Universitatea din București

Facultatea de Matematic**ă** și Informatic**ă**

LUCRARE DE LICENȚĂ

Cassandra Interface – Aplicație Web

COORDONATOR STIINTIFIC STUDENT

Conf. dr. Alin Ștefănescu Lina Luca-Cristian

Bucuresti 2019

Curpins

[Introducere 3](#_Toc10576701)

[Tehnologiile si arhitecturile folosite 5](#_Toc10576702)

[Apache Cassandra 3.11.3 5](#_Toc10576703)

[Java 8 8](#_Toc10576704)

[Spring, Spring Boot 2.0.5 si Spring Security 9](#_Toc10576705)

[Thymeleaf 14](#_Toc10576706)

[HTML si HTML5 16](#_Toc10576707)

[CSS si Bootstrap 17](#_Toc10576708)

[Javascript si Jquery 17](#_Toc10576709)

[Prezentarea aplicatiei 17](#_Toc10576710)

[Concluzii 17](#_Toc10576711)

[Lista figuri 17](#_Toc10576712)

[Bibliografie 17](#_Toc10576713)

Introducere

Ideea aplicatiei pleaca de la tehnologia Apache Cassandra, o baza de date NoSQL (not only SQL) destinata gestionarii cantitatilor mari de date. Din pacate interactionarea cu aceasta baza de date de catre un programator nu este foarte complexa.

Folosind limbajul CQL (Cassandra Query Language), un programator poate prelua data sau executa comenzi asupra bazei de date. Din pacate, din punct de vedere vizual si din punct de vedere al eficientei, aceasta procedura nu este una foarte benefica. Interfata consta intr-o consola care primeste comenzile CQL si returneaza rezultatul. Astfel nu exista nicio modalitate de a avea o privire de ansamblu a bazei de date fara a trece prin mai multi pasi inevitabil.

Am ales sa dezvolt aceasta aplicatie in special pentru acest motiv, de a eficientiza gestionarea acestei baze de date printr-o interfata prietenoasa, eficienta si accesibila oricui (de aici si numele aplicatiei).

Am ales ca acesta aplicatie sa fie web in primul rand pentru usurinta si accesibilitatea pe care o ofera interactiunii si comunicarii intre membri. In al doilea rand, pentru a introduce aplicatia mai usor in mediul online, care dupa parerea mea este cel mai eficient mod de a promova o creatie, mai ales aplicatii in genul acesteia.

Avand ca exemplu negativ interfata curenta a bazei de date, a fost usor de imaginat o aplicatie eficienta care sa-i ocupe locul. Astfel, comparatia dintre cele doua interfete poate fi destul de semnificativa.

\**De acum inainte voi folosi termenul de “interfata veche” si “interfata noua” pentru a face referire la interfata curenta a bazei de date Cassandra, reprezentata de consola de tip terminal, respectiv aplicatia(interfata) prezentata, Cassandra Interface.*

Spre deosebire de interfata veche, care putea fi accesata numai prin query-uri, interfata noua ofera atat facilitatea de a executa query-uri de tip text, cat si accesarea/executarea datelor prin doar cateva click-uri. Acesta contine cate o pagina speciala pentru acestea: “View/Edit”, “Console/Script”, prin care utilizatorul poate face (intr-un procent destul de mare) cam tot ceea ce poate face si interfata veche, insa mult mai usor.

Alta diferenta seminificativa, este aceea ca interfata veche nu ofera siguranta si comunicarea eficienta intre doi sau mai multi utilizatori care au acces la aceleasi date. Cassandra Interface ofera aceasta facilitate prin simplul fapt ca orice schimbare asupra bazei de date este salvata in Log si se poate accesa de catre membrii cu rol superior in orice moment. De asemenea orice schimbare de update sau delete asupra bazei de date creeaza automat un backup. Pe langa aceste lucruri, utilizatorii cu acces la aceleasi date, pot avea roluri si acces diferit asupra lor, facand totul sa se desfasoare intr-un mediu sigur. Astfel, siguranta datelor in participarea colectiva este asigurata.

Principalul scop in dezvoltarea acestei aplicatii fiind eficienizarea din punct de vedere vizual si temporal, Cassandra Interface ofera acest lucru prin fiecare componenta a aplicatiei. Fiecare pagina in parte ofera ceva aparte utilizatorului, care poate beneficia ce acel lucru intr-un timp cat mai scurt.

Pentru a putea beneficia de calitate in dezvoltarea acestei aplicatii atat din punct de vedere al programatorului cat si din perspectiva unui utilizator, am folosit tehnologii complexe in acest sens.

Din motive evidente, baza de date folosita pentru aplicatie este Apache Cassandra, impreuna cu limbajul CQL.

Ca limbaj principal de programare pentru backend am folosit Java 8, fiind unul dintre cele mai bune tehnologii in dezvoltarea aplicatiilor web. De asemenea, este si o preferinta subiectiva. Impreuna cu acesta, am folosit framework-ul Spring Boot integrat pentru Java, oferind o usurinta si o calitate mai mare in dezvoltarea aplicatiei. Pentru o securitate eficace am folosit o tehnologie derviata din Spring si anume Spring Security.

Comunicarea intre backend si frontend este realizata prin tehnologia Thymeleaf, o librarie din Java, care aplica transformari si proceseaza template-urile afisand datele produse de aplicatie. De asemenea, a fost posibila integrarea cu Spring Security si crearea template-urilor de tip layout prin adaugarea unor dialecte la aceasta librarie.

Pentru frontend, fiind o aplicatie web, am folosit HTML5, impreuna cu elementele de design alte CSS-ului si Bootstrap-ului si crearea de scripturi in Javascript si Jquery.

