

UNIwersytet Gdański

Wydział Matematyki, Fizyki i Informatyki

Artur Rybak

nr albumu: 179796

Projektowanie RESTful API na potrzeby aplikacji mobilnych

Praca magisterska na kierunku:

INFORMATYKA

Promotor:

dr Włodzimierz Bzyl

Gdańsk 2015

Streszczenie

Problematyka tej pracy dotyczy schematu komunikacji między urządzeniami mobilnymi a serwerem, jaki narzuca projekt interfejsu w postaci WebAPI. Przeanalizuję konsekwencje, jakie niesie za sobą obranie podejścia RESTful jako wzorca projektowego oraz gdy grupa klientów ograniczona zostanie do aplikacji mobilnych, tj. takich, które uruchamiane są z poziomu urządzeń przenośnych: telefonów, tabletów, zegarków, opasek, odzieży. Kierując się potrzebami oraz wymaganiami wpisanymi w sprzęt, przedstawię oraz zutylizuję w istniejącym projekcie praktyki, powszechnie uważane za właściwe. Słuszność rzeczonych praktyk potwierdzona będzie oszczędnym gospodarowaniem baterią urządzenia, minimalizowaniem zużycia dostępnego pakietu danych czy poprawnym reagowaniem na osiągalność i topologię sieci.

Modelowe przykłady API które oferują firmy takie, jak Facebook, Twitter, Flickr czy Google posłużą jako punkt odniesienia oraz źródło wzorców, którymi powinniśmy się kierować podczas tworzenia własnego WebAPI. Skupię się na najczęstszych problemach i możliwych rozwiązaniach struktury interfejsu, ekstensywnym wykorzystaniu protokołu HTTP oraz powszechnych rozwiązaniach dla sztywnych problemów takich jak autentykacja, wersjonowanie, rozszerzalność, monitoring czy obsługa błędów.

Wreszcie, zajmę się analizą wdrożonego interfejsu, na kilku przykładach po-

staram się przetransformować API w ten sposób, by intuicyjnie spełniało założenia REST. Przedstawię, jak moja implementacja odbiega od modelowej oraz ocenię czy stuprocentowa użycie podejścia RESTful jest zawsze wskazane i wymagane.

Słowa kluczowe

WebAPI, RESTful, aplikacje, mobilne, telefony, komunikacja, wzorce

Spis treści

Wprowadzenie	7
1. Rozumienie WebAPI	10
1.1. Jakie API jest użyteczne?	11
1.2. Praca z Web API	11
1.3. API First	11
2. Projekt WebAPI	12
2.1. Wartość biznesowa	12
2.2. Budowa poprawnego modelu	12
2.2.1. Rzeczowniki - zasoby, kontra czasowniki - akcje	12
2.2.2. Często popełniane błędy	12
2.2.3. Ograniczenia	12
3. Wykonanie WebAPI	13
3.1. Implementacja na wybranej platformie	13
3.2. Składnia zapytań i odpowiedzi	13
3.3. Utylizacja protokołu HTTP	13
3.4. Obsługa błędów	13

3.5. Wersjonowanie i rozszerzalność	13
4. Ewaluacja WebAPI	14
4.1. Porównanie funkcjonalne istniejącego web serwisu i RESTful API	14
4.2. Testowanie	14
4.2.1. Behawioralne	14
4.2.2. Wydajnościowe	14
4.2.3. Narzędzia testowania	14
Zakończenie	15
Bibliografia	16
Oświadczenie	18

Wprowadzenie

Urządzenia mobilne opanowały sposób w jaki komunikujemy się ze światem. Smartfon jest nam bliższy niż niejeden z członków rodziny. Nie ma w tym jednak nic dziwnego, skoro stał się naszą bramą do poznania, doświadczania i dzielenia się światem i przeżyciami. Mobilność sięga jednak coraz dalej. Telefon, tablet, zegarek, opaska na nadgarstku, okulary, buty, a niedługo pozostałe części naszej garderoby są lub będą częścią świata mobilnego. Niezaprzeczalny jest wpływ telefonu i tabletu na nasz rytm życia, wobec czego, biorąc pod uwagę powszechność rozwiązań mobilnych, postanowiłem przyjrzeć się kluczowej kwestii, jaką jest wydajny, dobrze zaprojektowany interfejs komunikacyjny, pozwalający trzymać w ryzach nawet najbardziej zaawansowany system.

