Google Sheets Connected System for Appian

V2.0.3

appian

Appian Corporation

Version 2.0.3

Table of Contents

Overview

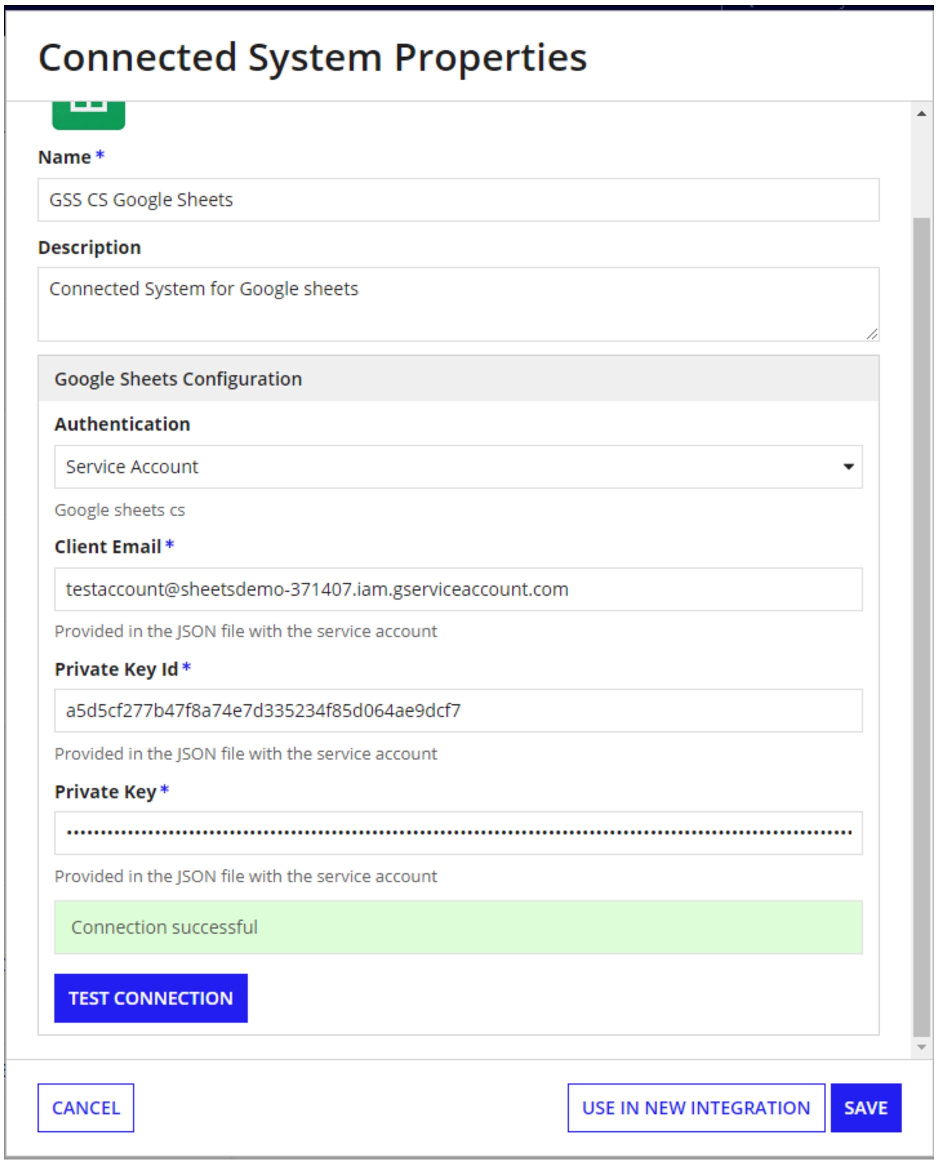
Google Sheets is a spreadsheet program included as part of the free, web-based Google Docs office suite offered by Google within its Google Drive service. The app is compatible with Microsoft Excel file formats and is available as a web application, mobile app for Android, iOS, Windows, BlackBerry, and as a desktop application on Google's ChromeOS. You can easily build different integrations using the Google Sheets connected system and access various functionalities directly in Appian by using the desired google sheet ID as input.

Features

- Spreadsheet Creation - Create Spreadsheets and specify the share with the specified Email addresses.
- Spreadsheets operation - Modifies a spreadsheet with Add Rows, Update Rows, Clear Rows, Delete Rows, Add Columns, and Delete Columns integration.
- Get Spreadsheet details - Get the specified cell range data with the Get Rows Integration and get spreadsheet details with the Get Sheet Details Integration.
- Data Fabric - Set up a record type with Data Source Integration and Sync Integration.

Connected System Configuration

1. Service Account Authentication


Connected System Properties

Name *

GSS CS Google Sheets

Description

Connected System for Google sheets

Google Sheets Configuration

Authentication

Service Account

Google sheets cs

Client Email *

testaccount@sheetsdemo-371407.iam.gserviceaccount.com

Provided in the JSON file with the service account

Private Key Id *

a5d5cf277b47f8a74e7d335234f85d064ae9dcf7

Provided in the JSON file with the service account

Private Key *

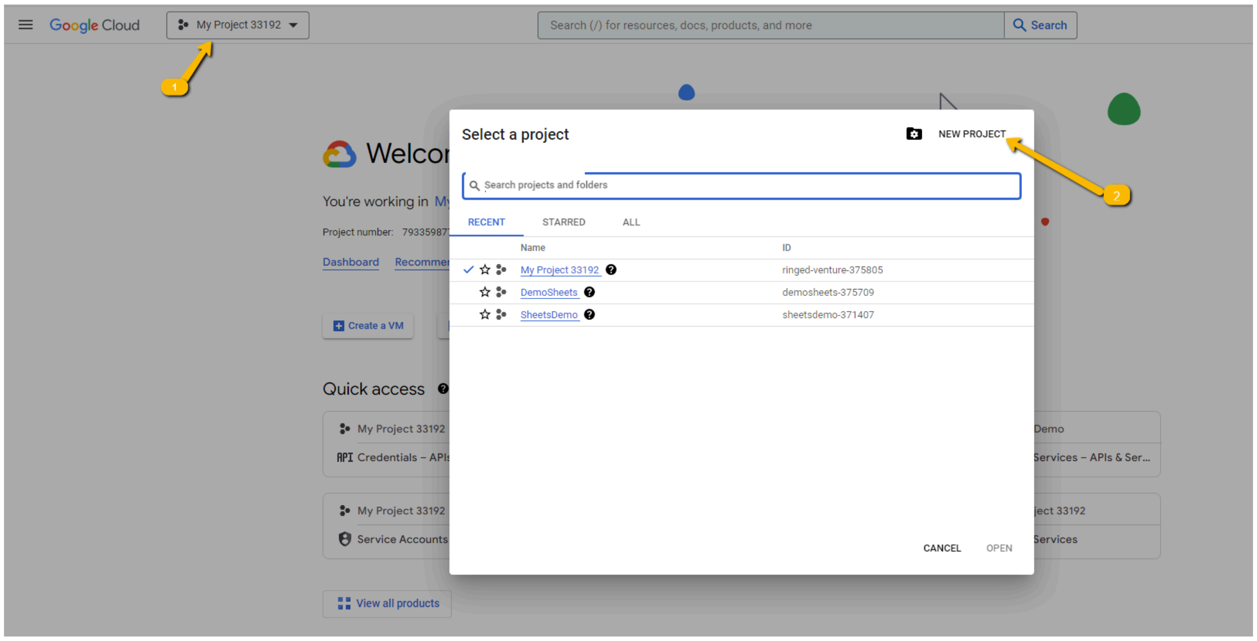
••

Provided in the JSON file with the service account

Connection successful

Steps to get service account credentials:

1. Go to Google Developer Console.

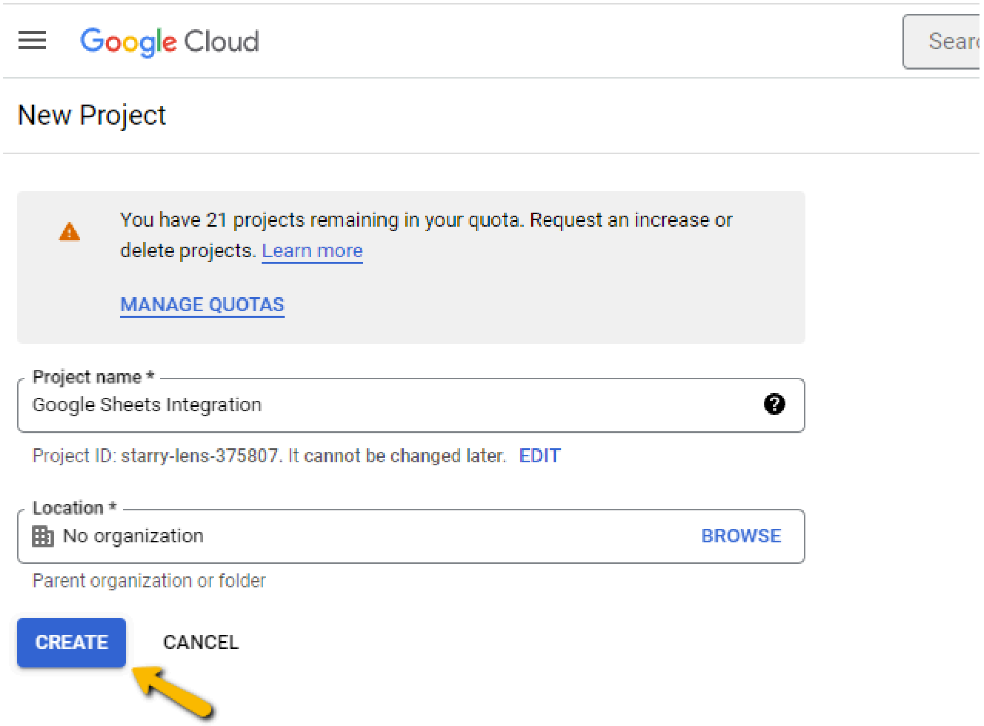


Google Cloud
My Project 33192
Search (/) for resources, docs, products, and more
Search
Select a project
NEW PROJECT
Search projects and folders
RECENT
STARRED
ALL

Name	ID
My Project 33192	ringed-venture-375805
DemoSheets	demosheets-375709
SheetsDemo	sheetsdemo-371407

2. Click on **NEW PROJECT** to create a new console project.

3. Enter a name for the project and select Organization as per needs.



You have 21 projects remaining in your quota. Request an increase or delete projects. Learn more

