FrontPage Custom News

II A3

Back-end

Documentatie Modul Back-End

1. Tehnologii utilizate











->MongoDb

S-a ales acest mod de a construi si de a administra baza de date aferenta proiectului, intrucat este o baza de date NoSql si stocheaza informatiile in colectii formate din documente individuale, spre deosebire de bazele de date relationale. De asemenea, se permite crearea de noi inregistrari fara a defini initial structura bazei de date. Mai mult, se integreaza foarte bine cu Java. Astfel, lucrul cu MongoDB este mult mai usor si rapid.

->Java Spring Framework + Maven

S-a ales acest framework, intrucat ajuta la "legarea" eficienta a componentelor in cazurile in care acestea sunt foarte multe si este necesara combinarea lor in diverse modalitati, folosindu-se de implementarea pattern-design-ului Model-Controller-Viewer (MVC). De asemenea, framework-ul Spring ofera optiunea de sincronizare de task-uri (@Scheduled annotation) ce constituie un element important al proiectului, si anume programarea parserelor sa fie executate la un anumit interval de timp, preluandu-se astfel cele mai noi articole de pe site-urile sursa.

Maven ajuta la gestionarea dependintelor care sunt instalate automat.

->Jsoup Library

S-a folosit aceasta librarie pentru construirea parserelor, ajutand la extragerea si manipularea datelor din codurile sursa HTML ale site-urilor de pe care au fost preluate articolele. Este foarte usor de utilizat si flexibil, permitand ajungerea pe o cale mai usoara la rezultatul dorit.

->TranslatorAPI

S-a utilizat acest API, intrucat are posibilitatea de a traduce articole in mai multe limbi, oferind o gama larga de limbi disponibile. Nu este un API oficial, deoarece cele oficiale cereau foarte multe date personale si nu erau complet gratuite – aveam la dispozitie doar un numar restrans de caractere pentru traducerea gratuita. (S-a incercat cu API-ul de traducere de la Microsoft – Bing si de asemenea cu YandexTranslatorAPI)

Ideea la baza careia sta acest API de traducere este generarea continua de HTTP request-uri catre API-ul de traducere Google. In acest request trebuie precizat textul, limba textului, limba in care se va face traducerea si encoding-ul (UTF-8). Aceste request-uri se pot face in numar nelimitat, deoarece imita request-urile trimise de un browser ("User-Agent Mozilla/5.0"). Rezultatul este apoi parsat cu ajutorul unui **JSONArray**, care contine atat textul tradus, cat si textul de provenienta si limba.

In urma implementarii, s-a observat ca intr-un request nu se pot trimite mai mult de 2000-3000 de caractere. Din aceasta cauza, s-a implementat in metoda statica translate(String text, Language textLanguage, Language resultedLanguage) care face traducerea efectiva, o modalitate

prin care textul este impartit in paragrafe, urmand ca apoi sa se formeze request-uri cu fiecare paragraf in parte. In final, prin concatenarea paragrafelor traduse in limba precizata, va rezulta textul complet.

->Stanford CoreNLP

S-a folosit aceasta librarie pentru sumarizarea articolelor, intrucat ofera un set de instrumente pentru analiza si prelucrare a textului. Poate determina structura propozitiilor si din punct de vedere al frazelor si al cuvintelor ce depind unele de altele si poate deduce cuvintele care se refera la acelasi entitati. Este foarte flexibil si usor de utilizat, nefiind necesare prea multe linii de cod.

2. Detalii de implementare

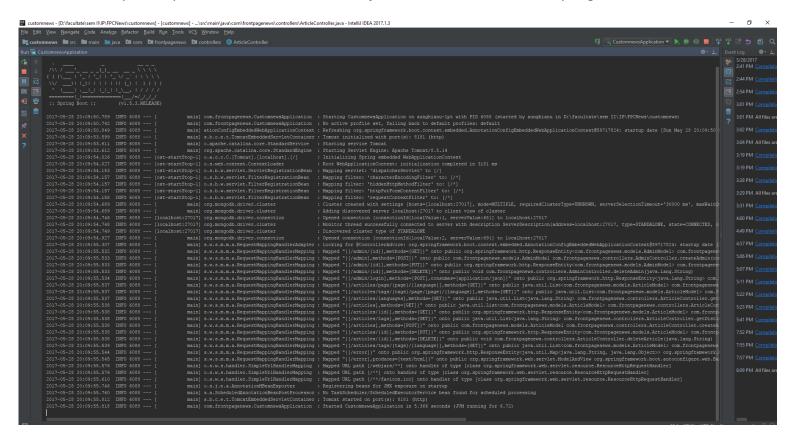
Pasii efectuati:

- ✓ S-au realizat diagramele use-case si UML pentru a stabili structura aplicatiei
- ✓ S-a scris planul de lucru
- ✓ S-a hotarat structura bazei de date
- ✓ S-au creat models, controllers, repositories si services pentru a defini caracteristicile si operatiile ce vor fi efectuate asupra articolelolor si a administratorilor
- ✓ S-au realizat parserele ce vor prelua articolele
- ✓ S-a efectuat sincronizarea executiei fiecarui parser cu ajutorul @Scheduled
- √ S-au scris primele request-uri
- ✓ S-a ales API-ul pentru traducerea articolelor, insa acesta a fost schimbat ulterior din cauza ca cel ales avea o limita de traduceri pe zi
- ✓ S-a efectuat sumarizarea articolelor
- ✓ S-au adaugat request-uri
- ✓ S-a facut hash-uirea parolelor astfel incat acestea nu pot fi vizibile, intrucat sunt criptate (s-a folosit SHA-1)
- ✓ S-a adaugat utilizarea de token ce ajuta la securizarea bazei de date. Mecanism:

 Cand un admin se va loga, se va genera un string (asemanator unui id), cu ajutorul java.util.UUID, ce poarta denumirea de token si care va fi memorat in baza de date, in campul token al admin-ului respectiv, si va fi transmis catre front-end. Apoi, la fiecare request efectuat de catre acel admin, back-end-ul va cere obligatoriu de la front-end ca in request sa existe un header cu numele Token ce va contine valoarea generata la logare. Astfel, ne asiguram ca nu oricine poate accesa request-urile.
- ✓ S-a realizat trimiterea erorilor ce pot aparea in timpul rularii aplicatiei

Pornirea aplicatiei pe partea de back-end:

Aplicatia poate fi rulata direct din intellij sau folosind comanda mvn spring-boot:run.



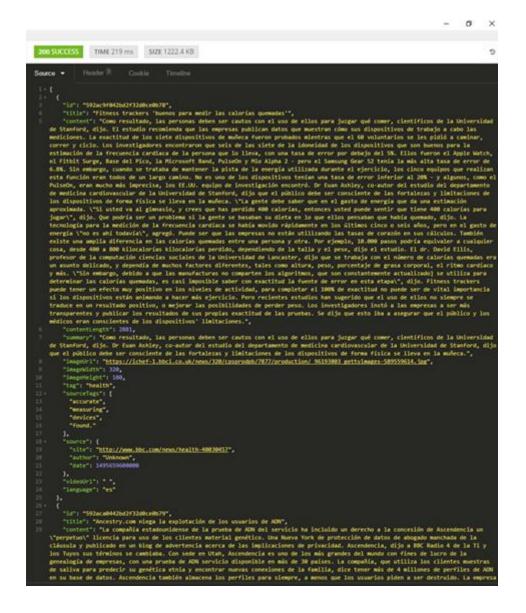


and 200 SUCCESS TIME 32 ms SIZE 38.4 KB Source . Header III Cookie Timeline "10": "9920e85443c072804a7fe2",
"title": "I low you' - Francesco Totti bids an emotional farewell to Roma face",
"content" " Totti's departure has been continued for some time now but that didn't stop his goodbye being an emotionally charged
affair. The day started with fatti paraeling around the pitch before the match, as he took his now contoursy position on the bench.
Totti mas brought on in the second-half and he was on the pitch as Diego Perceti scored a domastic above that saw Roma quality for ment
sesson's Champions League group stage. After the match has finished it was all about Totti as he walked around the ground again, this
time with his family, and eventually took the microphone to address the fans. Totti spoke of his love for the fans and how he was
"afraid" of shat the future any hold for bian. Mis speech in full reads as follows: "Thank you know, "Thank you to my watter and
father, my brother, my relatives and my friends. Thank you to my wife and to my three children. I wonted to start from the end - from
the goodbye. - because I don't know if I'll be able to read these lines. It is appossible to sum up 28 years in a few sentences. I'd
like to do so with a song or posm, but I can't write any. 'Yover the years, I've tried to express myself through my feet, which have
made everything simpler for me ever since I was a child. Speaking of childbood, can you guess what my favourith toy wou'? A football, of
course! And it still is today. At some point in life, you grow up - that's what I've been told and that's what time has decided.

