



# Dodatkowe funkcjonalności

# Co w module

- Co zrobić, by “szafa czarna” wyszukało także komody?
- Jak przeszukiwać produkty z USA (w \$) posługując się PLN?
- Analiza języka - czyli jak rozkładać go na części pierwsze, a nawet wykrywać czy to angielski czy francuski
- Komentarze w sklepie - jak zbudować strukturę, aby móc zagnieżdżać dokumenty?

# Synonymy

**Kupujemy szafę!**

# Zakup szafy

- Zakupy: Allegro. Kupujemy szafę.

A screenshot of a search bar from the Allegro website. The search bar is a horizontal rectangle with a blue border. It contains a search input field with the text "szafa", a green "szukaj wielu" button, a dropdown menu showing "Wszystkie kategorie" with a downward arrow, and an orange "SZUKAJ" button.

szafa	szukaj wielu	Wszystkie kategorie ▼	SZUKAJ
-------	--------------	-----------------------	--------

- **PROBLEM:**  
Ale jakiej szafy? Komody? Regału? Kredensu?

# Zakup szafy

**Rozwiązanie:** odpowiednie nazewnictwo produktu.



Promowane ⓘ

SZAFA GARDEROBA S 90cm 2D 2SZ BIAŁA NOWOCZESNA

od ⓘ Super Sprzedawcy

Stan: Nowy Materiał: płyta laminowana Głębokość mebla: 51 cm Szerokość mebla: 90 cm

**516,13 zł**

419,62 zł netto

60 dni na spłatę

616,13 zł z dostawą

Produkt: Szafa Akord S 90 x 180 x ...

dostawa pojutrze



Promowane ⓘ

**Szafa Przesuwana VISTA 05 150 Lustro + półki**

Stan: Nowy Materiał: płyta laminowana Głębokość mebla

**889,00 zł**

722,76 zł netto

60 dni na spłatę

Produkt: Szafa Idźczak Meble Vista...

1089,00 zł z dostawą



Promowane ⓘ

Szafa, Regał zamknięty BIAŁY , LOREN, półka

od ⓘ Super Sprzedawcy

Stan: Nowy Materiał: płyta laminowana Głębokość m

**389,00 zł**

60 dni na spłatę

1 Moneta

449,00 zł z dostawą

# Lepsze rozwiązanie: **SYNONIMY**

# Synonimy

## Czym są?

### Dwie definicje podane przez SJP:

1. Wyraz mający **takie samo lub podobne** znaczenie co inny wyraz, np. auto i samochód; bliskoznacznik, wyraz bliskoznacznym;
2. Przenośnie: symbol, **odpowiednik czegoś**, np. Zachód jako synonim nowoczesności



# Co nam to da w Solr?

- Na etapie analizy możemy określić, że dane pole ma być przefiltrowane pod kątem synonimów.
- Dzięki temu możemy dodać *“komoda”* jako synonim *“szafa”*.
- **Efekt:** produkt o tytule *“Biała komoda”* może być wyszukany także, jeśli wpiszemy *“szafa”*

# Synonimy w Solr

## konfiguracja

1. Przygotuj plik z synonimami. Może być w głównym folderze lub w którymś z podfolderów.
2. Dodaj analizator (*analyzer*) do pola, które ma być “*zsynonimizowane*”.
  - SynonymGraphFilterFactory
  - ~~SynonymFilterFactory~~ (deprecated)
3. Zaindeksuj (/reindeksuj) pliki i przeszukuj z wykorzystaniem synonimów.

# Synonimy w Solr

## Plik

- Plik z synonimami musi być dostarczony w ramach konfiguracji, ale nie musi być w tym samym folderze
- Jest to plik \*.txt
- Dodajemy “gołe” synonimy, bez ubierania w jakikolwiek schemat (json/xml itd)
- Dwa rodzaje synonimów:
  - Ze znakiem “=>” - gdy od jednego słowa prowadzi do drugiego, ale bez wzajemności. Dobrze do wyłapywania literówek (np. “teh=>the”). **Uwaga! Finalnie oryginalny wyraz nie znajdzie się w wyniku.**
  - Oddzielone przecinkiem - gdy wszystkie są dla siebie nawzajem synonimami i finalnie wszystkie są przetwarzane (włącznie z oryginalnym tokenem)
- Oba rodzaje można mieszać w jednym pliku

# Synonimy w Solr

## Synonym Graph Filter Factory

- Filter, który pozwala nam dodać synonimy do analizatora.
- Ustawiamy go w managed-schema.
- Zastąpił on Synonym Filter Factory, który aktualnie jest *“deprecated”*.
- Jeśli używamy w procesie indexingu, musimy po nim umieścić Flatten Graph Filter, który spłaszczy nam tokeny

# Synonimy w Solr

## Najważniejsze parametry

- `synonyms` (wymagane) - ścieżka do pliku z synonimami.
- `ignoreCase` (domyślnie `false`) - jeśli `true`, synonimy będą dopasowane bez uwzględniania wielkości znaków.
- `expand` (domyślnie `true`) - jeśli `true`, tokeny będą rozszerzone o wszystkie synonimy. Jeśli `false`, będą zastąpione pierwszym z listy.

# Synonimy w Solr

## Przykład konfiguracji

```
<analyzer type="index">
  <tokenizer class="solr.StandardTokenizerFactory"/>
  <filter class="solr.SynonymGraphFilterFactory" synonyms="mysynonyms.txt"/>
  <filter class="solr.FlattenGraphFilterFactory"/> <!-- required on index analyzers after graph filters -->
</analyzer>
<analyzer type="query">
  <tokenizer class="solr.StandardTokenizerFactory"/>
  <filter class="solr.SynonymGraphFilterFactory" synonyms="mysynonyms.txt"/>
</analyzer>
```

```
couch,sofa,divan
teh => the
huge,ginormous,humungous => large
small => tiny,teeny,weeny
```

**Wejście:** "teh small couch"

**Tokeny:** "teh"(1), "small"(2), "couch"(3)

**Wyjście (po synonimach):**

"the"(1), "tiny"(2), "teeny"(2), "weeny"(2),  
"couch"(3), "sofa"(3), "divan"(3)

# Synonimy w Solr

## Przykład konfiguracji

polish\_synonyms.txt

```
sukces , wygrana , zwyciestwo  
espansja=>ekspansja
```

# Synonimy w Solr

## Przykład konfiguracji

```
<fieldType name="text_polish" class="solr.TextField" positionIncrementGap="100">
  <analyzer type="index">
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.StopFilterFactory"
      ignoreCase="true"
      words="lang/stopwords_pl.txt"
    />
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter class="solr.SynonymGraphFilterFactory" synonyms="polish_synonyms.txt"/>
    <filter class="solr.FlattenGraphFilterFactory"/>
    <filter class="solr.StempelPolishStemFilterFactory"/>
  </analyzer>
  <analyzer type="query">
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.StopFilterFactory"
      ignoreCase="true"
      words="lang/stopwords_pl.txt"
    />
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter class="solr.SynonymGraphFilterFactory" synonyms="polish_synonyms.txt"/>
    <filter class="solr.StempelPolishStemFilterFactory"/>
  </analyzer>
</fieldType>
```