MANAGE QUOTAS

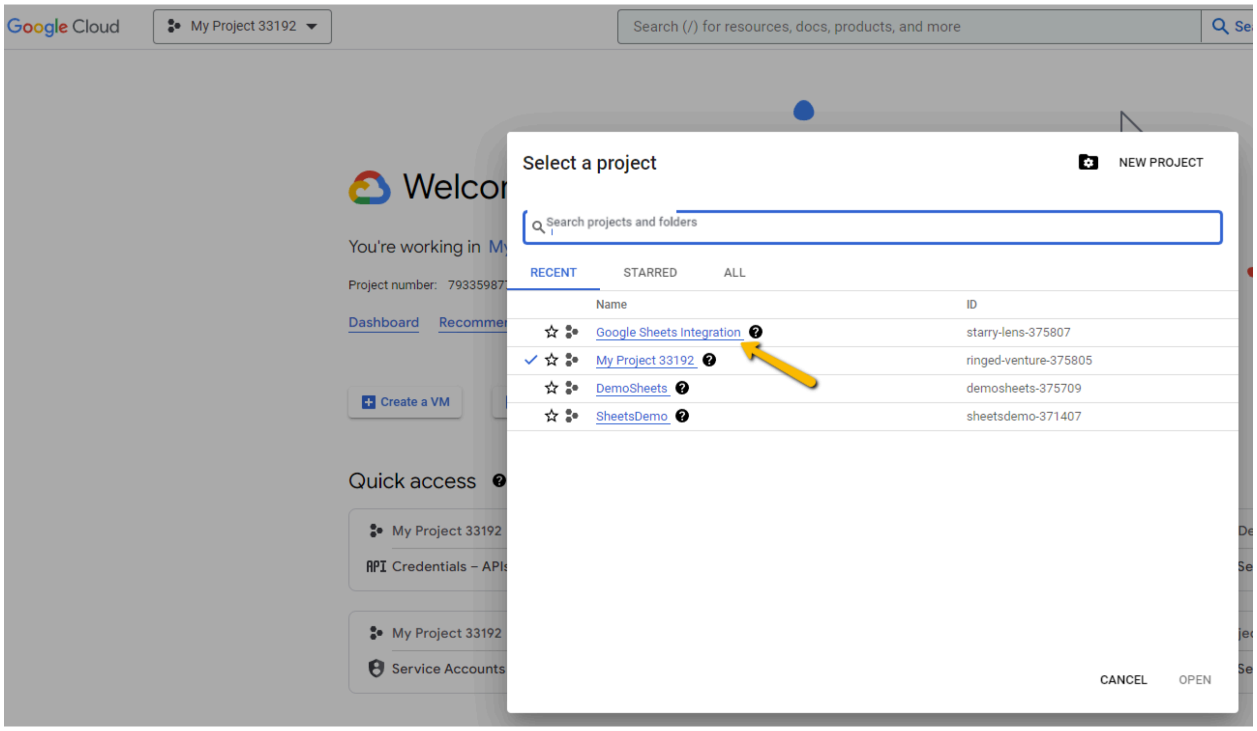
Project name * Google Sheets Integration

Project ID: starry-lens-375807. It cannot be changed later. EDIT

Location * No organization BROWSE

Parent organization or folder

4. Now, Select the project from the Dropdown.


Select a project

NEW PROJECT

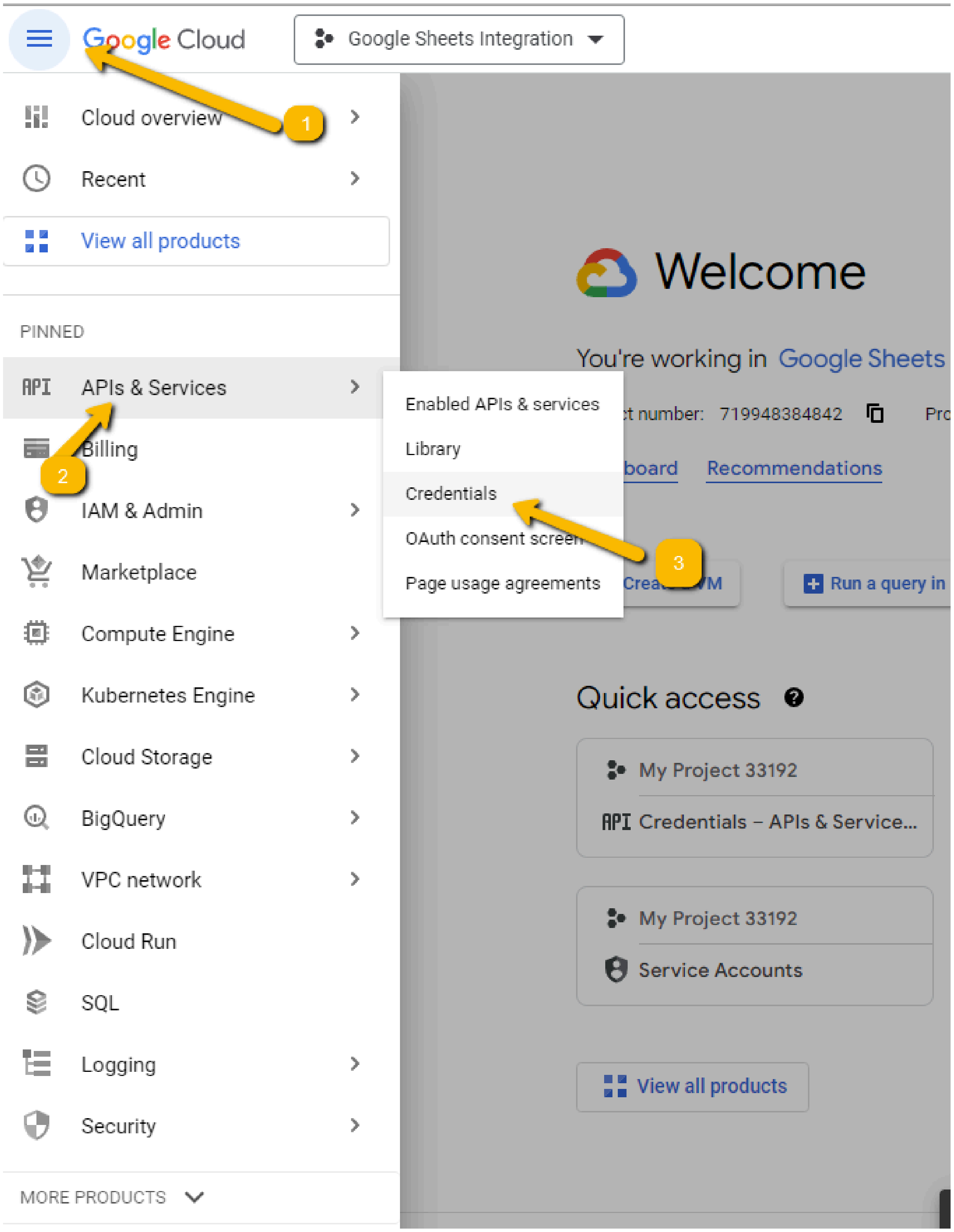
Search projects and folders

RECENT STARRED ALL

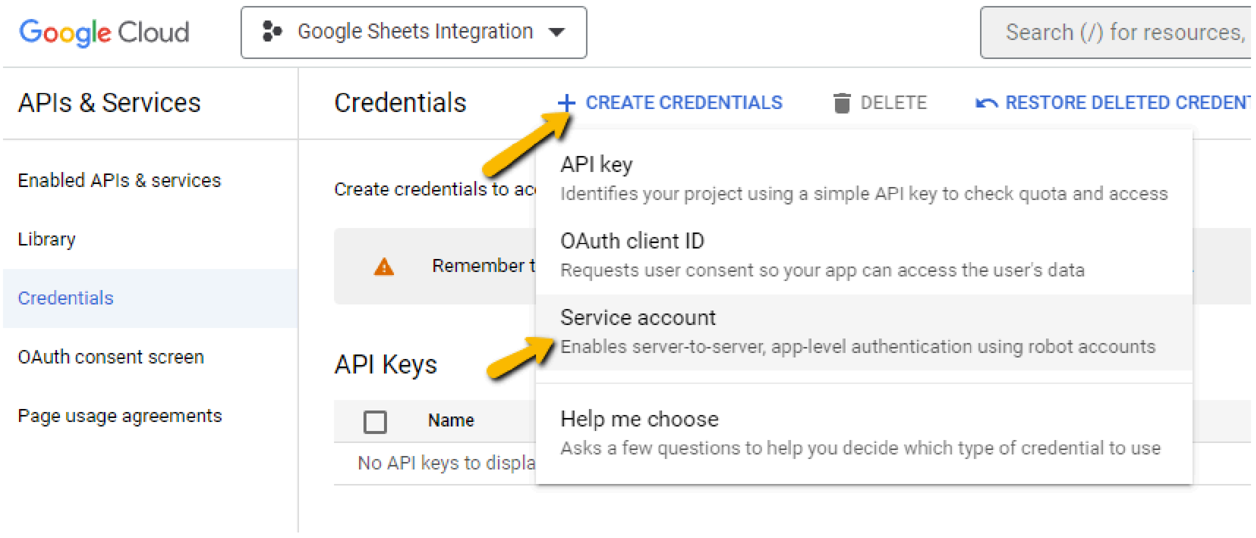
Name	ID
Google Sheets Integration	starry-lens-375807
✓ My Project 33192	ringed-venture-375805
DemoSheets	demosheets-375709
SheetsDemo	sheetsdemo-371407

CANCEL OPEN

5. Now select **Credentials** from **APIs & Services** from Menu.


Google Cloud
Google Sheets Integration
Cloud overview
1
Recent
View all products
PINNED
APIs & Services
2
Billing
IAM & Admin
Marketplace
Compute Engine
Kubernetes Engine
Cloud Storage
BigQuery
VPC network
Cloud Run
SQL
Logging
Security
MORE PRODUCTS
Enabled APIs & services
Library
Credentials
3
OAuth consent screen
Page usage agreements
Welcome
You're working in Google Sheets
t number: 719948384842
board
Recommendations
Run a query in
Quick access
My Project 33192
Credentials – APIs & Service...
My Project 33192
Service Accounts
View all products

6. Select **Service Account** under **Create Credentials**



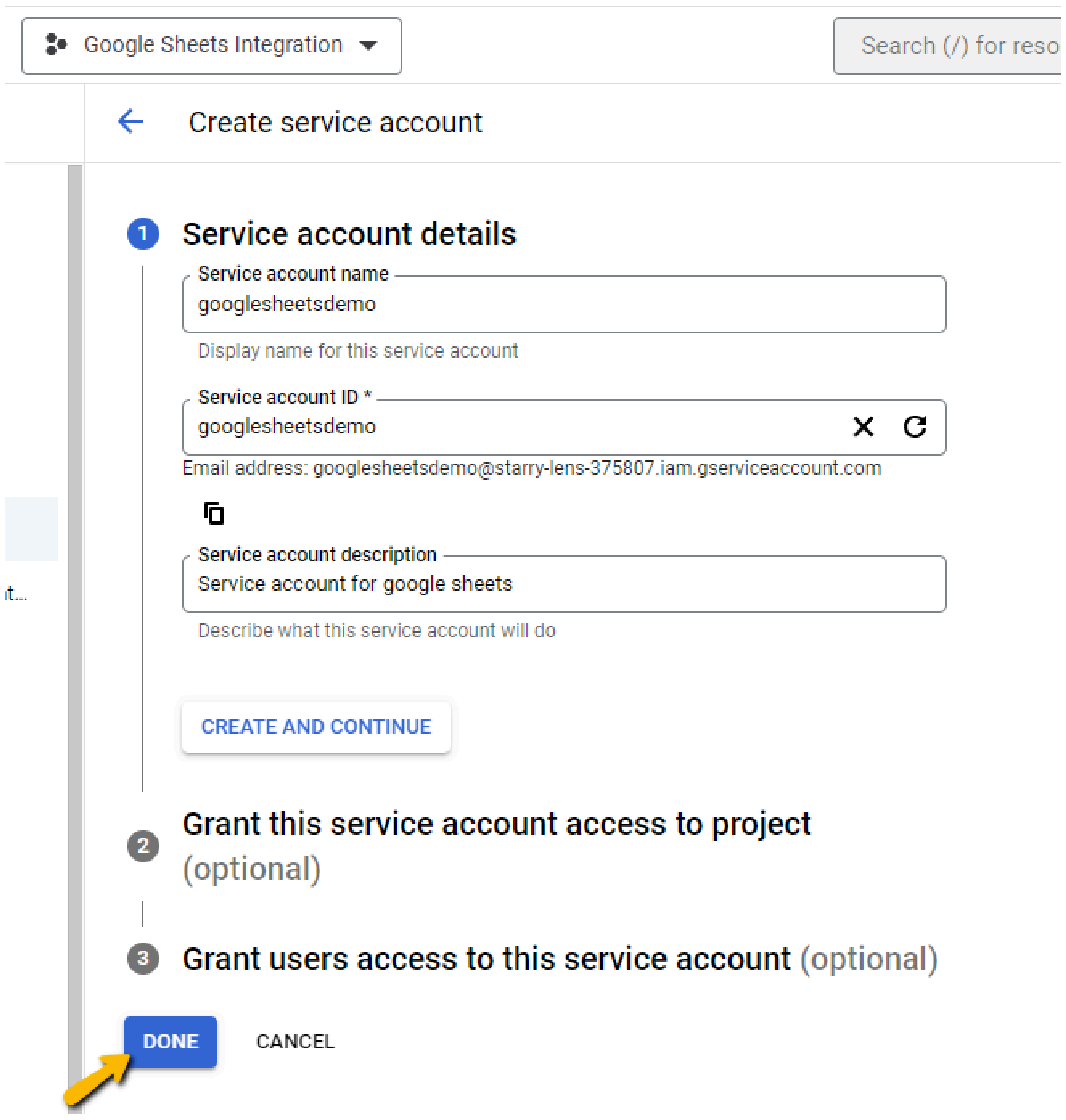
API key
Identifies your project using a simple API key to check quota and access

