DataOps.onETL

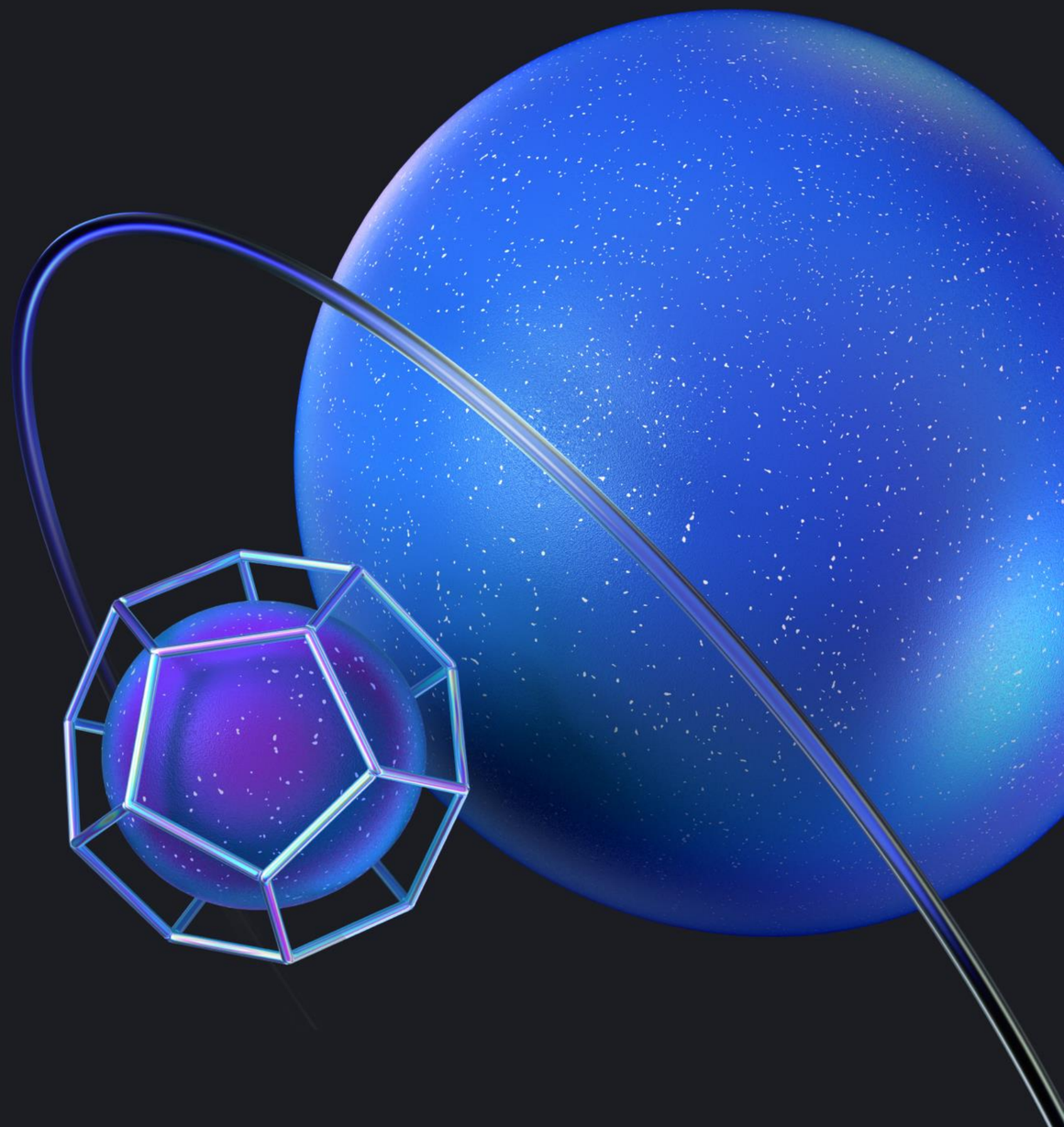
Unit#4

Объекты подключения и манипуляции данными с использованием структуры DataFrame

Саттар Гюльмамедов

PO команды ETL

МТС Тета x DataOps Platform



onETL урок # 4

→ File DataFrame Connections

→ FileDF Reader

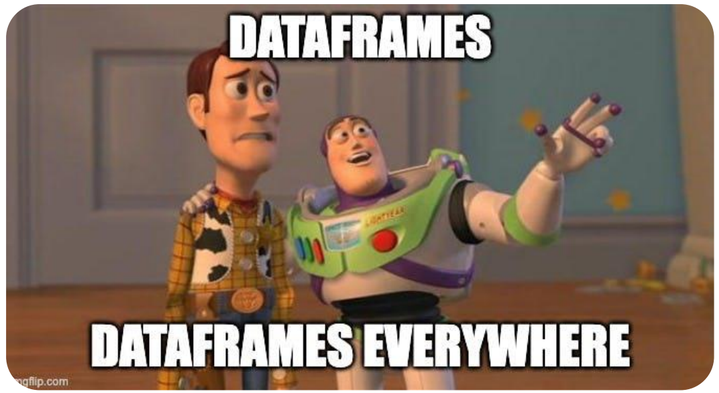
→ FileDF Writer

→ Где найти документацию

МТС Тета x DataOps Platform

DataFrame

spark.df != pandas.df



FileDataFrameConnections

→ LocalFS

→ HDFS

→ S3

	A	B	C	D	E	F	G	H	I	J	K	L
2	ABW	0,810230588	0,749301039	0,69161526	0,637959162	0,590062487	0,537295706	0,494795263	0,451969659	0,134255302	-0,045044623	-0,086392283
3	AFE	2,740404834	2,780206834	2,774990291	2,802585821	2,72815853	2,655671599	2,688371428	2,691134381	2,678183716	2,607471694	2,543756525
4	AFG	4,077627729	3,466788304	3,657576065	3,121341229	2,581549399	2,866492148	2,885207973	2,908529093	3,134746908	2,851357654	2,534498317
5	AFW	2,812850935	2,761838977	2,750730568	2,723317337	2,71305851	2,706266076	2,669238668	2,633982069	2,61564614	2,573377402	2,539799442
6	AGO	3,758702522	3,735525429	3,684428967	3,617677518	3,586211007	3,550986636	3,464456984	3,395277792	3,268348405	3,166029862	3,096752671
7	ALB	-0,16515104	-0,183211385	-0,207047	-0,291205787	-0,159880412	-0,091972294	-0,246732042	-0,426007367	-0,574206959	-0,926918062	-1,21579032
8	AND	0,630034575	0,497261876	0,355274942	0,17437769	1,10060299	1,772182713	1,580147087	1,757491257	1,761891295	1,702288225	0,994607149
9	ARB	2,157196193	2,299823498	2,259225675	2,155966313	2,109697128	2,068738846	2,096194015	2,062686901	1,757899007	1,623335187	1,78833854
10	ARE	1,041344608	0,997641826	0,956397624	0,911950036	0,863868923	0,819744473	0,789450108	0,779087177	0,817694386	0,834812788	0,808075151
11	ARG	1,136905694	1,1191092	1,099461091	1,078001344	1,057181578	1,037133897	1,015808349	0,993397493	0,970053991	0,947490959	0,925835474
12	ARM	-0,49817016	-0,448296309	-0,395592881	-0,392995249	-0,444257167	-0,486625263	-0,540251082	-0,564065543	-0,533006607	-0,522963243	-0,377102034
13	ASM	-1,146298234	-1,304782022	-1,478945711	-1,639270188	-1,807230769	-1,971818747	-2,122935674	-2,304138699	-2,42124971	-2,530170926	-1,706495481
14	ATG	1,083707561	0,934326293	0,831589248	0,786935419	0,690288329	0,610956145	0,554872966	0,534443401	0,592053943	0,597151595	0,581875833
15	AUS	1,745820001	1,72115144	1,491566489	1,439216653	1,561940498	1,646827797	1,495996711	1,477490711	1,233428251	0,140895083	1,238638797
