

O'REILLY®



Interfejs API Strategia programisty

POZNAJ POTENCJAŁ INTERFEJSÓW API!

Helion

Daniel Jacobson,
Greg Brail, Dan Woods

Tytuł oryginału: APIs: A Strategy Guide

Tłumaczenie: Piotr Pilch

ISBN: 978-83-283-0558-8

© 2015 Helion S.A.

Authorized Polish translation of the English edition of APIs: A Strategy Guide,
ISBN 9781449308926 © 2012 Evolved Media.

This translation is published and sold by permission of O'Reilly Media, Inc.,
which owns or controls all rights to publish and sell the same.

All rights reserved. No part of this book may be reproduced or transmitted in any
form or by any means, electronic or mechanical, including photocopying, recording
or by any information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości
lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione.
Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie
książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie
praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi
bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte
w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej
odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne
naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION
nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe
z wykorzystania informacji zawartych w książce.

Wydawnictwo HELION

ul. Kościuszki 1c, 44-100 GLIWICE

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

http://helion.pl/user/opinie/inapst_ebook

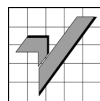
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

- [Poleć książkę na Facebook.com](#)
- [Kup w wersji papierowej](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Interfejs API. Strategia programisty

Daniel Jacobson, Greg Brail, Dan Woods



HELION

O'REILLY™

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

Spis treści

Przedmowa	9
1. Możliwości interfejsów API	13
Dlaczego napisano tę książkę?	15
Dla kogo przeznaczona jest ta książka?	17
Czym jest interfejs API?	17
Czym interfejs API różni się od witryny internetowej?	18
Interfejsy API i witryny internetowe mają jednakże wiele wspólnego	20
Kto korzysta z interfejsu API?	21
Typy interfejsów API	21
Dlaczego teraz?	23
2. Interfejsy API jako strategia biznesowa	25
Rozwój interfejsów API	28
Dlaczego interfejs API może okazać się potrzebny?	30
Potrzebujesz drugiej aplikacji dla urządzeń przenośnych	30
Klienci lub partnerzy pytają o interfejs API	31
Witryna internetowa zaczyna być używana do wyodrębniania wyświetlanych danych	32
Wymagasz większej elastyczności w zakresie udostępniania treści	32
Istnieją dane, które mają zostać udostępnione	33
Konkurencja z branży korzysta z interfejsu API	33
Chcesz umożliwić potencjalnym partnerom poznanie nowości	34
Wymagasz skalowania integracji z klientami i partnerami	34
Interfejs API ulepsza architekturę techniczną	36
3. Łańcuch wartości interfejsu API	37
Definiowanie łańcucha wartości: zadawanie kluczowych pytań	38
Tworzenie łańcucha wartości prywatnego interfejsu API	41
Metody użycia prywatnego interfejsu API	42
Zalety prywatnych interfejsów API	45
Zagrożenia związane z prywatnymi interfejsami API	46

Tworzenie łańcucha wartości publicznych interfejsów API	47
Metody wykorzystania publicznego interfejsu API	48
Zalety publicznych interfejsów API	51
Zagrożenia związane z publicznymi interfejsami API	52
Przemiana: publiczny interfejs API zamiast prywatnego,	
prywatny interfejs zamiast publicznego	53
Netflix: zamiana publicznego interfejsu API na prywatny	54
Modele biznesowe z interfejsami API	
używane do współpracy z partnerami	56
Zwiększanie zasięgu: więcej aplikacji, więcej platform	56
Osiąganie pośredniego dochodu	57
Zwiększanie innowacji we współpracy z partnerami	58
Zwiększanie wartości aplikacji poprzez integrację	58
Wariant użycia po części darmowy, a po części płatny	59
Postrzeganie modeli biznesowych interfejsów API przez firmę	
ProgrammableWeb.com	60
4. Przygotowywanie strategii produktów z interfejsami API	65
Określanie jasnego celu biznesowego	66
Określenie wizji dotyczącej interfejsu API	66
Podstawy strategii dotyczącej interfejsu API	68
Interfejsy API wymagają sponsora biznesowego	69
Typy strategii związanych z interfejsami API	70
Strategie dotyczące prywatnych interfejsów API	71
Strategie dotyczące publicznych interfejsów API	72
Tworzenie zespołu	73
Promotor projektantów	74
Zastrzeżenia dotyczące interfejsów API	76
5. Kluczowe zasady projektowania interfejsów API	81
Projektowanie interfejsów API dla konkretnych grup odbiorców	82
Projektowanie pod kątem projektantów	83
Projektowanie pod kątem użytkowników aplikacji	85
Najlepsze praktyki związane z projektem interfejsów API	85
Odróżnij własny interfejs API	86
Zapewnij łatwość testowania i używania interfejsu API	87
Zapewnij łatwość zrozumienia interfejsu API	88
Nie rób niczego dziwnego	89
Mniej znaczy więcej	90
Ukierunkowanie na konkretny segment projektantów	91

