



# Spark & Hadoop

## -Projekat 1-

# Bike Washington

Big Data Systems

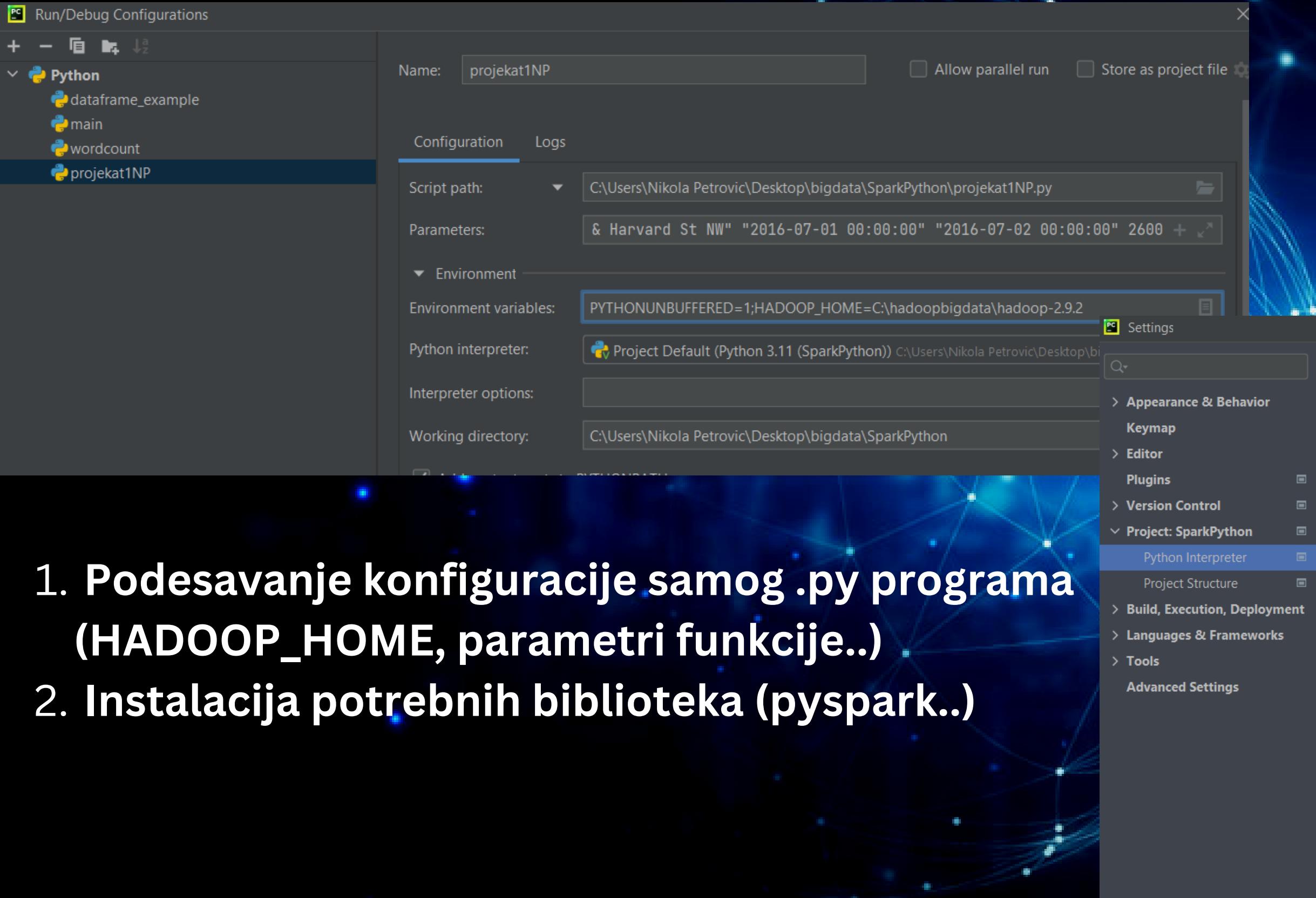
Mentor:

Prof. dr Dragan H. Stojanovic

Student:

Petrovic Nikola 1466

# Podesavanje PyCharm okruzenja



1. Podesavanje konfiguracije samog .py programa  
(**HADOOP\_HOME**, parametri funkcije..)
2. Instalacija potrebnih biblioteka (pyspark..)

