

# IBM Domino a Apache Solr

Ondřej Fuxa

[YOUR SYSTEM, spol. s r.o.](#)

Jan Krejčířek

**SUTOL**  
**Conference2017**

29. listopadu, GreenPoint Praha

# Jan Krejčárek



IBMCHAMPION 



**@jan\_krejcarek**

# Ondřej Fuxa



IBM+Java enthusiast



**@fuxa\_ondrej**



# IBM Domino a Solr

**DMS - Hledání**  
Úvod / DMS / Hledání

Administrator

your

Hledat Zrušit

Celkem nalezeno a filtrováno záznamů: 7.  
Výsledek hledání obsahuje pouze dokumenty DMS ke kterým máte příslušné oprávnění (maximálně 100 záznamů).

Filtrovat

- Typ**  
Faktura (7)  
Zrušit filtr
- Firma**  
YOUR SYSTEM, spol. s r.o. (7)  
Zrušit filtr
- Částka**  
[0 - 50000] (7)  
Zrušit filtr
- Rok**  
2017 (7)  
Zrušit filtr
- Vyřizuje**  
Odbor IS/IT (7)  
Zrušit filtr

Výsledek hledání

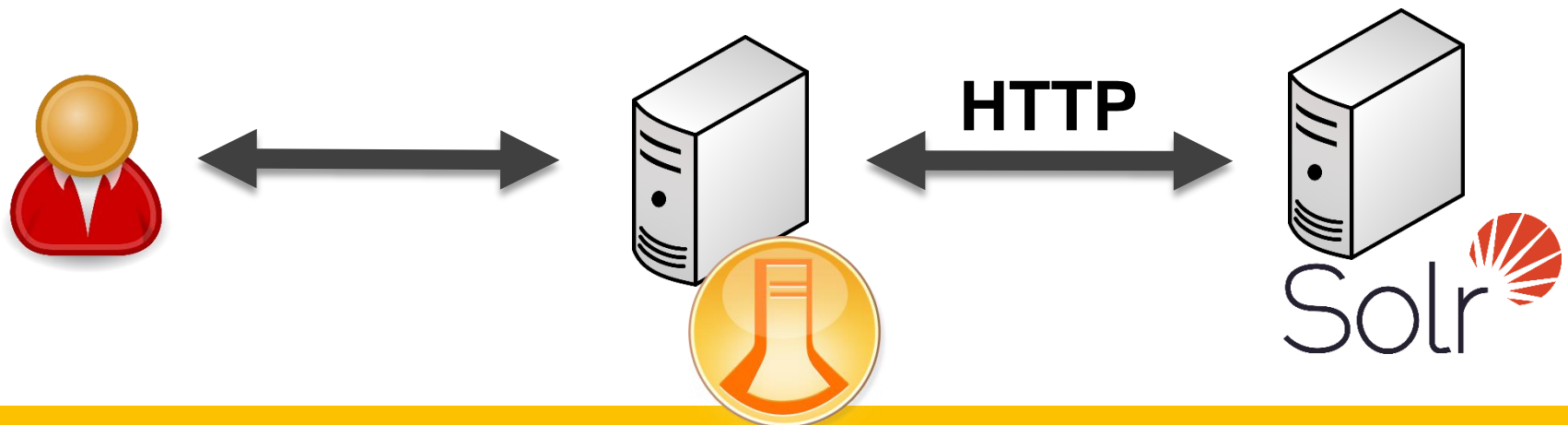
Název	Typ	Firma	Částka v Kč
[redacted]	Faktura	YOUR SYSTEM, spol. s r.o.	[redacted]
[redacted]	Faktura	YOUR SYSTEM, spol. s r.o.	[redacted]
[redacted]	Faktura	YOUR SYSTEM, spol. s r.o.	[redacted]
[redacted]	Faktura	YOUR SYSTEM, spol. s r.o.	[redacted]
[redacted]	Faktura	YOUR SYSTEM, spol. s r.o.	[redacted]
[redacted]	Faktura	YOUR SYSTEM, spol. s r.o.	[redacted]
[redacted]	Faktura	YOUR SYSTEM, spol. s r.o.	[redacted]