# Synonimy w Solr

## Przykład konfiguracji

q=description\_polish:wygrana

hl=true

hl.fl=description\_polish

```
Response: { numFound: 1, start: 0, numDocuments: 1, docs: [
  {
    "id": "5",
    "title": "Iverson Zycie to nie gra",
    "suggest_field": ["Iverson Zycie to nie gra",
      "Niesamowity sukces i niewiarygodny upadek złotego dziecka koszykówki\n\nSyn alkoholiczki i recydywisty, który życia uczył si
    "author": "Kent Babb",
    "price___s_ns": "PLN",
    "price": "98,PLN",
    "description_polish": ["Niesamowity sukces i niewiarygodny upadek złotego dziecka koszykówki\n\nSyn alkoholiczki i recydywisty,
    "title_polish": ["Iverson Zycie to nie gra"],
    "_version_": 1758360631374774272}}
],
"highlighting": {
  "5": {
    "description_polish": ["Niesamowity <em>sukces</em> i niewiarygodny upadek złotego dziecka koszykówki\n\nSyn alkoholiczki i"]}}
```

# Synonimy w Solr

## Narzędzie “Analysis”

Field Value (Index)  
espansja wygrana

Analyse Fieldname / FieldType:  [? Schema Browser](#)

	Field	Value	Field	Value
ST	text	espansja	wygrana	
	raw_bytes	[65 73 70 61 6e 73 6a 61]	[77 79 67 72 61 6e 61]	
	start	0	9	
	end	8	16	
	positionLength	1	1	
	type	<ALPHANUM>	<ALPHANUM>	
	termFrequency	1	1	
	position	1	2	
SF	text	espansja	wygrana	
	raw_bytes	[65 73 70 61 6e 73 6a 61]	[77 79 67 72 61 6e 61]	
	start	0	9	
	end	8	16	
	positionLength	1	1	
	type	<ALPHANUM>	<ALPHANUM>	
	termFrequency	1	1	
	position	1	2	
LCF	text	espansja	wygrana	
	raw_bytes	[65 73 70 61 6e 73 6a 61]	[77 79 67 72 61 6e 61]	
	start	0	9	
	end	8	16	
	positionLength	1	1	
	type	<ALPHANUM>	<ALPHANUM>	
	termFrequency	1	1	
	position	1	2	

# Synonimy w Solr

## Narzędzie "Analysis"

LCF	text	expansja	wygrana		
	raw_bytes	[65 73 70 61 6e 73 6a 61]	[77 79 67 72 61 6e 61]		
	start	0	9		
	end	8	16		
	positionLength	1	1		
	type	<ALPHANUM>	<ALPHANUM>		
	termFrequency	1	1		
	position	1	2		
SGF	text	ekspansja	sukces	zwyciestwo	wygrana
	raw_bytes	[65 6b 73 70 61 6e 73 6a 61]	[73 75 6b 63 65 73]	[7a 77 79 63 69 65 73 74 77 6f]	[77 79 67 72 61 6e 61]
	start	0	9	9	9
	end	8	16	16	16
	positionLength	1	1	1	1
	type	SYNONYM	SYNONYM	SYNONYM	<ALPHANUM>
	termFrequency	1	1	1	1
	position	1	2	2	2
FGF	text	ekspansja	sukces	zwyciestwo	wygrana
	raw_bytes	[65 6b 73 70 61 6e 73 6a 61]	[73 75 6b 63 65 73]	[7a 77 79 63 69 65 73 74 77 6f]	[77 79 67 72 61 6e 61]
	start	0	9	9	9
	end	8	16	16	16
	positionLength	1	1	1	1
	type	SYNONYM	SYNONYM	SYNONYM	<ALPHANUM>
	termFrequency	1	1	1	1
	position	1	2	2	2
SF	text	ekspansja	sukces	zwyciestwo	wygrać
	raw_bytes	[65 6b 73 70 61 6e 73 6a 61]	[73 75 6b 63 65 73]	[7a 77 79 63 69 65 73 74 77 6f]	[77 79 67 72 61 c4 87]
	start	0	9	9	9
	end	8	16	16	16
	positionLength	1	1	1	1
	type	SYNONYM	SYNONYM	SYNONYM	<ALPHANUM>
	termFrequency	1	1	1	1
	position	1	2	2	2
	keyword	false	false	false	false