1. Importovanje biblioteka i pomocnih funkcija
2. Ucitavanje prosledjenih argumenata
3. Kreiranje spark sesije
4. Ucitavanje podataka iz ulaznog foldera

```
import sys
from pyspark.sql import SparkSession
from pyspark.sql.functions import lit, hour, trunc, round, stddev, min, max

if __name__ == "__main__":
    #check the number of arguments
    if len(sys.argv) < 2:
        print("Usage: projekat1NP.py <parameters> ")
        exit(-1)

    #Set a name for the application
    appName = "DataFrame Example"
    input_folder = sys.argv[1]
    input_station = sys.argv[2] 2.
    input_date1 = sys.argv[3]
    input_date2 = sys.argv[4]
    duration=sys.argv[5]

    #create a new Spark application and get the Spark session object
    spark = SparkSession.builder.appName(appName).getOrCreate() 3.

    #read in the CSV dataset as a DataFrame
    #inferSchema option forces Spark to automatically specify data column types
    #header option forces Spark to automatically fetch column names from the CSV file
    dataset = spark.read \
        .option("inferSchema", True) \
        .option("header", True) \ 4.
        .csv(input_folder)
```

```
dataset.select(min("duration"), max("duration"), mean("duration"), stddev("duration")).show(10)
```

min(duration)	max(duration)	avg(duration)	stddev_samp(duration)
60	86394	1165.864400526216	2299.2018698659117

```
print("\n Station details:\n")
result = dataset.filter((col("Start station") == input_station))
# Show only top 5 rows
result.show(5)

print("\n Station details from-to date:\n")
result = result.filter((col("Start date") >= lit(input_date1)) & (col("Start date") <= lit(input_date2)))
# Show only top 5 rows

result.show(5)
```

Station details from-to date:

Duration	Start date	End date	Start station number	Start station	End station number	End station	Bike number	Member
623	2016-07-01 00:00:11	2016-07-01 00:10:34	31105	14th & Harvard St NW	31105	14th & Harvard St NW	W20029	Mem
429	2016-07-01 00:32:55	2016-07-01 00:40:05	31105	14th & Harvard St NW	31404	9th & Upshur St NW	W20563	Mem
248	2016-07-01 02:10:00	2016-07-01 02:14:08	31105	14th & Harvard St NW	31602	Park Rd & Holmead...	W21940	Mem
573	2016-07-01 05:35:16	2016-07-01 05:44:49	31105	14th & Harvard St NW	31507	1st & Washington ...	W01418	Mem
1885	2016-07-01 05:56:30	2016-07-01 06:27:55	31105	14th & Harvard St NW	31108	4th & M St SW	W01057	Mem

```
rows = result.count()
print(f"\nNumber of rides with duration greater than {duration} : {rows}\n")
```

```
Number of rides from 2016-07-01 00:00:00 to 2016-07-02 00:00:00 : 35
```

```
# Most common days for ride in two ways
# I
most_popular_day = dataset.groupBy("startDay").count().orderBy(desc("count")).first()["startDay"]

# Most common days for ride
# II
days = dataset.groupBy("startDay").agg(count("*").alias("count"))
print("\nPopularity of a day for biking\n")
days = days.sort(desc("count"))
days.show()
print("People in Washington likes to drive bike in the middle of the week")
print("\n Most popular day:\n" + most_popular_day)
```

Popularity of a day for biking

startDay	count
Wednesday	971821
Tuesday	952071
Friday	951455
Thursday	940094
Monday	894586
Saturday	884199
Sunday	851035

People in Washington likes to drive bike in the middle of the week

Most popular day:  
Wednesday

Display the sorted data in tabular format with hour and average number of bike rides for each hour

```
+-----+  
|hour|avg_rides|  
+-----+  
| 17| 29787.46|  
| 18| 24468.17|  
| 8 | 21875.71|  
| 16| 20777.38|  
| 19| 16667.13|  
| 15| 16322.42|  
| 13| 15226.92|  
| 12| 15225.25|  
| 14| 14852.29|  
| 7 | 14724.71|  
| 9 | 13088.29|  
| 11| 12894.58|  
| 20| 11676.58|  
| 10| 10835.38|  
| 21| 8417.63|  
| 22| 6032.13|  
| 6 | 5689.67|  
| 23| 3621.58|  
| 0 | 2069.38|  
| 5 | 1678.96|  
+-----+
```

