Searchlight

Concepts & overview
1.x-alpha

drupal db

doc: node

id

content

content

attr

attr
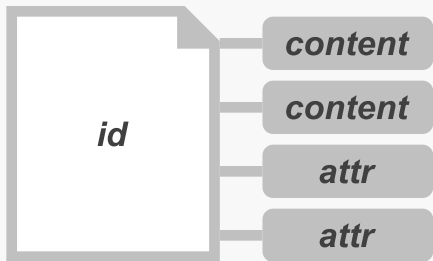
The datasource describes the Drupal db's fields and tables using the Views API,

d 🔍

datasource: node table

To the search backend as a document index with content and attribute fields.

datasource:
node table

backend:
Sphinx, Solr

conf

configuration
file(s)

The Search backend plugin uses the
datasource configuration to write one or
more configuration file(s) specific to the
search service (Sphinx, Solr, etc.)

content

content

attr

attr

id

doc: node
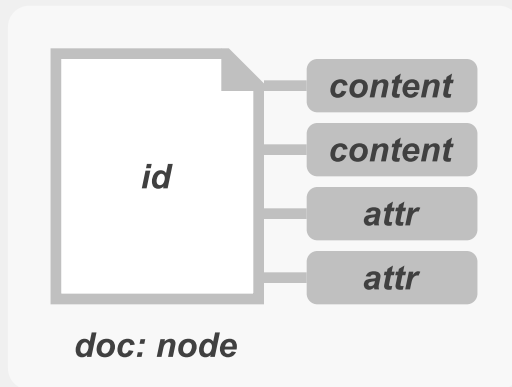
Sphinx *pulls* documents from the Drupal DB using SQL queries.

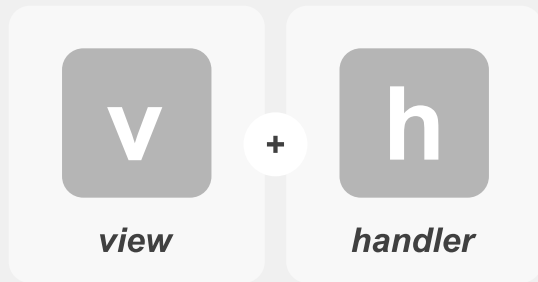Documents are *pushed* to Solr using an XML document format.

i

vs

i

index (Sphinx)

index (Solr)

v + **h**

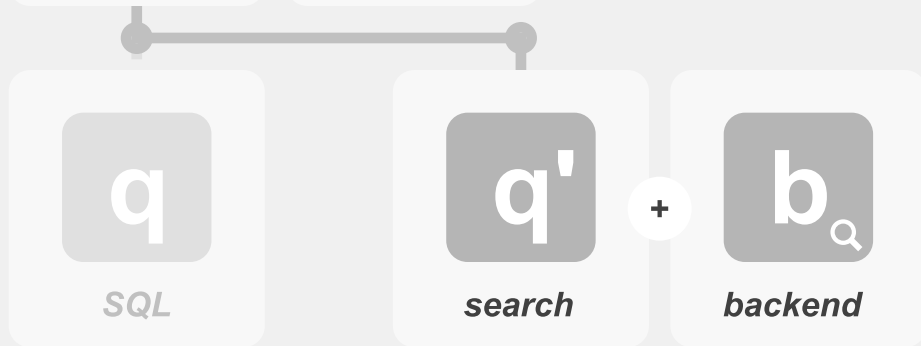*view*   *handler*

**q**   **q'** + **b**

SQL   *search*   *backend*

When a view with a Searchlight handler (filter, argument, etc) is built, the default SQL query plugin is replaced with a Searchlight query plugin that interfaces views with the Search backend.

v

view

+

h

filters

+

h

sorts

q'

search

b

backend

add_where()

add_orderby()

setFilter()

setSort()

As views handlers build the views query, the Searchlight query plugin translates SQL into attribute filters, sorts, etc. for the backend plugin.

