

*Računarstvo i informatika*

*Katedra za računarstvo  
Elektronski fakultet u Nišu*

Napredne baze podataka  
**Full-text pretraživanje  
informacija**

Zimski semestar 2020/2021



# Pretraživanje informacija

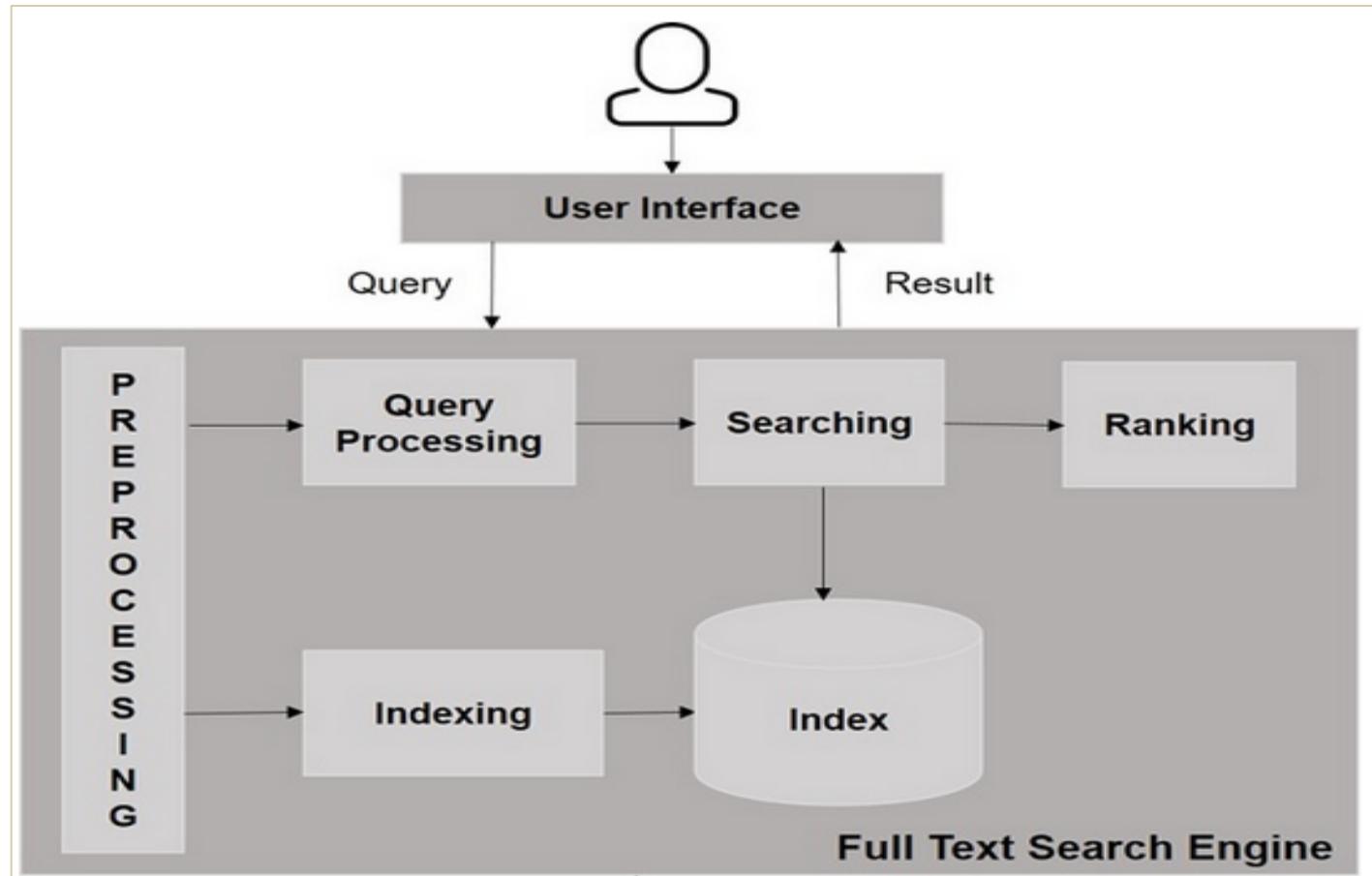
- Information retrieval – Pretraživanje informacija
- Pretraživanje informacija predstavlja aktivnost pribavljanje relevantnih resursa iz kolekcije resursa na osnovu zadatog kriterijuma.
- Resurs najčešće predstavlja nestrukturani dokument.
- Sama pretraga može biti bazirana na metapodacima ili bazirana na full-text (ili drugim content-based) indeksima.
- Najpoznatiji sistem za pretragu informacija danas predstavljaju internet pretraživači.
- Pronalaženje dokumenata radi po principu upoređivanja korisničkog upita, koji može da varira od nekoliko reči do čitavog opisa, sa nestruktuiranim tekstualnim dokumentima (web stranica, novinski članak, pdf knjiga, cv obrazac itd.)



