Concept - Resources Module / Faceted Search / SolR

Abstract

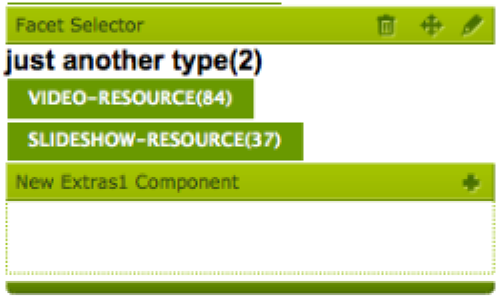
This concept page explains how it would be possible to add solr's faceting possibilities to Magnolia.

Enhancing the existing search with SolR's faceting possibilities

Use Cases

Enabling this, the following use cases could be answered.

- Easy Configuration, adding facets is easy through access to solr fields and teh possibility to facet on evrything that is indexed.

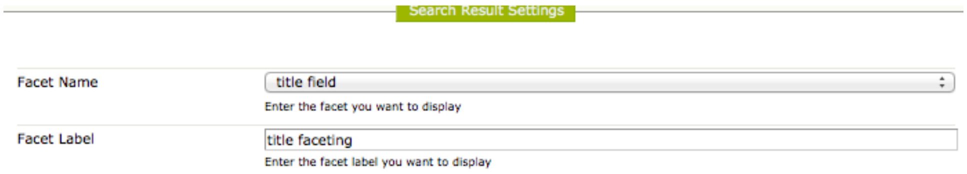
  Facet Selector

  just another type(2)

  VIDEO-RESOURCE(84)

  SLIDESHOW-RESOURCE(37)