v + h

view **handler**

q q' + b

SQL **search** **backend**

r i

view->result *index*
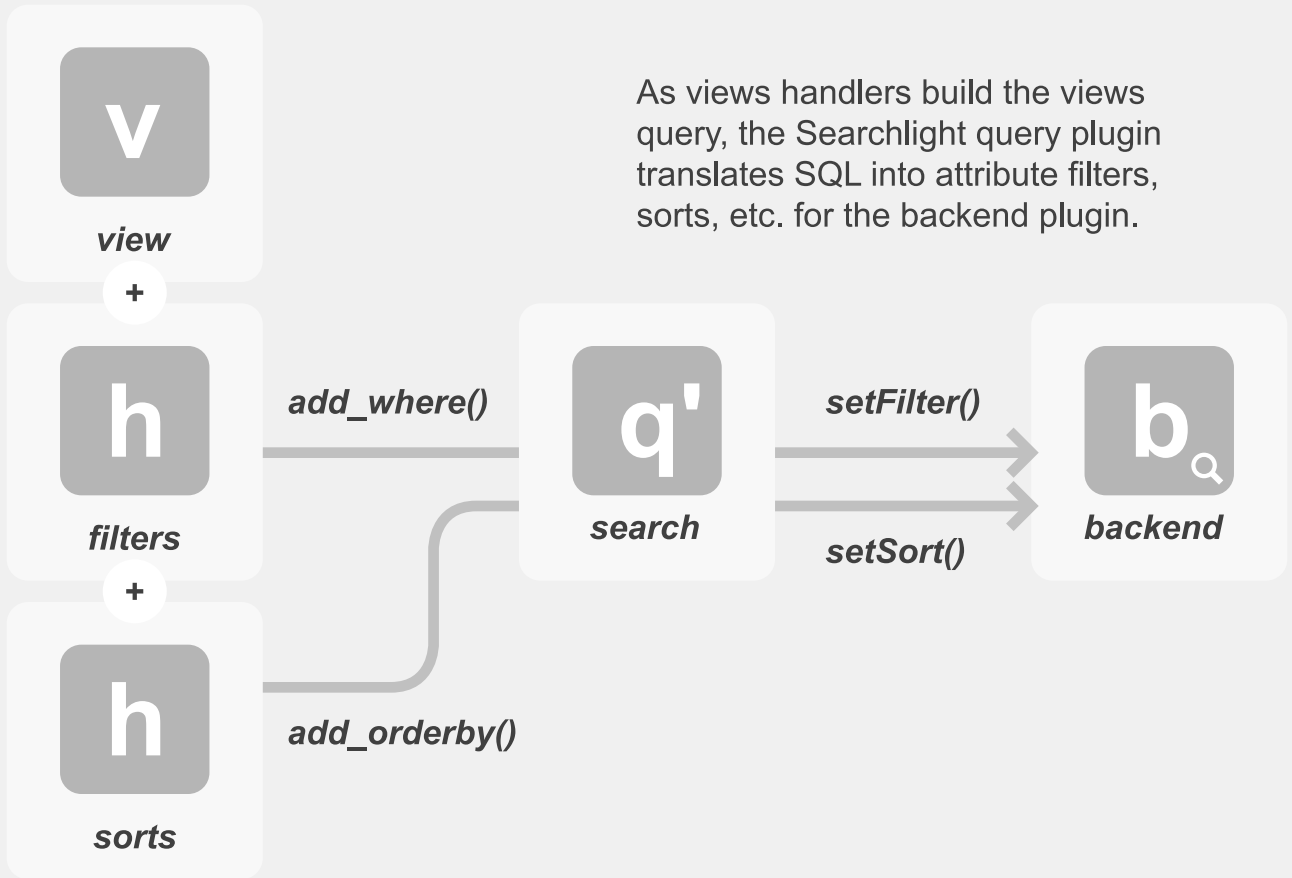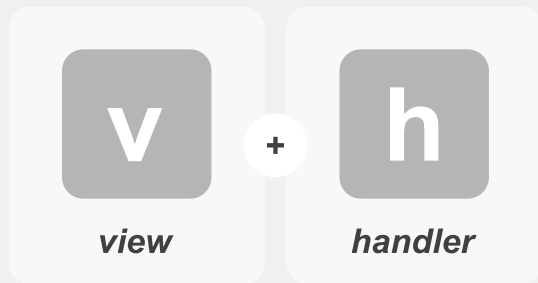
When the view is executed the backend
dispatches a query against the search
service and retrieves the results that will
be passed through the rest of the views
render stack.