Custom Bootcamp week-5

25-09-2023 to 29-09-2023

25-09-2023

In [2]:

```
from pyspark.sql import SparkSession

spark = SparkSession.builder.getOrCreate()
test_df = [("James","Sales","NY",90000,34,10000),
    ("Michael","Sales","NY",86000,56,20000),
    ("Robert","Sales","CA",81000,30,23000),
    ("Maria","Finance","CA",90000,24,23000),
    ("Raman","Finance","CA",99000,40,24000),
    ("Scott","Finance","NY",83000,36,19000),
    ("Jen","Finance","NY",79000,53,15000),
    ("Jeff","Marketing","CA",80000,25,18000),
    ("Kumar","Marketing","NY",91000,50,21000)
  ]


ud_scehma = ["employee_name","department","state","salary","age","bonus"]

df = spark.createDataFrame(data=test_df,schema = ud_scehma)
```

```
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/09/25 04:18:44 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... usin
g builtin-java classes where applicable
```

In [3]:

```
df.show()
```

```
+-------------+----------+-----+------+---+-----+
|employee_name|department|state|salary|age|bonus|
+-------------+----------+-----+------+---+-----+
|        James|     Sales|   NY| 90000| 34|10000|
|      Michael|     Sales|   NY| 86000| 56|20000|
```

25-09-2023

In [5]:
```
df.cache().show()
```

```
23/09/25 04:19:31 WARN CacheManager: Asked to cache already cached data.
+-------------+----------+-----+------+---+-----+
|employee_name|department|state|salary|age|bonus|
+-------------+----------+-----+------+---+-----+
|        James|     Sales|   NY| 90000| 34|10000|
|      Michael|     Sales|   NY| 86000| 56|20000|
|       Robert|     Sales|   CA| 81000| 30|23000|
|        Maria|   Finance|   CA| 90000| 24|23000|
|        Raman|   Finance|   CA| 99000| 40|24000|
|        Scott|   Finance|   NY| 83000| 36|19000|
|          Jen|   Finance|   NY| 79000| 53|15000|
|         Jeff| Marketing|   CA| 80000| 25|18000|
|        Kumar| Marketing|   NY| 91000| 50|21000|
+-------------+----------+-----+------+---+-----+
```

In [14]:
```
from pyspark import StorageLevel

df.persist(StorageLevel.MEMORY_ONLY)
```

```
23/09/25 04:27:22 WARN CacheManager: Asked to cache already cached data.
```

Out[14]:
```
DataFrame[employee_name: string, department: string, state: string, salary: bigint, age: bigint, bonu
s: bigint]
```

In [15]:
```
df.show()
```

```
+-------------+----------+-----+------+---+-----+
|employee_name|department|state|salary|age|bonus|
+-------------+----------+-----+------+---+-----+
|        James|     Sales|   NY| 90000| 34|10000|
|      Michael|     Sales|   NY| 86000| 56|20000|
```

25-09-2023

In [16]:
```
df.unpersist()
```

Out[16]:
```
DataFrame[employee_name: string, department: string, state: string, salary: bigint, age: bigint, bonu
s: bigint]
```

In [19]:
```
from pyspark.sql.functions import *
df.cache().count()
```

Out[19]:
```
9
```

In [20]:
```
import sys
print(sys.version)
```

```
3.11.4 (main, Jul  5 2023, 14:15:25) [GCC 11.2.0]
```

In [21]:
```
test = StorageLevel(useDisk=False,useMemory=True,useOffHeap=False,deserialized=False)
```

In [23]:
```
df.persist(storageLevel=test)
```

```
23/09/25 04:38:12 WARN CacheManager: Asked to cache already cached data.
```

Out[23]:
```
DataFrame[employee_name: string, department: string, state: string, salary: bigint, age: bigint, bonu
s: bigint]
```

25-09-2023

In [32]:
```
df.groupBy('department','state').agg(sum('salary').alias("sum")\
                                    ,avg('salary').alias("avg"))\
.filter(col('avg')>90000).show()
```

```
+----------+-----+------+-------+
|department|state|   sum|    avg|
+----------+-----+------+-------+
|   Finance|   CA|189000|94500.0|
| Marketing|   NY| 91000|91000.0|
+----------+-----+------+-------+
```

In [29]:
```
df.createOrReplaceTempView('emp')
```

In [30]:
```
spark.sql('select * from emp').show()
```

```
+-------------+----------+-----+------+---+-----+
|employee_name|department|state|salary|age|bonus|
+-------------+----------+-----+------+---+-----+
|        James|     Sales|   NY| 90000| 34|10000|
|      Michael|     Sales|   NY| 86000| 56|20000|
|       Robert|     Sales|   CA| 81000| 30|23000|
|        Maria|   Finance|   CA| 90000| 24|23000|
|        Raman|   Finance|   CA| 99000| 40|24000|
|        Scott|   Finance|   NY| 83000| 36|19000|
|          Jen|   Finance|   NY| 79000| 53|15000|
|         Jeff| Marketing|   CA| 80000| 25|18000|
|        Kumar| Marketing|   NY| 91000| 50|21000|
+-------------+----------+-----+------+---+-----+
```

25-09-2023

In [33]:
```
spark.sql('create database idashell')
```

Out[33]: DataFrame[]

In [34]:
```
spark.sql('use idashell')
```

Out[34]: DataFrame[]

In [45]:
```
df.write.saveAsTable('table01')
```

In [46]:
```
spark.sql('describe table01').show()
```

```
+-------------+---------+-------+
|     col_name|data_type|comment|
+-------------+---------+-------+
|employee_name|   string|   null|
|   department|   string|   null|
|        state|   string|   null|
|       salary|   bigint|   null|
|          age|   bigint|   null|
|        bonus|   bigint|   null|
+-------------+---------+-------+
```

