

Data\_pipeline is our entry point to our application, ingest.py is our data Source – to może być Hive lub cokolwiek innego. Data pipeline wywoła te 3 pliki w określonej sekwencji.

Driver tworzy spark session whhichc is entry point to each spark application (czyli spark session tworzę tylko raz w całej aplikacji a później ją re-use). Once you have spark session – you have access to all spark methods.

Config file

[loggers]  
keys=root,Ingest,Transform,Persist  
  
[handlers]  
keys=consoleHandler *# bo printujemy log do konsoli*[formatters]  
keys=sampleFormatter  
  
[logger\_root] *# mamy tutaj root level*level=DEBUG *# tutaj może być DEBUG, INFO i inne*handlers=consoleHandler  
  
[logger\_Ingest] *# robimy to for each python class  
# and each python class is in seperate file*level=INFO  
handlers=consoleHandler  
qualname=Ingest  
propagate=0 *# upewnia, że log jest drukowany tylko raz*[logger\_Transform]  
level=ERROR  
handlers=consoleHandler  
qualname=Transform  
propagate=0 *# i tak samo dla każdej klasy*[logger\_Persist]  
level=WARN  
handlers=consoleHandler  
qualname=Persist  
propagate=0  
  
[handler\_consoleHandler] *# oraz handler level - używając handler level  
# można kontrolować logging to different places*class=StreamHandler  
level=INFO *# przykładowo mogę mieć debug statement into console  
# but if you have another handler który drukuje do jakiegoś pliku lub wysyła  
# jako email - to mogę mieć też inne log level.*formatter=sampleFormatter  
args=(sys.stdout,)  
  
*# opdsumowując, hanlder will over write to co wpisane w logger\_root pod atrybutem level  
# pod warunkiem że level w handler ma wyższy podziom niż w logger\_root  
# jak mamy w logger\_root - leve=DEBUG i w handler\_consoleHandler - level = WARN  
# to printują się tylko WARN i ERROR messages*[formatter\_sampleFormatter] *# under what format log is printed*format=%(acctime)s - % (name)s - %(levelname)s - %(message)s