

# Содержание

- Общие сведения об API сервисов Google** ..... 3
- Получение доступа к API через Google Cloud Console** ..... 4
  - Создание «проекта» в Google Cloud Console ..... 4
  - Подключение к «проекту» библиотеки API Google Sheets ..... 4
  - Создание сервисного аккаунта Google для доступа к API «проекта» ..... 6
  - Создание ключа доступа к сервисному аккаунту ..... 7
- Настройка доступа в Google-таблице** ..... 9
- Примеры программного кода** ..... 10
  - Общие сведения о примерах ..... 10
  - Подготовка LINQ-запроса в ADVANTA ..... 11



# Экспорт данных из ADVANTA в Google-таблицы

Google предоставляет множество сервисов для совместной работы с документами распространённых офисных форматов. В их числе - **Google Sheets**, облачный редактор электронных таблиц, аналогичный Excel.

В некоторых ситуациях требуется переносить данные о проектах, сформированных в ADVANTA, в таблицы Google Sheets.

Если передавать данные требуется регулярно, можно автоматизировать этот процесс. Для этого можно написать приложение, которое будет получать данные из ADVANTA через ее API и записывать их в электронную таблицу через API системы Google Sheets.

Шаги, которые надо выполнить для автоматизации передачи информации из ADVANTA в таблицы Google Sheets:

1. Зарегистрировать «сервисный аккаунт» в системе «Google Cloud Console» для вызова к API.
2. Настроить доступ к файлу электронной таблицы в Google Sheets для связанной с «сервисным аккаунтом» учётной записи.

Также ниже рассмотрены примеры двух максимально простых приложений на C#, которые получают данные из ADVANTA и записывают их в таблицы Google. На основе этих примеров Вы можете создавать собственные приложения для решения задач по автоматизации передачи данных из ADVANTA в облачные сервисы Google.

## Общие сведения об API сервисов Google

Доступ к API-функциям облачных сервисов Google настраивается в системе [Google Cloud Console](#) - это веб-сервис для разработчиков и системных администраторов организаций. Требуется создать в [Google Cloud Console](#) «проект» и подключить в него библиотеки API нужных сервисов Google.

При вызове API программа должна проходить авторизацию в «проекте». Есть несколько способов её организовать в зависимости от того, каким именно образом программа будет использоваться:

1. Если предполагается создание приложения с визуальным интерфейсом, в котором пользователи будут работать через персональные учётные записи Google, следует настроить доступ к «проекту» через протокол **OAuth**. При вызове API из такого приложения, будет открываться веб-страница авторизации Google, где пользователь должен ввести свои логин и пароль вручную.
2. Для приложений, работающих в автоматическом режиме, требуется создать специальную учётную запись Google - «**сервисный аккаунт**». Ключ доступа к такой учётной записи экспортируется в файл и включается в код программы.

Непосредственный вызов API Google может быть выполнен либо с помощью библиотек, либо напрямую, через REST API. В первом случае код намного проще, но библиотеки надо устанавливать в проект в среде разработки.

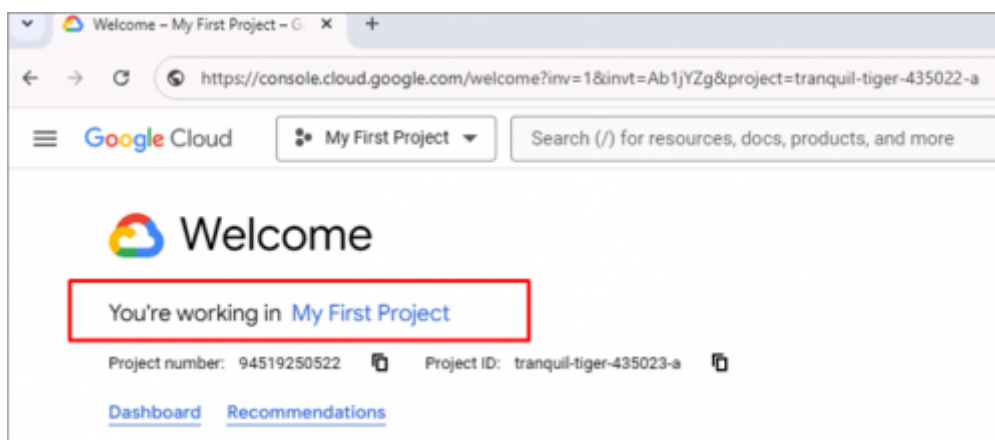
## Получение доступа к API через Google Cloud Console

Для вызова API-функций Google-таблиц требуется выполнить шаги:

1. Создать в сервисе [Google Cloud Console](#) новый проект.
2. Подключить к этому проекту библиотеку API-функций «Google Sheets».
3. Создать сервисный аккаунт – специальную учётную запись, которую можно использовать лишь для идентификации приложений.
4. Создать ключ доступа к сервисному аккаунту - файл формата JSON, который содержит данные для автоматической авторизации приложения.