In continuare urmeaza descrierea in amanunt a tehnologiilor folosite si a aplicatiei.

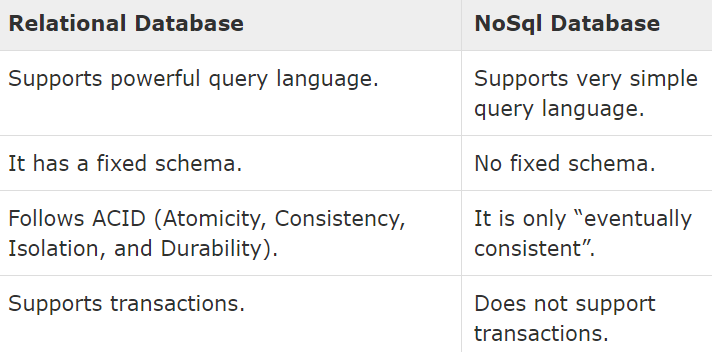
Tehnologiile si arhitecturile folosite

Apache Cassandra 3.11.3

Apache Cassandra este o baza de date “open-source”, distribuita (baza de date in care nu toate dispozitivele de stocare sunt atasate la acelasi procesor, fiind posibila stocarea pe mai multe calculatoare fie locale, fie dispersate intr-o retea de calculatoare interconectate [15]), non-relationara (NoSQL) de tip “wide column store”, putand fi interpretata ca o depozitare cheie-valoare in doua dimensiuni [16], creata pentru putea face fata unui numar foarte mare de date [3].

Avinash Lakshman si Prashant Malik au dezolvtat initial Cassandra la Facebook pentru a imbunatati motorul de cautare, apoi Facebook a lansat oficial Cassandra ca proiect “open source” la codul Google in iulie 2008 [3].

O baza de date NoSQL prezinta un mod de a depozita si recupera date diferit fata de mecanismul traditional al bazelor de date relationare, avand avantajul procesarii unui numar foarte mare de date mult mai eficient [17].



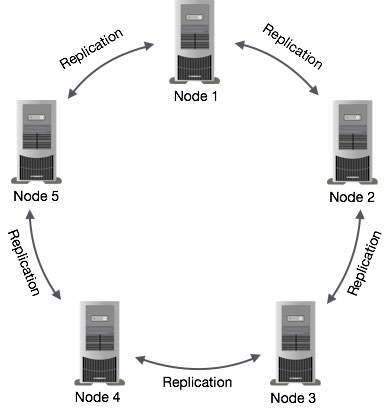
[17]

Cassandra a devenit populara datorita numeroaselor caracteristici de care dispune, cum ar fi capacitatea foarte mare de scalabilitate, avand posibilitatea de adaugare de echipament pentru mai multe date. Pe langa aceasta, Apache Cassandra este facuta in asa fel incat nu poate avea niciun punct de eroare, fiind disponibila in orice moment. De asemenea, cu cat baza de date este mai complexa in arhitectura cu atat viteza este mai buna.

Spre deosebire de alte baze de date NoSQL, Cassandra a fost dezvoltata pentru “fast writes”, fiind capabila sa stocheze sute de tera de date fara a ingreuna citirea acestora. Un alt atribut este si flexibilitatea tipruilor de date, putand depozita toate formatele de tipuri de date, incluzand cele structurate si semi-structurate.

Arhitectura consta intr-un sistem “peer-to-peer” distribuit de-a lungul mai multor noduri (in care sunt distribuite la randul lor datele), aceste noduri apartinand de un cluster (grup). Un cluster contine unul sau mai multe centre de date (un centru de date este format din mai multe noduri). Fiecare nod dintr-un cluster are acelasi rol si este independent si in acelasi timp interconectat cu celelalte noduri. Atributul principal este ca fiecare nod poate accepta cereri de scriere/citire, iar daca un nod pateste ceva, aceste cereri pot fi satisfacute de celealte noduri.

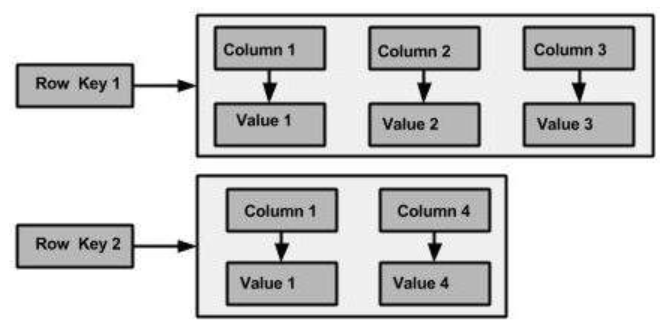
Cassandra lucreaza prin replicarea datelor, astfel unul sau mai multe noduri dintr-un cluster contine replicari pentru o anumita bucata din datele din acel cluster. Fiecare cerere de scriere este captata de un log din fiecare nod, apoi detele sunt trimite la “mem-table”, un tabel de memorie unde sunt trimise temporar datele, inainte de a fi partitionate si replicate in cluster. Daca “mem-table” este plin, datele se vor stoca de asemenea temporar in “SStable”, un fisier pe disk, pe care Cassandra il va interoga periodic pentru a elimina datele inutile. Imaginea de mai jos arata o schema de noduri intr-un cluster si cum functioneaza replicarea astfel incat sa nu existe niciun fel de eroare [17].



