
Spis treści

Słowo wstępne	15
1. Podstawowe pojęcia związane z kontenerem aplikacji społecznościowych	21
Czym jest kontener aplikacji społecznościowych?	22
Profil użytkownika	23
Znajomi i powiązania użytkownika	24
Strumień aktywności użytkowników	24
Implementacja zastrzeżonych i otwartych standardów	25
Implementacja zastrzeżona	25
Implementacja typu open source	26
Dlaczego w tej książce zostaną omówione otwarte standardy?	27
Wbudowana aplikacja — tworzenie rozwiązań w ramach czarnej skrzynki	27
Wbudowane zabezpieczenia aplikacji	29
Ataki XSS	30
Zasada tego samego pochodzenia i starsze przeglądarki	30
Pobieranie plików bez wiedzy użytkownika	31
Zabezpieczanie aplikacji	31
Aplikacja zewnętrzna — integracja danych serwisu społecznościowego poza kontenerem	31
Widoki aplikacji	32
Widok domowy (mały)	33
Widok profilu (mały)	34
Widok kanwy (duży)	35
Domyślny widok (dowolny)	36
Zagadnienia związane z uprawnieniami aplikacji	37
Aplikacje strony klienckiej kontra aplikacje serwera	38
Stosowanie systemów szablonów w warstwie znaczników	38
Stosowanie mieszanego środowiska serwera i klienta	39
Opóźnianie ładowania mniej ważnej treści	40
Kiedy dobra aplikacja okazuje się zła?	40
Przenośna aplikacja z animacjami	41
Niedopracowany widok	42

Aplikacja kopiująca widoki	43
Aplikacja prezentująca zbyt dużo informacji	43
Nierentowna aplikacja	44
Aplikacja informacyjna	45
Studia przypadków dla modeli aplikacji	46
Studium przypadku: gra społecznościowa ze znajomymi	46
Studium przypadku: aplikacje sprzedaży produktów	50
Studium przypadku: aplikacje uwzględniające położenie użytkownika	53
Krótkie wskazówki na początek	56
Należy zdefiniować docelowych odbiorców	57
Możliwie wczesne budowanie punktów integracji z serwisem społecznościowym	57
Budowanie z myślą o elementach komercyjnych	57
Tworzenie dopracowanych, atrakcyjnych widoków	58
2. Odzworowywanie relacji użytkowników na podstawie grafu powiązań społecznościowych	59
Graf powiązań społecznościowych w internecie	59
Stosowanie grafu rzeczywistych powiązań społecznościowych w wirtualnym świecie	61
Automatyczne dzielenie użytkowników na klastry	62
Prywatność i bezpieczeństwo	62
Budowanie zaufania	63
Udostępnianie prywatnych danych użytkownika: model opt-in kontra model opt-out	63
Model udostępniania za zgodą użytkownika (opt-in)	63
Model wyłączania udostępniania na wniosek użytkownika (opt-out)	64
Zrozumienie modelu relacji	65
Model śledzenia	65
Model połączeń	66
Model grupowy	67
Relacje kontra podmioty	71
Budowanie związków społecznościowych — analiza grafu powiązań społecznościowych Facebooka	72
Budowanie na bazie rzeczywistych tożsamości	72
Zrozumienie najskuteczniejszych kanałów komunikacji	73
Budowanie grup użytkowników	74
Unikanie grafów nieistotnych powiązań społecznościowych	74
Wskazywanie lubianych i nielubianych podmiotów za pośrednictwem protokołu OpenLike	75
Integracja widgetu OpenLike	75
Sposób prezentowania oznaczeń „Lubię to”	76
Podsumowanie	76