# Pretraživanje informacija

- Kod pretrage bazirane na metapodacima pretražuju se samo polja koja opisuju (apstrahuju) sam dokument (naslov, autori, datum izdavanja, žanrovi i sl.).
- Full-text pretraga predstavlja tehniku pretrage dokumenata kod koje se pretražuje ceo dokument tj. njegov sadržaj.
- Dva najbitnija aspekta full-text pretrage:
  - **Relevantnost** – mogućnost da se rezultati pretrage rangiraju po tome koliko su relevantni za dati upit (query).
  - **Analiza** – mogućnost pretvaranja bloka teksta u posebne normalnizovane tokene, na osnovu kojih se kreira indeks za pretragu.

# Pretraživanje informacija



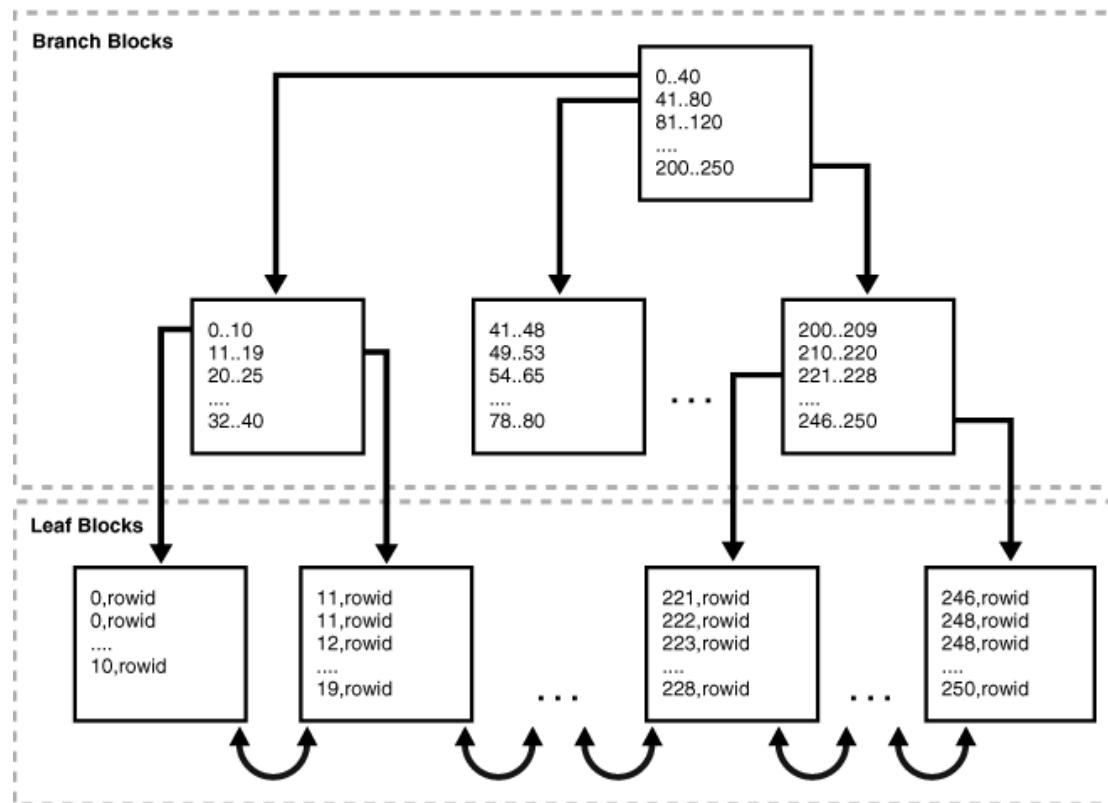


# Pretraživanje informacija

- Osnovne aktivnosti koje treba obezbiti kod svakog mehanizma za full-text pretragu:
  - **Procesiranje i indeksiranje dokumenata** – obavlja se nad dokumentima tokom njihovog dodavanja u sistem.
    - Dokumenti se svode na predefinisani standardni format za skladištenje. Tokom ovog procesa vrši se kreiranje novog ili ažuriranje postojećeg indeksa koji se koristi prilikom pretrage.
    - Ovaj proces se obično odvija u offline režimu.
  - **Procesiranje upita** - interpretacija korisnikovog zahteva odnosno pitanja na osnovu čega se upit (query)
    - Transformacije koje se primenjuju na dokumentom primenjuju se i nad korisničkim zahtevom
    - Obavljaju se dodatne transformacije
    - Vrši se dinamički u real-time režimu