OAuth client ID
Requests user consent so your app can access the user's data

Service account
Enables server-to-server, app-level authentication using robot accounts

Help me choose
Asks a few questions to help you decide which type of credential to use

7. Now fill the details of the service account and click **Done** as the other fields are optional.


Google Sheets Integration
Search (/) for reso
Create service account
1
Service account details
Service account name: googlesheetsdemo

Display name for this service account

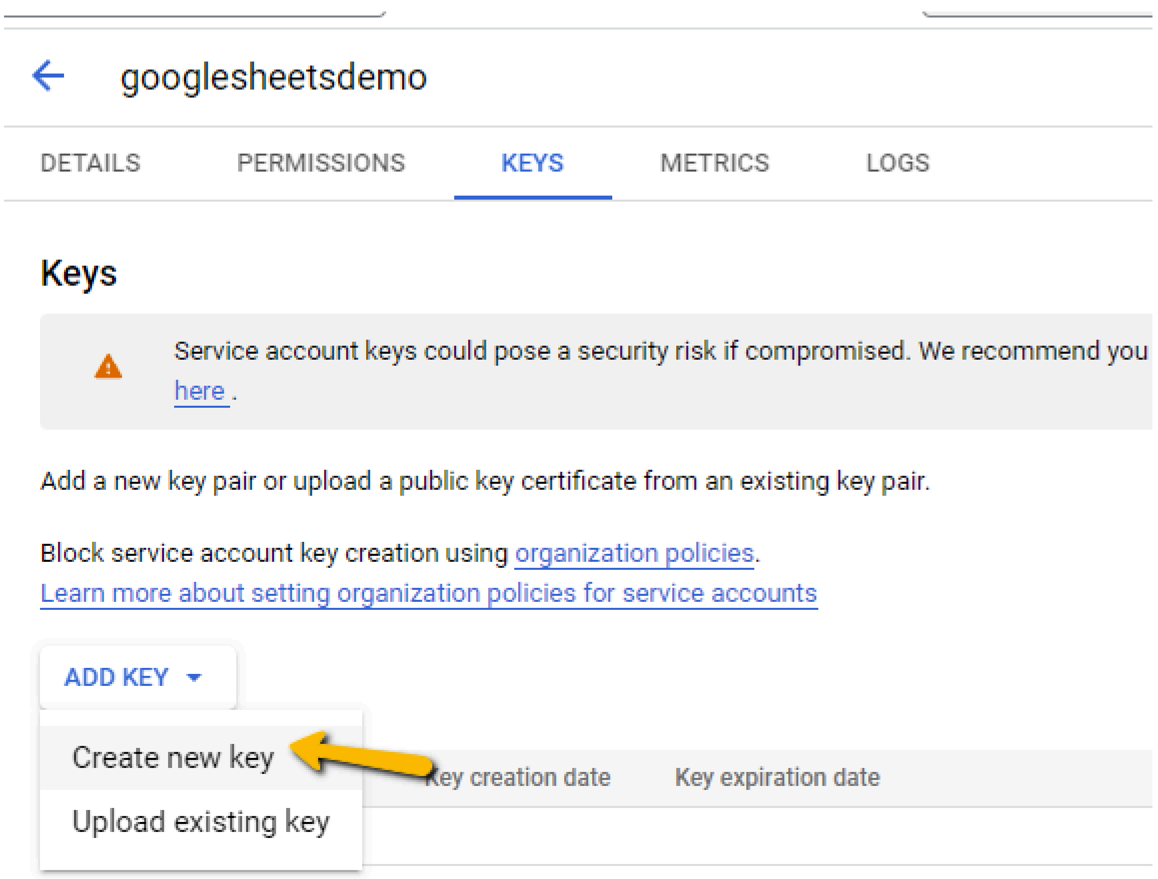
Service account ID *: googlesheetsdemo

Email address: googlesheetsdemo@starry-lens-375807.iam.gserviceaccount.com

Service account description: Service account for google sheets

Describe what this service account will do
CREATE AND CONTINUE
2
Grant this service account access to project (optional)
3
Grant users access to this service account (optional)
DONE
CANCEL
it...

8. Now click on the created Service Account name and create a new key under the **KEYS** tab.


googlesheetsdemo

DETAILS PERMISSIONS KEYS METRICS LOGS

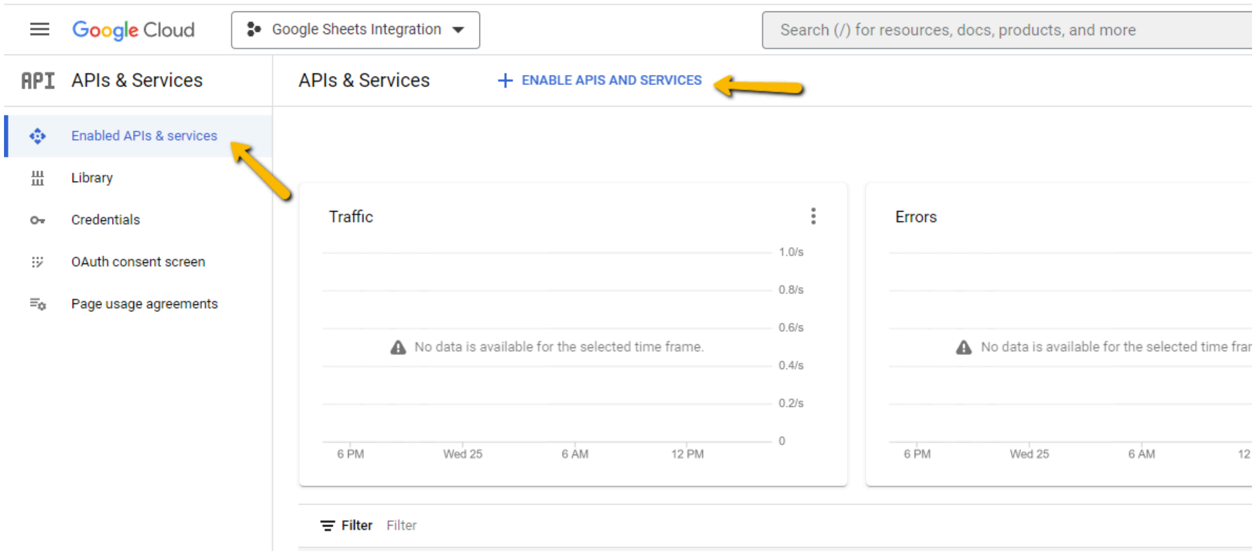
Keys

Service account keys could pose a security risk if compromised. We recommend you here .

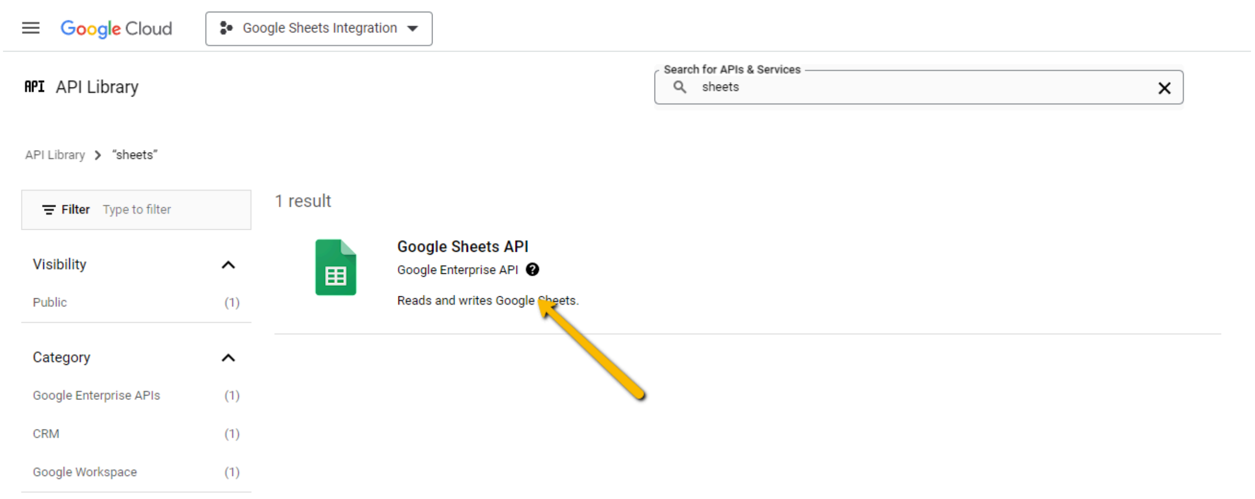
Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using organization policies.
Learn more about setting organization policies for service accounts

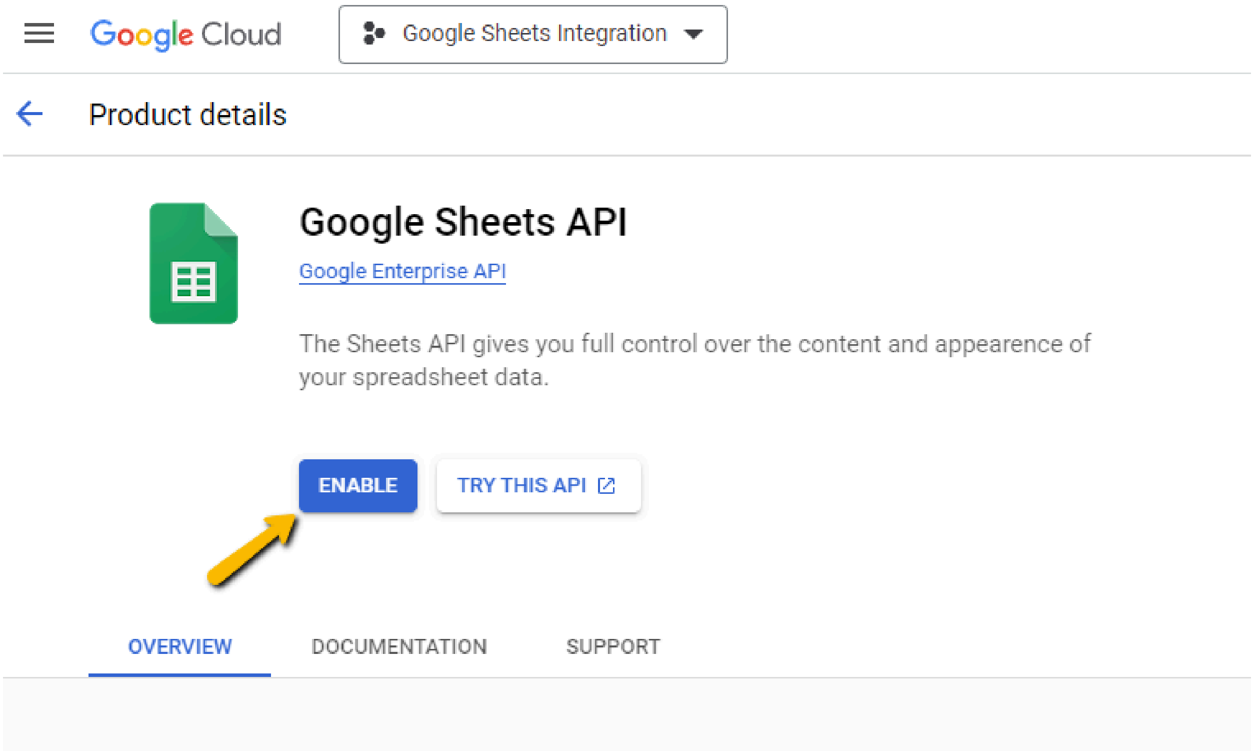
9. Select **JSON** and click **create.** A JSON file will be downloaded automatically.
10. Use the values of client email, private key id and private key from the JSON file in the Connected system.
11. In the console, select **Enabled APIs & services** from the menu and click on **Enable APIs and Services.**