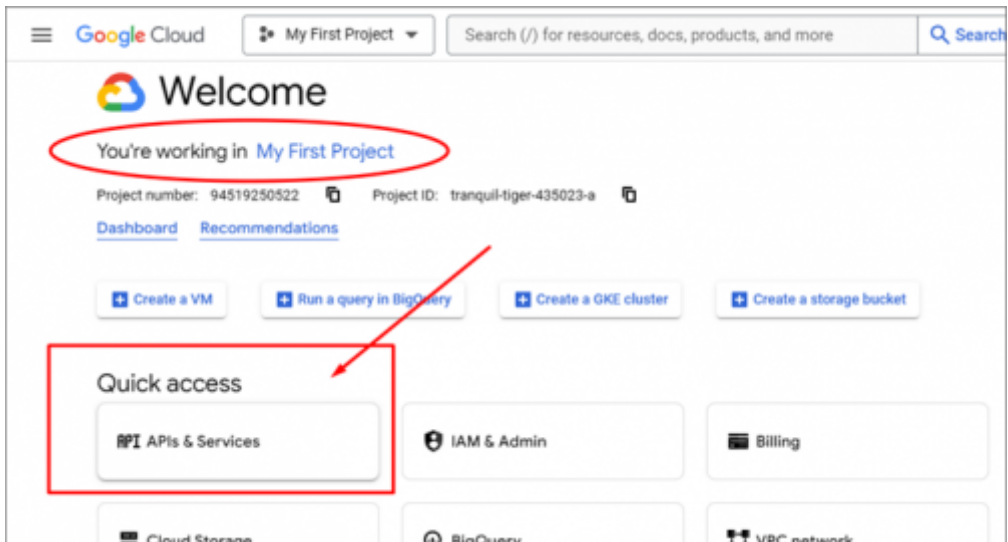
### Создание «проекта» в Google Cloud Console

Перейти на сайт [Google Cloud Console](#). При первом входе система автоматически создаст новый «проект» с названием «My First Project».

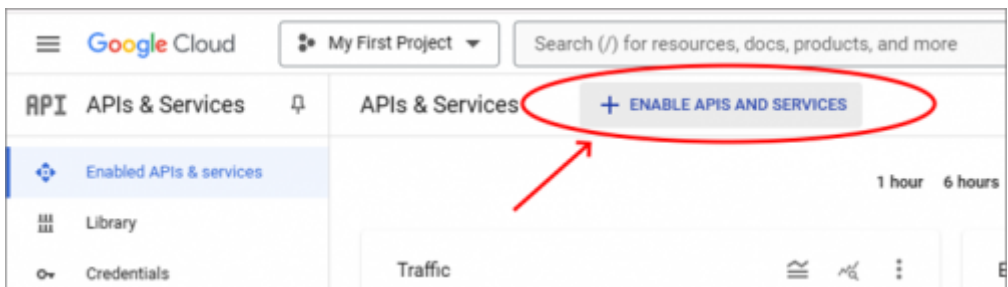


### Подключение к «проекту» библиотеки API Google Sheets

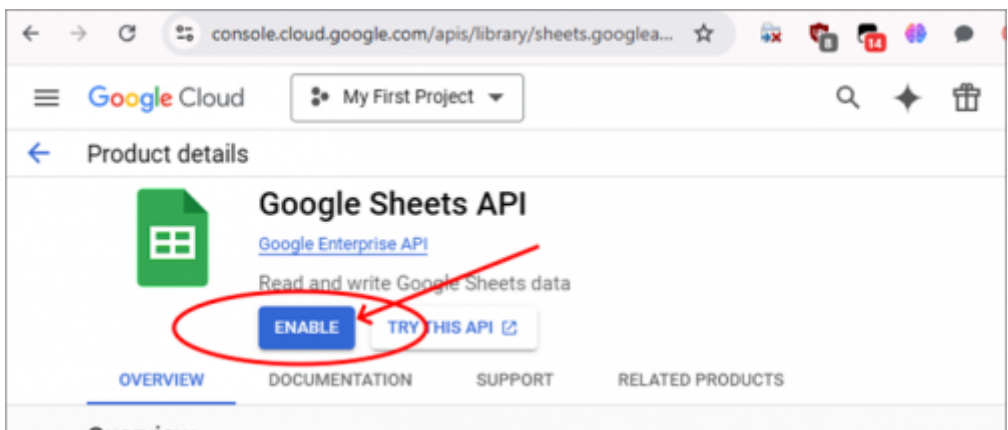
Для подключения к созданному проекту библиотеки API-функций Google Sheets необходимо перейти в раздел «APIs & Services» по кнопке под названием проекта.