Kwestie techniczne projektu interfejsów API	91
Usługa REST	92
Przykład: projektowanie z wykorzystaniem pragmatycznej usługi REST	97
Kontrola wersji i projekt interfejsu API	100
Projektowanie infrastruktury dla interfejsów API	106
Centrum danych czy chmura?	106
Strategie buforowania	107
Kontrolowanie ruchu sieciowego generowanego przez interfejsy API	109
6. Zabezpieczenia interfejsów API i zarządzanie użytkownikami	111
Zarządzanie użytkownikami	112
Czy niezbędne jest rozpoczynanie od podstaw?	113
Pytania, które należy zadać odnośnie do zarządzania użytkownikami	113
Identyfikacja	114
Uwierzytelnianie: potwierdzanie tożsamości użytkownika	116
Nazwy użytkowników i hasła	116
Uwierzytelnianie oparte na sesji	117
Inne metody uwierzytelniania	118
Protokół OAuth	118
Ulepszanie uwierzytelniania za pomocą protokołu SSL	120
Szyfrowanie	122
Wykrywanie zagrożeń i zapobieganie im	123
„Wstrzykiwanie” kodu SQL	124
Ataki z wykorzystaniem formatów XML i JSON	124
Maskowanie danych	125
Ogólne zalecenia	126
Zalecenia dotyczące ochrony danych interfejsu API	126
Zalecenia dotyczące zabezpieczeń interfejsów API	126
7. Kwestie prawne związane ze strategią interfejsu API	129
Zarządzanie prawami	130
Praktyka: zarządzanie prawami w organizacji NPR	130
Umowy i warunki użytkowania	133
Zasady prywatności	135
Zasady utrzymywania danych	136
Przypisywanie właściciela treści i budowanie świadomości marki	137
Reagowanie na niewłaściwe użycie	137

8. Obsługa interfejsu API i zarządzanie nim	139
Obsługa interfejsu API	140
Informacje operacyjne na żądanie:	
strona statusu interfejsu API	141
Radzenie sobie z problemami operacyjnymi	142
Umowy SLA	143
Zarządzanie problemami	143
Monitorowanie i wsparcie operacyjne	144
Dokumentowanie interfejsu API	145
Zestaw procedur operacyjnych	147
Metody zarządzania ruchem sieciowym	148
Zarządzanie ruchem sieciowym na poziomie biznesowym	148
Operacyjne zarządzanie ruchem sieciowym	152
Zarządzanie ruchem sieciowym i skalowalność	154
Bramy interfejsów API	155
9. Określanie miary sukcesu interfejsu API	159
Obsługa metryk interfejsów API	160
Dlaczego gromadzone są wzorce wykorzystania	
na potrzeby metryk?	161
Żądania i odpowiedzi	162
Wyświetlenia	162
Lojalność	163
Metryki operacyjne	164
Metryki związane z efektywnością	166
Metryki związane z wydajnością	166
Kluczowe pytania związane z wydajnością interfejsów API	167
10. Angażowanie projektantów w proces adaptacji	173
Co motywuje projektantów?	174
Kluczowe elementy programu dla projektantów	174
Produkt (czyli najpierw musisz dysponować znakomitym	
interfejsem API!)	175
Dostęp do interfejsu API i jego twórcy	175
Warunki biznesowe i oczekiwania względem umów SLA	176
Treść	177
Świadomość istnienia interfejsu API	177
Skoncentruj się na pełnym komforcie pracy projektantów	178
Społeczność	179

Anatomia portalu projektantów	179
Działania zalecane i niezalecane w procesie angażowania projektantów	184
Działania zalecane	184
Działania niezalecane	188
11. Epilog: to dopiero początek	191
Skorowidz	193

Przedmowa

Konwencje zastosowane w książce

W książce zastosowano następujące konwencje typograficzne:

Kursywa lub **pogrubienie**

Styl identyfikuje nowe pojęcia, adresy URL, adresy e-mail, nazwy plików i ich rozszerzenia.