# Waluty

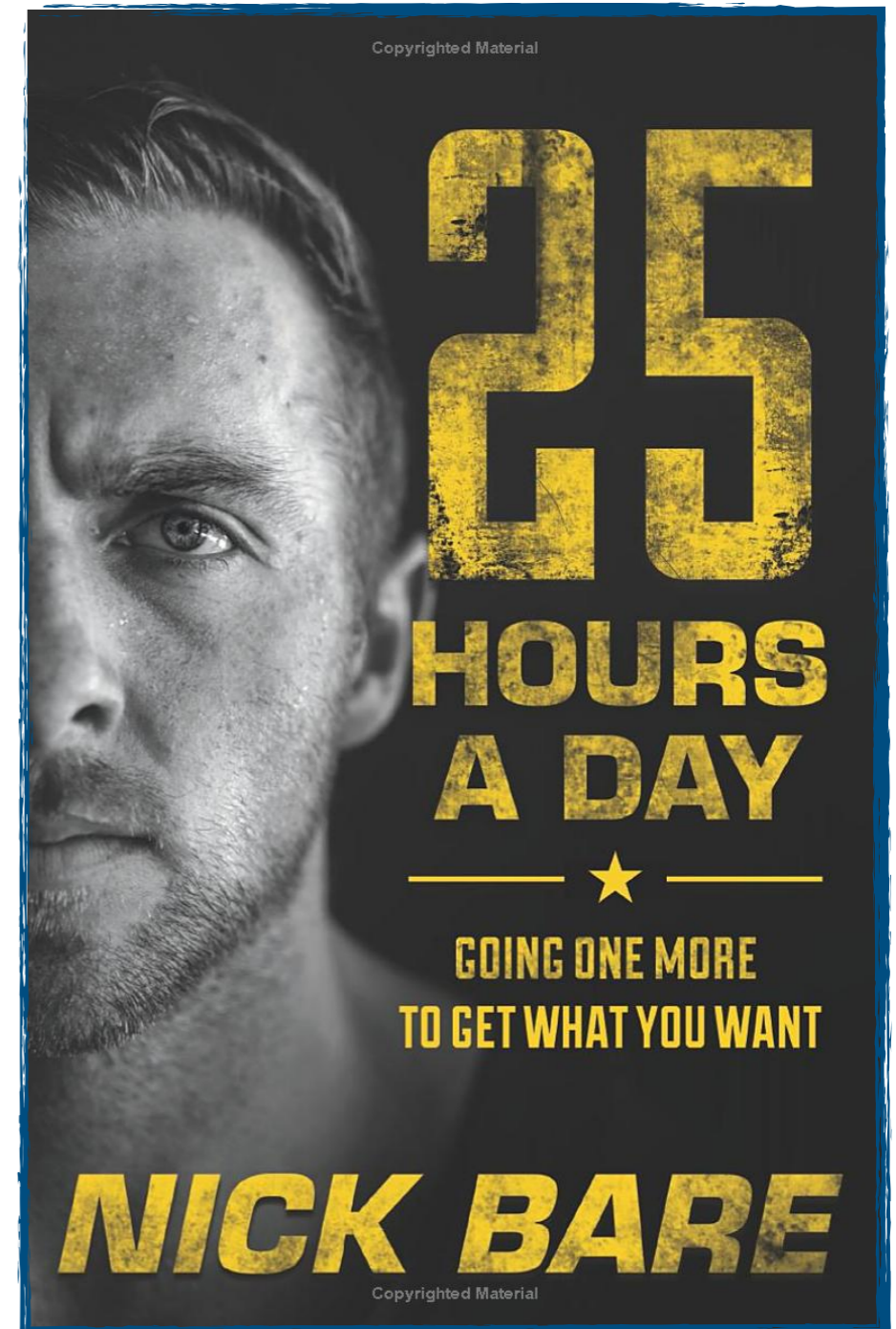
# Międzynarodowy Sklep Internetowy

## Wyszukiwanie produktu

# Problem

## Szukamy książki

*“25 hours a Day: Going one more to get what you want”*



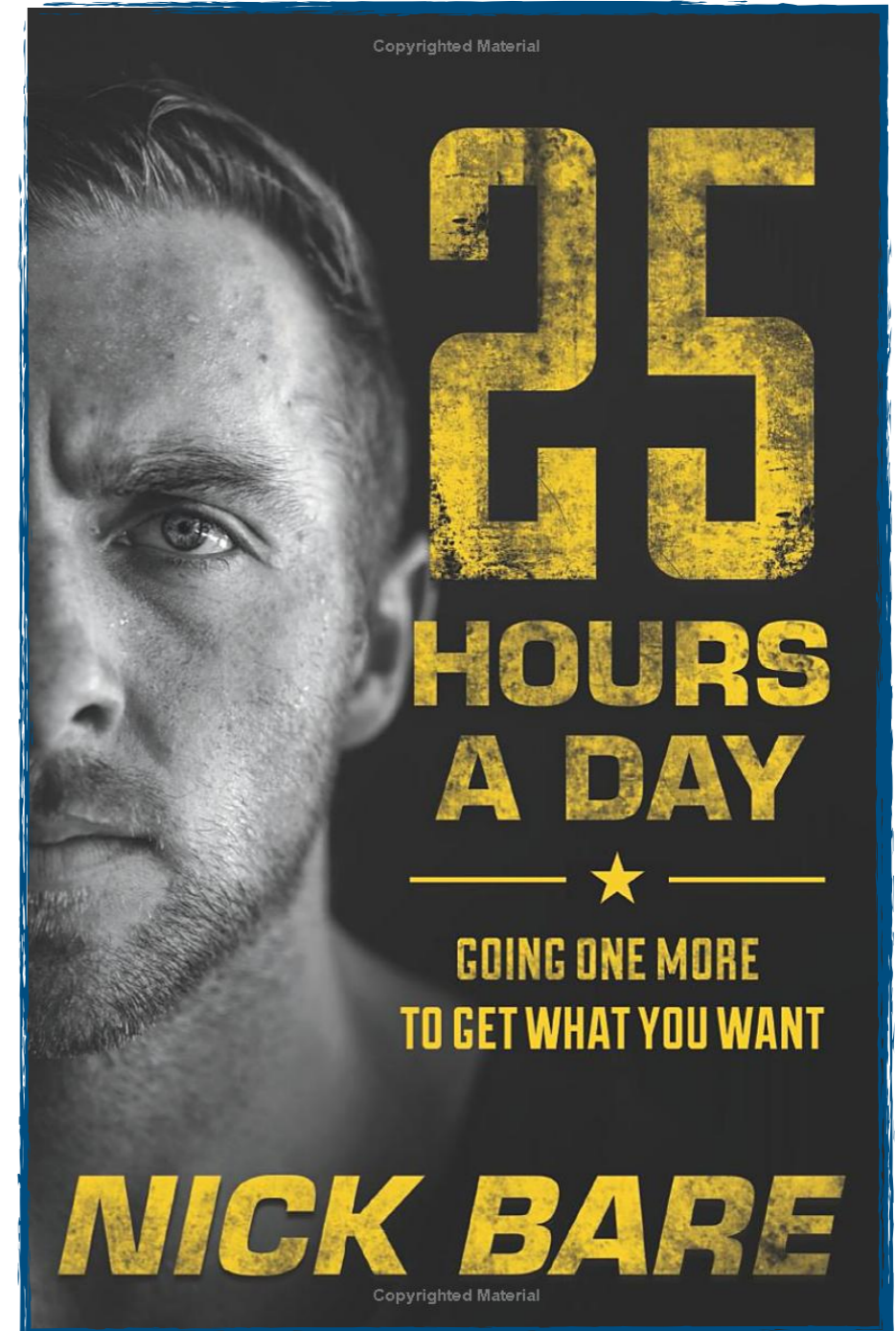
# Problem

**Szukamy książki**


*“25 hours a Day: Going one more to get what you want”*

*Możemy przeznaczyć  
maksymalnie*

**60 zł**



# Problem


**Buy new:** \$15.99 


No Import Fees Deposit & \$12.23  
Shipping to Poland [Details](#) ▾

**Delivery Thursday, August 24.**  
Order within **21 hrs 37 mins**



# Problem

**Buy new:** \$15.99 

No Import Fees Deposit & \$12.23  
Shipping to Poland [Details](#) 

Delivery **Thursday, August 24.**  
Order within 21 hrs 37 mins

**WYSTARCZY 60 PLN?**

# Problem

**Podstawowy problem:**  
Ile PLN to 15.99\$ ?

# Problem

**Podstawowy problem:**

Ile PLN to 15.99\$ ?

**Rozszerzony problem:**

Jak znaleźć książki w \$ poniżej 60 PLN?

# Rozwiązanie

Solr oferuje możliwość wymiany walut  
**podczas przeszukiwania**

# Implementacja walut

- Waluty powinny mieć swój własny typ.
- Można je dodać z grubsza na dwa sposoby
  - Korzystając z własnego pliku wymiany walut
  - Korzystając z portalu OpenExchangeRatesOrgProvider, dzięki któremu będzie można na bieżąco sprawdzać kursy walut (opcja płatna powyżej 1000 requestów/miesiąc)
- Przy dodawaniu danych, całość od setnych oddzielamy kropką, a wartość od nazwy waluty oddzielamy przecinkiem, np: “10.23,PLN”

# Waluty

## Korzystanie z własnego pliku

1. Określ typ w managed-schema:

```
<fieldType name="currency" class="solr.CurrencyFieldType"
amountLongSuffix="_l_ns" codeStrSuffix="_s_ns" defaultCurrency="PLN"
currencyConfig="currency.xml" />
```

2. Określ pole, które będzie przechowywało pieniądze.

```
<field name="price" type="currency" indexed="true" stored="true"
required="true" multiValued="false" />
```

3. Określ dwa pola pomocnicze - jedno na walutę, drugie na wartość.

```
<dynamicField name="*_l_ns" type="long" indexed="true" stored="true"
multiValued="false" />
<dynamicField name="*_s_ns" type="string" indexed="true" stored="true"
multiValued="false" />
```