12. Search for sheets in the search box and click on **Google Sheets API**.

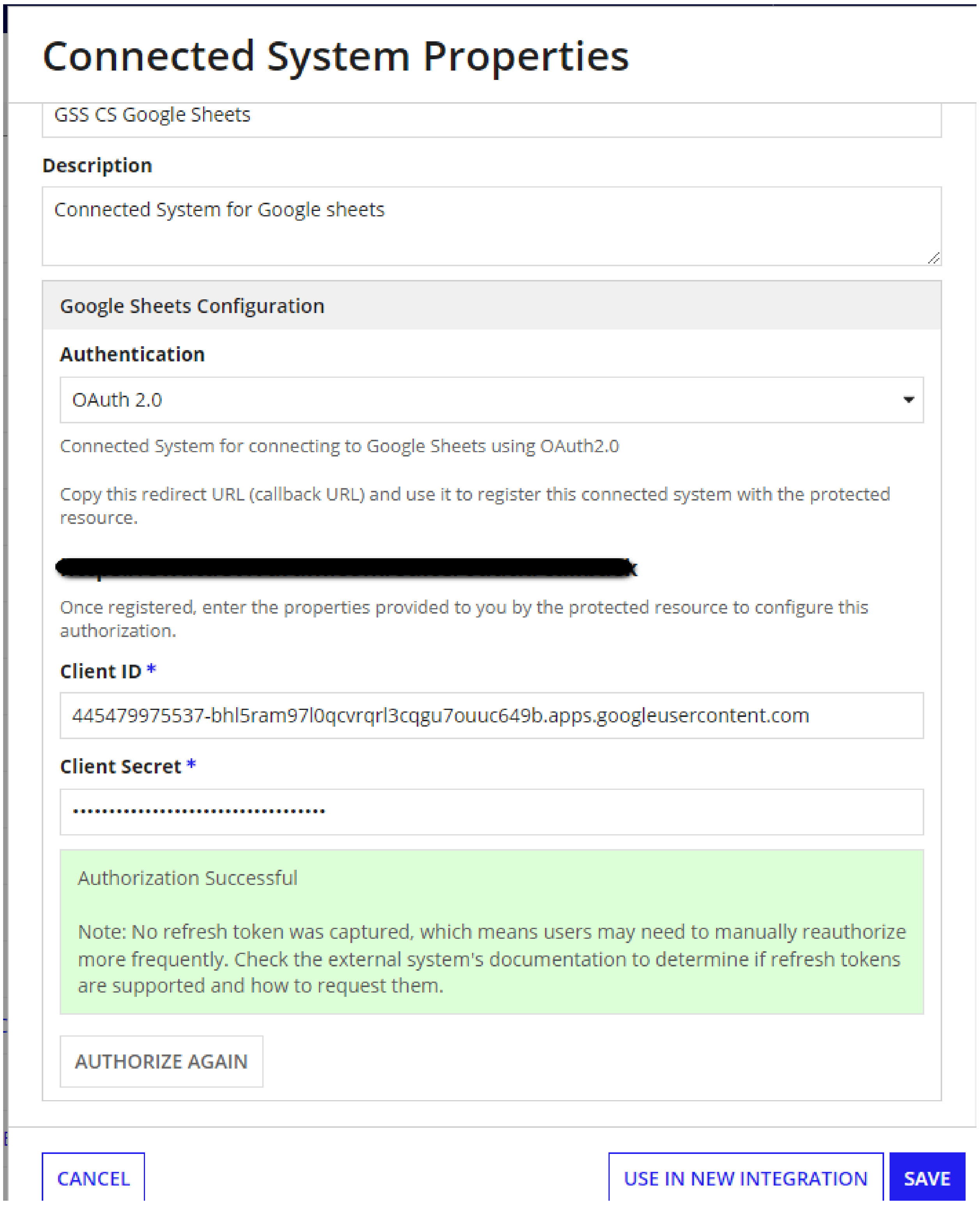


13. Click on **Enable** button to enable sheets API.


Google Cloud
Google Sheets Integration
Product details
Google Sheets API
Google Enterprise API
The Sheets API gives you full control over the content and appearence of your spreadsheet data.
ENABLE
TRY THIS API
OVERVIEW
DOCUMENTATION
SUPPORT

14. Necessary setup in the Google console is now complete.

2. OAuth 2.0 Authentication



Connected System Properties

GSS CS Google Sheets

Description

Connected System for Google sheets

Google Sheets Configuration

Authentication

OAuth 2.0

Connected System for connecting to Google Sheets using OAuth2.0

Copy this redirect URL (callback URL) and use it to register this connected system with the protected resource.

[redacted]

Once registered, enter the properties provided to you by the protected resource to configure this authorization.

Client ID *

445479975537-bhl5ram97l0qcvrqrl3cqgu7ouuc649b.apps.googleusercontent.com

Client Secret *

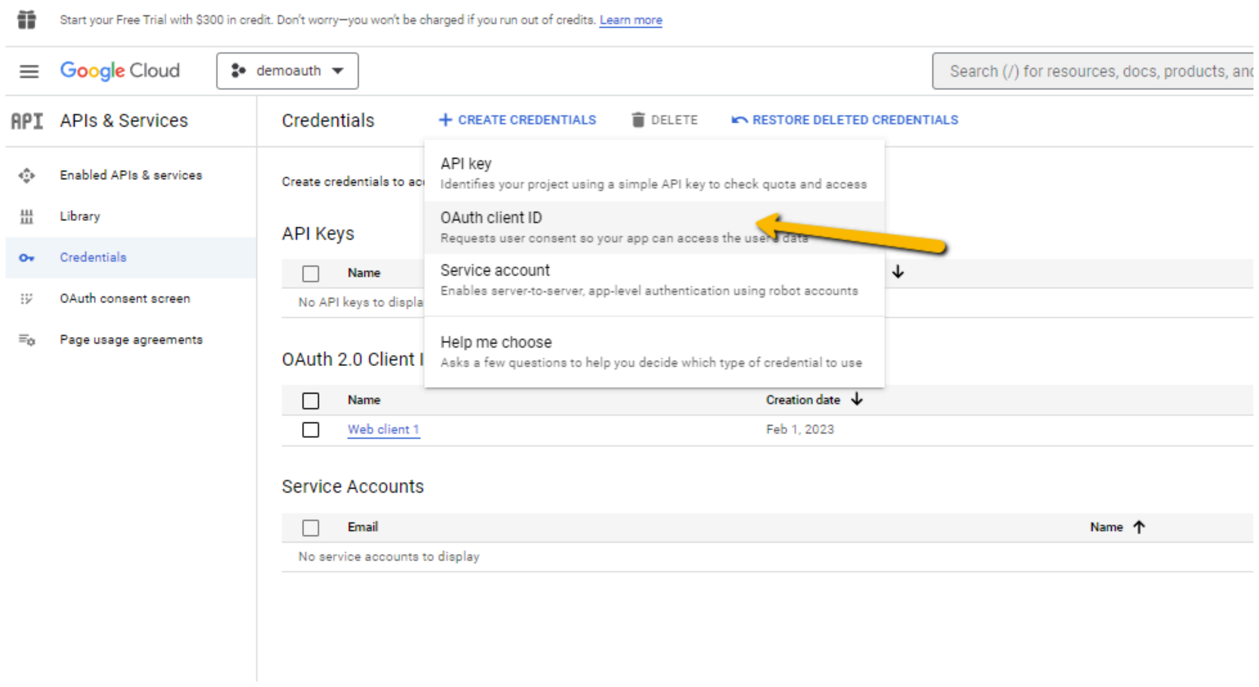
••••••••••••••••••••••••••••••••••••

Authorization Successful

Note: No refresh token was captured, which means users may need to manually reauthorize more frequently. Check the external system's documentation to determine if refresh tokens are supported and how to request them.

Steps to get OAuth 2.0 credentials:

1. Select credentials from the menu and click on **Create Credentials** which we did for creating a service account and click on **OAuth client ID** for OAuth authentication.



Start your Free Trial with $300 in credit. Don't worry—you won't be charged if you run out of credits. Learn more

Google Cloud | demoauth | Search (/) for resources, docs, products, and

APIs & Services | Credentials | + CREATE CREDENTIALS | DELETE | RESTORE DELETED CREDENTIALS

Enabled APIs & services | Library | Credentials | OAuth consent screen | Page usage agreements

Create credentials to ac

- API key — Identifies your project using a simple API key to check quota and access
- OAuth client ID — Requests user consent so your app can access the user's data
- Service account — Enables server-to-server, app-level authentication using robot accounts
- Help me choose — Asks a few questions to help you decide which type of credential to use