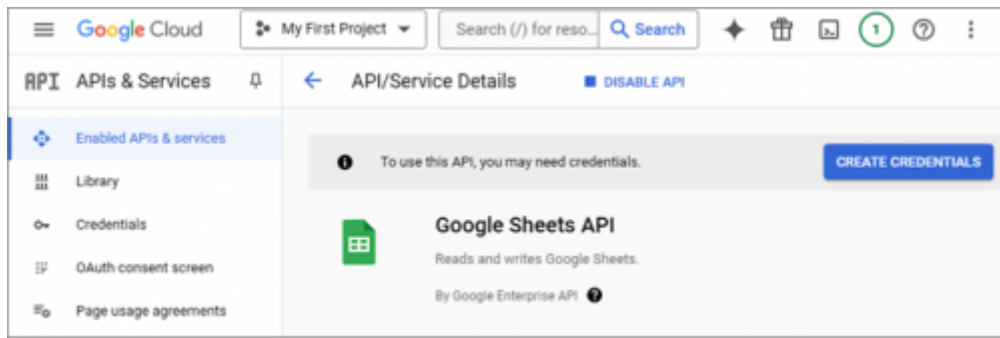
Затем необходимо открыть список библиотек кнопкой «Enable APIs and services».



В поле поиска ввести «Sheets», нажать **Enter** и выбрать найденную библиотеку «Google Sheets API». Включить библиотеку кнопкой **Enable**.

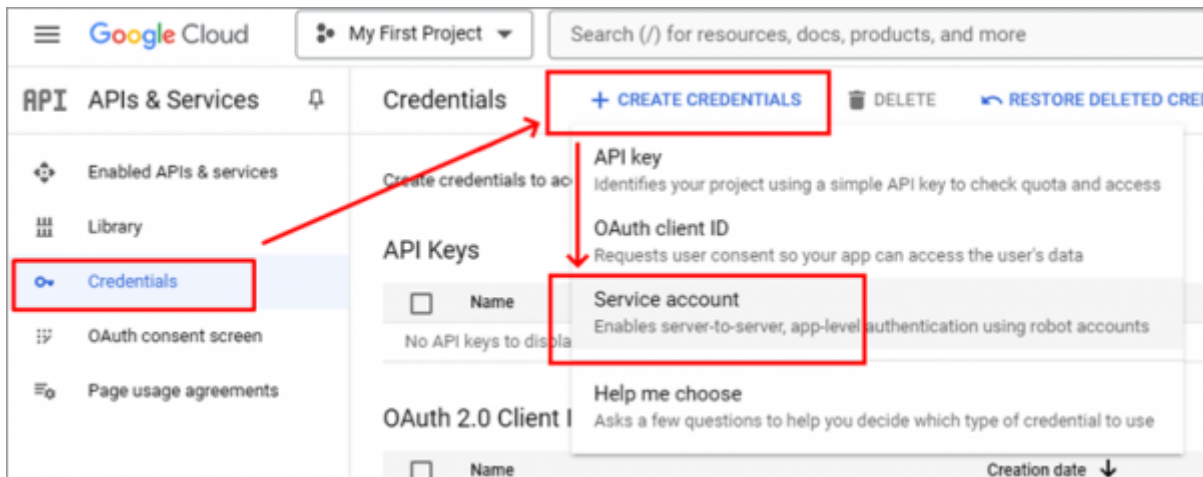


Библиотека появится в проекте, на странице «Enabled APIs & services».