4. Stwórz plik z wymianą walut.

# Waluty

## Korzystanie z własnego pliku

Plik z wymianą walut  
currency.xml

```
<currencyConfig version="1.0">
  <rates>
    <!-- Updated from http://www.exchangerate.com/ at 2011-09-27 -->
    <rate from="USD" to="ARS" rate="4.333871" comment="ARGENTINA Peso" />
    <rate from="USD" to="AUD" rate="1.025768" comment="AUSTRALIA Dollar" />
    <rate from="USD" to="EUR" rate="0.743676" comment="European Euro" />
    <rate from="USD" to="CAD" rate="1.030815" comment="CANADA Dollar" />

    <!-- Cross-rates for some common currencies -->
    <rate from="EUR" to="GBP" rate="0.869914" />
    <rate from="EUR" to="NOK" rate="7.800095" />
    <rate from="GBP" to="NOK" rate="8.966508" />

    <!-- Asymmetrical rates -->
    <rate from="EUR" to="USD" rate="0.5" />
  </rates>
</currencyConfig>
```

# Waluty

## Korzystanie Open Exchanges Rates

1. Jest portal, na którym na bieżąco można sprawdzać wymiany walut: [www.openexchangerates.org](http://www.openexchangerates.org)
2. Można stworzyć swoje konto i podpiąć się pod API
3. Po uzyskaniu tokenu (App ID), możemy użyć API OER w naszej konfiguracji.
4. UWAGA! Token jest płatny, ale do 1000 requestów miesięcznie jest free.



# Waluty

managed-schema:

```
<fieldType name="currency" class="solr.CurrencyFieldType" amountLongSuffix="_l_ns" codeStrSuffix="_s_ns"  
  defaultCurrency="PLN" providerClass="solr.OpenExchangeRatesOrgProvider" refreshInterval="60" ratesFileLocation="http://www.openexchangerates.org/api/latest.json?app_id=86f0700e89b14a109f382391c  
1cc40de" />
```

# Waluty

q=\*:\*

```
"response": {"numFound": 6, "start": 0, "numFoundExact": true, "docs": [
  {
    "id": "1",
    "title": "25 hours a day",
    "author": "Nick Bare",
    "price__s_ns": "PLN",
    "price": "43,PLN",
    "description": "You have big dreams that fire you up, and yet a fear of failure is holding you back. You see the success others have achieved",
    "category": "Biznes",
    "language": "en",
    "description_english": ["You have big dreams that fire you up, and yet a fear of failure is holding you back. You see the success others have",
    "title_english": ["25 hours a day"],
    "languages": ["en"],
    "_version_": 1757942012577841152},
  {
    "id": "2",
    "title": "Hooked: How to Build Habit-Forming Products",
    "author": "Nir Eyal",
    "price__s_ns": "PLN",
    "price": "59,PLN",
    "description": "Nir Eyal reveals how successful companies create products people can't put down - and how you can too\n\nWhy do some products",
    "category": "Biznes",
    "language": "en",
    "description_english": ["Nir Eyal reveals how successful companies create products people can't put down - and how you can too\n\nWhy do some",
    "title_english": ["Hooked: How to Build Habit-Forming Products"],
    "languages": ["en"],
    "_version_": 1757942012593569792},
  {
    "id": "3",
    "title": "Allgäu 2: Ostallgäu und vorderes Lechtal 55 Touren mit GPS-Tracks",
    "author": "Mark Zahel",
    "price__s_ns": "PLN",
    "price": "69,PLN",
    "description": "Die Schlösser Neuschwanstein und Hohenschwangau, Bergseen und Weiher, die Wieskirche und die Altstadt von Füssen, sanfte Al",
    "category": "Poradniki",
    "language": "de",
    "description_german": ["Die Schlösser Neuschwanstein und Hohenschwangau, Bergseen und Weiher, die Wieskirche und die Altstadt von Füssen, sa",
    "title_german": ["Allgäu 2: Ostallgäu und vorderes Lechtal 55 Touren mit GPS-Tracks"],
    "languages": ["de"],
    "_version_": 1757942012603006976},
  {
    "id": "4",
```

# Waluty

q=price: [50,PLN TO \*]

```
"response":{"numFound":4,"start":0,"numFoundExact":true,"docs":[{"id":"2","title":"Hooked: How to Build Habit-Forming Products","author":"Nir Eyal","price__s_ns":"PLN","price":"59,PLN","description":"Nir Eyal reveals how successful companies create products people can't put down - and how you can too\n\nWhy do some products","category":"Biznes","language":"en","description_english":["Nir Eyal reveals how successful companies create products people can't put down - and how you can too\n\nWhy do some","title_english":["Hooked: How to Build Habit-Forming Products"],"languages":["en"],"_version_":1757942012593569792}, {"id":"3","title":"Allgäu 2: Ostallgäu und vorderes Lechtal 55 Touren mit GPS-Tracks","author":"Mark Zahel","price__s_ns":"PLN","price":"69,PLN","description":"Die Schlösser Neuschwanstein und Hohenschwangau, Bergseen und Weiher, die Wieskirche und die Altstadt von Füssen, sanfte Alm","category":"Poradniki","language":"de","description_german":["Die Schlösser Neuschwanstein und Hohenschwangau, Bergseen und Weiher, die Wieskirche und die Altstadt von Füssen, sa","title_german":["Allgäu 2: Ostallgäu und vorderes Lechtal 55 Touren mit GPS-Tracks"],"languages":["de"],"_version_":1757942012603006976}, {"id":"4","title":"Karate Shorin-ryu (deluxe edition): L'eredità delle guardie del re di Okinawa: 1","author":"Emanuel Giordano","price__s_ns":"PLN","price":"212,PLN","description":"Lo Shorin-ryu è lo stile di Karate più antico esistente. Da questo stile marziale, adatto a chiunque, e basato su movimenti m","category":"Sport","language":"it","description_italian":["Lo Shorin-ryu è lo stile di Karate più antico esistente. Da questo stile marziale, adatto a chiunque, e basato su mo","title_italian":["Karate Shorin-ryu (deluxe edition): L'eredità delle guardie del re di Okinawa: 1"],"_version_":1757942012610347008}, {"id":"5"
```

# Waluty

q=price: [40,USD TO \*]

```
"_": "1676503899630"}},  
"response": {"numFound": 1, "start": 0, "numFoundExact": true, "docs": [  
  {  
    "id": "4",  
    "title": "Karate Shorin-ryu (deluxe edition): L'eredità delle guardie del re di Okinawa: 1",  
    "author": "Emanuel Giordano",  
    "price___s_ns": "PLN",  
    "price": "212,PLN",  
    "description": "Lo Shorin-ryu è lo stile di Karate più antico esistente. Da questo stile marziale,  
    "category": "Sport",  
    "language": "it",  
    "description_italian": ["Lo Shorin-ryu è lo stile di Karate più antico esistente. Da questo stile  
    "title_italian": ["Karate Shorin-ryu (deluxe edition): L'eredità delle guardie del re di Okinawa:  
    "languages": ["it"],  
    "_version_": 1757942012610347008}]  
  }  
}]
```

# Analiza języka

# Problem

## **Szukane:**

“Szafa z nalepka”

## **Zaindeksowane:**

“Biała komoda z żółtą  
nalepką na drzwiach”

# Problem

**Szukane:**

“Szafa z nalepka”

**Zaindeksowane:**

“Biała komoda z żółtą  
nalepką na drzwiach”



**0 wyników**

# Rozwiązanie

- Potrzebna jest analiza języka
- Czasami nawet wykrycie języka!



