

Testy jednostkowe

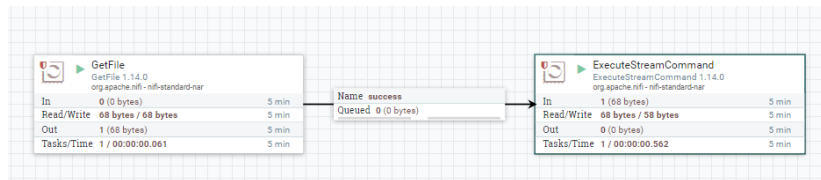
Agata Kopyt, Nikola Miszalska

January 2024

1 Testy funkcjonalne

1.1 Wywoływanie skryptu .sh

- cel: sprawdzenie, czy wywołanie skryptu .sh, który wywoła skrypt pythonowy działa poprawnie.
- kroki: uruchomienie przepływu Nifi Flow.
- oczekiwany wynik: pobranie z Api danych z dnia 2024-01-01 do katalogów stage i archive .



Rysunek 1: Nifi Flow

```
vagrant@node1:~$ cd project/nifi_in
vagrant@node1:~/project/nifi_in$ ls archive
vagrant@node1:~/project/nifi_in$ ls stage
vagrant@node1:~/project/nifi_in$
```

Rysunek 2: Zawartość katalogów stage i archive przed odpaleniem procesu.

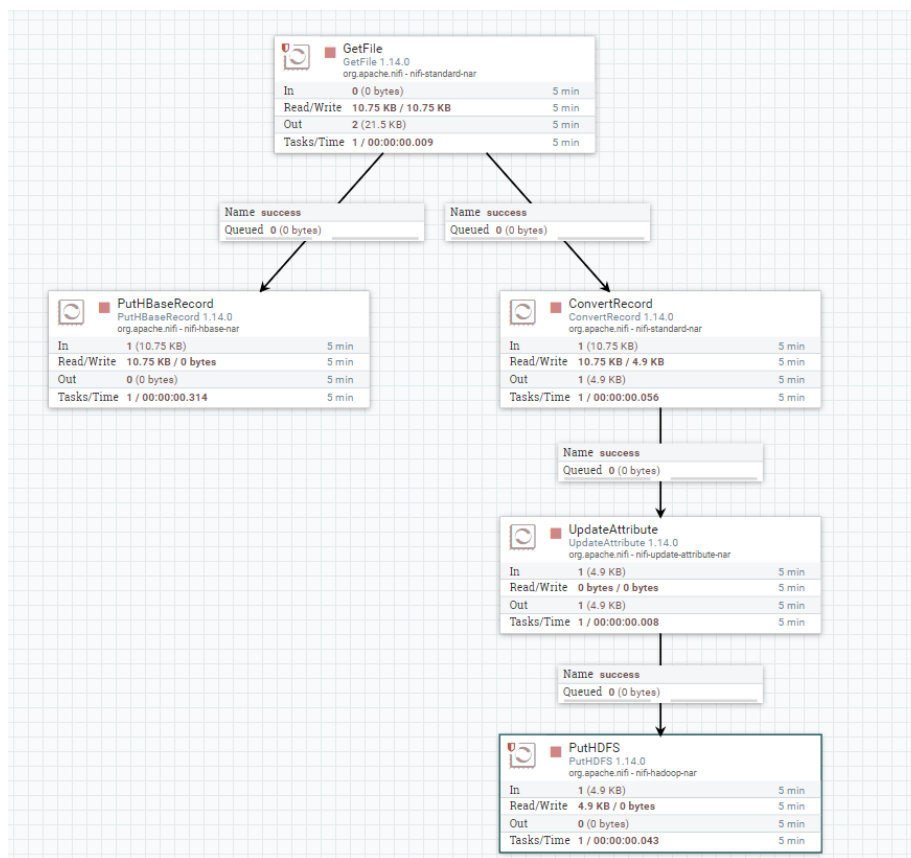
```

vagrant@node1:~/project/nifi_in$ ls archive
songs_2024-01-01.json
vagrant@node1:~/project/nifi_in$ ls stage
songs_2024-01-01.json
vagrant@node1:~/project/nifi_in$

```

Rysunek 3: Zawartość katalogów stage i archive po odpaleniu procesu.

