



FACULTATEA DE ELECTRONICĂ, TELECOMUNICAȚII
ȘI TEHNOLOGIA INFORMAȚIEI

Documentare proiect Java

Andrei Mihăescu

Capitolul 1

Descriere aplicație

Această aplicație reprezintă scheletul unei platforme de video la cerere (eng. Video on Demand) care va constitui lucrarea mea de dizertație. Aplicația se află în fază incipientă și reprezintă mai degrabă un "proof of concept" și o validare a faptului că un astfel de proiect este fezabil în cadrul unei lucrări de dizertație, folosind tehnologii open-source.

Parțea cheie a unei astfel de aplicații o reprezintă transmiterea fluxurilor video către clienți. Așadar o prima problemă de rezolvat a fost găsirea unei soluții în acest sens. În urma consultării site-urilor de specialitate și având în minte dorința de a crea o aplicație modernă, care să utilizeze ultimele tehnologii, am ales să folosesc o arhitectură client-server pentru a realiza o aplicație web.

Motivul pentru care am optat pentru o aplicație web este că pentru a avea un astfel de produs de succes, o platformă de video la cerere trebuie să deservească un număr cât mai mare de clienți. Pe baza acestor considerente alegerea aplicației web a fost una foarte simplă, deoarece majoritatea terminalelor dispun de un browser. În plus, tehnologiile recente permit crearea de pagini web de tip "responsive" capabile să se adapteze la rezoluția dispozitivului pe care este afișată pagina și deci experiența unui utilizator care folosește un dispozitiv mobil să fie la fel de plăcută ca cea a unui care folosește un terminal fix.

Soluția aleasă pentru a crea o aplicație modernă a fost să creez un serviciu web de tip REST care să fie consumat de o parte front-end realizată în HTML5. Utilizând HTML5 și mai exact redarea unui clip video se face foarte simplu tag-ul dedicat `<video>`. Acesta permite specificarea căi ce va conține fișierul video. Această cale urmează să fie cea pe care o va expune serviciul web.

Partea server a aplicației așa cum am spus este constituită dintr-un serviciu web de tip REST. Alegerea folosirii acestui tip de serviciu se datorează în principiu simplității cu care acestea se pot realiza folosind framework-uri. Un alt motiv este de asemenea și ușurința cu care acestea se pot testa, fiind disponibile foarte multe unelte gratuite. Serviciile REST sunt apelabile prin intermediul link-urilor de tip `http://hostname:port/path/to/resource`.

O altă parte la fel de importantă pentru acest tip de aplicație este aspectul securitate, deoarece nu se dorește accesul utilizatorilor neautorizați la conținutul video. Securitatea este asigurată de un filtru web care scanează toate cererile care vin către server și verifică dacă acestea sosesc de la un utilizator autentificat.

Capitolul 2

Tehnologii folosite

În materie de tehnologii folosite aplicația le utilizează pe cele mai recente, atât pe parte de client, cât și pe cea de server.

Partea de client, după cum am menționat și mai sus este scrisă în HTML5 și CSS3 și folosește javascript pentru a adăuga o componentă dinamică paginilor. Însă aceste limbaje nu sunt folosite pentru ca atare ci prin intermediul librăriilor deja dezvoltate.

Una din caracteristicile moderne ale acestei aplicații este faptul că este una de tip single-page, ceea ce înseamnă ca întreaga aplicație este de fapt o singură pagină al cărei conținut este schimbat în mod dinamic. Acest fapt face ca aceasta să fie mult mai puțin consumatoare de bandă deoarece la o acțiune a utilizatorului ceea ce este transmis de la server nu sunt decât schimbările necesare și nu întreaga pagină cum este cazul multora aplicații vechi.

