

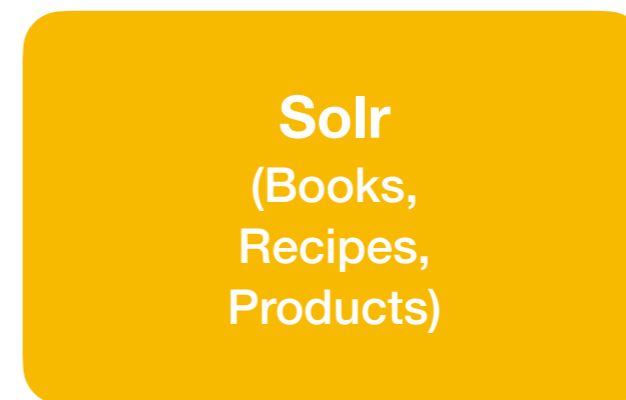
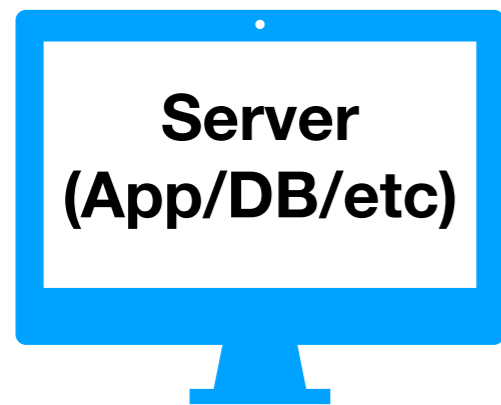
Solr