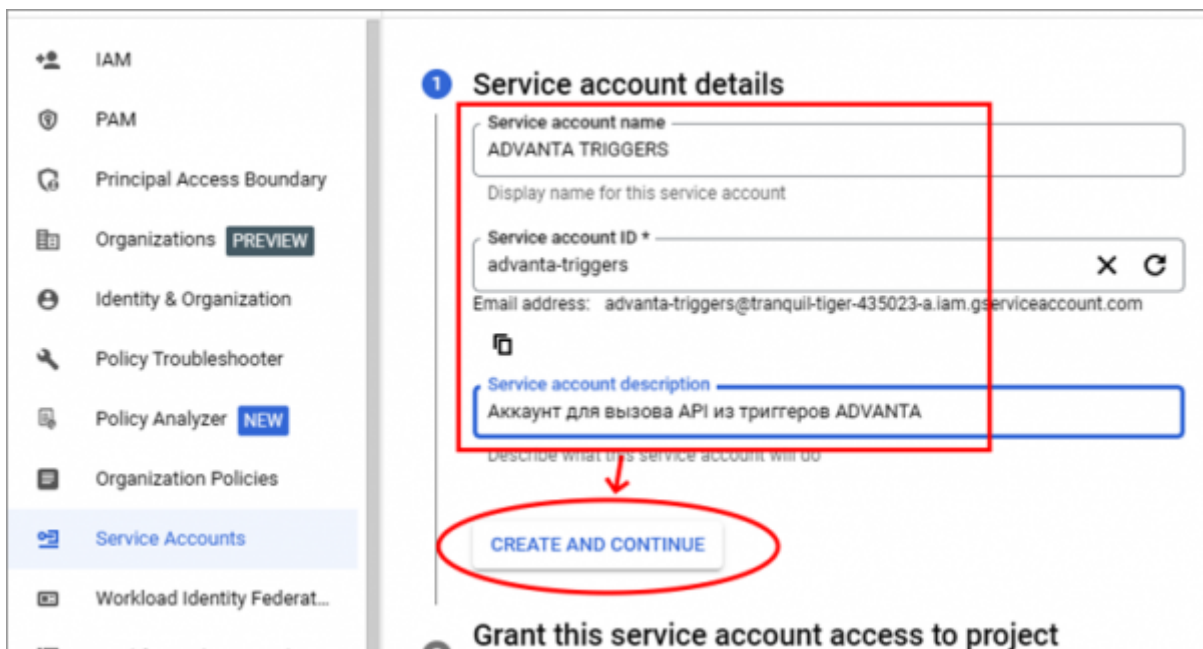


## Создание сервисного аккаунта Google для доступа к API «проекта»

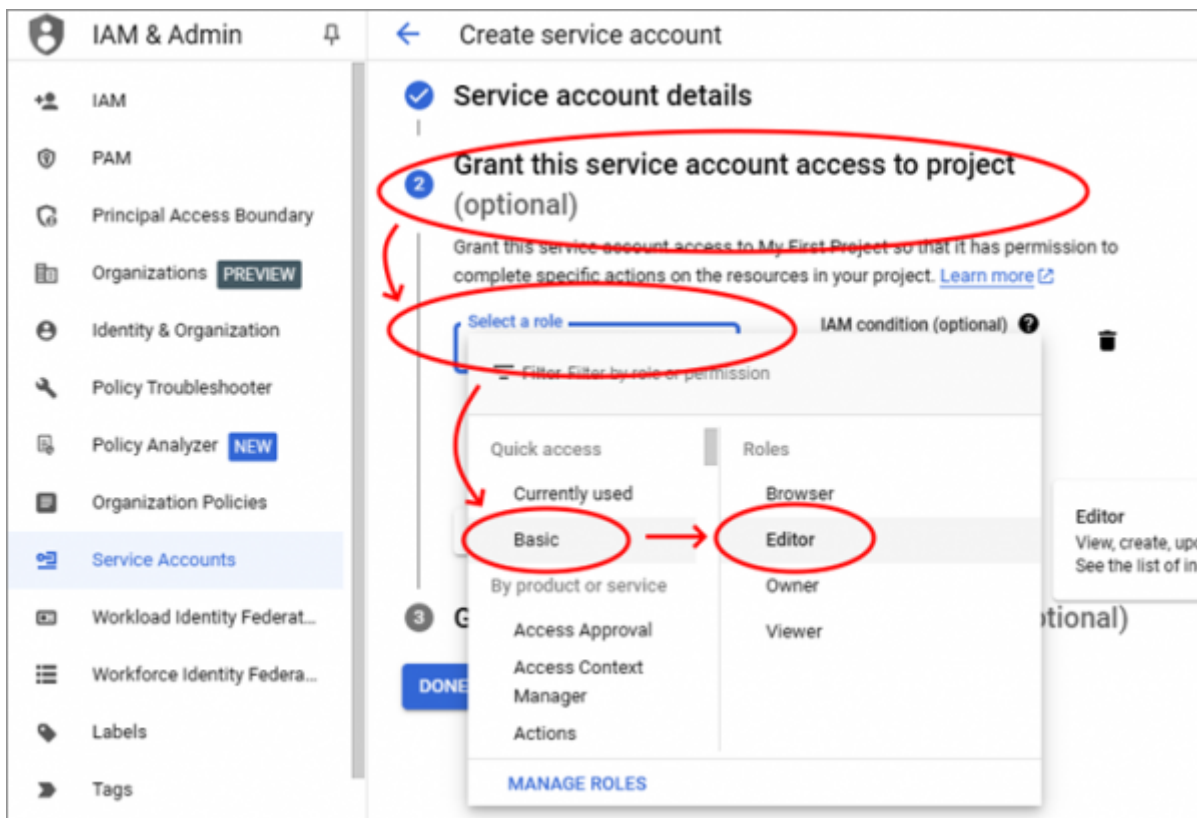
Для создания сервисного аккаунта Google необходимо в разделе «Credentials» вызвать меню «Create credentials > Service account».