16	AUT	0,455937472	0,589387255	0,781541633	1,120992502	1,081396299	0,694621104	0,487071925	0,444673639	0,415176732	0,435671684	0,956287797
17	AZE	1,328763827	1,293447027	1,248208997	1,191209858	1,11785721	0,981261805	0,866316916	0,846646726	0,684365325	0,441197736	0,039507866
18	BDI	3,529972848	3,551107704	3,345863383	2,18870609	1,629024754	2,287301447	2,983823553	3,26424857	2,867082195	2,672478462	2,66016084
19	BEL	0,620163579	0,471340144	0,443929288	0,579446242	0,506300002	0,385228018	0,455184695	0,540461327	0,430996815	0,411601997	0,856132433
20	BEN	2,915012122	2,899920609	2,92622898	2,951249123	2,949828144	2,946321561	2,922392187	2,887074085	2,829138113	2,759705179	2,702041089
21	BFA	3,031880077	3,008421955	2,979778499	2,972345608	2,934811289	2,865655421	2,768681163	2,703875567	2,688787667	2,650375915	2,559988391
22	BGD	1,243571246	1,267157297	1,245960208	1,191061127	1,230795354	1,249724066	1,161378463	1,11317249	1,14420974	1,149318476	1,074836743
23	BGR	-0,579220596	-0,559647217	-0,568389264	-0,638069468	-0,701382098	-0,730443175	-0,722080405	-0,703905641	-0,600241518	-0,814846472	-6,187252982
24	BHR	1,055562986	2,954757119	3,845379359	3,816604913	3,429078922	3,291633803	2,072370164	0,459362585	-1,125242678	-0,96602484	0,611005553
25	BHS	1,081836998	0,934994066	0,898582547	0,912227428	0,831528122	0,765793759	0,720668948	0,657441085	0,471994445	0,352416997	0,508137884
26	BIH	-1,854258767	-1,558328902	-1,293477828	-1,317606589	-1,237305784	-1,18362668	-1,166594811	-1,166081335	-1,266771023	-1,440652596	-1,150513985
27	BLR	-0,1566176	-0,03838	0,056151568	0,132853237	0,087721088	-0,109782323	-0,213824207	-0,201786601	-0,42347525	-0,828232721	-0,804228427
28	BLZ	2,256631567	2,238167082	2,194276752	2,116320634	2,046871171	1,98926784	1,948634454	1,823016119	1,486221516	1,285629921	1,301640251
29	BMU	0,361775877	0,312791571	0,210544188	0,151869636	-1,05247098	-1,060534596	0,070427498	-0,010952131	-0,028168133	-0,202104145	-0,364505179

```
[20]: df.show(31, truncate=False)
+------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
|country_code|y2012              |y2013              |y2014              |y2015              |y2016              |y2017              |y2018              |y2019              |y2020              |y2021              |y2022              |
+------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
