

## 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)

```
private static String callUrlAndParseResult(String langFrom, String langTo, String word) throws Exception
{
    String url = "https://translate.googleapis.com/translate_a/single?" +
        "client=gtx:" +
        "sl=" + langFrom +
        "tl=" + langTo +
        "&dt=t&q=" + URLEncoder.encode(word, "UTF-8");

    URL obj = new URL(url);
    HttpURLConnection con = (HttpURLConnection) obj.openConnection();
    con.setRequestProperty("User-Agent", "Mozilla/5.0");

    BufferedReader in = new BufferedReader(
        new InputStreamReader(con.getInputStream()));
    String inputLine;
    StringBuffer response = new StringBuffer();

    while ((inputLine = in.readLine()) != null) {
        response.append(inputLine);
    }
    in.close();

    return parseResult(response.toString());
}
```

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 articolelor si a administratorilor
- ✓ S-au realizat parsele 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`.

```
customnews - [D:\facultate\sem II\IP\PPCNews\customnews] - [customnews] - \src\main\java\com\frontpagenews\controllers\ArticleController.java - IntelliJ IDEA 2017.1.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Run CustomnewsApplication
Run CustomnewsApplication
:: Spring Boot :: (v1.5.3.RELEASE)
2017-05-28 20:09:50.789 INFO 6088 --- [main] com.frontpagenews.CustomnewsApplication : Starting CustomnewsApplication on aumhianu-lpt with PID 6088 (started by aumhianu in D:\facultate\sem II\IP\PPCNews\customnews)
2017-05-28 20:09:50.792 INFO 6088 --- [main] com.frontpagenews.CustomnewsApplication : No active profile set, falling back to default profiles: default
2017-05-28 20:09:50.849 INFO 6088 --- [main] ationConfigEmbeddedWebApplicationContext : Refreshing org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicationContext@859717024: startup date [Sun May 28 20:09:50:849 PM 2017]; root: org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicationContext@859717024
2017-05-28 20:09:53.699 INFO 6088 --- [main] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat initialized with port(s): 9181 (http)
2017-05-28 20:09:53.611 INFO 6088 --- [main] org.apache.catalina.core.StandardService : Starting service Tomcat
2017-05-28 20:09:53.612 INFO 6088 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet Engine: Apache Tomcat/8.5.14
2017-05-28 20:09:54.026 INFO 6088 --- [ost-startStop-1] o.a.c.c.c.localhost : Initializing Spring embedded WebApplicationContext
2017-05-28 20:09:54.027 INFO 6088 --- [ost-startStop-1] o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 3101 ms
2017-05-28 20:09:54.153 INFO 6088 --- [ost-startStop-1] o.s.b.w.s.ServletRegistrationBean : Mapping servlet: 'dispatcherServlet' to [/]
2017-05-28 20:09:54.157 INFO 6088 --- [ost-startStop-1] o.s.b.w.s.ServletRegistrationBean : Mapping filter: 'characterEncodingFilter' to [/]
2017-05-28 20:09:54.157 INFO 6088 --- [ost-startStop-1] o.s.b.w.s.ServletRegistrationBean : Mapping filter: 'hiddenHttpMethodFilter' to [/]
2017-05-28 20:09:54.157 INFO 6088 --- [ost-startStop-1] o.s.b.w.s.ServletRegistrationBean : Mapping filter: 'httpPutFormContentFilter' to [/]
2017-05-28 20:09:54.158 INFO 6088 --- [ost-startStop-1] o.s.b.w.s.ServletRegistrationBean : Mapping filter: 'requestContextFilter' to [/]
2017-05-28 20:09:54.689 INFO 6088 --- [main] org.mongodb.driver.cluster : Cluster created with settings (hosts=[localhost:27017], mode=MULTIPLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=5000)
2017-05-28 20:09:54.689 INFO 6088 --- [main] org.mongodb.driver.cluster : Adding discovered server localhost:27017 to client view of cluster
2017-05-28 20:09:54.746 INFO 6088 --- [localhost:27017] org.mongodb.driver.connection : Opened connection [connectionId{localValue:1, serverValue:68}] to localhost:27017
2017-05-28 20:09:54.749 INFO 6088 --- [localhost:27017] org.mongodb.driver.cluster : Discovered cluster type of STANDALONE
2017-05-28 20:09:54.927 INFO 6088 --- [main] org.mongodb.driver.connection : Opened connection [connectionId{localValue:2, serverValue:69}] to localhost:27017
