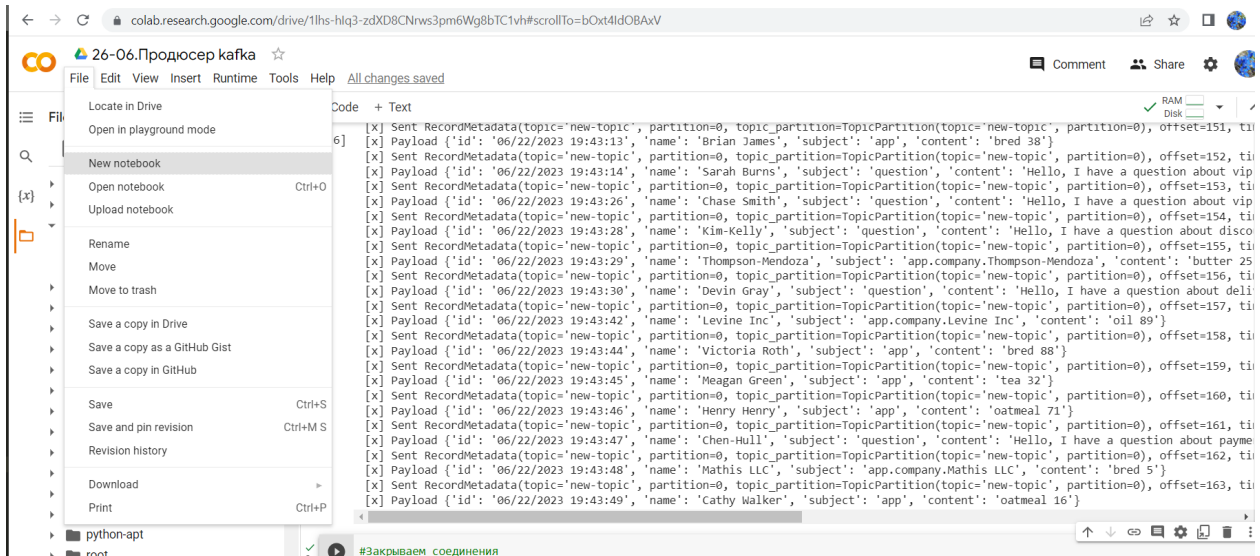


# Практическая работа

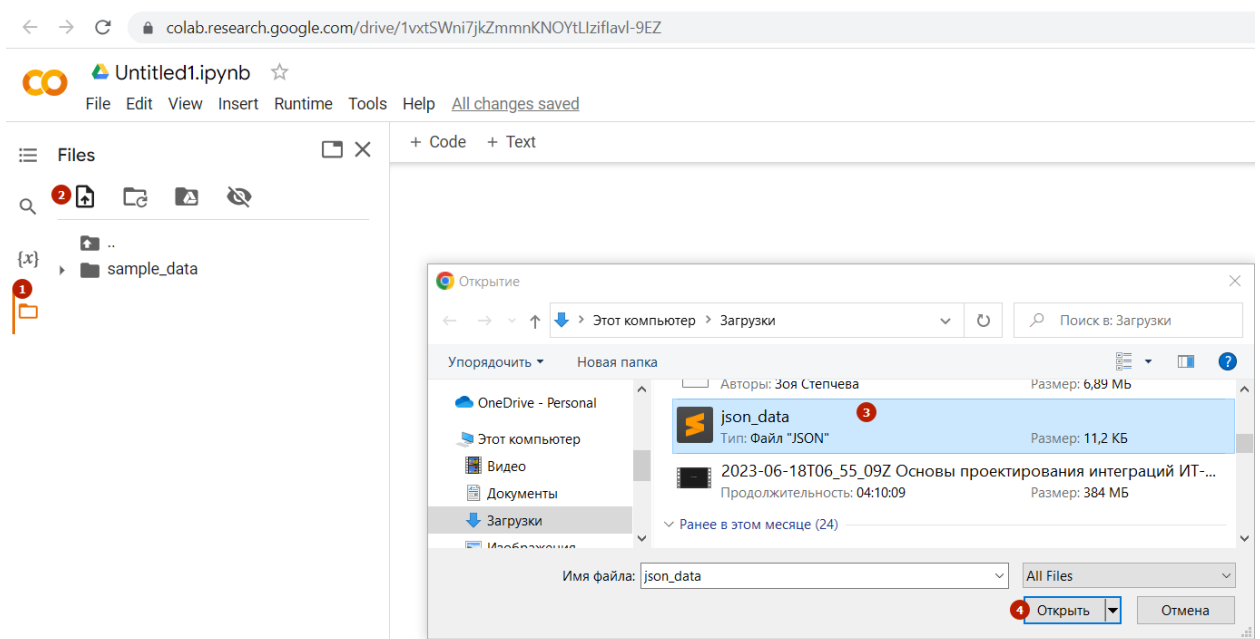
## Публикация и чтение сообщений из kafka