API Keys

Name
No API keys to displa

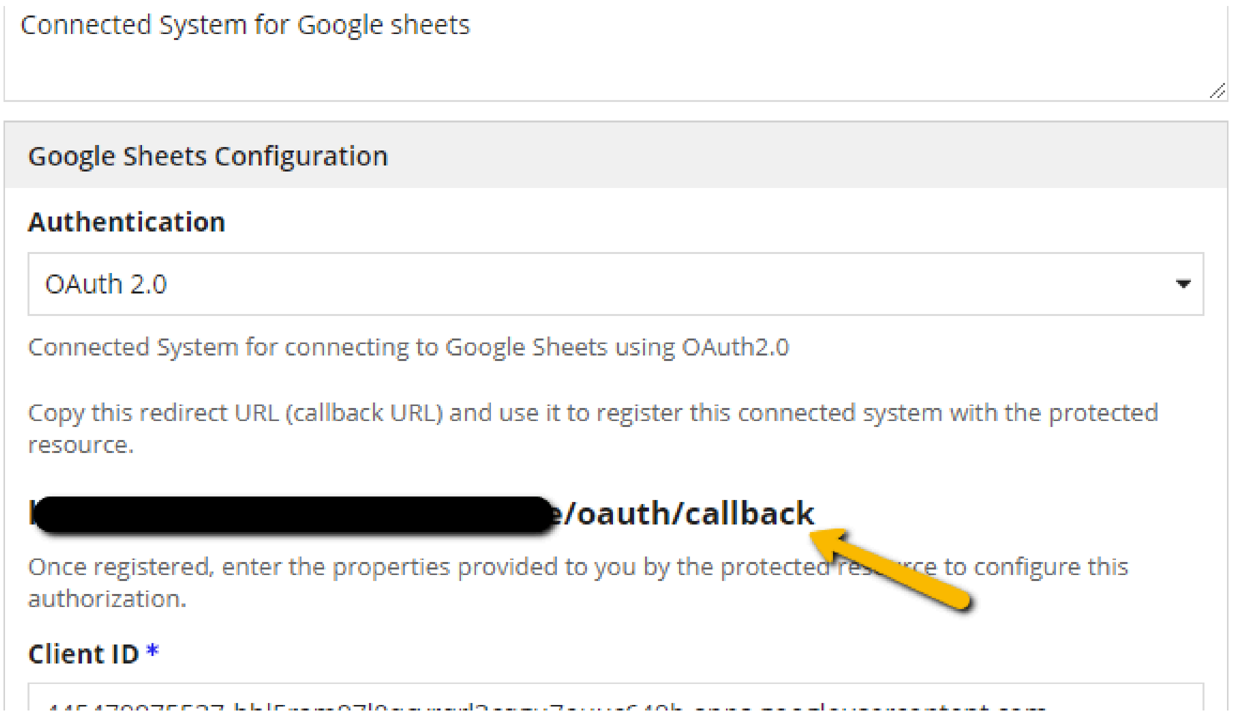
OAuth 2.0 Client I

Name	Creation date
Web client 1	Feb 1, 2023

Service Accounts

Email	Name
No service accounts to display	

2. Before proceeding to next step, copy the callback url from the connected system shown below



Connected System for Google sheets

Google Sheets Configuration

Authentication

OAuth 2.0

Connected System for connecting to Google Sheets using OAuth2.0

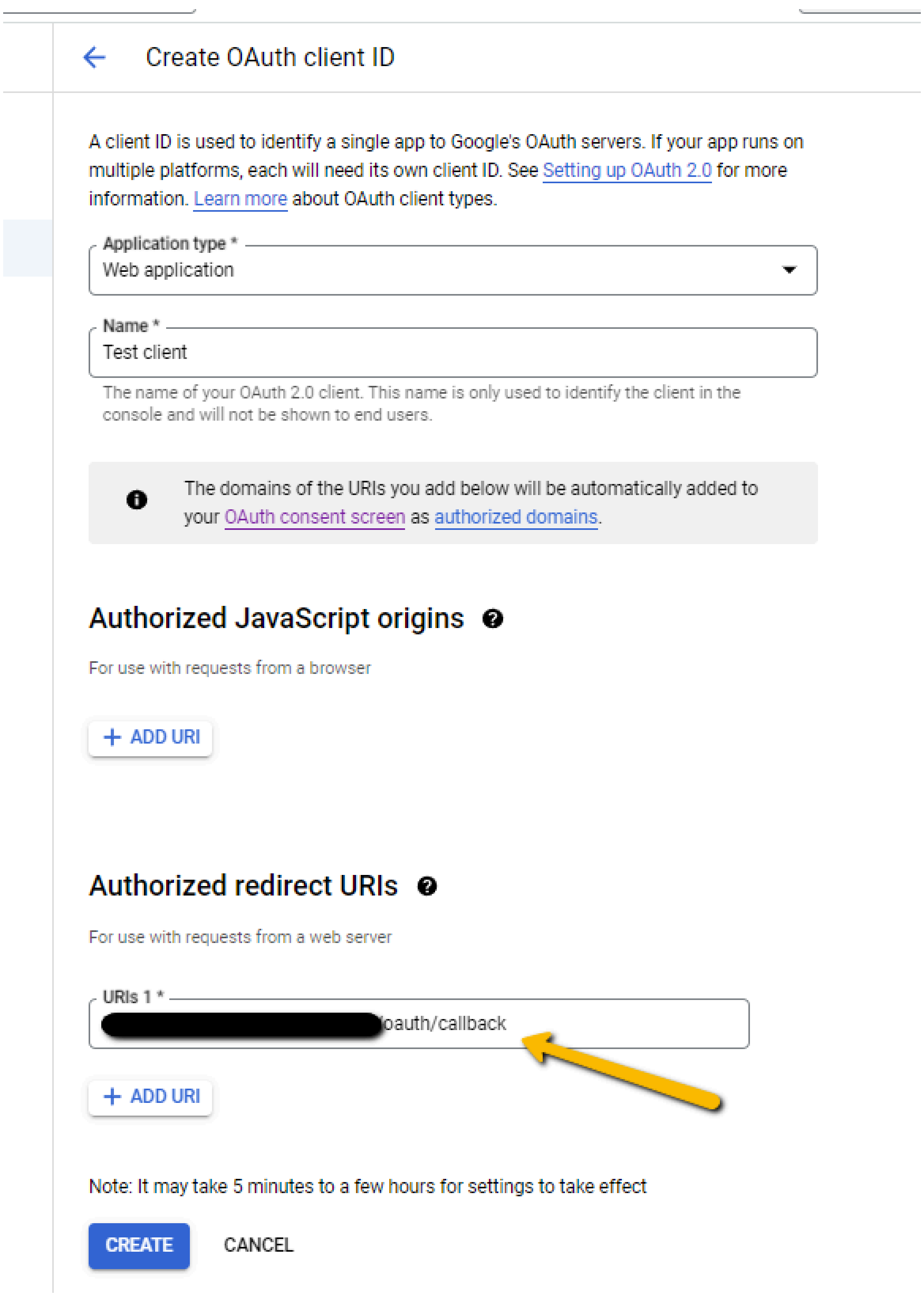
Copy this redirect URL (callback URL) and use it to register this connected system with the protected resource.

[redacted]/oauth/callback

Once registered, enter the properties provided to you by the protected resource to configure this authorization.

Client ID *

3. Now select **Web Application** in Application type dropdown and provide **Name** for the credential. In the **Authorized redirect URIs** section, add the redirect URI you have copied from the connected system and click on **create.**


Create OAuth client ID

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See Setting up OAuth 2.0 for more information. Learn more about OAuth client types.

Application type *
Web application

Name *
Test client

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

The domains of the URIs you add below will be automatically added to your OAuth consent screen as authorized domains.

Authorized JavaScript origins

For use with requests from a browser

Authorized redirect URIs

For use with requests from a web server

URIs 1 *
[redacted]/oauth/callback

Note: It may take 5 minutes to a few hours for settings to take effect

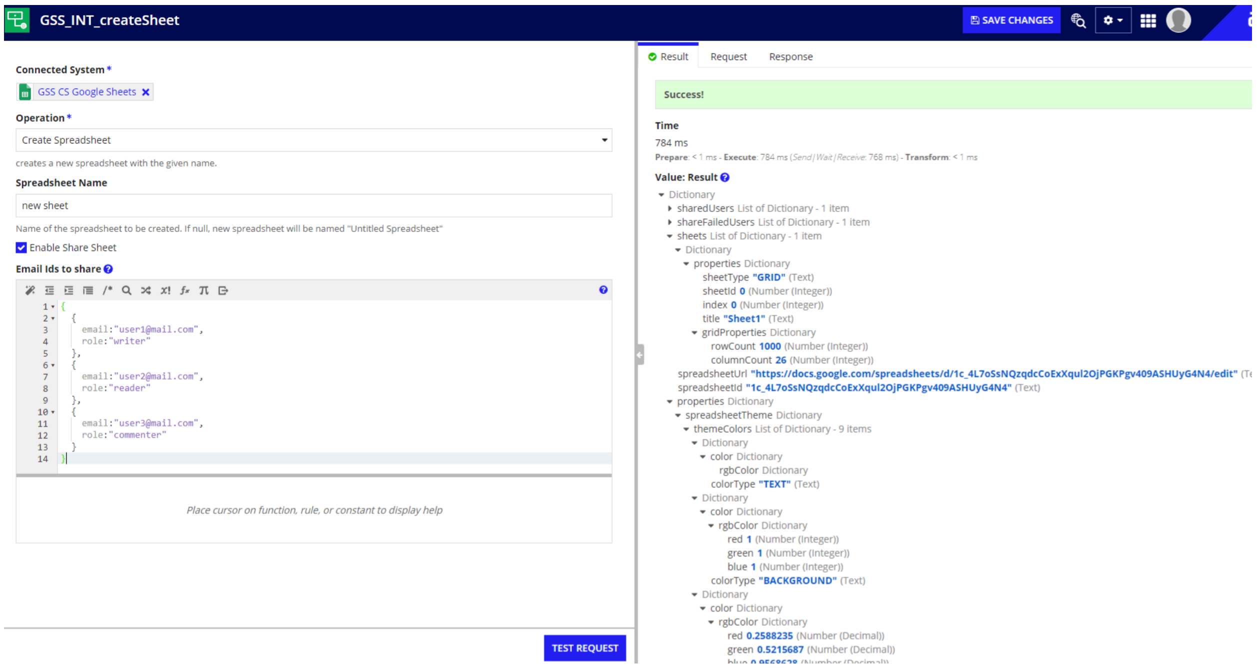
4. The OAuth credential is now created with client Id and client secret .You can download the JSON file for using the credentials in the connected system.

Integration Configuration

Create Spreadsheet

Creates a new spreadsheet and shares the spreadsheet to users.

Note: This Integration only works with OAuth Authentication.



GSS_INT_createSheet

Connected System *
GSS CS Google Sheets

Operation *
Create Spreadsheet
creates a new spreadsheet with the given name.

Spreadsheet Name
new sheet
Name of the spreadsheet to be created. If null, new spreadsheet will be named "Untitled Spreadsheet"

☑ Enable Share Sheet

Email Ids to share

```
{
  {
    email:"user1@mail.com",
    role:"writer"
  },
  {
    email:"user2@mail.com",
    role:"reader"
  },
  {
    email:"user3@mail.com",
    role:"commenter"
  }
}
```

Place cursor on function, rule, or constant to display help

Success!

Time
784 ms
Prepare: < 1 ms - Execute: 784 ms (Send / Wait / Receive: 768 ms) - Transform: < 1 ms

Value: Result

```
▾ Dictionary
  ▸ sharedUsers List of Dictionary - 1 item
  ▸ shareFailedUsers List of Dictionary - 1 item
  ▾ sheets List of Dictionary - 1 item
    ▾ Dictionary
      ▾ properties Dictionary
          sheetType "GRID" (Text)
          sheetId 0 (Number (Integer))
          index 0 (Number (Integer))
          title "Sheet1" (Text)
        ▾ gridProperties Dictionary
            rowCount 1000 (Number (Integer))
            columnCount 26 (Number (Integer))
    spreadsheetUrl "https://docs.google.com/spreadsheets/d/1c_4L7oSsNQzqdcCoExXqul2OjPGKPgv409ASHUyG4N4/edit" (Te
    spreadsheetId "1c_4L7oSsNQzqdcCoExXqul2OjPGKPgv409ASHUyG4N4" (Text)
  ▾ properties Dictionary
    ▾ spreadsheetTheme Dictionary
      ▾ themeColors List of Dictionary - 9 items
        ▾ Dictionary
          ▾ color Dictionary
              rgbColor Dictionary
            colorType "TEXT" (Text)
        ▾ Dictionary
          ▾ color Dictionary
            ▾ rgbColor Dictionary
                red 1 (Number (Integer))
                green 1 (Number (Integer))
                blue 1 (Number (Integer))
            colorType "BACKGROUND" (Text)
        ▾ Dictionary
          ▾ color Dictionary
            ▾ rgbColor Dictionary
                red 0.2588235 (Number (Decimal))
                green 0.5215687 (Number (Decimal))
                blue 0.9568628 (Number (Decimal))
```

