# 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
Как запустить примеры кода	13
Подготовка каталога для примеров	13
Запуск примера «ExportToGoogleSheet1Libs» (с библиотеками Google)	13

Last update: 08.04.2025 product:api:integration\_examples:google\_sheets https://wiki.a2nta.ru/doku.php/product/api/integration\_examples/google\_sheets?rev=174409588907:04

# Экспорт данных из ADVANTA в Googleтаблицы

<head> <meta property=«og:title» content=«Экспорт данных из ADVANTA в Google-таблицы»/> <meta property=«og:description» content=«Статья о настройке экспорта данных из ADVANTA в Google-таблицы»/> <meta property=«og:image»

content=«https://wiki.a2nta.ru/lib/exe/fetch.php/product/api/integration\_examples/62d10bd608997.png?t=1744095770&w=500&h=90&tok=ed7a50»/> <meta property=«og:url» content=«https://wiki.a2nta.ru/doku.php/product/api/integration\_examples/google\_sheets»/> </head>

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

Если требуется регулярно передавать данные о проектах в таблицы Google, можно автоматизировать этот процесс. Для этого надо 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 API и получить «ключ доступа» к нему.
- 2. Настроить доступ к файлу электронной таблицы в Google Sheets для связанной с «сервисным аккаунтом» учётной записи.
- 3. Настроить в ADVANTA LINQ-запрос, который сформирует таблицу данных.
- 4. Написать приложение, которое выполнит LINQ-запрос и запишет полученные данные в Google Sheets.

Ниже рассмотрены примеры двух максимально простых приложений на С#, 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.

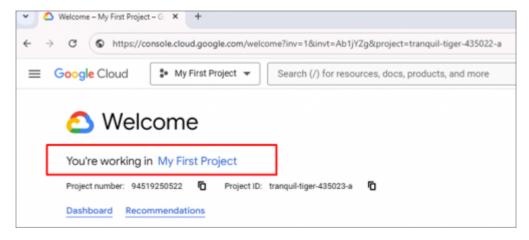
### Получение доступа к API через Google Cloud Console

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.

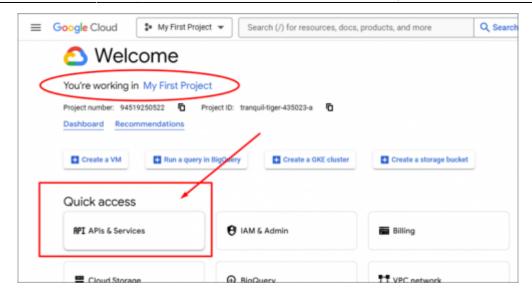
### Создание «проекта» в Google Cloud Console

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

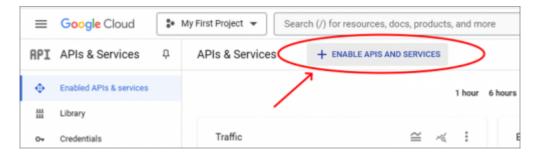


### Подключение к «проекту» библиотеки API Google Sheets

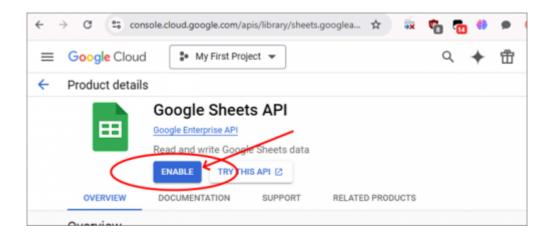
Для подключения к созданному проекту библиотеки API-функций Google Sheets необходимо перейти в раздел «APIs & Services» по кнопке под названием проекта.



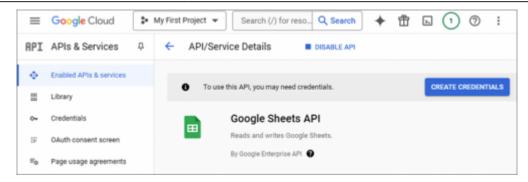
Затем необходимо открыть список библиотек кнопкой «Enable APIs and services».