|ABW         |0.810230587788097  |0.749301039347625  |0.691615260101675  |0.63795916173479   |0.590062487333077  |0.537295706145068  |0.49479526304698 9 |0.451969658863314  |0.134255302359952  |-0.045044623090632 9|-0.086392282654992 7|
|AFE         |2.740404834239655  |2.780206833556065  |2.774990291343476  |2.80258582087383   |2.728158529987581 5|2.655671599230146 3|2.688371428074276 5|2.691134380563980 2|2.678183716419013 6|2.607471694095579 6|2.543756524554581 2|
|AFG         |4.077627728587795  |3.466788304109782  |3.657576065630364  |3.121341228908737  |2.581549398956592  |2.866492147500737 3|2.885207973048492  |2.908529092687743  |3.134746907791634  |2.851357654499312  |2.534498316667334  |
|AFW         |2.812850934621252 5|2.761838976751036 5|2.750730567984831 5|2.723317337417171 3|2.713058510369876 3|2.706266075863510 4|2.669238668027347  |2.633982069113443 5|2.615646139857915 4|2.573377401687949 4|2.539799442479179  |
|AGO         |3.758702529194126  |3.735525428575565  |3.684428966834926  |3.617677517620956  |3.586211000683242  |3.550986636619956  |3.464456983680985  |3.395277791970976  |3.268348404584563  |3.166029861530256  |3.096752670673266  |
|ALB         |-0.165151040121679 |-0.183211384606402 |-0.207046999760594 |-0.291205786840436 |-0.159880412127734 |-0.091972293744249 5|-0.246732042281782 |-0.426007367032238 |-0.574206959244013 |-0.926918061639398 |-1.215790320125329 |
|AND         |0.630034575245577  |0.497261875887559  |0.355274942185653  |0.174377690367456  |1.100602989798862  |1.772182712878631  |1.580147086561455  |1.757491257369977  |1.761891295907448  |1.702288225345615  |0.994607149162366  |
|ARB         |2.157196193328985 6|2.299823498884862  |2.259225674977135  |2.155966313224255  |2.109697128451259  |2.068738846072413  |2.096194015288574 5|2.062686901070722  |1.757899006959186  |1.623335187238524  |1.788385403946193  |
|ARE         |1.041344607830885  |0.997641825826328  |0.956397623844095  |0.911950036242349  |0.863868922846647  |0.819744473334198  |0.789450107726376  |0.779087177439076  |0.817694385518168  |0.834812788038646  |0.808075150558113  |
|ARG         |1.136905693378967  |1.11910919999772   |1.09946109101836   |1.07800134449604   |1.057181577762752  |1.037133896865145  |1.015808349080002  |0.993397493293046  |0.970053991438048  |0.947490959333989  |0.925835473662058  |
|ARM         |-0.498170160314836 |-0.448296309047071 |-0.395592881362063 |-0.392995248824912 |-0.444257166904663 |-0.486625263030829 |-0.540251081979475 |-0.564065542556681 |-0.533006606500067 |-0.522963242641951 |-0.377102034390101 |
