

```
ssh -i ~/.ssh/id_rsa_student898_2 student898_2@37.139.41.176
hdfs dfs -ls
hdfs dfs -mkdir input_csv_for_stream
hdfs dfs -ls
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
igoreigor-MS-7808:~$ ssh -i ~/.ssh/id_rsa_student898_2 student898_2@37.139.41.176
Last login: Thu Jan  6 14:58:21 2022 from 109-252-20-121.nat.spd-mgts.ru
[student898_2@bigdataanalytics-worker-3 ~]$ hdfs dfs -ls
Found 4 items
drwx----- - student898_2 student898_2          0 2022-01-05 18:00 .Trash
drwxr-xr-x - student898_2 student898_2          0 2022-01-05 10:18 .sparkStaging
drwxr-xr-x - student898_2 student898_2          0 2021-12-15 22:13 for_stream
drwxr-xr-x - student898_2 student898_2          0 2022-01-04 14:47 shadrin_iris_console_checkpoint
[student898_2@bigdataanalytics-worker-3 ~]$ hdfs dfs -mkdir input_csv_for_stream
[student898_2@bigdataanalytics-worker-3 ~]$ hdfs dfs -ls
Found 5 items
drwx----- - student898_2 student898_2          0 2022-01-05 18:00 .Trash
drwxr-xr-x - student898_2 student898_2          0 2022-01-05 10:18 .sparkStaging
drwxr-xr-x - student898_2 student898_2          0 2021-12-15 22:13 for_stream
drwxr-xr-x - student898_2 student898_2          0 2022-01-10 18:08 input_csv_for_stream
drwxr-xr-x - student898_2 student898_2          0 2022-01-04 14:47 shadrin_iris_console_checkpoint
[student898_2@bigdataanalytics-worker-3 ~]$
```

```
ls for_stream/
cat for_stream/archive.csv
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
drwxr-xr-x - student898_2 student898_2          0 2022-01-04 14:47 shadrin_iris_console_checkpoint
[student898_2@bigdataanalytics-worker-3 ~]$ ls for_stream/
archive.csv  data.json  dataset.csv  iris.json  product_list1.csv  product_list2.csv  product_list3.csv  product_list4.csv  product_list.csv
[student898_2@bigdataanalytics-worker-3 ~]$ cat for_stream/archive.csv
Year,Honor,Name,Country,Birth Year,Death Year,Title,Category,Context
1927,Man of the Year,Charles Lindbergh,United States,1902,1974,US Air Mail Pilot,,First Solo Transatlantic Flight
1928,Man of the Year,Walter Chrysler,United States,1875,1940,Founder of Chrysler,Economics,Chrysler/Dodge Merger
1929,Man of the Year,Owen D. Young,United States,1874,1962,Member of the German Reparations International Commission,Diplomacy,Young Plan
1930,Man of the Year,Mahatma Gandhi,India,1869,1948,,Revolution,Salt March
1931,Man of the Year,Pierre Laval,France,1883,1945,Prime Minister of France,Politics,
1932,Man of the Year,Franklin D. Roosevelt,United States,1882,1945,President of the United States,Politics,Presidential Election
1933,Man of the Year,Hugh S. Johnson,United States,1882,1942,Director of the National Recovery Administration,Politics,New Deal
1934,Man of the Year,Franklin D. Roosevelt,United States,1882,1945,President of the United States,Politics,
1935,Man of the Year,Haile Selassie,Ethiopia,1892,1975,Emperor of Ethiopia,War,Colonial War
1936,Woman of the Year,Wallis Simpson,United States,1896,1986,Duchess of Windsor,Politics,Edward VIII Abdication Crisis
1937,Man and Wife of the Year,Chiang Kai-shek,China,1887,1975,Premier of the Republic of China,War,World War II
1937,Man and Wife of the Year,Soong Mei-ling,China,1898,2003,First Lady of the Republic of China,War,World War II
1938,Man of the Year,Adolf Hitler,Germany,1889,1945,Chancellor of Germany,War,World War II
```

Запускаем `pyspark`

Инициализация стрима

В командной строке `pyspark` импортируем нужные методы и определяем функцию `console_output` для вывода стрима в консоль.

```
from pyspark.sql import functions as F
from pyspark.sql.types import StructType, StringType
def console_output(df, freq):
    return df.writeStream \
        .format("console") \
        .trigger(processingTime='%s seconds' % freq) \
        .options(truncate=False) \
        .start()
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

version 2.3.2.3.1.4.0-315

Using Python version 2.7.5 (default, Nov 16 2020 22:23:17)
SparkSession available as 'spark'.
>>> from pyspark.sql import functions as F
>>> from pyspark.sql.types import StructType, StringType
>>> def console_output(df, freq):
...     return df.writeStream \
...         .format("console") \
...         .trigger(processingTime='%s seconds' % freq) \
...         .options(truncate=False) \
...         .start()
...
>>>
```