2017-05-28 20:09:55.307 INFO 6088 --- [main] s.w.s.m.a.a.RequestMappingHandlerAdapter : Looking for @ControllerAdvice: org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicationContext@859717024: startup date [Sun May 28 20:09:55:307 PM 2017]; root: org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplicationContext@859717024
2017-05-28 20:09:55.532 INFO 6088 --- [main] s.w.s.m.a.a.RequestMappingHandlerMapping : Mapped "[/admin/{id}],method=[GET]" onto public org.springframework.http.ResponseEntity<com.frontpagenews.models.AdminModel> com.frontpagenews.controllers.AdminController.createAdmin(java.lang.String)
2017-05-28 20:09:55.533 INFO 6088 --- [main] s.w.s.m.a.a.RequestMappingHandlerMapping : Mapped "[/admin],method=[POST]" onto public void com.frontpagenews.controllers.AdminController.deleteAdmin(java.lang.String)
2017-05-28 20:09:55.533 INFO 6088 --- [main] s.w.s.m.a.a.RequestMappingHandlerMapping : Mapped "[/admin/{id}],method=[PUT]" onto public org.springframework.http.ResponseEntity<com.frontpagenews.models.AdminModel> com.frontpagenews.controllers.AdminController.updateAdmin(java.lang.String)
2017-05-28 20:09:55.534 INFO 6088 --- [main] s.w.s.m.a.a.RequestMappingHandlerMapping : Mapped "[/admin/login],method=[POST],consumes=[application/json]" onto public org.springframework.http.ResponseEntity<java.lang.String> com.frontpagenews.controllers.AdminController.login(java.lang.String,java.lang.String)
2017-05-28 20:09:55.537 INFO 6088 --- [main] s.w.s.m.a.a.RequestMappingHandlerMapping : Mapped "[/articles/page/{page}/{language}],method=[GET]" onto public java.util.List<com.frontpagenews.models.ArticleModel> com.frontpagenews.controllers.ArticleController.getPageArticles(java.lang.String,int)
2017-05-28 20:09:55.537 INFO 6088 --- [main] s.w.s.m.a.a.RequestMappingHandlerMapping : Mapped "[/articles/{language}],method=[GET]" onto public java.util.List<com.frontpagenews.models.ArticleModel> com.frontpagenews.controllers.ArticleController.getArticlesByLanguage(java.lang.String)
2017-05-28 20:09:55.538 INFO 6088 --- [main] s.w.s.m.a.a.RequestMappingHandlerMapping : Mapped "[/articles],method=[GET]" onto public org.springframework.http.ResponseEntity<com.frontpagenews.models.ArticleModel> com.frontpagenews.controllers.ArticleController.getArticles()
2017-05-28 20:09:55.538 INFO 6088 --- [main] s.w.s.m.a.a.RequestMappingHandlerMapping : Mapped "[/articles/tags],method=[GET]" onto public java.util.List<com.frontpagenews.models.ArticleModel> com.frontpagenews.controllers.ArticleController.getArticlesByTag(java.lang.String)
2017-05-28 20:09:55.538 INFO 6088 --- [main] s.w.s.m.a.a.RequestMappingHandlerMapping : Mapped "[/articles],method=[POST]" onto public void com.frontpagenews.controllers.ArticleController.createArticle(java.lang.String,java.lang.String,java.lang.String)
2017-05-28 20:09:55.538 INFO 6088 --- [main] s.w.s.m.a.a.RequestMappingHandlerMapping : Mapped "[/articles/{id}],method=[PUT]" onto public org.springframework.http.ResponseEntity<com.frontpagenews.models.ArticleModel> com.frontpagenews.controllers.ArticleController.updateArticle(java.lang.String)
2017-05-28 20:09:55.538 INFO 6088 --- [main] s.w.s.m.a.a.RequestMappingHandlerMapping : Mapped "[/articles/{id}],method=[DELETE]" onto public void com.frontpagenews.controllers.ArticleController.deleteArticle(java.lang.String)
2017-05-28 20:09:55.539 INFO 6088 --- [main] s.w.s.m.a.a.RequestMappingHandlerMapping : Mapped "[/articles/tags/{tags}/{language}],method=[GET]" onto public java.util.List<com.frontpagenews.models.ArticleModel> com.frontpagenews.controllers.ArticleController.getArticlesByTagAndLanguage(java.lang.String,java.lang.String)
2017-05-28 20:09:55.544 INFO 6088 --- [main] s.w.s.m.a.a.RequestMappingHandlerMapping : Mapped "[/error]" onto public org.springframework.http.ResponseEntity<java.lang.String, java.lang.Object> org.springframework.web.servlet.mvc.annotation.annotation.AnnotationMethodHandlerAdapter.handle(org.springframework.http.HttpServletRequest,org.springframework.http.HttpServletResponse,org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter.HandlerMethod)
2017-05-28 20:09:55.545 INFO 6088 --- [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/webjars/**] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2017-05-28 20:09:55.576 INFO 6088 --- [main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**/favicon.ico] onto handler of type [class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
