# Lab5-Task3: Guided Solution for Producing Messages to Kafka Topic

**Let's break down the code and explain each section:**

**Import Required Modules & Set Environment Variables**

```
import time
from pyspark.sql import SparkSession
from pyspark.sql import functions as F
from pyspark.sql import types as T
from kafka import KafkaProducer
```

Here, we're importing necessary modules and setting the correct environment paths for PySpark, Hadoop, and Python.

**Initialize SparkSession:**

```
spark = SparkSession.builder.master("local").appName('ex5_reviews_producer').getOrCreate()
```

This creates a new SparkSession with the name 'ex5_reviews_producer', running on a single local node.

**Load Data:**

```
# Load data from Parquet file into a DataFrame.
data_df = spark.read.parquet('s3a://spark/data/source/google_reviews/')
data_df.show(6)
```

Here, the processed Google reviews data is loaded from a Parquet file into a DataFrame and displayed.

**Convert Data to JSON:**

```
#Convert the DataFrame records to JSON format.
data = data_df.toJSON()
print(data.take(6))
```

Each record in the DataFrame is converted to a JSON string, which is a suitable format for sending messages to a Kafka topic. A sample of six JSON records is printed.

**Initialize Kafka Producer**:

```
#Set up a Kafka producer.
producer = KafkaProducer(bootstrap_servers='course-kafka:9092', value_serializer=lambda v: v.encode('utf-8'))
```

A KafkaProducer instance is initialized to send messages to the Kafka cluster. The producer will serialize the messages (JSON strings) into bytes using UTF-8 encoding.

**Produce Messages to Kafka Topic**:

```
i = 0

for json_data in data.collect():
    i = i + 1
    producer.send(topic='gps-user-review-source', value=json_data)
    if i == 50:
        producer.flush()
        time.sleep(5)
        i = 0
```

In this loop, each JSON string is sent as a message to the **DAVID_TEST_9** Kafka topic. After sending 50 messages, the producer buffers are flushed, ensuring that all messages are sent.

Then the script waits for 10 seconds before resuming. This pattern is used to space out the message production, sending 50 messages every 10 seconds.

**Close Kafka Producer & Terminate SparkSession**:

```
producer.close()
spark.stop()
```

After all messages have been sent, the Kafka producer is closed to release resources, followed by terminating the SparkSession.

**Summery**

This solution reads the processed Google Reviews data, converts it to JSON, and then produces the messages to a Kafka topic in batches of 50, with a 10-second interval between batches.

**Full code solution**

```
import time
from pyspark.sql import SparkSession
from pyspark.sql import functions as F
from pyspark.sql import types as T
from kafka import KafkaProducer

spark = SparkSession.builder.master("local").appName('ex5_reviews_producer').getOrCreate()

data_df = spark.read.parquet('s3a://spark/data/source/google_reviews')

data = data_df.toJSON()

producer = KafkaProducer(bootstrap_servers='course-kafka:9092', value_serializer=lambda v: v.encode('utf-8'))

i = 0

for json_data in data.collect():
    i = i + 1
    producer.send(topic='gps-user-review-source', value=json_data)
    if i == 50:
        producer.flush()
        time.sleep(5)
        i = 0

producer.close()

spark.stop()
```