  New Extras1 Component

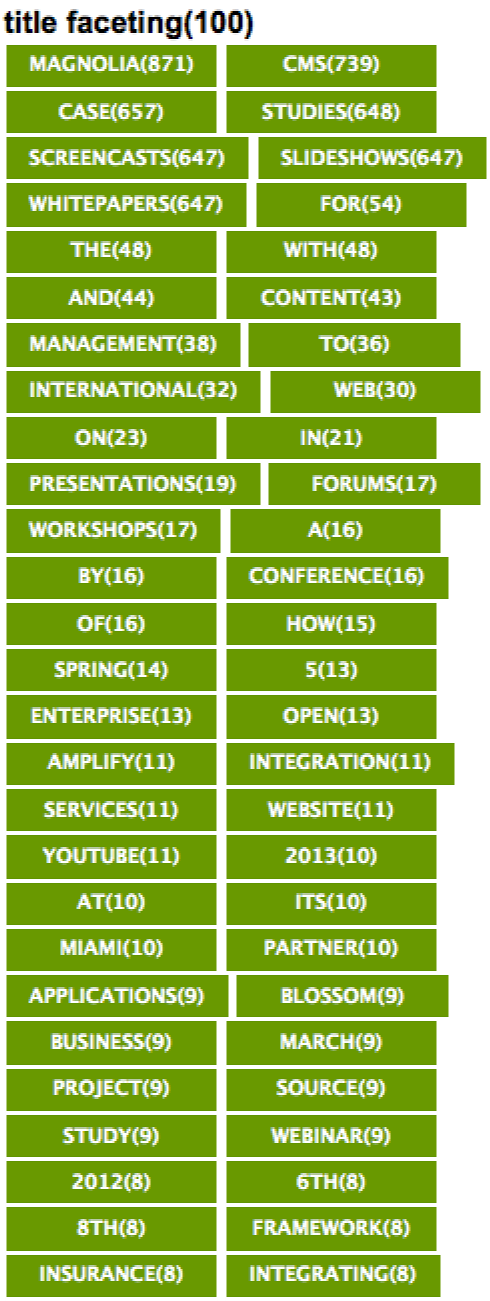
- 

  Search Result Settings

  Facet Name: title field
  Enter the facet you want to display

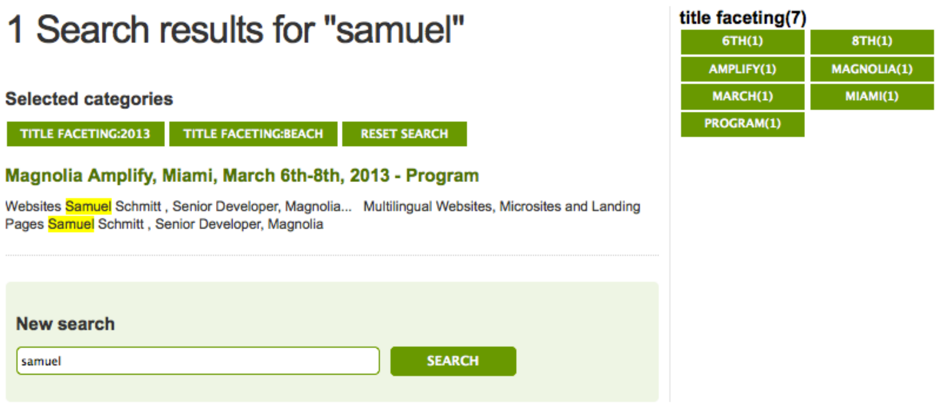
  Facet Label: title faceting
  Enter the facet label you want to display

-

- 

  title faceting(100)

  MAGNOLIA(871) CMS(739)
  CASE(657) STUDIES(648)
  SCREENCASTS(647) SLIDESHOWS(647)
  WHITEPAPERS(647) FOR(54)
  THE(48) WITH(48)
  AND(44) CONTENT(43)
  MANAGEMENT(38) TO(36)
  INTERNATIONAL(32) WEB(30)
  ON(23) IN(21)
  PRESENTATIONS(19) FORUMS(17)
  WORKSHOPS(17) A(16)
  BY(16) CONFERENCE(16)
  OF(16) HOW(15)
  SPRING(14) 5(13)
  ENTERPRISE(13) OPEN(13)
  AMPLIFY(11) INTEGRATION(11)
  SERVICES(11) WEBSITE(11)
  YOUTUBE(11) 2013(10)
  AT(10) ITS(10)
  MIAMI(10) PARTNER(10)
  APPLICATIONS(9) BLOSSOM(9)
  BUSINESS(9) MARCH(9)
  PROJECT(9) SOURCE(9)
  STUDY(9) WEBINAR(9)
  2012(8) 6TH(8)
  8TH(8) FRAMEWORK(8)
  INSURANCE(8) INTEGRATING(8)

- 

  1 Search results for "samuel"

  Selected categories

  TITLE FACETING:2013 TITLE FACETING:BEACH RESET SEARCH

  Magnolia Amplify, Miami, March 6th-8th, 2013 - Program

  Websites Samuel Schmitt , Senior Developer, Magnolia... Multilingual Websites, Microsites and Landing Pages Samuel Schmitt , Senior Developer, Magnolia

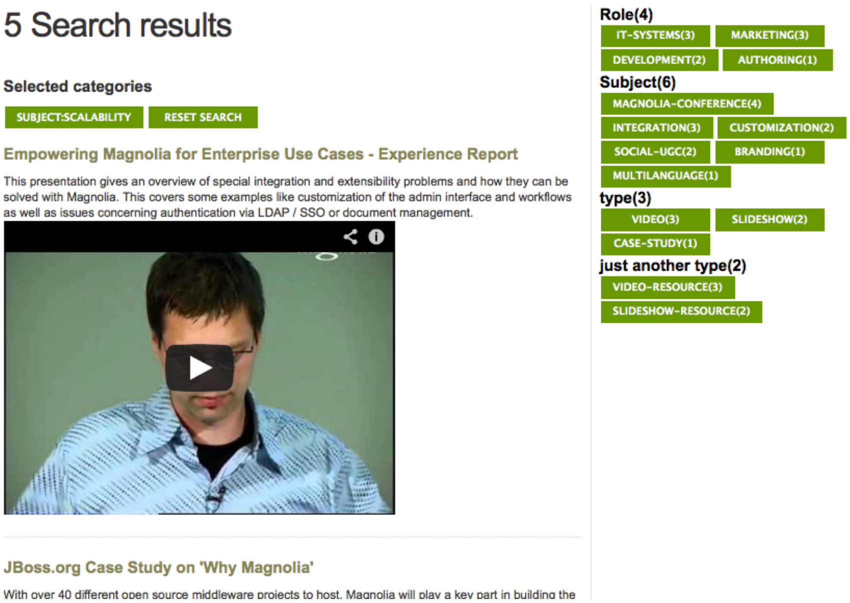
  New search

  samuel SEARCH

  title facting(7)