Stała szerokość

Styl jest stosowany w akapitach do wyróżniania elementów programów, takich jak nazwy zmiennych lub funkcji, a także baz danych, typów danych, zmiennych środowiskowych, instrukcji i słów kluczowych.

Stała szerokość z pogrubieniem

Styl służy do wyróżniania poleceń bądź innego tekstu, który powinien być wprowadzony przez użytkownika.



Ta ikona oznacza wskazówkę, sugestię lub ogólną uwagę.



Ta ikona wskazuje na ostrzeżenie.

Korzystanie z przykładowych kodów

Książka ma na celu ułatwienie wykonania konkretnej pracy. Ogólnie rzecz biorąc, kod zawarty w książce może być używany w tworzonych programach i dokumentacji. Nie ma potrzeby kontaktowania się z wydawnictwem w celu uzyskania zgody, o ile nie jest ponownie wykorzystywana znaczna ilość kodu. Na przykład tworzenie programu, w którym umieszczono kilka małych fragmentów kodu z książki, nie wymaga zgody. Inaczej jest w przypadku sprzedaży lub dystrybucji przez wydawnictwo dysków CD-ROM z przykładami. Udzielanie odpowiedzi na pytanie z dołączeniem cytatu z książki, a także przykładowego kodu nie wymaga zgody. Zgoda jest niezbędna w przypadku uwzględnienia w dokumentacji własnego produktu znacznej ilości przykładowego kodu zawartego w książce.

Wzmianka o twórcach jest mile widziana, lecz nie konieczna. Informacja taka zwykle obejmuje tytuł, autora, wydawnictwo i numer ISBN. Na przykład: *Interfejs API. Strategia programisty*, Daniel Jacobson, Greg Brail i Dan Woods (wydawnictwo Helion). Copyright 2012 Evolved Media, 9781449308926.

Jeśli czytelnik uzna, że użycie przykładowych kodów nie jest objęte powyższymi przypadkami wymogu uzyskania zgody, powinien skontaktować się z wydawnictwem, korzystając z adresu <ftp://ftp.helion.pl/przyklady/inapst.zip>.

Podziękowania

Książka nie powstałaby bez Scotta Regana, czyli niewymienionego autora. Scott był niespożytym źródłem energii, przywództwa i wsparcia. Szczególnie dobrze radził sobie ze znajdowaniem praktycznych przykładów, które ożywiają narrację.

John Musser zarówno uczestniczył w tworzeniu treści książki, jak i zapewnił mnóstwo spostrzeżeń, bazując na swoim dużym doświadczeniu z interfejsami API zdobytym podczas korzystania z serwisu Programmable Web. John okazał się cennym propagatorem idei i doradcą w zakresie kwestii ogólnych, a także szczegółów technologii.

Sam Ramji poświęcił swój czas i służył przemyślanym przywództwem w przypadku swoich wywiadów i wyrażanych opinii. W podobny sposób Brian Mulloy zaoferował nam swój czas i umiejętności. Harold Neal znalazł trochę czasu w swoim napiętym harmonogramie związanym z pracą

w instytucie Center for American Progress, aby uczestniczyć w wywiadach i recenzjach. Shanley Kane podzieliła się z nami swoją wiedzą na temat zarządzania społecznościami związanymi z interfejsami API. Szczególnie chcemy podziękować Chetowi Kapoorowi z firmy Apigee za przedstawienie jego punktu widzenia dotyczącego rynków, a także za wsparcie tego projektu.

Chcemy również podziękować osobom aktywnie działającym w świecie interfejsów API, które umożliwiły nam przeprowadzenie z nimi wywiadu. Są to: Derek Willis i Derek Gottfrid, którzy zajmowali się tworzeniem interfejsu API wydawnictwa „The New York Times”, Steve Smith i Chris Patti z firmy AccuWeather, Tim Madewell z firmy Innotas, Jason Sirota z firmy XO Group Inc. oraz Kin Lane, promotor interfejsów API. Wypowiedzi wszystkich tych osób przyczyniły się do powstania tej książki, a ponadto sprawiły, że teoria została zastosowana praktycznie.

