

Exp. No : 6**Handling JSON data using HDFS and Python****1. Create emp.json file**

```
GNU nano 7.2 emp.json
{
  "name": "John Doe",
  "age": 30,
  "department": "HR",
  "salary": 50000
},
{
  "name": "Jane Smith",
  "age": 25,
  "department": "IT",
  "salary": 60000
},
{
  "name": "Alice Johnson",
  "age": 35,
  "department": "Finance",
  "salary": 70000
},
{
  "name": "Bob Brown",
  "age": 28,
  "department": "Marketing",
  "salary": 55000
},
{
  "name": "Charlie Black",
  "age": 45,
  "department": "IT",
  "salary": 80000
}
```

2. Install jq package

```
Copr repo for PyCharm owned by phracek      454 B/s | 1.8 kB    00:04
Fedora 40 - x86_64                        3.5 kB/s | 10 kB    00:02
Fedora 40 openh264 (From Cisco) - x86_64  1.4 kB/s | 989 B    00:00
Fedora 40 - x86_64 - Updates              4.2 kB/s | 7.6 kB    00:01
Fedora 40 - x86_64 - Updates              843 kB/s | 4.7 MB    00:05
google-chrome                             1.5 kB/s | 1.3 kB    00:00
google-chrome                             1.0 kB/s | 1.8 kB    00:01
RPM Fusion for Fedora 40 - Nonfree - NVIDIA Dri 6.3 kB/s | 16 kB    00:02
RPM Fusion for Fedora 40 - Nonfree - NVIDIA Dri 702 B/s | 4.9 kB    00:07
RPM Fusion for Fedora 40 - Nonfree - Steam    5.8 kB/s | 15 kB    00:02
RPM Fusion for Fedora 40 - Nonfree - Steam    326 B/s | 1.5 kB    00:04
Package jq-1.7.1-7.fc40.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

3. Execute jq . emp.json command

```
lksh@fedora:~$ nano emp.json
lksh@fedora:~$ jq . emp.json
[
  {
    "name": "John Doe",
    "age": 30,
    "department": "HR",
    "salary": 50000
  },
  {
    "name": "Jane Smith",
    "age": 25,
    "department": "IT",
    "salary": 60000
  },
  {
    "name": "Alice Johnson",
    "age": 35,
    "department": "Finance",
    "salary": 70000
  },
  {
    "name": "Bob Brown",
    "age": 28,
    "department": "Marketing",
    "salary": 55000
  },
  {
    "name": "Charlie Black",
    "age": 45,
    "department": "IT",
    "salary": 80000
  }
]
lksh@fedora:~$
```

4. pip install pandas

```

bash: pip: command not found...
Install package 'python3-pip' to provide command 'pip'? [N/y] y


* Waiting in queue...
* Loading list of packages....
The following packages have to be installed:
python3-pip-23.3.2-1.fc40.noarch      A tool for installing and managing Python3 packages
Proceed with changes? [N/y] y

* Waiting in queue...
* Waiting for authentication...
* Waiting in queue...
* Downloading packages...
* Requesting data...
* Testing changes...
* Installing packages...
Defaulting to user installation because normal site-packages is not writeable
Collecting pandas
  Downloading pandas-2.2.2-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (19 kB)
Collecting numpy>=1.26.0 (from pandas)
  Downloading numpy-2.1.1-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (60 kB)
    60.9/60.9 kB 527.6 kB/s eta 0:00:00
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/lib/python3.12/site-packages (from pandas) (2.8.2)
Collecting pytz>=2020.1 (from pandas)
  Downloading pytz-2024.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas)
  Downloading tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: six>=1.5 in /usr/lib/python3.12/site-packages (fr

```

5. pip install hdf5

```

Defaulting to user installation because normal site-packages is not writeable
Collecting hdf5
  Downloading hdf5-2.7.3.tar.gz (43 kB)
    