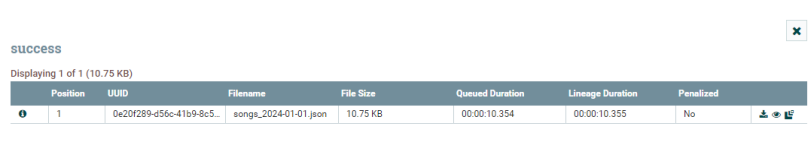
1.2 Składowanie danych dynamicznych w HDFS oraz HBase



Rysunek 4: Nifi Flow

Test 1: Pobranie pliku z katalogu stage

- cel: sprawdzenie, czy plik z katalogu stage pobiera się poprawnie.
- kroki: uruchomienie procesora GetFile.
- oczekiwany wynik: Dodanie do kolejki pliku `songs_2024-01-01.json` z katalogu stage, bez zostawiania go tam.



success

Displaying 1 of 1 (10.75 KB)

	Position	UUID	Filename	File Size	Queued Duration	Lineage Duration	Penalized
0	1	0e20f289-d56c-41b9-8c5...	songs_2024-01-01.json	10.75 KB	00:00:10.354	00:00:10.355	No

Rysunek 5: Zawartość kolejki po odpaleniu procesu GetFile

```
vagrant@node1:~$ ls project/nifi_in/stage/  
vagrant@node1:~$
```

Rysunek 6: Zawartość katalogu stage po odpaleniu procesora GetFile

Test 2: Zmiana rozszerzenia

- cel: sprawdzenie, czy zmiana rozszerzenia pliku przebiega poprawnie..
- kroki: uruchomienie procesora ConvertRecord oraz UpdateAttribute.
- oczekiwany wynik: dodanie do kolejki pliku `songs_2024-01-01.parquet`.



success

Displaying 1 of 1 (4.90 KB)

	Position	UUID	Filename	File Size	Queued Duration	Lineage Duration	Penalized
0	1	4645442c-3977-4aa0-569a-4...	songs_2024-01-01.parquet	4.90 KB	00:00:38.327	00:03:47.020	No

Rysunek 7: Zawartość kolejki po przejściu plik przez procesory.

Test 3: Składowanie danych w HBase

- cel: sprawdzenie, czy dane poprawnie ładują się do tabeli w Hbase.
- kroki: uruchomienie procesora PutHBSaeRecord.
- oczekiwany wynik: dodanie świeżych danych do tabelki w HBase.

```
hbase(main):010:0> scan 'daily_data'
ROW                                COLUMN+CELL
0 row(s)
Took 0.1294 seconds
hbase(main):011:0> scan 'daily_data'
ROW                                COLUMN+CELL
086THPnabbulzfDjRspoN             column=data:artist, timestamp=2024-01-01T21:27:41.796, value=Elley Duh\u00c3\xA9
086THPnabbulzfDjRspoN             column=data:date_added, timestamp=2024-01-01T21:27:41.796, value=2024-01-01T09:04:46
086THPnabbulzfDjRspoN             column=data:popularity, timestamp=2024-01-01T21:27:41.796, value=79
086THPnabbulzfDjRspoN             column=data:rank, timestamp=2024-01-01T21:27:41.796, value=46
086THPnabbulzfDjRspoN             column=data:track_name, timestamp=2024-01-01T21:27:41.796, value=MONEY ON THE DASH - SPE
D UP
0DHkkQjcrOXLBr9eCtcf             column=data:artist, timestamp=2024-01-01T21:27:41.796, value=Blacha 2115
0DHkkQjcrOXLBr9eCtcf             column=data:date_added, timestamp=2024-01-01T21:27:41.796, value=2024-01-01T09:04:46
0DHkkQjcrOXLBr9eCtcf             column=data:popularity, timestamp=2024-01-01T21:27:41.796, value=68
0DHkkQjcrOXLBr9eCtcf             column=data:rank, timestamp=2024-01-01T21:27:41.796, value=12
0DHkkQjcrOXLBr9eCtcf             column=data:track_name, timestamp=2024-01-01T21:27:41.796, value=Kevin sam w domu
0h3XyqV4apMraBSNuM8U7Z           column=data:artist, timestamp=2024-01-01T21:27:41.796, value=CYRIL
0h3XyqV4apMraBSNuM8U7Z           column=data:date_added, timestamp=2024-01-01T21:27:41.796, value=2024-01-01T09:04:46
0h3XyqV4apMraBSNuM8U7Z           column=data:popularity, timestamp=2024-01-01T21:27:41.796, value=70
0h3XyqV4apMraBSNuM8U7Z           column=data:rank, timestamp=2024-01-01T21:27:41.796, value=22
```

Rysunek 8: Zawartość tabeli przed i po uruchomieniu procesora.