### Часть 1. Создание скрипта продюсера

#### 1. Откройте новый ноутбук google colab (File->New notebook):

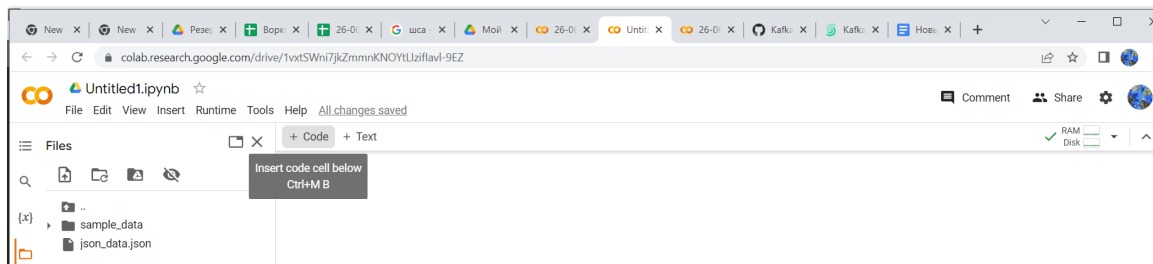


#### 2. Добавьте файл json\_data.json с сообщениями:

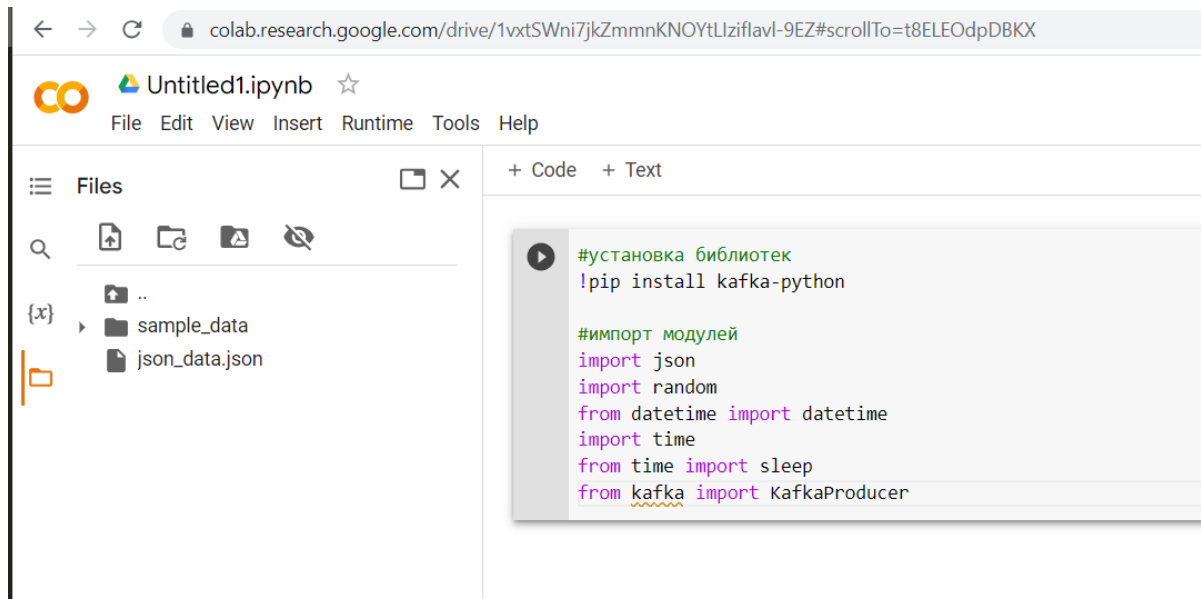


Файл с исходными данными для брокера kafka при этом будет доступен по адресу  
/content/json\_data.json