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

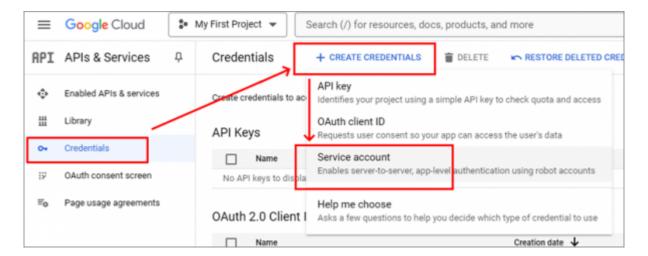


Библиотека появится в проекте, на странице «Enabled APIs & services».

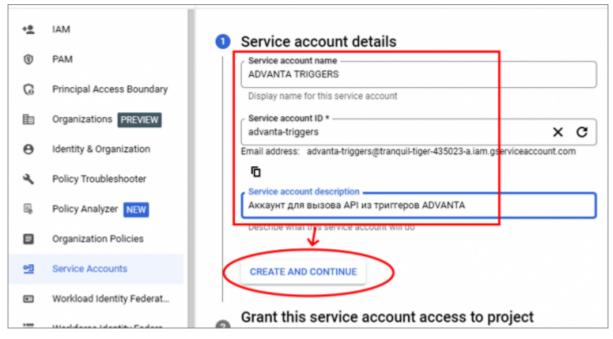


### Создание сервисного аккаунта Google для доступа к API «проекта»

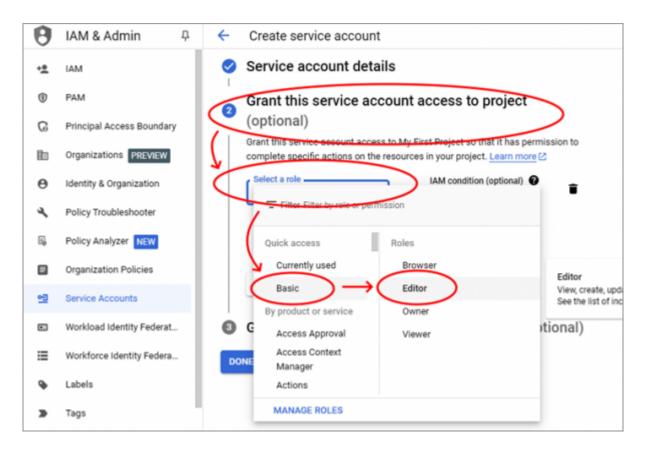
Для создания сервисного аккаунта Google необходимо в разделе «Credentials» вызвать меню «Create credentials > Service account».



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».



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



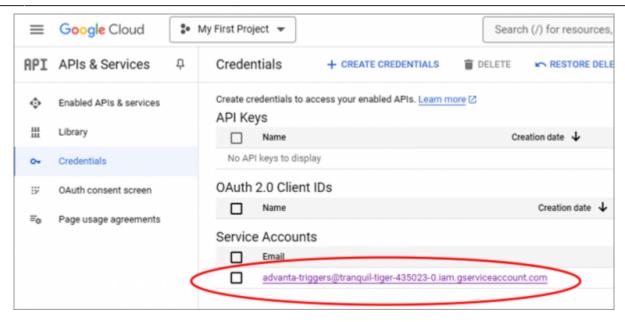
Для перехода к следующему шагу нажать **Continue**.

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.

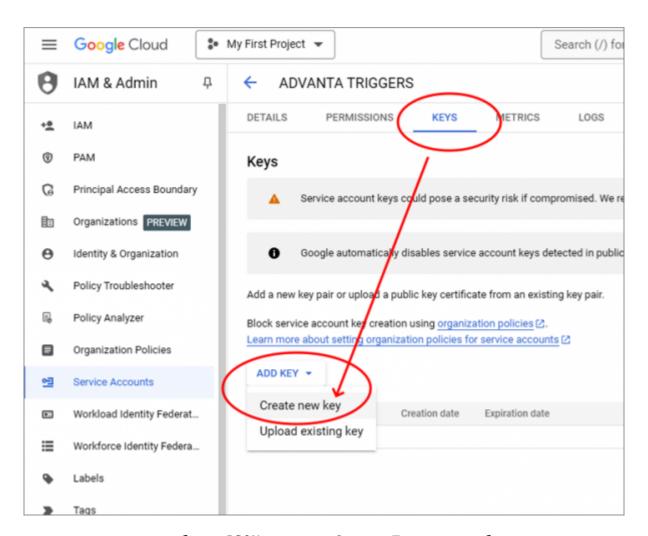
#### Создание ключа доступа к сервисному аккаунту

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

Перейти по ссылке с именем аккаунта а 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».

