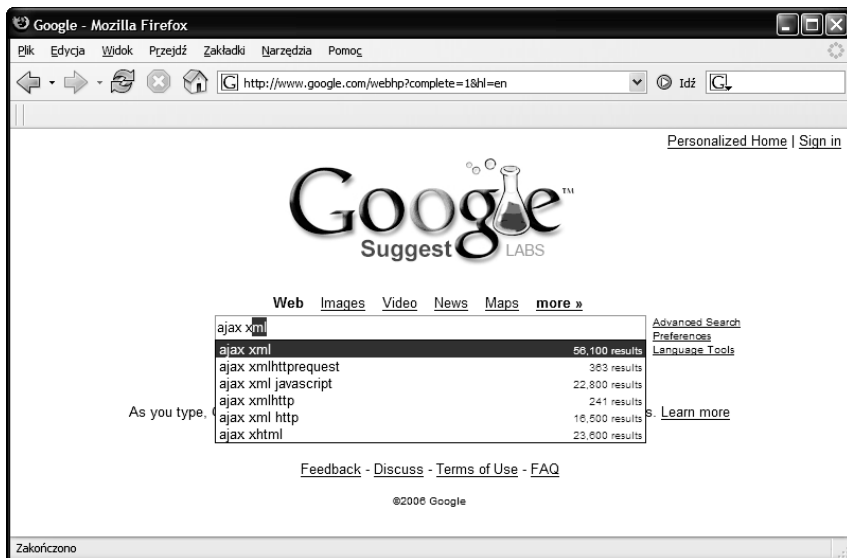
## Prawdziwa aplikacja Ajax — Google Suggest

Aby zobaczyć przykład aplikacji Ajax w akcji, spójrzmy na *Google Suggest*. Aplikacja ta rozszerza interfejs znanej wyszukiwarki Google, oferując użytkownikowi sugestie dla odpowiednich terminów, w oparciu o to, co zostało w danym momencie wpisane.

Za każdym naciśnięciem klawisza przez użytkownika warstwa Ajax aplikacji odpytuje serwer Google, czy istnieje odpowiednio podobna fraza; jeśli tak, to wyświetla zwrócone dane w liście rozwijanej. Wraz z każdą sugerowaną frazą listowana jest również liczba rezultatów, których spodziewałaby się wyszukiwarka dla poszukiwanego przeprowadzonego dla tej frazy. W każdym momencie użytkownik ma możliwość wyboru jednej z tych sugestii zamiast kontynuowania pisania, a wyszukiwarka Google wykona przeszukiwanie dla wybranej frazy.

Ponieważ serwer jest odpytywany przy każdym naciśnięciu klawisza, lista rozwijana jest dynamicznie aktualizowana, bez zbędnego czekania na odświeżenie strony lub innych podobnych przerw.

Rysunek 7.4 pokazuje działający program. Możesz sam wypróbować tę aplikację, korzystając z hiperłącza strony domowej Google <http://www.google.com/webhp?complete=1&hl=en>.



**Rysunek 7.4.** Przykład aplikacji Ajax — Google Suggest

A teraz zidentyfikujemy poszczególne komponenty takiej aplikacji Ajax oraz zobaczymy, jak one współpracują.



Google zaprezentował również inne aplikacje Ajax, które możesz wypróbować, jak na przykład witrynę internetową *gmail*, będącą serwisem poczty elektronicznej. Innym przykładem może też być *Google Maps*. Po szczegóły sięgnij pod adres: <http://www.google.com>.

## Części składowe Ajax

Teraz przeanalizujemy poszczególne komponenty aplikacji Ajax.

### Obiekt XMLHttpRequest

Kiedy klikasz łącze lub zatwierdzasz formularz HTML, wysyłasz żądanie HTTP do serwera, który odpowiada wysyłając do Ciebie nową lub zmodyfikowaną stronę. Aby jednak Twoja aplikacja internetowa mogła pracować asynchronicznie,

musisz posiadać możliwość wysłania żądania HTTP do serwera bez powiązania go z żądaniem wyświetlenia nowej strony.

Możesz to osiągnąć dzięki **obiekowi** XMLHttpRequest. Ten obiekt JavaScript potrafi utworzyć połączenie z serwerem, a następnie wysłać do niego żądania HTTP bez potrzeby powiązania ich z wczytywaniem strony.