Далее на странице «Create service account» ввести название аккаунта в поле «Service account name» и описание в поле «Service account description». Система автоматически заполнит поле «Service account ID». Для продолжения настройки нажать «Create and continue».



На странице «Grant this service account access to project» в выпадающем списке «Role» выбрать роль «Basic / Editor».



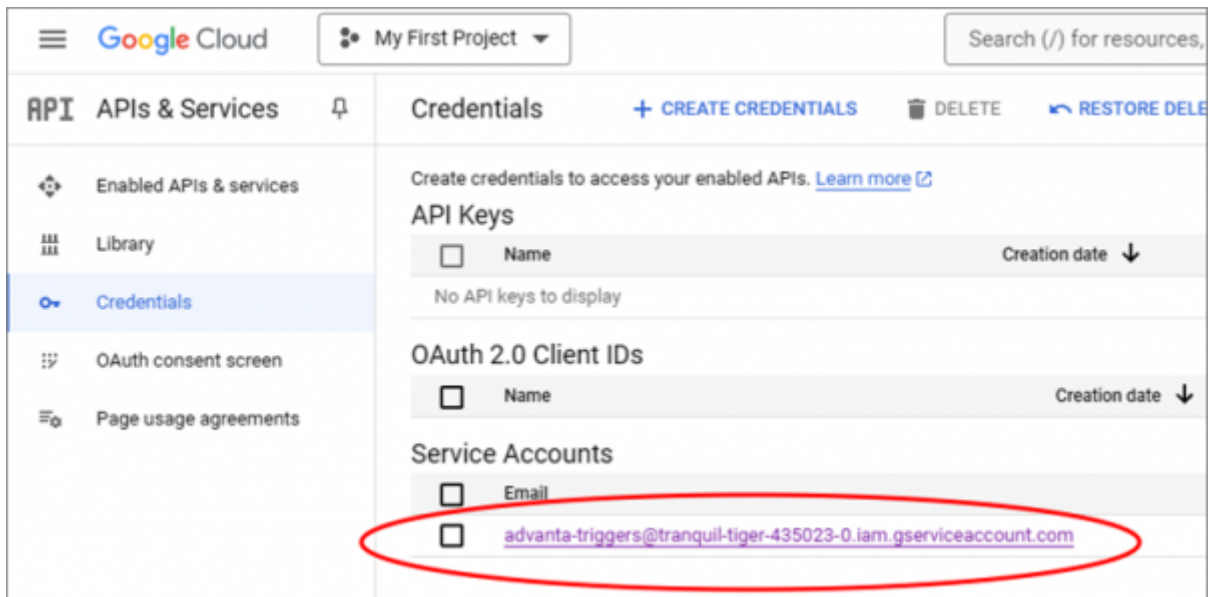
Для перехода к следующему шагу нажать **Continue**.

В разделе «Grant users access to this service account» ничего вводить не нужно. Нажать **Done**, чтобы завершить создание сервисного аккаунта. Сервисный аккаунт будет создан и появится в поле «Service account». Адрес электронной почты этого сервисного аккаунта нужно будет указать далее при настройке доступа к Google-таблице.