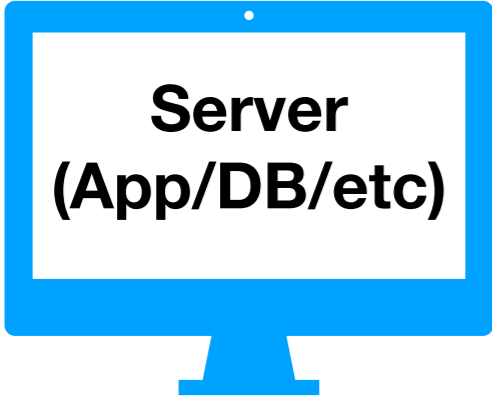
Search

Ogólna budowa

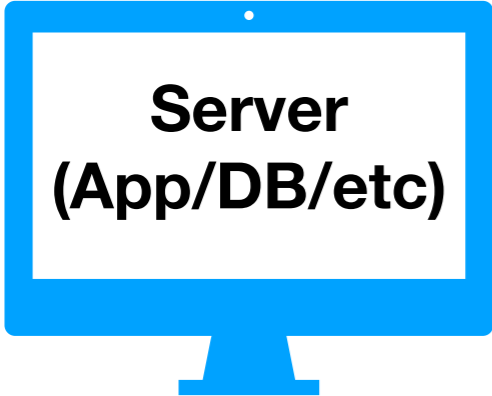
**W jakim punkcie
Jesteśmy?**

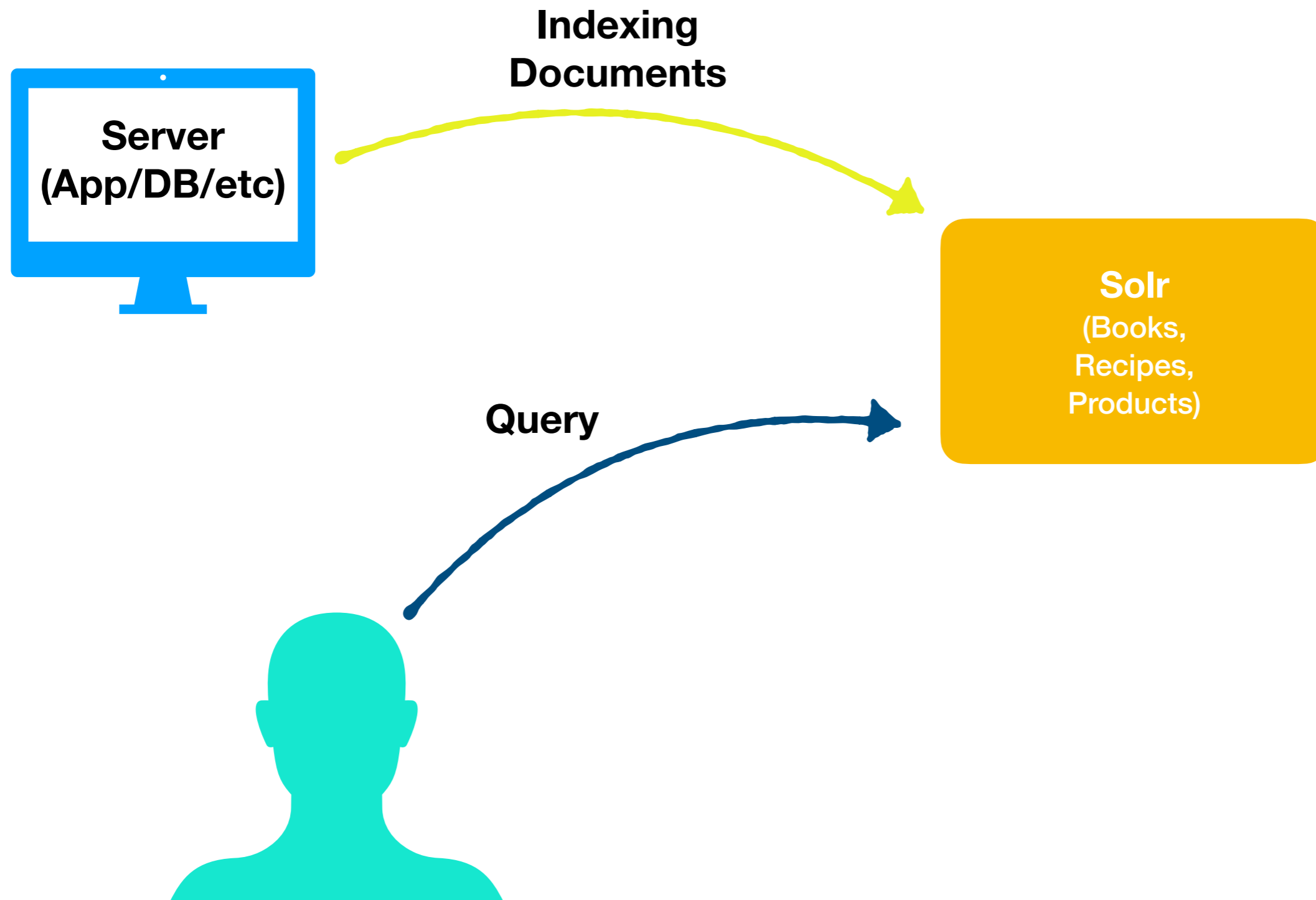


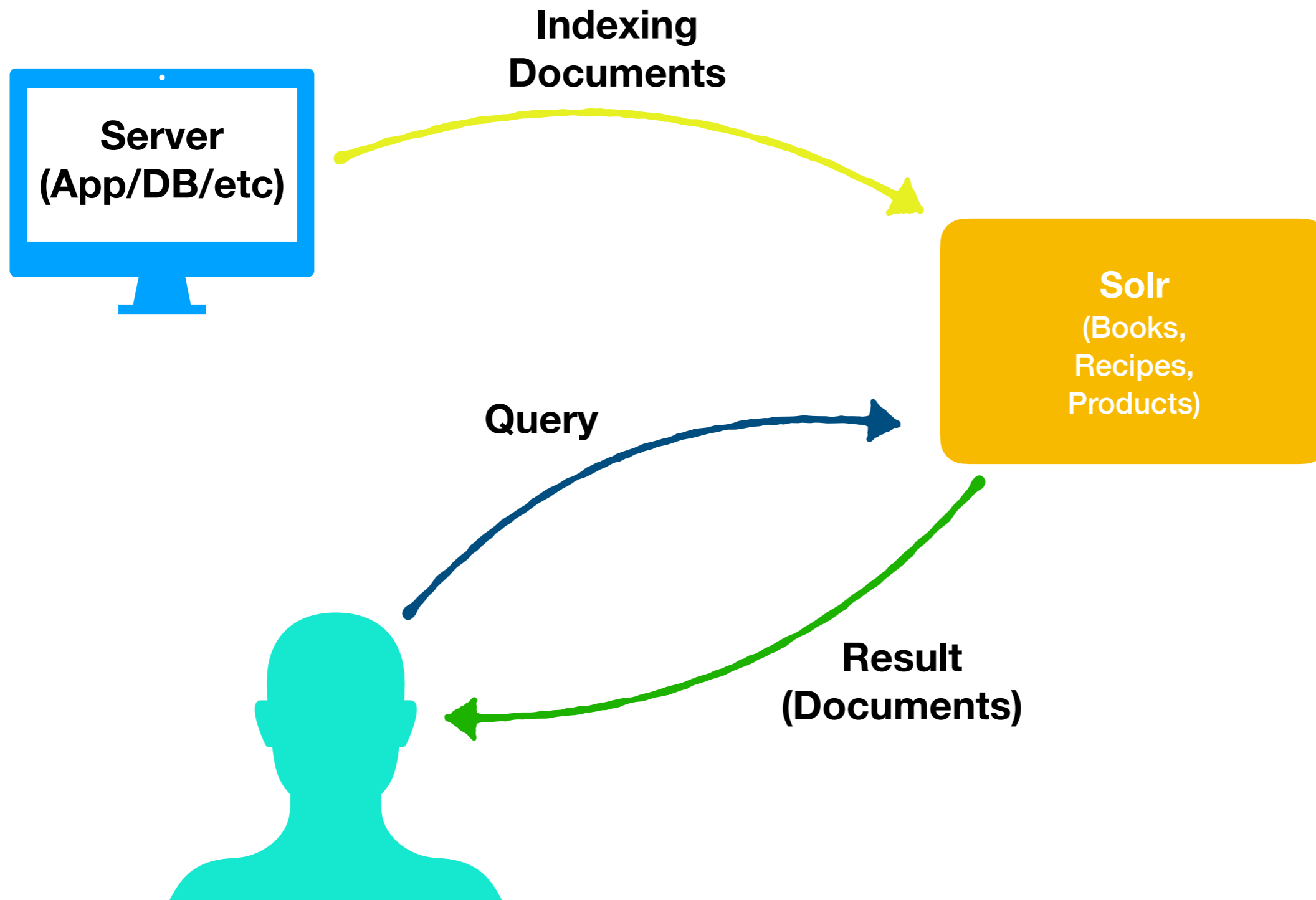
**Indexing
Documents**

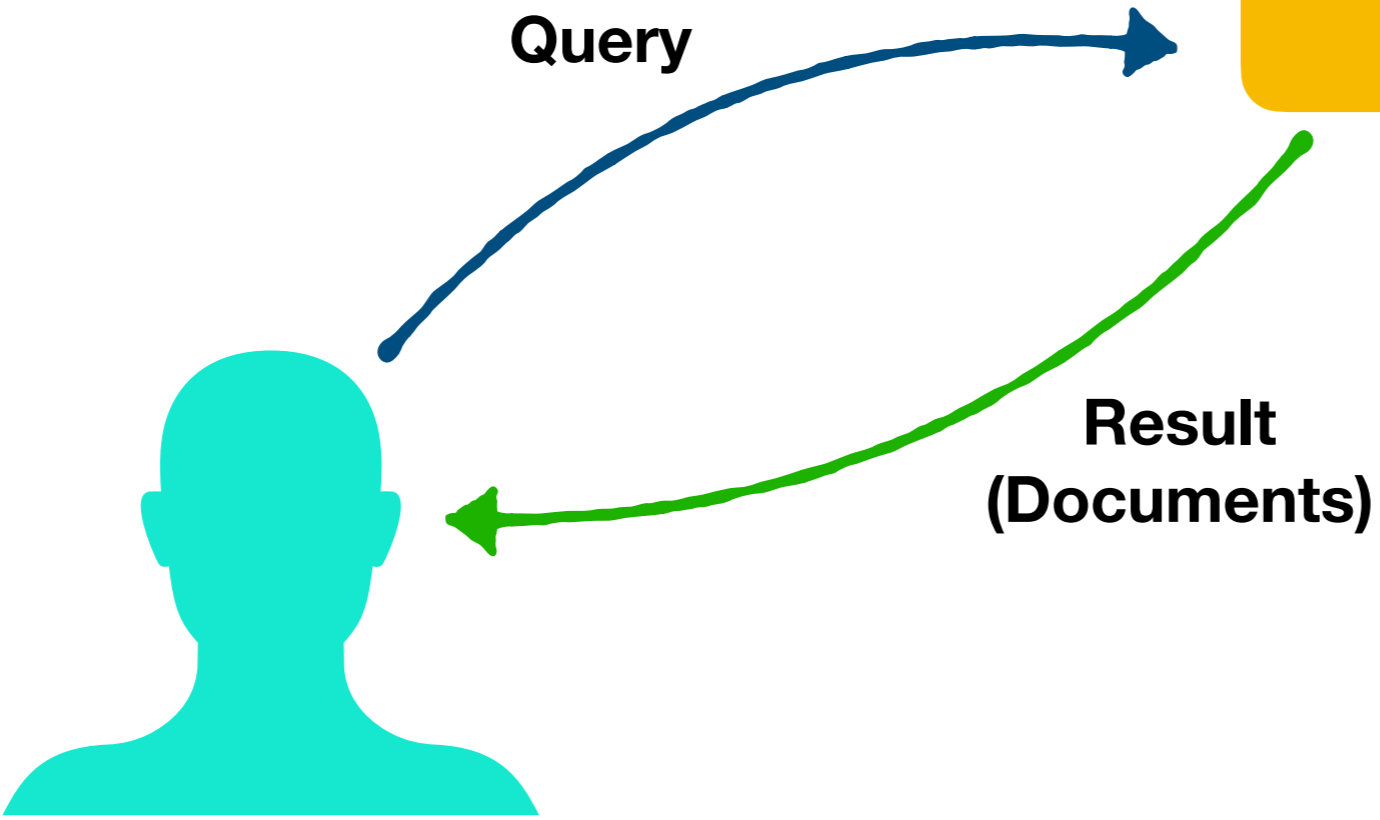
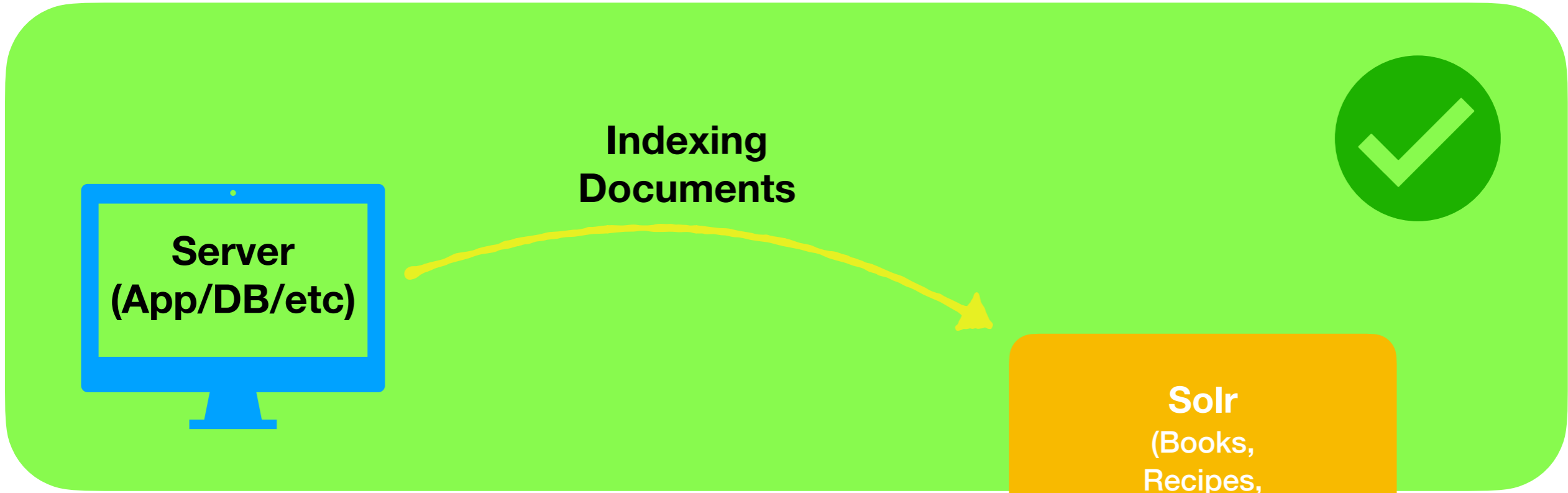