[17]

Termenul de keyspace reprezinta un container pentru o lista de una mai mai multe “column families” (structura in care sunt depozitate datele – tabel). Ca si atribute, un keyspace poate fi determinat atat de un factor de replicare, care reprezinta numarul de copieri a datelor in clusterul din care face parte, cat si de strategia pentru a replica aceste date.

O familie de coloane la randul ei este un container de colectii de linii, fiecare linie continand coloane ordonate, fiecare linie avand cel putin o coloana ca facand parte din “partion key” (cheia primara). Cheia primara poate fi simpla sau compusa (din mai multe coloane) si este responsabila pentru distribuirea datelor in partitia curenta, astfel cautarea se va face implicit dupa aceasta cheie. Pentru o cautare in care nu este mentionata cheia primara, performanta va scadea senimificativ; in acest caz, pentru unele coloane care nu fac parte din cheia primara, se introduce termenul de index pe coloana respectiva, adaugandu-i anumite atribuite pentru a facilita cautarea. Pe langa cheia primara mai exista si coloane “clustering key”, care sunt reponsabile pentru sortarea datelor in partitie. Figura de mai jos reprezinta schema unei familii de coloane [17].



[17]

Apache Cassandra ofera un limbaj ca suport sub forma unui terminal pentru comunicarea intre user si baza de date, si anume CQL (Cassandra Query Language), avand posibilitatea de a crea keyspace-uri, insera date sau executa diferite query-uri [17]. Atat sintaxa limbajului cat si tipurile de date seamana destul de mult cu cea de la bazele de date relationare cum ar fi SQL, astfel exista comenzi asemanatoare pentru: selectare, creare, alterare, stergere, inserare, actualizare, etc. Pe langa tipurile de date cunoscute cum ar fi: text, int, boolean, date, etc, Cassandra introduce termeunul de “frozen”, un tip de data care accepta atat tipuri create de utilizator (UDT – user data type), cat si colectii cunoscute ca liste, seturi si harti. Ca si in alte baze de date, aceasta ofera posibilitatea crearii de functii si triggere asupra tabelelor. Mai mult decat atat, cautarea fiind restrictionata de cheia primara, se poate crea un “materialized view”, adica un alt tabel conectat cu tabelul de baza, in care pe langa cheia primara inititala, se adauga si alte coloane la aceasta, facand posibila cautarea dupa acestea (fara a altera tabelul initial, iar modificarile sunt sincronizate). Mai jos sunt prezentate cateva exemple de sintaxa pentru acest limbaj.

CREATE KEYSPACE IF NOT EXISTS admin WITH REPLICATION = { ‘class’ : ‘SimpleStrategy’, ‘replication\_factor’ : 3 } AND DURABLE\_WRITES = true;

CREATE TABLE IF NOT EXISTS admin.users (user\_id text PRIMARY KEY, age int, name frozen<fullname>, data list<text>);

CREATE TYPE admin.fullname (first\_name text, last\_name text);

INSERT INTO admin.users (id,age) values (‘1’,21);

CREATE MATERIALIZED VIEW IF NOT EXISTS admin.users\_by\_age

AS SELECT \* FROM admin.users

WHERE user\_id IS NOT NULL AND age IS NOT NULL

PRIMARY KEY (age, user\_id);

SELECT (age,user\_id) FROM admin.users\_by\_age;

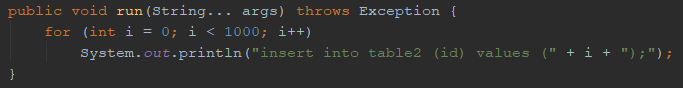
Motivul pentru care am ales aceasta baza de date este prezentat in Introducere.

Java 8

Java este un limbaj de programare orientat pe obiecte, creat de catre James Gosling la inceputul anilor ’90, fiind lansat in 1995. Este un limbaj cu o intindere larga, fiind folosit in numeroase aplicatii distribuite, incluzand cele mobile.

In sintaxa limbajului se regasesc o mare parte din elemente de C sau C++, dar modelul obiectelor este mai simplu si are mai putine facilitati de nivel jos. Ca de exemplu, Java nu are supraincarcare operatorului, iar mostenirea multipla este valabila doar pentru interfete nu si pentru clase. Avantajul major este ca un program Java corect si compilat, poate fi rulat fara modificari pe orice platforma care are instalata o masina virtuala Java. Acest lucru este posibil deoarece sursele sunt compilate intr-un format numit cod de octeti, acesta fiind independent de tipul platformei.

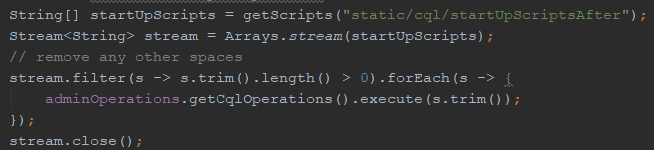
Pentru a demonstra aceste referinte, urmeaza o imagine sustrasa din proiect, care contine un bloc de cod care se apeleaza la pornirea serverului afiseaza in consola acestuia (System.out.println) 1000 de linii (for de la 0 la 999), fiecare continand un query de inserare pentru baza de date. Acest bloc a fost folosit pentru testare si nu are un rol functional in aplicatie.



Dintre cele 4 platforme furnizate de Oracle (Java Card, Java ME – micro edition, SE – standard edition si EE – Enterprise Edition, aplicatia Cassandra Interface foloseste versiunea 8 a platformei standard, care este destinanta sistemelor de tip workstation [4].

Java SE 8 a aparut pe 18 martie 2014 si este considerata cea mai mare actualizare, modificarile asigurand faptul ca Java ramane o prioritate la nivelul aplicatilor de intreprindere. Spre deosebire de celelalte versiuni, acesta aduce imbunatatiri seminificative.