Smartfon - będący naszym osobistym centrum sterowania musi w sposób efektywny porozumiewać się z użytkownikiem oraz światem zewnętrznym. Ubogie byłyby bowiem aplikacje mobilne z których korzystamy gdyby nie pomoc web serwisów. Z jednej strony dostarczają nam one niezbędnych informacji o pogodzie, ruchu drogowym, zmianach cen ropy czy naszych osiągnięciach na siłowni. Z drugiej konsumują to wszystko, co chcemy zachować dla siebie lub czym chcemy pochwalić się przed innymi. Zmieniają sposób w jaki jesteśmy połączeni z ludźmi z najbliższego otoczenia ale i najdalszych zakątków globu.

Intuicyjnie, jako ludzie, potrafimy się porozumiewać. W większości sytuacji wiemy, czy to, co i o czym mówimy kogoś rani, czy bawi. Wiemy, czy dialog z naszym rozmówcą "klei się" w całość, czy raczej jest to nerwowe odbijanie pytań i odpowiedzi. Całą naszą umiejętność i intuicję w tym zakresie opieramy na doświadczeniu. O ile nie jest więc dla nas problemem wysyłanie i odbieranie prostych komunikatów, o tyle zdolność do porozumiewania się między narodami wspólnym językiem biznesowym z zachowaniem wrażliwości na kulturę i poglądy rozmówców jest niezaprzeczalnie wyzwaniem.

Podobnie jest z aplikacjami mobilnymi: o ile łatwo jest z pomocą dostępnych narzędzi wysłać wiadomość w świat a może nawet odebrać odpowiedź na zadane pytanie, o tyle trudniej robić to w sposób wydajny i dobrze ustrukturyzowany. W ciągu ostatnich lat udało się nam jednak wypracować kilka reguł i wytycznych, które czynią komunikację szybką, czytelną, bezpieczną. Reguł, które czynią komunikację wydajną.

Od trzech lat usilnie przyglądam się światu przez pryzmat telefonu i jego możliwości. Przez ten czas dostrzegłem różne próby komunikacji, gdzie jako medium służy smartfon. Pierwotna jego funkcja - możliwość wykonywania połączeń głosowych za pośrednictwem sieci GSM jest dobrze zdefiniowana i wykorzystana ale funkcjonalnie mało rozszerzalna. Krótkie wiadomości tekstowe urozmaiciły formę komunikacji ale prawdziwą rewolucję przyniósł tani mobilny internet oraz upo-

wszechnienie się darmowych punktów dostępowych sieci WiFi. Procentowy udział w ruchu internetowym, jaki jest kreowany przez platformy mobilne wzrasta z upływem czasu. Dzieje się to za sprawą coraz doskonalszych aplikacji, przemyślanych interfejsów i wygodzie dostępu do informacji, jaką zyskujemy będąc w podróży czy odpoczywając po ciężkim dniu na kanapie, bo i ta nie jest naturalnym środowiskiem w którym korzystamy z komputera stacjonarnego czy laptopa.

Jako developer aplikacji mobilnych jestem świadomy, że sukces napisanych przeze mnie programów nie ma jednego źródła. Składają się na niego: przemyślany i przyjemny design aplikacji, rozpoznanie, zrozumienie i prawidłowe modelowanie potrzeb użytkownika oraz przypadków użycia. To i tak tylko wierzchołek góry lodowej. Doświadczenie jednak podpowiada, że dobrze zaprojektowana komunikacja klient (aplikacja mobilna) - serwer (w postaci WebAPI) wiele rzeczy upraszcza i pozwala się skupić na problemach funkcjonalnych, zamiast marnotrawić czas na kolejne techniczne *gotchas* (z ang. zasadzki).

Rozumienie WebAPI

Zanim skupimy się na kwestiach technicznych oraz na projektowaniu API, powinniśmy wiedzieć, po co w ogóle chcemy stwarzać możliwość na manipulację naszymi zasobami za pomocą zewnętrznego interfejsu. Pytanie zgoła podstawowe. Istotne jest jednak byśmy wracali do niego często w trakcie projektowania czy implementacji. Łatwo bowiem zgubić ideę którą chcieliśmy się podzielić podczas gdy pochłonie do reszty walka z przeciwnościami technicznymi oraz dylematy związane ze strukturą naszego interfejsu.

Projektowanie API nie jest żadnym wyjątkiem. Podobnie jest z tą pracą mądrą, podobnie z każdym pomysłem na biznes. Cel powinien nie tylko uświęcać środki ale i wyznaczać dalszą drogę. Cel powinien być zawsze widoczny.