2017-05-28 20:09:55.710 INFO 6088 --- [main] o.s.j.e.a.AnnotationBeanExporter : Registering beans for JMX exposure on startup
2017-05-28 20:09:55.760 INFO 6088 --- [main] s.a.ScheduledAnnotationBeanPostProcessor : No TaskScheduler/ScheduledExecutorService bean found for scheduled processing
2017-05-28 20:09:55.812 INFO 6088 --- [main] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat started on port(s): 9181 (http)
2017-05-28 20:09:55.918 INFO 6088 --- [main] com.frontpagenews.CustomnewsApplication : Started CustomnewsApplication in 5.366 seconds (JVM running for 6.72)
```

Send 200 SUCCESS TIME 187 ms SIZE 89 B

Source	Header	Cookie	Timeline
1 = {			
2 "health",			
3 "sport",			
4 "science",			
5 "movie",			
6 "politics",			
7 "technology",			
8 "travel",			
9 "computer security"			
10 }			

Source ▾ Header ▾ Cookie Timeline

```
1 - {
2   {
3     "id": "592b685442b2f22894a7f8e1",
4     "title": "'I love you' - Francesco Totti bids an emotional farewell to Roma fans",
5     "content": "Totti's departure has been confirmed for some time now but that didn't stop his goodbye being an emotionally charged affair. The day started with Totti parading around the pitch before the match, as he took his new customary position on the bench. Totti was brought on in the second-half and he was on the pitch as Diego Perotti scored a dramatic winner that saw Roma 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 fans. Totti spoke of his love for the fans and how he was 'afraid' of what the future may hold for him. His speech in full reads as follows: \"Thank you, Rome. \"Thank you to my mother and father, my brother, my relatives and my friends. Thank you to my wife and to my three children. I wanted to start from the end - from the goodbyes - because I don't know if I'll be able to read these lines. It is impossible to sum up 28 years in a few sentences. I'd like to do so with a song or poem, but I can't write any. \"Over 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 childhood, can you guess what my favourite toy was? 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. \"Damned time. Back on 17 June 2001, we all wanted time to pass a little more quickly. We couldn't wait to hear the referee blow the final whistle. I still get goose bumps now when I think back to it. \"Today, time has come to tap me on the shoulder and say: 'We have to grow up. As of tomorrow, you'll be an adult. Take off those shorts and boots because starting today, you are a man. You can no longer enjoy the smell of the grass, the sun on your face as you bear down on the opposition's goal, the adrenaline consuming you, the joy of celebrating. \"Over the past few months, I've asked myself why I'm being awoken from this dream. Imagine you're a child having a good dream, and your mother wakes you up to go to school. You want to keep dreaming, you try to slip back into the dream but you never can. This time, it's not a dream, but reality. And I can no longer slip back in. \"I want to dedicate this letter to all of you - to all the children that have supported me. To the children of yesterday, who have grown up and become parents and to the children of today, who perhaps shout 'Tottigol'. I'd like to think that for you, my career has become a fairytale for you to pass on. \"It's really over now. I'm taking off that jersey for the final time. I'll fold it away, even though I'm not ready to say 'enough' and perhaps I never will be. I forgive me for not giving interviews and clarifying my thoughts, but it's not easy to turn out the light. I'm afraid. It's not the same fear you feel when you're standing in front of the goal, about to take a penalty. \"This time, I can't see what the future looks like through the holes of the net. Allow me to be afraid. This time, it's me who needs you and the love that you've always shown me. With your support, I will succeed in turning the page and throwing myself into a new adventure. Now, it's time for me to thank all of the team-mates, coaches, directors, presidents and everyone who has worked alongside me during this time. \"To the fans and the Curva Sud, a guiding light for all Romans and Romanisti. Being born Roman and Romanisti is a privilege. Being the captain of this team is an honour. You are - and will always be - my life. I will no longer entertain you with my feet, but my heart will always be there with you. \"Now, I will go down the stairs and enter the dressing room that welcomed me as a child and that I now leave as a man. \"I'm proud and happy to have given you 28 years of love. \"I love you.