#### 3. Добавьте новую ячейку 1 с кодом:



4. Добавьте **ячейку 1** код для установки библиотек и импорта модулей:



5. Запустите код ячейки на выполнение, и оцените результат:




6. Создайте новую **ячейку 2**. Добавьте в **ячейку 2** код для подключения к брокеру kafka сервиса <https://console.upstash.com/kafka>. Добавьте пароль для доступа к кластеру:

com/kafka/87c5bfb-70eb-480a-bf72-5f798e903119?tab=details

## 1 SIO-1204-cluster

Free Tier · Single Replica · 10,000 messages per day


Credentials: Default 


[Details](#) [Usage](#) [Topics](#) [Credentials](#) [Connectors](#) **NEW**

**2**

Region	Endpoint	Username	<b>5</b> Password
eu-west-1	striking-snipe-11469-eu1-kafka.upstash.io:9092	*****	*****

### Connect to your cluster **3**

[Java](#) [Node](#) **[Python](#)** [Go](#) [CSharp](#) [Properties](#) [Cloudflare Workers](#) 

Library: [kafka-python](#) 


**4** **Producer**

```
1 from kafka import KafkaProducer
2
3 producer = KafkaProducer(
4     bootstrap_servers=['striking-snipe-11469-eu1-kafka.upstash.io:9092'],
5     sasl_mechanism='SCRAM-SHA-256',
6     security_protocol='SASL_SSL',
7     sasl_plain_username='c3RyaWtpbmctc25pcGUtMTE0NjkkKjipBit7uF-RUpZHBimX87sz0v1chKwCHEU',
8     sasl_plain_password='*****',
9 )
```

Добавьте в параметры подключения средства сериализации:


`value_serializer=lambda v: json.dumps(v).encode('utf-8')`

Оцените результат:

```
 # объявление продюсера Kafka и параметров подключения
producer = KafkaProducer(
    bootstrap_servers=['striking-snipe-11469-eu1-kafka.upstash.io:9092'],
    sasl_mechanism='SCRAM-SHA-256',
    security_protocol='SASL_SSL',
    sasl_plain_username='c3RyaWtpbmctc25pcGUtMTE0NjkkKjipBit7uF-RUpZHBimX87sz0v1chKwCHEU',
    sasl_plain_password='*****',
    value_serializer=lambda v: json.dumps(v).encode('utf-8')
)
```

Заполните оставшуюся часть ячейки кодом для отправки сообщений.

- Создайте **ячейку 3** с кодом. Внесите в нее данные для закрытия соединения с брокером:

 **0s** **[33]** **#Закрываем соединения**  
`producer.close()`

- Запустите скрипт на выполнение. Оцените результат.
- Прервите выполнение скрипта в ячейке 2. Выполните код ячейки 3 для закрытия соединения с брокером.

## Часть 2. Создание скрипта консьюмера

1. Создайте новый ноутбук для скрипта консьюмера.
2. В ячейку 1 добавьте импорт библиотек:

```
+ Code + Text

[1] #установка библиотек
!pip install kafka-python

import json
import random

from kafka import KafkaConsumer
from json import loads
from kafka.structs import TopicPartition
```

3. В ячейку 2 добавьте данные для подключения к брокеру, с указанием топика и необходимых партиций:

