

Przedmowa (7)

O autorze (9)

Wstęp (11)

Rozdział 1. Podstawy technologii Ajax (17)

- 1. Określenie zgodności przeglądarki internetowej za pomocą obiektu żądania (21)
- 2. Użycie obiektu żądania do przekazania danych POST do serwera (24)
- 3. Użycie własnej biblioteki z XMLHttpRequest (29)
- 4. Otrzymywanie danych w postaci XML (32)
- 5. Pobieranie zwykłych starych ciągów tekstowych (37)
- 6. Otrzymywanie danych w postaci liczb (40)
- 7. Otrzymywanie danych w formacie JSON (45)
- 8. Obsługa błędów obiektu żądania (51)
- 9. Zagłębienie się w odpowiedź HTTP (56)
- 10. Generowanie stylizowanej wiadomości wykorzystującej plik arkusza stylów (61)
- 11. Generowanie wiadomości stylizowanej "w locie" (66)

Rozdział 2. Formularze sieciowe (71)

- 12. Wysyłanie do serwera wartości pól tekstowych lub elementów textarea bez odświeżania przeglądarki (71)
- 13. Wyświetlanie wartości pola tekstowego lub elementu textarea za pomocą danych serwera (78)
- 14. Wysyłanie do serwera wybranych wartości z listy bez korzystania z komunikacji dwustronnej (84)
- 15. Dynamiczne generowanie nowej listy wyborów za pomocą danych serwera (91)
- 16. Rozbudowa istniejącej listy wyboru (98)
- 17. Wysyłanie do serwera wartości pól wyboru bez konieczności korzystania z komunikacji dwustronnej (103)
- 18. Dynamiczne generowanie nowej grupy pól wyboru na podstawie danych z serwera (111)
- 19. Zapełnienie istniejącej grupy pól wyboru danymi z serwera (117)
- 20. Zmiana nieuporządkowanych list za pomocą odpowiedzi HTTP (124)
- 21. Wysyłanie do komponentu serwera wartości ukrytych znaczników (131)

Rozdział 3. Sprawdzanie poprawności (137)

- 22. Sprawdzanie poprawności pól tekstowych i elementów textarea pod kątem wystąpienia pustych pól (137)
- 23. Sprawdzanie poprawności składni adresu e-mail (139)
- 24. Sprawdzanie poprawności unikalnych nazw użytkowników (149)
- 25. Sprawdzanie poprawności numeru karty kredytowej (152)
- 26. Sprawdzanie poprawności kodu bezpieczeństwa karty kredytowej (160)
- 27. Sprawdzanie poprawności kodu pocztowego (164)

Rozdział 4. Super sposoby dla programistów sieciowych (169)

- 28. Uzyskanie dostępu do API Google Maps (169)
- 29. Użycie obiektu żądania API Google Maps (171)
- 30. Użycie Ajaksu z Google Maps i Yahoo! Maps (177)

- 31. Wyświetlanie danych XML pobranych z witryny Weather.com (186)
- 32. Użycie Ajaksu z Yahoo! Maps i GeoURL (193)
- 33. Debugowanie w przeglądarce Firefox znaczników wygenerowanych przez Ajax (197)
- 34. Pobranie kodu pocztowego (200)
- 35. Tworzenie dużych, łatwych w obsłudze zakładek (207)
- 36. Używanie trwałego magazynu danych po stronie klienta dla aplikacji Ajax (208)
- 37. Sterowanie historią przeglądarki internetowej za pomocą iframes (211)
- 38. Wysyłanie wartości cookie do programu serwera (214)
- 39. Użycie XMLHttpRequest do wydobywania cen energii ze strony internetowej (221)
- 40. Wysyłanie wiadomości e-mail za pomocą obiektu XMLHttpRequest (226)
- 41. Odszukanie informacji lokalizacyjnych przeglądarki (232)
- 42. Tworzenie czytnika kanałów RSS (235)

Rozdział 5. Direct Web Remoting (DWR) dla zapaleńców Javy (243)