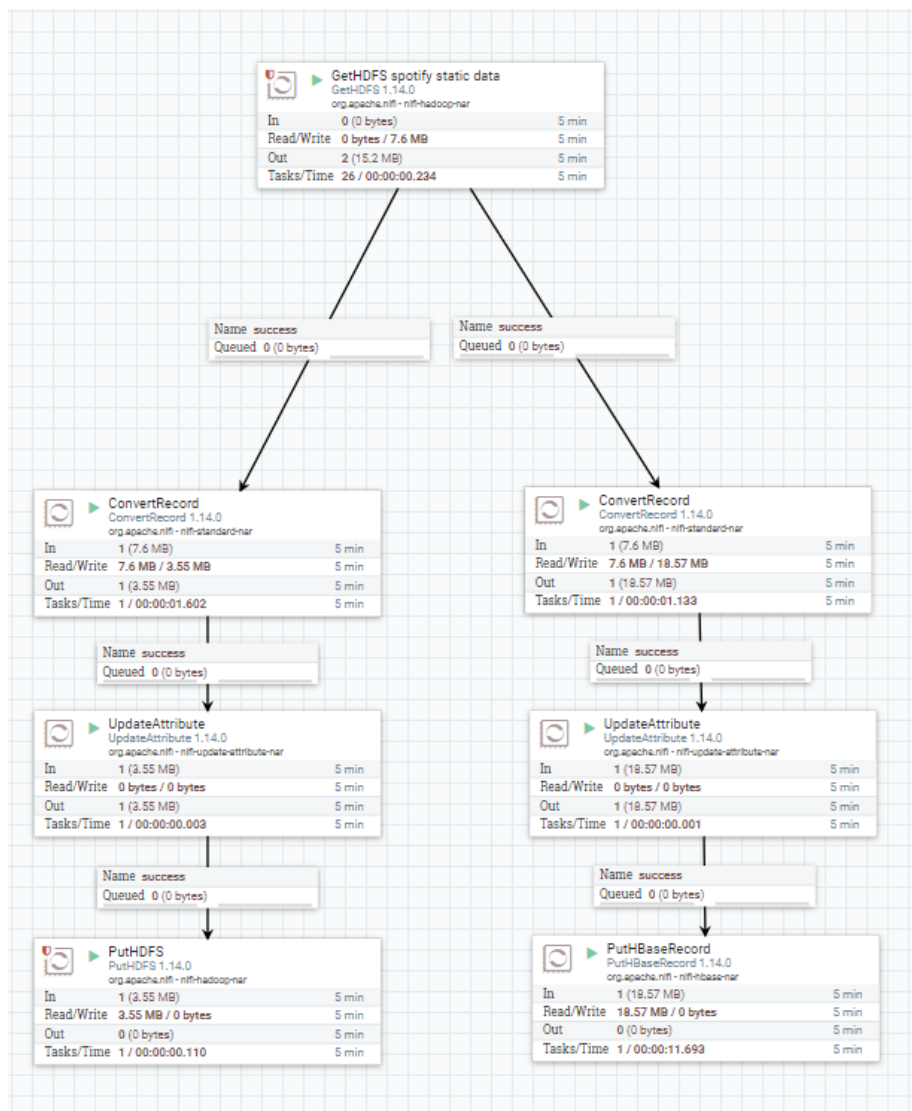
Test 4: Składowanie danych w HDFS

- cel: sprawdzenie, czy dane dodają się do katalogu `daily_data/` w HDFS.
- kroki: uruchomienie procesora PutHDFS.
- oczekiwany wynik: Pojawienie się nowego pliku `songs_2024-01-01.parquet` w HDFS.

```
vagrant@node1:~$ hdfs dfs -ls /user/miszalskan/project/nifi_out/daily_data/  
Found 1 items  
-rw-r--r--  1 root supergroup      5015 2024-01-01 21:30 /user/miszalskan/project/nifi_out/daily_data/songs_2024-01-01  
.parquet
```

Rysunek 9: Zawartość katalogu przed i po uruchomieniu procesora.