25-09-2023

In [47]:
```
spark.sql('describe extended table01').show()
```

```
+--------------------+--------------------+-------+
|            col_name|           data_type|comment|
+--------------------+--------------------+-------+
|       employee_name|              string|   null|
|          department|              string|   null|
|               state|              string|   null|
|              salary|              bigint|   null|
|                 age|              bigint|   null|
|               bonus|              bigint|   null|
|                    |                    |       |
|# Detailed Table ...|                    |       |
|             Catalog|       spark_catalog|       |
|            Database|            idashell|       |
|               Table|             table01|       |
|        Created Time|Mon Sep 25 05:50:...|       |
|         Last Access|             UNKNOWN|       |
|          Created By|         Spark 3.4.1|       |
|                Type|             MANAGED|       |
|            Provider|             parquet|       |
|            Location|file:/home/labuse...|       |
+--------------------+--------------------+-------+
```

In [39]:
```
df.write.option("path","/home/labuser/Documents/database").saveAsTable('table001')
```

25-09-2023

In [48]:
```
spark.sql('drop table idashell.table01')
```

Out[48]: DataFrame[]

In [49]:
```
spark.sql('drop table idashell.table001')
```

Out[49]: DataFrame[]

In [51]:
```
df.write.partitionBy("department").csv("/home/labuser/Documents/depfolder")
```

In [52]:
```
df.write.partitionBy("department",'state').csv("/home/labuser/Documents/depstate")
```

In [53]:
```
data = [
    '{"name":"sushant","age":23}',
    '{"name":"virat","age":30}'
]
```

In [54]:
```
from pyspark.sql.types import *
```

25-09-2023

In [54]:
```
from pyspark.sql.types import *
```

In [55]:
```
schema = StructType([\
                    StructField("name",StringType(),True),\
                    StructField("age",IntegerType(),True)])
```

In [57]:
```
df = spark.read.schema(schema).json(spark.sparkContext.parallelize(data))
```

In [58]:
```
json_data = [
    '{"name": "Alice", "age": 25, "address": {"city": "New York", "state": "NY"}}',
    '{"name": "Bob", "age": 30, "address": {"city": "San Francisco", "state": "CA"}}',
    '{"name": "Charlie", "age": 35, "address": {"city": "Los Angeles", "state": "CA"}}'
]

schema = StructType([
    StructField("name", StringType(), True),
    StructField("age", IntegerType(), True),
    StructField("address", StructType([
        StructField("city", StringType(), True),
        StructField("state", StringType(), True)
    ]), True)
])


df = spark.read.schema(schema).json(spark.sparkContext.parallelize(json_data))
```

26-09-2023

Cmd 1

Python

```python
dbutils.secrets.listScopes()
```

```
[SecretScope(name='shellkey')]
```

Command took 0.27 seconds -- by a user at 9/27/2023, 10:32:53 AM on unknown compute

Cmd 2

```python
dbutils.fs.mount(source = "wasbs://input@shellgen2account.blob.core.windows.net",mount_point ="/mnt/input",extra_configs = {"fs.azure.account.key.shellgen2account.blob.core.windows.net":dbutils.secrets.get(scope = "gen2scope", key = "gen2secret")})
```

Shift+Enter to run

Shift+Ctrl+Enter to run selected text

26-09-2023

Cmd 1

Python

```python
dbutils.fs.mount(source = "wasbs://input@storageshell00001.blob.core.windows.net",mount_point ="/mnt/input",extra_configs = {"fs.azure.account.key.storageshell00001.blob.core.windows.net":dbutils.secrets.get(scope = "shellkey", key = "shellkey")})
```

```
True
```

Command took 12.81 seconds -- by a user at 9/27/2023, 10:36:40 AM on unknown compute

Cmd 2

```python
dbutils.secrets.listScopes()
```

```
[SecretScope(name='shellkey')]
```

Command took 0.09 seconds -- by a user at 9/27/2023, 10:37:38 AM on unknown compute

Cmd 3

```python
dbutils.fs.ls('/mnt/input')
```

```
[]
```

Command took 0.28 seconds -- by a user at 9/27/2023, 10:38:07 AM on unknown compute

26-09-2023

```
1  dbutils.fs.mounts()
```

```
[MountInfo(mountPoint='/databricks-datasets', source='databricks-datasets', encryptionType=''),
 MountInfo(mountPoint='/Volumes', source='UnityCatalogVolumes', encryptionType=''),
 MountInfo(mountPoint='/databricks/mlflow-tracking', source='databricks/mlflow-tracking', encryptionType=''),
 MountInfo(mountPoint='/databricks-results', source='databricks-results', encryptionType=''),
 MountInfo(mountPoint='/databricks/mlflow-registry', source='databricks/mlflow-registry', encryptionType=''),
 MountInfo(mountPoint='/mnt/input', source='wasbs://input@storageshell00001.blob.core.windows.net', encryptionType=''),
 MountInfo(mountPoint='/Volume', source='DbfsReserved', encryptionType=''),
 MountInfo(mountPoint='/volumes', source='DbfsReserved', encryptionType=''),
 MountInfo(mountPoint='/', source='DatabricksRoot', encryptionType=''),
 MountInfo(mountPoint='/volume', source='DbfsReserved', encryptionType='')]
```

Command took 0.27 seconds -- by a user at 9/27/2023, 11:09:53 AM on unknown compute

Cmd 5

```
1  df  = spark.read.csv("/mnt/input/zipcodes.csv",inferSchema=True,header=True)
```

df: pyspark.sql.dataframe.DataFrame = [RecordNumber: integer, Zipcode: integer ... 18 more fields]

Command took 1.05 seconds -- by a user at 9/27/2023, 11:12:25 AM on unknown compute

26-09-2023

Cmd 6

```
df.write.option('path','/mnt/input/externaltable').saveAsTable('zipcodes')
```

Command took 6.88 seconds -- by a user at 9/27/2023, 11:13:21 AM on unknown compute

Cmd 7

```
df.display()
```

Table +

	RecordNumber	Zipcode	ZipCodeType	City	State	LocationType	Lat	Long	Xa
1	1	704	STANDARD	PARC PARQUE	PR	NOT ACCEPTABLE	17.96	-66.22	0.3
2	2	704	STANDARD	PASEO COSTA DEL SUR	PR	NOT ACCEPTABLE	17.96	-66.22	0.3
3	10	709	STANDARD	BDA SAN LUIS	PR	NOT ACCEPTABLE	18.14	-66.26	0.3
4	61391	76166	UNIQUE	CINGULAR WIRELESS	TX	NOT ACCEPTABLE	32.72	-97.31	-0.
5	61392	76177	STANDARD	FORT WORTH	TX	PRIMARY	32.75	-97.33	-0.
6	61393	76177	STANDARD	FT WORTH	TX	ACCEPTABLE	32.75	-97.33	-0.
7	4	704	STANDARD	URB EUGENE RICE	PR	NOT ACCEPTABLE	17.96	-66.22	0.