3. Tworzenie podstawowych elementów platformy aplikacji społecznościowych	79
Czego nauczysz się w tym rozdziale?	79
Apache Shindig	79
Konfiguracja kontenera Shindig	80
Instalacja kontenera Shindig w systemie Mac OS X (Leopard)	81
Instalacja kontenera Shindig w systemie Windows	84
Testowanie instalacji kontenera Shindig	86
Partuza	87
Wymagania	88
Instalacja kontenera Partuza w systemie Mac OS X (Leopard)	88
Instalacja kontenera Partuza w systemie Windows	91
Testowanie instalacji kontenera Partuza	96
Specyfikacja gadżetu OpenSocial w formacie XML	96
Konfigurowanie aplikacji za pomocą węzła ModulePrefs	97
Elementy Require i Optional	98
Element Preload	98
Element Icon	99
Element Locale	99
Element Link	100
Definiowanie preferencji użytkownika	101
Wyliczeniowe typy danych	103
Treść aplikacji	103
Definiowanie widoków treści	104
Treść wbudowana kontra treść zewnętrzna	110
Budowanie kompletnego gadżetu	111
 4. Definiowanie funkcji za pomocą odwołań JavaScriptu do elementów standardu OpenSocial	 115
Czego nauczysz się w tym rozdziale?	115
Dołączanie bibliotek JavaScriptu z funkcjami standardu OpenSocial	116
Dynamiczne ustawianie wysokości widoku gadżetu	117
Umieszczanie animacji Flash w ramach gadżetu	118
Wyświetlanie komunikatów dla użytkowników	119
Tworzenie komunikatu	120
Określanie położenia okien komunikatów	123
Definiowanie stylów komunikatów i okien	125
Zapisywanie stanu z preferencjami użytkownika	127
Programowe ustawianie tytułu gadżetu	129
Integracja interfejsu użytkownika gadżetu z podziałem na zakładki	130
Podstawowy gadżet	131
Tworzenie zakładki na podstawie kodu języka znaczników	131
Tworzenie zakładki w kodzie JavaScriptu	132
Uzyskiwanie i ustawianie informacji na temat obiektu TabSet	134

Rozszerzanie kontenera Shindig o własne biblioteki języka JavaScript	136
Budowanie kompletnego gadżetu	140
Przygotowanie specyfikacji XML gadżetu	140
Wyświetlanie gadżetu przy użyciu kontenera Shindig	144
5. Przenoszenie aplikacji, profili i znajomych	145
Czego nauczysz się w tym rozdziale?	145
Ocena obsługi standardu OpenSocial	145
Podstawowe elementy specyfikacji OpenSocial	147
Specyfikacja podstawowego serwera API	148
Specyfikacja podstawowego kontenera gadżetów	148
Specyfikacja serwera społecznościowego interfejsu API	149
Specyfikacja kontenera gadżetów społecznościowych	149
Specyfikacja kontenera OpenSocial	150
Tworzenie rozwiązań dla wielu kontenerów i przenoszenie aplikacji	150
Stosowanie mieszanego środowiska klient-serwer	151
Oddzielanie funkcji społecznościowych od podstawowego kodu aplikacji	151
Unikanie znaczników właściwych konkretnym kontenerom	151
Przenoszenie aplikacji z Facebooka do kontenera OpenSocial	152
Stosowanie ramek iframe dla konstrukcji niebędących aplikacjami społecznościowymi	152
Wyodrębnianie logiki funkcji Facebooka	153
Oddzielenie kodu znaczników (wizualizacji) od logiki programu	153
Stosowanie punktów końcowych REST zamiast języka FQL	153
Stosowanie implementacji z zasadniczą częścią kodu po stronie serwera	154
Personalizacja aplikacji na podstawie danych zawartych w profilu	154
Obiekt Person	154
Metody wymiany danych obiektu Person	155
Pola dostępne w ramach obiektu Person	160
Rozszerzanie obiektu Person	183
Uzyskiwanie profilu użytkownika	189
Promowanie aplikacji z wykorzystaniem znajomych użytkownika	191
Generowanie żądań dotyczących znajomych użytkownika	192
Budowanie kompletnego gadżetu	193
Specyfikacja gadżetu	193
Kod języka znaczników	194
Kod języka JavaScript	195
Uruchamianie gadżetu	197
6. Aktywność użytkowników, publikowanie powiadomień aplikacji i żądanie danych w kontenerze OpenSocial	199
Czego nauczysz się w tym rozdziale?	200
Promocja aplikacji za pomocą strumienia aktywności w kontenerze OpenSocial	200
Personalizacja aplikacji na podstawie powiadomień w strumieniu aktywności	201
Generowanie powiadomień w celu zwiększania liczby użytkowników	202

