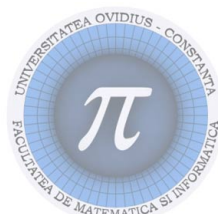


UNIVERSITATEA OVIDIUS



FACULTATEA DE MATEMATICĂ ȘI
INFORMATICĂ

LUCRARE DE LICENȚĂ

Dezvoltare aplicație web SPA bazată pe Python/Django și AngularJS

Student:

Ștefan Daniel MIHĂILĂ

Profesor îndrumător:

Lect. dr. Andrei RUSU

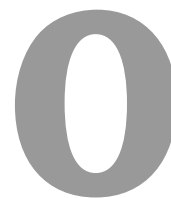
iulie 2015

Cuprins

0	Introducere	2
0.1	Modalități de dezvoltare de SPA	3
0.1.1	Framework-uri JavaScript	3
0.1.2	3

Internetul este un amalgam de tehnologii, legate împreună cu bandă adezivă, sfoară și gumă de mestecat. Nu este ceva proiectat într-un mod elegant, pentru că este un organism în creștere, nu o mașinărie construită cu intenție.

Mattias Petter Johansson (Programator la Spotify)



Introducere

Internetul a evoluat continuu și a ajuns în punctul în care poate face o mulțime de lucruri pentru care nici măcar nu a fost creat. Aproape toți programatorii din ziua de azi sunt programatori web, iar aplicațiile web seamănă tot mai mult cu aplicațiile desktop. În aceste condiții, a devenit foarte important pentru dezvoltatori să poată crea astfel de aplicații într-un mod rapid și eficient, iar uneltele pe care le au la dispoziție au fost reînnoite permanent cu altele mai bune.

Arhitectura web clasică este una client server, în care clientul (browserul) cere o pagină folosind protocolul HTTP, serverul o crează dinamic folosind un limbaj de programare server-side (C#, Java, Python, PHP, Scala etc.) și o trimite browserului pentru afișare. Prin HTTP, conexiunile sunt întotdeauna inițiate de către client, care cere pagina web.

Această arhitectură este limitată. Să ne imaginăm de exemplu că avem o pagină web care afișează în timp real scorurile unor partide de fotbal. După încărcarea paginii, server-ul nu-i poate comunica browserului că un scor s-a schimbat. Browserul va afișa scorurile neactualizate până când utilizatorul re-împrospătează pagina.

Această problemă a fost rezolvată prin intermediul AJAX¹, o tehnică ce permite browserului să facă cereri asincrone către server după ce pagina a fost încărcată, prin intermediul JavaScript.

Următoarea etapă în acest proces incremental a fost SPA - Single-Page Application. Într-un SPA, tot codul HTML, JavaScript și CSS este fie descărcat în momentul în care pagina este încărcată prima dată, fie în mod asincron, de obicei ca răspuns la acțiunile utilizatorului. SPA oferă utilizatorului senzația unei aplicații fluide și poate uneori să ofere iluzia că aceasta răspunde la acțiuni

¹Asynchronous JavaScript and XML; în aplicațiile moderne se utilizează cu preferință JSON (JavaScript Object Notation) în loc de XML, dar denumirea a rămas.

imediat, fără să mai aștepte răspunsul serverului. Vom vedea în aplicația construită pentru această lucrare, de exemplu, că atunci când utilizatorul dorește ștergerea unei resurse, această resursă este întâi înlăturată din UI, apoi o cerere asincronă îi spune serverului să șteargă resursa din baza de date. Desigur, pentru că se comunică cu serverul prin TCP/IP, această comunicare poate eșua, caz în care un mesaj de eroare este afișat și resursa re apare în UI, dar în mai mult de 90% din cazuri, când totul merge bine, utilizatorul are senzația că resursa este ștearsă instant.

0.1 Modalități de dezvoltare de SPA

0.1.1 Framework-uri JavaScript

Anumite framework-uri JavaScript pentru crearea de aplicații web, cum ar fi AngularJS², Ember.js³, React⁴ și Backbone.js⁵, și-au propus să ușureze dezvoltarea de aplicații web SPA.

Printre avantajele folosirii unui framework unui astfel de framework, enumerăm:

- Aceste framework-uri oferă de obicei și posibilitatea organizării codului folosind MVC⁶. Într-un framework JavaScript MVC, view-ul sunt reprezentate de șabloane HTML, controller-ul este un obiect JS care se ocupă de comunicarea dintre view și model, iar modelul este un obiect JS care, de obicei, mapează obiectele din baza de date de pe server la obiecte stocate în client. Un proiect MVC este mai ușor de navigat, este mai ușor de modificat și mai ușor de înțeles. De asemenea, colaborarea dintre designer și programator este mai ușoară cu un framework MVC.
- Viteza de dezvoltare.
- Ușurința de dezvoltare.
- Diminuarea codului necesar a fi scris.

Bineînțeles, există și dezavantaje:

- Necesitatea învățării unei tehnologii noi. Ce este și mai trist, este că foarte posibil această tehnologie va fi depășită în doar câțiva ani.
- Unele framework-uri (AngularJS) au o curbă de învățare abruptă.

0.1.2

²<https://angularjs.org/>

³<http://emberjs.com/>

⁴<http://facebook.github.io/react/>

⁵<http://backbonejs.org/>

⁶Model View Controller