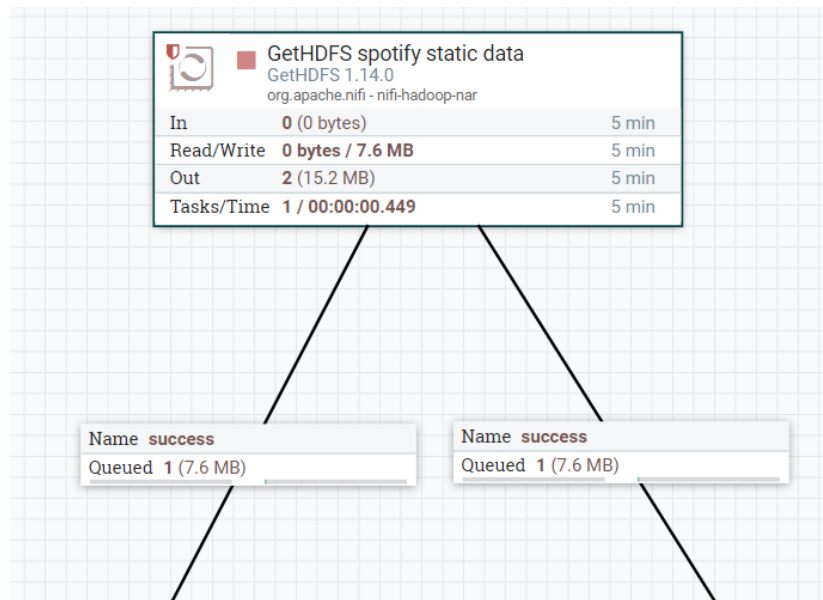
1.3 Składowanie danych statycznych w HDFS oraz HBase



Rysunek 10: Nifi Flow

Test 1: Pobranie pliku z katalogu /nifi_in/spotify z HDFS

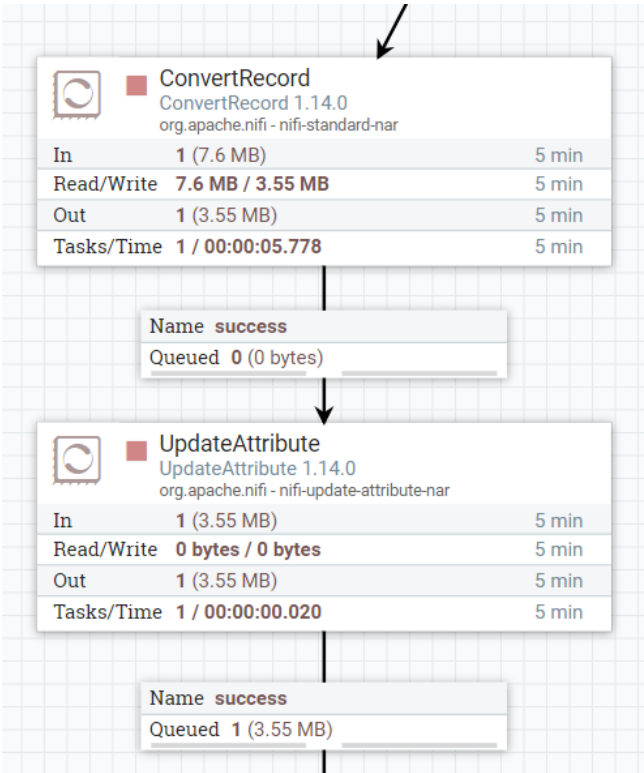
- cel: sprawdzenie, czy plik pobiera się poprawnie.
- kroki: uruchomienie procesora GetHDFS.
- oczekiwany wynik: dodanie do 2 kolejek plików



Rysunek 11: Kolejki po uruchomieniu procesora GetHDFS

Test 2: Zmiana rozszerzenia na parquet



- cel: sprawdzenie, czy zmiana rozszerzenia pliku na .parquet przebiega poprawnie..
- kroki: uruchomienie procesora ConvertRecord oraz UpdateAttribute.
- oczekiwany wynik: dodanie do kolejki pliku spotify_songs.parquet.