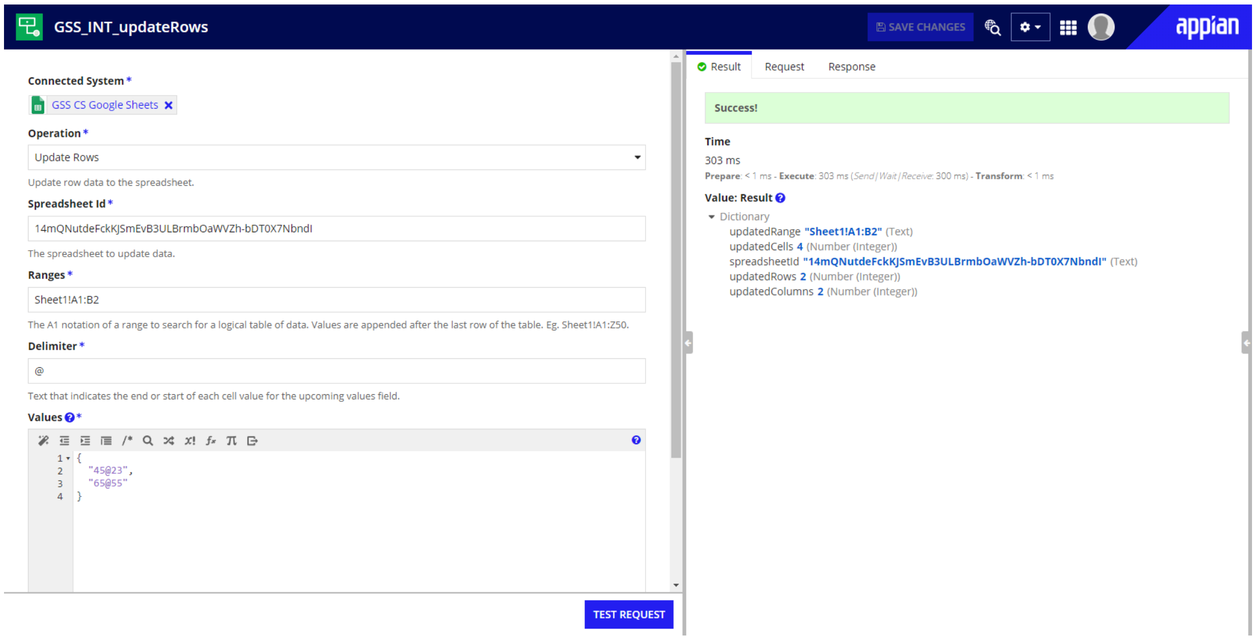
Spreadsheet Id - A new sheet will be created under this name. If it is empty then, it will be named "Untitled Spreadsheet".
Enable Share Sheet - Check box to share the created spreadsheet. Below Expression box will be displayed only when this checkbox is checked.
Email Ids to share - Accepts list of dictionary in which each dictionary corresponds to

```
{
   email:"user1@mail.com",
   role:"<<role>>"
 }
```

   email- Email Id of the user to be shared.
   role- should be one of the following: "reader","writer","commenter"

Update Rows

Clears cell values in the given range.



GSS_INT_updateRows

SAVE CHANGES

Connected System *

GSS CS Google Sheets

Operation *

Update Rows

Update row data to the spreadsheet.

Spreadsheet Id *

14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI

The spreadsheet to update data.

Ranges *

Sheet1!A1:B2

The A1 notation of a range to search for a logical table of data. Values are appended after the last row of the table. Eg. Sheet1!A1:Z50.

Delimiter *

@

Text that indicates the end or start of each cell value for the upcoming values field.

Values *

```
{
  "45@23",
  "65@55"
}
```

TEST REQUEST

Result | Request | Response

Success!

Time

303 ms

Prepare: < 1 ms - **Execute**: 303 ms (*Send / Wait / Receive*: 300 ms) - **Transform**: < 1 ms

Value: Result

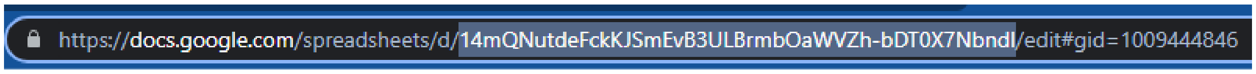
```
Dictionary
    updatedRange "Sheet1!A1:B2" (Text)
    updatedCells 4 (Number (Integer))
    spreadsheetId "14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI" (Text)
    updatedRows 2 (Number (Integer))
    updatedColumns 2 (Number (Integer))
```

Spreadsheet Id - Alphanumeric characters present in the sheet url
Ranges - Cell range in A1 notation.
Delimiter- Text that indicates the end or start of each cell value for the upcoming values field.
Values - List of text values in which each value represents row values. Each cell value is delimited by delimiter..
Example: If you want to update the values in the cells as follows
A1: Text1
A2: Text2
B1: Text3
B2: Text4 in Sheet1,

Then *Sheet1!A1:B2* should be given in Ranges field and
{
"Text1!~!Text3",
"Text2!~!Text4"
}
should be given in the Values field.

Note:

- You can extract spreadsheet Id from the URL like shown below . The highlighted text is the spreadsheet Id



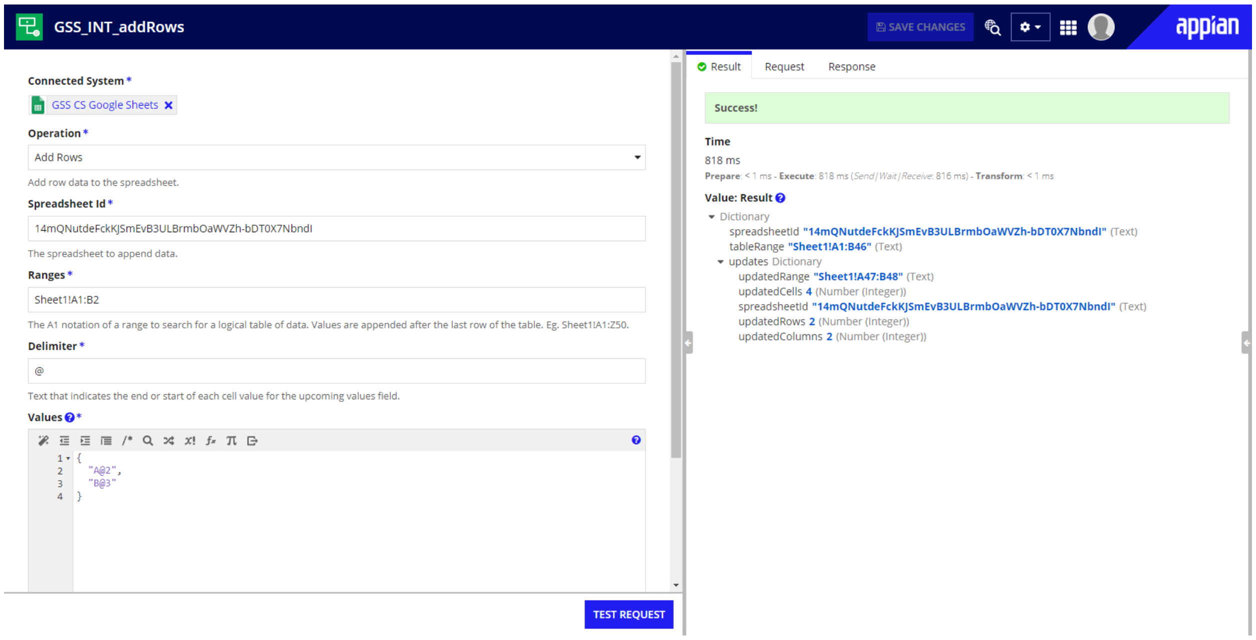
https://docs.google.com/spreadsheets/d/14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7Nbndl/edit#gid=1009444846

- A1 notation should be used inside the ranges field. Refer to this link to know about A1 notation.

Add Rows

Inserts the given data in the specified range. If the cells are already filled, then the next empty row cell will be added.

.



Connected System *
GSS CS Google Sheets

Operation *
Add Rows
Add row data to the spreadsheet.

Spreadsheet Id *
14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI
The spreadsheet to append data.

Ranges *
Sheet1!A1:B2
The A1 notation of a range to search for a logical table of data. Values are appended after the last row of the table. Eg. Sheet1!A1:Z50.

Delimiter *
@
Text that indicates the end or start of each cell value for the upcoming values field.

Values *

```
{
  "A@2",
  "B@3"
}
```

Success!

Time
818 ms
Prepare: < 1 ms - **Execute**: 818 ms (*Send / Wait / Receive*: 816 ms) - **Transform**: < 1 ms

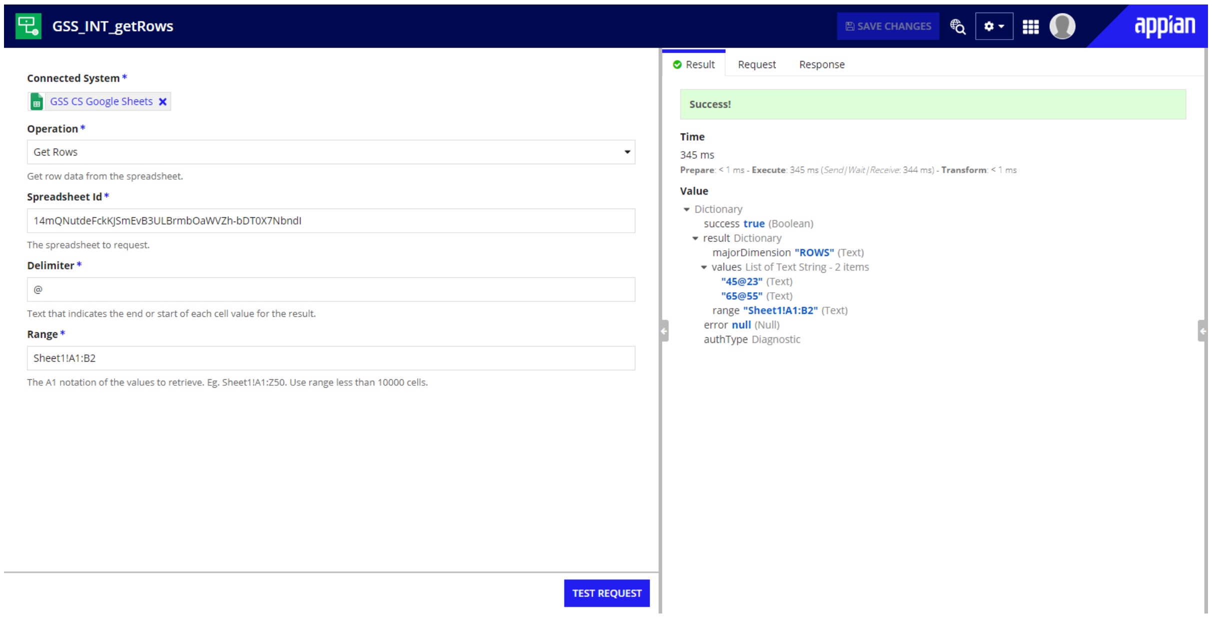
Value: Result

```
Dictionary
    spreadsheetId "14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI" (Text)
    tableRange "Sheet1!A1:B46" (Text)
  updates Dictionary
      updatedRange "Sheet1!A47:B48" (Text)
      updatedCells 4 (Number (Integer))
      spreadsheetId "14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI" (Text)
      updatedRows 2 (Number (Integer))
      updatedColumns 2 (Number (Integer))
```

Spreadsheet Id - Alphanumeric characters present in the sheet url
Ranges - Cell range in A1 notation.
Delimiter- Text that indicates the end or start of each cell value for the upcoming values field.
Values - List of text values in which each value represents row values. Each cell value is delimited by the delimiter.

Get Rows

Returns the cell data in the given range.



GSS_INT_getRows

Connected System *
GSS CS Google Sheets

Operation *
Get Rows
Get row data from the spreadsheet.

Spreadsheet Id *
14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI
The spreadsheet to request.

Delimiter *
@
Text that indicates the end or start of each cell value for the result.

Range *
Sheet1!A1:B2
The A1 notation of the values to retrieve. Eg. Sheet1!A1:Z50. Use range less than 10000 cells.

Success!