# Invertovani indeks

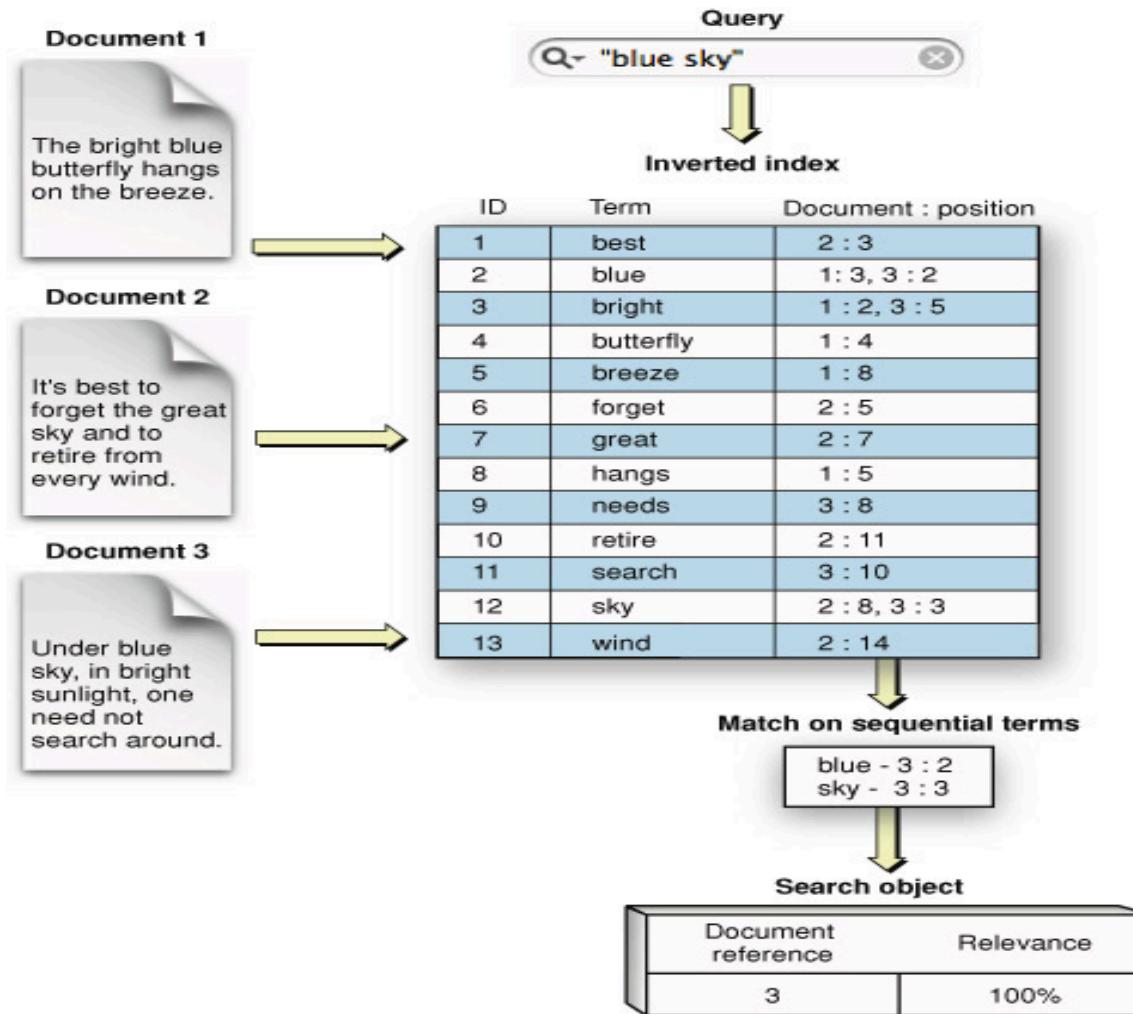
- Standardni indeks kod relacionih baza podataka
  - B/stablo
  - Sortirano za range upite
  - Složenost operacija  $O(\log(n))$



# Invertovani indeks

- Centralna struktura koja se koristi kod full-text pretrage.
- Indeksna struktura kod koje se sadržaj dokumenta, reči ili brojevi, mapira na lokaciju u okviru dokumenta ili skupa dokumenata.
- Ovakva struktura omogućava izuzetno brzu pretragu, pri čemu zahteva dodatno procesiranje prilikom dodavanja novih dokumenata u kolekciju.
- Struktura invertvanog indeksa:
  - **Vokabular (vocabulary)** – lista svih reči koje se pojavljuju u dokumentu ili skupu dokumenata
  - **Pojavljivanja (Occurrences)** - lista koja sadrži informacije o svakoj reči iz vokabulara (dokumenti u kojima se reč pojavljuje, pozicija u okviru dokumenta, frekvencija pojavljivanja i sl.)