W następnych rozdziałach dowiesz się, jak utworzyć instancję tego obiektu oraz jak jego właściwości i metody mogą być wykorzystane przez procedury JavaScript, zawarte w stronie internetowej, aby ustanowić asynchroniczne połączenie z serwerem.



Ze względów bezpieczeństwa obiekt XMLHttpRequest może generalnie łączyć się z adresami URL tej samej domeny, co wywołana strona, oraz nie może bezpośrednio łączyć się ze zdalnym serwerem.

Rozdział 8., „Obiekt XMLHttpRequest”, omawia, jak utworzyć instancję obiektu XMLHttpRequest oraz opisuje pokrótce właściwości i metody obiektu.

## Rozmawianie z serwerem

W tradycyjnym modelu strony internetowej, kiedy wysyłasz żądanie poprzez hiperłącze lub zatwierdzenie strony, serwer przyjmuje to żądanie, wykonuje wszelkie wymagane czynności po stronie serwerowej, by wreszcie wysłać do Ciebie nową stronę z zawartością odpowiadającą akcji, jaką wykonałeś.

Gdy to przetwarzanie ma miejsce, interfejs użytkownika jest skutecznie zamrożony. Zapewne wiesz już, kiedy serwer kończy swoje zadanie — kiedy w przeglądarce pojawia się nowa lub zmodyfikowana strona.

Jednakże w przypadku asynchronicznych żądań serwera komunikacja odbywa się w tle, a ich ukończenie niekoniecznie musi być powiązane z odświeżaniem strony czy wczytywaniem nowej. Dlatego też musisz ustalić, w jakim stopniu serwer obsłużył dane żądanie.

Obiekt XMLHttpRequest posiada wygodną właściwość, która raportuje postęp obsługi żądania serwera. Możesz sprawdzić tę właściwość, używając procedur JavaScript celem określenia momentu, w którym serwer zakończy pracę, a wyniki jego działania są dostępne do wykorzystania.

Z tego powodu Twój kod Ajax musi zawierać procedurę monitorującą status żądań serwera i odpowiednio się zachowywać. Przyjrzymy się temu zagadnieniu bliżej w rozdziale 9., „Rozmawianie z serwerem”.

## Co dzieje się na serwerze?

Dotychczas, gdy interesowaliśmy się skryptami serwerowymi, komunikacja za pomocą obiektu `XMLHttpRequest` była po prostu innym żądaniem HTTP. Aplikacje Ajax niewiele „przejmują się” tym, jakie języki programowania lub środowisko jest obecne na serwerze. Przyjmując, że strona kliencka warstwy Ajax otrzymuje na czas poprawnie sformatowane odpowiedzi HTTP z serwera, wszystko będzie pracować poprawnie.

Możliwe jest napisanie prostej aplikacji Ajax w całości bez skryptów serwerowych, po prostu wywołując obiekt `XMLHttpRequest` ze statycznym zasobem serwera, jak na przykład dokumentem XML lub plikiem tekstowym.

Aplikacje Ajax mogą wywoływać rozmaite inne serwerowe zasoby, takie jak na przykład **usługi sieciowe** (*ang. web services*). W dalszej części niniejszej książki zapoznamy się z pewnymi przykładami wywoływania usług sieciowych, korzystając z takich protokołów jak SOAP czy REST.



W niniejszej książce będziemy korzystać z popularnego serwerowego języka skryptowego PHP w procedurach serwerowych. Jeśli jednak czujesz się bardziej komfortowo używając takich języków jak ASP, JSP czy innych, nic nie stoi na przeszkodzie, abyś z nich korzystał.

## Obsługa odpowiedzi serwera

Kiedy jesteś informowany o tym, że asynchroniczne żądanie zostało z powodzeniem wykonane, możesz wykorzystać informacje zwrócone przez serwer.

Ajax pozwala, aby zwracane dane mogły mieć różną postać, w tym na przykład tekst ASCII oraz dane XML.

W zależności od natury aplikacji, informacje te mogą zostać następnie skonwertowane do innej postaci, wyświetlone lub przetworzone w pewien założony sposób w obrębie bieżącej strony.

Skupimy się na tych kwestiach w rozdziale 10., „Używanie zwróconych danych”.

