Package ‘pysparklyr’

October 4, 2023

**Title**  Provides a 'PySpark' Back-End for the 'sparklyr' Package

**Version**  0.1.0

**Description**  It enables 'sparklyr' to integrate with 'Spark Connect', and 'Databricks Connect' by providing a wrapper over the 'PySpark' 'python' library.

**License**  MIT + file LICENSE

**Encoding**  UTF-8

**RoxygenNote**  7.2.3

**Imports**  cli, DBI, dplyr, dbplyr, glue, purrr, reticulate (>= 1.31), methods, rlang, sparklyr (>= 1.8.3), tidyselect, fs, magrittr, tidyr, vctrs, processx, httr2

**URL**  https://github.com/mlverse/pysparklyr

**BugReports**  https://github.com/mlverse/pysparklyr/issues

**Suggests**  crayon, R6, testthat (>= 3.0.0), tibble, withr

**Config/testthat/edition**  3

**NeedsCompilation**  no

**Author**  Edgar Ruiz [aut, cre], Posit Software, PBC [cph, fnd]

**Maintainer**  Edgar Ruiz <edgar@posit.co>

**Repository**  CRAN

**Date/Publication**  2023-10-04 16:10:05 UTC

### R topics documented:

- `installed_components` .................................................. 2
- `install_pyspark` ....................................................... 2
- `spark_connect_service_start` ...................................... 3

**Index**  5
**installed_components**  
*Lists installed Python libraries*

**Description**
Lists installed Python libraries

**Usage**
```
installed_components(list_all = FALSE)
```

**Arguments**
- `list_all` Flag that indicates to display all of the installed packages or only the top two, namely, pyspark and databricks.connect

**Value**
Returns no value, only sends information to the console. The information includes the current versions of 'sparklyr', and 'pysparklyr', as well as the 'Python' environment currently loaded.

---

**install_pyspark**  
*Installs PySpark and Python dependencies*

**Description**
Installs PySpark and Python dependencies
Installs Databricks Connect and Python dependencies

**Usage**
```
install_pyspark(
    version = NULL,
    envname = NULL,
    python_version = ">=3.9",
    new_env = TRUE,
    method = c("auto", "virtualenv", "conda"),
    ...
)
```

```
install_databricks(
    version = NULL,
    cluster_id = NULL,
    envname = NULL,
    python_version = ">=3.9",
)```
spark_connect_service_start

    new_env = TRUE,
    method = c("auto", "virtualenv", "conda"),
    ...
)

Arguments

version Version of 'databricks.connect' to install
envname The name of the Python Environment to use to install the Python libraries. Default to NULL. If NULL, a name will automatically be assigned based on the version that will be installed
python_version The version of Python to use to create the Python environment.
new_env If TRUE, any existing Python virtual environment and/or Conda environment specified by envname is deleted first.
method The installation method to use. If creating a new environment, "auto" (the default) is equivalent to "virtualenv". Otherwise "auto" infers the installation method based on the type of Python environment specified by envname.
... Passed on to reticulate::py_install()
cluster_id Target of the cluster ID that will be used with. If provided, this value will be used to extract the cluster's version

Value

It returns no value to the R session. This function purpose is to create the 'Python' environment, and install the appropriate set of 'Python' libraries inside the new environment. During runtime, this function will send messages to the console describing the steps that the function is taking. For example, it will let the user know if it is getting the latest version of the Python library from 'PyPi.org', and the result of such query.

spark_connect_service_start

    Starts and stops Spark Connect locally

Description

Starts and stops Spark Connect locally

Usage

spark_connect_service_start(
    version = "3.4",
    scala_version = "2.12",
    include_args = TRUE,
    ...
)

spark_connect_service_stop(version = "3.4", ...)
Arguments

- **version**: Spark version to use (3.4 or above)
- **scala_version**: Acceptable Scala version of packages to be loaded
- **include_args**: Flag that indicates whether to add the additional arguments to the command that starts the service. At this time, only the 'packages' argument is submitted.
- ... Optional arguments; currently unused

Value

It returns messages to the console with the status of starting, and stopping the local Spark Connect service.
Index

install_databricks (install_pyspark), 2
install_pyspark, 2
installed_components, 2

reticulate::py_install(), 3

spark_connect_service_start, 3
spark_connect_service_stop
  (spark_connect_service_start), 3