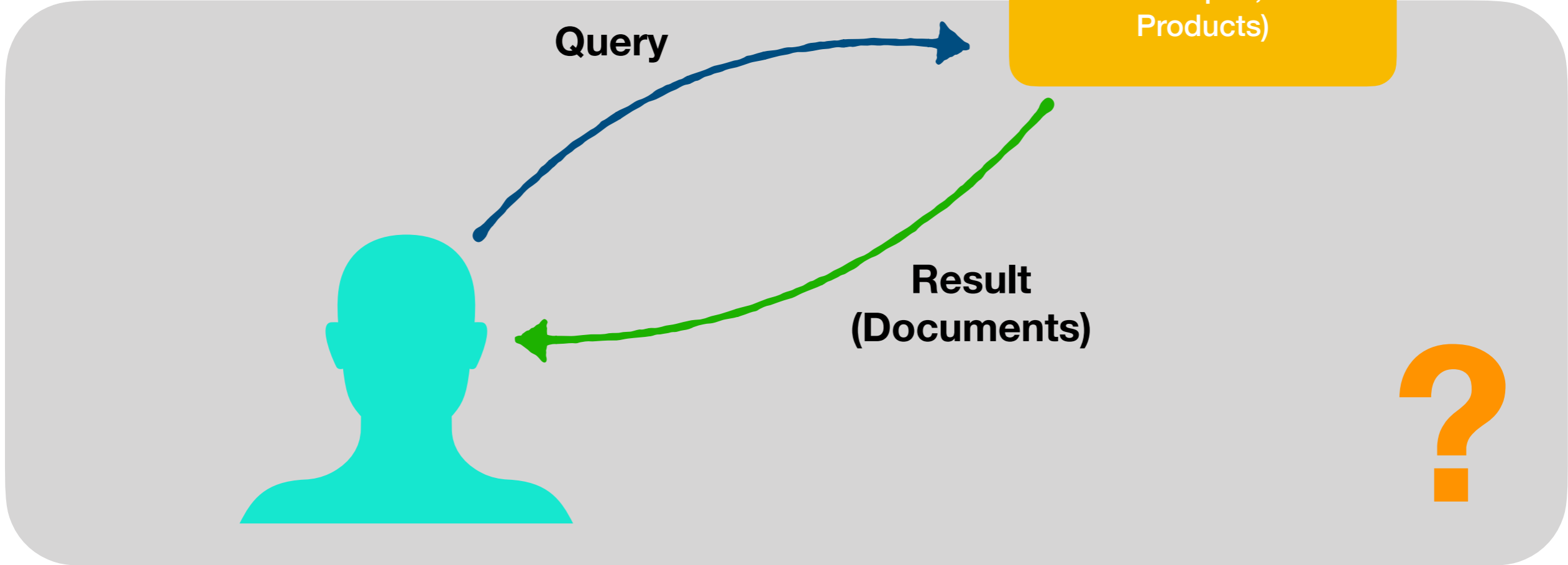
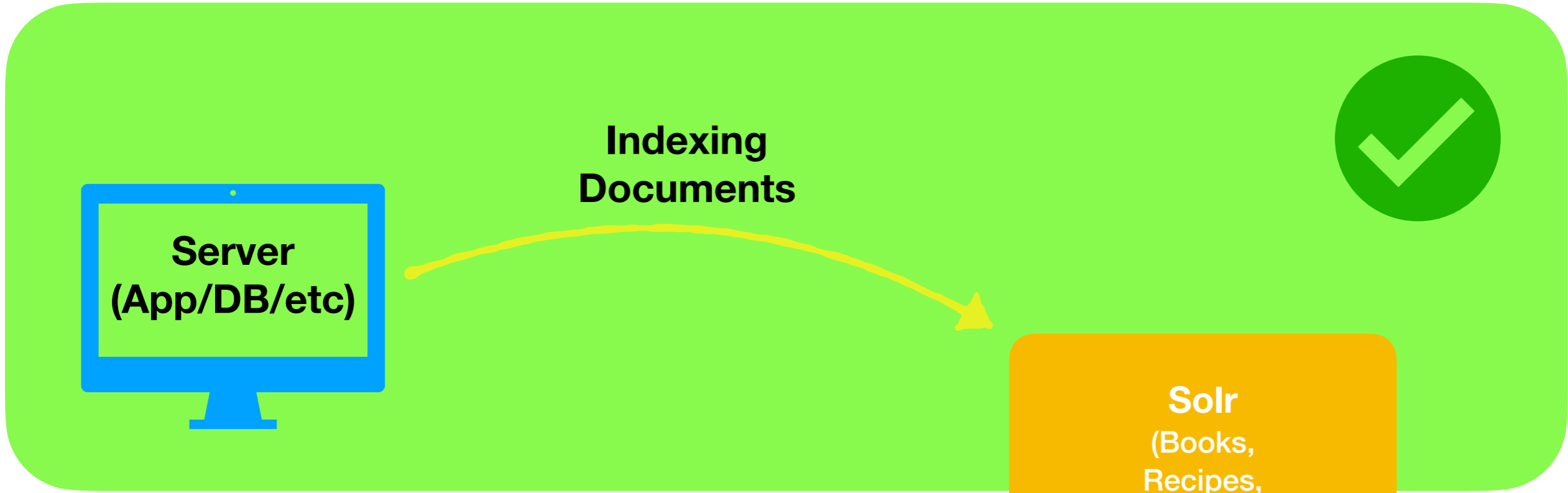
Indexing Documents











Przeszukiwanie

Ogólna budowa

1. Request Handler

2. Query Parser

3. Response Writer

Przeszukiwanie

Request Handler

- /select (class: 'solr.SearchHandler')
- /query (class: 'solr.SearchHandler')
- /suggest (class: 'solr.SuggestComponent')
- /spellcheck (class: 'solr.SpellCheckComponent')
- /update
- Całość ustawiamy w solrconfig.xml

Przeszukiwanie

Query Parser

- Standard Query Parser (“*Lucene Parser*”)
- DisMax Query Parser
- Common Query Parameters

Przeszukiwanie

Response Writer

- XML Response Writer
- JSON
- CSV
- PHP
- Python
- I inne...