Time
345 ms
Prepare: < 1 ms - **Execute**: 345 ms (*Send/Wait/Receive*: 344 ms) - **Transform**: < 1 ms

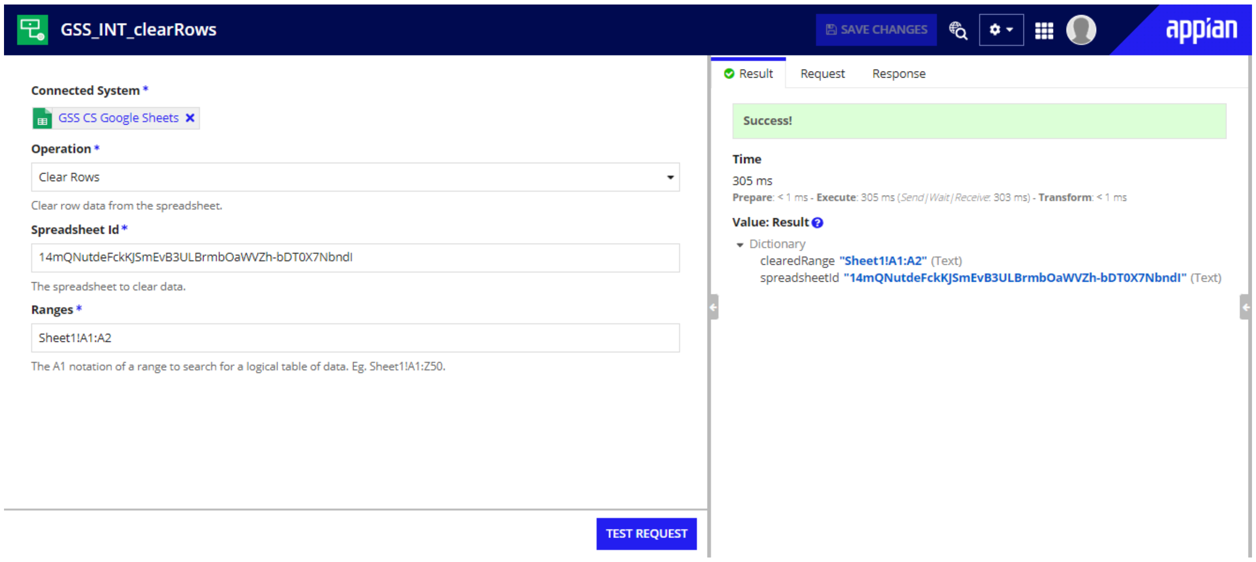
Value

```
▾ Dictionary
    success true (Boolean)
  ▾ result Dictionary
      majorDimension "ROWS" (Text)
    ▾ values List of Text String - 2 items
        "45@23" (Text)
        "65@55" (Text)
      range "Sheet1!A1:B2" (Text)
    error null (Null)
    authType Diagnostic
```

Spreadsheet Id - Alphanumeric characters present in the sheet url
Delimiter- Text that indicates the end or start of each cell value for the result.
Ranges - Cell range in A1 notation

Clear Rows

Clears cell values in the given range.



GSS_INT_clearRows

Connected System *
GSS CS Google Sheets

Operation *
Clear Rows
Clear row data from the spreadsheet.

Spreadsheet Id *
14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI
The spreadsheet to clear data.

Ranges *
Sheet1!A1:A2
The A1 notation of a range to search for a logical table of data. Eg. Sheet1!A1:Z50.

Success!

Time
305 ms
Prepare: < 1 ms - **Execute**: 305 ms (*Send/Wait/Receive*: 303 ms) - **Transform**: < 1 ms

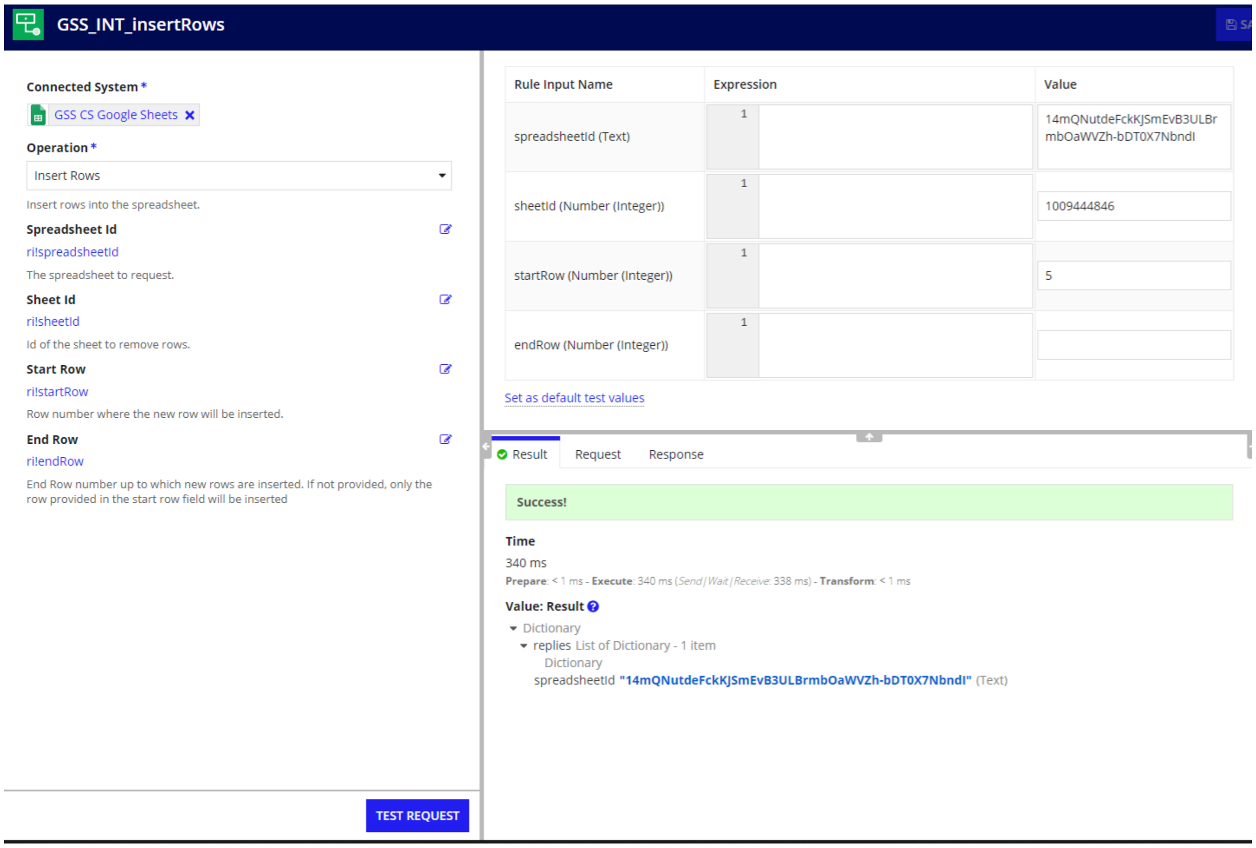
Value: Result

```
▾ Dictionary
    clearedRange "Sheet1!A1:A2" (Text)
    spreadsheetId "14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI" (Text)
```

Spreadsheet Id - Alphanumeric characters present in the sheet url
Ranges - Cell range in A1 notation

Insert Rows

Inserts blank rows in the sheet.


GSS_INT_insertRows

Connected System *
GSS CS Google Sheets

Operation *
Insert Rows

Insert rows into the spreadsheet.

Spreadsheet Id
ri!spreadsheetId
The spreadsheet to request.

Sheet Id
ri!sheetId
Id of the sheet to remove rows.

Start Row
ri!startRow
Row number where the new row will be inserted.

End Row
ri!endRow
End Row number up to which new rows are inserted. If not provided, only the row provided in the start row field will be inserted

Rule Input Name	Expression	Value
spreadsheetId (Text)	1	14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI
sheetId (Number (Integer))	1	1009444846
startRow (Number (Integer))	1	5
endRow (Number (Integer))	1	

Set as default test values

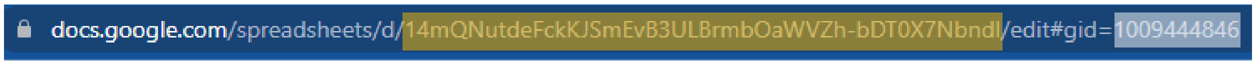
Result Request Response

Success!

Time
340 ms
Prepare: < 1 ms - Execute: 340 ms (Send/Wait/Receive: 338 ms) - Transform: < 1 ms

Value: Result
```
▾ Dictionary
  ▾ replies List of Dictionary - 1 item
      Dictionary
    spreadsheetId "14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI" (Text)
```

Spreadsheet Id - Alphanumeric characters present in the sheet url
Sheet Id - number in the url which is after "gid=".

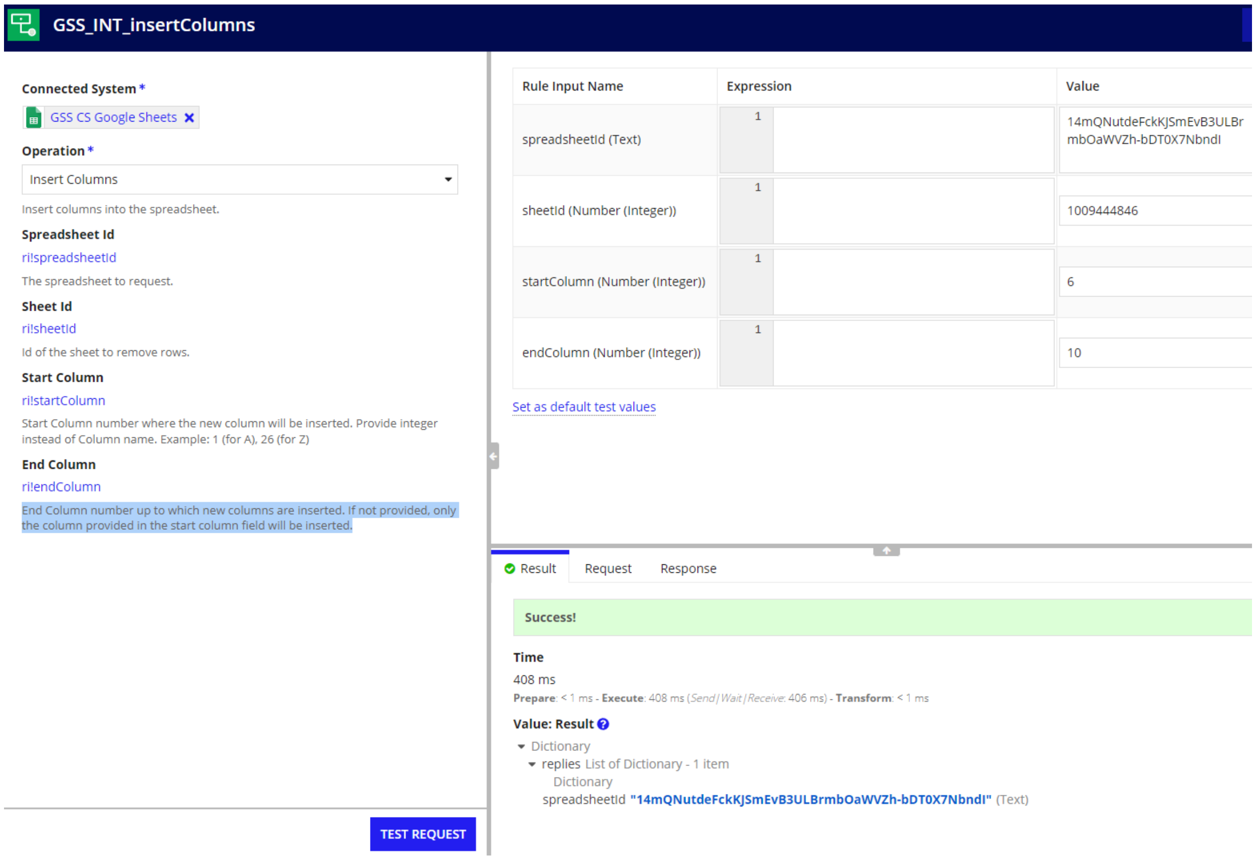

docs.google.com/spreadsheets/d/14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI/edit#gid=1009444846

In the above image, the text highlighted in yellow is Spreadsheet Id and the text highlighted in white is the Sheet Id
Start Row - Row number where the new row will be inserted.
End Row - End Row number up to which new rows are inserted. If not provided, only the row provided in the start row field will be inserted.