  6TH(1) 8TH(1)
  AMPLIFY(1) MAGNOLIA(1)
  MARCH(1) MIAMI(1)
  PROGRAM(1)

- Facet/Categorize on all fields submitted to the index and for all content inside the index (DMS/DATA, WEBSITE, Third party)



5 Search results

Selected categories

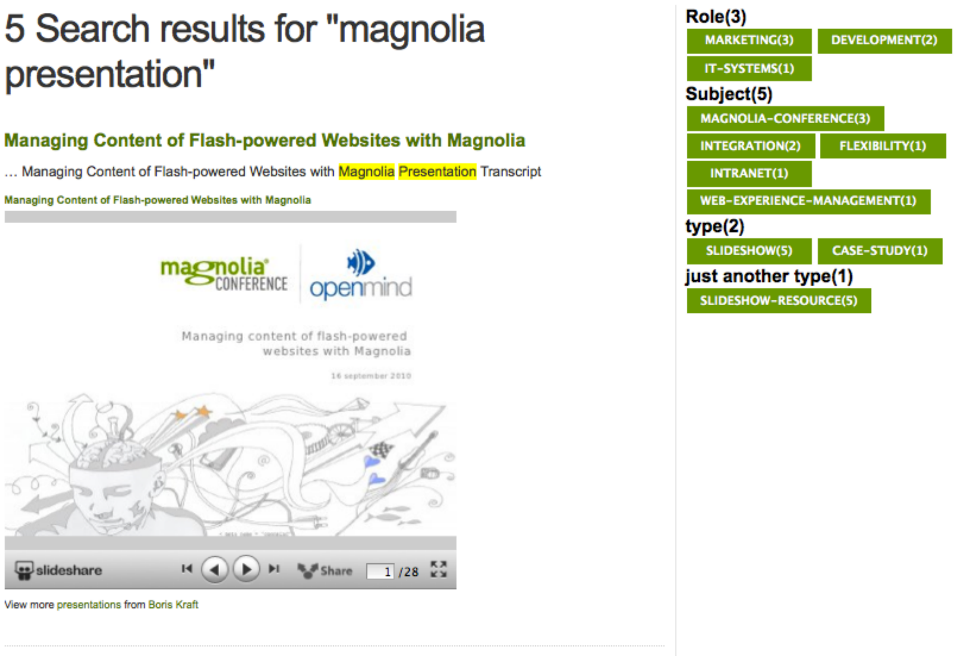
Empowering Magnolia for Enterprise Use Cases - Experience Report

This presentation gives an overview of special integration and extensibility problems and how they can be solved with Magnolia. This covers some examples like customization of the admin interface and workflows as well as issues concerning authentication via LDAP / SSO or document management.

JBoss.org Case Study on 'Why Magnolia'

With over 40 different open source middleware projects to host, Magnolia will play a key part in building the

- Do keyword based searches in faceted content, get the current facets for a specific keyword search.