**Faceted search**

# Apache Solr – možnosti

- **Kategorizování výsledků a postupné filtrování (faceted search)**
- **Zvýrazňování**
- **Návrhy**
- **„Geospatial search“ pro vyhledávání dle zeměpisných dat**
- **Prioritizace polí při vyhledávání**
- **Statistiky**
- **Větší kontrola nad tím, co se indexuje a jak**
- **Zdarma**

- **Solr je samostatný aplikační server, používá HTTP pro indexaci dat a zpracování dotazů**
- **Uživatelé s ním přímo nepracují, vaše aplikace indexuje data a prezentuje výsledky vyhledávání uživateli**



## Apache Solr

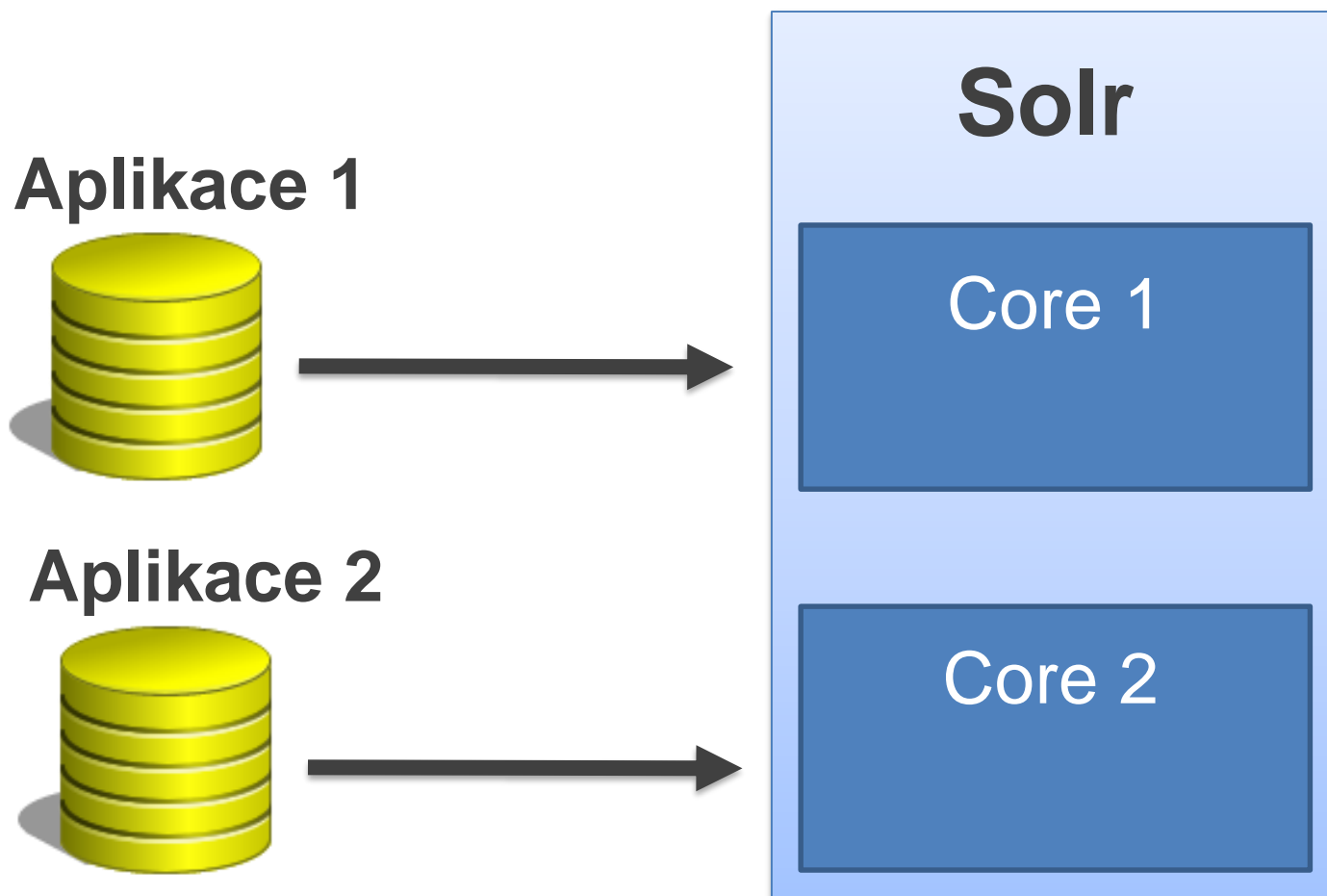
Core 1

Core 2

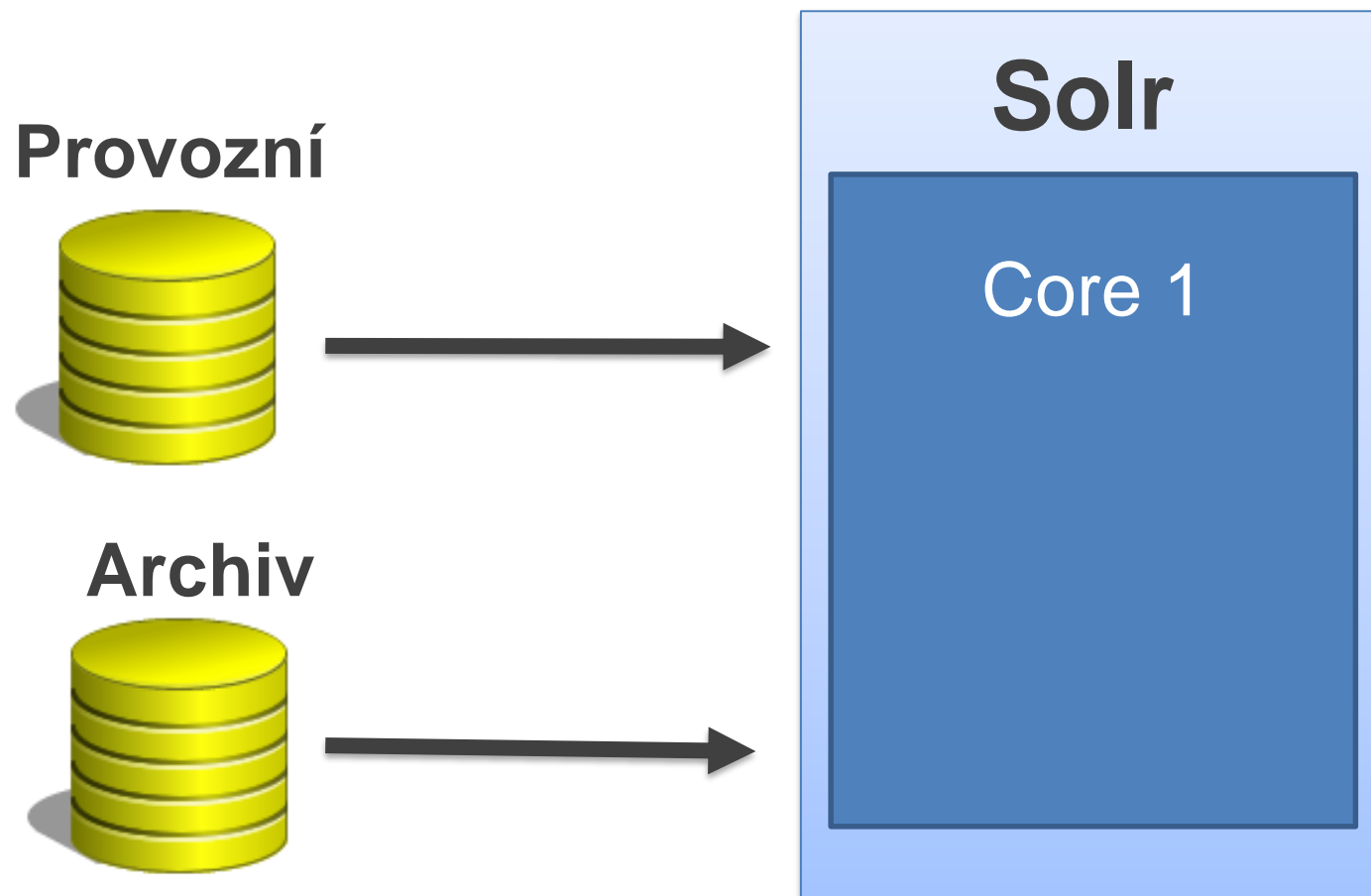
Core N

- **Solr udržuje indexovaná data v oddělených jádrech („core“)**
- **Každý core má vlastní konfiguraci**