Printre acestea se numara expresiile lamda, care sunt considerate cele mai mari caracteristici in Java 8. Aceste expresii actioneaza ca o functie anonima, intr-o bucata de cod foarte scurta, ajutand astfel si la coding style. De asemenea, tot in acesta versiune au fost introduse stream-urile, o alta actualizare majora, care vine cu diferite tipuri de operatiuni de flux (filtrare, sortare, etc), folosite in mod eficient impreuna cu expresiile lamda [5]. Urmeaza o imagine din aplicatie care contine un astfel de exemplu, care pentru un vector de stringuri, il filtreaza, eliminand stringurile vide, iar pentru fiecare dintre acestea, executa o anumita comanda.



Am ales acest limbaj in primul rand pentru performanta pe care o ofera atat ca viteza si usurinta de scriere, dar si pentru calitatea dovedita de-a lungul anilor, fiind platforma pentru milioane de dispozitive. In al doilea rand pentru diversitatea de care dispune acest limbaj, avand posibilitatea de integrare cu aproape orice domeniu.

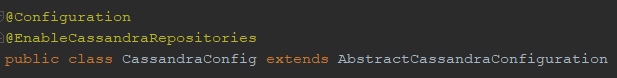
Spring, Spring Boot 2.0.5 si Spring Security

Spring este un framework de aplicatii pentru platforma Java, iar prima versiune a sa aparut pe 1 octombrie 2002 si a fost scrisa de Rob Johnson. Framework-ul poate fi folosit de orice aplicatie Java, insa principalul sau atribut faciliteaza dezvoltarea aplicatiilor web.

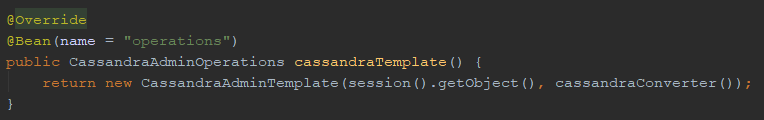
Specific pentru Spring este container-ul IoC (inversion of control), care produce configuratii si se ocupa de obiectele din Java cu ajutorul reflexiei (abilitatea unui proces de a-si examina si modifica propria structura si comportament [8]). Container-ul se ocupa de organizarea duratei de viata (creare, apelare, configurare, etc) a obiectelor care se mai numesc “beans” si poate fi configurat cu ajutorul unui fisier XML (Extensible Markup Language) sau prin detectarea unor adnotari specifice asupra claselor de configurare. Aceste obiecte la randul lor pot fi obtinute printr-o cerere catre container pentru un obiect cu un anumit nume sau de un anumit tip (dependency lookup) sau prin-o metoda care trimite obiectul dupa nume intr-un alt obiect (dependency injection) [6]. Imaginea de mai jos prezinta a doua metoda, in care obiectul (bean-ul) UserService creat si stocat in contextul Spring, este “injectat” intr-un obiect intr-o clasa cu ajutorul adnotarii @Autowired.



Spring are propriul AOP (aspect-oriented programming), care modularizeaza relatiile de tip cross-cutting in aspecte. Cu alte cuvinte, spre deosebire de modulele traditionale care incapsuleaza obiectele si contin codul necesar pentru ca acestea sa implementeze serciviile sitemului, modulele AOP contin codul necesar implementarii unui singur serviciu a sistemului, considerant toate obiectele posibile [9]. In imaginea urmatoare, se poate observa o clasa pe care framework-ul o va recunoaste ca fiind una de configurare si o va manipula in consecinta (o va instantia ca un bean) datorita adnotatiei corespunzatoare.



Alta caracteristica importanta este accesarea datelor in Spring. Acesta ofera suport pentru cele mai populare framework-uri de acces de date din Java cum ar fi Cassandra (in cazul de fata), JDBC, Hibernate, JDO, JPA, etc. De asemenea, tranzactiile au o diversitate destul de mare, de la tranzactii locale si globale pana la tranzactii in aproape toate mediile de platforme din Java [6]. Imaginea de mai jos arata crearea unui Bean pentru un obiect dintr-o clasa de configurare a bazei de date Cassandra si anume CassandraAdminTemplate, prin care se va permite executarea comenzilor si preluarea datelor asupra bazei de date.



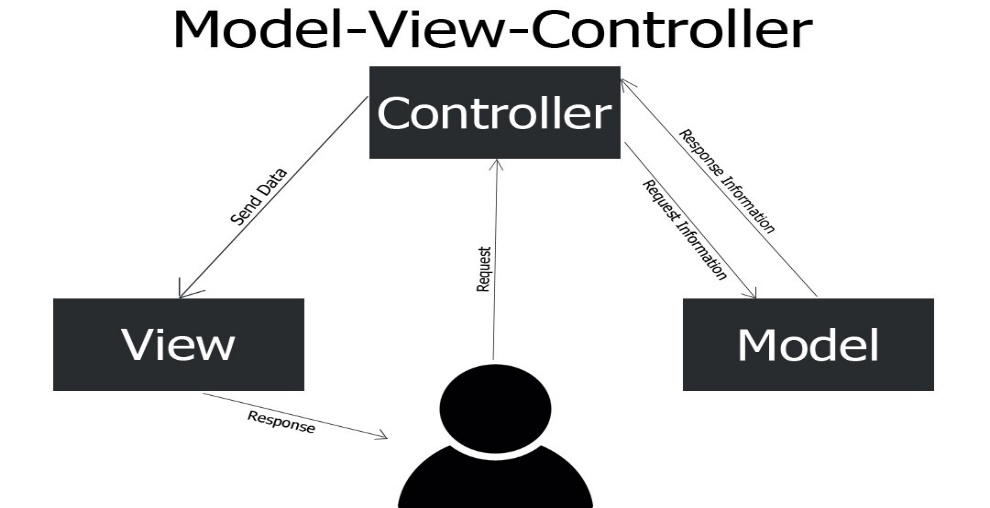
Framework-ul Spring foloseste ca arhitectura propriul MVC (Model-View-Controller) pentru aplicatii web. Aceasta arhitectura este folosita in mare parte pentru dezvoltarea interfetelor pentru utilizatori si imparte aplicatia in trei parti interconectate:

Model, componenta centrala, independenta de interfata si care se ocupa de logica si regulile cu privire la datele aplicatiei (baza de date);

View, reprezentand informatia vizuala pe care o vede utilizatorul, partea de frontend;

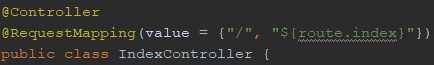
Controller, care primeste intrari si trimite mai departe catre model sau view.

Cu alte cuvinte, modelul este responsabil pentru organizarea datelor aplicatiei si primeste intrari din view cu ajutorul controllerului, view-ul reprezentand astfel prezentarea modelului intr-un format aparte, iar controllerul raspunde intrarilor din view, optional le valideaza si actioneaza in consecinta asupra modelului [10].

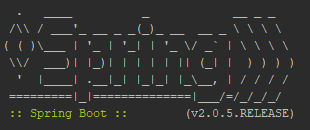


[11]

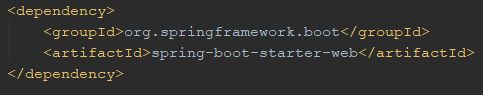
Figura urmatoare arata cum framework-ul va recunoaste, printr-o adnotare, o clasa ca fiind o componenta a arhitecturii MVC si anume controller si va trata intreaga clasa cu componentele ei ca atare.



Spring Boot reprezinta o solutie bazata pe o conventie de configuratii considerate ca fiind unele dintre cele mai bune din platforma Spring pentru crearea de aplicatii “stand alone”, aplicatii care pornesc de la o configurare minima si eficienta pentru a putea rula cat mai rapid (“just run”). Acesta ofera simplitate, configurand automat cand este cazul si folosind Project Object Models (POMs), simplificand astfel configuratiile Maven [6].



Un POM este un fisier XML care contine informatii despre proiect si despre configuratiile ce contribuie la constructia acestuia. In acest fisier pot fi incluse toate dependintele proiectului, plugin-urile, versiunile acestora si multe altele, oferind astfel atat simplitate cat si complexitate in dezvoltarea aplicatiei [13]. Imaginea urmatoare reprezinta o dependinta Maven din POM-ul proiectului care permite integrarea web cu framework-ul Spring Boot.

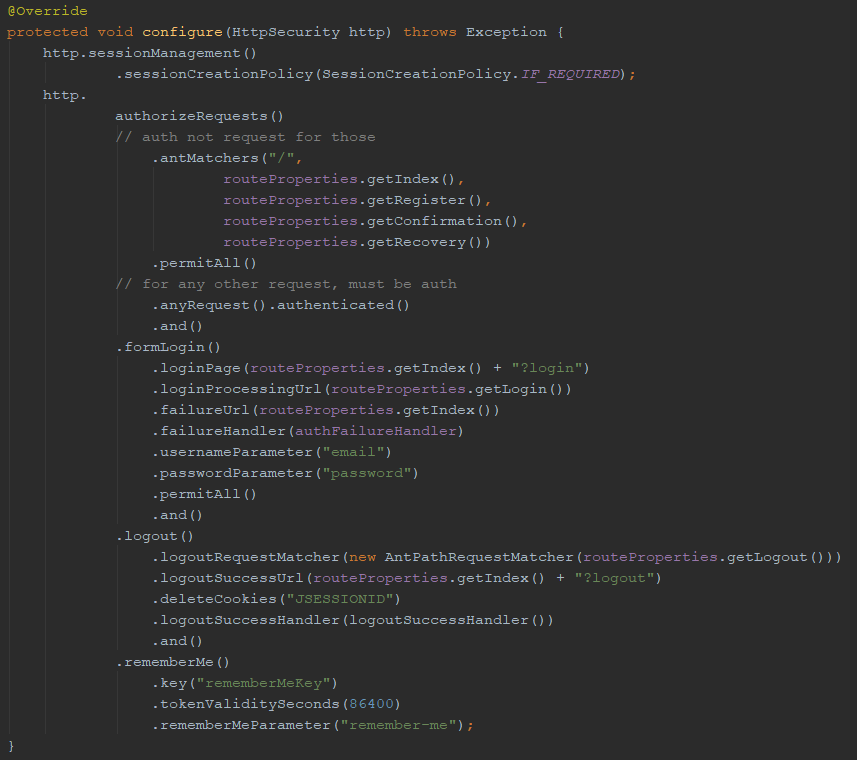


Spring Boot cuprinde implicit si server-ul pe care va rula aplicatia, si anume Tomcat. Apache Tomcat este un mediu de server web in care un program in Java poate rula, implementand implicit cateva specificatii din Java EE incluzand Java Servlet, JSP (JavaServer Page), Java EL si WebSocket [12], fiecare facand parte din componentele unei aplicatii web. Figura urmatoare arata initializarea server-ului la pornirea Spring Boot-ului.