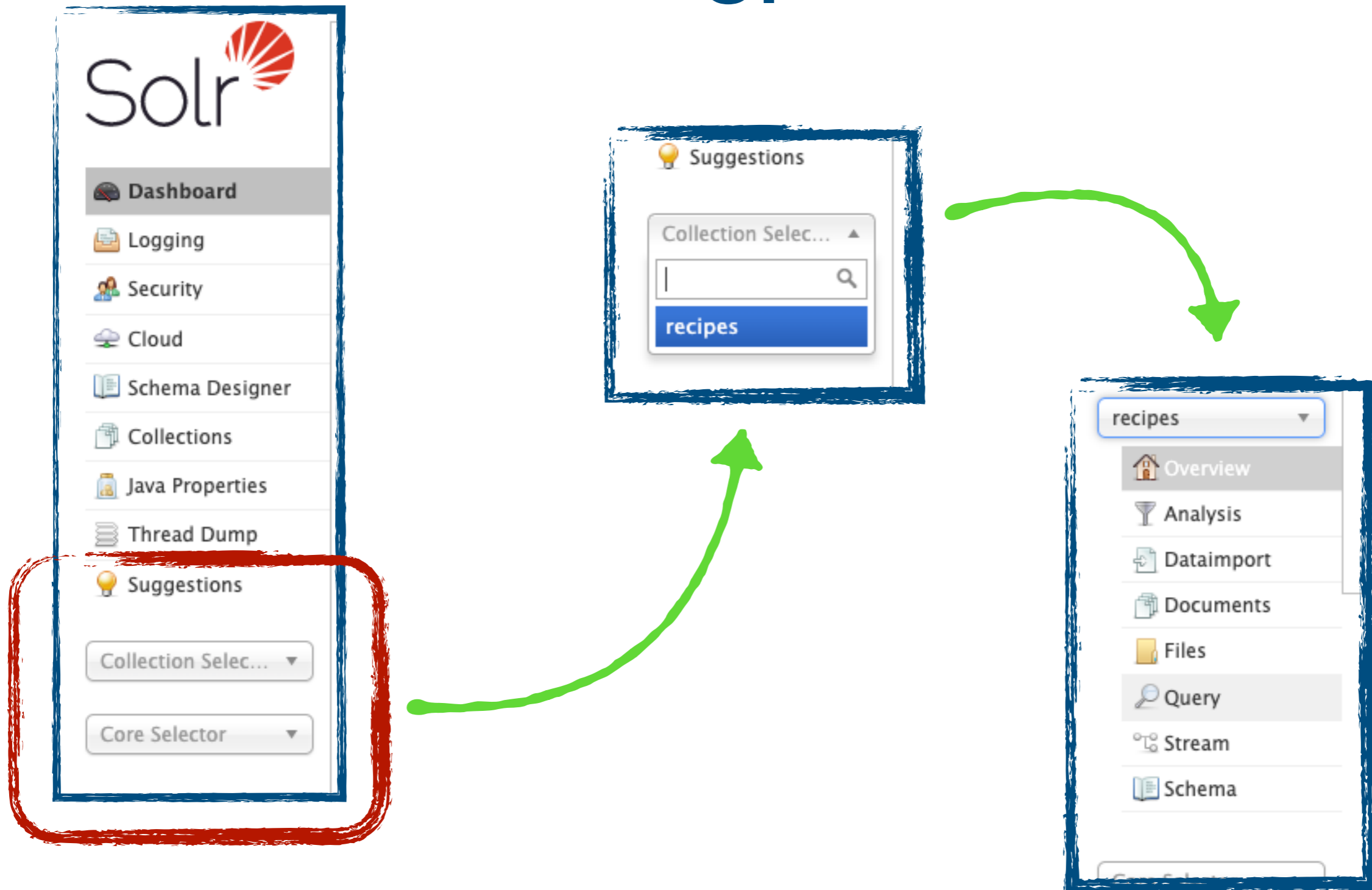
Przeszukiwanie

Sposoby

- Można przeszukiwać za pomocą:
 - HTTP (API restowe)
 - API (Java)
 - UI (HTTP obudowane aplikacją UI)

Przeszukiwanie

UI



Przeszukiwanie

UI

Request-Handler (qt)

/select

— common —

q

.

q.op

OR

fq

sort

start, rows

0 10

fl

df

wt

indent on

debugQuery

defType

lucene

hl

facet

spatial

spellcheck

Raw Query Parameters

key1=val1&key2=val2

Execute Query

http://localhost:8983/solr/recipes/select?indent=true&q.op=OR&q=%3A*

```
{
  "responseHeader":{
    "zkConnected":true,
    "status":0,
    "QTime":1,
    "params":{
      "q":"*.*",
      "indent":"true",
      "q.op":"OR",
      "_":"1690790461309"}},
  "response":{"numFound":10,"start":0,"numFoundExact":true,"docs":[
    {
      "id":"1",
      "title":"Mięсно-ryżowe klopsiki w warzywnym sosie",
      "ingredients":["300g mięsa mielonego wieprzowego",
        "100g ugotowanego ryżu",
        "1 jajko",
        "1/2 szklanki mąki- do obtoczenia klopsików",
        "1 papryka (użyłam pół żółtej i pół czerwonej)",
        "1 cebula",
        "1/2 cukinii",
        "2 ząbki czosnku",
        "1 i 1/2 szklanki bulionu",
        "3 łyżki koncentratu pomidorowego",
        "przyprawy: sól, pieprz, majeranek, bazylia, oregano, papryka",
        "2 łyżki posiekanej natki pietruszki, do posypania",
        "olej do smażenia",
        "opcjonalnie: 2 łyżki śmietanki"],
      "description":"W misce łączymy mięso mielone z ugotowanym i wyst",
      "time":"2h",
      "price__s_ns":"PLN",
      "price":"40,PLN",
      "_version_":1772652631391469568},
    {
      "id":"2",
      "title":"Udka z kurczaka ze śliwkami i boczkiem",
      "ingredients":["500-600 g uda z kurczaka bez kości (ok 5-6 kawał",
        "Suszone śliwki (można kupić w stokrotce)",
        "Przyprawy: papryka słodka mielona oraz papryka wędzona słodka",
        "Boczek (ugotowany lub surowy) w plastrach"],
      "description":"Udo kurczaka kroimy (na dwa), tłuczemy i przypraw",
      "time":"2h",
```

Przeszukiwanie Podstawy

Przeszukiwanie - podstawy

- Ogólne parametry przeszukiwania
- Jak wyglądają wyniki?
- Jak wygląda Standard Query Parser?
- Query posiada swoją konstrukcję!

Common Query Parameters

Common Query Parameters

- Podstawowe Query Parsery posiadają wspólne parametry query, nazwane *“common query parameters”*.
- **defType** - wybiera query parser. Domyślnie Standard Query Parser (defType=lucene)
- **sort** - określa sortowanie rosnące (asc) lub malejące (desc)
- **start** - określamy offset (domyślnie 0)
- **rows** - używamy do paginacji (domyślnie 10)
- **fq** (filter query) - używamy do ograniczenia zbioru z którego będziemy wybierać.
- **fl** (field list) - lista pól, które są brane pod uwagę

Common Query Parameters

- **debug** - dodaje dodatkowych informacji w odpowiedziach (wartości: query,timing,results,all/true)
- **timeAllowed** - maksymalny czas w milisekundach, jaki możemy przeznaczyć na przeszukanie.
- **omitHeader** - jeśli true, pomija header w odpowiedziach (domyślnie false)
- **wt** - Response Writer (domyślnie json)
- **cache** (true/false) - domyślnie cache'uje wyniki wszystkich query i fq.

Common Query Parameters

- **echoParams** - określa jakie informacje na temat parametrów z requestu mają być dołączone do headera (w odpowiedzi/response)
- **explicit** (domyślnie) - tylko parametry z aktualnego query
- **All** - wszystkie parametry, które biorą udział w query (także to co jest określone w solrconfig.xml)
- **None** - wyrzuca parametry z headera odpowiedzi.

Common Query Parameters

echoParams

explicit

```
{
  "responseHeader": {
    "status": 0,
    "QTime": 0,
    "params": {
      "q": "solr",
      "indent": "true",
      "wt": "json",
      "_": "1458227751857"
    }
  },
  "response": {
    "numFound": 0,
    "start": 0,
    "docs": []
  }
}
```

all

```
{
  "responseHeader": {
    "status": 0,
    "QTime": 0,
    "params": {
      "q": "solr",
      "df": "text",
      "preferLocalShards": "false",
      "indent": "true",
      "echoParams": "all",
      "rows": "10",
      "wt": "json",
      "_": "1458228887287"
    }
  },
}
```


Standard Query Parser

Standard Query Parser

- q - podstawowy parametr, w którym przekazujemy zapytanie (query).
- q.op (AND/OR) - domyślny operator dla wyrażeń query (query expressions).
- df - domyślne pole, po którym przeszukujemy.
- sow (true/false) - (split on whitespace) Jeśli true, analiza tekstu jest osobno dla każdego wyrazu oddzielonego znakiem specjalnym
- Domyślne wartości parametrów są opisane w solrconfig.xml. Jeśli my dodamy je do query, zostaną nadpisane.

Standard Query Parser

Response

- `<responseHeader>`
- `<result>`
- `<lst>` jeśli włączony jest *debug* (z informacjami dodatkowymi).