\" In total Totti played for Roma for 25 seasons, racking up a remarkable 786 appearances. 639 of those were in Serie A which places him joint-second with Gianluigi Buffon, behind Paolo Maldini (647). Totti is the all-time record appearance maker for Roma and their top scorer of all time with 307 goals in all competitions. No player has scored in more Serie A seasons than his 23 and he was the youngest captain in Serie A history when he was appointed at the age of 22. It is still unconfirmed what he will do next, but an offer from Roma is on the table to become a director with a meeting scheduled on Monday.\".",
6     "contentlength": 4547,
7     "summary": "Totti's departure has 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 Diego Perotti scored a dramatic winner that saw Roma 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 fans. Totti spoke of his love for the fans and how he was 'afraid' of what the future may hold for him. Now, it's time for me to thank all of the team-mates, 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. \".",
8     "imageUrl": "http://i.eurport.com/2017/05/18/2893546-53866139-648-368.jpg",
9     "imageWidth": 648,
10    "imageHeight": 368,
11    "tag": "sport",
12    "sourceTags": {
13      "Diego Perotti",
14      "Gianluigi Buffon",
15      "Paolo Maldini"
16    },
17    "source": {
18      "site": "http://eurport.com/football/serie-a/2016-2017/i-love-you-francesco-totti-bids-an-emotional-farewell-to-roma-fans-510618647/story.html",
19      "author": "",
20      "date": 1496835200000
21    },
22    "videoId": ""
23  }
```



```
200 SUCCESS TIME 219 ms SIZE 1222.4 KB

Source Header Cookie Timeline

1 {
2 {
3 "id": "592aca942bd2f320fce0b78",
4 "title": "Fitness trackers 'buenos para medir las calorías quemadas'",
5 "content": "Como resultado, las personas deben ser cautas con el uso de ellos para juzgar qué comer, científicos de la Universidad de Stanford, dijo. El estudio recomienda que las empresas publiquen datos que muestran cómo sus dispositivos de trabajo a cabo las mediciones. La exactitud de los siete dispositivos de muñeca fueron probados mientras que el 48 voluntarios se les pidió a caminar, correr y ciclismo. Los investigadores encontraron que seis de los siete de la idoneidad de los dispositivos que son buenos para la estimación de la frecuencia cardíaca de la persona que lo lleva, con una tasa de error por debajo del 5%. Ellos fueron el Apple Watch, el Fitbit Surge, Base del Pico, la Microsoft Band, PulseOn y Mio Alpha 2 - pero el Samsung Gear S2 tenía la más alta tasa de error de 6.8%. Sin embargo, cuando se trataba de mantener la pista de la energía utilizada durante el ejercicio, los cinco equipos que realizan esta función eran todos de un largo camino. No es uno de los dispositivos tenían una tasa de error inferior al 20% - y algunos, como el PulseOn, eran mucho más imprecisa, los EE.UU. equipo de investigación encontró. Dr Juan Ashley, co-autor del estudio del departamento de medicina cardiovascular de la Universidad de Stanford, dijo que el público debe ser consciente de las fortalezas y limitaciones de los dispositivos de forma física se lleva en la muñeca. "La gente debe saber que en el gasto de energía que da una estimación aproximada. "Si usted va al gimnasio, y crees que has perdido 400 calorías, entonces usted puede sentir que tiene 400 calorías para jugar", dijo. Que podría ser un problema si la gente se basaban su dieta en lo que ellos pensaban que había quemado, dijo. La tecnología para la medición de la frecuencia cardíaca se había movido rápidamente en los últimos cinco o seis años, pero en el gasto de energía "no es ahí todavía", agregó. Puede ser que las empresas no están utilizando las tasas de corazón en sus cálculos. También existe una amplia diferencia en las calorías quemadas