# TASK

We want you to do the following tasks:

1. Produce message and publish to Kafka topic (input\_topic):

You will have to write a producer that publishes events (json object) to kafka topic.

1. Consume (read) messages from topic (input topic), aggregate the messages by userId and produce a summary item to another kafka topic (output\_topic)

# IMPLEMENTATION

To publish JSON objects as messages to a kafka topic (input\_topic) in the following format (sample input):

{

"userId":"userid-1",

"type":"event",

"metadata":{

"messageId":"123sfdafas-32487239857dsh98234",

"sentAt":1534382478,

"timestamp":1534382478,

"receivedAt":0,

"apiKey":"apikey1",

"spaceId":"space1",

"version":"v1"

},

"event":"Played Movie",

"eventData":{

"MovieID":"MIM4ddd4"

}

}

The summary item should have following information (sample json) and should publish the json object to kafka topic (output\_topic):

{

"userId":"j11288090",

"firstSeen":1534382478,

"lastSeen":1534386588

}

You can install kafka to run kafka cluster locally or can use confluent docker image to run Kafka cluster and create kafka topics.

Kafka docker image and quickstart guide:

<https://docs.confluent.io/current/quickstart/ce-docker-quickstart.html>

Or

Kafka installation guide:

<https://kafka.apache.org/quickstart>

We would like to see following:

1. Your ability to publish messages to kafka
2. Ability to aggregate these messages in memory over the lifetime of the application and publish the result to a different kafka topic.
3. Code quality, documentation, unit test case and approach you take to solve this problem.

You should supply your answer to us in the form of a Git repository. If you’d like to keep it in a private repo on GitHub, that’s fine; just add us as collaborators. Or you can send us a blob of your code; use git bundle. Work in whatever programming language you want to. You will need to tell us how to build and run your code, however. Ideally this will be nothing more than stack build, or sbt test, or python producer.py, or ./configure && make && make install or whatever is blatantly idiomatic in your language of choice

Any questions please feel free to contact us