# Najpopularniji sati u danu za voznju

```
# Extract the hour of the day from the "Start date" column  
dataset = dataset.withColumn("hour", hour(col("Start date")))  
# Group the data by the hour of the day  
grouped_data = dataset.groupBy("hour")  
# Count the number of bike rides for each hour  
ride_counts = grouped_data.agg(count("*"))  
# Divide the counts by the number of hours in a day  
avg_rides = ride_counts.withColumn("avg_rides", col("count(1)") / 24)  
# Round the avg_rides column to 2 decimal places  
avg_rides = avg_rides.withColumn("avg_rides", round(col("avg_rides"), 2))  
# Sort the hours by the average number of bike rides in descending order  
sorted_avg_rides = avg_rides.sort(col("avg_rides").desc())  
# The most popular hours will be the first rows of the sorted averages  
most_popular_hours = sorted_avg_rides.select("hour", "avg_rides")  
# Display the data in tabular format with hour and average number of bike rides for each hour  
print("\nDisplay the sorted data in tabular format with hour and average number of bike rides for each hour")  
most_popular_hours.show()
```

# Kreiranje potrebnih kontejnera

Koriscen je Windows operativni sistem, gde je instalirana pomocna aplikacija "Docker Desktop".



## Koraci u kreiranju kontejnera:

- Pozicioniranje u cmd-u na lokaciju gde se nalaze docker-compose.yml i hadoop.env fajlovi
- Pozivanje komande: **docker run -it --network bde --env-file hadoop.env -p 4040:4040 --name spark bde2020/spark-base:3.1.2-hadoop3.2 bash**

Navedenom komandom se postize:

- Kreiranje kontejnera (namenode, nodemanager, resourcemanager, datanode, historyserver, spark-master, spark-worker-1, spark-worker-2) konfiguracija se nalazi u docker-compose.yml fajlu
- Kreiranje mreze "bde" na koju su povezani svi navedeni kontejneri, kako bi medjusobno bili vidljivi
- Kreiranje spark kontejnera koji ce biti vidljiv na portu 4040

# Izgled aplikacije Docker Desktop nakon kreiranja kontejnera

The screenshot shows the Docker Desktop interface with the following details:

**Containers** (Selected)

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)

Only show running containers

Search

Extensions BETA

Add Extensions

NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
spark a17a84c04f9d	<a href="#">bde2020/spark-base:3.1.2-hadoop3.2</a>	Exited	4040:4040		<span>⋮</span> <span>⋮</span> <span>⋮</span>
spark-hadoop	-	Running (8/8)			<span>⋮</span> <span>⋮</span> <span>⋮</span>
spark-worker-2 1ba81bfe9505	<a href="#">bde2020/spark-worker:3.1.2-hadoop3.2</a>	Running	<a href="#">8072:8071</a>	1 minute ago	<span>⋮</span> <span>⋮</span> <span>⋮</span>
spark-worker-1 50cc6f7612d4	<a href="#">bde2020/spark-worker:3.1.2-hadoop3.2</a>	Running	<a href="#">8071:8071</a>	1 minute ago	<span>⋮</span> <span>⋮</span> <span>⋮</span>
datanode 69ae9eea519b	<a href="#">bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-jav8</a>	Running		5 minutes ago	<span>⋮</span> <span>⋮</span> <span>⋮</span>
resourcemanager 1b4c5d6f440c	<a href="#">bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2</a>	Running		4 minutes ago	<span>⋮</span> <span>⋮</span> <span>⋮</span>
spark-master 54d4508ea505	<a href="#">bde2020/spark-master:3.1.2-hadoop3.2</a>	Running	<a href="#">7077:7077</a> <a href="#">8070:8070</a>	1 minute ago	<span>⋮</span> <span>⋮</span> <span>⋮</span>
historyserver 94e424e57989	<a href="#">bde2020/hadoop-historyserver:2.0.0-hadoop3.2.1-jav</a>	Running		5 minutes ago	<span>⋮</span> <span>⋮</span> <span>⋮</span>
namenode 4cf5d2ab044e	<a href="#">bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-jav8</a>	Running	<a href="#">9000:9000</a> <a href="#">9870:9870</a>	5 minutes ago	<span>⋮</span> <span>⋮</span> <span>⋮</span>
nodemanager 0cf53d3ffe15	<a href="#">bde2020/hadoop-nodemanager:2.0.0-hadoop3.2.1-jav</a>	Running		5 minutes ago	<span>⋮</span> <span>⋮</span> <span>⋮</span>

Activate Windows  
Go to Settings to activate [Showing 16 items](#)

# Izgled portainer.io prozora nakon kreiranja mreže i povezivanja kontejnera na istu

The screenshot shows the Portainer Business Edition interface running in a web browser. The URL in the address bar is <https://localhost:9443/#/2/docker/networks/d578bf8f4e14825251471ef3ddbbe778b87570381e910bf509bcda93f06d1ec>. The browser's status bar indicates "Not secure".

