# Wydajność części front-endu i back-endu

Front end to część strony czy aplikacji webowej, którą użytkownik może zobaczyć lub „doświadczyć”. Ponieważ ludzi uważa się za wzrokowców, doświadczenia użytkownika są bardzo ważne, aby przyciągnąć uwagę odbiorcy.

Na „front end” składa się między innymi hipertekstowy język znaczników – HTML. Kaskadowe arkusze styli w plikach z rozszerzeniem .CSS, mające na celu „ozdobienie” wcześniej wspomnianych znaczników czy zawartych w nich informacji. Współczesnym użytkownikom często serwuje się także odrobinę „magii” w postaci skryptów języka JavaScript czy biblioteki jQuery. Wszelkie pojawiające się i znikające elementy, ruszające się przyciski czy wyskakujące okienka, można stworzyć pisząc skrypty, które później są dołączane do plików .html.

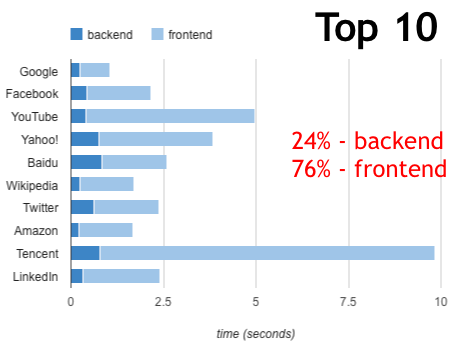
Prócz dostarczania User eXperience zadaniem front endu jest także komunikacja z back endem aplikacji przekazywanie jej interakcji użytkownika lub wyświetlanie mu informacji od serwera. Back end zazwyczaj jest dzielony na trzy części: serwer, aplikację (działającą na serwerze) oraz bazę danych. Każda z nich jest w pewien sposób uniwersalna np. gdyby aplikację (a raczej logikę aplikacji) przenieść z jednego serwera na inny oraz podpiąć do bazy danych z innymi (ale spójnymi) danymi, aplikacja też by działała. Żeby zaprogramować aplikację działającą po tzw. stronie serwera najczęściej korzysta się z takich języków programowania jak: PHP, Java, Ruby czy Python.

Wybór konkretnej technologii to bardzo ważny element tworzenia aplikacji webowej, jednak dyskusja na temat, który z nich jest najszybszy, najwydajniejszy, czy którego kod jest najłatwiejszy w utrzymaniu może pochłonąć kilka lat. Przede wszystkim należy dobrać odpowiednio język do tworzonej aplikacji. Dla wszystkich jednak można korzystać ze wspólnych koncepcji backendowych, wymienionych między innymi w opisanym projekcie „Snail Project”. Sugestie zawarte w wynikach stanowią uniwersalne rozwiązania, które każdy programista może zastosować w swoim projekcie aby jego aplikacja działała bardziej wydajnie.

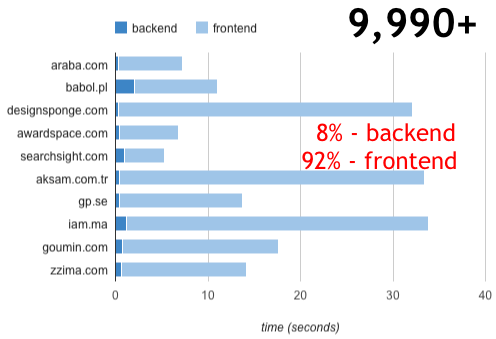
Działania części określanej jako front end oraz back end mimo różnych zadań są ze sobą połączone. Przykładowo osoba chcąca zarezerwować bilet lotniczy, po wpisaniu odpowiedniego adresu w swojej przeglądarce internetowej w pierwszej kolejności doświadczy kontaktu z wyglądem strony. Wszystkie informacje, które można od razu przeczytać na pierwszej stronie zazwyczaj nie wymagają od razu połączenia z bazą danych. Dopiero w momencie sprawdzania aktualnej informacji i ilości miejsc dla danego lotu, następuje komunikacja z serwerem, wyszukanie odpowiednich informacji w bazie danych, ewentualna filtracja ich przez kontrolery i zwrócenie ich do części wizualnej, gdzie użytkownik sam dokona wyboru czy dokonać kolejnej akcji np. rezerwacji biletu. Ciężko by było stworzyć taką aplikację gdyby nie wykorzystać bazy danych znajdującej się po stronie serwera czy nie używać arkusza styli lub skryptu pobierającego wartość maksymalnej ceny biletu podanej przez użytkownika.

Steve Souders na swoim blogu(PRZYPIS) opublikował testy dotyczące czasu jaki jest potrzebny do załadowania się strony internetowej pod względem podziału na front end oraz back end. Wyniki te dowodzą, że front end to nawet 92% czasu jaki jest potrzebny na gotowość aplikacji.

Poniżej przytoczono wyniki jakie uzyskał Steven Souders. Sprawdzian został przeprowadzony dla serwisów znajdujących się w rankingu TOP 10 najczęściej odwiedzanych stron internetowych na świecie.



Top 10 najpopularniejszych stron WWW



Ostatnie 10 stron WWW rankingu top 10000

Powyższy wykres przedstawia sytuację dla serwisów internetowych dla dziesięciu ostatnich z rankingu TOP 10 000.

Opublikowane rezultaty są jednoznaczne, back end to część pracy programisty, który w najmniejszym stopniu pływa na wydajność działania aplikacji webowej. Kiedy nie zastosuje się rad, które sugeruje wykonany i opisany „Snail project” nawet najwydajniejszy na świecie serwer nie poradzi sobie z odpowiedziami na „niepotrzebne” żądania czy przesyłaniem do klienta „ciężkich” plików graficznych lub niezoptymalizowanych skryptów blokujących ładowanie się pozostałej części strony.

http://www.stevesouders.com/blog/2012/02/10/the-performance-golden-rule/