5 Search results for "magnolia presentation"

Managing Content of Flash-powered Websites with Magnolia

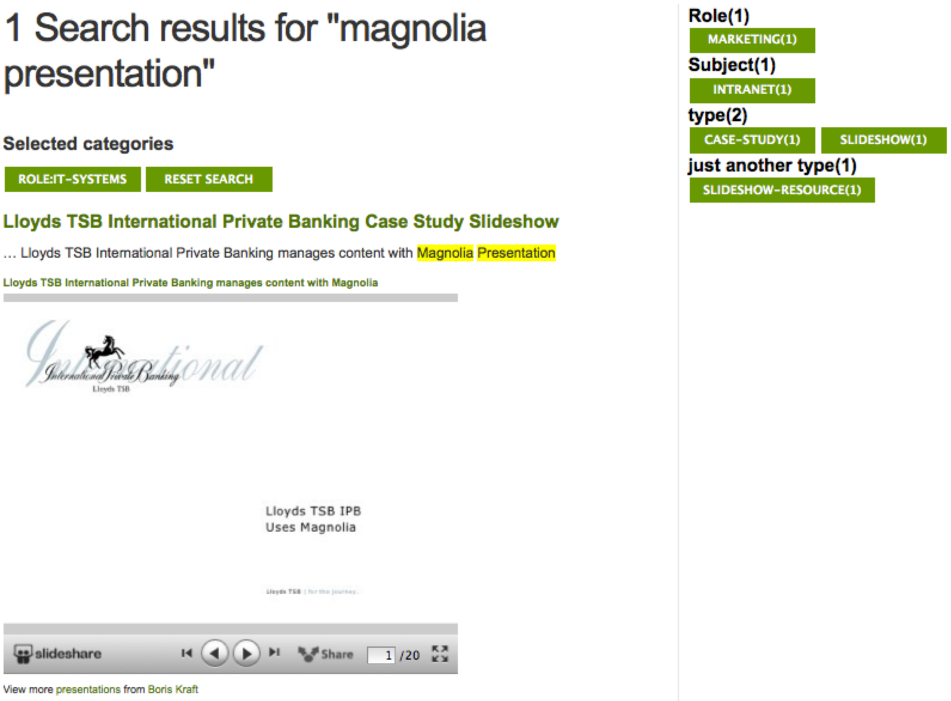
... Managing Content of Flash-powered Websites with Magnolia Presentation Transcript

Managing Content of Flash-powered Websites with Magnolia

View more presentations from Boris Kraft

-

    - The above keyword search gives the associated categories that are available for teh specific search, refining is possible by clicking again on one of the items, for instance clicking on IT Systems will give us teh only result matching IT-Systems and "Magnolia Presentation".



1 Search results for "magnolia presentation"

Selected categories

ROLE:IT-SYSTEMS | RESET SEARCH

Lloyds TSB International Private Banking Case Study Slideshow

... Lloyds TSB International Private Banking manages content with Magnolia Presentation

Lloyds TSB International Private Banking manages content with Magnolia

View more presentations from Boris Kraft

Role(1)

MARKETING(1)

Subject(1)

INTRANET(1)

type(2)

CASE-STUDY(1) | SLIDESHOW(1)

just another type(1)

SLIDESHOW-RESOURCE(1)

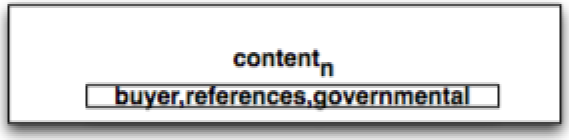
-
- Do range faceting (price/dates) propose a general search interface for product/e-commerce sites.
- Be able to provide user context content paths, maybe this could be another concept page on its own.
  - First all content is categorized, each content must have at least one user profile categorization (developer, marketing, buyer, ...)