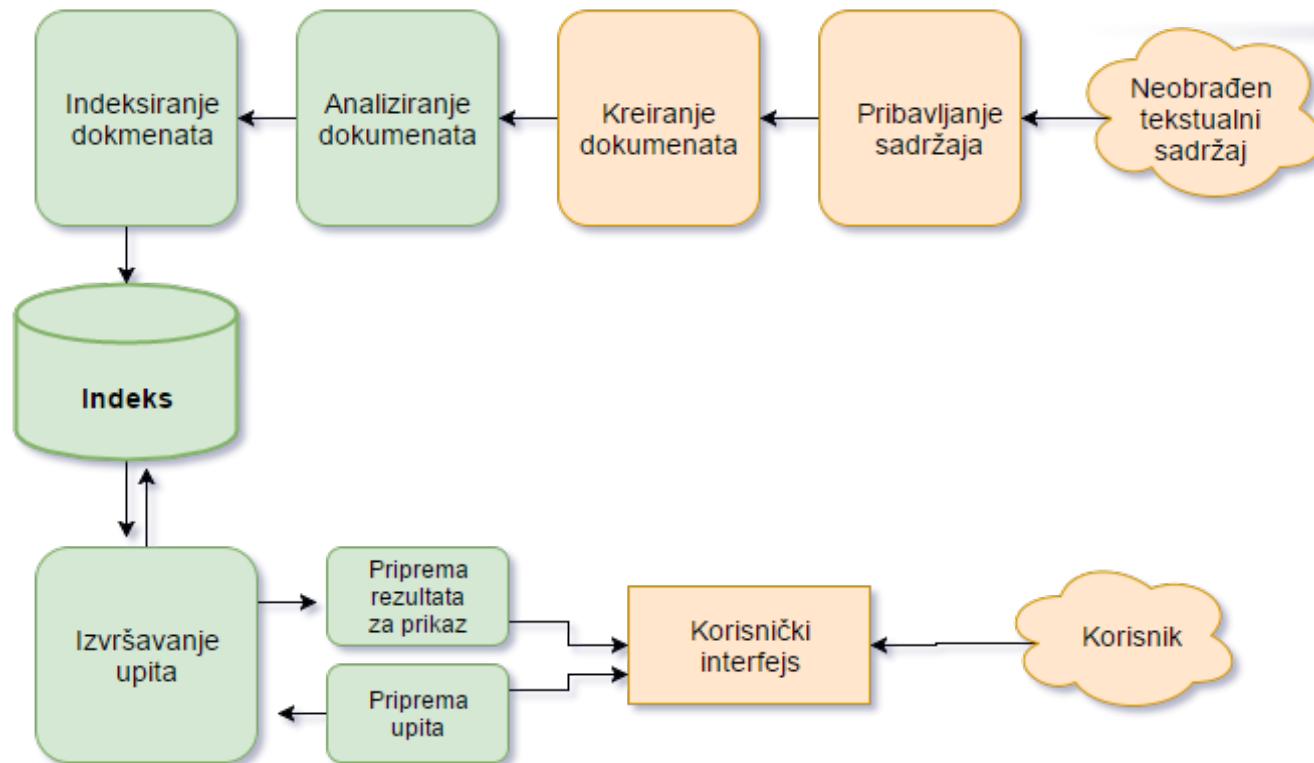
# Invertovani indeks



# Lucene bibliotekna

- Lucene je biblioteka za pretraživanje informacija koja omogućava skalabilnu pretragu sa visokim performansama.
- Koristi se za pretraživanje dokumenata, tj. informacija unutar dokumenata ili metapodataka o dokumentima.
- Bazira se na implementaciji invertovanog indeksa.
- Besplatna biblioteka otvorenog koda.
- Implementirana je u Java programskom jeziku.
- Pod okriljem Apache Software Foundation organizacije.
- Jedna od najpopularniju biblioteku ovog tipa.

# Lucene biblioteka





# Lucene biblioteka

- U osnovi Lucene biblioteka definiše model dokumenta.
- Lucene indeks se sastoji od dokumenata.
- Dokument se sastoji od jednog ili više polja.
- Svako polje ima attribute:
  - Koji je tip polja.
  - Kako obraditi sadržaj polja (Analyzers,Tokenizers).
  - Da li se polje čuva ili se samo koristi kod indeksiranja.
  - Težina polja koja određuje relevantnost sadržaja polja prilikom pretrage.
- Primeri polja koja se najčešće javljaju u dokumentu su: naslov, sadržaj, autor, metapodaci i URL.