Pasywne i bezpośrednie publikowanie powiadomień aplikacji	205
Bezpośrednie publikowanie powiadomień aplikacji	206
Pasywne publikowanie powiadomień aplikacji	207
Zrównoważone publikowanie powiadomień	209
Generowanie żądań AJAX i żądań dostępu do danych zewnętrznych	210
Generowanie standardowych żądań dostępu do danych	211
Umieszczanie treści w żądaniach danych	212
Używanie autoryzowanych żądań do zabezpieczania połączeń	213
Budowanie kompletnego gadżetu	221
7. Zaawansowane techniki OpenSocial i przyszłość tego standardu	225
Czego nauczysz się w tym rozdziale?	225
Potokowe przesyłanie danych	225
Rodzaje żądań danych	228
Udostępnianie danych dla żądań zewnętrznych	233
Korzystanie z potokowego przesyłania danych po stronie klienta	234
Obsługa błędów generowanych przez potok danych	237
Parametry dynamiczne	238
Szablony OpenSocial	240
Alternatywny model kodu języka znaczników i danych	241
Wyświetlanie szablonów	243
Wyrażenia	247
Zmienne specjalne	248
Wyrażenia warunkowe	250
Przetwarzanie treści w pętli	253
Łączenie potokowego przesyłania danych i szablonów	258
Pozostałe znaczniki specjalne	260
Biblioteki szablonów	262
Interfejs API języka JavaScript	265
Kilka dodatkowych znaczników — język znaczników OpenSocial	270
Wyświetlanie nazwiska użytkownika — znacznik os:Name	271
Lista wyboru użytkownika — znacznik os:PeopleSelector	271
Wyświetlanie odznaki użytkownika — znacznik os:Badge	272
Ładowanie zewnętrznego kodu HTML — znacznik os:Get	272
Obsługa lokalizacji za pomocą pakietów komunikatów	272
Biblioteki API protokołu OpenSocial REST	275
Dostępne biblioteki	275
Przyszłość standardu OpenSocial: obszary rozwoju	276
Kontenery korporacyjne	276
Mobilna rewolucja	277
Rozproszone frameworki internetowe	277
Standard OpenSocial i rozproszone frameworki internetowe	277
Standard Activity Streams	278
Protokół PubSubHubbub	278