Определяем схему наших файлов

```
schema = StructType().add("Year", StringType()).add("Honor", StringType()).add("Name", StringType()).add("Country", StringType()).add("Birth Year", StringType()).add("Death Year", StringType())
```

Создаём стрим чтения из файла (с параметром `.format("csv")`). В `options` указываем папку на HDFS, из которой будут читаться файлы`

```
raw_files = spark \
    .readStream \
    .format("csv") \
    .schema(schema) \
    .options(path="input_csv_for_stream", header=True) \
    .load()
```

Запускаем стрим

```
out = console_output(raw_files, 5)
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

>>> from pyspark.sql.types import StructType, StringType
>>> def console_output(df, freq):
...     return df.writeStream \
...         .format("console") \
...         .trigger(processingTime='%s seconds' % freq) \
...         .options(truncate=False) \
...         .start()
...
>>> schema = StructType().add("Year", StringType()).add("Honor", StringType()).add("Name", StringType()).add("Country", StringType()).add("Birth Year", StringType()).add("Death Year", StringType())
>>> raw_files = spark \
...     .readStream \
...     .format("csv") \
...     .schema(schema) \
...     .options(path="input_csv_for_stream", header=True) \
...     .load()
>>> out = console_output(raw_files, 5)
>>>
```

В соседнем терминале подключаемся к удалённому серверу `worker-3` и переходим в каталог с загруженными файлами`

```
ls
ll for_stream
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
student898_2@bigdataanalytics-worker-3:~
igor@igor-MS-7808:~$ ssh -i ~/.ssh/id_rsa_student898_2 student898_2@37.139.41.176
Last login: Mon Jan 10 18:07:02 2022 from 109.252.20.121
[student898_2@bigdataanalytics-worker-3 ~]$ ls
for_stream
[student898_2@bigdataanalytics-worker-3 ~]$ ll for_stream
итого 120
-rw-rw-r-- 1 student898_2 student898_2 11686 янв  3 18:18 archive.csv
-rw-rw-r-- 1 student898_2 student898_2 25867 янв  3 19:25 data.json
-rwxr-xr-x 1 student898_2 student898_2 43320 дек 29 20:29 dataset.csv
-rw-rw-r-- 1 student898_2 student898_2 15802 янв  1 10:37 iris.json
-rw-rw-r-- 1 student898_2 student898_2   98 дек 16 18:20 product_list1.csv
-rw-rw-r-- 1 student898_2 student898_2  126 дек 16 19:21 product_list2.csv
-rw-rw-r-- 1 student898_2 student898_2  128 дек 16 19:17 product_list3.csv
-rw-rw-r-- 1 student898_2 student898_2  128 дек 16 19:19 product_list4.csv
-rw-rw-r-- 1 student898_2 student898_2  125 дек 16 18:10 product_list.csv
[student898_2@bigdataanalytics-worker-3 ~]$
```

Копируем файл на HDFS

```
hdfs dfs -put for_stream/archive.csv input_csv_for_stream
```

Завершаем стрим командой `out.stop()`

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
student898_2@bigdataanalytics-worker-3:~
|1935|Man of the Year      |Haile Selassie   |Ethiopia   |1892   |1975   |
|1936|Woman of the Year    |Wallis Simpson  |United States|1896   |1986   |
|1937|Man and Wife of the Year|Chiang Kai-shek |China      |1887   |1975   |
|1937|Man and Wife of the Year|Soong Mei-ling  |China      |1898   |2003   |
|1938|Man of the Year        |Adolf Hitler    |Germany    |1889   |1945   |
|1939|Man of the Year        |Joseph Stalin   |Soviet Union|1878   |1953   |
|1940|Man of the Year        |Winston Churchill|United Kingdom|1874   |1965   |
|1941|Man of the Year        |Franklin D. Roosevelt|United States|1882   |1945   |
|1942|Man of the Year        |Joseph Stalin   |Soviet Union|1878   |1953   |
|1943|Man of the Year        |George Marshall |United States|1880   |1959   |
|1944|Man of the Year        |Dwight D. Eisenhower|United States|1890   |1969   |
|1945|Man of the Year        |Harry S. Truman |United States|1884   |1972   |
+-----+-----+-----+-----+-----+
only showing top 20 rows

out.stop()
>>> out.stop()
>>>
```

Попробуем запустить стрим с другими опциями.

Параметр `maxFilesPerTrigger` определяет сколько файлов будет прочитано в одном батче. При этом, если необработанных файлов меньше чем `maxFilesPerTrigger`, то они не будут прочитаны и батч не появится