# Lucene biblioteka

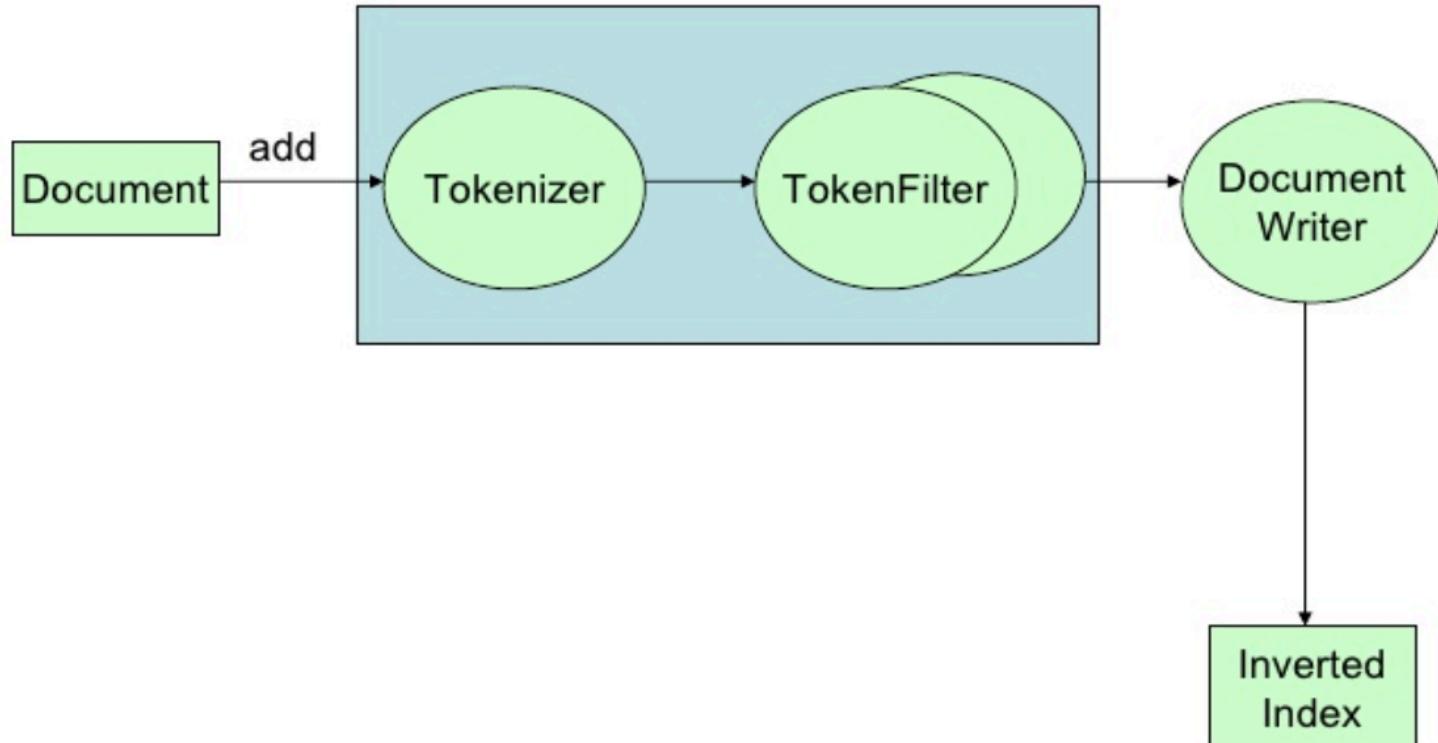
- Komponenta za kreiranje dokumenata ima zadatak da izvrši ekstrakciju teksta iz pribavljenog sadržaja.
- Ukoliko je pribavljeni sadržaj već u formi teksta, ovaj korak je trivijalan.
- Međutim, ovaj postupak može biti i složen ukoliko je tekst u okviru binarne datoteke (PDF, Microsoft Office, Open Office, Adobe Flash...) ili sadrži određene oznake ili tagove (XML, HTML, RDF), koje je neophodno ukloniti.
- Lucene sadrži skup funkcija za kreiranje dokumenata i njihovih polja, ali ne sadrži nikakve mehanizme za njihovo automatsko kreiranje iz prikupljenog sadržaja zbog toga što se ti postupci razlikuju od slučaja do slučaja.