Protokół Salmon	279
Protokół Open Graph	280
Budowanie kompletnego gadżetu	281

8. Zagadnienia związane z bezpieczeństwem aplikacji społecznościowych287

Czego nauczysz się w tym rozdziale?	287
Wykonywanie zewnętrznego kodu za pośrednictwem ramek iframe	288
Bezpieczny model — projekt Caja	288
Dlaczego warto używać kompilatora Caja?	289
Rodzaje ataków — jak Caja chroni użytkownika?	289
Przekierowywanie użytkowników bez ich zgody	290
Śledzenie historii przeglądarki użytkownika	290
Wykonywanie dowolnego kodu za pomocą funkcji <code>document.createElement</code>	291
Rejestrowanie klawiszy naciskanych przez użytkownika	291
Konfiguracja kompilatora Caja	293
Przetwarzanie skryptów za pomocą kompilatora Caja z poziomu wiersza poleceń	295
Zabezpieczanie kodu HTML-a i JavaScriptu	295
Zmiana docelowego formatu kodu	300
Uruchamianie kompilatora Caja z poziomu aplikacji internetowej	301
Stosowanie kompilatora Caja dla gadżetu OpenSocial	303
Dodawanie kompilatora Caja do gadżetu	303
Praktyczny przykład	304
Wczesne wykrywanie niebezpiecznych elementów JavaScriptu za pomocą narzędzia JSLint	305
Eksperymenty w środowisku Caja Playground	306
Wskazówki dotyczące pracy w środowisku Caja	306
Implementacja modułowego kodu — kompilatora Caja nie należy stosować dla całego projektu	307
Stosowanie wstępnie przetworzonych bibliotek JavaScriptu	308
Nie należy używać Firebuga dla przetworzonego kodu źródłowego JavaScriptu	309
Nie należy umieszczać zdarzeń w kodzie języka znaczników	309
Centralizacja kodu JavaScriptu — stosowanie wyłącznie żądań danych i kodu języka znaczników	311
Lżejsza alternatywa dla kompilatora Caja: narzędzie ADsafe	312
ADsafe kontra Caja — którego narzędzia używać?	313
Jak zaimplementować środowisko ADsafe?	314
Konfiguracja obiektu środowiska ADsafe	314
Obiekt DOM	315
Wybór konkretnych węzłów DOM za pomocą zapytań	317
Praca z obiektami pakietów	321
Dołączanie zdarzeń	327
Definiowanie bibliotek	328

Budowanie kompletnego gadżetu	329
Źródło danych	330
Sekcja nagłówkowa: dołączane skrypty i style	330
Ciało: warstwa języka znaczników	332
Ciało: warstwa języka JavaScript	332
Ostateczny wynik	334
Podsumowanie	335
 9. Zabezpieczanie dostępu do grafu powiązań społecznościowych za pomocą standardu OAuth	337
Punkt wyjścia — uwierzytelnianie podstawowe	337
Implementacja uwierzytelniania podstawowego — jak to działa?	338
Wady stosowania uwierzytelniania podstawowego	339
Standard OAuth 1.0a	340
Przepływ pracy w standardzie OAuth 1.0a	341
Standard OAuth z perspektywy użytkownika końcowego	348
Dwuetapowa autoryzacja OAuth kontra trzyetapowa autoryzacja OAuth	350
Przykład implementacji trzyetapowej autoryzacji OAuth	354
Narzędzia i wskazówki związane z diagnozowaniem problemów	369
OAuth 2	373
Przepływ pracy w standardzie OAuth 2	373
Przykład implementacji: Facebook	381
Przykład implementacji: żądanie dodatkowych informacji na temat użytkownika w procesie autoryzacji OAuth w serwisie Facebook	392
Przykład implementacji: aplikacja z perspektywy użytkownika końcowego	394
Wskazówki dotyczące diagnozowania problemów z żądaniami	396
Podsumowanie	400
 10. Przyszłość serwisów społecznościowych: definiowanie obiektów społecznościowych za pośrednictwem rozproszonych frameworków sieciowych	401
Czego nauczysz się w tym rozdziale?	401
Protokół Open Graph — definiowanie stron internetowych jako obiektów społecznościowych	402
Wzloty i upadki metadanych	403
Działanie protokołu Open Graph	403
Implementacja protokołu Open Graph	404
Rzeczywisty przykład: implementacja protokołu Open Graph w serwisie Facebook	410
Praktyczna implementacja: odczytywanie danych protokołu Open Graph ze źródła w internecie	413
Wady protokołu Open Graph	419
Strumień aktywności: standaryzacja aktywności społecznościowych	420
Dlaczego warto zdefiniować standard dla aktywności?	421
Implementacja standardu Activity Streams	421