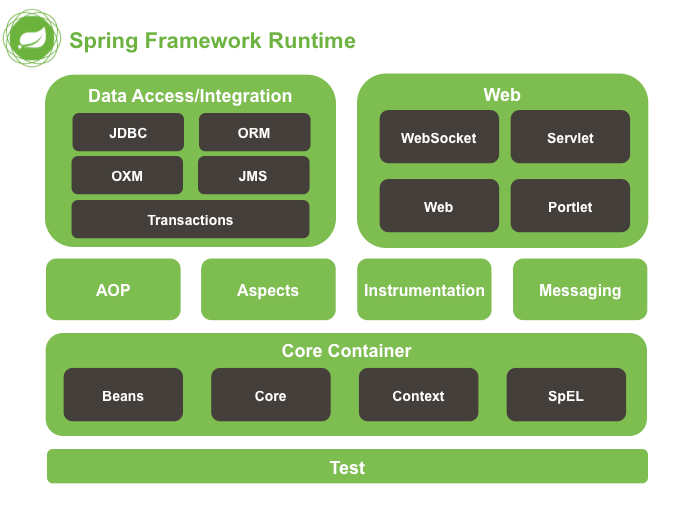


Spring Security este un framework care aduce conceptul de autentificare si autorizare pentru securizarea aplicatiilor Java. Autentficiarea reprezinta procesul de identificare a utilizatorului ce vrea sa acceseze aplicatia, iar autorizarea se ocupa de accesul in executarea actiunilor asupra ei (cereri web, metode, acces pe un anumit domeniu, etc). Alte avantaje constau in suportul oferit pentru configurarile in Java, posibilitatea de integrare cu Spring MVC, portabilitate, CSRF (Cross-site request forgery) protection (diferite atacuri asupra aplicatiei prin comenzi neautorizate provenite de la un utilizator in care aplicatia are incredere [20]), etc [19].

Imaginea de mai jos reprezinta functia de configurare a framework-ului pentru Cassandra Interface. Acesta configurare consta in restrictionarea accesului asupra paginilor web ale aplicatiei utilizatorilor care nu sunt autentificati; in acest caz, accesul lor va fi limitat la pagina de pornire si la cele care tin de autentificare. De asemenea apar si setarile subiective pentru pagina de login si logout (urls, handlers, parametrii); in plus, se configureaza si facilitatea de “remember-me”, care memoreaza user-ul la autentificare pentru un anumit interval de timp.



Am ales acest framework pentru performanta si complexitatea de care dispune, fiind unul dintre cele mai folosite framework-uri pentru aplicatii web in Java.



[14]

Thymeleaf

Thymeleaf este o librarie din Java pentru template-uri XML/XHTML/HTML5, care poate rula atat pe un mediu web cat si pe un mediu non-web, aplicand transformari asupra acestor template-uri in urma carora se pot afisa date produse de aplicatie [21], permitand astfel “comunicarea” intre backend si frontend.

Arhitectura Thymeleaf permite procesarea rapida prin parsarea fisierelor astfel incat sa se execute cat mai putine operatii.

O puternica unealta a acestei librarii o constituie extensibilitatea acestuia, permitand definirea anumitor noduri (DOM – Document Object Model), care impreuna cu alte “artefacte”, formeaza un dialect [2]. Thymeleaf ofera default dialectul Standard, insa mai pot fi adaugate si altele.

In imaginea de mai jos se intampla urmatoarele: prin aplicarea precompilarii Thymeleaf cu dialectul Standard, acest element de HTML (<div>) va fi atasat paginii, numai in cazul in care in request va exista obiectul param.login (th:if). In caz afirmativ, continutul div-ului va fi inlocuit cu mesajul corespunzator cheii “login.redirect.message” din fisierul de mesaje al aplicatiei (th:text).



Alt dialect foarte folosit este cel pentru Layout, care permite “decorarea” paginilor dupa un anumit layout, fiind foarte eficient pentru a nu duplica aceasi cod pentru fiecare pagina in parte.

Prin aceasta imagine este reprezentata sintaxa dialectului Layout care spune librariei ca acesta pagina HTML este layout pentru alte pagini.



Iar prin acesta sintaxa, Thymeleaf va stii ca acest bloc HTML este suport (fragment) pentru pagina care va “decora” acest layout. Astfel, daca intr-o pagina care foloseste acest layout exista acelasi fragment dar cu continut, acel continut va fi inlocuit in pagina de layout in fragmentul respectiv.



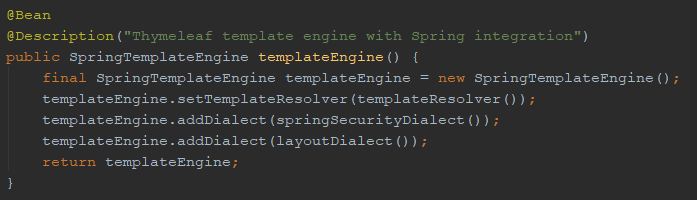
Dialectul pentru Spring Security este de asemenea folosit. Prin integrarea cu acest framework, Thymeleaf va putea avea acces la statusul legat de autentificare si autorizare a utilizatorului care acceseaza pagina. Astfel se pot impune restrictii direct asupra blocurilor de html din frontend, in functie de accesul pe care il are la momentul respectiv.

Imaginile de mai jos fac referire la sintaxa dialectului care restrictioneaza blocul respectiv de html (<li>) utilizatorilor care sunt autentificati, respectiv utilizatorului (autentificat si el) cu rol de ADMIN.





Ca ultim exemplu, in figura de mai jos este un fragment din configurarea din Cassandra Interface a librariei Thymeleaf, in care se creeaza bean-ul (integrare cu Spring) pentru motorul care va fi folosit in compilarea template-urilor, unde se adauga cele doua dialecte (cel standard este inclus automat).