|ASM         |-1.146298234333733 |-1.304782022416283 |-1.478945711166667 |-1.639270187580843 |-1.807230768904222 |-1.971818747440715 |-2.122935673924733 |-2.304138698905233 |-2.421249709990733 |-2.530170926471207 |-1.706495480618955 |
|ATG         |1.083707560137     |0.934326293295174  |0.831589247616481  |0.786935419380774  |0.690288328652049  |0.610956144890505  |0.554872969562746  |0.534443400511753  |0.592053942582792  |0.597151594527623  |0.581875832853236  |
|AUS         |1.745820000067359  |1.721151439989     |1.491566489128471  |1.439216652599525  |1.561940498103999  |1.646827797350817  |1.495996710413285  |1.477490710584383  |1.233428251331378  |1.140895083292287  |1.238638796575531  |
|AUT         |0.455937472231891  |0.589387254692989  |0.781541632525841  |1.120992502369581  |1.081396298728002  |0.694621103604983  |0.487071924768999  |0.444673639004876  |0.415176731565556  |0.435671684056384  |0.956287797076634  |
|AZE         |1.328763827333755  |1.293447027005911  |1.248208997735588  |1.191209858078018  |1.117857210139111  |0.981261804504575  |0.866316916378041  |0.846646726414021  |0.684365325388487  |0.441197735816311  |0.039507866238038  |
|BDI         |3.529972847776032  |3.551107704043353  |3.345863383396     |2.188706090456616  |1.629024753883778  |2.287301446895888  |2.983823553303175  |3.264248570258887  |2.867082195327444  |2.672478461796476  |2.660160840004174  |
|BEL         |0.620163579337378  |0.471340143966627  |0.443929288475221  |0.579446241677791  |0.506300002451376  |0.385228018372369  |0.455184695275022  |0.540461327098122  |0.430996815099722  |0.411601996532257  |0.856132433382837  |
|BEN         |2.915012122173111  |2.899920609174312  |2.926228979846691  |2.951249122668277  |2.949828143925949  |2.946321561188807  |2.922392186810889  |2.887074084933347  |2.829138112822106  |2.759705179333777  |2.702041089333956  |
|BFA         |3.031880077224777  |3.008421955462805  |2.979778499233872  |2.972345608216331  |2.934811289124846  |2.865655421084027  |2.768681162588888  |2.703875567252288  |2.688787667306686  |2.650375915122147  |2.559988391274344  |
|BGD         |1.243571246468827  |1.267157297217878  |1.245960208087779  |1.191061127233824  |1.230795354270544  |1.249724066577957  |1.161378462704926  |1.113172490016756  |1.144209749001656  |1.149318476331586  |1.074836743115885  |