21 rows | 0.37 seconds runtime

Refreshed 4 days ago

26-09-2023

Cmd 8

SQL

```
%sql

delete from zipcodes where RecordNumber=1
```

_sqldf: pyspark.sql.dataframe.DataFrame = [num_affected_rows: long]

Table +

	num_affected_rows
1	1

1 row | 6.44 seconds runtime

Refreshed 4 days ago

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[11]`. Learn more

Command took 6.44 seconds -- by a user at 9/27/2023, 11:17:53 AM on unknown compute

26-09-2023

Cmd 9

SQL

```
%sql

vacuum zipcodes retain 169 hours
```

_sqldf: pyspark.sql.dataframe.DataFrame = [path: string]

Table +

	path
1	dbfs:/mnt/input/externaltable

1 row | 45.38 seconds runtime

Refreshed 4 days ago

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[12]`. Learn more

Command took 45.38 seconds -- by a user at 9/27/2023, 11:22:36 AM on unknown compute

Cmd 10

26-09-2023

Cmd 10

SQL

```
%sql

optimize zipcodes zorder by (RecordNumber)
```

_sqldf: pyspark.sql.dataframe.DataFrame = [path: string, metrics: struct]

Table +

	path	metrics
1	dbfs:/mnt/input/externaltable	{"numFilesAdded": 0, "numFilesRemoved": 0, "filesAdded": {"min": null, "max": null, "avg": 0, "totalFiles": 0, "totalSize": 0}, "filesRemoved": {"min": null, "max": null, "avg": 0, "totalFiles": 0, "totalSize": 0}, "partitionsOptimized": 0, "zOrderStats": {"strategyName": "minCubeSize(107374182400)", "inputCubeFiles": {"num": 0, "size": 0}, "inputOtherFiles": {"num": 1, "size": 6418}, "inputNumCubes": 0, "mergedFiles": {"num": 0, "size": 0}, "numOutputCubes": 0, "mergedNumCubes": null}, "numBatches": 0, "totalConsideredFiles": 1, "totalFilesSkipped": 1, "preserveInsertionOrder": false, "numFilesSkippedToReduceWriteAmplification": 0, "numBytesSkippedToReduceWriteAmplification": 0, "startTimeMs": 1695794047563, "endTimeMs": 1695794048225, "totalClusterParallelism": 4, "totalScheduledTasks": 0, "autoCompactParallelismStats": null, "deletionVectorStats": {"numDeletionVectorsRemoved": 0, "numDeletionVectorRowsRemoved": 0}, "numTableColumns": 20, "numTableColumnsWithStats": 20, "totalTaskExecutionTimeMs": 0, "skippedArchivedFiles": 0, "clusteringMetrics": null}

26-09-2023

Python

```python
df.write.partitionBy('City','State').option('path','/mnt/input/partitiontable').saveAsTable('idazip')
```

Command took 21.17 seconds -- by a user at 9/27/2023, 11:40:13 AM on unknown compute

Cmd 12

```sql
%sql

select * from zipcodes version as of 0
```

_sqldf: pyspark.sql.dataframe.DataFrame = [RecordNumber: integer, Zipcode: integer ... 18 more fields]

Table +

	RecordNumber	Zipcode	ZipCodeType	City	State	LocationType	Lat	Long	Xa
1	1	704	STANDARD	PARC PARQUE	PR	NOT ACCEPTABLE	17.96	-66.22	0.3
2	2	704	STANDARD	PASEO COSTA DEL SUR	PR	NOT ACCEPTABLE	17.96	-66.22	0.3
3	10	709	STANDARD	BDA SAN LUIS	PR	NOT ACCEPTABLE	18.14	-66.26	0.3
4	61391	76166	UNIQUE	CINGULAR WIRELESS	TX	NOT ACCEPTABLE	32.72	-97.31	-0.
5	61392	76177	STANDARD	FORT WORTH	TX	PRIMARY	32.75	-97.33	-0.
6	61393	76177	STANDARD	FT WORTH	TX	ACCEPTABLE	32.75	-97.33	-0.

26-09-2023

Cmd 13

SQL

```
%sql

restore zipcodes version as of 0
```

_sqldf: pyspark.sql.dataframe.DataFrame = [table_size_after_restore: long, num_of_files_after_restore: long ... 4 more fields]

Table +

	table_size_after_restore	num_of_files_after_restore	num_removed_files	num_restored_files	removed_files_size	restored_files_siz
1	6482	1	1	1	6418	6482

1 row | 14.62 seconds runtime

Refreshed 4 days a

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[16]`. Learn more

26-09-2023

Cmd 1

Python

```python
from pyspark.sql.functions import *

def add_column(df,col1,col2,newcol):
    return df.withColumn(newcol,concat(col1,lit(' '),col2))
```

Command took 0.12 seconds -- by a user at 9/26/2023, 2:56:10 PM on unknown compute

Cmd 2

```python
def f1(x):
    return x['City'].upper()
```

Command took 0.05 seconds -- by a user at 9/26/2023, 3:17:49 PM on unknown compute

Cmd 3

```python

```

27-09-2023

Cmd 1

Python

```
%run /Users/shellunext_1693422231860@npunext.onmicrosoft.com/notebooks/UtilityNotebook
```

Command took 0.22 seconds -- by a user at 9/26/2023, 3:16:20 PM on unknown compute

Cmd 2

```
df = spark.read.option("header",True).option('InferSchema',True).csv('/mnt/input/zipcodes.csv')
```

df: pyspark.sql.dataframe.DataFrame = [RecordNumber: integer, Zipcode: integer ... 18 more fields]

Command took 16.02 seconds -- by a user at 9/26/2023, 2:17:03 PM on unknown compute

Cmd 3

```
df.display()
```

Table +

	RecordNumber	Zipcode	ZipCodeType	City	State	LocationType	Lat	Long	Xa
1	1	704	STANDARD	PARC PARQUE	PR	NOT ACCEPTABLE	17.96	-66.22	0.3
2	2	704	STANDARD	PASEO COSTA DEL SUR	PR	NOT ACCEPTABLE	17.96	-66.22	0.3
3	10	709	STANDARD	BDA SAN LUIS	PR	NOT ACCEPTABLE	18.14	-66.26	0.3
4	61391	76166	UNIQUE	CINGULAR WIRELESS	TX	NOT ACCEPTABLE	32.72	-97.31	-0.
5	61392	76177	STANDARD	FORT WORTH	TX	PRIMARY	32.75	-97.33	-0.