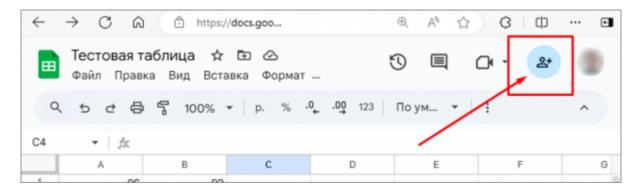
Далее он потребуется при создании кода приложения. Пример структуры файла «service-account.json»:

```
{} service-account.json > ...
 1
 2
        "type": "service_account",
        "project_id": "tranquil-tiger-435022-a0",
 3
        "private_key_id": "52b77cc59e50573be4a2a573e4bf3d86392fcaf0",
 4
        "private key": "----BEGIN PRIVATE KEY----\nMIIEvwIBADANBgkqhkiG9w0BAQEFAASCBKkwggS.
 5
        "client_email": "advanta-triggers@tranquil-tiger-435022-a0.iam.gserviceaccount.com",
 6
 7
        "client_id": "106608638608969228656",
        "auth_uri": "https://accounts.google.com/o/oauth2/auth",
        "token_uri": "https://oauth2.googleapis.com/token",
 9
        "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
10
        "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/advanta-te
11
        "universe_domain": "googleapis.com"
12
13
14
```

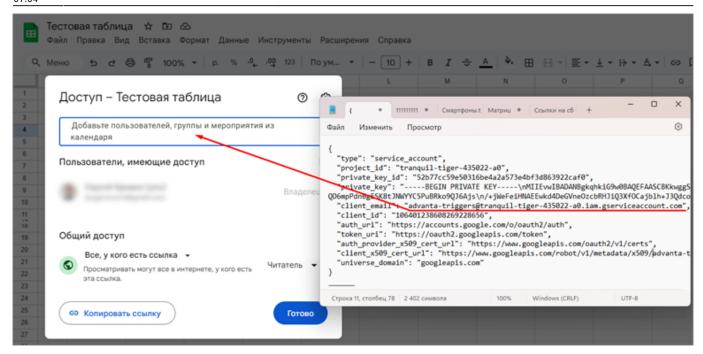
В реальных проектах важно обеспечить ограниченный доступ к файлу ключа и по возможности шифровать. Если ключ попадёт в поисковые системы, Google автоматически заблокирует его.

## Настройка доступа в Google-таблице

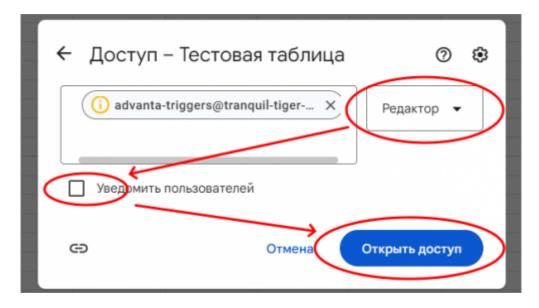
В Google-таблице, в которую приложение будет записывать данные, надо дать доступ на редактирование сервисной учётной записи. Диалог настройки вызывается кнопкой «Доступ» справа на панели инструментов.



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:
  - exporttogooglesheets2nolibs.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:

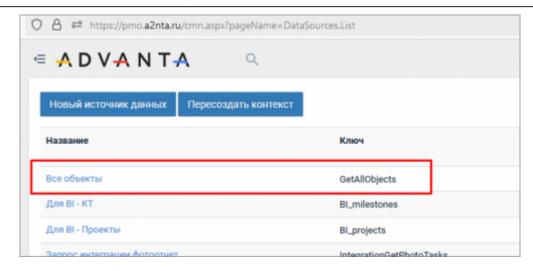
- 1. Проверяется доступ к Google-таблице, с помощью данных авторизации, настроенный в «Google Cloud Console» и сохранённых в файл.
- 2. Проводится авторизация в ADVANTA.
- 3. Выполняется запрос списка объектов с помощью заранее настроенного в ADVANTA LINQзапроса.
- 4. Выполняется очистка Google-таблицы 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».