|BGR         |-0.579220596364389 |-0.559647217418859 |-0.568389264852903 |-0.638069468160719 |-0.701382097841331 |-0.730443175298289 |-0.722080405222853 |-0.703905641117455 |-0.600241518465542 |-0.814846471528947 |-6.187252982049585 |
|BHR         |1.055562985562381  |2.954757111852975  |3.845379353863732  |3.816604912611651  |3.429078921505881  |3.291633803232261  |2.072370164333842  |0.459362584678761  |-1.125242678338333 |-0.966024840061971 |0.611005553402689  |
|BHS         |1.081836998383634  |0.934994065798907  |0.898582547265484  |0.912227428271637  |0.831528122187361  |0.765793758966215  |0.720668948364001  |0.657441084996389  |0.471994444906908  |0.352416997336031  |0.508137884368941  |
|BIH         |-1.854258766676769 |-1.558328902265336 |-1.293477828044997 |-1.317606589222995 |-1.237305783537664 |-1.183626679669333 |-1.166594811287262 |-1.166081334650816 |-1.266771023115278 |-1.440652596041193 |-1.150513984594433 |
|BLR         |-0.156617600248906 |-0.038380000266784 |0.056151567510169  |0.132853236725342  |0.087721087870644 9|-0.109782322950581 |-0.213824206997671 |-0.201786600970723 |-0.423475250279 49 |-0.828232720866807 |-0.804228426880244 |
|BLZ         |2.256631566774737  |2.238167082342999  |2.194276752381771  |2.116320634018391  |2.046871171057766  |1.989267840251 2   |1.948634453864 43  |1.823016115906 52  |1.486221515756 32  |1.285629920655 72  |1.301640251412 76  |
|BMU         |0.361775877335588  |0.312791570829084  |0.210544188304464  |0.151869636092964  |-1.052470979682 71 |-1.060534596076 2  |0.070427497805 0419|-0.010952131373 9413|-0.028168132639 0652|-0.202104144558 527|-0.364505179849 171|
|BOL         |1.654635398874 86  |1.629573199155 05  |1.603315106722 19  |1.573144940148 3   |1.547288296347 13  |1.520108881053 33  |1.487474196724 42  |1.457504391417 79  |1.339739220431 11  |1.193486718678 83  |1.190274813262 8   |
|BRA         |0.900343360601223  |0.868346150167205  |0.857834828724811  |0.845992601345206  |0.811256478 2781   |0.792263400422 27  |0.793768153848091  |0.766107802084815  |0.665176682246802  |0.528590452919416  |0.459583548488331  |
|BRB         |0.257756861693629  |0.241564354388319  |0.226568487744075  |0.212392263810218  |0.203329499651748  |0.192888292188534  |0.179288797746607  |0.175755759089405  |0.182929163306018  |0.180461453548187  |0.154574639677083  |
+------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+-------------------+
only showing top 31 rows
```