- Každá aplikace má zpravidla vlastní konfiguraci a index



- **Více příbuzných aplikací může sdílet jeden index**





# Solr

## Core

Konfigurace

Dokumenty

- **managed-schema**
- **solrconfig.xml**

# Konfigurační soubor managed-schema

- Obsahuje definici polí a typů polí
- Každé pole má název a typ
- Pole mohou být použita pro indexování (indexed = “true”)
- Pole mohou být uložena, později pak mohou být načtena (stored = “true”) – např. obsah dlouhého článku není třeba ukládat, jen se použije pro indexování. Později se načte ID článku a ten je zobrazen z databáze

```
<field name="title" type="text_general",  
indexed="true" stored="true"  
required="true" />
```

# managed-schema

```
<fieldType name="text_general" class="solr.TextField"
positionIncrementGap="100" multiValued="true">
  <analyzer type="index">
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.StopFilterFactory"
ignoreCase="true" words="stopwords.txt" />
    <filter class="solr.LowerCaseFilterFactory"/>
  </analyzer>
  <analyzer type="query">
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.StopFilterFactory"
ignoreCase="true" words="stopwords.txt" />
    <filter class="solr.SynonymGraphFilterFactory"
synonyms="synonyms.txt" ignoreCase="true" expand="true"/>
    <filter class="solr.LowerCaseFilterFactory"/>
  </analyzer>
</fieldType>
```

# Definice typu pole

- **Určuje, co se stane s obsahem pole při indexování a při dotazování. Např. že**
  - **Obsah má být rozdělen do částí (např. slov)**
  - **Častá slova mají být odstraněna**
  - **Text má být převeden na malá písmena**
- **Obsah jednoho pole může být zkopírován do jiného pole jiného typu pro rozdílnou analýzu (velmi časté)**

```
<copyField source="author" dest="author_full"/>
```

# solrconfig.xml

```
<requestHandler name="/search"
class="solr.SearchHandler">
  <lst name="defaults">
    <str name="echoParams">explicit</str>
    <int name="rows">20</int>
    <str name="df">subject</str>
    <str name="defType">edismax</str>
    <str name="qf">subject^20 description^0.8</str>
    <str name="q.alt">*:*</str>
    <str name="mm">100%</str>
  </lst>
</requestHandler>
```