## Inne zadania zarządzające

Od aplikacji Ajax wymagane będzie również wykonanie wielu innych zadań. Przykładem może być wykrywanie błędów oraz ich odpowiednia obsługa, czy też informowanie użytkownika o stanie przesłanych żądań Ajax.

W kolejnych rozdziałach tej książki zostanie zaprezentowanych wiele przykładów.

## Złożenie wszystkiego w jedną całość

Przypuśćmy, że chcesz zaprojektować nową aplikację Ajax lub zaktualizować odziedziczoną aplikację internetową, tak by obsługiwała techniki Ajax. Jak się możesz z tym uporać?

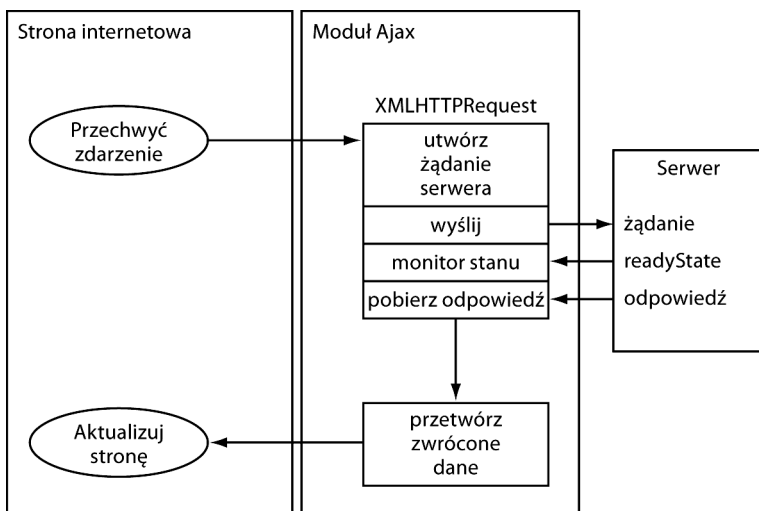
Po pierwsze musisz zdecydować, jakie zdarzenia strony i akcje użytkownika będą odpowiedzialne za wysyłanie asynchronicznych żądań HTTP. Możesz przykładowo podjąć decyzję, że zdarzenie `onMouseOver` wybranego obrazka będzie skutkowało wysłaniem żądania do serwera i pobraniem informacji na temat tegoż obrazu. Innym przykładem może być zdarzenie `onClick` przycisku, które generować będzie żądanie serwera pobierające informacje na temat tego, jak wypełnić pola formularza.

W rozdziale 4., „Programowanie po stronie klienta przy użyciu JavaScript”, przekonałeś się, jak JavaScript może być wykorzystany do wykonania instrukcji połączonych z takimi zdarzeniami, poprzez wykorzystanie procedur obsługi zdarzeń. W swojej aplikacji Ajax metody takie odpowiedzialne będą za zapoczątkowanie asynchronicznych żądań HTTP poprzez wykorzystanie obiektu `XMLHttpRequest`.

Mając gotowe żądania, musisz napisać procedury monitorujące postęp tych żądań, reagujące na otrzymanie sygnału z serwera, że żądanie zostało pomyślnie zakończone.

W końcu, po otrzymaniu powiadomienia, że serwer zakończył swoje zadania, potrzebujesz procedury pobierającej zwrócone informacje z serwera, aby zastosować je w aplikacji. Możesz przykładowo zechcieć wykorzystać nowo zwrócone dane do zmiany zawartości tekstu strony, wypełnienia pól formularza lub otwarcia wyskakującego okienka z informacją.

Rysunek 7.5 prezentuje diagram przepływu dla tego procesu.



readyState – stan oznaczający wykonanie żądania

**Rysunek 7.5.** Diagram prezentujący współpracę komponentów aplikacji Ajax

W rozdziale 11., „Nasza pierwsza aplikacja Ajax”, wykorzystamy świeżo nabytą wiedzę do zbudowania kompletnej aplikacji Ajax.

## Podsumowanie

W rozdziale tym opisaliśmy ułomności tradycyjnego interfejsu internetowego, wskazując poszczególne problemy, które chcemy rozwiązać. Zaprezentowaliśmy różne bloki budulcowe aplikacji Ajax oraz omówiliśmy, jak one współpracują.

W następnych rozdziałach części II przyjrzymy się tym komponentom dokładniej, wykorzystując je do zbudowania kompletnej aplikacji Ajax.