'Yomened time. Back on I'll have 2001, we all sunted list to pass a little zone calcily. We condoin't wait to hear the referee blook the
final whistle. I still get goose bumps now when I think back to it. 'Y'foday, time has come to tap me on the shoulder and say: 'We have
to grow up. As of tomorrow, you'll be an adult. Tain of if those shorts and boost because tarting today, you are a man. You can no longer
to grow up. As of tomorrow, you'll be an adult. Tain of if those unconfirmed what he will do ment, but an offer from Romm is on the table to become a director with a meeting scheduled on Roman.",
"contentingshi's Most are been confirmed for some time now but that didn't stop his goodbye being an emotionally charged
affair. Totti was brought on in the second-half and he was on the pitch as Diggo Perretti scored a drawatic winner that same found upon the second-half and he was on the pitch as Diggo Perretti scored a drawatic winner that same man qualify
for next season's Champions League group stage. After the match had finished it was all about Totti as he walked around the ground
again, this time with his family, and eventually took the microphone to address the fame. Totti spoke of his love for the fame and how
he was 'afraid' of what the viture may hold for him. Now, it's time for ne to thems all of the team-maters, coaches, directors,
presidents and everyone who has worked alongside me during this time. Totti is the all-time record appearance maker for Roma and their
top scorer of all time with 307 goals in all competitions."

"ImageNotics" these/ficeurosport.com/2817/85/28/2893566-53866139-648-586.igg",
"ImageNotics" the score of "imagetr1": "fitte://
'imagetridth": 648,
"imageridint": 560,
"tag": "sport",
"sourcriags": [
"Diego Perotti",
"Giamluigi Buffon"
"Peolo Meldini" "source"; {
 "source"; {
 "source"; {
 "ste': "bttp://curoscert.com/football/serie a/2016-2012/1-love you francesco totti-bids an emotional farraell to rome.
 fans sted3047[Instory.shtml",
 "suther"; " "
 "sete"; 1400035200061

9



Structura aplicatiei (ierarhia de fisiere si rolurile lor):

- APIs -> contine o parte din API-urile utilizate (YandexTranslatorAPI pentru traducerea articolelor)
- **controllers** -> AdminController, ArticleController gestioneaza request-urile, folosind REST API. Request-urile implementate sunt:

*Pentru articole:

- **GET /articles** (toate articolele)
- POST /articles (crearea unui articol, trebuie sa aiba headerul 'Token')
- **GET /articles/page/{language}** (primele 10 articole in ordinea datei articolului pentru pagina 1, urmatoarele 10 pentru pagina 2, etc, in limba aleasa (en, es, it, de, fr))

- **GET /articles/tags/{tags}/{language}** (toate articolele in limba aleasa cu tag-urile din query string ('&' intre ele))
- **GET /articles/tags/{tags}/page/{page}/{language}** (la fel ca cea de dinainte, dar de data aceasta paginate, cate 10 articole pe pagina, in ordinea datelor articolelor)
- GET /articles/{id} (articolul cu id-ul din query string)
- **GET /articles/tags** (toate tagurile)
- GET /articles/languages (toate limbile disponibile)
- PUT /articles/{id} (update/create articol, trebuie sa aiba header-ul 'Token')
- **DELETE /articles/{id}** (sterge articolul cu id-ul din query string, trebuie sa aiba header-ul 'Token')

*Pentru admin:

- **POST /admin** (update/create admin, trebuie sa aiba 'Token')
- POST /admin/login cu body {"username": "..", "password": ".."} (returneaza {"OK": 1} daca exista, {"OK": 0} daca nu exista)
 - **GET /admin/{id}** (admin-ul cu id-ul din query string)
 - **DELETE /admin/{id}** (sterge admin-ul cu id-ul din query string, trebuie sa aiba 'Token')
- models -> <u>AdminModel</u>), <u>ArticleModel</u>, <u>SourceModel</u> constituie "scheletul" cu datele ce caracterizeaza fiecare articol, fiecare admin si fiecare sursa (acestea sunt si campurile din baza de date):

-pentru admins:

```
private String id;

private String title;

private String content;

private int contentLength;

private String summary;

private String imageUrl;

private int imageWidth;

private int imageHeight;

private String tag;

private List<String> sourceTags;

private SourceModel source;

private String videoUrl;

private String language;
```

-pentru articole:

```
private String id;

private String title;

private String content;

private int contentLength;

private String summary;

private String imageUrl;

private int imageWidth;

private int imageHeight;

private String tag;

private List<String> sourceTags;

private SourceModel source;

private String videoUrl;

private String language;
```

-pentru surse:

```
private String site;
private String author;
private Date date;
```

parsers -> contine toate parserele pentru fiecare sursa de unde sunt preluate articolele. Pentru
fiecare website sunt preluate cele mai noi articole de pe o pagina principala, apoi pentru fiecare URL
al fiecarui articol in parte se preiau cu ajutorul librariei Jsoup toate informatiile necesare. Acestea
sunt memorate in variabile intermediare care apoi sunt "adunate" intr-un nou obiect de tipul
ArticleModel dupa care acesta este salvat in baza de date.