Standard Query Parser

Response

```
<response>
<responseHeader><status>0</status><QTime>1</QTime></responseHeader>
<result numFound="1" start="0">
  <doc>
    <arr name="cat"><str>electronics</str><str>hard drive</str></arr>
    <arr name="features"><str>7200RPM, 8MB cache, IDE Ultra ATA-133</str>
      <str>NoiseGuard, SilentSeek technology, Fluid Dynamic Bearing (FDB) motor</str></arr>
    <str name="id">SP2514N</str>
    <bool name="inStock">>true</bool>
    <str name="manu">Samsung Electronics Co. Ltd.</str>
    <str name="name">Samsung SpinPoint P120 SP2514N - hard drive - 250 GB - ATA-133</str>
    <int name="popularity">6</int>
    <float name="price">92.0</float>
    <str name="sku">SP2514N</str>
  </doc>
</result>
</response>
```

Przykłady Query

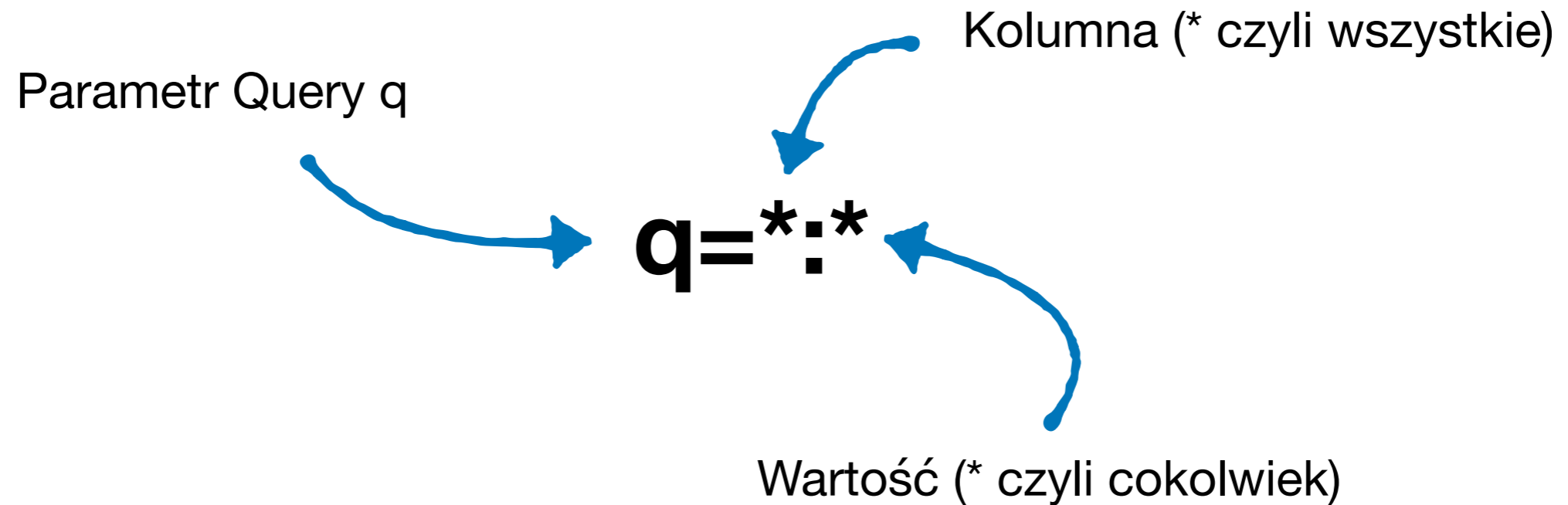
Przykłady query

Scenariusz nr 1:

Chcemy wyciągnąć wszystkie dokumenty

Przykłady query

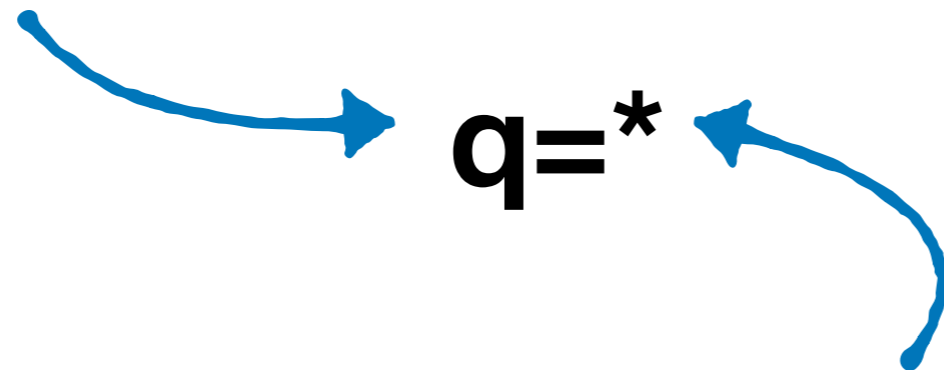
#1



Przykłady query

#1

Parametr Query q



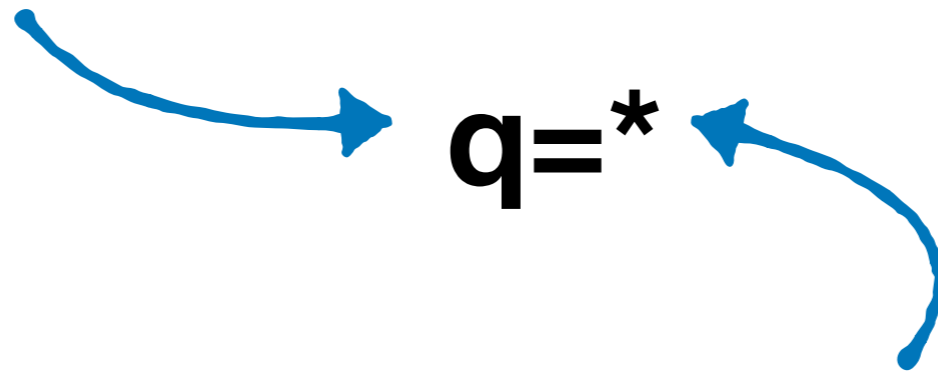
q=*

Wartość (* czyli cokolwiek)

Przykłady query

#1

Parametr Query q



Wartość (* czyli cokolwiek)

Request

http://solrhost:8983/solr/books_spellcheck/query?indent=true&q.op=OR&q=*

Request-Handler (qt)

/query

— common —

q

*

q.op

OR

fq

sort

start, rows

0

10

fl

df

wt

indent on

debugQuery

defType

lucene

hl

facet

spatial

spellcheck

Raw Query Parameters

key1=val1&key2=val2

Execute Query

http://192.168.100.23:8983/solr/books_spellcheck/query?indent=true&q.op=OR&q=*

```
"responseHeader": {
  "zkConnected": true,
  "status": 0,
  "QTime": 0,
  "params": {
    "q": "*",
    "indent": "true",
    "q.op": "OR",
    "_forwardedCount": "1",
    "_": "1676649826414"}},
"response": { "numFound": 6, "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
    "author": "Nick Bare",
    "price___s_ns": "PLN",
    "price": "43,PLN",
    "description_english": ["You have big dreams that fire you up, and yet
    "title_english": ["25 hours a day"],
    "_version_": 1758100787610779648},
  {
    "id": "2",
    "title": "Hooked: How to Build Habit-Forming Products",
    "suggest_field": ["Hooked: How to Build Habit-Forming Products",
      "Nir Eyal reveals how successful companies create products people c
    "author": "Nir Eyal",
    "price___s_ns": "PLN",
    "price": "59,PLN",
    "description_english": ["Nir Eyal reveals how successful companies cre
    "title_english": ["Hooked: How to Build Habit-Forming Products"],
    "_version_": 1758100787622313984},
  {
    "id": "3",
    "title": "Allgäu 2: Ostallgäu und vorderes Lechtal 55 Touren mit GPS-T
    "suggest_field": ["Allgäu 2: Ostallgäu und vorderes Lechtal 55 Touren
      "Die Schlösser Neuschwanstein und Hohenschwangau, Bergseen und Weih
    "author": "Mark Zahel",
    "price___s_ns": "PLN",
```

Przykłady query

Scenariusz nr 2:

Wyszukujemy dokumenty, które mają słowo “hours” LUB
“build” tytule

Przykłady query

#2

Parametr Query q

Wartość ("hours" oraz "build")

q=hours build

Parametr Query df
(default field)

df=title

Wartość ("title" - przeszukujemy po tytule, chyba że w q podamy inaczej)

Parametr Query q.op
(operator logiczny)

q.op=OR

Wartość ("OR" - LUB)

Przykłady query

#2

Wartość ("hours" oraz "build")

Parametr Query q

q=hours build

Parametr Query df
(default field)

df=title

Wartość ("title" - przeszukujemy po tytule, chyba że w q podamy inaczej)

Parametr Query q.op
(operator logiczny)

q.op=OR

Wartość ("OR" - LUB)

Request

<http://solrhost:8983/solr/books/select?df=title&indent=true&q.op=OR&q=hours%20build>

Request-Handler (qt)