26-06.Консумер kafka ☆

File Edit View Insert Runtime Tools Help All changes saved

```
+ Code + Text

[21] #объявление потребителя Kafka
consumer = KafkaConsumer(
    bootstrap_servers=['striking-snipe-11469-eu1-kafka.upstash.io:9092'],
    sasl_mechanism='SCRAM-SHA-256',
    security_protocol='SASL_SSL',
    sasl_plain_username='c3RyaWtpbmctc25pcGUtMTE0NjkkKjIPBit7uF-RUPZHBimX87sz0vlchkwCHEU',
    sasl_plain_password='56cc5507a847888c9809f586c8210f',
    group_id='1',
    auto_offset_reset='earliest',
    enable_auto_commit=True
)

# подписка потребителя на определенный раздел topic partition
topic='myNewTopic'
topic_partition_0 = TopicPartition(topic, 0) # указываем имя топика и номер раздела: #все вопросы записывать в раздел 0
topic_partition_1 = TopicPartition(topic, 1) # указываем имя топика и номер раздела: #все корпоративные заявки записывать в раздел 1
topic_partition_2 = TopicPartition(topic, 2) # указываем имя топика и номер раздела: #заявки от частных лиц записывать в раздел 2
consumer.assign([topic_partition_0, topic_partition_1, topic_partition_2])
```

4. В ячейку 3 добавьте код для чтения сообщений из брокера, в ячейку 4 – код для закрытия соединения.

```
+ Code + Text

for message in consumer:
    payload=message.value.decode("utf-8")
    data=json.loads(payload)

    #вывод исходных данных в консоль Goggle Colab
    print(f' [x] Partition {message.partition}')
    print(data)

[25] #отписываем потребителя и закрываем соединение
consumer.unsubscribe()
consumer.close()
```

Запустите последовательно код продюсера и код консьюмера. Оцените результат.

Самостоятельное задание: Измените скрипт так, чтобы читать из брокера только сообщения с вопросами.

### Часть 3. Отправка данных консьюмером в google-таблицы

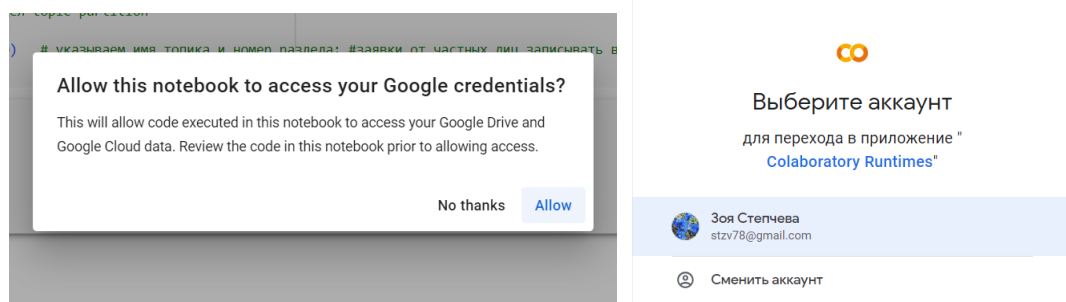
1. Создайте новый ноутбук. Скопируйте в **ячейку 1** код из предыдущего ноутбука консьюмера.
2. Создайте новую google-таблицу, сохраните ее идентификатор (указан в адресной строке). Например, для адреса таблицы [https://docs.google.com/spreadsheets/d/1KQ\\_7h0NTvkadNqEHTOmZTWozauplPIdKx6rdUr-7mW8/edit#gid=0](https://docs.google.com/spreadsheets/d/1KQ_7h0NTvkadNqEHTOmZTWozauplPIdKx6rdUr-7mW8/edit#gid=0) ее идентификатор: 1KQ\_7h0NTvkadNqEHTOmZTWozauplPIdKx6rdUr-7mW8, измените название **Листа 1** google-таблицы на `topic_partition_0`
3. В **ячейку 2** добавьте импорт модуля авторизации для google-таблиц и код для подключения созданной на шаге 2 таблицы по ее идентификатору:

```
#импорт модулей для работы с google-таблицами
from google.colab import auth
auth.authenticate_user()
import gspread
from google.auth import default
creds, _ = default()

##### соединение с google-таблицами
#Google Sheets Autentificate
googleClient = gspread.authorize(creds)

#Открытие заранее созданного файла Гугл-таблицы по идентификатору
googleTableId = '1KQ_7h0NTvkadNqEHTOmZTWozauplPIdKx6rdUr-7mW8'
table = googleClient.open_by_key(googleTableId)
workSheet = table.worksheet("topic_partition_0") #в какой лист гугл-таблиц будем записывать данные
```