Această funcționalitate este realizată cu ajutorul librăriei javascript AngularJS. Acesta permite compoziția paginilor web și astfel la acțiunile utilizatorului AngularJS dispune de un ruter de URL-uri care va intercepta aceste acțiuni și va încărca pagină specificată pentru URL-ul respectiv. Unul din altele funcționalități pentru care am preferat să folosesc această bibliotecă a fost datorită "data binding" ceea ce permite o legare foarte simplă a elementelor ce permit introducerea de date de către utilizator de variabile javascript care pot fi mai departe transmise către serviciul web.

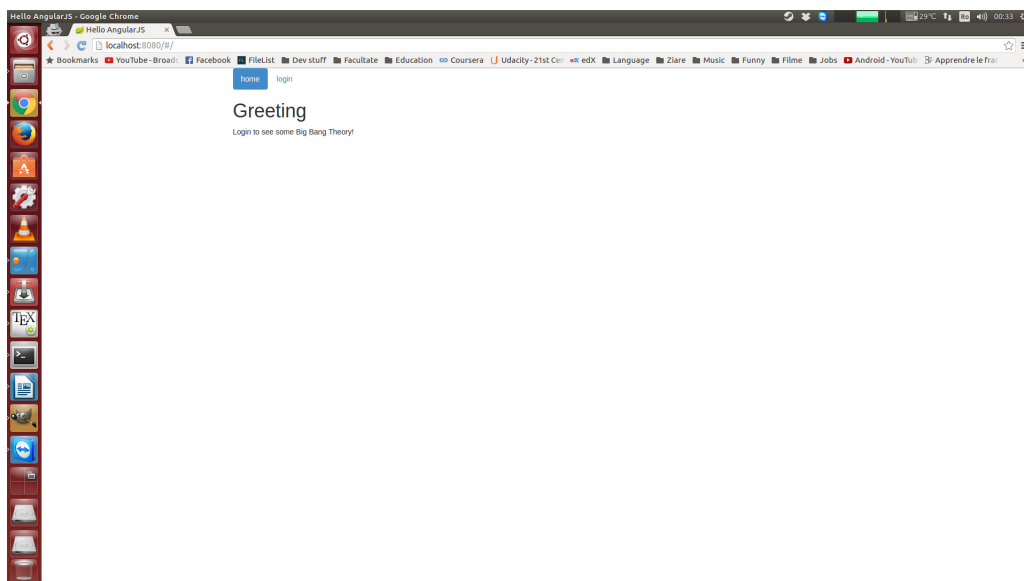
Pe partea de client am folosit librăriile puse la dispoziție de Spring și prin intermediul anotărilor am reușit să creez foarte simplu un serviciu REST. Pentru rularea aplicației am ales să nu folosesc un server separat așa cum este cazul majorității aplicațiilor, ci unul integrat în aplicație. Aceasta este de dorit în defavoarea folosirii unui server dedicat deoarece elimină problemele legate de compatibilitate ce pot apărea între aplicație și mediile pe care aceasta rulează. Folosirea unui server integrat în aplicație presupune de asemenea și unele particularități în materie de configurarea. Aceasta se va face prin intermediul unei clase de configurare și nu printr-un fisier, din moment ce întreaga aplicație va fi livrată ca o arhivă de tip jar.

Folosirea unui astfel de server este foarte mult facilitată de folosirea librăriilor Spring, deoarece acestea oferă o soluție numită Springboot care se bazează pe Apache Tomcat, un server foarte des folosit pentru rularea aplicațiilor web create în Java. Pentru obținerea acestor librării am folosit **maven**, o unealtă care permite gestiunea dependențelor unui proiect Java, dar care și gestionează aspecte cum ar fi compilare și împachetarea arhivelor. În plus Spring pune la dispoziție librăriile prin intermediul maven și include chiar pachete complete legate de anumite arii tipice ale unei aplicații cum ar fi : web, securitate, acces la o bază de date.