27-09-2023

Cmd 4

```
df.write.parquet('/mnt/input/parquet')
```

Command took 2.86 seconds -- by a user at 9/26/2023, 2:20:10 PM on unknown compute

Cmd 5

```
df.write.saveAsTable('table01')
```

Command took 15.04 seconds -- by a user at 9/26/2023, 2:21:47 PM on unknown compute

Cmd 6

```
df.write.option('path','/mnt/input/table/zip/').saveAsTable('table1')
```

Command took 5.28 seconds -- by a user at 9/26/2023, 2:24:50 PM on unknown compute

Cmd 7

```
%sql

select * from table1
```

_sqldf: pyspark.sql.dataframe.DataFrame = [RecordNumber: integer, Zipcode: integer ... 18 more fields]

27-09-2023

Cmd 8

```
%sql
describe extended table1
```

_sqldf: pyspark.sql.dataframe.DataFrame = [col_name: string, data_type: string ... 1 more field]

Table +

	col_name	data_type
1	RecordNumber	int
2	Zipcode	int
3	ZipCodeType	string
4	City	string
5	State	string
6	LocationType	string
7	Lat	double

37 rows | 0.70 seconds runtime

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[10]`. Learn more

Command took 0.70 seconds -- by a user at 9/26/2023, 2:26:50 PM on unknown compute

27-09-2023

Cmd 9

SQL

```
%sql
describe extended table1
```

_sqldf: pyspark.sql.dataframe.DataFrame = [col_name: string, data_type: string ... 1 more field]

Table +

	col_name	data_type	co
1	RecordNumber	int	nul
2	Zipcode	int	nul
3	ZipCodeType	string	nul
4	City	string	nul
5	State	string	nul
6	LocationType	string	nul
7	Lat	double	nul

37 rows | 0.53 seconds runtime

Refreshed 5 days ago

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[11]`. Learn more

Command took 0.53 seconds -- by a user at 9/26/2023, 2:27:36 PM on unknown compute

Cmd 10

27-09-2023

```
%sql
describe extended table01
```

_sqldf: pyspark.sql.dataframe.DataFrame = [col_name: string, data_type: string ... 1 more field]

Table

	col_name	data_type	co
1	RecordNumber	int	nul
2	Zipcode	int	nul
3	ZipCodeType	string	nul
4	City	string	nul
5	State	string	nul
6	LocationType	string	nul
7	Lat	double	nul
8	Long	double	nul
9	Xaxis	double	nul
10	Yaxis	double	nul
11	Zaxis	double	nul
12	WorldRegion	string	nul

38 rows | 0.44 seconds runtime

Refreshed 5 days ago

This result is stored as PySpark data frame `_sqldf` and in the IPython output cache as `Out[12]`. Learn more

27-09-2023

Cmd 12

Python

```
add_column(df,'City','State','new_column').display()
```

Table +

	RecordNumber	Zipcode	ZipCodeType	City	State	LocationType	Lat	Long	Xa
1	1	704	STANDARD	PARC PARQUE	PR	NOT ACCEPTABLE	17.96	-66.22	0.3
2	2	704	STANDARD	PASEO COSTA DEL SUR	PR	NOT ACCEPTABLE	17.96	-66.22	0.3
3	10	709	STANDARD	BDA SAN LUIS	PR	NOT ACCEPTABLE	18.14	-66.26	0.3
4	61391	76166	UNIQUE	CINGULAR WIRELESS	TX	NOT ACCEPTABLE	32.72	-97.31	-0.
5	61392	76177	STANDARD	FORT WORTH	TX	PRIMARY	32.75	-97.33	-0.
6	61393	76177	STANDARD	FT WORTH	TX	ACCEPTABLE	32.75	-97.33	-0.
7	4	704	STANDARD	URB EUGENE RICE	PR	NOT ACCEPTABLE	17.96	-66.22	0.

21 rows | 0.36 seconds runtime

Refreshed 5 days ago

Command took 0.36 seconds -- by a user at 9/26/2023, 2:59:11 PM on unknown compute

Cmd 13

```
df1 = df.rdd.map(lambda x: x['City'].lower())
```

Command took 0.06 seconds -- by a user at 9/26/2023, 3:21:11 PM on unknown compute

27-09-2023

```
df1.collect()
```

```
['parc parque',
 'paseo costa del sur',
 'bda san luis',
 'cingular wireless',
 'fort worth',
 'ft worth',
 'urb eugene rice',
 'mesa',
 'mesa',
 'hilliard',
 'holder',
 'holt',
 'homosassa',
 'bda san luis',
 'sect lanausse',
 'spring garden',
 'springville',
 'spruce pine',
 'ash hill',
 'asheboro',
 'asheboro']
```

Command took 0.29 seconds -- by a user at 9/26/2023, 3:22:39 PM on unknown compute

27-09-2023

Cmd 1

Python

```python
dbutils.fs.ls("dbfs:/user/hive/warehouse/zipcodes_csv")
```

```
[FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_csv/_delta_log/', name='_delta_log/', size=0, modificationTime=1695788344000),
 FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_csv/part-00000-fa61ed1f-3042-4acd-9da6-a449d61c6cd5.c000.snappy.parquet', name
='part-00000-fa61ed1f-3042-4acd-9da6-a449d61c6cd5.c000.snappy.parquet', size=6482, modificationTime=1695788357000)]
```