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting docopt (from hdf5)
  Downloading docopt-0.6.2.tar.gz (25 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: requests>=2.7.0 in /usr/lib/python3.12/site-packages (from hdf5) (2.31.0)
Requirement already satisfied: six>=1.9.0 in /usr/lib/python3.12/site-packages (from hdf5) (1.16.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/lib/python3.12/site-packages (from requests>=2.7.0->hdf5) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/lib/python3.12/site-packages (from requests>=2.7.0->hdf5) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/lib/python3.12/site-packages (from requests>=2.7.0->hdf5) (1.26.19)
Building wheels for collected packages: hdf5, docopt
  Building wheel for hdf5 (pyproject.toml) ... done
  Created wheel for hdf5: filename=hdf5-2.7.3-py3-none-any.whl size=34205 sha256=0d536af61228b7f0d53e3b48d95259498753e9777c49cd399bff47eeec7511a2
  Stored in directory: /home/hayagreevan/.cache/pip/wheels/97/ae/d9/536505928dd3a458b206013b02625df8f12d22fa154f2bfd65
  Building wheel for docopt (pyproject.toml) ... done
  Created wheel for docopt: filename=docopt-0.6.2-py2.py3-none-any.whl size=13674 sha256=8355c4921fa97d2181cbc04fbfabf5706c5121b8b5ad260fc656fe8c25dee200
  Stored in directory: /home/hayagreevan/.cache/pip/wheels/1a/bf/a1/4cee4f7678c68c5875ca89eaccf460593539805c3906722228
Successfully built hdf5 docopt
Installing collected packages: docopt, hdf5
Successfully installed docopt-0.6.2 hdf5-2.7.3

```

6. Create process_data.py

```

GNU nano 7.2                                process_data.py
from hdfs import InsecureClient
import pandas as pd
import json

# Connect to HDFS
hdfs_client = InsecureClient('http://localhost:9870', user='hdfs')

# Read JSON data from HDFS
try:
    with hdfs_client.read('/home/hadoop/emp.json', encoding='utf-8') as reader:
        json_data = reader.read() # Read the raw data as a string
        if not json_data.strip(): # Check if data is empty
            raise ValueError("The JSON file is empty.")
        print(f"Raw JSON Data: {json_data[:1000]}") # Print first 1000 characters
        data = json.loads(json_data) # Load the JSON data
except json.JSONDecodeError as e:
    print(f"JSON Decode Error: {e}")
    exit(1)
except Exception as e:
    print(f"Error reading or parsing JSON data: {e}")
    exit(1)

# Convert JSON data to DataFrame
try:
    df = pd.DataFrame(data)
except ValueError as e:
    print(f"Error converting JSON data to DataFrame: {e}")
    exit(1)

# Projection: Select only 'name' and 'salary' columns
projected_df = df[['name', 'salary']]

# Aggregation: Calculate total salary
total_salary = df['salary'].sum()

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line

```

```

lksh@fedora:~$ nano process_data.py
lksh@fedora:~$ nano process_data.py
lksh@fedora:~$ hdfs dfs -mkdir /exp6
lksh@fedora:~$ hdfs dfs -put emp.json /exp6
lksh@fedora:~$

```

Output:

```
lksh@fedora:~$ nano process_data.py
lksh@fedora:~$ python3 process_data.py
Filtered JSON file saved successfully.
Projection: Select only 'name' and 'salary' columns
```

	name	salary
0	John Doe	50000
1	Jane Smith	60000
2	Alice Johnson	70000
3	Bob Brown	55000
4	Charlie Black	80000

```

Aggregation: Total salary of all employees
Total Salary: 315000

Count: Number of employees earning more than 50000
Number of High Earners (>50000): 4

Top 5 Earners:
```

	name	age	department	salary
4	Charlie Black	45	IT	80000
2	Alice Johnson	35	Finance	70000
1	Jane Smith	25	IT	60000
3	Bob Brown	28	Marketing	55000
0	John Doe	30	HR	50000

```

Skipped DataFrame (First 2 rows skipped):
```

	name	age	department	salary
2	Alice Johnson	35	Finance	70000
3	Bob Brown	28	Marketing	55000
4	Charlie Black	45	IT	80000