- 43. Integracja DWR z aplikacją sieciową Javy (243)
- 44. Użycie DWR do wypełnienia listy wyboru wartościami z tablicy Javy (246)
- 45. Użycie DWR do utworzenia listy select na podstawie wartości z obiektu Map Javy (251)
- 46. Wyświetlanie na stronie internetowej kluczy i wartości z obiektu HashMap Javy (253)
- 47. Użycie DWR do wypełnienia listy uporządkowanej wartościami z tablicy Javy (256)
- 48. Dostęp do własnego obiektu Javy z poziomu JavaScript (260)
- 49. Wywołanie wbudowanego obiektu Javy z poziomu JavaScriptu za pomocą DWR (265)

Rozdział 6. Sposoby na Ajax z bibliotekami Prototype i Rico (269)

- 50. Użycie narzędzi Ajax biblioteki Prototype we własnych aplikacjach (269)
- 51. Uaktualnienie zawartości elementu HTML danymi pochodzącymi z serwera (274)
- 52. Tworzenie obserwatorów pól strony internetowej (278)
- 53. Użycie biblioteki Rico do uaktualnienia kilku elementów za pomocą jednej odpowiedzi Ajax (282)
- 54. Utworzenie księgarni typu "przeciągnij i upuść" (286)

Rozdział 7. Praca z Ajaksem wraz z Ruby on Rails (293)

- 55. Instalacja Ruby on Rails (294)
- 56. Monitorowanie zdalnych wywołań za pomocą platformy Rails (299)
- 57. Udostępnienie kodu JavaScript aplikacjom platformy Rails (305)
- 58. Dynamiczne generowanie listy select w szablonie platformy Rails (307)
- 59. Określenie, czy technologia Ajax jest wywoływana w żądaniu (311)
- 60. Dynamiczne generowanie listy select za pomocą danych pochodzących z bazy danych (312)
- 61. Okresowe przeprowadzanie zdalnych wywołań (316)
- 62. Dynamiczne przeglądanie informacji o żądaniu dla obiektu XMLHttpRequest (320)

Rozdział 8. Urok biblioteki JavaScript script.aculo.us (325)

- 63. Integracja efektów wizualnych biblioteki script.aculo.us z aplikacją Ajax (325)
- 64. Tworzenie okna logowania, które "wzrusza się", gdy zostaną podane nieprawidłowe dane (328)
- 65. Utworzenie autouzupełniającego się pola za pomocą biblioteki script.aculo.us (332)
- 66. Tworzenie pola edycji tekstu (336)
- 67. Utworzenie formularza sieciowego, który znika po wysłaniu (339)

Rozdział 9. Opcje i wydajność (341)

- 68. Naprawa przycisku Wstecz przeglądarki internetowej w aplikacjach Ajax (342)
- 69. Obsługa zakładek i przycisków Wstecz za pomocą RSH (349)
- 70. Ustawienie ograniczenia czasu dla żądania HTTP (360)
- 71. Usprawnienie możliwości obsługi, wydajności i niezawodności dużych aplikacji JavaScript (363)
- 72. Zaciemnianie kodu JavaScript i Ajax (369)
- 73. Użycie dynamicznego znacznika script do przeprowadzenia żądań usług sieciowych (374)
- 74. Konfiguracja serwera Apache ze względu na kwestie związane z przejściami między różnymi domenami (379)
- 75. Uruchomienie wewnątrz przeglądarki internetowej mechanizmu wyszukiwania (381)
- 76. Użycie deklaratywnych znaczników za pomocą mechanizmu XForms zamiast znacznika script (386)
- 77. Utworzenie bufora po stronie klienta (392)
- 78. Tworzenie autouzupełniającego się pola (400)
- 79. Dynamiczne wyświetlenie większej ilości informacji na dany temat (403)
- 80. Użycie ciągów tekstowych i tablic w celu dynamicznego generowania kodu HTML (406)

Skorowidz (411)