Site-urile sursa sunt:

- BBC Health articole de sanatate
- CNN Money -articole de stiri
- Eurosport articole sportive
- HackerNews articole din domeniul IT
- National Geographic articole stiintifice
- MovieWeb articole despre filme
- Reuters Politics articole despre politica
- TechCrunch articole despre tehnologie
- The Guardian Travel articole despre calatorii
- The Verge articole tech

De asemenea, tot aici se realizeaza atat **traducerea** articolelor efectuata prin request-urile HTTP explicate anterior, cat si **sumarizarea** articolelor.

• **repositories** -> AdminRepository, ArticleRepository contine operatiile disponibile ce sunt efectuate efectiv asupra bazei de date:

-pentru admins:

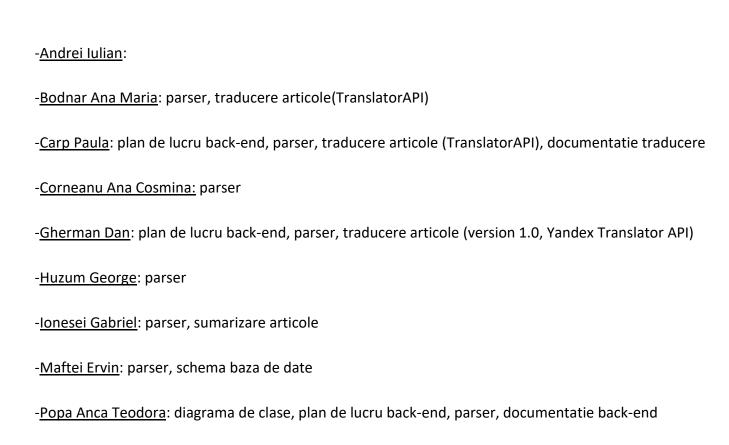
```
public List<AdminModel> findAll();
public AdminModel findOne(String id);
public AdminModel findByUsernameAndPassword(String username, String password);
public AdminModel findByUsername(String username);
public AdminModel findByToken(String token);
public AdminModel save(AdminModel admin);
public void delete(String id);
public long count();
boolean exists(String id);
-pentru articole
 public List<ArticleModel> findAll();
 public List<ArticleModel> findAll(Sort sort);
 public ArticleModel findOne(String id);
 public ArticleModel findByTitle(String title);
 public List<ArticleModel> findByLanguage(String language);
 public List<ArticleModel> findByLanguage(String language, Sort sort);
 public List<ArticleModel> findByLanguageAndTagIn(String language,List<String> tags, Sort sort);
 public List<ArticleModel> findByTagIn(List<String> tags);
 public List<ArticleModel> findByTagIn(List<String> tags, Sort sort);
 public ArticleModel findByTag(String tag);
 public ArticleModel findByTag(String tag, Sort sort);
 public ArticleModel save(ArticleModel article);
 public void delete(String id);
 public long count();
 boolean exists(String id);
```

- services -> constituie legatura intre repositories si controllers
- **summar** -> executa sumarizarea articolelor. Summary genereaza continutul sumarizat al unui text dat ca parametru, dar ofera si posibilitatea de a genera si cuvinte importante din text care pot fi folosite pe post de keywords pentru respectivul articol. Algoritmul de sumarizare consta in procesarea textului si in organizarea lui pe propozitii ,iar apoi in functie de procentul dorit de utilizator sunt extrase propozitiile care sunt considerate mai importante ,fara a le schimba cuvintele sau intelesul. Importanta propozitiilor este calculata folosind algoritmul HITS (https://en.wikipedia.org/wiki/HITS algorithm).

```
Summar summar=new Summar();
summar=new Summar(f_content);
String summary = summar.getSummary();
article.setSummary(summary);
```

3.Realizare task-uri

-Popa Andrei: parser



-<u>Unghianu Anda Costina</u>: diagrama de clase, plan de lucru back-end, schelet java spring cu conexiune la mongo, clasele pentru controllere, modele, servicii si repositories, task scheduling pentru parsere, request-uri (REST API) (15), secure password hashing, token based authentication, trimiterea erorilor pe front-end, corectarea si fixarea erorilor in cod