

Welcome to the org.apache.forrest.plugin.output.solr Plugin

Table of contents

1 Apache Forrest - org.apache.forrest.plugin.output.solr Plugin.....	1
2 Dispatcher activation.....	1
3 Solr activation.....	2
4 Samples.....	3

1. Apache Forrest - org.apache.forrest.plugin.output.solr Plugin

Generates [solr](#) documents from xdos. Further when run with the dispatcher in static mode it allows you to update a Solr server with the content of your website at the same time you generate the site (solr-add contract). In dynamic mode (as servlet) it provides a GUI to manage your project in solr (solr-actionbar contract) and a search interface (solr-search contract) to search your solr server.

2. Dispatcher activation

You can see some [screenshots](#) and activate the different contracts by using following snippets. Explanation given within xml comments.

```
<!-- default plugin CSS -->
<forrest:contract name="branding-css-links">
  <forrest:property name="branding-css-links-input">
    <css url="solr.css" media="screen" theme="Pelt"/>
  </forrest:property>
</forrest:contract>
<!-- In static mode solr-add allows you to update a Solr server with
the content of your
website at the same time you generate the site. -->
<forrest:contract name="solr-add">
  <forrest:property name="request">#{ $getRequest }</forrest:property>
</forrest:contract>
<forrest:hook name="solr">
```

```
<forrest:hook class="solrSearch">
  <!-- Search interface (solr-search contract) to
  search your solr server. -->
  <forrest:contract name="solr-search">
    <forrest:property
name="contextPath">#{$contextPath}</forrest:property>
    <forrest:property name="hidden">
      <input name="hl" type="hidden" value="true"/>
      <input name="hl.fl" type="hidden" value="content"/>
      <input name="fl" type="hidden" value="id,title"/>
      <input name="indent" type="hidden" value="on"/>
      <input name="rows" type="hidden" value="10"/>
      <input name="start" type="hidden" value="0"/>
    </forrest:property>
  </forrest:contract>
</forrest:hook>
<!-- If the userAgent is the cocoon crawler then we do not want to
output the actionbar. The cocoon crawler does not provide the request
header "User-Agent".-->
<!-- HEADSUP
  The jx:set var="userAgent" must be defined in the main structurer
  and cannot go into a tiles! -->
<jx:set var="userAgent"
value="#{$cocoon.request.getHeader('User-Agent')}" />
<jx:if test="#{$!userAgent.equals('')&null!=userAgent}">
  <forrest:hook class="solrActionBar">
    <!-- GUI to manage your project in solr. -->
    <forrest:contract name="solr-actionbar">
      <forrest:property
name="request">#{$getRequest}</forrest:property>
      <forrest:property name="contextPath">
        #{$contextPath}</forrest:property>
    </forrest:contract>
  </forrest:hook>
</jx:if>
</forrest:hook>
```

You can as well use the above aggregated to a tiles, like (make sure you copy both lines!):

```
<jx:set var="userAgent" value="#{$cocoon.request.getHeader('User-Agent')}" />
<jx:import uri="cocoon://prepare.tiles.solrbar"/>
```

3. Solr activation

The host server urls can be configured by adding the following properties to your project forrest.properties.xml in case you do not use the default values.

```
<property name="solr.select.url"
value="http://localhost:8983/solr/select"/>
<property name="solr.update.url"
value="http://localhost:8983/solr/update"/>
```

In your solr server add the following field definitions to the schema.xml. We assume "id" is already set as "uniqueKey" and you changed the "defaultSearchField" to "content" otherwise

you need to prefix your queries.

```
<field name="content" type="text" indexed="true" stored="true" />
<field name="title" type="text" indexed="true" stored="true"/>
<field name="subtitle" type="text" indexed="true" stored="true"/>
<field name="abstract" type="text" indexed="true" stored="true"/>
<field name="version" type="text" indexed="true" stored="true"/>
<field name="author" type="text" indexed="true" stored="true"
multiValued="true"/>
```

4. Samples

This section includes links to a number of samples using this plugin. The plugin is intended to be self documenting through these samples. If you require further help please ask on the user mailing list.

- [index.solr](#) - the xdocs-to-solrDoc.xsl transforms xdocs to solr docs.
- [index.solr.add](#) - surround the solr docs with an add statement.
- [index.solr.delete](#) - surround the solr docs id with a delete statement.
- [solr.commit](#) - commit statement for solr.
- [solr.optimize](#) - optimize statement for solr.
- [index-creation.solr.add](#) - is generating and adding an "add" statement for solr with the whole site as solr docs as input. This will index your whole project if you add ".do.html" to the url.

Above pages suffixed with `.do.html` will trigger the underlying statement on the solr server. We do not link them in this document since the links would be crawled. This is as well the reason why the gui is deactivated. Each the gui would trigger the delete, add, commit and optimize statement against the solr server for every page.