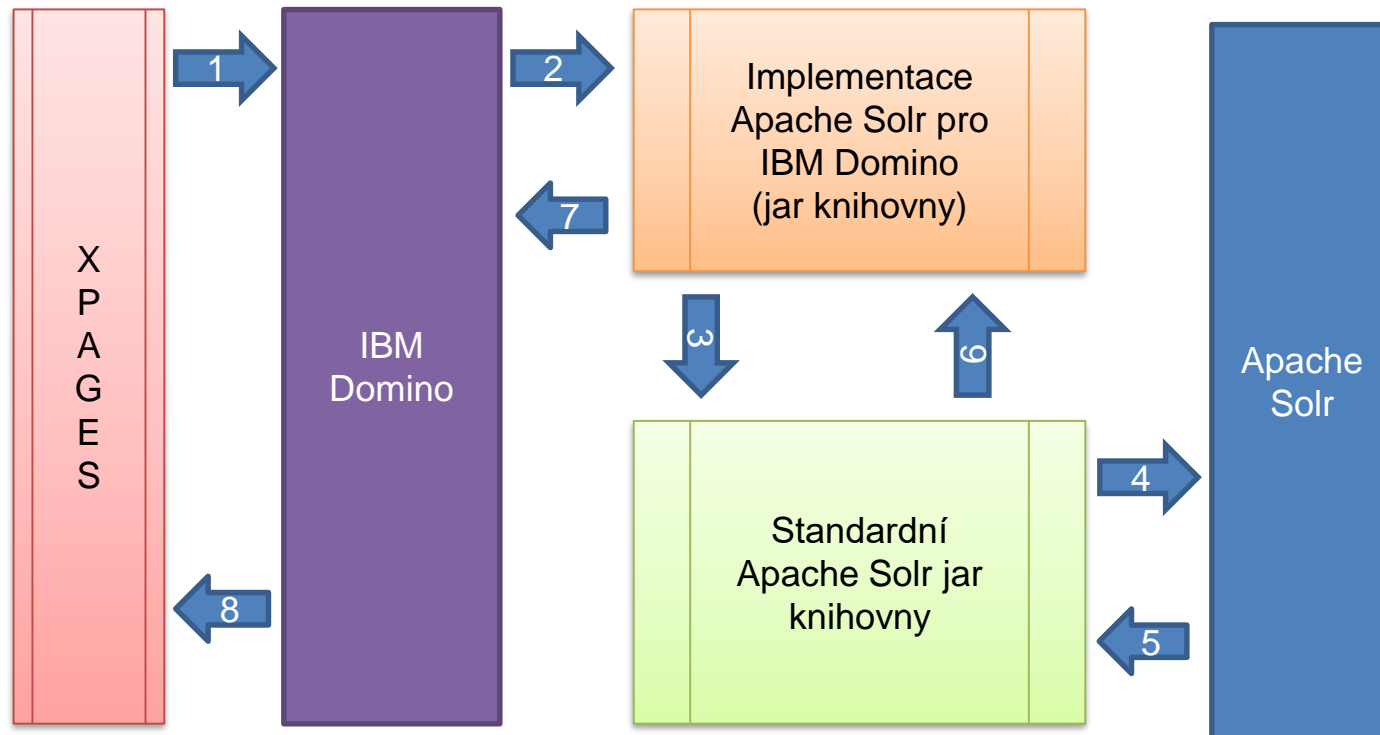
## Další možnosti konfigurace

- **Extrakce obsahu a metadat ze souborů (Word, Excel, PDF, HTML, XML, JPEG, video a dalších ...)**
- **Data Import Handler pro načítání dat z externích zdrojů (relační databáze, RSS, email)**

# Solr a IBM Domino - architektura

- **Stav věci**
  - IBM Domino nekomunikuje s Apache Solr přímo.
  - Apache Solr poskytuje JAVA knihovnu pro integraci.
    - Programátor neřeší HTTP komunikaci ani zpracování JSON nebo XML.
- **Co s tím?**
  - Implementace nadstavby nad standardní Apache Solr knihovnu, která ji rozšíří o funkce pro převod notes dokumentu do Solr dokumentu.
  - Výsledek tohoto snažení použijeme v XPages, v Notes agentovi.

# SOLR a IBM Domino - architektura



1. Uživatel hledá
2. Aplikace zavolá příslušnou metodu Solr rozšíření pro IBM Domino
3. Požadavek je transformován z notes světa do Solr světa
4. Odejde přes HTTP požadavek v interním formátu na Solr server
5. Solr vrátí výsledek hledání
6. Odpověď je transformována do standardních Java tříd Apache Solr
7. Následuje transformace do notes světa
8. Zobrazení odpovědi uživateli



## **Solr a IBM Domino – co jsme udělali?**

- **Abstraktní vrstvu mezi IBM a Solr**
- **Obecná transformace mezi IBM dokumentem a Solr dokumentem**
- **Metody, které vrátí výsledky a facets ve formátu vhodném pro IBM komponenty XPages**
- **Vyhledávání, které respektuje READERS pole**

# Solr a IBM Domino – uniqueKey

- V rámci Solr schématu je potřeba mít identifikaci záznamu v Notes aplikaci (uniqueKey)
  - Update změn
  - Dohledání záznamu pro otevření ze seznamu zobrazených výsledků
- Pokud jedna aplikace, tak stačí **UNID**
- Pokud jeden server, tak stačí **replika ID a UNID**
- V „nejhorším“ případě **název serveru, replikace ID a UNID**

# Solr a IBM Domino – první naplnění dat

- **Jednorázový notes agent**
- **Zpracuje všechny historické záznamy a prvotně naplní Solr**
- **Zkušenost přibližně 80 tis. záznamů trvá 4-5 minut**
  - **Pozor na soft / hard commit 😊**

# Solr a IBM Domino – okamžitá indexace

- **PROBLÉM:** Jak zajistit, že vytvořený / aktualizovaný záznam v notes aplikaci je ihned dostupný přes vyhledávání?
- **ŘEŠENÍ:** TriggerHappy
  - <http://www.openntf.org/main.nsf/project.xsp?r=project/Trigger%20Happy>
  - OpenNTF DLL rozšíření IBM Domino
  - Ověřena funkčnost na IBM Domino 9.0.1 FP9 (WINDOWS)
- Reaguje na nízko úrovněvé operace v nadefinované Notes databázi a když nastanou, tak ihned spustí standardního agenta.
  - Agent zajistí okamžitou indexaci a hard commit tj. Solr search ihned vrací nová data.
- **ZÁVĚR:** Rychlejší než standardní FullText vyhledávání od IBM 😊