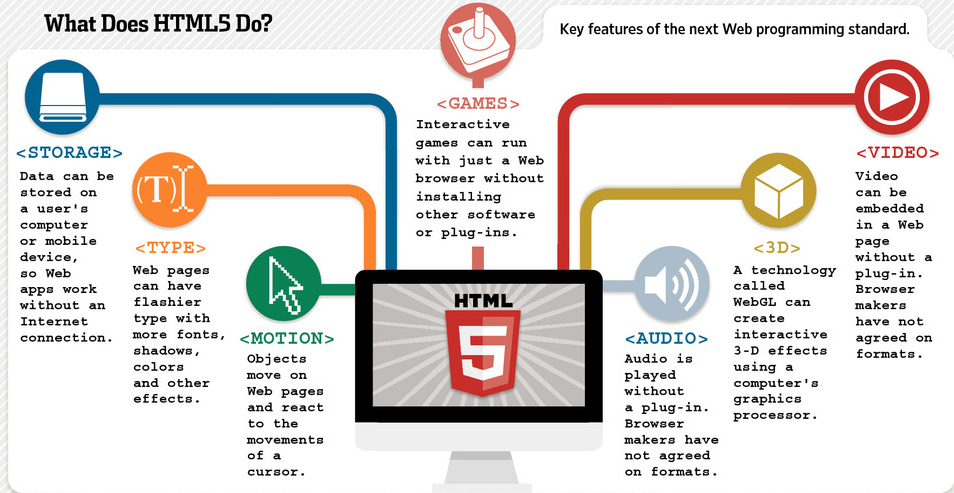


HTML si HTML5

HTML (Hype Text Markup Language) este un limbaj de marcare (metoda de formatarea unui text intr-o pagina web) folosit pentru crearea site-urilor web. Browserele web primesc documente HTML fie din spatiul local fie de la un server web si le parseaza in pagini web.

HTML creeaza documente structurate folosind blocuri pentru texte cum ar fi paragrafe, liste, link-uri, imagini si alte obiecte de tip bloc. Acestea sunt delimitate prin tag-uri ca <p> </p> pentru paragrafe sau <img> </img> pentru imagini. O pagina web este delimitata de tag-ul <html> </html>, iar continutul vizibil pentru user este delimitat de tag-ul <body> </body> [22]. Astfel, intreg documentul este o structura arborescenta, fiecare tag putand include alte tag-uri. De asemenea fiecare tag in parte poate avea informatii suplimentare (atribute) despre obiectul respectiv, informatii pe care HTML le va folosi in parsarea documentului.

HTML5 este a cinea si cea mai buna versiune a limbajului HTML, aparut pe 22 ianuarie 2008 si are ca scop imbunatatirea limbajului astfel incat sa poata avea suport pentru cele mai noi fisiere multimedia si alte caracteristici, fiind in acelasi timp citibil atat pentru user dar si pentru calculatoare. Faciliteaza si incorporarea cu diferite API-uri pentru aplicatii web mai complexe [23]. Pentru a declara documentului ca este folosit HTML se foloseste intaintea tag-ului <html>, tag-ul <!DOCTYPE html>.



[24]

CSS si Bootstrap

CSS (Cascading Style Sheet) este un limbaj de stilizare pentru prezentarea unui document scris intr-un limbaj de marcare (HTML). Numele de “cascading” vine de la puterea limbajului de a determina prioritatea regulilor (pe care sa o aleaga) in caz ca exista mai multe.

CSS-ul are o sintaxa simpla, astfel ca un fisier de stilizare (style sheet), contine o lista de reguli, fiecare regula avand unul sau mai multi selectori si un bloc de declarare. Selectorul determina carei parti din document ii este aplicata stilizarea.

Mai jos este un exemplu de sintaxa a CSS-ului, care seteaza culoarea rosie si textul bold pentru continutul tuturor paragrafelor care au clasa “paragraf” dintr-un document, “p.paragraf” fiind selectorul, iar ceea ce se afla intre acolade este blocul de reguli.

p.paragraf {

color: red;

font-weight: bold;

}

Boostrap este un framework “open source” pentru CSS. A aparut initial sub numele de Twitter Blueprint, fiind creat de catre Mark Otto si Jacon Thornton pentru platforma Twitter. Au aparut apoi mai multe versiuni, iar Boostrap 3 (cel folosit in Cassandra Interface) s-a stabilit pe 19 august 2013, urmat apoi de Boostrap 4 pe 29 octombrie 2014 in versiunea alpha si finalizat pe 18 ianuarie 2018 [26].

Boostrap este format in mare parte din o colectie de reguli CSS si optional sripturi de JavaScript, care folosite si apelate sub forma unor atribute ale elementelor dintr-un document, transforma elementele respective si le stilizeaza pe baza regulilor definite de framework.

Astfel, avand posibilitatea de a modifica numai acele atribute dupa bunul plac al utilizatorului, stilizand prin acestea documentul, Boostrap devine un framework in acelasi timp complex, dar si maleabil si usor de folosit, economisind astfel timp si cod.