Rysunek 12: Nifi Flow.

success

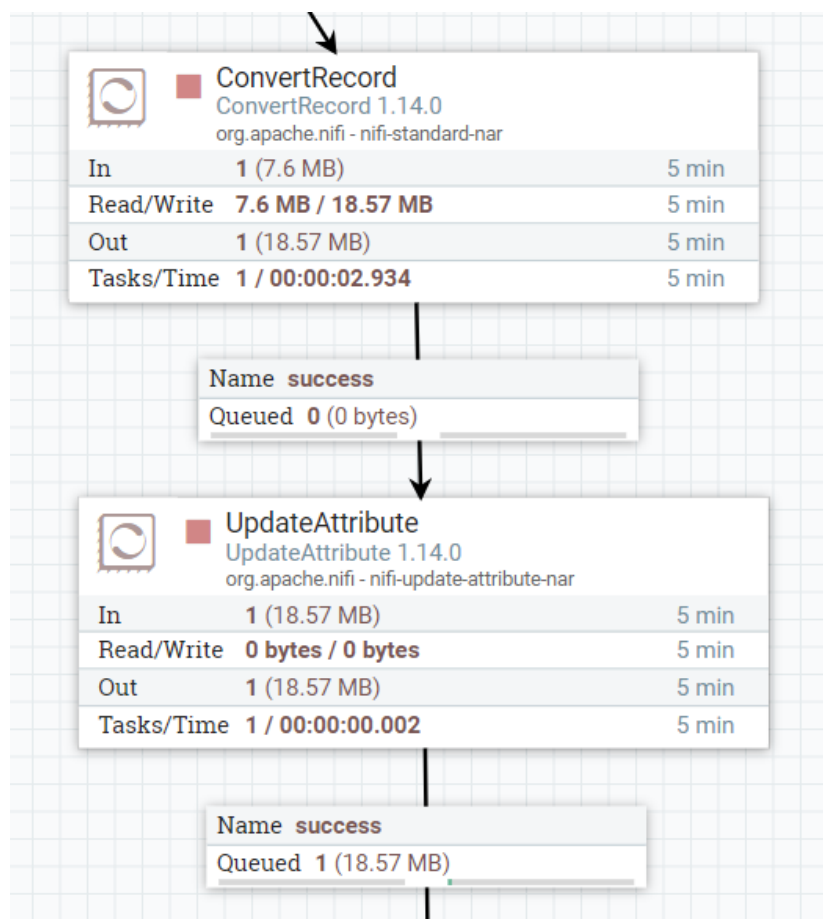
Displaying 1 of 1 (3.55 MB)

	Position	UUID	Filename	File Size	Queued Duration	Lineage Duration	Penalized	
0	1	bec007db-9819-435...	spotify_songs.parquet	3.55 MB	00:01:27.440	00:04:38.363	No	 

Rysunek 13: Zawartość kolejki po przejściu plik przez procesory.

Test 3: Zmiana rozszerzenia na json

- cel: sprawdzenie, czy zmiana rozszerzenia pliku na .json przebiega poprawnie..
- kroki: uruchomienie procesora ConvertRecord oraz UpdateAttribute.
- oczekiwany wynik: dodanie do kolejki pliku `spotify_songs.json`.



Rysunek 14: Nifi Flow.

success

Displaying 1 of 1 (18.57 MB)

	Position	UUID	Filename	File Size	Queued Duration	Lineage Duration	Penalized	
1	1	eb456fb9-b74c...	spotify_songs.json	18.57 MB	00:02:36.162	00:05:49.185	No	

Rysunek 15: Zawartość kolejki po przejściu plik przez procesory.

