intake_solr Documentation

Release 0.0.4+10.g37a313a.dirty

Joseph Crail

Jul 23, 2018

Contents:

CHAPTER 1

Quickstart

`intake-solr` provides quick and easy access to tabular data stored in Apache SOLR

This plugin reads SOLR query results without random access: there is only ever a single partition.

1.1 Installation

To use this plugin for intake, install with the following command:

```
conda install -c intake intake-solr
```

1.2 Usage

1.2.1 Ad-hoc

After installation, the functions `intake.open_solr_table` and `intake.open_solr_sequence` will become available. The former method can be used to return the results of a SOLR query into a dataframe, but the latter will produce a generic sequence of dictionaries.

Given the query `text:test`, the following would load into a dataframe:

```
import intake
source = intake.open_solr_dataframe("text:test")
dataframe = source.read()
```

Three parameters are of interest when defining a data source:

- query: the query to execute, which can be defined either using Lucene or **'JSON'_** syntax, both of which are to be provided as a string.

1.2.2 Creating Catalog Entries

To include in a catalog, the plugin must be listed in the plugins of the catalog:

```
plugins:
  source:
    - module: intake_solr
```

and entries must specify `driver: solr_table` or `driver: solr_sequence`. The further arguments are exactly the same as for the `open_solr_*` functions.

1.2.3 Using a Catalog

Assuming a catalog file called `cat.yaml`, containing a SOLR source `data`, one could load it into a dataframe as follows:

```python
import intake
cat = intake.Catalog('cat.yaml')
df = cat.data.read()
```

The type of the output will depend on the plugin that was defined in the catalog. You can inspect this before loading by looking at the `.container` attribute, which will be either `"dataframe"` or `"python"`.

CHAPTER 2

API Reference

intake_solr.source.SOLRTableSource(query, . . .)	Execute a query on SOLR, return as dataframe
intake_solr.source.SOLRSequenceSource(query, . . .)	Execute a query on SOLR

class intake_solr.source.**SOLRTableSource**(*query*, *base_url*, *core*, *qargs=None*, *metadata=None*, *auth=None*, *cert=None*, *zoocollection=False*)

Execute a query on SOLR, return as dataframe

Parameters

query: str Query to execute, in Lucene syntax, e.g., `"*:*"`

base_url: str Connection on which to reach SOLR, including protocol (http), server, port and base path. If using Zookeeper, this should be the full comma-separated list of service:port/path elements.

core: str Named segment of the SOLR storage to query

qargs: dict Further parameters to pass with the query (e.g., highlighting)

metadata: dict Additional information to associate with this source

auth: None, "kerberos" or (username, password) Authentication to attach to requests

cert: str or None Path to SSL certificate, if required

zoocollection: bool or str If using Zookeeper to orchestrate SOLR, this is the name of the collection to connect to.

Attributes

datashape

description

`hvplot` Returns a hvPlot object to provide a high-level plotting API.

`plot` Returns a hvPlot object to provide a high-level plotting API.

Methods

`close()`	Close open resources corresponding to this data source.
`discover()`	Open resource and populate the source attributes.
`read()`	Load entire dataset into a container and return it
`read_chunked()`	Return iterator over container fragments of data source
`read_partition(i)`	Return a (offset_tuple, container) corresponding to i-th partition.
`to_dask()`	Return a dask container for this data source
`yaml()`	Return YAML representation of this data-source

class `intake_solr.source.`**`SOLRSequenceSource`**(*query*, *base_url*, *core*, *qargs=None*, *metadata=None*, *auth=None*, *cert=None*, *zoocollection=False*)

Execute a query on SOLR

Parameters

query: str Query to execute, in Lucene syntax, e.g., `"*:*"`

base_url: str Connection on which to reach SOLR, including protocol (http), server, port and base path. If using Zookeeper, this should be the full comma-separated list of service: port/path elements.

core: str Named segment of the SOLR storage to query

qargs: dict Further parameters to pass with the query (e.g., highlighting)

metadata: dict Additional information to associate with this source

auth: None, "kerberos" or (username, password) Authentication to attach to requests

cert: str or None Path to SSL certificate, if required

zoocollection: bool or str If using Zookeeper to orchestrate SOLR, this is the name of the collection to connect to.

Attributes

datashape

description

`hvplot` Returns a hvPlot object to provide a high-level plotting API.

`plot` Returns a hvPlot object to provide a high-level plotting API.

Methods

`close()`	Close open resources corresponding to this data source.
`discover()`	Open resource and populate the source attributes.

Continued on next page

Table 3 – continued from previous page

`read()`	Load entire dataset into a container and return it
`read_chunked()`	Return iterator over container fragments of data source
`read_partition(i)`	Return a (offset_tuple, container) corresponding to i-th partition.
`to_dask()`	Return a dask container for this data source
`yaml()`	Return YAML representation of this data-source

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Index