content$_1$

developer,java,spring

content$_2$

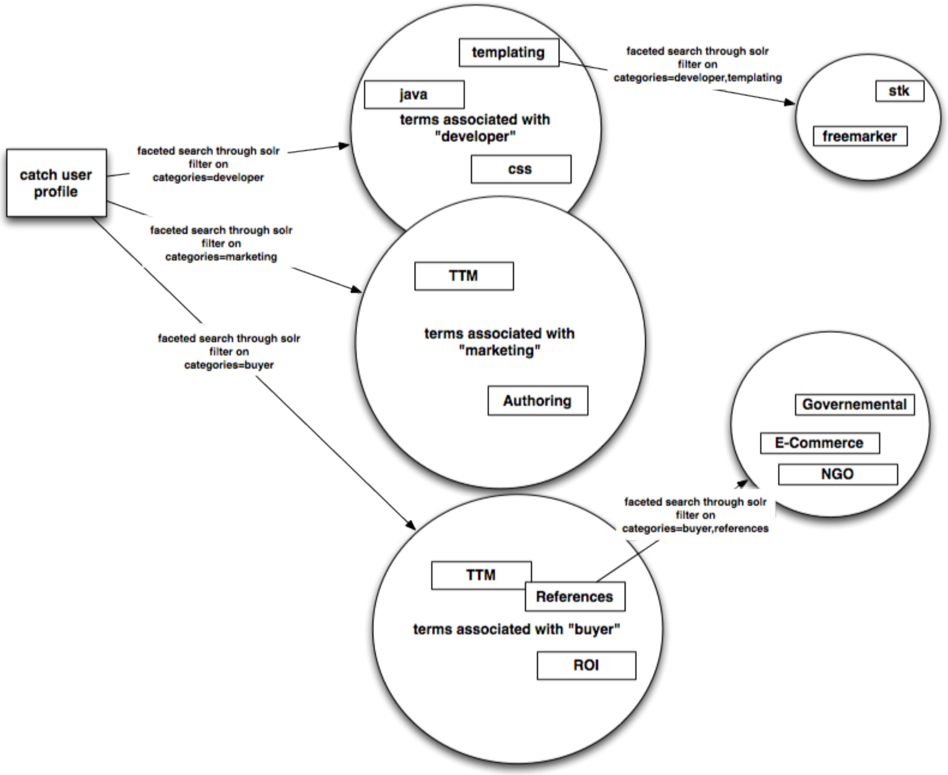
developer,templating,freemarker



content$_n$

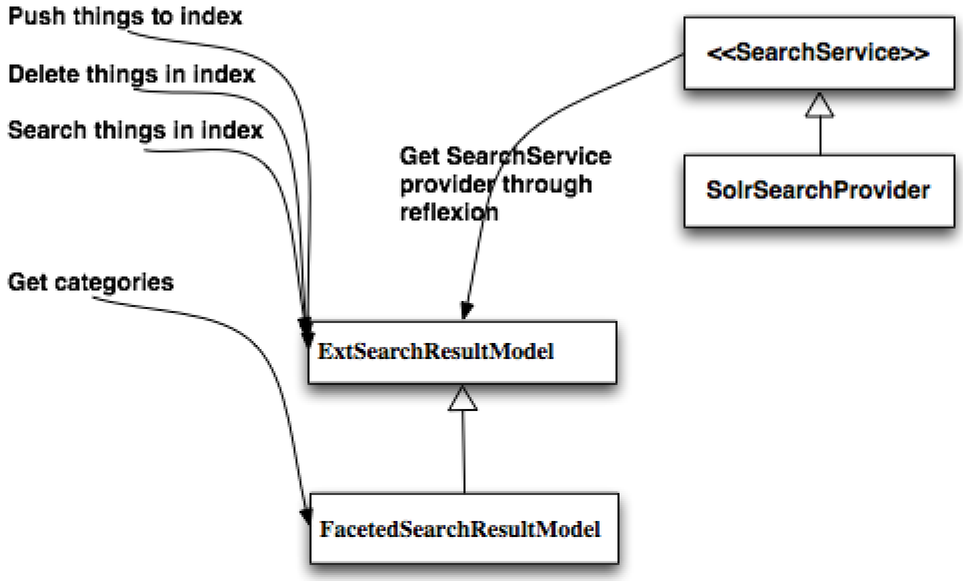
buyer,references,governmental

  -
  - Then, navigation is done through solr's faceted search.Based on a few initial choices, different layouts can be proposed after each refining.


catch user profile
faceted search through solr filter on categories=developer
templating
java
terms associated with "developer"
css
faceted search through solr filter on categories=developer,templating
stk
freemarker
faceted search through solr filter on categories=marketing
TTM
terms associated with "marketing"
Authoring
faceted search through solr filter on categories=buyer
Governemental
E-Commerce
NGO
faceted search through solr filter on categories=buyer,references
TTM
References
terms associated with "buyer"
ROI

Proposed architecture

We will try to make things as generic as possible, to be able to use as well other search providers, we extend the ExtSearchResultModel Class with the FacetedSearchResultModel Class which will only contain the specific getters/setters for faceting.



How do we push the categories to the index ?

Two things have to be distinguished here, content from the website and assets like documents, movies and other stuff.

Website content

Content from the website is already picked up by the Heritrix crawler that calls the provider instance through the Extended Search configuration and pushes urls to the solr server which will extract all content and index it.

To add categorization tags, we can add categories to a meta field in the page by adding a script in the HtmlHeader template as follows.

```
[#assign categories = pageModel.categories!]
[#assign hasCategories = categories?has_content]

[#function getCats itemlist]
    [#-- Assigns: Get Content from List Item--]
    [#local cats = ""]
    [#list itemlist as item]
       [#local itemName = item.@name]
       [#local itemDisplayName = item.displayName!itemName]
       [#local cats = itemDisplayName + "," + cats]
    [/#list]
    [#return cats]
[/#function]


[#if hasCategories]
<meta name="categories" content="${getCats(categories)}"/>
[/#if]
```

Enabling this, Solr's tika parser will pick up stuff in meta categories field, and index it if teh solr scheme has a corresponding categories field, now this is nice but what if we want to create other facets, like it is done with the resources module. In the resources module we have "root" categories that we can call facets like resources_role, resources_subject, you would not like to modify your scheme each time you add other facets to magnolia no ?