# Lucene biblioteka

- Komponenta za analizu dokumenata ima zadatak da tekst u okviru dokumenta podeli na individualne nedeljive elemente – tokene. Svaki token odgovara jednoj reči u jeziku, a ova komponenta određuje na koji način se tekst deli na niz tokena.
- Najčešći problemi koje ova komponenta treba da reši su:
  - Da li treba izvršiti korekciju pravopisnih grešaka?
  - Kako tretirati složenice?
  - Da li treba uključiti i sinonime za određene reči kako bi pretraga bila uspešnija?
  - Da li treba smatrati jedninu i množinu iste reči istim tokenom?
  - Da li treba voditi računa o velikim i malim slovima?
- Lucene nudi više ugrađenih analizatora čijim se kombinovanjem (rednim vezivanjem) može implementirati ova komponenta.
- Moguće je implementirati i specijalizovani analizator.

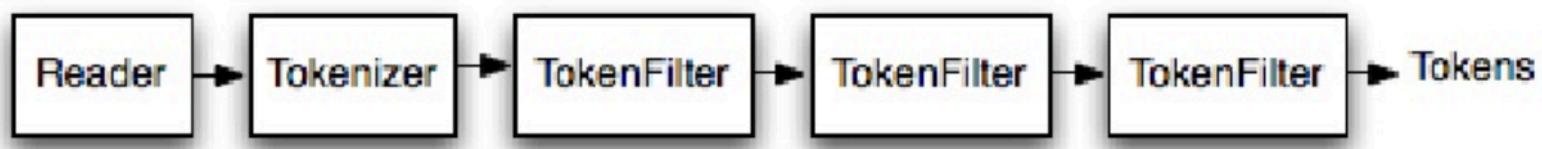
# Lucene biblioteka

- Proces analize dokumenata



# Lucene biblioteka

- Proces analize dokumenata



- 1 Tokenizer
- N TokenFilters



# Lucene biblioteka

"The quick brown fox jumped over the lazy dogs"

**WhitespaceAnalyzer :**

[The] [quick] [brown] [fox] [jumped] [over] [the] [lazy] [dogs]

**SimpleAnalyzer :**

[the] [quick] [brown] [fox] [jumped] [over] [the] [lazy] [dogs]

**StopAnalyzer :**

[quick] [brown] [fox] [jumped] [over] [lazy] [dogs]

**StandardAnalyzer:**

[quick] [brown] [fox] [jumped] [over] [lazy] [dogs]

"XY&Z Corporation - xyz@example.com"

**WhitespaceAnalyzer:**

[XY&Z] [Corporation] [-] [xyz@example.com]

**SimpleAnalyzer:**

[xy] [z] [corporation] [xyz] [example] [com]

**StopAnalyzer:**

[xy] [z] [corporation] [xyz] [example] [com]

**StandardAnalyzer:**

[xy&z] [corporation] [xyz@example.com]

# Lucene biblioteka

- Priprema upita je postupak koji je neophodan nakon zadavanja upita od strane korisnika kako bi upit bio prilagođen Lucene biblioteci
- Paket QueryParser omogućava omogućava postupa kreiranja upita.
- Upiti mogu biti prosti – jedna reč ili fraza, mogu sadržati i više reči povezanih logičkom operatorima, wildcard karaktere itd.
- Izvršavanje upita je proces u okviru koga se pretražuje indeks, pribavljuju dokumenti koji odgovaraju upitu i rezultat se sortira na odgovarajući način.

# Lucene biblioteka

- U teoriji pretraživanja informacija najčešće se pominju tri modela pretraživanja:
  - **Bulov model** – Dokumenti zadovoljavaju ili ne zadovoljavaju dati upit.
  - **Vektorski model** – Dokumenti i upiti se modeluju kao vektori u višedimenzionalnom prostoru, a relevantnost se računa kao rastojanje između tih vektora.
  - **Probabilistički model** – U ovom modelu se računa verovatnoća da dokument zadovoljava upit (koristi se u biblioteci Xapian).
- Lucene biblioteka kombinuje vektorski model i Bulov model.
- Lucene biblioteka nudi mogućnost izbora modela na nivou svake pretrage.

# Lucene biblioteka

- Prednosti korišćenja:
  - Brzina
  - Jednostavan API
  - Konkurentno indeksiranje i pretraživanje
  - Inkrementalno indeksiranje
  - Near real-time performanse
- Nedostaci:
  - Ne podržava spojeve
  - Nema delimičnog ažuriranja dokumenata
  - Ažuriranej dokumenta se svodi na operaciju brisanja I dodavanja novog dokumenta.