The left sidebar shows the navigation menu:

- Home
- local
  - Dashboard
  - App Templates
  - Stacks
  - Containers
  - Images
  - Networks
  - Volumes
  - Events
  - Host
- Settings
- Users
- Environments
- Registries
- Licenses
- Authentication logs
- Notifications
- Settings

The "Networks" option is selected in the sidebar.

The main content area displays network configuration details and a list of containers in the network:

Internal	false
IPV4 Subnet	- 172.23.0.0/16
IPV4 Gateway	- 172.23.0.1
IPV4 IP Range	-
IPV4 Excluded IPs	-

**Access control**

Ownership: administrators ⓘ

**Containers in network**

Container Name	IPv4 Address	IPv6 Address	MacAddress	Actions
spark-worker-2	172.23.0.9/16	-	02:42:ac:17:00:09	<button>Leave Network</button>
spark-worker-1	172.23.0.8/16	-	02:42:ac:17:00:08	<button>Leave Network</button>
datanode	172.23.0.2/16	-	02:42:ac:17:00:02	<button>Leave Network</button>
resourcemanager	172.23.0.4/16	-	02:42:ac:17:00:04	<button>Leave Network</button>
spark-master	172.23.0.7/16	-	02:42:ac:17:00:07	<button>Leave Network</button>
historyserver	172.23.0.3/16	-	02:42:ac:17:00:03	<button>Leave Network</button>
namenode	172.23.0.5/16	-	02:42:ac:17:00:05	<button>Leave Network</button>
nodemanager	172.23.0.6/16	-	02:42:ac:17:00:06	<button>Leave Network</button>

Activate Windows.  
Go to Settings to activate Windows.

# Postavljanje podataka na HDFS

```
Command Prompt - docker exec -it namenode bash
Microsoft Windows [Version 10.0.19044.2486]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Nikola Petrovic>cd Desktop
C:\Users\Nikola Petrovic\Desktop>cd bigdata
C:\Users\Nikola Petrovic\Desktop\bigdata>cd SparkPython
C:\Users\Nikola Petrovic\Desktop\bigdata\SparkPython>docker cp projekat1NP.py namenode:/data
C:\Users\Nikola Petrovic\Desktop\bigdata\SparkPython>docker cp Data namenode:/Data

C:\Users\Nikola Petrovic\Desktop\bigdata\SparkPython>docker exec -it namenode bash
root@4cf5d2ab044e:/# hdfs dfs -put /data/projekat1NP.py /proj
put: `/proj/projekat1NP.py': File exists
root@4cf5d2ab044e:/# hdfs dfs -put /data/projekat1NP.py /proj
2023-01-27 23:52:44,664 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
root@4cf5d2ab044e:/# hdfs dfs -put /data/Data /proj
```

Pozicioniranje na ciljni folder iz koga se kopiraju podaci na hdfs

Kopiranje podataka na hadoop namenode

Ako folder ne postoji prvo ga treba kreirati: hdfs dfs -mkdir /proj:

Pokretanje namenode

Kopiranje podataka na hdfs

# Izgled podataka na HDFS-u port: 9870

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities ▾

## Browse Directory

/proj Go!   

Show 25 entries Search:

<input type="checkbox"/>	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
<input type="checkbox"/>	drwxr-xr-x	root	supergroup	0 B	Jan 24 22:23	0	0 B	.idea	
<input type="checkbox"/>	drwxr-xr-x	root	supergroup	0 B	Jan 24 22:23	0	0 B	Data	
<input type="checkbox"/>	-rw-r--r--	root	supergroup	3.09 KB	Jan 24 22:23	3	128 MB	dataframe_example.py	
<input type="checkbox"/>	drwxr-xr-x	root	supergroup	0 B	Jan 24 22:23	0	0 B	input_folder	
<input type="checkbox"/>	drwxr-xr-x	root	supergroup	0 B	Jan 24 22:23	0	0 B	venv	

Showing 1 to 5 of 5 entries Previous 1 Next

Hadoop, 2019.