МТС Тета x DataOps Platform

LocalFS

```python
from onetl.connection import SparkLocalFS
from pyspark.sql import SparkSession

spark = SparkSession.builder \
    .master("local") \
    .appName("spark-app-name") \
    .getOrCreate()

local_fs = SparkLocalFS(spark=spark)
local_fs.check()
```

https://onetl.readthedocs.io/en/stable/connection/file_df_connection/spark_local_fs.html

МТС Тета x DataOps Platform

HDFS

```python
from onetl.connection import Hive
from pyspark.sql import SparkSession

spark = (
    SparkSession.builder.appName("spark-app-name")
    .option(
        "spark.kerberos.access.hadoopFileSystems",
        "hdfs://namenode1.domain.com:8020",
    )
    .option("spark.kerberos.principal", "user")
    .option("spark.kerberos.keytab", "/path/to/keytab")
    .enableHiveSupport()
    .getOrCreate()
)
hdfs = SparkHDFS(
    host="namenode1.domain.com",
    cluster="cluster_name",
    spark=spark,
)
hdfs.check()
```

https://onetl.readthedocs.io/en/stable/connection/file_df_connection/spark_hdfs/index.html

S3

```python
from onetl.connection import SparkS3
from pyspark.sql import SparkSession

maven_packages = SparkS3.get_packages(spark_version="3.5.0")
excluded_packages = [
    "com.google.cloud.bigdataoss:gcs-connector",
    "org.apache.hadoop:hadoop-aliyun",
    "org.apache.hadoop:hadoop-azure-datalake",
    "org.apache.hadoop:hadoop-azure",
]
spark = (
    SparkSession.builder.appName("spark-app-name")
    .config("spark.jars.packages", ",".join(maven_packages))
    .config("spark.jars.excludes", ",".join(excluded_packages))
    .config("spark.hadoop.fs.s3a.committer.magic.enabled", "true")
    .config("spark.hadoop.fs.s3a.committer.name", "magic")
    .config(
        "spark.hadoop.mapreduce.outputcommitter.factory.scheme.s3a",
        "org.apache.hadoop.fs.s3a.commit.S3ACommitterFactory",
    )
    .config(
        "spark.sql.parquet.output.committer.class",
        "org.apache.spark.internal.io.cloud.BindingParquetOutputCommitter",
    )
    .config(
        "spark.sql.sources.commitProtocolClass",
        "org.apache.spark.internal.io.cloud.PathOutputCommitProtocol",
    )
    .getOrCreate()
)

s3 = SparkS3(
    host="domain.com",
    protocol="http",
    bucket="my-bucket",
    access_key="ACCESS_KEY",
    secret_key="SECRET_KEY",
    extra={
        "path.style.access": True,
    },
    spark=spark,
)
s3.check()
s3.close()
```

МТС Тета x DataOps Platform

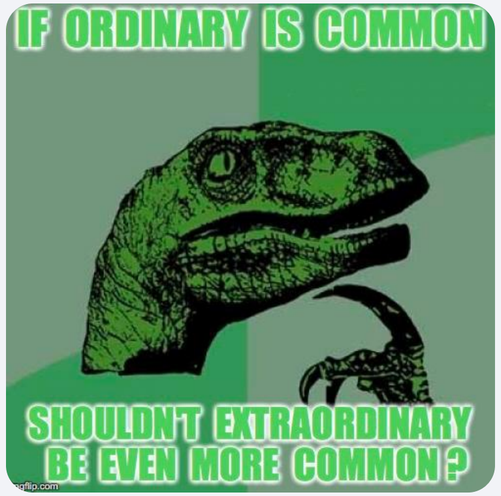
S3()

Обычные:

- → host
- → [port]
- → protocol (https | http)
- → spark

Специфические

- → [access_key]
- → [secret_key]
- → [session_token]
- → [region]
- → bucket



https://onetl.readthedocs.io/en/stable/connection/file_df_connection/spark_s3/index.html

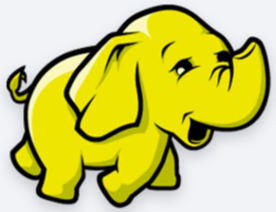
S3() - extra

```python
extra = {
    "path.style.access": True,
    "committer.magic.enabled": True,
    "committer.name": "magic",
    "connection.timeout": 300000,
}
```

https://hadoop.apache.org/docs/current/hadoop-aws/tools/hadoop-aws/index.html#General_S3A_Client_configuration

S3 — Troubleshooting

```python
spark.sparkContext.setLogLevel("debug")
```



https://onetl.readthedocs.io/en/stable/connection/file_df_connection/spark_s3/troubleshooting.html