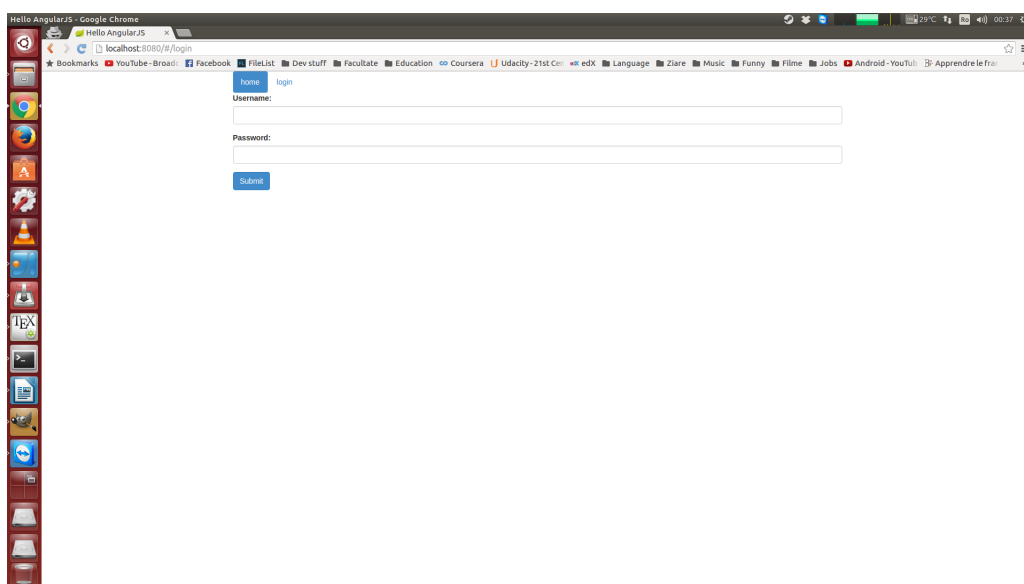
Capitolul 3

Funcționarea aplicației

Aplicația este accesibilă la următoarea adresă `http://localhost:8080`. Utilizatorul accesând acest URL va ajunge pe pagina de mai jos:

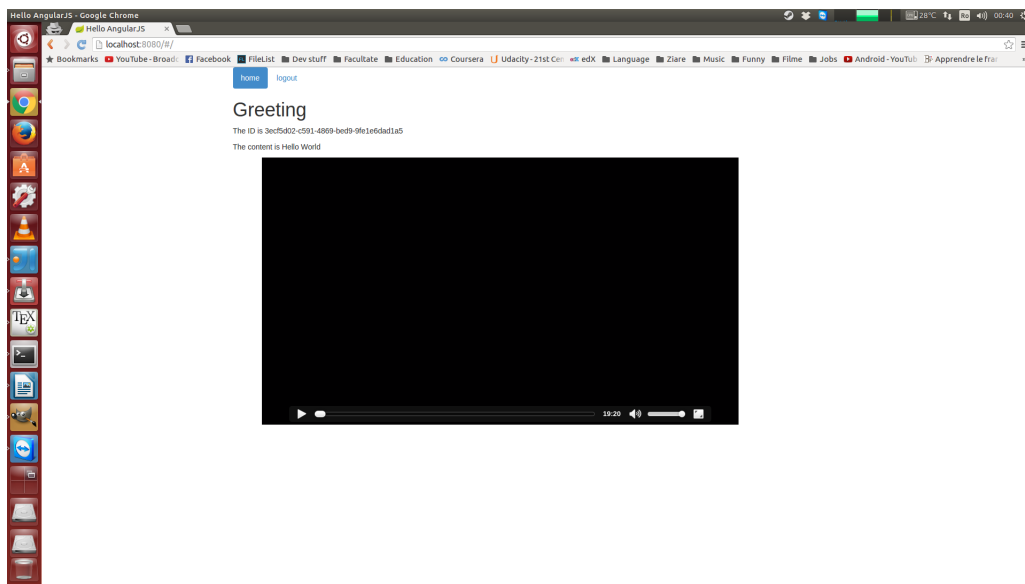


Apăsând pe butonul de login se observă faptul de URL-ul este schimbat și că ruterul se URL al librăriei AngularJS a interceptat schimbarea și a schimbat conținutul paginii.