Test 4: Składowanie danych w HBase

- cel: sprawdzenie, czy dane poprawnie ładują się do tabeli w Hbase.
- kroki: uruchomienie procesora PutHBaseRecord.
- oczekiwany wynik: pojawienie się danych w tabeli `spotify_songs` w HBase.

```

hbase(main):004:0> scan 'spotify_songs'
ROW
COLUMN+CELL
0 row(s)
Took 0.1957 seconds
hbase(main):005:0> scan 'spotify_songs'
ROW
COLUMN+CELL
0017A6SjgTbfQVU2EtsPNo column=data:acousticness, timestamp=2024-01-07T13:57:54.457, value=0.279
0017A6SjgTbfQVU2EtsPNo column=data:danceability, timestamp=2024-01-07T13:57:54.457, value=0.682
0017A6SjgTbfQVU2EtsPNo column=data:duration_ms, timestamp=2024-01-07T13:57:54.457, value=235440
0017A6SjgTbfQVU2EtsPNo column=data:energy, timestamp=2024-01-07T13:57:54.457, value=0.401
0017A6SjgTbfQVU2EtsPNo column=data:instrumentalness, timestamp=2024-01-07T13:57:54.457, value=0.0117
0017A6SjgTbfQVU2EtsPNo column=data:key, timestamp=2024-01-07T13:57:54.457, value=2
0017A6SjgTbfQVU2EtsPNo column=data:liveness, timestamp=2024-01-07T13:57:54.457, value=0.0887
0017A6SjgTbfQVU2EtsPNo column=data:loudness, timestamp=2024-01-07T13:57:54.457, value=-10.068
0017A6SjgTbfQVU2EtsPNo column=data:mode, timestamp=2024-01-07T13:57:54.457, value=1
0017A6SjgTbfQVU2EtsPNo column=data:playlist_genre, timestamp=2024-01-07T13:57:54.457, value=rock
0017A6SjgTbfQVU2EtsPNo column=data:playlist_id, timestamp=2024-01-07T13:57:54.457, value=37i9dQZF1DWYDQ8wBxd7xt
0017A6SjgTbfQVU2EtsPNo column=data:playlist_name, timestamp=2024-01-07T13:57:54.457, value=Pinoy Classic Rock
0017A6SjgTbfQVU2EtsPNo column=data:playlist_subgenre, timestamp=2024-01-07T13:57:54.457, value=classic rock
0017A6SjgTbfQVU2EtsPNo column=data:speechiness, timestamp=2024-01-07T13:57:54.457, value=0.0236
0017A6SjgTbfQVU2EtsPNo column=data:tempo, timestamp=2024-01-07T13:57:54.457, value=97.091
0017A6SjgTbfQVU2EtsPNo column=data:track_album_id, timestamp=2024-01-07T13:57:54.457, value=1srJQ0nJEQgd8w4XSqI4JQ
0017A6SjgTbfQVU2EtsPNo column=data:track_album_name, timestamp=2024-01-07T13:57:54.457, value=Trip
0017A6SjgTbfQVU2EtsPNo column=data:track_album_release_date, timestamp=2024-01-07T13:57:54.457, value=2001-01-01
0017A6SjgTbfQVU2EtsPNo column=data:track_artist, timestamp=2024-01-07T13:57:54.457, value=Barbie's Cradle
0017A6SjgTbfQVU2EtsPNo column=data:track_name, timestamp=2024-01-07T13:57:54.457, value=Pangarap
0017A6SjgTbfQVU2EtsPNo column=data:track_popularity, timestamp=2024-01-07T13:57:54.457, value=41

```

Rysunek 16: Zawartość tabeli przed i po uruchomieniu procesora.

Test 5: Składowanie danych w HDFS

- cel: sprawdzenie, czy dane dodają się do katalogu `spotify` w HDFS.
- kroki: uruchomienie procesora PutHDFS.
- oczekiwany wynik: Pojawienie się nowego pliku `spotify_songs` w HDFS.

```
vagrant@node1:~$ hdfs dfs -ls /user/miszalskan/project/nifi_out/spotify/
vagrant@node1:~$
vagrant@node1:~$ hdfs dfs -ls /user/miszalskan/project/nifi_out/spotify/
Found 1 items
-rw-r--r--  1 root supergroup    3722730 2024-01-07 13:57 /user/miszalskan/project/nifi_out
/spotify/spotify_songs.parquet
```

Rysunek 17: Zawartość katalogu przed i po uruchomieniu procesora.