4. Запустите последовательно ячейки 1-2.  
При запуске ячейки 2 укажите разрешения модулю использовать ваши учетные данные google-аккаунта для работы с google-таблицами:



Укажите, какой аккаунт для этого использовать.

5. Скопируйте в **ячейку 3** следующий код для подключения к разделу брокера kafka из ноутбука консьюмера предыдущей части практической работы:

```

#объявление потребителя Kafka
consumer = KafkaConsumer(
    bootstrap_servers=['striking-snipe-11469-eu1-kafka.upstash.io:9092'],
    sasl_mechanism = 'SCRAM-SHA-256',
    security_protocol = 'SASL_SSL',
    sasl_plain_username = 'c3RyaWtpbmctc25pcGUTMTE0NjkkKjipBit7uF-RUPZHBimX87sz0v1chKwCHEU',
    sasl_plain_password = '9bccd3067a84408dbc5883f5d6c8218f',
    group_id = '1',
    auto_offset_reset = 'earliest',
    enable_auto_commit = True
)

# подписка потребителя на определенный раздел topic partition
topic='myNewTopic'
topic_partition_0 = TopicPartition(topic, 0) # указываем имя топика и номер раздела: #заявки от
consumer.assign([topic_partition_0])

```

Добавьте в текущую **ячейку 3** код обработки чтения сообщений из топика kafka и записи в лист google-таблицы:

+ Code + Text

```

consumer.assign([topic_partition_0])

#начальный номер строки для записи данных в лист гугл-таблиц
tableRow = 1

#считывание из топика Kafka
for message in consumer:
    payload = message.value.decode("utf-8")
    data = json.loads(payload)

    #вывод исходных данных в консоль Goggle Colab
    print(data)

    #парсинг JSON-сообщения полезной нагрузки
    id = data['id']
    name = data['name']
    subject = data['subject']
    content = data['content']
    created_at = data['created_at']

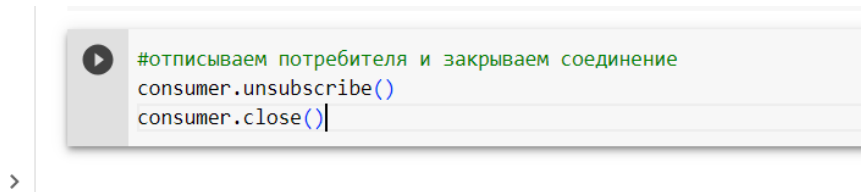
    #обновление данных в в Гугл-таблице

    #переход на следующую строку в гугл-таблицах
    tableRow = tableRow + 1

    #запись данных в ячейки гугл-таблицы
    workSheet.update_cell(tableRow, 1, id)
    workSheet.update_cell(tableRow, 2, name)
    workSheet.update_cell(tableRow, 3, content)
    workSheet.update_cell(tableRow, 4, created_at)

```

6. Добавьте в **ячейку 4** код для закрытия соединения с kafka:

A code editor snippet with a light gray background. On the left, there is a vertical line and a small gray square containing a black play button icon. To the right of the icon, the text "#отписываем потребителя и закрываем соединение" is written in green. Below this, the code "consumer.unsubscribe()" and "consumer.close()" are written in black, with a blue cursor at the end of the second line.

```
#отписываем потребителя и закрываем соединение
consumer.unsubscribe()
consumer.close()
```

Для проверки скрипта запустите код с ноутбука продюсера, а затем – код текущего консьюмера. Оцените результат работы скриптов, перейдя в лист google-таблицы.

Самостоятельное задание: измените скрипт так, чтобы считать сообщения заявки от физических лиц - в отдельную вкладку google-таблицы.