Files Formats

- → CSV
- → Excel
- → JSON
- → JSONLine
- → XML
- → Avro
- → ORC
- → Parquet

CSV

Параметры:

- → sep
- → encoding
- → quote
- → escape
- → header
- → lineSep

Методы:

- → -

https://onetl.readthedocs.io/en/stable/file_df/file_formats/csv.html

https://spark.apache.org/docs/latest/sql-data-sources-csv.html

Excel

Параметры:

→ header

→ inferSchema

Методы:

→ get_packages

https://onetl.readthedocs.io/en/stable/file_df/file_formats/excel.html

https://github.com/crealytics/spark-excel

МТС Тета x DataOps Platform

JSON

Параметры:

→ multiLine

→ encoding

→ lineSep

Методы:

→ -

https://onetl.readthedocs.io/en/stable/file_df/file_formats/json.html

https://spark.apache.org/docs/latest/sql-data-sources-json.html

МТС Тета x DataOps Platform

JSONLine

Параметры:

→ multiLine

→ encoding

→ lineSep

Методы:

→ -

https://onetl.readthedocs.io/en/stable/file_df/file_formats/jsonline.html

https://spark.apache.org/docs/latest/sql-data-sources-json.html

XML

Параметры:

→ rowTag

Методы:

→ get_packages

https://onetl.readthedocs.io/en/stable/file_df/file_formats/xml.html

https://github.com/databricks/spark-xml

МТС Тета x DataOps Platform

Avro

Параметры:

→ avroSchema

→ avroSchemaUrl

Методы:

→ get_packages

https://onetl.readthedocs.io/en/stable/file_df/file_formats/avro.html

https://spark.apache.org/docs/latest/sql-data-sources-avro.html

ORC

Параметры:

→ -

Методы:

→ -

https://onetl.readthedocs.io/en/stable/file_df/file_formats/orc.html

https://spark.apache.org/docs/latest/sql-data-sources-orc.html

МТС Тета x DataOps Platform

Parquet

Параметры:

→ -

Методы:

→ -

https://onetl.readthedocs.io/en/stable/file_df/file_formats/parquet.html

https://spark.apache.org/docs/latest/sql-data-sources-parquet.html

FileDF Reader

→ FileDFReader()

→ run()

FileDFReader()

- → connection
- → source_path
- → format

df_schema

Python

```python
from pyspark.sql.types import (
    DoubleType,
    IntegerType,
    StringType,
    StructField,
    StructType,
    TimestampType,
)

df_schema = StructType(
    [
        StructField("_id", IntegerType()),
        StructField("text_string", StringType()),
        StructField("hwm_int", IntegerType()),
        StructField("hwm_datetime",
TimestampType()),
        StructField("float_value", DoubleType()),
    ],
)
```

options

https://onetl.readthedocs.io/en/stable/file_df/file_df_reader/options.html

https://spark.apache.org/docs/latest/sql-data-sources-generic-options.html

run()

Python

```python
from onetl.connection import SparkLocalFS
from onetl.file import FileDFReader
from onetl.file.format import CSV

csv = CSV(delimiter=",")
local_fs = SparkLocalFS(spark=spark)

reader = FileDFReader(
    connection=local_fs,
    format=csv,
    source_path="/path",
)
df = reader.run()
```

FileDF Writer

→ FileDFWriter()

→ run(df)

FileDFWriter()

- → connection
- → target_path
- → format

options

https://onetl.readthedocs.io/en/stable/file_df/file_df_writer/options.html

https://spark.apache.org/docs/latest/sql-data-sources-load-save-functions.html

run(df)

