Kod szkolenia: **FULLTEXT**

Tytuł szkolenia: **Technologie wyszukiwania pełnotekstowego**

Dni: 3

Partner merytoryczny SigDelta

Opis:

Adresaci szkolenia:

Szkolenie adresowane jest do osób zainteresowanych funkcjonalnością wyszukiwania pełnotekstowego i różnych spoosbów jej wykorzystania.

Cel szkolenia:

Uczestnicy dowiedzą się czym jest wyszukiwanie pełnotekstowe, zagadnienia z nim związane, jakie są możliwości jego realizacji oraz najlepsze praktyki. Dodatkowo poznają szeroką gamę produktów realizujących tą funkcjonalność uwzględniając rozwiązania zarówno bazodanowe jak i dedykowane biblioteki programistyczne, w szczególności te wiodące na rynku jak Apache Lucene, Apache Solr, Sphinx Search czy Hibernate Search. Po szkoleniu uczestnicy będą potrafili zaprojektować oraz oprogramować system dostarczający funkcjonalność wyszukiwania pełnotekstowego. Szkolenie traktuje temat w bardzo szerokim aspekcie tak, aby uczestnik był przygotowany do realizacji tego zadania tak, aby spełniło ono wszystkie wymagania oraz potrafił wybrać rozwiązanie najlepsze w danej sytuacji.

Mocne strony szkolenia:

Program obejmuje całościowo i wyczerpująco zagadnienia związane z wyszukiwaniem pełnotekstowym, przedstawia wiele ciekawych aspektów, typowe problemy oraz najlepsze praktyki. W przeciwieństwie do większości tego typu szkoleń, zawiera ono część warsztatową, która pozwoli na ugruntowanie wiedzy również w praktycznym jej aspekcie.

Wymagania:

Od uczestników wymagana jest podstawowa umiejętność znajomość baz danych i języka SQL oraz programowania w języku Java.

Parametry szkolenia:

3*8 godzin (3*7 netto) wykładów i warsztatów. Wielkość grupy: maks. 8-10 osób

Sages sp. z o.o., ul. Nowogrodzka 62c, 02-002 Warszawa
tel. +48 22 203 56 00 fax: +48 22 203 56 01
e-mail: biuro@sages.com.pl web: www.sages.com.pl

sages

Program szkolenia:

1. Koncepcja wyszukiwania pełnotekstowego
    I. Idea wyszukiwania pełnotekstowego (full-text search, FTS)
    II. Gdzie przydatne jest wyszukiwanie pełnotekstowe?
        i. Strony WWW
        ii. Aplikacje webowe
        iii. Bazy danych
        iv. Poczta e-mail
2. Wyszukiwanie pełnotekstowe w szczegółach, typowe problemy
    I. Użycie LIKE
    II. Realizacja
        i. Stop-słowa
        ii. Lematyzacja
        iii. Stemming
        iv. Morfologia języków
        v. Indeksy TF-IDF
        vi. Ocena dopasowania szukanego tekstu do tekstu znalezionego
        vii. Wyszukiwanie aproksymacyjne
        viii. Indeksy w bazie danych
            A. Konfiguracja indeksów w bazie danych
            B. Miejsce przetrzymywania indeksów
            C. Wydajne indeksowanie
            D. Utrzymywanie indeksów
            E. Transakcyjność
    III. Architektura separacji odpowiedzialności komend i zapytań (Command Query Responsibility Separation - CQRS)
    IV. Rozproszenie wyszukiwania
    V. Przeszukiwanie pełnotekstowe w plikach różnego formatu (PDF, XML, MS Office, ...)
    VI. Skalowalność
3. Wsparcie wyszukiwania pełnotekstowego na poziomie źródeł danych
    I. Opensource'owe bazy danych
        i. MySQL
            A. Możliwosci Fulltext
            B. Realizacja
                @. Tabele typu MyISAM
                @. Wyszukiwanie z rozwijaniem zapytania, MATCH
        ii. PostgreSQL
            A. Możliwości
            B. Realizacja
                @. TSVECTOR column
                @. Dedykowane indeksy
                    a. GIN
                    b. GiST
                @. Utrzymywanie kolumny TSVECTOR