Wyrazy wdzięczności kierujemy do następujących osób: Sophie Jasson-Holt, Deb Cameron, Dana Safarik, Deb Gabriel i Henry’ego Coupeta z zespołu firmy Evolved Media. Dziękujemy wszystkim, którzy zapewнили nam wsparcie w zakresie redagowania i zarządzania projektem. Przyczynili się oni do szybkiego powstania książki i zapewnienia jej wysokiej jakości.

Daniel chce również podziękować Michaelowi Hartowi, który rozpoczął realizowanie programu tworzenia interfejsu API firmy Netflix. Program ten niezauważalnie wpływa na treść książki w postaci różnych przykładów odnoszących się do firmy Netflix. Chcemy też podziękować Zachowi Brandowi, który udostępnił nam aktualne obrazy i statystyki dotyczące interfejsu API organizacji NPR.

Choć książka w dużej mierze bazuje na naszym osobistym doświadczeniu związanym ze światem interfejsów API, jest ono wzbogacone naszymi rozmowami z wieloma wspaniałymi liderami w tej dziedzinie.

W szczególności wszyscy mieliśmy przyjemność współpracy z kilkoma z najbardziej utalentowanych osób w branży, które wpłynęły na kształt książki zarówno w sposób subtelny, jak i mniej subtelny. Co więcej, z czasem nasze punkty widzenia zaczęły się przenikać, do czego przyczyniły się artykuły, prezentacje, nieformalne rozmowy i inne interakcje z wieloma osobami, które sprawiły, że rozwój interfejsów API osiągnął obecnie taki, a nie inny etap. Wszystkim tym osobom (wicie, o kim mowa) dziękujemy za ich pośredni wkład — niecierpliwie będziemy śledzić, w jakim kierunku nastąpi rozwój branży interfejsów API.

Możliwości interfejsów API

Interfejsy API to coś poważnego, co cały czas zyskuje na znaczeniu. Pionierskie w tej dziedzinie firmy, takie jak Google, Facebook, Apple i Twitter, udostępniły publicznie zadziwiające rozwiązania technologiczne, powodując transformację istniejących firm i stworzenie nowych branż. Kluczową rolę w sukcesie odniesionym przez te firmy odgrywają interfejsy API, które powiązują użytkowników i ich urządzenia przetwarzające z bazowymi platformami. Platformy te obsługują każdą działalność biznesową i łączą ze sobą wewnętrzne mechanizmy poszczególnych firm.

Świat zaczął się już zmieniać. Przeanalizuj następujące przykłady:

- Firma Salesforce.com stworzyła duży i rozbudowany „ekosystem” partnerów, udostępniając im podstawowe usługi w celu rozszerzania ich i wprowadzania do nich innowacji. Obecnie interfejs API serwisu Salesforce obsługuje większy ruch sieciowy niż witryna internetowa tej firmy. W połowie roku 2011 ponad 60% ruchu sieciowego kierowanego do tego serwisu trafiało do niego za pośrednictwem interfejsów API.
- Firma Amazon udostępniła swoją podstawową infrastrukturę obliczeniową jako usługi AWS (*Amazon Web Services*), z których można korzystać za pośrednictwem kilku interfejsów API. Obecnie większy ruch sieciowy firma obsługuje przy użyciu usług AWS niż za pomocą jej wszystkich razem wziętych globalnych witryn sklepów.
- Twitter to najbardziej oczywisty przykład firmy, która prawie całkowicie bazuje na interfejsie API i „ekosystemie” aplikacji programistów.
- Firma Netflix zupełnie zmieniła to, w jaki sposób pobierane są filmy i programy telewizyjne metodą strumieniowania na setki różnych urządzeń. Firma ta nie tylko spowodowała poważne zmiany w branży

wypożyczalni wideo, ale też wpłynęła na duże segmenty pokrewne, takie jak telewizja kablowa. Interfejsy API umożliwiają firmie Netflix obsługę mnóstwa urządzeń w przystępny sposób.