# Solr a IBM Domino – bezpečnost

- **Solr neřeší oprávnění ke čtení záznamů**
  - Proto nemá být přímo dostupný pro uživatele
  - FW omezení na port Solr
- **Obecná třída zajišťuje naplnění schématu informacemi o READERS**
  - jména, role, skupiny
- **Automatické doplnění filtru READERS dle přihlášeného uživatele**
  - FILTR, protože neovlivňuje pořadí výsledku dle skóre

# Solr a IBM Domino – XPages (UI)

- **Custom Control**
  - Obecná implementace, tak aby se Facets přidávaly a ubíraly dle toho jak programátor v čisté Java třídě nadefinuje
- **Bootstrap (responzivní design)**
- **MVC přístup**
  - Viz přednáška [Just a View: An Introduction To Model-View-Controller Pattern in XPages](#)
  - Použití view scope proměnné „page“ vložené pomocí ApplicationListener
  - Custom Control čistě UI
  - Kód jen v čisté Java třídě



# Solr a IBM Domino – XPages (UI)

- **1. Custom Control**
  - Rozložení vlastní stránky
  - Pole „Hledat“
  - IBM komponenta „repeat“ pro zobrazení všech facets, které jsou v odpovědi
- **2. Custom Control (ccSearchData)**
  - Pro jednotlivý facet zobrazí všechny hodnoty v něm obsažené
  - Užití opět IBM komponenty „repeat“
  - Vstupní parametr „filterMod“ určuje, zda je možné užít jen jeden filtr v jeden okamžik, či se dají filtry kombinovat

# Solr a IBM Domino – XPages auto facets

```
<xp:repeat id="facetGroups" var="facetGroup" value="#{page.facetGroupNames}" rows="30"
removeRepeat="true"
    repeatControls="false">
  <xp:this.dataContexts>
    <xp:dataContext var="facetItemName" indexVar="facetItemName"
      value="#{javascript:page.getFacetItemName(facetGroup)}">
      </xp:dataContext>
    </xp:this.dataContexts>
    <xp:text tagName="strong" styleClass="icon-left-fa fa-file" value="#{facetGroup}"/>
    <xc:ccSearchData refreshId="content" filterMode="add" categoryName="#{facetItemName}">
      <xc:this.facetResults>
        <![CDATA[#{javascript:page.getFacets(facetGroup)}]]>
      </xc:this.facetResults>
    </xc:ccSearchData>
    <xp:link escape="true" id="removeFilter" title="Zrušit tento filtr"
      rendered="#{javascript:page.isUseFilter(facetItemName)}" styleClass="text-
danger">
      <xp:eventHandler event="onclick" submit="true" refreshMode="partial"
refreshId="content">
        <xp:this.action>
          <![CDATA[#{javascript:page.removeFilter(facetItemName)}]]>
        </xp:this.action>
      </xp:eventHandler>
      <span class="fa fa-times"/>&#160;Zrušit filtr
    </xp:link>
  </xp:repeat>
```



# Solr a IBM Domino – JAVA implementace

```
public long search() {
    SolrDataAccess sda = new SolrDataAccess();

    // nastaveni umistení serveru Apache Solr
    ...

    // nastavení facetu pro rychlou filtraci výsledku
    sda.addFacet("Typ", "type");
    sda.addFacet("Firma", "company_str");
    sda.addFacetInterval("Částka", "amount",
        new String[] {"[0,50000]", "[50001,150000]", "[150001,250000]", "[250001,*]"});
    sda.addFacet("Vyřizuje", "responsible_str");

    // nastavení seznamu použitých filtrů
    ...

    // filtrace dle prav člověka
    ...

    // provede vyhledání položek v Solr
    resultsFound = sda.search(buildQuery());
    pageResults = sda.getCurrentPageResults();
    facets = pageResults.getFacetGroupCollection();

    return resultsFound;
}
```

# Zdroje

- <http://lucene.apache.org/solr/>
- [Solr Reference Guide](#)
- [Apache Solr Enterprise Search Server,  
third edition](#)