II. Komercyjne bazy danych
    i. MS SQL Server
        A. Możliwości
        B. Realizacja
            @. Reguły językowe
            @. Dedykowany serwis
            @. Tworzenie katalogów
            @. Tworzenie indeksów unikalnych i FULLTEXT
            @. Utrzymywanie indeksów FULLTEXT
            @. Stoplisty
            @. Synonimy
            @. Zadawanie zapytań
                a. CONTAINS
                b. FREETEXT
                c. CONSTAINSTABLE
                d. FREETEXTTABLE
            @. Rodzaje wyrażeń
                a. Proste
                b. Oparte o prefixowanie
                c. Z wariantami
                d. Oparte o bliskość elementów
                e. Ważone
        C. Dostępność w produktach Microsoft Exchange i Microsoft SharePoint
        D. Zalety i wady
    ii. Oracle
        A. Możliwości Oracle Text
        B. Realizacja
            @. Typy indeksów
                a. Standard
                b. Catalog
                c. Classification
                d. Substring
                e. Prefix
    iii. IBM DB2
III. NoSQL

4. Wspracie wyszukiwania na poziomie dedykowanych bibliotek programistycznych i serwerów
    I. Apache Lucene
        i. Możliwości
        ii. Realizacja
            A. Indeksowanie
                @. Elementy
                    a. Index
                    b. Document
                    c. Field

- @. Klasy
  - a. IndexWriter
  - b. Directory
  - c. Analyzer
  - d. Document
  - e. Field
- @. Możliwości przetrzymywania indeksów (plik, pamięć, dowolna baza danych)
- @. Utrzymywanie indeksów, strategie odświeżania indeksów

B. Wyszukiwanie
- @. Klasy
  - a. IndexReader
  - b. IndexSearcher
  - c. Query
  - d. QueryParser
  - e. TopDocs/ScoreDocs
  - f. Document
- @. Typy zapytań (Term, Range, Prefix, Boolean, Phrase, WilCard, Fuzzy)

C. Narzędzie do inspekcji indeksów - Luke

II. Apache Solr
- i. Relacja z Apache Lucene
- ii. Możliwości
- iii. Przykłady wdrożeń
- iv. Konfiguracja (Schema.xml, Solrconfig.xml)
- v. Realizacja
  - A. Architektura
  - B. Dokumenty
  - C. Indexy
  - D. Pola dynamiczne
  - E. Pluginy
  - F. Faceted search
    - @. Wprowadzenie
    - @. Przykłady zastosowania
    - @. Realizacja

III. Sphinx Search
- i. Możliwości
- ii. Realizacja
  - A. SphinxAPI, SphinxQL,
  - B. Główne moduły
    - @. Indexer
    - @. Search
    - @. Searchd
    - @. Sphinxapi
  - C. Konfiguracja – plik sphinx.conf

D. Budowanie indexów
E. Zapytania

IV. Hibernate Search
    i. Możliwości
    ii. Realizacja
        A. Integracja z Apache Lucene
        B. Konfiguracja: hiberante.cfg.xml i hibernate.properties, API do konfiguracji
        C. Architektura:
            @. Index manager
            @. Aktualizacja indeksu a transakcja
                a. Dwa tryby: no-scope i transactional
                b. Zalety trybu transactional: spełnienie ACID, wydajność
            @. Indeksy współdzielone, środowisko klastrowe
            @. Strategie czytania – współdzielone, niewspółdzielone (Reader Strategy)
            @. DirectoryProvider (przetrzymywanie indeksu w pamięci, w systemie plików)
            @. Ustawienia wpływające na wydajność (merge_factor, max_merge_docs itd.)
        D. Adnotacje Hibernate Search: Indexed, Analyzer, ClassBridge, DocumentId
        E. Mapowanie encji
        F. Hibernate Analyzer
        G. Typy zapytań: TermQuery, WildcardQuery, PrefixQuery, PhraseQuery i inne
        H. API do wyszukiwania: Query, QueryBuilder i inne
        I. Strategie przebudowywania indeksu

V. Inne
    i. Solandra
    ii. Elasticsearch
    iii. DataparkSearch
    iv. Ferret
    v. mnoGoSearch
    vi. Xapian
    vii. Google Custom Search Index