- Firma NPR natchnęła swoim interfejsem API środowisko inżynierskie branży mediów cyfrowych. Interfejs ten obsługuje witrynę internetową, aplikacje dla urządzeń przenośnych oraz wszystkie inne formy dystrybucji i udostępniania, które są stosowane przez firmę. Interfejs API dokonał również transformacji relacji firmy z jej stacjami członkowskimi, a także tego, w jaki sposób firma NPR współużytkuje z nimi treść.

Przeanalizuj następujące trendy branżowe:

- Na początku roku 2011 sprzedaż smartfonów była większa niż sprzedaż komputerów PC. Firma Morgan Stanley przewidywała, że do końca 2012 r. na świecie będzie istnieć więcej podłączonych urządzeń przenośnych niż komputerów PC.
- Organizacja CTIA (stowarzyszenie z branży technologii bezprzewodowych) stwierdziła, że w Stanach Zjednoczonych jest już więcej urządzeń bezprzewodowych niż ludzi.
- Analitycy rywalizują ze sobą w przewidywaniu, ile urządzeń przenośnych będzie istnieć w przyszłości. Organizacja GSMA (kolejne stowarzyszenie z branży technologii bezprzewodowych) prognozuje, że do roku 2020 będzie 20 miliardów podłączonych urządzeń przenośnych. Hans Vestberg, dyrektor generalny w firmie Ericsson, szacuje tę liczbę na 50 miliardów. Tymczasem Marthin De Beer, główny wiceprezydent jednostki Emerging Technologies Group w firmie Cisco, przewiduje, że do roku 2020 liczba ta może przekroczyć bilion.
- Firma Cisco prognozuje, że choć internetowy ruch sieciowy pochodzący z komputerów PC nadal będzie się zwiększać o 33% rocznie, ruch sieciowy generowany przez inne urządzenia każdego roku do 2015 r. będzie ponad dwukrotnie większy.
- Gdy pisano tę książkę, firma Facebook odpowiadała za ponad 25% całkowitej liczby wyświetleń stron internetowych. Interfejsy API obsługują zarówno produkty tej firmy, jak i jej „ekosystem”.
- Ponad 30% internetowego ruchu sieciowego w najlepszym czasie antenowym w Stanach Zjednoczonych powiązane jest ze strumieniowaniem w serwisie Netflix, które jest zapewniane i zarządzane poprzez interfejsy API.

Powyższe statystyki wskazują nie tylko na eksplozję ogólnego internetowego ruchu sieciowego, ale też na ogromną zmianę w zakresie dystrybucji tego ruchu do aplikacji i urządzeń. Po przyjrzeniu się tym przyspieszającym trendom bardzo łatwo można sobie wyobrazić, że najprawdopodobniej w ciągu kilku lat interfejsy API będą obsługiwać większość internetowego ruchu sieciowego.

Dlaczego napisano tę książkę?

Jako autorzy zajmujemy się interfejsami API po zdobyciu niedawno doświadczenia „na polu bitwy”. Daniel Jacobson kierował projektowaniem systemu zarządzania treścią NPR i interfejsu API, który z nim współpracuje. Interfejs ten stanowi obecnie centralny element strategii dystrybucji mediów cyfrowych firmy NPR. Pozwala on firmie zwiększyć możliwości w zakresie dotarcia do swoich użytkowników w przypadku szerokiej gamy platform.

Aktualnie Daniel kieruje pracami projektowymi związanymi z interfejsami API w firmie Netflix, której strategia odnośnie do tych interfejsów stanowi kluczowy element usługi strumieniowania Netflix. Usługa ta pozwala firmie zapewnić rozbudowane funkcje wideo w przypadku setek urządzeń. Sprawiała ona, że znacznie zwiększyło się tempo budowania nowych implementacji i dostarczania ich do nowych urządzeń. Dzięki takiemu rozwiązaniu niesamowicie poszerzyła się baza użytkowników serwisu Netflix. W rezultacie liczba żądań obsługiwanych przez interfejs API w ciągu roku wzrosła z miliarda miesięcznie do ponad miliarda dziennie.

Greg Brail, jeden z autorów tej książki, korzysta z wiedzy zdobytej jako dyrektor ds. technicznych w firmie Apigee. Pomógł on zaimplementować tam dziesiątki programów z interfejsem API, a ponadto brał udział w znacznie większej liczbie projektów. Dzięki pełnieniu takiej funkcji Greg ma też możliwość korzystania z doświadczenia całego zespołu projektowego firmy Apigee, do którego należały setki programistów. Greg zdobywał także wiedzę, bazując na wielu programach API dla przedsiębiorstw.