Command took 0.36 seconds -- by a user at 9/27/2023, 10:08:53 AM on unknown compute

Cmd 2

```python
dbutils.fs.ls("dbfs:/user/hive/warehouse/zipcodes_csv/_delta_log/")
```

```
[FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_csv/_delta_log/00000000000000000000.crc', name='00000000000000000000.crc', size=
3404, modificationTime=1695788352000),
 FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_csv/_delta_log/00000000000000000000.json', name='00000000000000000000.json', siz
e=2325, modificationTime=1695788344000),
 FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_csv/_delta_log/00000000000000000001.crc', name='00000000000000000001.crc', size=
5077, modificationTime=1695788360000),
 FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_csv/_delta_log/00000000000000000001.json', name='00000000000000000001.json', siz
e=2155, modificationTime=1695788358000),
 FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_csv/_delta_log/__tmp_path_dir/', name='__tmp_path_dir/', size=0, modificationTim
e=1695788344000),
 FileInfo(path='dbfs:/user/hive/warehouse/zipcodes_csv/_delta_log/_copy_into_log/', name='_copy_into_log/', size=0, modificationTim
e=1695788355000)]
```

Command took 0.10 seconds -- by a user at 9/27/2023, 10:11:32 AM on unknown compute

28-09-2023

input

2

operation

Double

```python
import time
dbutils.widgets.text('input','1')
dbutils.widgets.text('operation','Double')

input_value = dbutils.widgets.get("input")
operation_type = dbutils.widgets.get("operation")

# Define a function to perform the specified operation
def perform_operation(value, operation):
    if operation == "Double":
        return value * 2
    elif operation == "Square":
        return value ** 2
    else:
        return "Invalid operation"

# Process the user input
try:
    input_value = float(input_value)
    result = perform_operation(input_value, operation_type)
    print(f"Result of {operation_type} operation on {input_value}: {result}")
except ValueError:
    print("Invalid input. Please enter a numeric value.")

```

28-09-2023

Cmd 1

Python

```python
from pyspark.sql.types import *

schema = StructType([StructField("lsoa_code", StringType(), True),\
                     StructField("borough", StringType(), True),\
                     StructField("major_category", StringType(), True),\
                     StructField("minor_category", StringType(), True),\
                     StructField("value", StringType(), True),\
                     StructField("year", StringType(), True),\
                     StructField("month", StringType(), True)])

Streamdf = spark.readStream.schema(schema).option("header",True).csv("/mnt/input/destination")

trimmedDF = Streamdf.select(
                                Streamdf.borough,
                                Streamdf.year,
                                Streamdf.month,
                                Streamdf.value
                                )\
                        .withColumnRenamed(
                                "value",
                                "convictions"
                                )
```

28-09-2023

Cmd 2

```
query = trimmedDF.writeStream\
                .outputMode("append")\
                .format("csv") \
                .option("path", "/mnt/input/processeddata") \
                .option("checkpointLocation", "/mnt/input/checkpoint") \
                .start()\
                .awaitTermination()
```

Cancelled

Command took 6.33 minutes -- by a user at 9/26/2023, 4:33:01 PM on unknown compute

Cmd 3

```
1
```

28-09-2023

Cmd 1

Python

```python
print("Sushant")
```

```
Sushant
```

Command took 0.07 seconds -- by a user at 9/27/2023, 2:03:52 PM on unknown compute

Cmd 2

```python
jdbc_url = "jdbc:sqlserver://idaservernew.database.windows.net:1433; databaseName=idadb02"

connections = {
    "user":"sqladmin",
    "password":"IDAshell@123",
    "driver":"com.microsoft.sqlserver.jdbc.SQLServerDriver"
}
```

Command took 0.09 seconds -- by a user at 9/27/2023, 2:43:39 PM on unknown compute

Cmd 3

```python
df = spark.read.jdbc(url = jdbc_url,table="SalesLT.Customer",properties=connections)

```

df: pyspark.sql.dataframe.DataFrame = [CustomerID: integer, NameStyle: boolean ... 13 more fields]

Command took 0.61 seconds -- by a user at 9/27/2023, 2:43:41 PM on unknown compute

28-09-2023

Cmd 4

Python

```
df.display()
```

Table +

	CustomerID	NameStyle	Title	FirstName	MiddleName	LastName	Suffix	CompanyNar
1	1	false	Mr.	Orlando	N.	Gee	null	A Bike Store
2	2	false	Mr.	Keith	null	Harris	null	Progressive Sp
3	3	false	Ms.	Donna	F.	Carreras	null	Advanced Bike
4	4	false	Ms.	Janet	M.	Gates	null	Modular Cycle
5	5	false	Mr.	Lucy	null	Harrington	null	Metropolitan S
6	6	false	Ms.	Rosmarie	J.	Carroll	null	Aerobic Exerci
7	7	false	Mr.	Dominic	P.	Gash	null	Associated Bik