By zrozumieć jak dobrze projektować API, postarajmy się więc najpierw zrozumieć jakie są wymagania i oczekiwania stawiane przed jego twórcami.

1.1. Jakie API jest użyteczne?

Dobrze zaprojektowane API pozwala deweloperom na tworzenie aplikacji "mashup", tj. takich, które łączą w sobie cechy kilku serwisów. Z pewnością istnieje uzasadnienie dla istnienia aplikacji, oferujących funkcjonalność specyficzną dla ich dziedziny. Nie potrzebują one elastyczności i gdyby porównać je do garniturów, to mielibyśmy do czynienia jedynie z sztytami na miarę. Takie aplikacje mogą bez problemów komunikować się z interfejsami bazującymi na akcjach lub za pomocą SOAP API. Pewną nadmiarowością byłoby także tworzenie generycznego serwisu tylko na potrzeby jednej aplikacji.

Jeśli jednak chcemy pobudzić społeczność i dać jej narzędzie, które będzie mogła KREATYWNIE wykorzystać, to być może wystawienie REST API jest dla nas najlepszym wyborem.

1.2. Praca z Web API

1.3. API First

ROZDZIAŁ 2

Projekt WebAPI

2.1. Wartość biznesowa

2.2. Budowa poprawnego modelu

2.2.1. Rzeczowniki - zasoby, kontra czasowniki - akcje

2.2.2. Często popełniane błędy

2.2.3. Ograniczenia

ROZDZIAŁ 3

Wykonanie WebAPI

3.1. Implementacja na wybranej platformie

3.2. Składnia zapytań i odpowiedzi

3.3. Utylizacja protokołu HTTP

3.4. Obsługa błędów

3.5. Wersjonowanie i rozszerzalność

ROZDZIAŁ 4

Ewaluacja WebAPI

4.1. Porównanie funkcjonalne istniejącego web serwisu i RESTful API

4.2. Testowanie

4.2.1. Behawioralne

4.2.2. Wydajnościowe

4.2.3. Narzędzia testowania

Zakończenie

Podsumowanie

Bibliografia

- [1] Kirsten L. Hunter. *Irresistible APIs. Create Web APIs that developers will love*. Manning Publications, 2015.
- [2] Kore.io [online], [dostęp: 15 czerwca 2015] dostępny w internecie: <http://kore.io/>.
- [3] Best practices for designing a pragmatic restful api [online], [dostęp: 15 czerwca 2015] dostępny w internecie: <http://www.vinaysahni.com/best-practices-for-a-pragmatic-restful-api>.
- [4] Chen-Che Huang, Jiun-Long Huang, Chin-Liang Tsai, Guan-Zhong Wu, Chia-Min Chen, and Wang-Chien Lee. Energy-efficient and cost-effective web api invocations with transfer size reduction for mobile mashup applications. *Wireless Networks*, 20(3):361–378, April 2014.
- [5] How to design rest apis for mobile? [online], [dostęp: 15 czerwca 2015] dostępny w internecie: <http://www.redotheweb.com/2012/08/09/how-to-design-rest-apis-for-mobile.html>.
- [6] Efficient mobile apis using apache thrift [online], [dostęp: 15 czerwca 2015] dostępny w internecie: <http://blog.whitepages.com/reducing-data-with-apache-thrift/>.

- [7] Rest and the promise of secure and efficient application delivery [online], [dostęp: 15 czerwca 2015] dostępny w internecie: <https://blog.akana.com/rest-and-the-promise-of-secure-and-efficient-application-delivery/>.
- [8] Creating an efficient rest api with http [online], [dostęp: 15 czerwca 2015] dostępny w internecie: <http://mark-kirby.co.uk/2013/creating-a-true-rest-api/>.
- [9] How to handle challenges with api security and efficiency [online], [dostęp: 15 czerwca 2015] dostępny w internecie: <http://searchsoa.techtarget.com/feature/How-to-handle-challenges-with-API-security-and-efficiency>.
- [10] Ilya Grigorik. *High Performance Browser Networking*. O'Reilly, 2013.
- [11] Walter Binder. *Designing and Implementing a Secure, Portable, and Efficient Mobile Agent Kernel: The J-SEAL2 Approach*. PhD thesis, der Technischen Universität Wien, April 2001.

Oświadczenie

Ja, niżej podpisany(a) oświadczam, iż przedłożona praca dyplomowa została wykonana przeze mnie samodzielnie, nie narusza praw autorskich, interesów prawnych i materialnych innych osób.

.....

data

.....

podpis