/select

— common —

q

hours build

q.op

OR

fq

sort

start, rows

0

10

fl

df

title

wt

indent on

debugQuery

defType

lucene

hl

facet

spatial

spellcheck

Raw Query Parameters

http://192.168.100.23:8983/solr/books/select?df=title&indent=true&q.op=OR&q=hours%20build

```
{
  "responseHeader":{
    "zkConnected":true,
    "status":0,
    "QTime":0,
    "params":{
      "q":"hours build",
      "df":"title",
      "indent":"true",
      "q.op":"OR",
      "_forwardedCount":"1",
      "_":"1676649826414"}},
  "response":{"numFound":2,"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",
      "category":"Biznes",
      "language":"en",
      "languages":["en"],
      "_version_":1757941861141446656},
    {
      "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",
      "category":"Biznes",
      "language":"en",
      "languages":["en"],
      "_version_":1757941861194924032}]
  }}
}
```

Przykłady query

Scenariusz nr 3:

Wyszukujemy dokumenty, które mają słowo “hours” i “build”
tytule

Przykłady query

#3

Parametr Query q

Wartość ("hours" oraz "build")

q=hours build

Parametr Query df
(default field)

df=title

Wartość ("title" - przeszukujemy
po tytule, chyba że
w q podamy inaczej)

Parametr Query q.op
(operator logiczny)

q.op=AND

Wartość ("AND" - i)

Przykłady query

#3

Wartość ("hours" oraz "build")

Parametr Query q

q=hours build

Parametr Query df
(default field)

df=title

Wartość ("title" - przeszukujemy po tytule, chyba że w q podamy inaczej)

Parametr Query q.op
(operator logiczny)

q.op=AND

Wartość ("AND" - i)

Request

<http://solrhost:8983/solr/books/select?df=title&indent=true&q.op=AND&q=hours%20build>

The screenshot displays the Solr Request-Handler (qt) interface. The URL bar shows the request: `http://192.168.100.23:8983/solr/books/select?df=title&indent=true&q.op=AND&q=hours%20build`. The form on the left contains the following fields:

- `Request-Handler (qt)`: `/select`
- `common`: (empty)
- `q`: `hours build`
- `q.op`: `AND`
- `fq`: (empty)
- `sort`: (empty)
- `start, rows`: `0` and `10`
- `fl`: (empty)
- `df`: `title`

The JSON response on the right is:

```
{
  "responseHeader": {
    "zkConnected": true,
    "status": 0,
    "QTime": 1,
    "params": {
      "q": "hours build",
      "df": "title",
      "indent": "true",
      "q.op": "AND",
      "_forwardedCount": "1",
      "_": "1676649826414"
    }
  },
  "response": { "numFound": 0, "start": 0, "numFoundExact": true, "docs": [] }
}
```

Request

<http://solrhost:8983/solr/books/select?df=title&indent=true&q.op=AND&q=hours%20build>

Przeszukiwanie Zaawansowane

Przeszukiwanie zaawansowane

- Jak znaleźć zakresy?
- Jak znaleźć słowa, które są “niepełne”?
- Jak wzmocnić niektóre frazy a osłabić inne?
- Jak uwzględniać literówki?
- Jak przeszukiwać efektywnie całą frazę po tekście?

Przedziały

Przedziały

podstawy

- Możemy określać w query przedziały, jeśli mamy wartości liczbowe i pokrewne (np. Daty, waluty itd)
- Używamy parametru q
- Przedziały opisujemy za pomocą nawiasów oraz słówka TO:
 - [x TO y] - jeśli wartości mają się zawierać (“włącznie”)
 - { x TO y } - jeśli wartości mają być wyłączone (“wyłącznie”)
- Np:
 - [2 TO 45] - od 2 do 45 (2 i 45 też się łapie)
 - {0 TO 100} - wszystko co jest pomiędzy 0 a 100, ale bez 0 (100 jest OK!)
- Możemy użyć wildcard (*) jako określenie obu skrajności, np:
 - [* TO 200] - mniejsze lub równe 200. (≤ 200)
 - {5 TO *} - większe niż 5 (> 5)

Przedziały

Daty

- Możemy także określić daty używając tych samych przedziałów
- Musimy zachować odpowiedni format danych - Solr domyślnie używa formatu ISO_INSTANT
 - YYYY-MM0DDThh:mm:ssZ
 - Np 2020-12-25T14:23:19Z
 - Zawsze podajemy w UTC
- [2000-01-01 TO 2019-12-31] obejmie pierwsze 20 lat naszego wieku
- NOW - obecny czas
- Matematyka: możemy w trakcie wyszukiwania robić podstawowe operacje na czasie, np:
 - NOW+3MONTHS
 - NOW-1DAY
 - NOW/HOUR - zaokrąglamy. Tutaj: od początku obecnej godziny.

Przykłady query

Scenariusz nr 4:

Wyszukujemy książki, które są droższe niż 20 zł i są po polsku LUB kosztują co najmniej 15\$ i nie są napisane po polsku

Przykłady query

#4

Parametr Query q

Pole "language" tylko o wartości "pl"

Przedział >20 zł (PLN to waluta domyślna)

Obie części połączone operatorem "LUB"

**q=(price: {20 TO *} AND language:pl) OR
(price: [15,USD TO *] AND !language:pl)**

Przedział >= 15\$
(Nawias kwadratowy)

Operator "I", który
łączy warunki w każdym
Z "podzapytań"

Negacja wyrażenia
"language:pl"

Request-Handler (qt)

/select

common

q
(price: {20 TO *} AND language:pl) OR (price: [15,USD TO *] AND !language:pl)

q.op
OR

fq

sort

start, rows
0 10

fl

df

wt

indent on

debugQuery

defType
lucene

hl

facet

spatial

spellcheck

Raw Query Parameters

key1=val1&key2=val2

Execute Query

http://192.168.100.23:8983/solr/books_spellcheck/select?indent=true&q.op=OR&q=(price%3A%20%7B20%20TO%20*%5D%20AND%20language%3Apl)%20OR%20(price%3A%20%5B15%20USD%20TO%20*%5D%20AND%

```
{
  "responseHeader": {
    "zkConnected": true,
    "status": 0,
    "qTime": 1,
    "params": {
      "q": "(price: {20 TO *} AND language:pl) OR (price: [15,USD TO *] AND !language:pl)",
      "indent": "true",
      "q.op": "OR",
      "_forwardedCount": "1",
      "_": "1676649826414"
    }
  },
  "response": {
    "numFound": 4,
    "start": 0,
    "numFoundExact": true,
    "docs": [
      {
        "id": "5",
        "title": "Iverson Zycie to nie gra",
        "suggest_field": ["Iverson Zycie to nie gra",
          "Niesamowity sukces i niewiarygodny upadek z\u0142otego dziecka koszyk\u00f3wki\n\nSyn alkoholicki i recydywisty, kt\u00f3ry \u017cy\u0107cia uczy\u0142 si\u0119 na niebezpiecznych ulicach Wirginii. Nie mi"],
        "author": "Kent Babb",
        "price__s_ns": "PLN",
        "price": "98,PLN",
        "description_polish": ["Niesamowity sukces i niewiarygodny upadek z\u0142otego dziecka koszyk\u00f3wki\n\nSyn alkoholicki i recydywisty, kt\u00f3ry \u017cy\u0107cia uczy\u0142 si\u0119 na niebezpiecznych ulic"],
        "title_polish": ["Iverson Zycie to nie gra"],
        "_version_": 1758100787642236928
      },
      {
        "id": "6",
        "title": "Tech Krytyka rozwoju srodowiska technologicznego",
        "suggest_field": ["Tech Krytyka rozwoju srodowiska technologicznego",
          "Wsp\u00f3\u0142czesna ekspansja technik doprowadzi\u0142a do wytworzenia nowego technologicznego srodowiska, kt\u00f3re jest coraz bardziej dynamiczne, ekspansywne i staje si\u0119 w coraz wi\u0119ks"],
        "author": "Jan Bia\u0142ek",
        "price__s_ns": "PLN",
        "price": "33,PLN",
        "description_polish": ["Wsp\u00f3\u0142czesna ekspansja technik doprowadzi\u0142a do wytworzenia nowego technologicznego srodowiska, kt\u00f3re jest coraz bardziej dynamiczne, ekspansywne i sta"],
        "title_polish": ["Tech Krytyka rozwoju srodowiska technologicznego"],
        "_version_": 1758100787648528384
      },
      {
        "id": "3",
        "title": "Allg\u00e4u 2: Ostallg\u00e4u und vorderes Lechtal 55 Touren mit GPS-Tracks",
        "suggest_field": ["Allg\u00e4u 2: Ostallg\u00e4u und vorderes Lechtal 55 Touren mit GPS-Tracks",
          "Die Schl\u00f6sser Neuschwanstein und Hohenschwangau, Bergseen und Weiher, die Wieskirche und die Altstadt von F\u00fcssen, sanfte Almen und schroffe Gipfel \u2013 das Ostallg\u00e4u und da"],
        "author": "Mark Zahel",
        "price__s_ns": "PLN",
        "price": "69,PLN",
        "description_german": ["Die Schl\u00f6sser Neuschwanstein und Hohenschwangau, Bergseen und Weiher, die Wieskirche und die Altstadt von F\u00fcssen, sanfte Almen und schroffe Gipfel \u2013"],
        "title_german": ["Allg\u00e4u 2: Ostallg\u00e4u und vorderes Lechtal 55 Touren mit GPS-Tracks"],
        "_version_": 1758100787630702592
      },
      {
        "id": "4",
        "title": "Karate Shorin-ryu (deluxe edition): L'eredit\u00e0 delle guardie del re di Okinawa: 1",
        "suggest_field": ["Karate Shorin-ryu (deluxe edition): L'eredit\u00e0 delle guardie del re di Okinawa: 1",
          "Lo Shorin-ryu \u00e8 lo stile di Karate pi\u00f9 antico esistente. Da questo stile marziale, adatto a chiunque, e basato su movimenti naturali, \u00e8 derivato il moderno Karate giappo"],
        "author": "Emanuel Giordano",
        "price__s_ns": "PLN",
        "price": "212,PLN",
        "description_italian": ["Lo Shorin-ryu \u00e8 lo stile di Karate pi\u00f9 antico esistente. Da questo stile marziale, adatto a chiunque, e basato su movimenti naturali, \u00e8 derivato il"]
      }
    ]
  }
}
```