# Elementy analizy języka

- ~~Synonimy~~ (Zaliczone;-))
- Stopwords
- Stemming i lemmatyzacja
- Zamiana znaków których nie ma na odpowiedniki w ASCII
- Phonetic Matching
- Wykrywanie języka
- I inne!

# Analiza języka

- Język analizujemy tak samo jak każdy inny tekst
- Potrzebujemy jednak wiedzieć, jaki język będzie znajdować się w danym polu.
- Co będzie przydatne:
  - Sprowadzenie do małych liter
  - Usunięcie stop-wordów
  - Stemming lub lemmatyzacja.

# Analiza języka

## Stopwords

- Stopwords to wyrazy, które są bardzo często spotykane, natomiast które nie stanowią realnej wartości.
- Jeśli je zostawimy na potrzeby analizy - mogą one zaburzyć scoring.
- Ponadto, usuwając je z analizy, oszczędzamy miejsce w indexie;-)
- Przykłady: “or”, “be”, “and”, “I”, “to”, “not”

# Analiza języka

## Stopwords

*“To be*

*or*

*not to be”*



# Analiza języka

## Stemming i lematyzacja

- **Stemming i Lematyzacja służą do tego samego** - sprowadzenia słów z tej samej rodziny do wspólnej postaci / wspólnej formy bazowej
- **Stemming - “obcinanie” do “korzenia”**. Taki “korzeń” nie musi mieć sensu ani poprawności gramatycznej.
  - Np. “Beautiful”, “Beautifully” => “Beauti”
- **Lematyzacja - sprowadzanie słowa do jego “rdzenia”**. Takim rdzeniem jest podstawowa forma słowa.
  - Np. “Beautiful”, “Beautifully” => “Beauty”

# Analiza języka

## Stemming

- Można utworzyć listę słów, które będą **chronione przed stemmingiem**. Aby to zrobić, należy utworzyć odpowiedni plik (protwords.txt) a następnie wykorzystać *KeywordMarkerFilterFactory*
- Przed stemmingiem można skopiować każdy token, aby znajdował się oryginalny oraz z **dodatkiem “KEYWORD”**. Dzięki temu zachowamy także niezestemowane słowa. Wykorzystujemy tu *KeywordRepeatFilterFactory*.
- Istnieje możliwość **nadpisania reguł stemmingu**. W tym celu tworzymy plik (stemdict.txt) i wykorzystujemy *StemmerOverrideFilterFactory*.
  - Np. “monkeys monkey”
  - Słowa oddzielamy tabem

# Analiza języka

## Dodatkowe mechanizmy

- Dzięki filtrowi ASCIIFoldingFilterFactory można przełożyć znaki których nie ma w pierwszych 127 znakach ASCII na takie, które są.
  - np. Rąbać => Rabac
- Phonetic Matching - możemy w Solr zastosować algorytmy phonetic matchingu, czyli wyszukiwania/analizy dwóch wyrazów, które są podobne w wymowie, ale różnią się w pisowni.
  - Np. “Steve”, “Stephe”.
  - Jest możliwość działania także w języku polskim, poprzez zastosowanie filtra Beider Morse Filter i wybrania języka polskiego.
  - Mechanizmy PM nie są dostępne defaultowo, należy je dołączyć jako osobne biblioteki.

# Analiza języka

## Beider-Morse Filter (phonetic matching)

### Przykładowa konfiguracja (Managed-schema)

```
<analyzer>  
  <tokenizer class="solr.StandardTokenizerFactory"/>  
  <filter class="solr.BeiderMorseFilterFactory" nameType="GENERIC" ruleType="APPROX" concat="true" languageSet="auto">  
  </filter>  
</analyzer>
```



# Analiza języka

## Polski

- Stemming - `StempelPolishStemFilterFactory`
- Lematyzacja - `MorphologikFilterFactory`
- Obie biblioteki dostępne w katalogu “contrib/analysis-extras”. Należy dołączyć je przed wykorzystaniem.

# Analiza języka

## Unicode Collation

- Unicode Collation to metoda sortowania Textu, która jest wrażliwa językowo (*language-sensitive*).
- Zamiast używać TextField, używamy typu ICUCollationField.
- Możemy wyszczególnić wiele języków oraz dialektów - np. Angielski brytyjski czy angielski amerykański, albo niemiecki szwajcarski itd.
- W managed-schema, definiując podajemy siłę (pole strength), z którą ma być sortowanie.
  - Mamy do wyboru primary, secondary, tertiary, quaternary, identical
  - Przykładowo primary oznacza że ignorujemy wszystkie różnice, secondary zignoruje wielkość znaków, ale uwzględni znaki diaktryczne.

# Analiza języka

## Unicode Collation - przykładowa konfiguracja

```
<fieldType name="polishCaseInsensitive" class="solr.ICUCollationField"  
  locale="pl_PL"  
  strength="secondary" />  
...  
<field name="city" type="text_general" indexed="true" stored="true"/>  
...  
<field name="city_sort" type="polishCaseInsensitive" indexed="true" stored="false"/>  
...  
<copyField source="city" dest="city_sort"/>
```

# Analiza języka

## Unicode Collation

- Można określać osobne sortowanie dla każdego z języków
- Możemy jednak określić sortowanie dla wszystkich języków jednocześnie. Jest to rozsądne przy potencjalnie wielu językach.
- Aby to zrobić, wpisujemy `/locale=""`, wybierając tryb Default/ROOT. Posiada on zasady działające dobrze dla większości języków.

```
<fieldType name="collatedROOT" class="solr.ICUCollationField"  
  locale=""  
  strength="primary" />
```

# Analiza języka

```
<fieldType name="text_polish" class="solr.TextField" positionIncrementGap="100">
  <analyzer>
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.StopFilterFactory"
      ignoreCase="true"
      words="lang/stopwords_pl.txt"
      />
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter class="solr.StempelPolishStemFilterFactory"/>
  </analyzer>
</fieldType>
```

# Wykrywanie języka

- Mamy możliwość automatycznego wykrywania i mapowania języka, dzięki mechanizmowi LangID.
- Wybieramy jedną z 3 dostępnych implementacji:
  - Tika
  - LangDetect
  - OpenNLP
- Dla ciekawskich: szczegółowe porównanie Tika i LangDetect znajdziemy tutaj: <https://blog.mikemccandless.com/2011/10/accuracy-and-performance-of-googles.html>

# Wykrywanie języka

## OpenNLP

- Mamy konieczność określenia modelu, który musimy dostarczyć (pole *langid.model*)

```
<processor class="org.apache.solr.update.processor.OpenNLPLangDetectUpdateProcessorFactory">
  <lst name="defaults">
    <str name="langid.fl">title,subject,text,keywords</str>
    <str name="langid.langField">language_s</str>
    <str name="langid.model">langdetect-183.bin</str>
  </lst>
</processor>
```

# Wykrywanie języka

## Podstawowe parametry

- `langid` - `true`, aby włączyć wykrywanie.
- `langid.fl` (wymagane) - lista pól, które będą badane.
- `langid.langField` (wymagane) - do tego pola spłynie język.
- `langid.langsField` - tu spłyną wszystkie języki (jeśli mamy włączone `langid.map.individual`).
- `langid.map.individual` - jeśli `true`, Solr wykryje dla każdego pola osobny język.



