EMS & Central Data Service (Payer) Design Document

* BatchService
  1. Configuration File to control file to process and call producerService accordingly
* ProducerService

1. Batch service calls the REST end point to send the message
   1. Endpoint : <http://localhost:8085/kafka/APPLICATIONNAME>

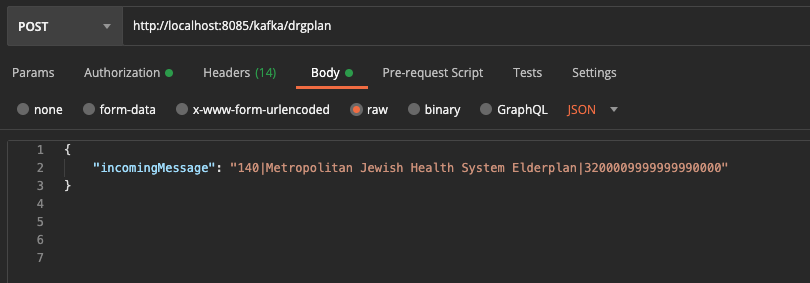
                                                              i.      Where APPLICATIONNAME Can be

* + - 1. DRGPAYER
      2. DRGPLAN
      3. NCPDP
      4. ANYNAME\_DEPENDING\_ON\_THE\_SERVICE \_*THAT\_WHAT*\_TO\_USE\_MECKESSON\_MESSAGING\_SERVICE

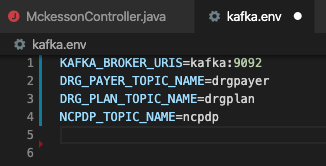
                                                            ii.      Sample END Points for BatchService

* + - 1. <http://localhost:8085/kafka/drgpayer>
      2. <http://localhost:8085/kafka/drgplan>
      3. <http://localhost:8085/kafka/ncpdp>
      4. <http://localhost:8085/kafka/xxxxxxxxxxxxx>

                                                          iii.      Sample json message



1. REST SERVICE :
   1. Maintains a set of Topic files in the kafka.env file

* Once the Service comes in it validates the application name in the URL (<http://localhost:8085/kafka/drgpayer>) with the list of accepted Topics in the kafka.env file.
* 

* If it’s one in the list of available topic it sends the message to kafka producer to the appropriate topic
* If it’s NOT in the list it will reject the message (Will not set to Producer )

* ConsumerService

1. CONSUMER
   1. In our Scenario, to keep it simple we can run with only one Consumer listening to all TOPICS, and distinguish the message based on TopicName

                                                              i.      Based on the Topic name the consumer can call the appropriate REST service of the Microservice which takes care of inserting to DB

* + - 1. Topic = DRGPAYER 🡺 call <http://localhost:8089/drgpayer/inserttodrgdatabase>
      2. Topic = DRGPAYER 🡺 call <http://localhost:8089/drgpayer/inserttodrgdatabase>
      3. Topic = ~~DRGPAYER~~ NDCPD🡺 call <http://localhost:8089/ncpdp/inserttoncpdbdatabase>
  1. DB - MySQL

Comment: (to be discussed)

* 1. One consume vs two ( we can make it one for now and if time allow , make it two) ,

                                                               i.      If we make it one, I still don’t want this consumer to be able to listen to all topic,

* + - 1. Need to make a noise topic to not be listened by the consumer
  1. Separated the consumer and service to insert into DB.

                                                               i.      For now, Jakeer can keep it as one, but separated as DBService etc… as needed.

* ProcessService (TBD) – Separated from Consumer??
* DataService
  + DB - Postgres