```
raw_files = spark \
    .readStream \
    .format("csv") \
    .schema(schema) \
    .options(path="input_csv_for_stream",
             header=True,
             maxFilesPerTrigger=1) \
    .load()
```

Так же добавим свою колонку `Live Year`

```
extra_files = raw_files \
    .withColumn("Live Year", F.col("Death Year") - F.col("Birth Year"))
```

Запускаем стрим

```
out = console_output(extra_files, 5)
```

Year	Honor	Name	Country	Birth Year	Death Year	Live Year
1927	Man of the Year	Charles Lindbergh	United States	1902	1974	72.0
1928	Man of the Year	Walter Chrysler	United States	1875	1940	65.0
1929	Man of the Year	Owen D. Young	United States	1874	1962	88.0
1930	Man of the Year	Mahatma Gandhi	India	1869	1948	79.0
1931	Man of the Year	Pierre Laval	France	1883	1945	62.0
1932	Man of the Year	Franklin D. Roosevelt	United States	1882	1945	63.0
1933	Man of the Year	Hugh S. Johnson	United States	1882	1942	60.0
1934	Man of the Year	Franklin D. Roosevelt	United States	1882	1945	63.0
1935	Man of the Year	Haile Selassie	Ethiopia	1892	1975	83.0
1936	Woman of the Year	Wallis Simpson	United States	1896	1986	90.0
1937	Man and Wife of the Year	Chiang Kai-shek	China	1887	1975	88.0
1937	Man and Wife of the Year	Soong Mei-ling	China	1898	2003	105.0
1938	Man of the Year	Adolf Hitler	Germany	1889	1945	56.0
1939	Man of the Year	Joseph Stalin	Soviet Union	1878	1953	75.0

Закрываем стрим и выходим из консоли `r pyspark`

`out.stop()`

`exit()`

Удаляем файлы из HDFS. Локально пока оставим, может пригодится

`hdfs dfs -rm -r input_csv_for_stream`

Year	Honor	Name	Country	Birth Year	Death Year	Live Year
1939	Man of the Year	Joseph Stalin	Soviet Union	1878	1953	75.0
1940	Man of the Year	Winston Churchill	United Kingdom	1874	1965	91.0
1941	Man of the Year	Franklin D. Roosevelt	United States	1882	1945	63.0
1942	Man of the Year	Joseph Stalin	Soviet Union	1878	1953	75.0
1943	Man of the Year	George Marshall	United States	1880	1959	79.0
1944	Man of the Year	Dwight D. Eisenhower	United States	1890	1969	79.0
1945	Man of the Year	Harry S. Truman	United States	1884	1972	88.0

only showing top 20 rows

```

out.stop()
>>> exit
Use exit() or Ctrl-D (i.e. EOF) to exit
>>> exit()
[student898_2@bigdataanalytics-worker-3 ~]$ hdfs dfs -rm -r input_csv_for_stream
22/01/10 18:26:35 INFO fs.TrashPolicyDefault: Moved: 'hdfs://bigdataanalytics-head-0.mcs.local:8020/user/student898_2/input_csv_for_stream' to trash at: hdfs://bigdataanalytics-head-0.mcs.local:8020/user/student898_2/.Trash/Current/user/student898_2/input_csv_for_stream
[student898_2@bigdataanalytics-worker-3 ~]$

```

Задание 2. Создать свой топик/топики, загрузить туда через консоль осмысленные данные с kaggle. Лучше в формате json. Много сообщений не нужно, достаточно штук 10-100. Прочитать свой топик так же, как на уроке.

`/usr/hdp/current/kafka-broker/bin/kafka-topics.sh --list --zookeeper bigdataanalytics-worker-3:2181`

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
student898_2@bigdataanalytics-worker-3:~
>>> exit
Use exit() or Ctrl-D (i.e. EOF) to exit
>>> exit()
[student898_2@bigdataanalytics-worker-3 ~]$ hdfs dfs -rm -r input_csv_for_stream
22/01/10 18:26:35 INFO fs.TrashPolicyDefault: Moved: 'hdfs://bigdataanalytics-head-0.mcs.local:8020/user/student898_2/input_csv_for_stream' to trash at: hdfs://bigdataanalytics-head-0.mcs.local:8020/user/student898_2/.Trash/Current/user/student898_2/input_csv_for_stream
[student898_2@bigdataanalytics-worker-3 ~]$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh --list --zookeeper bigdataanalytics-worker-3:2181
898_1
__consumer_offsets
alex_test
cherneev_test
incident_event_json
orders_json
sales_unknown
shadrin_iris
shadrin_iris_test
test
[student898_2@bigdataanalytics-worker-3 ~]$
```