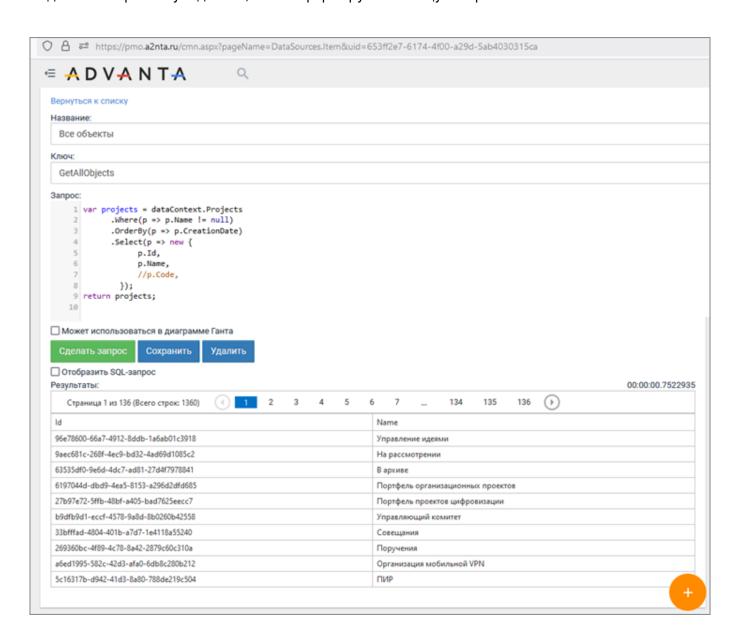
### Подготовка LINQ-запроса в ADVANTA

Для выгрузки списка объектов воспользуемся LINQ-запросом «Все объекты» (ключ «GetAllObjects»).

Необходимо убедиться, что запрос работает. Для этого, заходим в раздел LINQ-запросов Системы и проверяем, что запрос существует.



Далее необходимо перейти на страницу запроса кликом по названию, выполнить его кнопкой «Сделать запрос» и убедиться, что он формирует таблицу с перечнем объектов.



### Как запустить примеры кода

#### Подготовка каталога для примеров

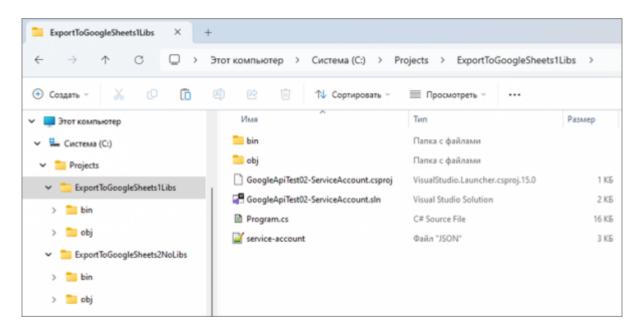
С помощью Проводника Windows необходимо создать на диске каталог C:\Projects и скопировать в него каталоги проектов из прилагаемых архивов

ExportToGoogleSheet1Libs.zip

И

#### ExportToGoogleSheet2NoLibs.zip

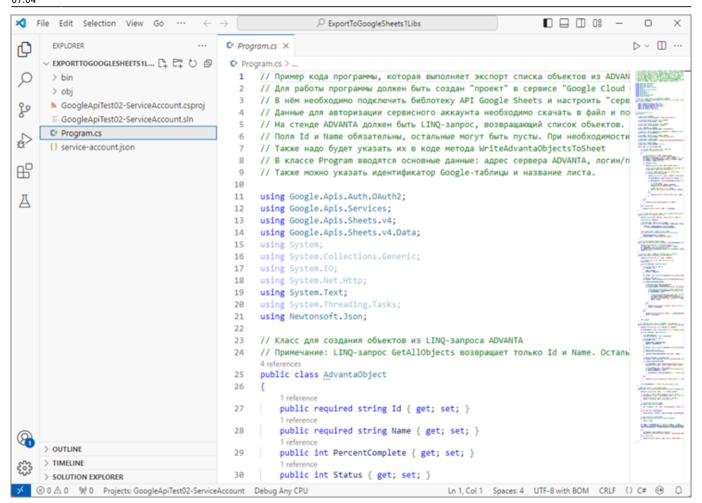
. В итоге, в каталоге «Projects» должны располагаться подкаталоги с программными файлами.