Typy obiektów	424
Czasowniki	426
WebFinger — rozszerzanie grafu powiązań społecznościowych na podstawie adresów poczty elektronicznej	429
Od finger do WebFinger: geneza protokołu WebFinger	429
Implementacja protokołu WebFinger	430
Wady protokołu WebFinger	432
Protokół OExchange — budowanie grafu udostępniania treści społecznościowych	433
Jak działa protokół OExchange?	433
Zastosowania protokołu OExchange	434
Implementacja protokołu OExchange	435
Protokół PubSubHubbub: rozpowszechnianie treści	440
Jak działa protokół PubSubHubbub?	441
Zalety z perspektywy wydawców i subskrybentów	443
Serwery hubów i usługi implementacji	445
Biblioteki przepływu pracy	445
Budowanie wydawcy w języku PHP	446
Budowanie wydawcy w języku Python	448
Budowanie subskrybenta w języku PHP	450
Budowanie subskrybenta w języku Python	452
Protokół Salmon: ujednolicenie stron konwersacji	455
Działanie protokołu Salmon	455
Budowanie rozwiązań na bazie protokołu PubSubHubbub	457
Ochrona przed nadużyciami i spamem	458
Przegląd implementacji	459
Podsumowanie	460
 11. Rozszerzanie grafu powiązań społecznościowych	
za pomocą standardu OpenID	461
Standard OpenID	461
Klucz do sukcesu — decentralizacja	462
Udoskonalenia względem tradycyjnego logowania	462
Dostęp do istniejącej bazy danych użytkowników i grafu powiązań społecznościowych	462
Czy już teraz dysponuję implementacją standardu OpenID?	
Gdzie mam jej szukać?	463
Procedura uwierzytelniania OpenID	464
Krok 1.: żądanie logowania przy użyciu identyfikatora OpenID	464
Krok 2.: operacja odkrywania w celu wyznaczenia adresu URL punktu końcowego	465
Krok 3.: żądanie uwierzytelnienia użytkownika	466
Krok 4.: udostępnienie stanu sukcesu lub niepowodzenia	467
Dostawcy OpenID	469
Omijanie problemów odkrywania domen w standardzie OpenID	469

Rozszerzenia standardu OpenID	471
Rozszerzenie Simple Registration	472
Rozszerzenie Attribute Exchange	473
Rozszerzenie Provider Authentication Policy Extension	479
Aktualnie tworzone rozszerzenia	483
Przykład implementacji: OpenID	484
Implementacja standardu OpenID w języku PHP	485
Implementacja standardu OpenID w języku Python	497
Typowe błędy i techniki diagnostyczne	508
Nie zgodność adresu URL wywołań zwrotnych	509
Brak możliwości odkrycia identyfikatora OpenID	509
Podsumowanie	510

12. Uwierzytelnianie hybrydowe

— wygoda użytkownika i pełen dostęp do profilu	511
Rozszerzenie hybrydy standardów OpenID i OAuth	511
Istniejące implementacje	512
Kiedy należy używać standardu OpenID, a kiedy jego hybrydy ze standardem OAuth?	512
Pytania, na które warto sobie odpowiedzieć przed wybraniem właściwego rozwiązania	512
Zalety i wady: standardowa implementacja OpenID	513
Zalety i wady: uwierzytelnianie hybrydowe	514
Przebieg uwierzytelniania w modelu hybrydowym na bazie standardów OpenID i OAuth	515
Kroki 1. i 2.: odkrywanie (pierwsze dwa kroki procedury OpenID)	516
Krok 3.: akceptacja uprawnień przez użytkownika	516
Krok 4.: przekazanie stanu akceptacji/odrzućenia żądania OpenID i parametrów rozszerzenia hybrydowego	517
Krok 5.: wymiana wstępnie zaakceptowanego tokenu żądania na token dostępu	519
Krok 6.: generowanie podpisanych żądań dostępu do chronionych danych użytkownika	520
Przykład implementacji: OpenID, OAuth i Yahoo!	521
Konfiguracja aplikacji: uzyskanie kluczy standardu OAuth na potrzeby procesu uwierzytelniania hybrydowego	521
Implementacja uwierzytelniania hybrydowego w języku PHP	522
Implementacja uwierzytelniania hybrydowego w języku Python	533
Podsumowanie	546

Dodatek A Podstawowe zagadnienia związane z budową aplikacji internetowych	547
---	------------

Dodatek B Słownik pojęć	563
--------------------------------------	------------

Skorowidz	567
------------------------	------------