Удалю топики shadrin_iris & shadrin_iris_test
/usr/hdp/current/kafka-broker/bin/kafka-topics.sh --delete --topic shadrin_iris --zookeeper bigdataanalytics-worker-3:2181
/usr/hdp/current/kafka-broker/bin/kafka-topics.sh --delete --topic shadrin_iris_test --zookeeper bigdataanalytics-worker-3:2181
Создаю топику shadrin_data
/usr/hdp/current/kafka-broker/bin/kafka-topics.sh --create --topic shadrin_data --zookeeper bigdataanalytics-worker-3:2181 --partitions 1 --replication-factor 1
В одном терминале запустим `console-consumer` чтобы прочесть запись
/usr/hdp/current/kafka-broker/bin/kafka-console-consumer.sh --topic shadrin_data --bootstrap-server bigdataanalytics-worker-3:6667

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
student898_2@bigdataanalytics-worker-3:~
shadrin_iris_test
test
[student898_2@bigdataanalytics-worker-3 ~]$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh --delete --topic shadrin_iris --zookeeper bigdataanalytics-worker-3:2181
Topic shadrin_iris is marked for deletion.
Note: This will have no impact if delete.topic.enable is not set to true.
[student898_2@bigdataanalytics-worker-3 ~]$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh --delete --topic shadrin_iris_test --zookeeper bigdataanalytics-worker-3:2181
Topic shadrin_iris_test is marked for deletion.
Note: This will have no impact if delete.topic.enable is not set to true.
[student898_2@bigdataanalytics-worker-3 ~]$ /usr/hdp/current/kafka-broker/bin/kafka-topics.sh --create --topic shadrin_data --zookeeper bigdataanalytics-worker-3:2181 --partitions 1 --replication-factor 1
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issues it is best to use either, but not both.
Created topic "shadrin_data".
[student898_2@bigdataanalytics-worker-3 ~]$ /usr/hdp/current/kafka-broker/bin/kafka-console-consumer.sh --topic shadrin_data --bootstrap-server bigdataanalytics-worker-3:6667
```

В другом терминале
less for_stream/data.json


```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

student898_2@bigdataanalytics-worker-3:~
... option("kafka.bootstrap.servers", kafka_brokers). \
... option("subscribe", "shadrin_data"). \
... option("startingOffsets", "earliest"). \
... option("endingOffsets", """"{"shadrin_data":{"0":20}}"""). \
... load()
>>> out = console_output(raw_data, 10)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 2, in console_output
  File "/usr/hdp/current/spark2-client/python/pyspark/sql/dataframe.py", line 238, in writeStream
    return DataStreamWriter(self)
  File "/usr/hdp/current/spark2-client/python/pyspark/sql/streaming.py", line 684, in __init__
    self._jwrite = df._jdf.writeStream()
  File "/usr/hdp/current/spark2-client/python/lib/py4j-0.10.7-src.zip/py4j/java_gateway.py", line 1257, in __call__
  File "/usr/hdp/current/spark2-client/python/pyspark/sql/utlis.py", line 69, in deco
    raise AnalysisException(s.split(':', 1)[1], stackTrace)
pyspark.sql.utlis.AnalysisException: u'writeStream' can be called only on streaming Dataset/DataFrame;"
>>>
```

```
raw_data = spark.readStream. \
    format("kafka"). \
    option("kafka.bootstrap.servers", kafka_brokers). \
    option("subscribe", "shadrin_data"). \
    option("startingOffsets", "earliest"). \
    option("maxOffsetsPerTrigger", "5"). \
    load()
```