# Lucene biblioteka

- Rešenja zasnovana na Lucene biblioteci:
  - Apache Solr
  - **Elasticsearch**
  - Katta
  - Bobo Search
  - Summa
  - Constellio

# Elasticsearch

- Platforma skladištenje i pretragu dokumenata bazirana na korišćenju Lucene indeksa.
- NoSQL document store implementiran korišćenjem Lucene indeksa.
- Implementiraj korišćenjem JAVA programskog jezika.
- JSON kao standardni format za skladištenje dokumenata.
- Pouzdano i skalabilno rešenje,
- Otpornost na pojavu grešaka.
- Podržane replikacije i distribuirano indeksiranje.
- Podržava balansiranej opterećenja prilikom izvršavanja upita.

# Elasticsearch

- **Klaster (Cluster)** – Kolekcija od jednog ili više servera (čvorova) koji zajedno čuvaju korisničke podatake i omogućuju indeksiranje i pretragu podataka u čvorovima koji se nalaze u okviru klastera.
- **Čvor (Node)** – Čvor je server na kome se čuvaju podaci i koji je deo klastera.
- **Indeks (Index)** - Indeks je kolekcija dokumenata koji imaju neke zajedničke karakteristike. Na pojam indeksa se može gledati kao na pojam baze podataka u relacionom modelu.
- **Tip (Type)** – U okviru indeksa mogu postojati više različitih tipova. Tip predstavlja logičko particionisanje indeksa. Generalno, dokumenta koja pripadaju istom tipu, sadrže ista polja tj. imaju istu strukturu (mapiranje). U koliko posmatramo paralelu sa relationalnim modelom, tip bi odgovarao pojmu tabele.

# Elasticsearch

- **Dokument (Document)** – Dokument je osnovna informativna jedinica koja se može indeksirati. Svaki dokument se nalazi u JSON formatu.
- **Indeksiranje dokumenata (to Index a document)** – Pod indeksiranjem dokumenta, podrazumeva se smeštanje dokumenta u indeks (index) odnosno pribavljanje, obrada, analiza dokumenta i ažuriranje indeksa.
- **Shard (particije)** – Indeks može da sadrži velike količine podataka, koji mogu da prevazilaze hardverske mogućnosti čvora na kome se nalaze, Elasticsearch deli indeks na mnogo manjih delova – particije. Na particiju može da se gleda kao na samostalni, u potpunosti funkcionalni indeks koji može da se nalazi na bilo kom čvoru u klasteru.

# Elasticsearch

- **Shard (particije)**

- Mehanizam podele indeksa na particije obezbeđuje horizontalno skaliranje, distribucija i paralelizacija pretrage na nivou particija.
- Za sam proces podele indeksa na particije i agregacija dokumenata iz particijaprilikom upita, je zadužen Elasticsearch i taj mehanizam je transparentan korisniku.

- **Replike** - Da ne bi došlo do otkazivanja sistema i gubljenja podataka usled otkaza u mreži, odnosno da bi se omogućila dostupnost sistema, svaki shard može da ima jednu ili više replika (replica shards). Replika shard se nikad ne nalazi na istom čvoru kao i njen original (primary shard).