Javascript si Jquery

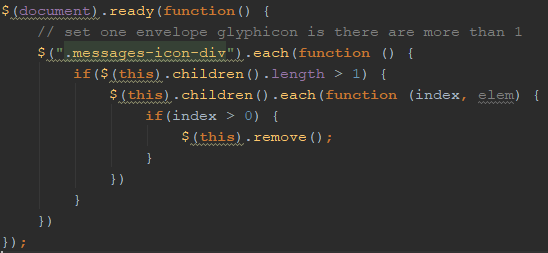
Javascript este un limbaj de programare interpretat (intructiunile sunt executate direct, fara compilare), orientat pe obiecte bazate pe prototipe (obiectele se pot refolosi). Alaturi de CSS si HTML, Javascript este unul dintre tehnologiile nucleu ale World Wide Web-ului si este o parte esentiala pentru aplicatiile web.

Chiar daca sunt unele asemanari intre Javascript si Java din punct de vedere al sintaxei si al librariilor, acesta a fost influentat in mare parte de limbaje de programare ca Self si Scheme. De asemenea serializarea JSON este bazata pe JavaScript [27].

O mare caracteristica a limbajului o consituie facilitatea de a putea comunica atat in partea de client (client-side), cat si in partea serverului (server-side), creand astfel o comunicare eficienta intre frontend si backend. O pagina web poate avea sau nu, unul sau mai multe fisiere (scripturi) javascript, fiecare rulandu-se atunci cand pagina web este incarcata. Unele dintre aceste scripturi creeaza triggere, parti de script care se vor apela atunci cand un anumit eveniment este intalnit in pagina.

Jquery este o librarie din JavaScript creata pentru a simplifica manipularea elementelor DOM din cadrul paginii HTML, a evenimentelor, animatiilor, cererilor ajax, etc, facand navigarea intr-un document mult mai usoara [28]. Astfel, scriptul scris in Jquery este mult mai scurt si mai lizibil decat unul scris in Javascript, incurajand in acest sens coduri mai complexe.

Imaginea de mai jos demonstreaza complexitatea sintaxei Jquery. Odata cu incarcarea paginii, scriptul se ruleaza (apelandu-se functia de incarcare completa a paginii ($(document).ready()), apoi prin Jquery, preiau toate elementele gasite cu care contin clasa “messages-icon-div” si pentru fiecare element gasit, daca are mai mult de un element-copil, sterg acele elemente pana ramane unul singur. Astfel scriptul elimina duplicatele din interioriul unui element prin doar cateva comenzi simple.



Prezentarea aplicatiei

Concluzii

Lista figuri

Bibliografie

[1] Apache Cassandra Documentation 4.0, <http://cassandra.apache.org/doc/latest/>

[2] Thymeleaf – Tutorial: Using Thymeleaf, <https://www.thymeleaf.org/doc/tutorials/2.1/usingthymeleaf.html>

[3] Apache Cassandra – Wikipedia, <https://en.wikipedia.org/wiki/Apache_Cassandra>

[4] Java (limbaj de programare) – Wikipedia, <https://ro.wikipedia.org/wiki/Java_(limbaj_de_programare)>

[5] Java 8 programming for beginners: Go from zero to hero, <https://www.infoworld.com/article/3130466/java-8-programming-for-beginners-go-from-zero-to-hero.html>

[6] Spring Framework – Wikipedia, <https://en.wikipedia.org/wiki/Spring_Framework>

[7] Spring Boot Refference, <https://docs.spring.io/spring-boot/docs/2.1.5.RELEASE/reference/pdf/spring-boot-reference.pdf>

[8] Reflexion (computer programming) – Wikipedia, <https://en.wikipedia.org/wiki/Reflection_(computer_programming)>

[9] Programarea orientata pe aspecte – Wikipedia, <https://ro.wikipedia.org/wiki/Programarea_orientat%C4%83_pe_aspecte>

[10] Model-view-controller – Wikipedia, <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>

[11] Girish Margan, Model-View-Controller (MVC), May 17 2018 - <https://medium.com/datadriveninvestor/model-view-controller-mvc-75bcb0103d66>

[12] Apache Tomcat – Wikipeida, <https://en.wikipedia.org/wiki/Apache_Tomcat>

[13] Maven, Apache Maven Project – Introduction to the POM, <https://maven.apache.org/guides/introduction/introduction-to-the-pom.html>

[14] Spring Framework – Introduction to the Spring Framework, <https://docs.spring.io/spring/docs/4.3.x/spring-framework-reference/html/overview.html>

[15] Distributed database – Wikipedia, <https://en.wikipedia.org/wiki/Distributed_database>

[16] Wide column store – Wikipedia, <https://en.wikipedia.org/wiki/Distributed_database>

[17] Tutorialspoint – Cassandra Tutorial, <https://www.tutorialspoint.com/cassandra>

[18] Datastax Documentation – CQL Refference, <https://docs.datastax.com/en/cql/3.3/cql/cql_reference/cqlCommandsTOC.html>

[19] Javapoint – Spring Security Tutorial, <https://www.javatpoint.com/spring-security-tutorial>

[20] Cross-site request forgery – Wikipedia, <https://en.wikipedia.org/wiki/Cross-site_request_forgery>

[21] Thymeleaf – Wikipedia, <https://en.wikipedia.org/wiki/Thymeleaf>

[22] HTML – Wikipedia, <https://en.wikipedia.org/wiki/HTML>

[23] HTML5 – Wikipedia, <https://en.wikipedia.org/wiki/HTML5>

[24] Web Techology and Trend: HTML5, <http://www-scf.usc.edu/~chenemil/itp104/webtech.html>

[25] Cascading Style Sheet – Wikipedia, <https://en.wikipedia.org/wiki/Cascading_Style_Sheets>

[26] Boostrap – Wikipedia, <https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)>

[27] JavaScript – Wikipedia, <https://en.wikipedia.org/wiki/JavaScript>

[28] Jquery – Wikipedia, <https://en.wikipedia.org/wiki/JQuery>