

Import Files into H2O

Source: `R/import.R` (<https://github.com/h2oai/h2o-3/blob/master/R/import.R>)

Imports files into an H2O cluster. The default behavior is to pass-through to the parse phase automatically.

```
h2o.importFile(  
  path,  
  destination_frame = "",  
  parse = TRUE,  
  header = NA,  
  sep = "",  
  col.names = NULL,  
  col.types = NULL,  
  na.strings = NULL,  
  decrypt_tool = NULL,  
  skipped_columns = NULL,  
  custom_non_data_line_markers = NULL,  
  partition_by = NULL,  
  quotechar = NULL,  
  escapechar = ""  
)
```

```
h2o.importFolder(  
  path,  
  pattern = "",  
  destination_frame = "",  
  parse = TRUE,  
  header = NA,  
  sep = "",  
  col.names = NULL,  
  col.types = NULL,  
  na.strings = NULL,  
  decrypt_tool = NULL,  
  skipped_columns = NULL,  
  custom_non_data_line_markers = NULL,  
  partition_by = NULL,  
  quotechar = NULL,  
  escapechar = "\\\"  
)
```

```
h2o.importHDFS(  
  path,  
  pattern = "",  
  destination_frame = "",  
  parse = TRUE,  
  header = NA,  
  sep = "",  
  col.names = NULL,  
  na.strings = NULL  
)
```

```
h2o.uploadFile(  
  path,  
  destination_frame = "",  
  parse = TRUE,  
  header = NA,
```

```

sep = "",
col.names = NULL,
col.types = NULL,
na.strings = NULL,
progressBar = FALSE,
parse_type = NULL,
decrypt_tool = NULL,
skipped_columns = NULL,
quotechar = NULL,
escapechar = "\\\"
)

```

Arguments

- path** The complete URL or normalized file path of the file to be imported. Each row of data appears as one line of the file.
- destination_frame** (Optional) The unique hex key assigned to the imported file. If none is given, a key will automatically be generated based on the URL path.
- parse** (Optional) A logical value indicating whether the file should be parsed after import, for details see `h2o.parseRaw`.
- header** (Optional) A logical value indicating whether the first line of the file contains column headers. If left empty, the parser will try to automatically detect this.
- sep** (Optional) The field separator character. Values on each line of the file are separated by this character. If `sep = ""`, the parser will automatically detect the separator.
- col.names** (Optional) An H2OFrame object containing a single delimited line with the column names for the file.
- col.types** (Optional) A vector to specify whether columns should be forced to a certain type upon import parsing.
- na.strings** (Optional) H2O will interpret these strings as missing.
- decrypt_tool** (Optional) Specify a Decryption Tool (key-reference acquired by calling `h2o.decryptionSetup`).
- skipped_columns** a list of column indices to be skipped during parsing.
- custom_non_data_line_markers** (Optional) If a line in imported file starts with any character in given string it will NOT be imported. Empty string means all lines are imported, NULL means that default behaviour for given format will be used
- partition_by** names of the columns the persisted dataset has been partitioned by.
- quotechar** A hint for the parser which character to expect as quoting character. None (default) means autodetection.
- escapechar** (Optional) One ASCII character used to escape other characters.

- pattern** (Optional) Character string containing a regular expression to match file(s) in the folder.
- progressBar** (Optional) When FALSE, tell H2O parse call to block synchronously instead of polling. This can be faster for small datasets but loses the progress bar.
- parse_type** (Optional) Specify which parser type H2O will use. Valid types are "ARFF", "XLS", "CSV", "SVMLight"

Details

`h2o.importFile` is a parallelized reader and pulls information from the server from a location specified by the client. The path is a server-side path. This is a fast, scalable, highly optimized way to read data. H2O pulls the data from a data store and initiates the data transfer as a read operation.

Unlike the `import` function, which is a parallelized reader, `h2o.uploadFile` is a push from the client to the server. The specified path must be a client-side path. This is not scalable and is only intended for smaller data sizes. The client pushes the data from a local filesystem (for example, on your machine where R is running) to H2O. For big-data operations, you don't want the data stored on or flowing through the client.

`h2o.importFolder` imports an entire directory of files. If the given path is relative, then it will be relative to the start location of the H2O instance. The default behavior is to pass-through to the parse phase automatically.

`h2o.importHDFS` is deprecated. Instead, use `h2o.importFile`.

See also

`h2o.import_sql_select`, `h2o.import_sql_table`, `h2o.parseRaw`

Examples

```
# NOT RUN {
h2o.init(ip = "localhost", port = 54321, startH2O = TRUE)
prostate_path = system.file (https://www.rdocumentation.org/packages/base/topics/system.
prostate = h2o.importFile(path = prostate_path)
class (https://www.rdocumentation.org/packages/base/topics/class)(prostate)
summary (https://www.rdocumentation.org/packages/base/topics/summary)(prostate)

#Import files with a certain regex pattern by utilizing h2o.importFolder()
#In this example we import all .csv files in the directory prostate_folder
prostate_path = system.file (https://www.rdocumentation.org/packages/base/topics/system.
prostate_pattern = h2o.importFolder(path = prostate_path, pattern = ".*.csv")
class (https://www.rdocumentation.org/packages/base/topics/class)(prostate_pattern)
summary (https://www.rdocumentation.org/packages/base/topics/summary)(prostate_pattern)
# }
```

Developed by Erin LeDell, Navdeep Gill, Spencer Aiello,
Anqi Fu, Arno Candel, Cliff Click, Tom Kraljevic, Tomas
Nykodym, Patrick Aboyoun, Michal Kurka, Michal
Malohlava, H2O.ai.

Site built with pkgdown (<https://pkgdown.r-lib.org/>)
1.3.0.