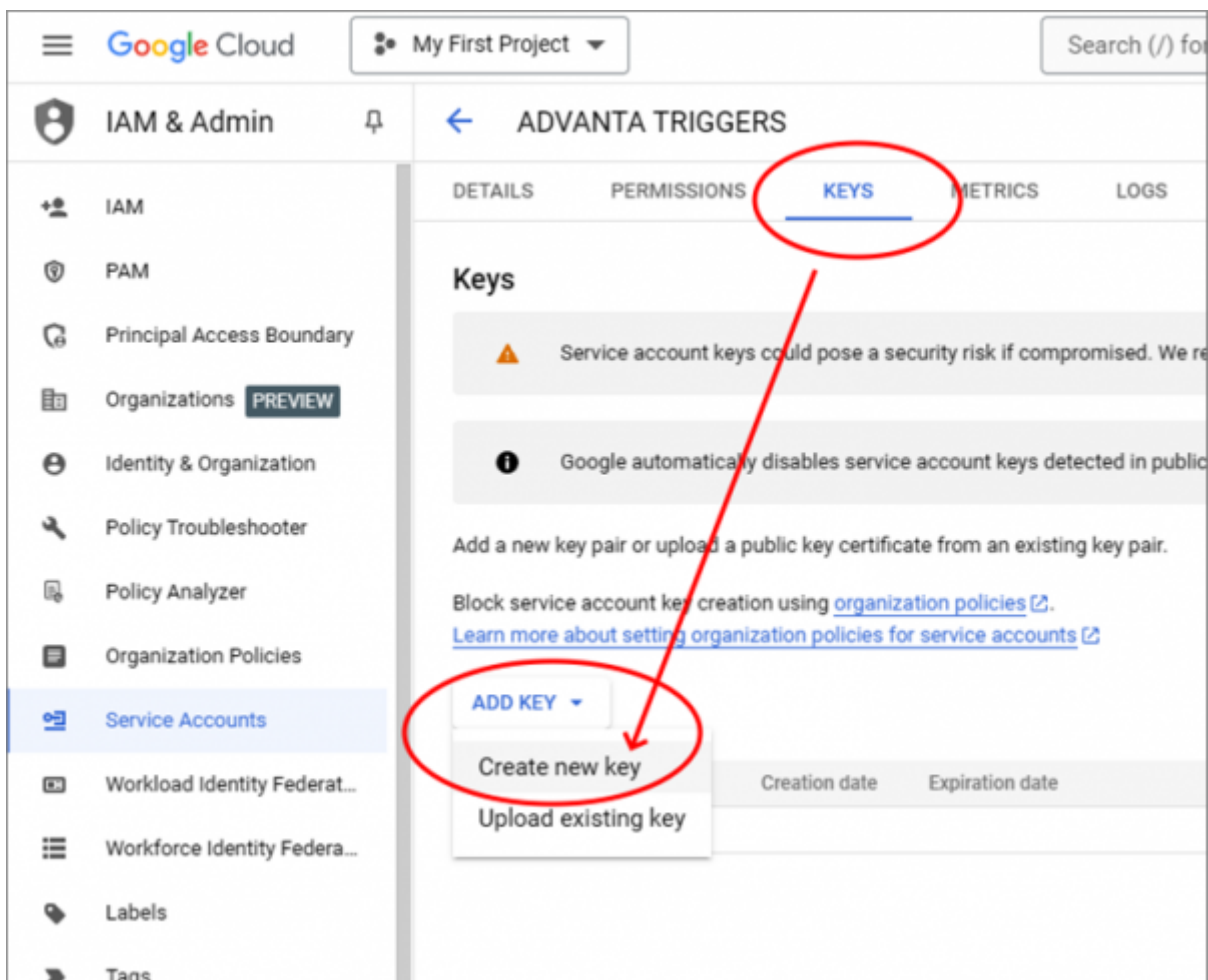
### Создание ключа доступа к сервисному аккаунту

Теперь нужно создать ключ для доступа к сервисному аккаунту из программы, которую будем разрабатывать.

Перейти по ссылке с именем аккаунта в разделе «Service Accounts»:



На странице настроек аккаунта нужно перейти на вкладку «Keys» и вызвать команду создания нового ключа через меню «Add key / Create new key».



По запросу о типа ключа выбрать JSON и нажать **Create**. Появится сообщение, что ключ сохранён на компьютере и одновременно с этим начнётся скачивание файла.

Полученный файл нужно сохранить на диске и изменить его название на «service-account.json».