Wildcard Search

Wildcard Search

- Solr (Lucene) pozwala wyszukiwać nie tylko po dokładnych frazach, ale także “domyślając się” o co może chodzić.
- Jednym z mechanizmów są wildcardy - możemy wstawić znak specjalny, aby Solr znalazł za nas “brakującą część”.
 - ? - oznacza, że chcemy, aby dopasował jeden znak (literę). Np “t?st” znajdzie odpowiedzi “test” i “tost”
 - * - oznacza, że chcemy, aby dopasował dowolną liczbą znaków. Np “t*st” znajdzie odpowiedzi “test”, “tost”, “toast” i “twist”.



Fuzzy Search

- Solr (Lucene) daje możliwość przeszukiwania z wykorzystaniem dystansu między wyszukiwaną frazą a danymi zaindeksowanymi.
- Dystans - odległość, która określa jak daleko od siebie są słowa. Odległość mierzona jest (najczęściej) w krokach. Każdy krok to jedna operacja:
 - Wstawienia litery
 - Wyjęcia litery
 - Przesunięcia liter
 - Np: “mama” leży w odległości 1 od słowa “mam”, ponieważ trzeba zabrać jedną literkę (“a”)

Fuzzy Search

- Podstawowym algorytmem używanym w Solr, Lucene, Elasticsearch i wszędzie indziej, jest algorytm dystansu Levensthein'a (ciekawskim polecam zaimplementować w dowolnym języku programowania)
- Można stosować także inne algorytmy odległości, jednak “utarło się”, że “Levensthein jest cool”.
- W Standard Query Parser wystarczy dodać na końcu wyrazu znak tyldy (~)
- Domyślnie mamy odległość = 2.
- Np: q=mama~
- Fuzzy search używamy tylko do wyrazów (terms).

Proximity Search

- To także wyszukiwanie rozmyte, tyle że w stosunku do fraz.
- Wyszukujemy tutaj jak daleko od siebie są wyrazy w tekście.
- Miarą odległości jest tym razem przestawienie wyrazów, nie znaków.
- Aby to osiągnąć podajemy “wymarzoną” frazę, znak tyldy oraz odległość
- Np: q=“how products”~4 wyszuka nam w odległości 4 kroków.

Przykłady query

Scenariusz nr 5 - proximity:

Wyszukujemy książki, które w tytule mają słowa “how products” w odległości maksymalnie 3 kroków (wyrazów).

Przykłady query

#5

q=title: "how products"~4

q=title: "how products"~3

```
"response": {"numFound": 1, "start": 0, "numFoundExact": true, "docs": [
  {
    "id": "2",
    "title": "Hooked: How to Build Habit-Forming Products",
    "suggest_field": ["Hooked: How to Build Habit-Forming Products",
      "Nir Eyal reveals how successful companies create products"],
    "author": "Nir Eyal",
    "price__s_ns": "PLN",
    "price": "59,PLN",
    "description_english": ["Nir Eyal reveals how successful companies create products"],
    "title_english": ["Hooked: How to Build Habit-Forming Products"],
    "_version_": 1758100787622313984}]
}}
```

```
"_": "1676649826414"}},
"response": {"numFound": 0, "start": 0, "numFoundExact": true, "docs": []
}}
```

Request

http://solrhost:8983/solr/books_spellcheck/select?indent=true&q.op=OR&q=title%3A%20%22how%20products%22~3

Przykłady query

Scenariusz nr 6:

Wyszukujemy książki, gdzie opis spełnia następujące wymagania:

Zawiera wyrazy podobne do “hour”,

Zawiera wyrazy pochodne od “success”

Zawiera wyraz “big”, ale są one dużo mniej istotne od reszty.

Przykłady query

#6

Parametr Query q

Fuzzy search (domyślnie 2 kroki)

q=description:(hour~ success* big^{0,2})

Success jako baza,
Wildcard pozwala
"Domyślić się"
Reszty wyrazu

Boostingiem
Oslabiamy
Wyraz "big"

Przykłady query

Parametr Query q

#6

Fuzzy search (domyślnie 2 kroki)

q=description:(hour~ success* big^0,2)

Success jako baza,
Wildcard pozwala
"Domyślić się"
Reszty wyrazu

Boostingiem
Oslabiamy
Wyraz "big"

Request

[http://solrhost:8983/solr/books_spellcheck/select?
indent=true&q.op=OR&q=description%3A\(hour~%20success*%20big%5E0%2C2\)](http://solrhost:8983/solr/books_spellcheck/select?indent=true&q.op=OR&q=description%3A(hour~%20success*%20big%5E0%2C2))

Przykłady query

#6

q=description:(hour~ success* big^0,2)

q=description:(hour~ success* big)

```

    "response": {"numFound": 3, "start": 0, "numFoundExact": true, "docs": [
      {
        "id": "2",
        "title": "Hooked: How to Build Habit-Forming Products",
        "suggest_field": ["Hooked: How to Build Habit-Forming Products",
          "Nir Eyal reveals how successful companies create products"],
        "author": "Nir Eyal",
        "price___s_ns": "PLN",
        "price": "59,PLN",
        "description_english": ["Nir Eyal reveals how successful companies create products"],
        "title_english": ["Hooked: How to Build Habit-Forming Products"],
        "_version_": 1758100787622313984},
      {
        "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"],
        "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"],
        "title_english": ["25 hours a day"],
        "_version_": 1758100787610779648},
      {

```

```

    "response": {"numFound": 3, "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"],
        "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"],
        "title_english": ["25 hours a day"],
        "_version_": 1758100787610779648},
      {
        "id": "2",
        "title": "Hooked: How to Build Habit-Forming Products",
        "suggest_field": ["Hooked: How to Build Habit-Forming Products",
          "Nir Eyal reveals how successful companies create products people"],
        "author": "Nir Eyal",
        "price___s_ns": "PLN",
        "price": "59,PLN",
        "description_english": ["Nir Eyal reveals how successful companies create products people"],
        "title_english": ["Hooked: How to Build Habit-Forming Products"],
        "_version_": 1758100787622313984},
      {

```

Faceting

Faceting

- Jak sprawdzić, ile osób żyje w danym województwie?
- Jak pogrupować liczbę książek w zakresy cenowe?