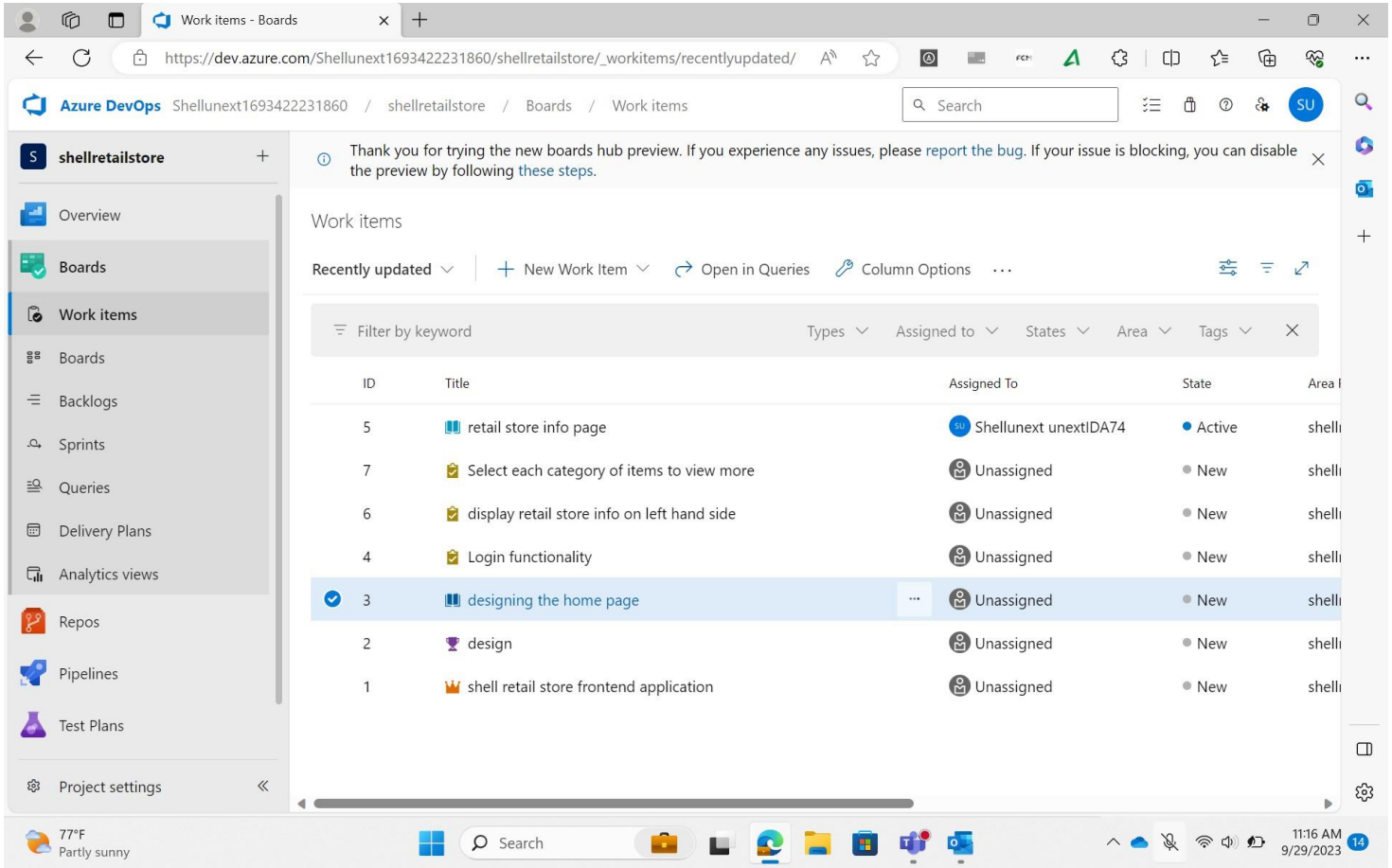
847 rows | 0.75 seconds runtime

Refreshed 4 days ago

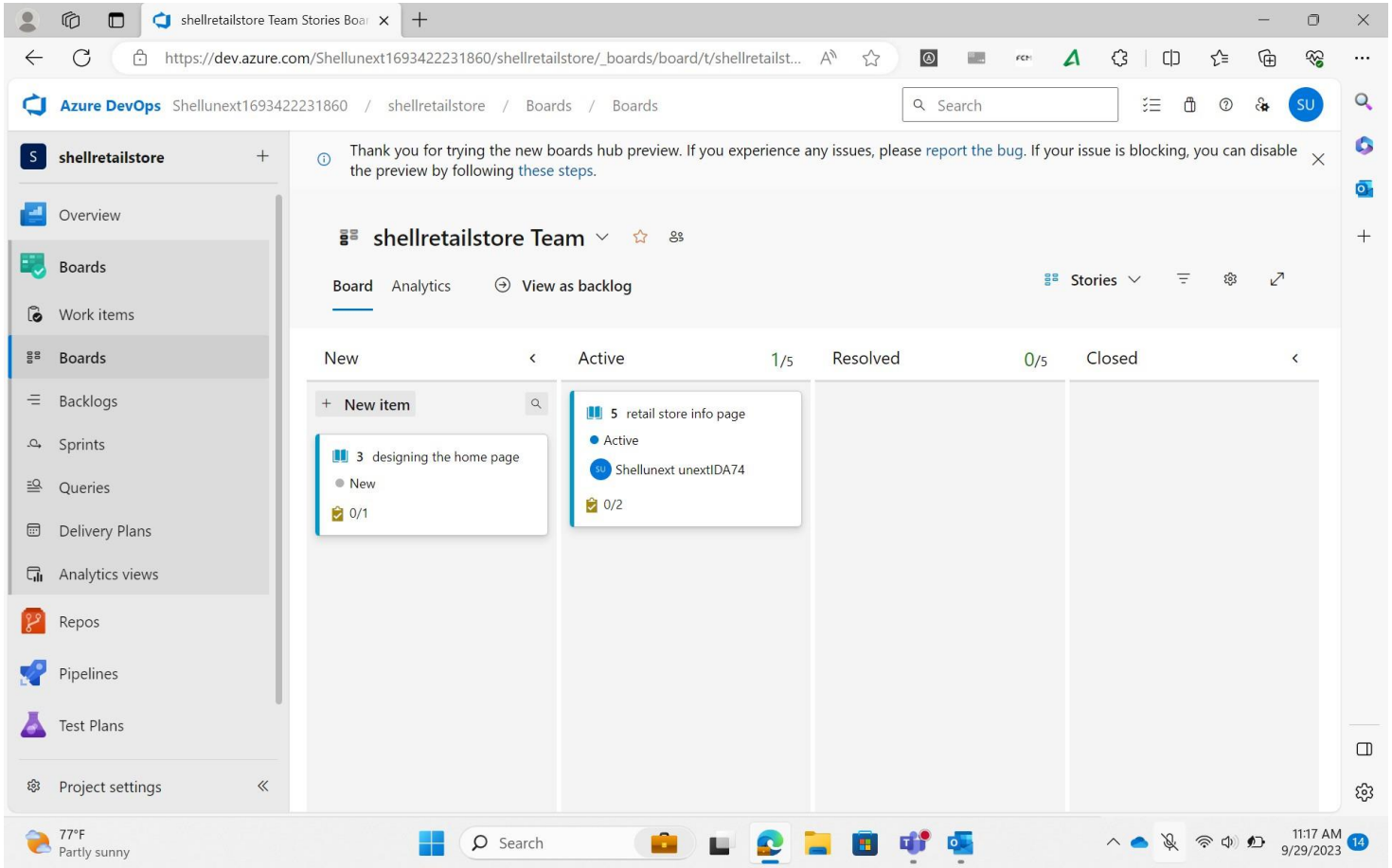
Command took 0.75 seconds -- by a user at 9/27/2023, 2:43:43 PM on unknown compute

