

How-To: Implement a multi-institutional search site

Introduction

This wiki page provides information about how to implement a customized search site with functionalities similar to <http://beta.vivosearch.org/>.

8/2013: Note that the code linked from this page is obsolete and is being redeveloped as [presented at the 2013 VIVO Conference](#)

- see also the [VIVO Multisite Search wiki space](#)

Useful resources:

- mailing list: [vivo-dev-all](#)
- list archive: [List Archive](#)
- main wiki: [Wiki Home](#)
- Live IRC chat room: [#VIVO](#)

2011 Downloads

- Apache Solr from <http://lucene.apache.org/solr/>
- Linked Data Index Builder from <https://sourceforge.net/p/vivo/tools/code/>
- Drupal from <http://drupal.org/>
- VIVO search Drupal module from <https://github.com/milesworthington/vivosolr>
- MySQL from <http://www.mysql.com/>

Setting up the applications

Dates from late 2011

1. Apache Solr

- Unpack the downloaded Apache Solr file
- Copy the files schema.xml and solrconfig.xml from /LinkedDataIndexer/solr to \$APACHE_SOLR_HOME/example/solr/conf
- Go to \$APACHE_SOLR_HOME/example and run `java -jar start.jar` to start solr
- Open <http://localhost:8983/solr> and check if solr is running

2. LinkedDataIndexer

- Go to the directory of LinkedDataIndexer
- Copy the example file /src/main/scala/edu/cornell/indexbuilder/configurations/ExampleConfig.scala to your customized scala file (e.g. MyInstitutionTest.scala)
- Modify the copied file from:

```
object ExampleConfig {
  val siteName = "Weill Cornell Medical College"
  val siteUrl = "http://vivo.med.cornell.edu"
  val siteVivoVersion = VitroVersion.r1dot2

  val solrUrl = "http://exampleSolrServer.example.edu:8080/solr"
  val classUris = List( " "http://vivoweb.org/ontology/core#Postdoc" " )
}
```

to your customized institutional configurations:

```
object MyInstitutionTest {
  val siteName = "My Institution"
  val siteUrl = "http://vivo.myinstitution.edu"
  val siteVivoVersion = VitroVersion.r1dot2

  val solrUrl = "http://localhost:8983/solr"
  val classUris = List( " "http://vivoweb.org/ontology/core#Postdoc" " )
}
```

- The file /LinkedDataIndexer/src/main/resources/akka.conf needs to be modified by changing the statement: `version = "1.1.3"` to `version = "1.2"`
- The following are needed to compile or run the index builder
- : scala from <http://www.scala-lang.org>
- : sbt from <https://github.com/harrah/xsbt/wiki/>
- : slf4s_2.9.1-1.0.7.jar

- : slf4j-api-1.6.3.jar
- : akka-actor-1.2.jar
- : commons-httpclient.jar
- : slf4j-log4j12-1.6.3.jar
- Go to the directory of LinkedDataIndexer
- Type `sbt` to start up
- Type `run`
- : Note: You may encounter errors complaining about missing jar files. In this case, you will need to add more jar files to the lib directory.
- Select your customized configuration to run the builder
- : Note: Apache Solr needs to be running at the time of building the index.

3. Drupal

- Follow the instructions to install and configure a basic Drupal site
- Copy vivosolr to `$DRUPAL_HOME/modules`
- Open your local drupal site
- Log in as administrator
- Click on the Modules tab and enable "VIVO Solr Search Integration"
- Click on the Configuration tab and configure "VIVO search" to your solr installation for "Solr server URL".
- To allow all viewers to use the search form without logging in, do the following:
- : Click on the Modules tab
- : Click on the "Permissions" text link for "Block"
- : Check the boxes for "ANONYMOUS USER" and "AUTHENTICATED USER" for "Use search" and "Use advanced search"
- Log out
- Test the search capability