
intake_accumulo Documentation

Release 0.1.1+0.gf6dd5e5.dirty

Joseph Crail

Feb 05, 2019

Contents:

1	Quickstart	3
1.1	Installation	3
1.2	Example: Reading Accumulo table without catalog	3
1.3	Example: Reading Accumulo table with catalog	4
2	API Reference	5
3	Indices and tables	7

This package enables the Intake data access and cataloging system to access data stored in Apache Accumulo.

This guide will show you how to get started using Intake to read an Accumulo table.

1.1 Installation

For conda users, the Intake Accumulo plugin is installed with the following commands:

```
conda install -c intake intake-accumulo
```

1.2 Example: Reading Accumulo table without catalog

The simplest use case for this plugin is to read an existing Accumulo table. Assuming the Accumulo instance is located at `localhost:42424` and the table is in the variable, `table`, this will read the entire table into a dataframe.:

```
>>> import intake
>>> ds = intake.open_accumulo(table)
>>> df = ds.read()
>>> df
```

	row	column_family	column_qualifier	column_visibility	time_u
↪ value					
0	row_0	cf1	cq1	2018-05-15 22:53:37.990	↪
↪ 0					
1	row_0	cf2	cq2	2018-05-15 22:53:38.009	↪
↪ 0					
2	row_1	cf1	cq1	2018-05-15 22:53:38.018	↪
↪ 1					
3	row_1	cf2	cq2	2018-05-15 22:53:38.026	↪
↪ 1					
4	row_2	cf1	cq1	2018-05-15 22:53:38.034	↪
↪ 2					

(continues on next page)

(continued from previous page)

5	row_2	cf2	cq2	2018-05-15 22:53:38.042	↳
↳ 2					
6	row_3	cf1	cq1	2018-05-15 22:53:38.049	↳
↳ 3					
7	row_3	cf2	cq2	2018-05-15 22:53:38.057	↳
↳ 3					
8	row_4	cf1	cq1	2018-05-15 22:53:38.065	↳
↳ 4					
9	row_4	cf2	cq2	2018-05-15 22:53:38.072	↳
↳ 4					

1.3 Example: Reading Accumulo table with catalog

This example is equivalent to the above example, except we now access the table through an existing catalog, `catalog.yml`:

```
>>> import intake
>>> c = intake.open_catalog("catalog.yml")
>>> df = c.basic.read()
>>> df
   row column_family column_qualifier column_visibility      time_
↳ value
0  row_0          cf1             cq1      2018-05-15 22:53:37.990  ↳
↳ 0
1  row_0          cf2             cq2      2018-05-15 22:53:38.009  ↳
↳ 0
2  row_1          cf1             cq1      2018-05-15 22:53:38.018  ↳
↳ 1
3  row_1          cf2             cq2      2018-05-15 22:53:38.026  ↳
↳ 1
4  row_2          cf1             cq1      2018-05-15 22:53:38.034  ↳
↳ 2
5  row_2          cf2             cq2      2018-05-15 22:53:38.042  ↳
↳ 2
6  row_3          cf1             cq1      2018-05-15 22:53:38.049  ↳
↳ 3
7  row_3          cf2             cq2      2018-05-15 22:53:38.057  ↳
↳ 3
8  row_4          cf1             cq1      2018-05-15 22:53:38.065  ↳
↳ 4
9  row_4          cf2             cq2      2018-05-15 22:53:38.072  ↳
↳ 4
```

<code>intake_accumulo.source. AccumuloSource(table)</code>	Read data from Accumulo table.
--	--------------------------------

class `intake_accumulo.source.AccumuloSource` (*table*, *host='localhost'*, *port=42424*,
username='root', *password='secret'*, *meta-*
data=None)

Read data from Accumulo table.

Parameters

table [str] The database table that will act as source
host [str] The server hostname for the given table
port [int] The server port for the given table
username [str] The username used to connect to the Accumulo cluster
password [str] The password used to connect to the Accumulo cluster

Attributes

cache_dirs
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.
plots List custom associated quick-plots

Methods

<code>close()</code>	Close open resources corresponding to this data source.
<code>discover()</code>	Open resource and populate the source attributes.
<code>read()</code>	Load entire dataset into a container and return it
<code>read_chunked()</code>	Return iterator over container fragments of data source
<code>read_partition(i)</code>	Return a part of the data corresponding to i-th partition.
<code>to_dask()</code>	Return a dask container for this data source
<code>to_spark()</code>	Provide an equivalent data object in Apache Spark
<code>yaml([with_plugin])</code>	Return YAML representation of this data-source

<code>set_cache_dir</code>	
----------------------------	--

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

A

AccumuloSource (*class in intake_accumulo.source*),
5