Faceting

- Faceting pozwala pogrupować wyszukiwanie po kategoriach bazując na wyszukiwanym query/frazie/wyrazie.
- Dzięki facetingowi możemy zbudować listę kategorii wraz z liczbą dokumentów przypisanych do nich. Dzięki temu można stworzyć m.in. prostszą nawigację.

Faceting

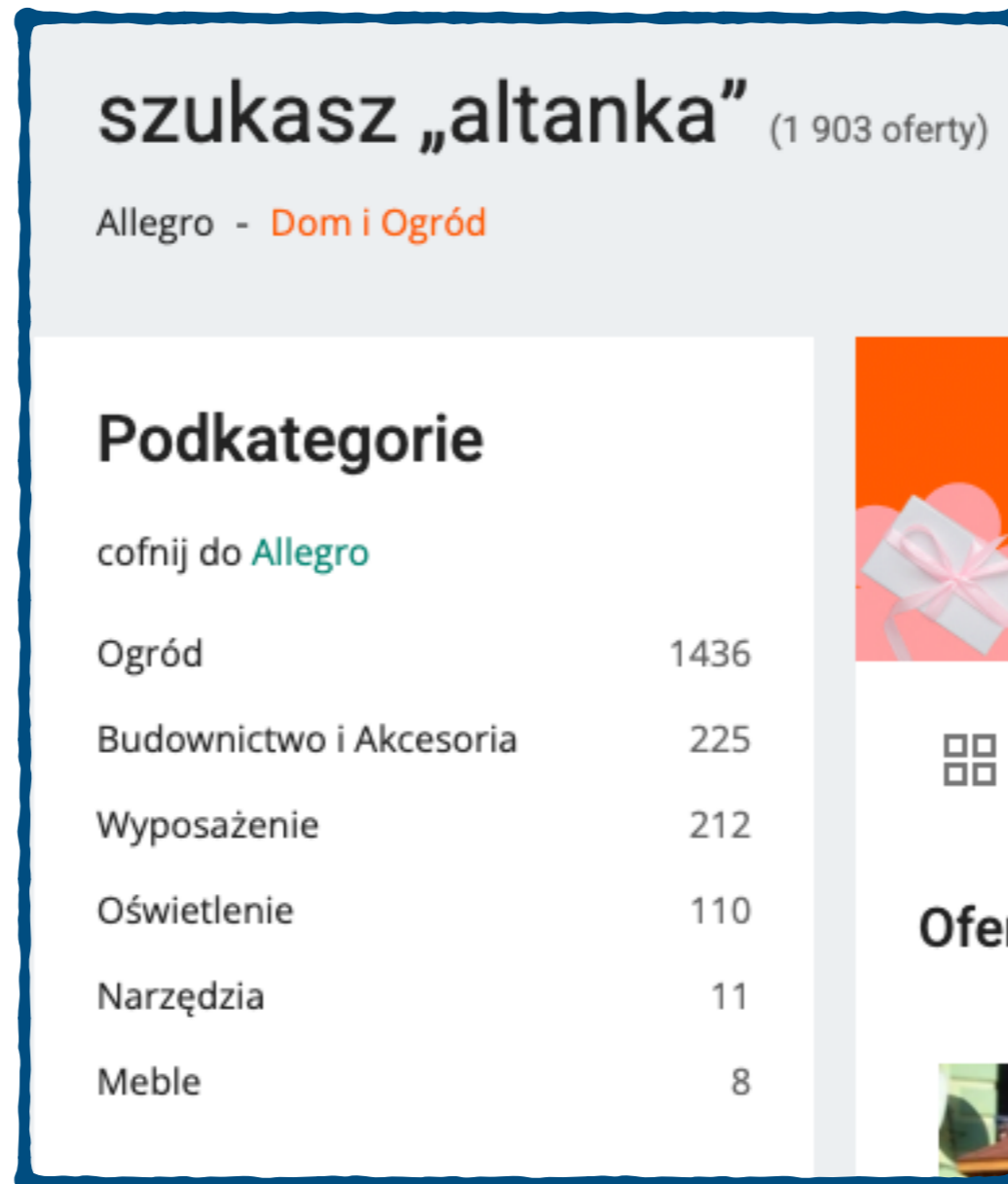
szukasz „altanka” (1 903 oferty)

Allegro - [Dom i Ogród](#)

Podkategorie

[cofnij do Allegro](#)

Ogród	1436
Budownictwo i Akcesoria	225
Wyposażenie	212
Oświetlenie	110
Narzędzia	11
Meble	8



Faceting

Podstawowy podział

1. Field-Value - klasyczne grupowanie po kluczu. Sprawdza się dobrze np. Do sprawdzenia kategorii w danym wyszukiwaniu.
2. Range faceting - grupowanie po zakresach. Sprawdza się dobrze m.in. przy cenach, ocenach, datach.

Faceting

Podstawowe parametry

- `facet (true/false)` - włącza lub wyłącza faceting.
- `facet.query` - query, dzięki któremu określamy jak konkretny faceting ma być zrealizowany.
- `facet.field` - określamy jakie pola mają zostać zgrupowane

Faceting

Podstawowe parametry

- facet.range - pole po jakim ma być przeprowadzony faceting zakresowy (*"range faceting"*)
 - facet.range.start - początek zakresu, np
f.price.facet.range.start=0
 - facet.range.end - analogicznie do start.
 - facet.range.gap - rozpiętość zakresu.
 - facet.range.other - określamy czy facet ma zawierać dodatkowe elementy poza naszymi obszarami (before, after, none, all)

Przykłady query

Scenariusz nr 7:

Sprawdźmy, do jakich kategorii oraz w jakich językach są napisane książki.

Przykłady query

#7

Parametr Query q Wszystkie dokumenty

q=*

Parametr facet

True - włączamy faceting.

facet=true

Parametr "field"
Określający pola
Po jakich chcemy
Robić faceting.

facet.field=[category,language]

Dwa pola, jako dwa niezależne
Facetingi.

Przykłady query

#7

```
"facet_counts":{
  "facet_queries":{},
  "facet_fields":{
    "category":[
      "biznes",2,
      "sport",2,
      "nauka",1,
      "poradniki",1],
    "language":[
      "en",2,
      "pl",2,
      "de",1,
      "it",1]},
  "facet_ranges":{},
  "facet_intervals":{},
  "facet_heatmaps":{}}
```

<http://solrhost:8983/solr/books/select?>

[df=description&facet.contains.ignoreCase=false&q=*&facet=true&facet.field=category&facet.field=language](http://solrhost:8983/solr/books/select?df=description&facet.contains.ignoreCase=false&q=*&facet=true&facet.field=category&facet.field=language)

Przykłady query

Scenariusz nr 8:

Sprawdźmy zakresy cenowe książek.

Przykłady query

#8

Parametr Query q

Wszystkie dokumenty

q=*

Parametr facet

True - włączamy faceting.

facet=true

Parametr facet.range
(Włączamy
faceting zakresowy)

Pole po którym
Będzie robiony
Zakres (price)

facet.range=price

Początek zakresu
Dla pola "price"

f.price.facet.range.start=0.0

Koniec zakresu
Dla pola "price"

f.price.facet.range.end=200.0

Każdy kolejny
Zakres co 40 zł

f.price.facet.range.gap=40

Przykłady query

#8

```
},  
"facet_counts":{  
  "facet_queries":{},  
  "facet_fields":{},  
  "facet_ranges":{  
    "price":{  
      "counts":[  
        "0.00,PLN",1,  
        "40.00,PLN",3,  
        "80.00,PLN",1,  
        "120.00,PLN",0,  
        "160.00,PLN",0],  
      "gap":"40.00,PLN",  
      "start":"0.00,PLN",  
      "end":"200.00,PLN"}},  
    "facet_intervals":{},  
    "facet_heatmaps":{}}
```

<http://solrhost:8983/solr/books/select?>

[df=description&facet.contains.ignoreCase=false&q=*&facet=true&facet.range=price&f.price.facet.range.start=0.0&f.price.facet.range.end=200.0&f.price.facet.range.gap=40](http://solrhost:8983/solr/books/select?df=description&facet.contains.ignoreCase=false&q=*&facet=true&facet.range=price&f.price.facet.range.start=0.0&f.price.facet.range.end=200.0&f.price.facet.range.gap=40)

Gratulacje!
Good job;-)