# Elasticsearch

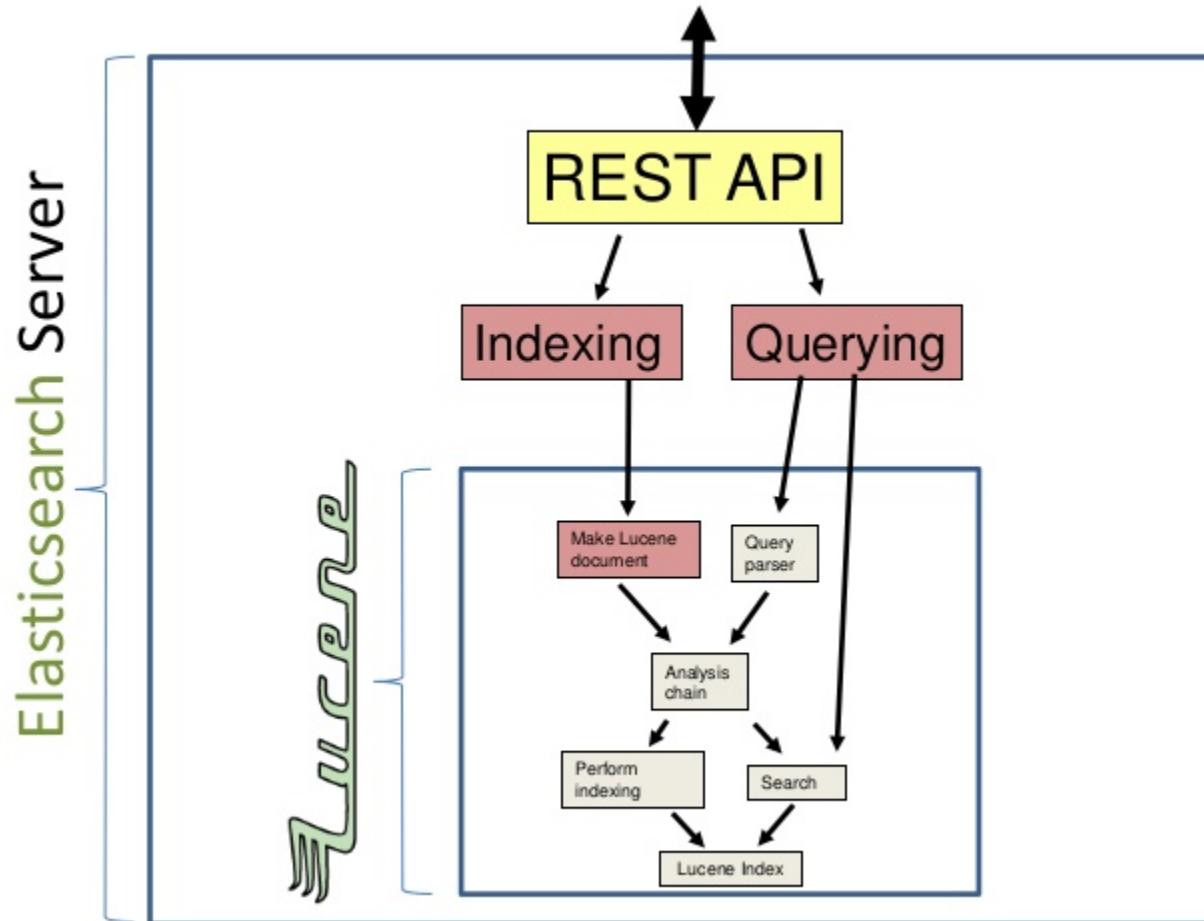
## TERMINOLOGY

| MySQL                   | Elastic Search        |
|-------------------------|-----------------------|
| Database                | Index                 |
| Table                   | Type                  |
| Row                     | Document              |
| Column                  | Field                 |
| Schema                  | Mapping               |
| Index                   | Everything is indexed |
| SQL                     | Query DSL             |
| SELECT * FROM table ... | GET http://...        |
| UPDATE table SET ...    | PUT http://...        |

# Elasticsearch

- Za svako tekstualno polje dokumenta moguće je navesti analizator.
- Ukoliko se ne navede nijedan, Elasticsearch će podrazumevano koristiti standardni analizator.
  - Ovaj analizator pruža tokenizaciju na osnovu Unicode Text Segmentation algoritma, koji se dobro pokazao u praksi za većinu jezika.
  - Pored ovoga, vrši i konverziju svih slova u mala i opciono izbacuje reči koje nemaju značaj za pretragu kao što su članovi, veznici, rečce i predlozi.
- Pored standardnog analizatora, Elasticsearch ima ugrađene i brojne druge. Neki od njih su:
  - **Simple analyzer** – deli tekst na tokene koji su ograničeni karakterima koji nisu slova i konvertuje sva slova u mala
  - **Whitespace analyzer** – deli tekst na tokene koji su ograničeni bilo kojim whitespace karakterima.
  - **Keyword analyzer** – čitavo tekstualno polje posmatra kao jedan token
  - **Pattern analyzer** – koristi regularne izraze za formiranje tokena.
  - **Language analyzers** – analizatori prilagođeni određenom jeziku.

# Elasticsearch





# Elasticsearch

A screenshot of a web browser window displaying the results of a Elasticsearch search. The URL in the address bar is `localhost:9200/test/_search?q=smashing`. The page content shows a JSON response:

```
{  
  "took": 2,  
  "timed_out": false,  
  "_shards": {  
    "total": 5,  
    "successful": 5,  
    "failed": 0  
  },  
  "hits": {  
    "total": 1,  
    "max_score": 0.15342641,  
    "hits": [  
      {  
        "_index": "test",  
        "_type": "stupid-hypes",  
        "_id": "gallon-smashing",  
        "_score": 0.15342641,  
        "_source": {  
          "name": "Gallon Smashing",  
          "stupidity_level": "10",  
          "lifetime": 30  
        }  
      }  
    ]  
  }  
}  
hits.hits[0]._source.name
```

# Elasticsearch

- Korisnici:
  - GitHub
  - Facebook
  - Stackoverflow
  - Soundcloud
  - StumbleUpon
  - Wikimedia Foundation
  - ...