databricks Explore sample datasets

List sample datasets available in Databricks CE

```
display(dbutils.fs.ls("/databricks-datasets"))
```

path
dbfs:/databricks-datasets/
dbfs:/databricks-datasets/README.md
dbfs:/databricks-datasets/Rdatasets/
dbfs:/databricks-datasets/SPARK_README.md
dbfs:/databricks-datasets/adult/
dbfs:/databricks-datasets/airlines/
dbfs:/databricks-datasets/amazon/
dbfs:/databricks-datasets/asa/
dbfs:/databricks-datasets/atlas_higgs/
dbfs:/databricks-datasets/bikeSharing/
dbfs:/databricks-datasets/cctvVideos/

```
selected_dataset = "sample_logs"
selected_dataset_path = "/databricks-datasets/" + selected_dataset

print(selected_dataset_path)
```

```
/databricks-datasets/sample_logs
```

List the files in a dataset

```
filelist = dbutils.fs.ls(selected_dataset_path)
display(filelist)
```

path
dbfs:/databricks-datasets/sample_logs/_SUCCESS
dbfs:/databricks-datasets/sample_logs/part-00000
dbfs:/databricks-datasets/sample_logs/part-00001

dbfs:/databricks-datasets/sample_logs/part-00002
dbfs:/databricks-datasets/sample_logs/part-00003
dbfs:/databricks-datasets/sample_logs/part-00004
dbfs:/databricks-datasets/sample_logs/part-00005
dbfs:/databricks-datasets/sample_logs/part-00006
dbfs:/databricks-datasets/sample_logs/part-00007

```
from pyspark.sql import *

filelistDF = sqlContext.createDataFrame(filelist)


just_readme = filelistDF.filter((filelistDF.name == "README.md") | (filelistDF.name ==
"readme.md"))
display(just_readme)
```

OK

```
filelistDF.createOrReplaceTempView("filelist_view")
```

```
%sql
select `path` from filelist_view where `name` like 'README%'
```

OK

```
#print(filelist)
print(filelist[0][1])

with open("/dbfs/" + selected_dataset_path + "/part-00004") as f:
  x = ''.join(f.readlines())

print(x)
```

```
_SUCCESS
2.2.2.2 - - [21/Mar/2014:10:00:00 -0400] "GET /endpoint_875 HTTP/1.1" 500 396
127.0.0.1 - - [21/Jun/2014:10:00:00 -0700] "GET /endpoint_314 HTTP/1.1" 200 220
```

```
4.4.4.4 - user1 [21/Mar/2014:10:00:00 -0400] "PUT /endpoint_996 HTTP/1.1" 200 24
3.3.3.3 - - [21/May/2014:10:00:00 -0600] "GET /endpoint_490 HTTP/1.1" 200 107
4.4.4.4 - - [21/Jul/2014:10:00:00 -0800] "PUT /endpoint_424 HTTP/1.1" 200 38
2.2.2.2 - - [21/May/2014:10:00:00 -0600] "POST /endpoint_222 HTTP/1.1" 200 464
3.3.3.3 - - [21/Mar/2014:10:00:00 -0400] "GET /endpoint_899 HTTP/1.1" 401 429
3.3.3.3 - user2 [21/Jul/2014:10:00:00 -0800] "GET /endpoint_772 HTTP/1.1" 200 461
2.2.2.2 - user1 [21/Jan/2014:10:00:00 -0200] "POST /endpoint_710 HTTP/1.1" 401 205
3.3.3.3 - user1 [21/Jun/2014:10:00:00 -0700] "POST /endpoint_906 HTTP/1.1" 200 476
1.1.1.1 - user1 [21/May/2014:10:00:00 -0600] "GET /endpoint_972 HTTP/1.1" 500 62
127.0.0.1 - user1 [21/Jun/2014:10:00:00 -0700] "PUT /endpoint_601 HTTP/1.1" 200 68
2.2.2.2 - user1 [21/Feb/2014:10:00:00 -0300] "GET /endpoint_542 HTTP/1.1" 200 372
3.3.3.3 - user1 [21/Jan/2014:10:00:00 -0200] "GET /endpoint_573 HTTP/1.1" 500 251
3.3.3.3 - - [21/Jun/2014:10:00:00 -0700] "POST /endpoint_447 HTTP/1.1" 500 417
1.1.1.1 - - [21/Jun/2014:10:00:00 -0700] "PUT /endpoint_405 HTTP/1.1" 401 231
1.1.1.1 - - [21/Jul/2014:10:00:00 -0800] "PUT /endpoint_751 HTTP/1.1" 401 150
4.4.4.4 - user1 [21/Mar/2014:10:00:00 -0400] "GET /endpoint_805 HTTP/1.1" 200 78
2.2.2.2 - - [21/May/2014:10:00:00 -0600] "PUT /endpoint_998 HTTP/1.1" 200 276
```

```
 export_json = filelistDF.toJSON()
print(export_json.take(2))
```

```
['{"path":"dbfs:/databricks-datasets/sample_logs/_SUCCESS","name":"_SUCCESS","size":0}
', '{"path":"dbfs:/databricks-datasets/sample_logs/part-00000","name":"part-00000","si
ze":245305}']
```

```
import re
from pyspark.sql import Row

apache_access_log_format = '^(\S+) (\S+) (\S+) \[([\w:/]+\s[+\-]\d{4})\] "(\S+) (\S+)
(\S+)" (\d{3}) (\d+)'

#Return a dictionary containing the parts of the Apache access log
def parse_apache_log_line(line):
  match = re.search(apache_access_log_format, line)
  if match is None:
    # Might want to just ignore, since it'a common to have dirty logs
    raise Error("Invalid log line format: %s" % line)
  return Row(
    ipAddress = match.group(1),
    clientIdentd = match.group(2),
    userId = match.group(3),
    dateTime = match.group(4),
    method = match.group(5),
    endpoint = match.group(6),
    protocol = match.group(7),
    responseCode = int(match.group(8)),
    contentSize = long(match.group(9)))
```

```
AttributeError: 'RemoteContext' object has no attribute 'textfile'
```