Python

```python
from onetl.connection import SparkLocalFS
from onetl.file import FileDFWriter
from onetl.file.format import CSV

csv = CSV(delimiter=",")
local_fs = SparkLocalFS(spark=spark)

writer = FileDFWriter(
    connection=local_fs,
    format=csv,
    target_path="/path/to/directory",
    options=FileDFWriter.Options(if_exists="replace_entire_directory"),
)

writer.run(df)
```

onETL урок # 4

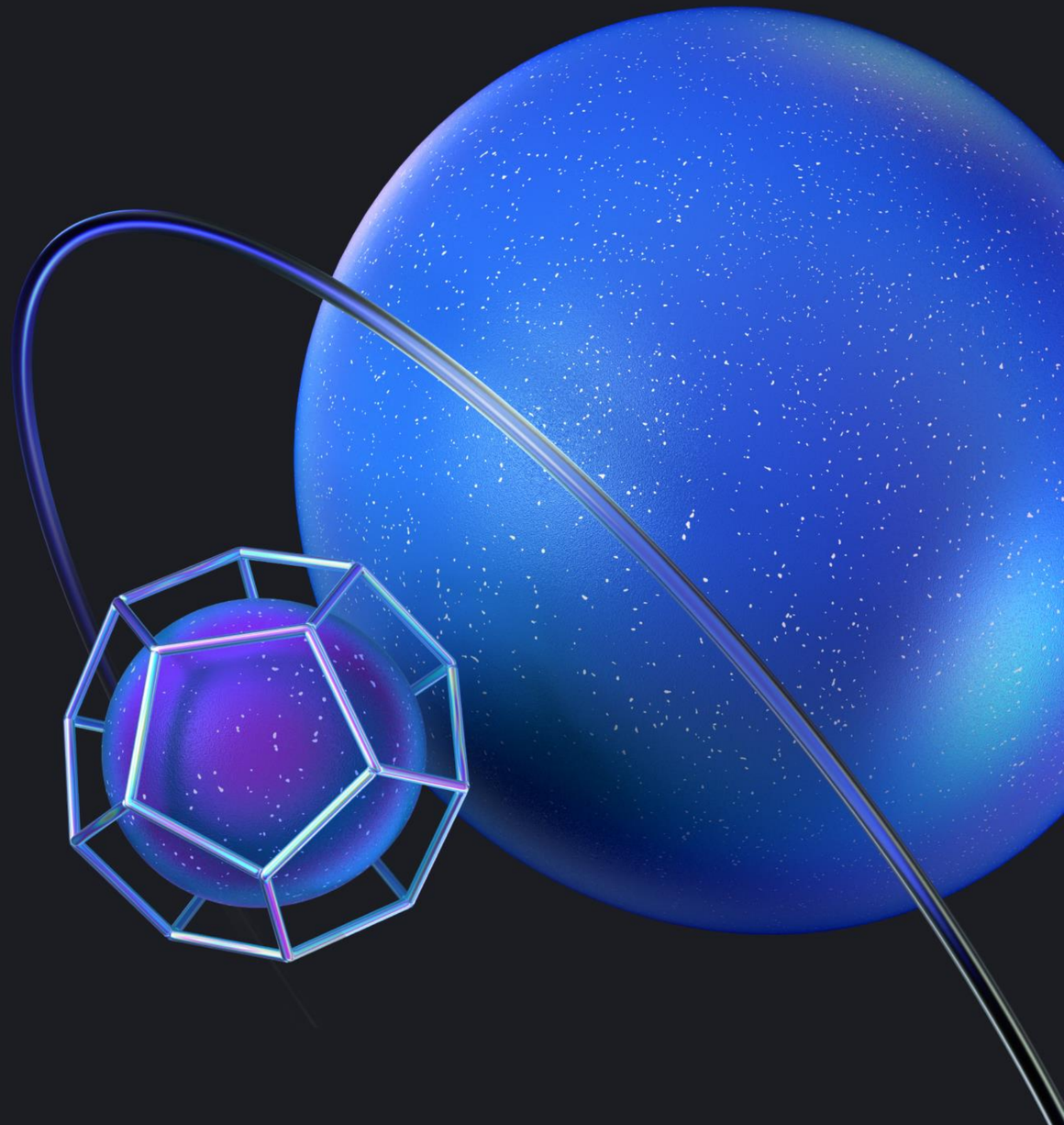
→ File DataFrame Connections

→ FileDF Reader

→ FileDF Writer

→ Где найти документацию

Спасибо!



onetools@mts.ru
https://t.me/c/1511728757/5

МТС Тета x DataOps Platform