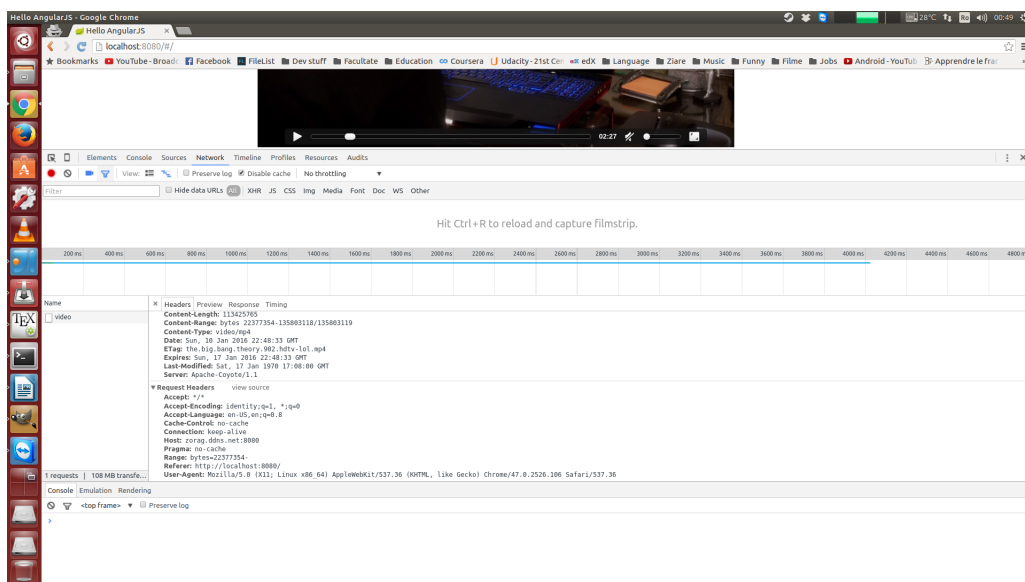
În momentul de față singurul utilizator existent este unul generic denumit **user** și care face parte din setările implicite pe care le oferă pachetul de librării springboot-security, care a fost folosit pentru realizarea părții de securitate.

După autentificare se poate observa că butonul "Login" a dispărut și că în locul acestuia a apărut "Logout". Mai jos se poate vedea chenarul în care va fi redat clipul video.



La click pe butonul play videoclipul va începe rularea.

Una din problemele care a necesitat un pic de investigare a fost faptul că în starea actuală nu era permis saltul la alt momente al clipului. Altfel spus utilizatorul nu putea să deruleze videoclipul acolo unde dorea. Aceasta se datora faptului că octeții erau transmiși secvențial, iar în momentul saltului acestia nu erau disponibili. Această problemă a fost remediată analizând cererile trimise de browser în momentul saltului și folosirea acelor informații pentru trimiterea octeților de informație utili.



După cum se poate vedea în screenshot-ul de mai sus browser-ul lansează o cerere către server, iar printre antetele transmise se numără și unul **range**.

Capitolul 4

Concluzii

În urma acestui experiment am putut constata că tehnologiile disponibile oferă posibilitatea dezvoltării facile a unei asemenea aplicații prin intermediul folosirii librăriilor. Deasemenea alegerea unei soluții adecvate este foarte importantă pentru că mai apoi eforturile dezvoltare să meargă în direcția bună. În cazul de față consider că am optat pentru o soluție scalabilă deoarece întreaga aplicație va fi instalabilă pe orice mediu care dispune de versiune adecvată de Java, fără niciun altfel de cerințe suplimentare.