Napisaliśmy tę książkę, aby ułatwić zrozumienie potencjału interfejsów API. Ponadto zależy nam na tym, by czytelnik mógł tworzyć interfejs API z pełną świadomością. Książka nie jest podręcznikiem nauki programowania, lecz zbiorem najlepszych rozwiązań. Niezbędne jest zaznajomienie się zarówno z możliwościami, jak i kwestiami natury taktycznej związanymi z konstruowaniem interfejsu API.

Książka wprowadzi również osoby kierujące firmą i specjalistów od technologii w obszar możliwości interfejsów API. Bez wątpienia świat interfejsów API obejmuje mnóstwo technologii, ale tym, co często traci się w całym zamieszaniu, jest wpływ interfejsów na działalność biznesową. W książce wyjaśniono, jak traktować interfejsy API z perspektywy biznesowej i w jaki sposób mogą one pozytywnie wpłynąć na biznes.

Zależy nam także na przekazaniu informacji na temat tego, z czym wiąże się decyzja o zaprojektowaniu interfejsu API, oraz tego, jakie są implikacje udostępniania interfejsu, nie tylko z technologicznego punktu widzenia, ale też w odniesieniu do strategii biznesowej, kwestii prawnych i relacji z partnerami.

Pokażemy, że interfejsy API mają znaczący wpływ na świat biznesu. Najwyższa pora na wykorzystanie tego faktu.

W przeciwieństwie do wielu innych omówień interfejsów API, w których zajęto się wyłącznie tym, w jaki sposób duże firmy internetowe korzystają z tych interfejsów publicznie, w tej książce zwrócono też uwagę na prywatny wariant ich użycia. Jesteśmy przekonani, że ta opcja ma nawet większe znaczenie niż wiele bardziej rozpowszechnionych publicznych programów API, o których często można przeczytać.

Jako autorzy po części musimy zająć się światem technologii, a po części światem biznesu. W związku z tym mamy nadzieję, że uda nam się poszerzyć wiedzę kreatywnych szefów firm i specjalistów od technologii w zakresie metod wykorzystania interfejsów API w kontekście ich własnej działalności biznesowej.

W książce omówiono następujące zagadnienia:

- możliwości biznesowe interfejsów API;
- przykłady firm wykorzystujących interfejsy API do transformacji swojej działalności biznesowej, a w niektórych przypadkach branż, z którymi są związane;
- modele biznesowe używane na potrzeby interfejsów API;
- postać łańcucha wartości interfejsów API oraz metody aktywowania jego różnych elementów;
- uwagi dotyczące przygotowywania strategii związanej z interfejsami API oraz reagowania na obawy i zastrzeżenia;
- kwestie związane z projektem interfejsów API, a zwłaszcza z metodą ułatwiającą zaznajomienie z nimi programistów;

- zagadnienia dotyczące bezpieczeństwa interfejsów API z uwzględnieniem protokołu OAuth;
- implikacje prawne związane z prowadzeniem działalności biznesowej bazującej na interfejsach API, w tym ochrona prywatności i prawa do danych;
- kwestie związane z obsługą interfejsu API przy rozwoju działalności;
- sposób analizowania pomiarów wydajności programu API;
- zaangażowanie projektantów i budowanie społeczności w celu spowodowania adaptacji utworzonego interfejsu API.

Podsumowując, książka zapewnia plan działania w przypadku zastosowania interfejsów API do transformacji prowadzonej działalności biznesowej.

Dla kogo przeznaczona jest ta książka?

Istnieje zestaw książek poświęconych technicznym aspektom interfejsów API, w tym publikacje dotyczące technologii REST, OAuth, XML, JSON i innych. Ta książka nie ma na celu rywalizowania z nimi. Okazuje się, że choć prawie niemożliwe jest zajmowanie się interfejsami API bez poruszenia kwestii związanych z rozwiązaniami technicznymi, książka nie jest kierowana do specjalistów od technologii, którzy budują interfejsy API lub bezpośrednio z nich korzystają. Przewidziano ją raczej dla osób, które muszą podjąć strategiczne decyzje dotyczące tego, czy w przypadku ich firmy interfejs API to dobry pomysł.