# Spark je dostupan na portu 8070

- URL adresa pristupa je vidljiva u gornjem levom uglu
  - Na ovom prozoru je moguce pracenje aktivnosti workera, dostupne memorije, statusa aplikacija, vremena izvršavanje..

localhost:8070/#completed-app

YouTube Добродошли на Ф... New chat Referee Store - Che... Rezultati: nogomet,... Welcome To TechBr... 2 12 13 provera da li je text... 1 SEMESTAR – Goo... 2 SEMESTAR – Goo...

**Apache Spark 3.1.2** Spark Master at spark://54d4508ea505:7077

**URL:** spark://54d4508ea505:7077

**Alive Workers:** 2

**Cores in use:** 8 Total, 0 Used

**Memory in use:** 10.2 GiB Total, 0.0 B Used

**Resources in use:**

**Applications:** 0 Running, 0 Completed

**Drivers:** 0 Running, 0 Completed

**Status:** ALIVE

**Workers (2)**

Worker Id	Address	State	Cores	Memory	Resources
worker-20230127233614-172.23.0.8-42135	172.23.0.8:42135	ALIVE	4 (0 Used)	5.1 GiB (0.0 B Used)	
worker-20230127233614-172.23.0.9-34373	172.23.0.9:34373	ALIVE	4 (0 Used)	5.1 GiB (0.0 B Used)	

**Running Applications (0)**

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

**Completed Applications (0)**

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

# spark-submit

```
C:\Users\Nikola Petrovic\Desktop\bigdata\spark-hadoop>docker run -it --network bde --env-file hadoop.env -p 4040:4040 --name spark bde2020/spark-base:3.1.2-hadoop3.2 bash
bash-5.0# /spark/bin/spark-submit --master spark://54d4508ea505:7077 hdfs://namenode:9000/proj/projekat1NP.py hdfs://namenode:9000/proj/Data "14th & Harvard St NW" "2016-07-01 00:00:00" "2016-07-03 00:00:00" 2700
```

- Pozicioniranje u spark bin folder i pozivanje komande spark-submit
- Postavljanje URL adrese spark mastera(**--master spark://54d4508ea505:7077**)
- Adresa programa i podataka sa HDFS-a(**hdfs://namenode:9000/proj/projekat1NP.py hdfs://namenode:9000/proj/Data** )
- Ostali parametri funkcije(**"14th & Harvard St NW" "2016-07-01 00:00:00" "2016-07-02 00:00:00" 2600**)

# Pracenje rada spark aplikacije

localhost:4040/jobs/

YouTube Добродошли на Ф... New chat Referee Store - Che... Rezultati: nogomet... Welcome To TechBr... 2 12 13 provera da li je text... 1 SEMESTAR – Goo... 2 SEMESTAR – Goo...

APACHE Spark 3.1.2 Jobs Stages Storage Environment Executors SQL DataFrame Example application UI

## Spark Jobs (?)

User: root  
Total Uptime: 10 min  
Scheduling Mode: FIFO  
Active Jobs: 1  
Completed Jobs: 4

Event Timeline

Enable zooming

Executors

- Added
- Removed

Jobs

- Succeeded
- Failed
- Running

10:21 10:22 10:23 10:24 10:25 10:26 10:27 10:28 10:29 10:30  
Sat 28 January

Executor driver added  
Executor 1 added  
Executor 0 added  
csv at Nat csv at NativeMethodAccessorImpl.java:0 (Job 1)  
count at NativeMe describe at NativeMethodAccessorImpl.java:0 (Job 4)

### Active Jobs (1)

Page: 1 1 Pages. Jump to 1 . Show 100 items in a page. Go

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
4	describe at NativeMethodAccessorImpl.java:0 describe at NativeMethodAccessorImpl.java:0 (kill)	2023/01/28 10:25:26	5.5 min	0/2	0/10 (8 running)

Page: 1 1 Pages. Jump to 1 . Show 100 items in a page. Go

### Completed Jobs (4)

Page: 1 1 Pages. Jump to 1 . Show 100 items in a page. Go

Activate Windows  
Go to Settings to activate Windows.