```
out = console_output(raw_data, 10)
out.stop()
```

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

student898_2@bigdataanalytics-worker-3:~
-----
Batch: 2
-----
+-----+-----+-----+-----+-----+-----+-----+
| key|      value|      topic|partition|offset|      timestamp|timestampType|
+-----+-----+-----+-----+-----+-----+-----+
| null|[20 20 20 20 20 2...|shadrin_data|      0|    10|2022-01-10 18:38:...|      0|
| null|[20 20 20 20 20 2...|shadrin_data|      0|    11|2022-01-10 18:38:...|      0|
| null|[20 20 20 20 20 2...|shadrin_data|      0|    12|2022-01-10 18:38:...|      0|
| null|[20 20 20 20 20 2...|shadrin_data|      0|    13|2022-01-10 18:38:...|      0|
| null|[20 20 20 20 20 2...|shadrin_data|      0|    14|2022-01-10 18:38:...|      0|
+-----+-----+-----+-----+-----+-----+-----+

out.stop()
.>>> .out.stop()
  File "<stdin>", line 1
    .out.stop()
    ^
SyntaxError: invalid syntax
>>> out.stop()
>>>
```

Парсинг сообщений.
Посмотрим в каком формате в Кафке хранятся сообщения.
raw_data.printSchema()


```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
student898_2@bigdataanalytics-worker-3:~
File "/usr/hdp/current/spark2-client/python/pyspark/sql/streaming.py", line 684, in __init__
self._jwrite = df._jdf.writeStream()
File "/usr/hdp/current/spark2-client/python/lib/py4j-0.10.7-src.zip/py4j/java_gateway.py", line 1257, in __call__
File "/usr/hdp/current/spark2-client/python/pyspark/sql/utils.py", line 69, in deco
raise AnalysisException(s.split(':', 1)[1], stackTrace)
pyspark.sql.utils.AnalysisException: u''writeStream' can be called only on streaming Dataset/DataFrame;"
>>> raw_data.printSchema()
root
|-- key: binary (nullable = true)
|-- value: binary (nullable = true)
|-- topic: string (nullable = true)
|-- partition: integer (nullable = true)
|-- offset: long (nullable = true)
|-- timestamp: timestamp (nullable = true)
|-- timestampType: integer (nullable = true)
>>>
```

`value` это всегда либо бинарный код, либо строка.
Определяем структуру данных нашего исходного датасета.
schema = StructType() \
 .add("sepalLength", FloatType()) \
 .add("sepalWidth", FloatType()) \
 .add("petalLength", FloatType()) \
 .add("petalWidth", FloatType()) \
 .add("species", StringType())
Traceback (most recent call last):
File "<stdin>", line 2, in <module>
NameError: name 'FloatType' is not defined

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка
student898_2@bigdataanalytics-worker-3:~
|-- value: binary (nullable = true)
|-- topic: string (nullable = true)
|-- partition: integer (nullable = true)
|-- offset: long (nullable = true)
|-- timestamp: timestamp (nullable = true)
|-- timestampType: integer (nullable = true)
>>> schema = StructType() \  
...     .add("sepalLength", FloatType()) \  
...     .add("sepalWidth", FloatType()) \  
...     .add("petalLength", FloatType()) \  
...     .add("petalWidth", FloatType()) \  
...     .add("species", StringType())  
Traceback (most recent call last):  
File "<stdin>", line 2, in <module>  
NameError: name 'FloatType' is not defined
>>>
```

value_iris = raw_data \
 .select(F.from_json(F.col("value").cast("String"), schema).alias("value"), "offset")
Traceback (most recent call last):
File "<stdin>", line 2, in <module>
NameError: name 'schema' is not defined

```
student898_2@bigdataanalytics-worker-3:~ - Терминал
Файл  Правка  Вид  Терминал  Вкладки  Справка

student898_2@bigdataanalytics-worker-3:~
|-- timestampType: integer (nullable = true)

>>> schema = StructType() \
...     .add("sepalLength", FloatType()) \
...     .add("sepalWidth", FloatType()) \
...     .add("petalLength", FloatType()) \
...     .add("petalWidth", FloatType()) \
...     .add("species", StringType())
Traceback (most recent call last):
  File "<stdin>", line 2, in <module>
NameError: name 'FloatType' is not defined

>>> value_iris = raw_data \
...     .select(F.from_json(F.col("value").cast("String"), schema).alias("value"), "offset")
Traceback (most recent call last):
  File "<stdin>", line 2, in <module>
NameError: name 'schema' is not defined

>>>
```

2.file source.py + 3. kafka source.py запускаются одной командой sparksubmit

применить, выполнить она повторяет эти команды не из консоли, а просто одной командой sparksubmit что то, что то, что то файл который скину, т. е. Запустить на прямую

Команда для запуска файлов Spark-Submit

spark-submit —master spark://VirtualBox:7077 /home/igor/3.kafka_source.py `2.file_source.py`