# Wykrywanie języka

## Parametry mapowania

- `langid.map` - true, aby zmapować.
- `langid.map.lcmap` - lista określająca mapowanie kodów językowych podczas mappingu. Można tu zamienić np. `en:english`, ale także sprowadzać konkretne dialekty, typu `en_GB:en` oraz `en_US:en`.
- `langid.map.keepOrigin` - zostawiamy także oryginalne pola.

# Wykrywanie języka

```
<doc>
  <field name="id">3</field>
  <field name="title">Allgäu 2: Ostallgäu und vorderes Lechtal 55 Touren mit GPS-Tracks</field>
  <field name="author">Mark Zahel</field>
  <field name="price">69</field>
  <field name="description">Die Schlösser Neuschwanstein und Hohenschwangau, Bergseen und Weiher, die Wieskirche und die Altstadt von Füssen, sanfte Almen und schroffe Gipfel – das Ostallgäu und das vordere Lechtal haben so viel zu bieten, dass man seine liebe Mühe hat, die Highlights zu benennen. In 55 vielfältigen Touren stellt der Rother Wanderführer »Allgäu 2« eine ganze Reihe davon näher vor. Die Spannweite der Touren ist weit gefächert: Malerische Seen im Ostallgäuer Alpenvorland locken, Paradegipfel wie Säuling, Thaneller und Aggenstein wollen erobert werden und wer die Einsamkeit sucht, wird an weniger bekannten Gipfeln wie Kienberg oder Leilachspitze fündig. Im Tiroler Außerfern geht es ins vordere Lechtal um Reutte, bis zu den Seitentälern von Hinterhornbach und Namlos und in das Tannheimer Tal. Viele der Touren führen an Berg- oder Alphütten vorbei, denn zu einer perfekten Wanderung gehört ja bekanntlich eine zünftige Einkehr. Tourensteckbriefe mit allen wichtigen Infos, ausführliche Wegbeschreibungen, aussagekräftige Höhenprofile und detaillierte Kartenausschnitte machen alle Touren leicht und sicher nachvollziehbar. Zudem stehen GPS-Tracks zum Download von der Internetseite des Rother Bergverlags bereit. Die farbige Schwierigkeitsbewertung erleichtert die Planung, und zahlreiche Farbfotos wecken die Lust, den Rucksack zu packen und loszuwandern.</field>
  <field name="category">Poradniki</field>
</doc>
```

```
<processor class="org.apache.solr.update.processor.LangDetectLanguageIdentifierUpdateProcessorFactory">
  <lst name="defaults">
    <str name="langid.fl">title,description</str>
    <str name="langid.langField">language</str>
    <str name="langid.langsField">languages</str>
    <str name="langid.map.individual">>true</str>
  </lst>
</processor>
```

# Wykrywanie i analiza języka

```
<doc>
  <field name="id">3</field>
  <field name="title">Allgäu 2: Ostallgäu und vorderes Lechtal 55 Touren mit GPS-Tracks</field>
  <field name="author">Mark Zahel</field>
  <field name="price">69</field>
  <field name="description">Die Schlösser Neuschwanstein und Hohenschwangau, Bergseen und Weiher, die Wieskirche und die Altstadt von Füssen, sanfte Almen und schroffe Gipfel – das Ostallgäu und das vordere Lechtal haben so viel zu bieten, dass man seine liebe Mühe hat, die Highlights zu benennen. In 55 vielfältigen Touren stellt der Rother Wanderführer »Allgäu 2« eine ganze Reihe davon näher vor. Die Spannweite der Touren ist weit gefächert: Malerische Seen im Ostallgäuer Alpenvorland locken, Paradegipfel wie Säuling, Thaneller und Aggenstein wollen erobert werden und wer die Einsamkeit sucht, wird an weniger bekannten Gipfeln wie Kienberg oder Leilachspitze fündig. Im Tiroler Außerfern geht es ins vordere Lechtal um Reutte, bis zu den Seitentälern von Hinterhornbach und Namlos und in das Tannheimer Tal. Viele der Touren führen an Berg- oder Alphütten vorbei, denn zu einer perfekten Wanderung gehört ja bekanntlich eine zünftige Einkehr. Tourensteckbriefe mit allen wichtigen Infos, ausführliche Wegbeschreibungen, aussagekräftige Höhenprofile und detaillierte Kartenausschnitte machen alle Touren leicht und sicher nachvollziehbar. Zudem stehen GPS-Tracks zum Download von der Internetseite des Rother Bergverlags bereit. Die farbige Schwierigkeitsbewertung erleichtert die Planung, und zahlreiche Farbfotos wecken die Lust, den Rucksack zu packen und loszuwandern.</field>
  <field name="category">Poradniki</field>
</doc>
```

```
<updateRequestProcessorChain name="langid">
  <processor class="org.apache.solr.update.processor.LangDetectLanguageIdentifierUpdateProcessorFactory" runtimeLib="true">
    <lst name="defaults">
      <str name="langid.fl">title,description</str>
      <str name="langid.langField">language</str>
      <str name="langid.langsField">languages</str>
      <str name="langid.map.individual">true</str>
      <str name="langid.map">true</str>
      <str name="langid.map.lcmap">en:english,de:german,pl:polish,it:italian</str>
      <str name="langid.map.keepOrig">true</str>
    </lst>
  </processor>
  <processor class="solr.LogUpdateProcessorFactory" />
  <processor class="solr.RunUpdateProcessorFactory" />
</updateRequestProcessorChain>
```

# Wykrywanie i analiza języka

```
<fieldType name="text_english" class="solr.TextField" positionIncrementGap="100">
  <analyzer>
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.StopFilterFactory"
      ignoreCase="true"
      words="lang/stopwords_en.txt"
    />
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter class="solr.EnglishPossessiveFilterFactory"/>
    <filter class="solr.KeywordMarkerFilterFactory" protected="protwords.txt"/>
    <filter class="solr.KStemFilterFactory"/>
  </analyzer>
</fieldType>

<fieldType name="text_polish" class="solr.TextField" positionIncrementGap="100">
  <analyzer>
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.StopFilterFactory"
      ignoreCase="true"
      words="lang/stopwords_pl.txt"
    />
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter class="solr.StempelPolishStemFilterFactory"/>
  </analyzer>
</fieldType>
```

```
<fieldType name="text_german" class="solr.TextField" positionIncrementGap="100">
  <analyzer>
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.StopFilterFactory"
      ignoreCase="true"
      words="lang/stopwords_de.txt"
    />
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter class="solr.GermanStemFilterFactory"/>
  </analyzer>
</fieldType>

<fieldType name="text_italian" class="solr.TextField" positionIncrementGap="100">
  <analyzer>
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.StopFilterFactory"
      ignoreCase="true"
      words="lang/stopwords_it.txt"
    />
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter class="solr.ItalianLightStemFilterFactory"/>
  </analyzer>
</fieldType>
```