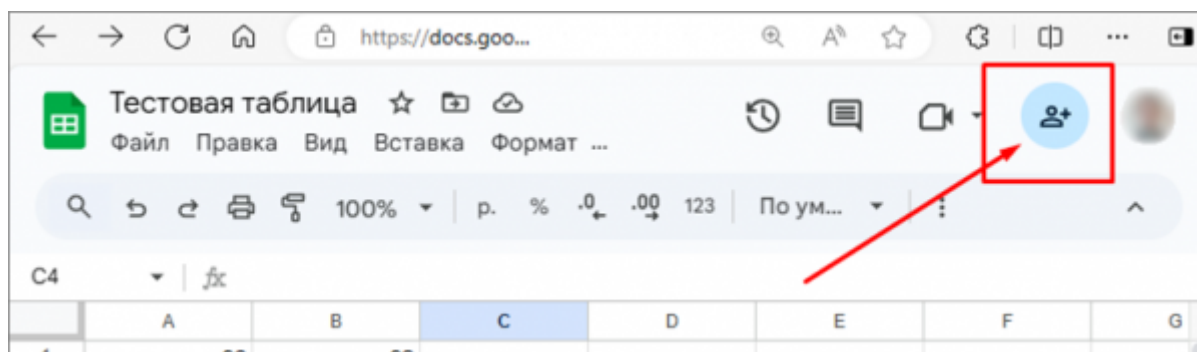
Далее он потребуется при создании кода приложения. Пример структуры файла «service-account.json»:

```
{  
  "type": "service_account",  
  "project_id": "tranquil-tiger-435022-a0",  
  "private_key_id": "52b77cc59e50573be4a2a573e4bf3d86392fcaf0",  
  "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEVwIBADANBgkqhkiG9w0BAQEFAASCBKkwwggS\n",  
  "client_email": "advanta-triggers@tranquil-tiger-435022-a0.iam.gserviceaccount.com",  
  "client_id": "106608638608969228656",  
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",  
  "token_uri": "https://oauth2.googleapis.com/token",  
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",  
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/advanta-t\n",  
  "universe_domain": "googleapis.com"  
}
```

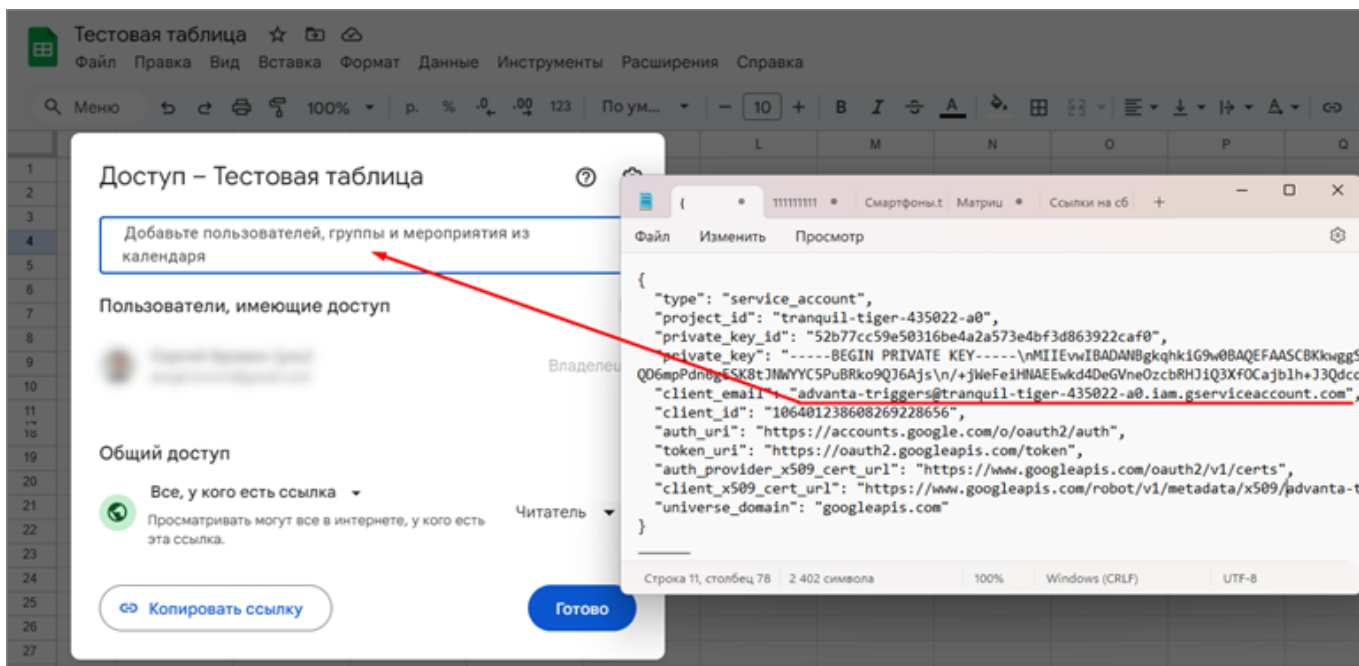
В реальных проектах важно обеспечить ограниченный доступ к файлу ключа и по возможности шифровать. Если ключ попадёт в поисковые системы, Google автоматически заблокирует его.