# Najpopularnija stanica za polazak

Ovo izracunavanje moze biti od koristi gradskim vlastima kako bi znali  
gde ujutru treba postaviti najvise bicikala:

Start station	count
Columbus Circle / Union Station	124700
Lincoln Memorial	118207
Jefferson Dr & 14th St SW	107015
Massachusetts Ave & Dupont Circle NW	90250
15th & P St NW	76684
Jefferson Memorial	72190
Smithsonian-National Mall / Jefferson Dr & 12th St SW	65571
14th & V St NW	61713
Thomas Circle	61337
New Hampshire Ave & T St NW	58773

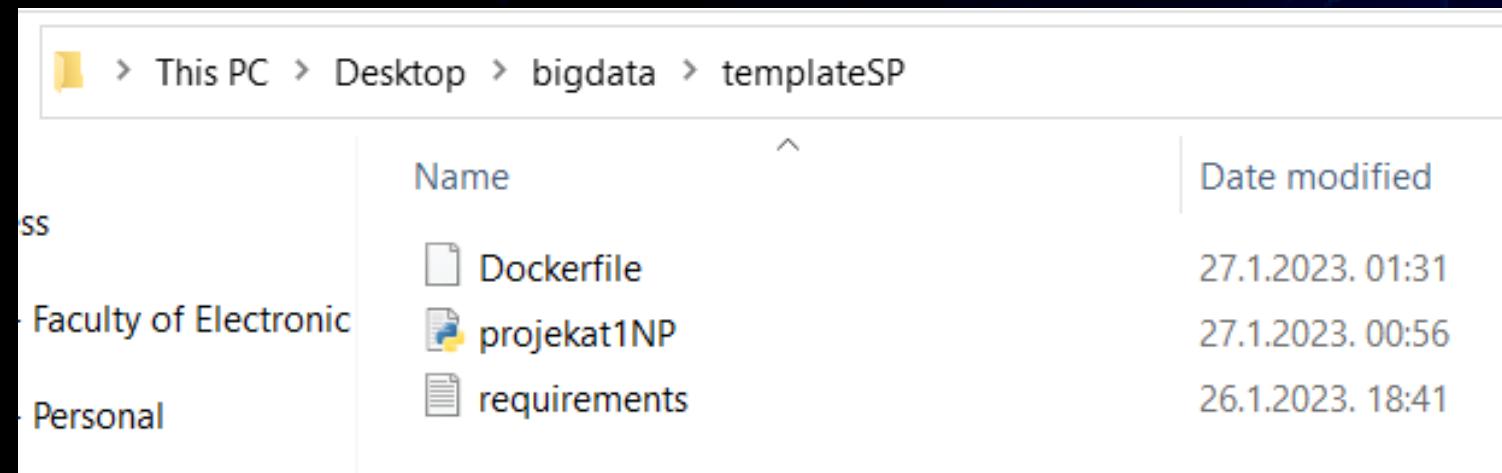
only showing top 10 rows

Most popular start station:

Columbus Circle / Union Station

# Pokretanje spark aplikacije na kontejnerima Nacin 2.

- Kreiranje fajla koji ce biti korisen, ciji sadrzaj cine fajlovi requirements.txt, Dockerfile i .py aplikacija.



```
FROM bde2020/spark-python-template:3.1.2-hadoop3.2
ENV SPARK_APPLICATION_PYTHON_LOCATION /app/projekat1NP.py
ENV SPARK_APPLICATION_ARGS "hdfs://namenode:9000/proj/Data" "14th & Harvard"
ENV SPARK_SUBMIT_ARGS --executor-memory 3G --executor-cores 3
```

Aplikacija Parametri  
Podesavanje memorije

```
pyspark==3.3.1
```

(u ovom slucaju to nije neophodno)