entre una persona y otra. Por ejemplo, 10,000 pasos podría equivaler a cualquier cosa, desde 400 a 800 kilocalorías kilocalorías perdido, dependiendo de la talla y el peso, dijo el estudio. El dr. David Ellis, profesor de la computación ciencias sociales de la Universidad de Lancaster, dijo que se trabaja con el número de calorías quemadas era un asunto delicado, y dependía de muchos factores diferentes, tales como altura, peso, porcentaje de grasa corporal, el ritmo cardíaco y más. "Sin embargo, debido a que las manufacturas no comparten los algoritmos, que son constantemente actualizado se utiliza para determinar las calorías quemadas, es casi imposible saber con exactitud la fuente de error en esta etapa", dijo. Fitness trackers puede tener un efecto muy positivo en los niveles de actividad, para completar el 100% de exactitud no puede ser de vital importancia si los dispositivos están animando a hacer más ejercicio. Pero recientes estudios han sugerido que el uso de ellos no siempre se traduce en un resultado positivo, o mejoran las posibilidades de perder peso. Los investigadores instó a las empresas a ser más transparentes y publicar los resultados de sus propias exactitud de las pruebas. Se dijo que esto iba a asegurar que el público y los médicos eran conscientes de las dispositivos' limitaciones.",
6 "contentlength": 2801,
7 "summary": "Como resultado, las personas deben ser cautas con el uso de ellos para juzgar qué comer, científicos de la Universidad de Stanford, dijo. Dr Juan Ashley, co-autor del estudio del departamento de medicina cardiovascular de la Universidad de Stanford, dijo que el público debe ser consciente de las fortalezas y limitaciones de los dispositivos de forma física se lleva en la muñeca.",
8 "imageurl": "https://ichef-1.thcl.co.uk/news/320/cosmosdb/7877/production/96333801/activimage-589559633.jpg",
9 "imageid": 320,
10 "imageheight": 180,
11 "tag": "health",
12 "sourceTag": {
13 "accurate",
14 "measuring",
15 "devices",
16 "found."
17 },
18 "source": {
19 "site": "http://www.thcl.co.uk/news/health-40030457",
20 "author": "Unknown",
21 "date": 1495456600000
22 },
23 "video": {
24 "language": "es"
25 },
26 {
27 "id": "592aca0442bd2f320fce0b79",
28 "title": "Ancestry.com niega la explotación de los usuarios de ADN",
29 "content": "La compañía estadounidense de la prueba de ADN del servicio ha incluido un derecho a la concesión de Ascendencia un 'perpetua' licencia para uno de los clientes material genético. Una Nueva York de protección de datos de abogado manchada de la cláusula y publicado en un blog de advertencia acerca de las implicaciones de privacidad. Ascendencia, dijo a BBC Radio 4 de la TI y los Tuyo sus términos se cambian. Con sede en Utah, Ascendencia es uno de los más grandes del mundo con fines de lucro de la genealogía de empresas, con una prueba de ADN servicio disponible en más de 30 países. La compañía, que utiliza los clientes muestras de saliva para predecir su genética etnia y encontrar nuevas conexiones de la familia, dice tener más de 4 millones de perfiles de ADN en su base de datos. Ascendencia también almacena los perfiles para siempre, a menos que los usuarios piden a ser destruido. La empresa
```

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/{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** -> AdminModel), ArticleModel, SourceModel 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 parsele 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 libreriei 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](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

- Andrei Iulian: request-uri (schimbate ulterior)
- Bodnar Ana Maria: parser, traducere articole(TranslatorAPI)
- Carp Paula: plan de lucru back-end, parser, traducere articole (TranslatorAPI), documentatie traducere
- Corneanu Ana Cosmina: parser
- Gherman Dan: plan de lucru back-end, parser, traducere articole (version 1.0, Yandex Translator API)
- Huzum George: parser
- Ionesei Gabriel: parser, sumarizare articole
- Maftai Ervin: parser, schema baza de date
- Popa Anca Teodora: diagrama de clase, plan de lucru back-end, parser, documentatie back-end
- Popa Andrei: parser
- Unghianu Anda Costina: 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