Cmd 5

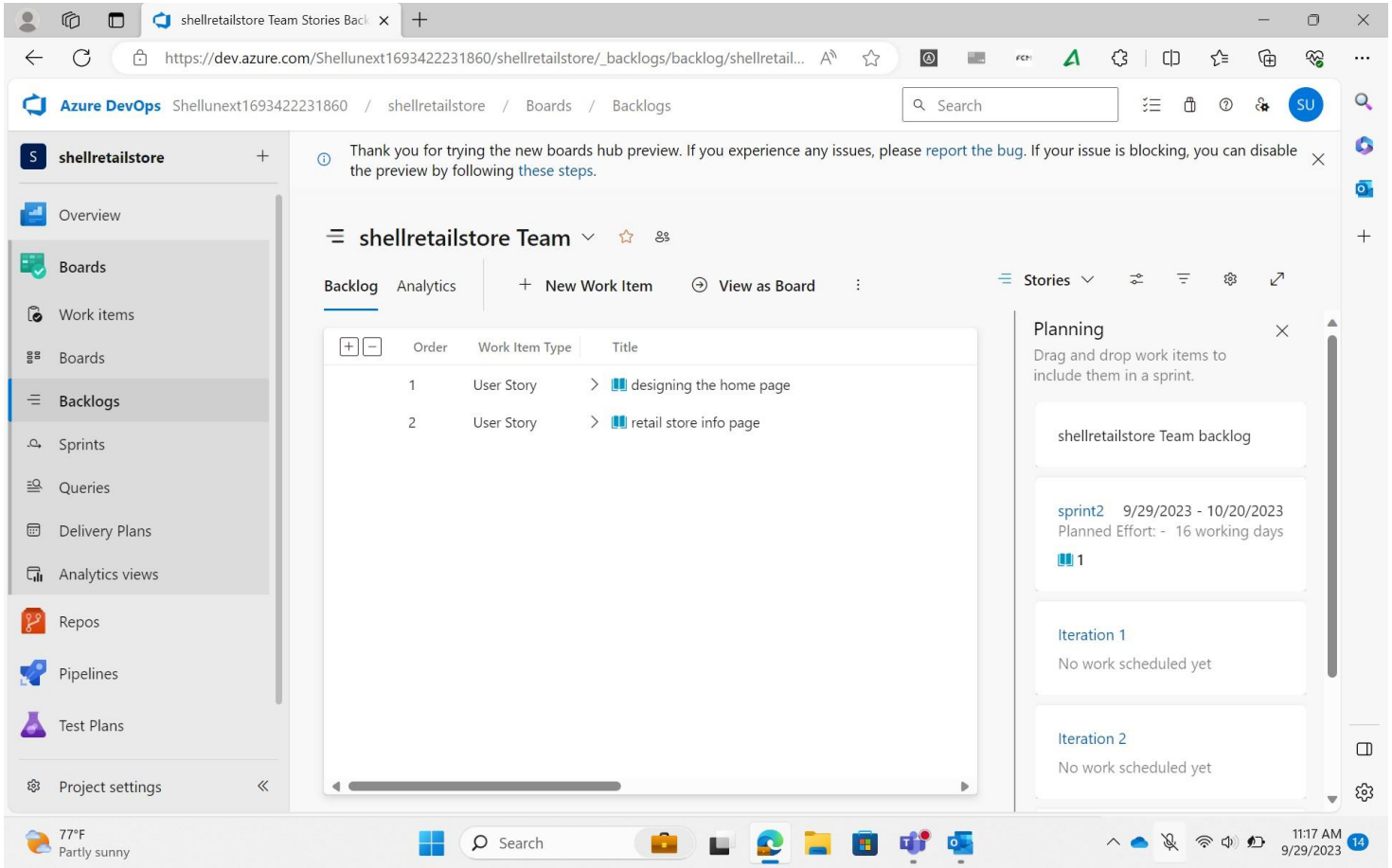
29-09-2023


Work items - Boards
https://dev.azure.com/Shellunext1693422231860/shellretailstore/_workitems/recently updated/
Azure DevOps Shellunext1693422231860 / shellretailstore / Boards / Work items
Search
SU
shellretailstore
Overview
Boards
Work items
Boards
Backlogs
Sprints
Queries
Delivery Plans
Analytics views
Repos
Pipelines
Test Plans
Project settings
Thank you for trying the new boards hub preview. If you experience any issues, please report the bug. If your issue is blocking, you can disable the preview by following these steps.
Work items
Recently updated
New Work Item
Open in Queries
Column Options
Filter by keyword
Types
Assigned to
States
Area
Tags
ID
Title
Assigned To
State
5 retail store info page Shellunext unextIDA74 Active
7 Select each category of items to view more Unassigned New
6 display retail store info on left hand side Unassigned New
4 Login functionality Unassigned New
3 designing the home page Unassigned New
2 design Unassigned New
1 shell retail store frontend application Unassigned New
77°F
Partly sunny
Search
11:16 AM
9/29/2023

