Komunikacja czasu rzeczywistego w PHP - czyli co z tymi WebSocketami

Abstrakt

Ilu z Was próbowało stworzyć choćby prosty czat w AJAX i PHP? Czy powalało to wydajnością? Co z poważniejszymi zastosowaniami, gdzie szybkość reakcji jest kluczowa? W trakcie prelekcji omówię te bardziej (PHP, NodeJS) i te trochę mniej (erlang) konwencjonalne rozwiązania pozwalające na skomunikowanie przeglądarki z Aplikacją przy jak najmniejszych opóźnieniach.

Porozmawiamy o wąskich gardłach, skalowaniu takich rozwiązań oraz ich utrzymaniu.

Ramowy plan prezentacji.

1. Teoria
    a. Omówienie problematyki komunikacji w czasie rzeczywistym w przeglądarce internetowej – rozwój technologii na przestrzeni lat.
    b. Jak działa protokół – krótki elementarz
        i. Budowa protokołu
        ii. Debugowanie komunikacji
    c. Prawdopodobne problemy
2. Wykorzystanie w praktyce – przytoczenie obecnie działających rozwiązań
3. Front-end
    a. Architektura przykładowej aplikacji
    b. Analiza dostępnych bibliotek pozwalających na wykorzystanie WebSocketów
    c. Progressive enhancement – w jaki sposób zadowolić jak największą ilość Gości, aby każdy z nich mógł korzystać z pełnej funkcjonalności serwisu.
4. Back-end
    a. Prosta aplikacja działająca jako prosta strona internetowa
    b. Rozbudowa o obsługę protokołu REST
5. Middleware
    a. Architektura aplikacji działającej w trybie „on-the-fly"
    b. Omówienie dostępnych technologii pozwalających na uruchomienie serwera WebStocket, m.in.
        i. Implementacja w czystym PHP
        ii. NodeJS
        iii. Erlang
        iv. Inne
    c. Porównanie powyższych ze względu na wydajność vs. wygoda vs. koszty utrzymania.
    d. Przygotowanie implementacji prostego middleware'u na przykładzie NodeJS.
6. Skalowalność aplikacji
    a. Forkowanie workerów
    b. Przykładowy projekt infrastruktury
    c. Współdzielenie sesji użytkowników
7. Pytania Publiczności

Przemysław „eRIZ" Pawliczuk
http://przemyslaw.pawliczuk.pl