W grupie odbiorców książki znajdują się: osoby zarządzające w firmach, członkowie zespołów zajmujących się rozwojem biznesowym, kierownicy projektów oraz propagatorzy rozwiązań technicznych. Oczywiście w książce zamieszczono wiele informacji dla specjalistów od technologii, ale na bardziej ogólnym poziomie.

Czym jest interfejs API?

API to skrót pochodzący od nazwy *Application Programming Interface*. Interfejs API może zapewnić współpracownikom, partnerom lub projektantom z zewnętrznych firm możliwość uzyskania dostępu do danych i usług w celu szybkiego tworzenia takich aplikacji jak przeznaczone dla urządzenia iPhone. Interfejsy API serwisów Twitter i Facebook to znane tego przykłady. Istnieją

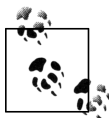
interfejsy API dostępne dla każdego projektanta, tylko dla partnerów, a także takie, które są używane wewnątrz firmy do ułatwienia procesu ulepszenia działalności biznesowej i współpracy między zespołami.

A zatem interfejs API to w zasadzie kontrakt. Po zastosowaniu go projektanci chętnie używają interfejsu, ponieważ wiedzą, że mogą na nim polegać. Kontrakt zwiększa zaufanie, a to powoduje większy poziom wykorzystania. Kontrakt zapewnia również znacznie efektywniejsze łącze między dostawcą a konsumentem, gdyż interfejsy są udokumentowane, spójne i przewidywalne.

Czym interfejs API różni się od witryny internetowej?

Interfejs API różni się dość znacznie od witryny internetowej. Witryna zapewnia informacje na żądanie. Firma udostępnia publicznie treść, a użytkownicy z niej korzystają. W przypadku witryn internetowych z użyciem treści nie są związane żadne kontrakty ani struktury. Jeśli treść w witrynie się zmienia, przy następnej wizycie odwiedzający ujrzą nową treść. Nie ma to wpływu na stosowane przez nich przeglądarki internetowe. Każda zmiana jest transparentna dla użytkownika. Jeżeli w znacznym stopniu przebudujesz witrynę internetową, będzie to dotyczyć użytkowników jedynie pod tym względem, że przyzwyczaili się do przeglądania treści rozmieszczonej w określony sposób. Ludzie znakomicie radzą sobie z wizualnym dopasowywaniem wzorców. Możemy szybko dostosować się do nowego wyglądu i znaleźć to, czego potrzebujemy. Nie oznacza to, że użytkownicy nie narzekają, gdy zostanie przebudowana ich ulubiona witryna, ale prawie zawsze z powodzeniem przyzwyczajają się do jej nowego wyglądu.

Interfejs API różni się od witryny znacząco, gdyż zawiera kontrakt, a programy są tworzone na jego bazie. W przeciwieństwie do ludzi programy nie są elastyczne i niemal zawsze kiepsko radzą sobie z dopasowywaniem wzorców. Jeśli cokolwiek zmienisz w kontrakcie interfejsu API, w przypadku aplikacji zbudowanych na jego bazie efekt uboczny tej zmiany może być potencjalnie dość duży.



Pamiętaj o tym, że struktura interfejsu API stanowi część kontraktu. Kontrakt to zobowiązanie, którego nie można w zwykły sposób zmienić.

Interfejs API należy traktować jako produkt w postaci oprogramowania, który uwzględnia sprawdzanie wersji, zgodność wstecz i wydłużenie czasu wymaganego do dostosowania wszystkich nowych funkcji. Powinna występować

Nasza definicja robocza interfejsu API

Oto definicja techniczna: interfejs API określa sposób komunikowania się ze sobą dwóch aplikacji komputerowych za pośrednictwem sieci (przeważnie internetu) z wykorzystaniem wspólnego, zrozumiałego dla nich obu języka.