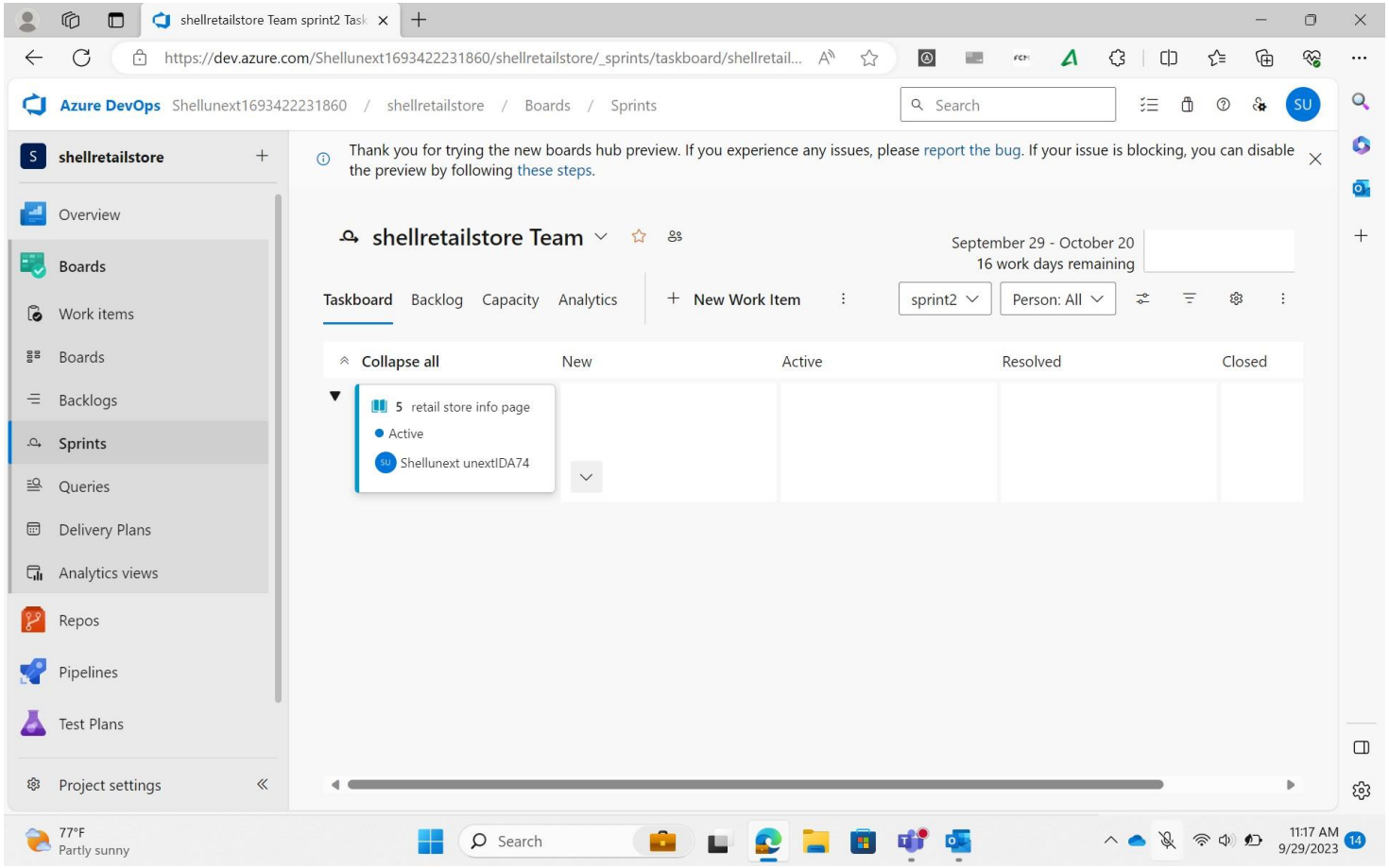
29-09-2023


shellretailstore Team Stories Boa
https://dev.azure.com/Shellunext1693422231860/shellretailstore/_boards/board/t/shellretailst...
Azure DevOps Shellunext1693422231860 / shellretailstore / Boards / Boards
Search
SU
shellretailstore
Overview
Boards
Work items
Boards
Backlogs
Sprints
Queries
Delivery Plans
Analytics views
Repos
Pipelines
Test Plans
Project settings
Thank you for trying the new boards hub preview. If you experience any issues, please report the bug. If your issue is blocking, you can disable the preview by following these steps.
shellretailstore Team
Board
Analytics
View as backlog
Stories
New
Active 1/5
Resolved 0/5
Closed
New item
3 designing the home page
New
0/1
5 retail store info page
Active
Shellunext unextIDA74
0/2
77°F
Partly sunny
Search
11:17 AM
9/29/2023

29-09-2023


shellretailstore Team Stories Back
https://dev.azure.com/Shellunext1693422231860/shellretailstore/_backlogs/backlog/shellretail...
Azure DevOps Shellunext1693422231860 / shellretailstore / Boards / Backlogs
Search
SU
shellretailstore
Overview
Boards
Work items
Boards
Backlogs
Sprints
Queries
Delivery Plans
Analytics views
Repos
Pipelines
Test Plans
Project settings
Thank you for trying the new boards hub preview. If you experience any issues, please report the bug. If your issue is blocking, you can disable the preview by following these steps.
shellretailstore Team
Backlog
Analytics
New Work Item
View as Board
Stories
Order
Work Item Type
Title
1
User Story
designing the home page
2
User Story
retail store info page
Planning
Drag and drop work items to include them in a sprint.
shellretailstore Team backlog
sprint2 9/29/2023 - 10/20/2023
Planned Effort: - 16 working days
1
Iteration 1
No work scheduled yet
Iteration 2
No work scheduled yet
77°F
Partly sunny
Search
11:17 AM
9/29/2023

29-09-2023


shellretailstore Team sprint2 Task
https://dev.azure.com/Shellunext1693422231860/shellretailstore/_sprints/taskboard/shellretail...
Azure DevOps Shellunext1693422231860 / shellretailstore / Boards / Sprints
Search
SU
shellretailstore
Overview
Boards
Work items
Boards
Backlogs
Sprints
Queries
Delivery Plans
Analytics views
Repos
Pipelines
Test Plans
Project settings
Thank you for trying the new boards hub preview. If you experience any issues, please report the bug. If your issue is blocking, you can disable the preview by following these steps.
shellretailstore Team
September 29 - October 20
16 work days remaining
Taskboard
Backlog
Capacity
Analytics
New Work Item
sprint2
Person: All
Collapse all
New
Active
Resolved
Closed
5 retail store info page
Active
Shellunext unextIDA74
77°F
Partly sunny
Search
11:17 AM
9/29/2023