# Wykrywanie i analiza języka

```
<field name="id" type="string" indexed="true" stored="true" required="true" multiValued="false" />
<uniqueKey>id</uniqueKey>
<field name="title" type="text_short" indexed="true" stored="true" required="false" multiValued="false" />
<field name="author" type="text_short" indexed="true" stored="true" required="true" multiValued="false" />
<field name="price" type="currency" indexed="true" stored="true" required="true" multiValued="false" />
<field name="description" type="text_general" indexed="true" stored="true" required="false" multiValued="false" />
<field name="category" type="text_short" indexed="true" stored="true" required="true" multiValued="false" />
<field name="language" type="string" indexed="true" stored="true" />
<field name="languages" type="string" indexed="true" stored="true" multiValued="true"/>

<field name="_version_" type="long" indexed="false" stored="false"/>

<dynamicField name="*_l_ns" type="long" indexed="true" stored="true" multiValued="false" />
<dynamicField name="*_s_ns" type="string" indexed="true" stored="true" multiValued="false" />
<dynamicField name="*_english" type="text_english" indexed="true" stored="true" multiValued="true"/>
<dynamicField name="*_italian" type="text_italian" indexed="true" stored="true" multiValued="true"/>
<dynamicField name="*_german" type="text_german" indexed="true" stored="true" multiValued="true"/>
<dynamicField name="*_polish" type="text_polish" indexed="true" stored="true" multiValued="true"/>
```

# Wykrywanie i analiza języka

```
{  
  "id": "1",  
  "title": "25 hours a day",  
  "author": "Nick Bare",  
  "price___s_ns": "PLN",  
  "price": "43,PLN",  
  "description": "You have big dreams that fire you up, and yet a fear of failure",  
  "category": "Biznes",  
  "language": "en",  
  "description_english": ["You have big dreams that fire you up, and yet a fear of failure"],  
  "title_english": ["25 hours a day"],  
  "languages": ["en"],  
  "_version_": 1757942
```

```
{  
  "id": "5",  
  "title": "Iverson Zycie to nie gra",  
  "author": "Kent Babb",  
  "price___s_ns": "PLN",  
  "price": "98,PLN",  
  "description": "Niesamowity sukces i niewiarygodny upadek złotego dziecka koszykówki\n\nSyn alkoholicki",  
  "category": "Sport",  
  "language": "pl",  
  "description_polish": ["Niesamowity sukces i niewiarygodny upadek złotego dziecka koszykówki\n\nSyn alkoholicki"],  
  "title_polish": ["Iverson Zycie to nie gra"],  
  "languages": ["pl"],  
  "_version_": 1757942012615589888},
```

# Wykrywanie i analiza języka

## Warto wiedzieć...

- “Langid” to UpdateRequestProcessor. Od kiedy są dodawane w chainie (z LogUpdateProcessorFactory oraz RunUpdateProcessorFactory), należy zaznaczyć właściwą opcję przy requeście.
- Aby się tym nie przejmować, warto dodać “update.chain” w initParams w solrconfig.xml

```
<initParams path="/update/**,/query,/select,/spell">  
  <lst name="defaults">  
    <str name="df">_text_</str>  
    <str name="update.chain">langid</str>  
  </lst>  
</initParams>
```

# Wykrywanie i analiza języka

## Warto wiedzieć...

- Langid oraz analiza języka korzysta z licznych bibliotek. Należy sprawić, aby były one dostępne.
- Jeden ze sposobów do dodanie opcji “<lib dir>” na początku solr config.

```
<lib dir="/solr/solr-8.11.2/dist" regex=".*\.jar"/>  
<lib dir="/solr/solr-8.11.2/contrib/langid/lib" regex=".*\.jar"/>  
<lib dir="/solr/solr-8.11.2/contrib/analysis-extras/lucene-libs" regex=".*\.jar"/>  
<lib dir="/solr/solr-8.11.2/contrib/analysis-extras/lib" regex=".*\.jar"/>
```



# Zagnieżdżone dokumenty

# Klasyczny blog z przepisami

- Tytuł
- Wstępny opis
- Czas
- Składniki
- Przepis
- ...?

# Klasyczny blog z przepisami

- Tytuł
- Wstępny opis
- Czas
- Składniki
- Przepis
- Komentarze!

# Jak dodać komentarze?

- Można dodać jako osobną kolekcję
- Ale lepiej wykorzystać mechanizm zagnieżdżonych dokumentów!

# Zagnieżdżone dokumenty

- **Zła wiadomość:** Lucene w ogóle nie wspiera zagnieżdżeń.

# Zagnieżdżone dokumenty

- **Zła wiadomość:** Lucene w ogóle nie wspiera zagnieżdżeń.
- **Dobra wiadomość:** twórcy Solr sobie z tym poradzili.

# Zagnieżdżone dokumenty

- **Zła wiadomość:** Lucene w ogóle nie wspiera zagnieżdżeń.
- **Dobra wiadomość:** twórcy Solr sobie z tym poradzili.
- **Jeszcze lepsza:** nie trzeba wiele robić, żeby zaindeksować zagnieżdżenia.

# Zagnieżdżone dokumenty

## Co trzeba zrobić?

- managed-schema
  - Pole “\_root\_” (string, musi być indeksowane)
  - Wszystkie pola, które występują w dokumencie “rodzicu” lub “dziecku” (warto przemyśleć, które są wymagane!)
- solrconfig.xml
  - Cache “PerSegFilter” - wprowadza możliwość funkcjonalności zagnieżdżonych elementów (dostępne w defaultowym configu oraz “configu RDF”)
- Dane
  - Zagnieżdżone dokumenty muszą mieć swoje unikalne ID



# Zagnieżdżone dokumenty

## Co warto zrobić? (managed-schema)

- Pole “*\_nest\_path\_*”
  - Solr automatycznie przekazuje to na każde dziecko (ale nie na rodzica)
  - Pozwala na prawidłowe identyfikowanie relacji przy używaniu Child Doc Transformer.
  - Jeśli nie używamy, rekomendowane jest używanie pola, które odróżni dziecko od rodzica.
- Pole “*\_nest\_parent\_*” - automatycznie przechowuje ID rodzica.