Interfejsy API są zgodne ze specyfikacją. Oznacza to, że:

- Dostawca interfejsów API opisuje dokładnie, jakie zaoferują one funkcje.
- Dostawca interfejsów API określa, kiedy funkcje będą dostępne oraz kiedy mogą się zmienić w niezgodny sposób.
- Dostawca interfejsów API może wyszczególnić dodatkowe występujące w nich ograniczenia techniczne, takie jak limity ilości, które kontrolują to, ile razy określona aplikacja lub użytkownik końcowy może skorzystać z interfejsu API w danej godzinie, dniu bądź miesiącu.
- Dostawca interfejsów API może podać dodatkowe ograniczenia natury prawnej albo biznesowej dotyczące korzystania z interfejsów, takie jak ograniczenia związane ze stosowaniem marek, rodzajem użycia itp.
- Projektanci zgadzają się na używanie interfejsu API zgodnie z podanym opisem, na korzystanie tylko z tych, które wyszczególniono, a także na postępowanie w zgodzie z regułami ustalonymi przez dostawcę.

Ponadto dostawca interfejsów API może oferować inne elementy, takie jak:

- mechanizmy zapewniające dostęp do interfejsu API i pozwalające zaznajomić się z warunkami ich użycia;
- dokumentacja ułatwiająca zrozumienie interfejsu API;
- zasoby, takie jak przykładowe programy, a także społeczności projektantów wspierające osoby korzystające z interfejsu API;
- informacje operacyjne o kondycji interfejsu API i poziomie jego wykorzystania.

równowaga między obsługą istniejącej bazy i jednoczesnym uwzględnianiem niezbędnych zmian, aby interfejs API rozwijał się wraz z prowadzoną działalnością biznesową oraz podążał za planowanymi innowacjami.

Nie oznacza to, że interfejs nigdy nie może ulec zmianie. Wręcz przeciwnie — interfejs ten jest produktem internetowym, który może się zmieniać prawie nieustannie w celu spełnienia wymagań biznesowych lub obsługi bieżącego obciążenia związanego z ruchem sieciowym w najefektywniejszy sposób.

Są to jednak modyfikacje dotyczące implementacji, a nie interfejsu. Jeśli zostaną one właściwie wprowadzone, implementacja interfejsu API może się zmieniać codziennie albo nawet częściej, a sam interfejs pozostanie spójny.

Interfejsy API i witryny internetowe mają jednakże wiele wspólnego

Podobnie jak w przypadku witryn internetowych oczekuje się, że interfejsy API będą dostępne przez 24 godziny na dobę każdego dnia roku. Tak jak użytkownicy witryn internetowych projektanci nie będą mieć zbyt wiele cierpliwości w sytuacji, gdy każdego niedzielnego poranka wystąpi zaplanowany przestój. Wszystko to może spowodować wyzwanie w postaci zbudowania interfejsu API z wykorzystaniem istniejącej infrastruktury technologicznej przedsiębiorstwa z systemami, które mogły zostać zaprojektowane na bazie cyklu jednodniowego. W tym przypadku takie systemy są wyłączane na koniec dnia roboczego i włączane ponownie następnego dnia (na przykład systemy bankowości).

Poprawnie zaprojektowane witryny internetowe mogą być i są nieustannie aktualizowane w celu zmodyfikowania ich wyglądu i dostrojenia funkcji. Jest to możliwe, ponieważ witryny to jednostki aktywne również poza obrębem sieci, które z łatwością mogą być modyfikowane bez dokonywania zmian po stronie klientów. Nie ma potrzeby kierowania aktualizacji oprogramowania do użytkowników.

Pod tym względem interfejsy API nie różnią się wiele od witryn. Zakładając, że używany interfejs API pozostaje zgodny wstecz, program API może wprowadzić nowe funkcje i zmienić implementację już istniejących bez ingerencji po stronie klientów. Dopóki zostanie utrzymany kontrakt między interfejsem API a używającymi go projektantami, może on być zmieniany zgodnie z harmonogramem dotyczącym witryny internetowej, a nie z harmonogramem obowiązującym dla działu informatycznego przedsiębiorstwa. Wynikiem tego jest lepszy i szybciej działający program API.

Okazuje się, że zmiany wprowadzane zarówno w interfejsach API, jak i witrynach internetowych mogą być inicjowane przez analizę zachowania aplikacji i użytkowników końcowych. W obu przypadkach zespół zarządzający produktami i dbający o dobry projekt nieustannie sprawdza dane analityczne, aby stwierdzić, jakie elementy witryny lub interfejsu API działają z powodzeniem, a jakie nie. Uzyskany wynik powinien być rozpatrywany na zwykłych spotkaniach projektantów, dzięki czemu z czasem zostanie zbudowany znacznie bardziej niezawodny interfejs API bądź witryna internetowa.