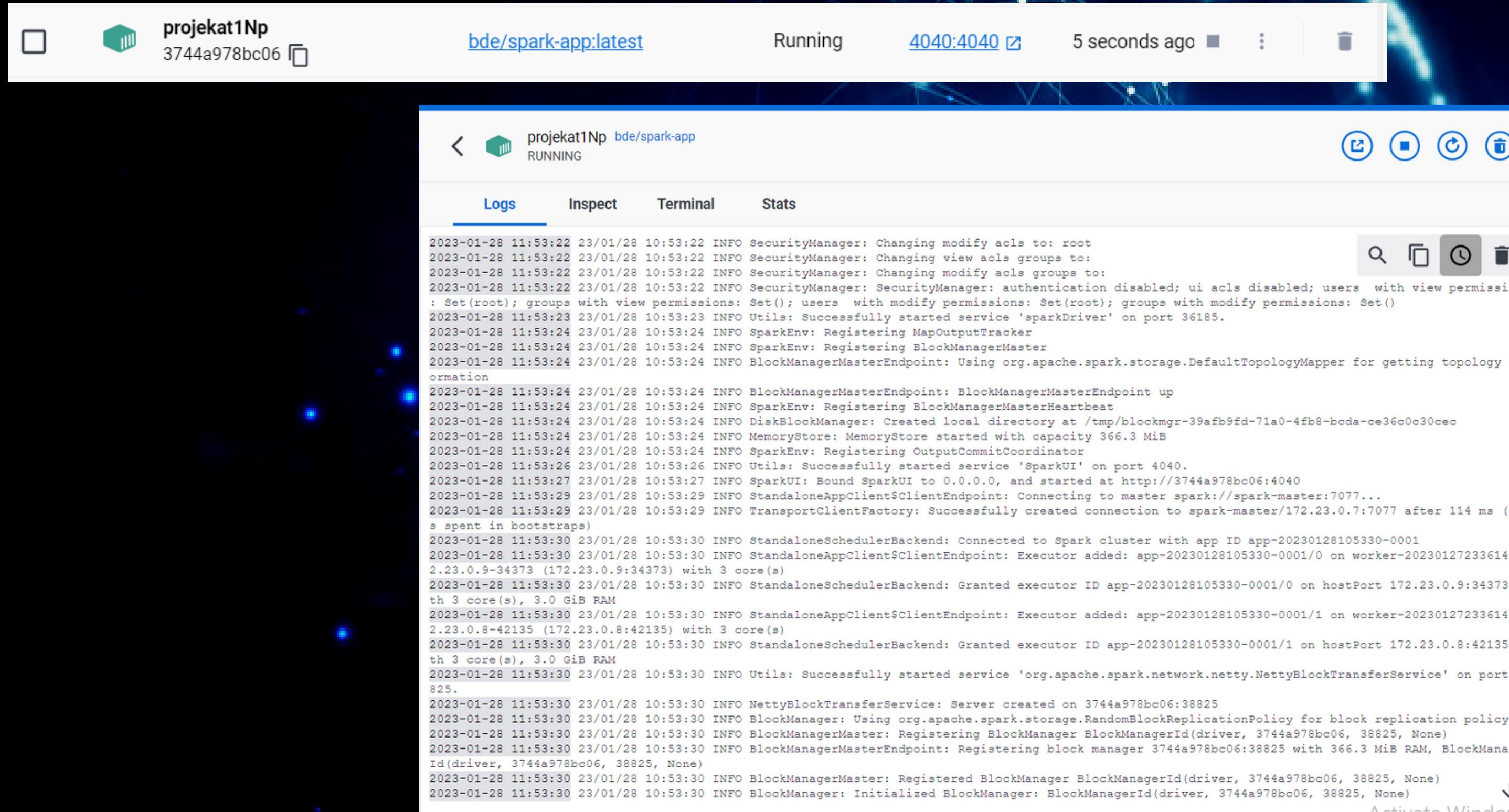
# Pokretanje aplikacije

```
C:\Users\Nikola Petrovic\Desktop\bigdata\templateSP>docker build --rm -t bde/spark-app .
[+] Building 3.4s (9/9) FINISHED
=> [internal] load build definition from Dockerfile                               0.5s
=> => transferring dockerfile: 32B                                              0.1s
=> [internal] load .dockerignore                                                 0.3s
=> => transferring context: 2B                                              0.1s
=> [internal] load metadata for docker.io/bde2020/spark-python-template:3.1.2-hadoop3.2 2.4s
=> [internal] load build context                                                 0.1s
=> => transferring context: 101B                                             0.0s
=> [1/1] FROM docker.io/bde2020/spark-python-template:3.1.2-hadoop3.2@sha256:905fcb3cf7890faaa21789aefb2e330d921 0.0s
=> CACHED [2/1] COPY requirements.txt /app/
=> CACHED [3/1] RUN cd /app && pip3 install -r requirements.txt                0.0s
=> CACHED [4/1] COPY . /app                                                     0.0s
=> exporting to image                                                       0.2s
=> => exporting layers                                                       0.0s
=> => writing image sha256:e5dd1e351ac4e18b6848b7607984d21c22228389b854a9ae1d3e163fd2f76882 0.0s
=> => naming to docker.io/bde/spark-app                                         0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