# Zagnieżdżone dokumenty

## Pole “\_nest\_path\_”

```
<fieldType name="_nest_path_" class="solr.NestPathField" />  
<field name="_nest_path_" type="_nest_path_" />
```

# Zagnieżdżone dokumenty

## Pole “\_nest\_parent\_”

```
<field name="_nest_parent_" type="string" indexed="true" stored="true" />
```

```
{  
  "content": "\"this is pretty poor position! 2/10. Boring!\"",  
  "id": "1/comments#1",  
  "_nest_parent_": "1",  
  "_version_": 1758065021458317312},  
{
```

# Zagnieżdżone dokumenty

```
<cache name="perSegFilter"  
  class="solr.search.LRUCache"  
  size="10"  
  initialSize="0"  
  autowarmCount="10"  
  regenerator="solr.NoOpRegenerator" />
```

# Zagnieżdżone dokumenty

## Identyfikacja

- Solr wymaga, aby każdy dokument-dziecko miał unikatowe ID **w skali indexu (!)**
- Jeśli nie zapewnimy takowego, Solr utworzy swoje.
- Zalecane jest zapewnianie ID w formacie compositeld. Np. Dla rodzica "Post" Id="P15", dzieci "comments" mogłyby być;
  - P15!C1
  - P15!C2
  - P15!C3

# Zagnieżdżone dokumenty

## Praktyka

```
<?xml version="1.0" encoding="UTF-8" ?>
<schema name="books-simple-schema" version="1.6">
  <field name="id" type="string" indexed="true" stored="true" required="true" multiValued="false" />
  <uniqueKey>id</uniqueKey>
  <field name="title" type="text_short" indexed="true" stored="true" required="false" multiValued="false" />
  <field name="author" type="text_short" indexed="true" stored="true" required="false" multiValued="false" />
  <field name="price" type="currency" indexed="true" stored="true" required="false" multiValued="false" />
  <field name="description" type="text_general" indexed="true" stored="false" required="false" multiValued="false" />
  <field name="category" type="text_short" indexed="true" stored="false" required="false" multiValued="false" />
  <field name="language" type="string" indexed="true" stored="false" />
  <field name="languages" type="string" indexed="true" stored="false" multiValued="true"/>

  <field name="_root_" type="string" indexed="true" stored="false" docValues="false" />
  <field name="_nest_path_" type="_nest_path_" />
  <field name="_nest_parent_" type="string" indexed="true" stored="true" />
  <field name="content" type="string" indexed="true" stored="true" required="false"/>

  <field name="_version_" type="long" indexed="false" stored="false"/>
</schema>
```

# Zagnieżdżone dokumenty

## Praktyka

```
<add>
  <doc>
    <field name="id">1</field>
    <field name="title">25 hours a day</field>
    <field name="author">Nick Bare</field>
    <field name="price">43</field>
    <field name="description">You have big dreams that fire you up, and yet a fear of failure
    doubt you could ever do what they've done. You tell yourself you lack the smarts, skills,
    there's a massive gap between what you think you can do and what you're actually capable of.

    In Twenty-Five Hours A Day, Nick shares the lessons he learned while building his business
    by flipping the switch and going all in—then breaking that switch so he could never get
    suck, living like you've got an extra hour, and harnessing the power of "one more."
    success.</field>
    <field name="category">Biznes</field>
    <field name="comments">
      <doc>
        <field name="content">"this is so great book! 10/10"</field>
      </doc>
      <doc>
        <field name="content">"this is pretty poor position! 2/10. Boring!"</field>
      </doc>
    </field>
  </doc>
</add>
```

```
"response": {"numFound": 3, "start": 0, "numFoundExact": true, "docs": [
  {
    "content": "\"this is so great book! 10/10\"",
    "id": "1/comments#0",
    "_nest_parent_": "1",
    "_version_": 1758065021458317312},
  {
    "content": "\"this is pretty poor position! 2/10. Boring!\"",
    "id": "1/comments#1",
    "_nest_parent_": "1",
    "_version_": 1758065021458317312},
  {
    "id": "1",
    "title": "25 hours a day",
    "author": "Nick Bare",
    "price___s_ns": "PLN",
    "price": "43,PLN",
    "description_english": ["You have big dreams that fire you up
    "title_english": ["25 hours a day"],
    "_version_": 1758065021458317312}]
  }
}]
```

# Zagnieżdżone dokumenty

## Przeszukiwanie

Standardowe query: **q=author:nick**

Response: Dokument bez dzieci

```
"response": {"numFound": 1, "start": 0, "numFoundExact": true, "docs": [
  {
    "id": "1",
    "title": "25 hours a day",
    "author": "Nick Bare",
    "price__s_ns": "PLN",
    "price": "43,PLN",
    "description_english": ["You have big dreams that fire you up, and"],
    "title_english": ["25 hours a day"],
    "_version_": 1758065021458317312}]
}}
```



# Zagnieżdżone dokumenty

## Przeszukiwanie

### Query:

q=author:nick

fl: \*,[child]

### Response:

Dokument z dziećmi

### Obserwacja:

numFound = 1

```
"response": {"numFound": 1, "start": 0, "numFoundExact": true, "docs": [
  {
    "id": "1",
    "title": "25 hours a day",
    "author": "Nick Bare",
    "price_currency": "PLN",
    "price": "43,PLN",
    "description_english": ["You have big dreams that fire you up, and y"],
    "title_english": ["25 hours a day"],
    "_version_": 1758065021458317312,
    "comments": [
      {
        "content": "\"this is so great book! 10/10\"",
        "id": "1/comments#0",
        "_nest_parent_": "1",
        "_version_": 1758065021458317312},
      {
        "content": "\"this is pretty poor position! 2/10. Boring!\"",
        "id": "1/comments#1",
        "_nest_parent_": "1",
        "_version_": 1758065021458317312}}]}
  ]
}
```

# Zagnieżdżone dokumenty

## Child Query Parser

- Jeśli użyjemy Child Query Parser, możemy wyłapać wszystkie dzieci bez rodziców.
- Query pozostaje to samo, czyli wyszukujemy rodziców!
- Całe query musi zostać jednak wzbogacone o wzorzec:

`q={!child of=<blockMask>}<someParents>`

Np: `q={!child of="*:* *_nest_path_*"}author:nick`

# Zagnieżdżone dokumenty

## Child Query Parser

### Query:

```
q=q={!child of="*:* -_nest_path_*"}author:nick
```

### Response:

Dokumenty-dzieci

```
"response":{"numFound":2,"start":0,"numFoundExact":true,"docs":[
  {
    "content":"\"this is so great book! 10/10\"",
    "id":"1/comments#0",
    "_nest_parent_":"1",
    "_version_":1758065021458317312},
  {
    "content":"\"this is pretty poor position! 2/10. Boring!\"",
    "id":"1/comments#1",
    "_nest_parent_":"1",
    "_version_":1758065021458317312}]
}}
```

# Zagnieżdżone dokumenty

## Parent Query Parser

- Odwrotnością Child Query Parser jest Parent Query Parser
- Pozwala wyszukać rodziców dzieci, które znajdziemy.
- Całe query musi zostać ułożone pod wzorzec:

`q={!parent which=<blockMask>}<someChildren>`

Np: `q={!parent which="*:* *_nest_path_*"}content:great`

# Zagnieżdżone dokumenty

## Parent Query Parser

### Query:

```
q={!parent which="*:* -_nest_path_*"}content:great
```

### Response:

Dokument-rodzic zmatchowanego dziecka

```
"response":{"numFound":1,"start":0,"numFoundExact":true,"docs":[{"id":"1",
"title":"25 hours a day",
"suggest_field":["25 hours a day",
  "You have big dreams that fire you up, and yet a fear of failure is holding you back. You see the success others h
"author":"Nick Bare",
"price___s_ns":"PLN",
"price":"43,PLN",
"description_english":["You have big dreams that fire you up, and yet a fear of failure is holding you back. You see
"title_english":["25 hours a day"],
"_version_":1758356462643445760}]
}]}
```

**Gratulacje!**  
**Good job;-)**