## Настройка доступа в Google-таблице

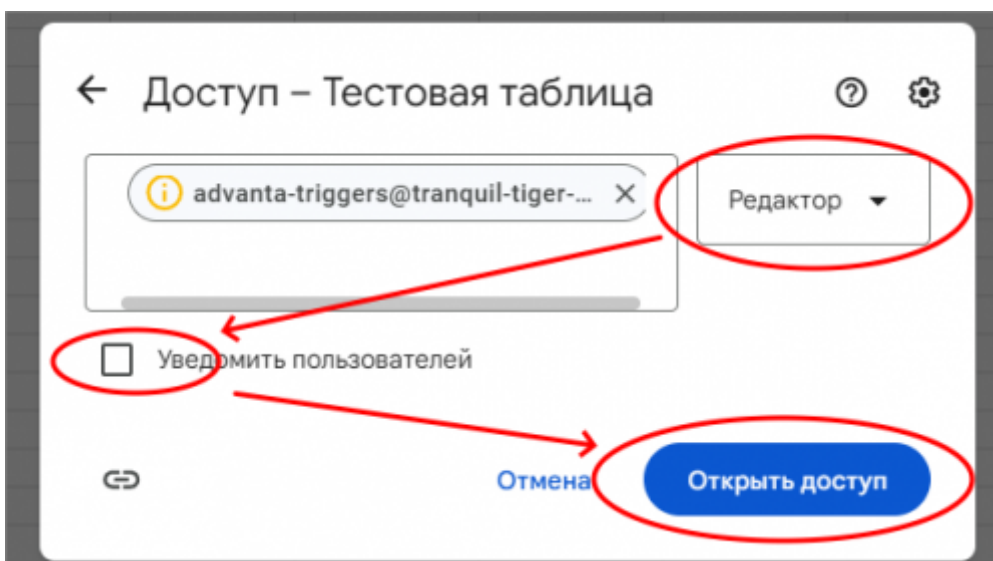
В Google-таблице, в которую приложение будет записывать данные, надо дать доступ на редактирование сервисной учётной записи. Диалог настройки вызывается кнопкой «Доступ» справа на панели инструментов.



В поле email-адреса необходимо ввести адрес почты сервисной учётной записи. Его можно найти в JSON-файле с данными авторизации, который был сохранен ранее. Он записан в элементе `client_email`.



Система предложит указать уровень доступа - нужно выбрать «Редактор». Опцию «Уведомить пользователей» можно отключить и нажать «Открыть доступ».



После сохранения настроек доступа, адрес сервисного аккаунта появится в списке пользователей таблицы, с выбранным уровнем доступа.

## Примеры программного кода

### Общие сведения о примерах

К инструкции прилагаются два примера кода, созданные в [Visual Studio Code](#):

- [exporttogooglesheets1libs.zip](#)

- демонстрирует выгрузку данных из ADVANTA в Google-таблицу с помощью библиотек Google, которые загружены в проект через NuGet. Библиотеки удобно использовать, если есть полный доступ к среде разработки, как например, при создании самостоятельного веб-сервиса:

- `exporttogooglesheets2nolib.zip`

- вариант экспорта данных без использования библиотек Google. Он значительно более сложный, но такой вариант можно использовать, например, в модуле триггеров, где нельзя подключить сторонние библиотеки.

Общий алгоритм работы примеров кода одинаков:

1. Проверяется доступ к Google-таблице, с помощью данных авторизации, настроенный в «Google Cloud Console» и сохранённых в файл.
2. Проводится авторизация в ADVANTA.
3. Выполняется запрос списка объектов с помощью заранее настроенного в ADVANTA [LINQ-запроса](#).
4. Выполняется очистка Google-таблицы от всех данных.
5. Список объектов записывается в Google-таблицу.

Чтобы открыть и запустить эти примеры, на ПК должен быть установлен [DotNet SDK 8](#) и сама среда разработки [Visual Studio Code](#). Для работы с C# в [Visual Studio Code](#) надо добавить расширение «C# Dev Kit». Для выгрузки данных из ADVANTA должен быть настроен LINQ-запрос.

## Подготовка LINQ-запроса в ADVANTA

From:  
<https://wiki.a2nta.ru/> - Wiki [3.x]

Permanent link:  
[https://wiki.a2nta.ru/doku.php/product/api/integration\\_examples/google\\_sheets?rev=1744023754](https://wiki.a2nta.ru/doku.php/product/api/integration_examples/google_sheets?rev=1744023754)

Last update: 07.04.2025 11:02