```
C:\Users\Nikola Petrovic\Desktop\bigdata\templateSP>docker run --name projekat1Np --net bde -p 4040:4040 -d bde/spark-ap
p
```

# Pracenje rada aplikacije moguce u logovima u aplikaciji Docker Desktop



Activate Window

# Provera trenutnog stanja na spark masteru

Spark Master at spark://54d4508ea505:7077

URL: spark://54d4508ea505:7077

**Alive Workers:** 2

**Cores in use:** 8 Total, 6 Used

**Memory in use:** 10.2 GiB Total, 6.0 GiB Used

**Resources in use:**

**Applications:** 1 Running, 1 Completed

**Drivers:** 0 Running, 0 Completed

**Status:** ALIVE

**Workers (2)**

Worker Id	Address	State	Cores	Memory	Resources
worker-20230127233614-172.23.0.8-42135	172.23.0.8:42135	ALIVE	4 (3 Used)	5.1 GiB (3.0 GiB Used)	
worker-20230127233614-172.23.0.9-34373	172.23.0.9:34373	ALIVE	4 (3 Used)	5.1 GiB (3.0 GiB Used)	

**Running Applications (1)**

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20230128105330-0001	(kill) DataFrame Example	6	3.0 GiB		2023/01/28 10:53:30	root	RUNNING	1.0 min

**Completed Applications (1)**

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20230128102110-0000	DataFrame Example	8	1024.0 MiB		2023/01/28 10:21:10	root	FINISHED	21 min

# Pracenje logova u Docker Desktop-u

The screenshot shows the Docker Desktop interface with a container named "projekat1NNpp" running. The container's image is "bde/spark-app". The "Logs" tab is selected, displaying the following log output:

```
2023-01-28 12:18:34 23/01/28 11:18:34 INFO BlockManagerInfo: Added broadcast_7_piece0 in memory on 172.23.0.10:39763 (size: 2 iB)
2023-01-28 12:18:35 23/01/28 11:18:35 INFO TaskSetManager: Finished task 0.0 in stage 4.0 (TID 18) in 1367 ms on 172.23.0.10
2023-01-28 12:18:35 23/01/28 11:18:35 INFO TaskSchedulerImpl: Removed TaskSet 4.0, whose tasks have all completed, from pool
2023-01-28 12:18:35 23/01/28 11:18:35 INFO DAGScheduler: ResultStage 4 (showString at NativeMethodAccessorImpl.java:0) finished in 1.639 s
2023-01-28 12:18:35 23/01/28 11:18:35 INFO DAGScheduler: Job 3 is finished. Cancelling potential speculative or zombie tasks for this job
2023-01-28 12:18:35 23/01/28 11:18:35 INFO TaskSchedulerImpl: Killing all running tasks in stage 4: Stage finished
2023-01-28 12:18:35 23/01/28 11:18:35 INFO DAGScheduler: Job 3 finished: showString at NativeMethodAccessorImpl.java:0, took 1.856180 s
2023-01-28 12:18:35 23/01/28 11:18:35 INFO CodeGenerator: Code generated in 279.477 ms
2023-01-28 12:18:35 +-----+
+-----+
2023-01-28 12:18:35 |Duration| Start date| End date|Start station number| Start station|End station number| End stat
ion|Bike number|Member type|
2023-01-28 12:18:35 +-----+-----+-----+-----+-----+-----+-----+
-----+
2023-01-28 12:18:35 | 623|2016-07-01 00:00:11|2016-07-01 00:10:34| 31105|14th & Harvard St NW| 31105|14th & Harvard St
NW| W20029| Member|
2023-01-28 12:18:35 | 430|2016-07-01 00:00:18|2016-07-01 00:07:29| 31023|Fairfax Dr & Wils...| 31025|Central Library /
...| W20623| Casual|
2023-01-28 12:18:35 | 696|2016-07-01 00:00:54|2016-07-01 00:12:31| 31620| 5th & F St NW| 31628| 1st & K St
SE| W00777| Member|
2023-01-28 12:18:35 | 456|2016-07-01 00:01:04|2016-07-01 00:08:40| 31202| 14th & R St NW| 31234|20th & O St NW /
...| W00068| Member|
2023-01-28 12:18:35 | 1099|2016-07-01 00:01:07|2016-07-01 00:19:26| 31268| 12th & U St NW| 31506|1st & Rhode Islan
...| W20726| Member|
2023-01-28 12:18:35 +-----+
-----+
2023-01-28 12:18:35 only showing top 5 rows
2023-01-28 12:18:35
2023-01-28 12:18:35
2023-01-28 12:18:35 Station details:
2023-01-28 12:18:35
```

Hvalana  
paznji!