This is where the power of solr enters the game, in solr you can add dynamic fields which will be created if they do not exist in the index, to do so we added the following field in solr's scheme.

```
<dynamicField name="category_*" type="category" stored="true" indexed="true" multiValued="true"/>
```

This tells Solr to automatically create a category field each time a facet that starts with category_ is added to the index, this means that if a meta field as follows is sent to the index;

```
<meta name="category_resources_role" content="constraint1, constraint2"/>
```

category_resources_role is created in the scheme and constraint1 and constraint2 are indexed under this field or facet.

> ⓘ The choice to either prepend the "category_" prefix to the categories "root" category in magnolia or to prepend it when submitting the content, especially when submitting resources from the JCR data repository is an implementation decision.

Now what about JCR content that is not accessible by the crawler.

This type of content can be send either by performing an extract of the data, converts it to have the correct solr syntax and submits it to the index on bulk or batch basis, or through a JCREventListener that will submit the content once it is available for publishing.

I wrote the following command to index video_resources and slideshow_resources to the solr index, this of course has to be enhanced by finding maybe a way through workflow to index or not the specified content.

```
 private final static String XpathData = "//*[((@jcr:primaryType='slideshow-resource') or (@jcr:
primaryType='video-resource'))]";


    /* (non-Javadoc)
     * @see info.magnolia.commands.MgnlCommand#execute(info.magnolia.context.Context)
     */
    @Override
    public boolean execute(Context context) throws Exception {

        Session session = context.getJCRSession("data");
        QueryManager qm = session.getWorkspace().getQueryManager();
        Query query = qm.createQuery(XpathData, "xpath");
        NodeIterator nodeIt = query.execute().getNodes();


        /**
         * This call is important since we ask access to the search provider instance
         *
         */
        SearchService<?, ?, ?, String> svc = EsUtil.getProviderInstance();

        while(nodeIt.hasNext()){
            Node current = nodeIt.nextNode();
            Map<String,String>things = this.prepareThings(current, session);
            if(things!=null){
                svc.addUpdate(RepositoryEntries.DAM.name(), things);
            }
        }


        return true;
    }

    private Map<String,String> prepareThings(Node current,Session session){

        Map<String,String>things = new HashMap<String,String>();
        ContentMap mp = new ContentMap(current);
        Map<String,List<String>> categorySet = extractCategories(((String[])mp.get("categories")),session);
        //put categories
        /**
         * Here we already give the correct syntax for usage with solr,
         * This is not ok since this class should be agnostic, we can overcome this by creating a generic
format and converters for each format in the provider package's logic
         *
         */
        for(String facet:categorySet.keySet()){
            things.put("literal.category_"+facet,StringUtils.join(categorySet.get(facet),","));
        }
        String abstrakt = (String)mp.get("abstract");
        things.put("literal.abstract", abstrakt);
        String itemType = (String) mp.get("itemtype");
        things.put("literal.type", itemType);
        String link = (String) mp.get("link");
        things.put("literal.htmllink", link);
        String url = extractURL(link);
        things.put("literal.url", url);
        String id=null;
        try {
            id = URLEncoder.encode(url,"UTF-8");
        } catch (UnsupportedEncodingException e) {
            log.warn("could not encode id"+e.getMessage());
            return null;
        }
        things.put("literal.id", id);
        String name = (String) mp.get("name");
        things.put("literal.title", name);
        log.debug(name+"<======>"+url+"<=====>"+StringUtils.join(categorySet.values(),"-"));
        //cr:lastModified,width,nodeDataTemplate,jcr:data,depth,jcr:uuid,size,extension,id,height,name,path,jcr:
mimeType,fileName,nodeType,jcr:primaryType
        String thumbnail = (String) ((ContentMap)mp.get("thumbnail")).get("name");
        log.debug("node's thumbnail name:"+thumbnail);

        return things;
    }
```