### Запуск примера «ExportToGoogleSheet1Libs» (с библиотеками Google)

Heoбходимо запустить Visual Studio Code и выполнить команду «File > Open Folder...». В окне «Open Folder» перейти в каталог C:\Projects\ExportToGoogleSheet1Libs и нажать «Выбор папки».

Содержащийся в каталоге код откроется в Visual Studio Code:



Прежде, чем запускать его на исполнение, необходимо проверить настройки параметров подключения к ADVANTA и к Google в коде:

1. Адрес сервера ADVANTA, логин и пароль пользователя, от имени которого будет выполняться LINQ-запрос:

```
// Основной класс программы
      0 references
35
      class Program
36
37
          // Адрес сервера ADVANTA
          2 references
38
          private static string a2serverUrl
                                                 "https://pmo.a2nta.ru";
39
40
          // Логин пользователя для подключения к ADVANTA
          1 reference
41
          private static string a2login =
42
43
          // Пароль пользователя для подключения к ADVANTA
          1 reference
          private static string a2password =
44
```

2. Имя ключа LINQ-запроса и количество объектов, которые программа из него получит. Большое количество объектов может потребовать большого объёма оперативной памяти компьютера, на котором выполняется программа:

```
// Ключ LINQ-запроса в ADVANTA
1 reference
private static string a2dataSourceKey = "GetAllObjects";

// Количество объектов, получаемых из LINQ-запроса в ADVANTA
1 reference
private static int a2numOfObjects = 7;
```

3. Далее нужно заменить в каталоге с кодом файл «service-account.json» на тот, который сгенерировали при настройке сервисного аккаунта в Google Cloud Console. При необходимости, можно разместить в каталоге с кодом несколько файлов авторизации и указать имя нужного в коде:

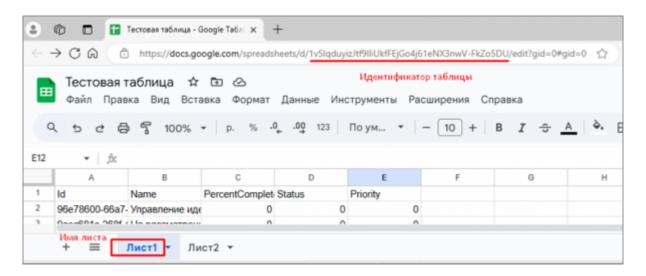
```
51
52 // Имя файла с данными для авторизации в Google
1 reference
53 private static string gFileName = "service-account.json";
54
```

Также нужно зарегистрировать дополнительный JSON в файле проекта «ExportToGoogleSheets1Lib.csproj», добавив в него код с новым именем файла, аналогично существующему:

```
10
       <ItemGroup>
          <PackageReference Include="Google.Apis.Auth" Version="1.68.0" />
11
          <PackageReference Include="Google.Apis.Sheets.v4" Version="1.68.0.3568" />
12
       </ItemGroup>
13
14
15
       <ItemGroup>
16
          <None Update="service-account.json">
              <CopyToOutputDirectory>Always</CopyToOutputDirectory>
17
18
          </None>
19
       </ItemGroup>
20
21
     </Project>
22
```

4. Далее указать идентификатор Google-таблицы, к которой дали доступ для сервисного аккаунта, а также имя листа в ней, на который требуется выгрузить данные:

**Идентификатор** - часть адреса страницы с таблицей, между /d/ и /edit, а имя листа – аналогично Excel:



Перед запуском примера необходимо обязательно сохранить изменения. После этого можно запускать решение. Для запуска служат команды справа на панели закладок Visual Studio Code.

#### Работа с примером

exporttogooglesheets2nolibs.zip полностью аналогична. Доступ к ADVANTA и Google настраивается точно также.

Если потребуется проводить собственные эксперименты с примерами в Visual Studio Code, рекомендуется создать новый проект, скопировать в него код из примера и подключить библиотеки Google.

From:

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

Permanent link:

https://wiki.a2nta.ru/doku.php/product/api/integration\_examples/google\_sheets?rev=1744095889

Last update: **08.04.2025 07:04** 