Example: If you want to insert 2 blank rows after row 4, then 5 should be provided in the start Row field and 6 should be provided in the End Row field.

Insert Columns

Inserts blank columns in the sheet.



GSS_INT_insertColumns

Connected System *

GSS CS Google Sheets

Operation *

Insert Columns

Insert columns into the spreadsheet.

Spreadsheet Id

ri!spreadsheetId

The spreadsheet to request.

Sheet Id

ri!sheetId

Id of the sheet to remove rows.

Start Column

ri!startColumn

Start Column number where the new column will be inserted. Provide integer instead of Column name. Example: 1 (for A), 26 (for Z)

End Column

ri!endColumn

End Column number up to which new columns are inserted. If not provided, only the column provided in the start column field will be inserted.

Rule Input Name	Expression	Value
spreadsheetId (Text)	1	14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI
sheetId (Number (Integer))	1	1009444846
startColumn (Number (Integer))	1	6
endColumn (Number (Integer))	1	10

Set as default test values

Result Request Response

Success!

Time

408 ms

Prepare: < 1 ms - **Execute**: 408 ms (*Send / Wait / Receive*: 406 ms) - **Transform**: < 1 ms

Value: Result

```
▾ Dictionary
  ▾ replies List of Dictionary - 1 item
      Dictionary
    spreadsheetId "14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI" (Text)
```

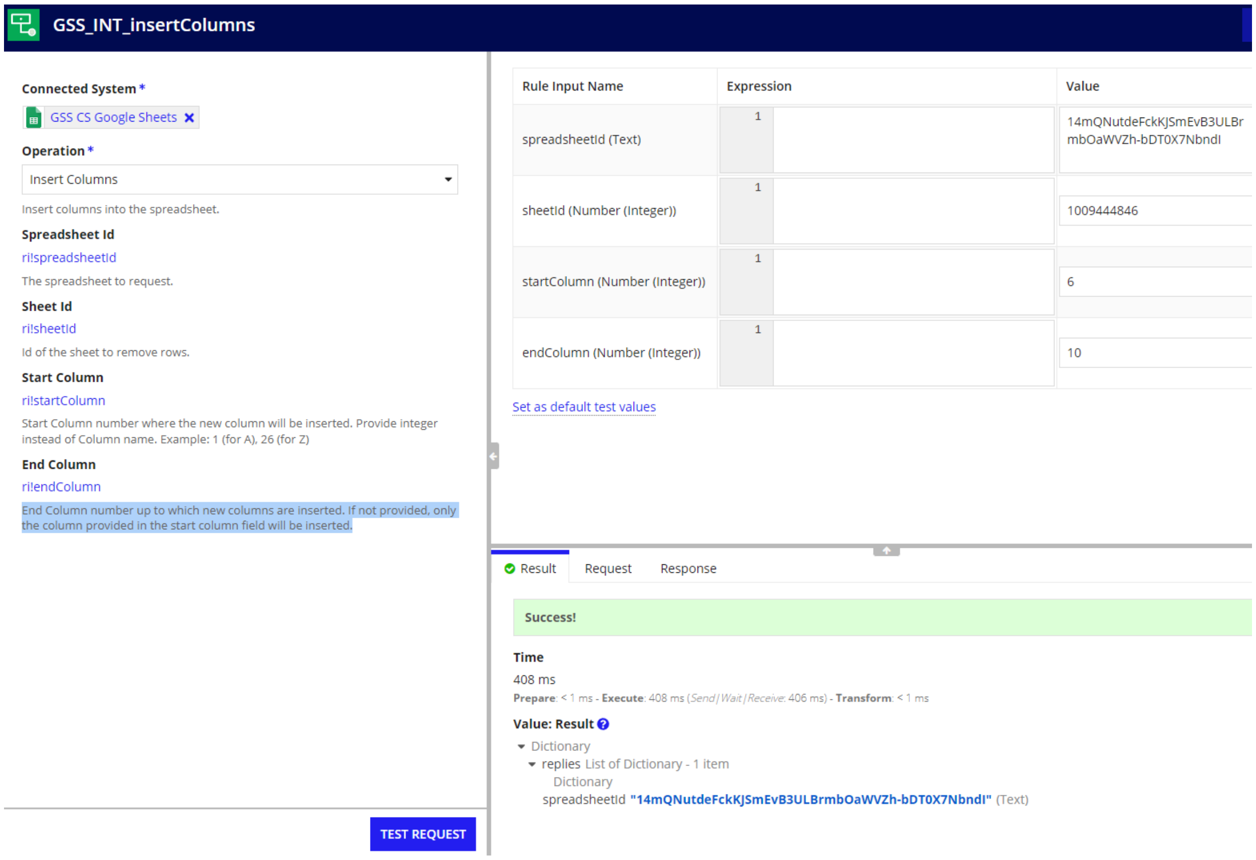
TEST REQUEST

Spreadsheet Id - Alphanumeric characters present in the sheet url
Sheet Id - number in the url which is after "gid=".
Start Column- Start Column number where the new column will be inserted. Provide integer instead of Column name. Example: 1 (for A), 26 (for Z).
End Column- End Column number up to which new columns are inserted. If not provided, only the column provided in the start column field will be inserted..

Example: If you want to insert 3 blank columns after column F, then 7 should be provided in the start Column field and 9 should be provided in the End Column field.

Delete Rows

Deletes rows in the sheet.



GSS_INT_insertColumns

Connected System *
GSS CS Google Sheets

Operation *
Insert Columns

Insert columns into the spreadsheet.

Spreadsheet Id
ri!spreadsheetId
The spreadsheet to request.

Sheet Id
ri!sheetId
Id of the sheet to remove rows.

Start Column
ri!startColumn
Start Column number where the new column will be inserted. Provide integer instead of Column name. Example: 1 (for A), 26 (for Z)

End Column
ri!endColumn
End Column number up to which new columns are inserted. If not provided, only the column provided in the start column field will be inserted.

Rule Input Name	Expression	Value
spreadsheetId (Text)	1	14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI
sheetId (Number (Integer))	1	1009444846
startColumn (Number (Integer))	1	6
endColumn (Number (Integer))	1	10

Set as default test values

Result | Request | Response

Success!

Time
408 ms
Prepare: < 1 ms - Execute: 408 ms (Send/Wait/Receive: 406 ms) - Transform: < 1 ms

Value: Result
Dictionary
replies List of Dictionary - 1 item
Dictionary
spreadsheetId "14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI" (Text)

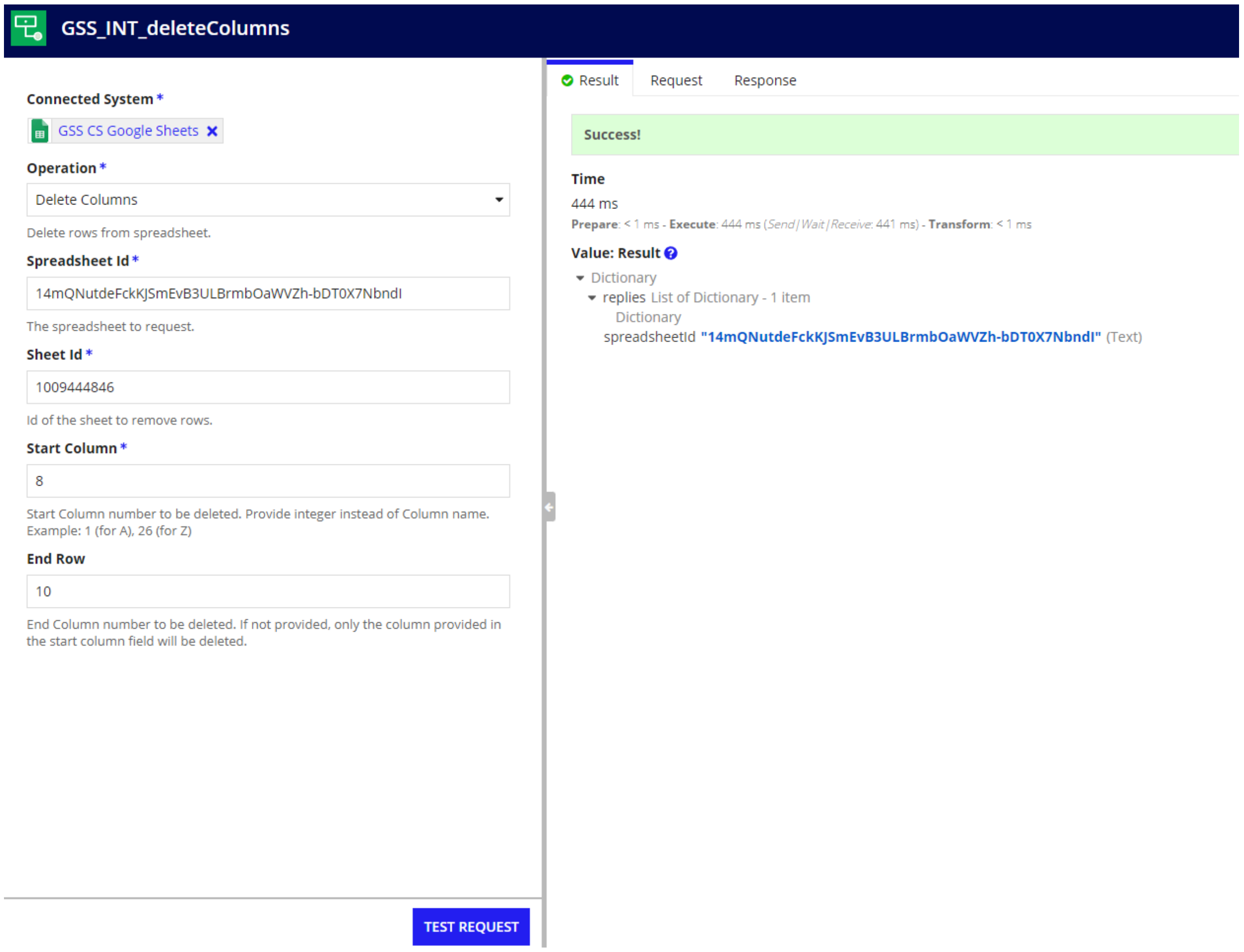
TEST REQUEST

Spreadsheet Id - Alphanumeric characters present in the sheet url
Sheet Id - number in the url which is after "gid=".
Start Row - Start Row number to be deleted..
End Row - End Row number to be deleted. If not provided, only the row provided in the start row field will be deleted.

Example: If you want to delete rows 7 to 10, then 7 should be provided in the start Row field and 10 should be provided in the End Row field.

Delete Columns

Deletes columns in the sheet.


GSS_INT_deleteColumns

Connected System *
GSS CS Google Sheets

Operation *
Delete Columns
Delete rows from spreadsheet.

Spreadsheet Id *
14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI
The spreadsheet to request.

Sheet Id *
1009444846
Id of the sheet to remove rows.

Start Column *
8
Start Column number to be deleted. Provide integer instead of Column name. Example: 1 (for A), 26 (for Z)

End Row
10
End Column number to be deleted. If not provided, only the column provided in the start column field will be deleted.

TEST REQUEST

Result | Request | Response

Success!

Time
444 ms
Prepare: < 1 ms - Execute: 444 ms (Send/Wait/Receive: 441 ms) - Transform: < 1 ms

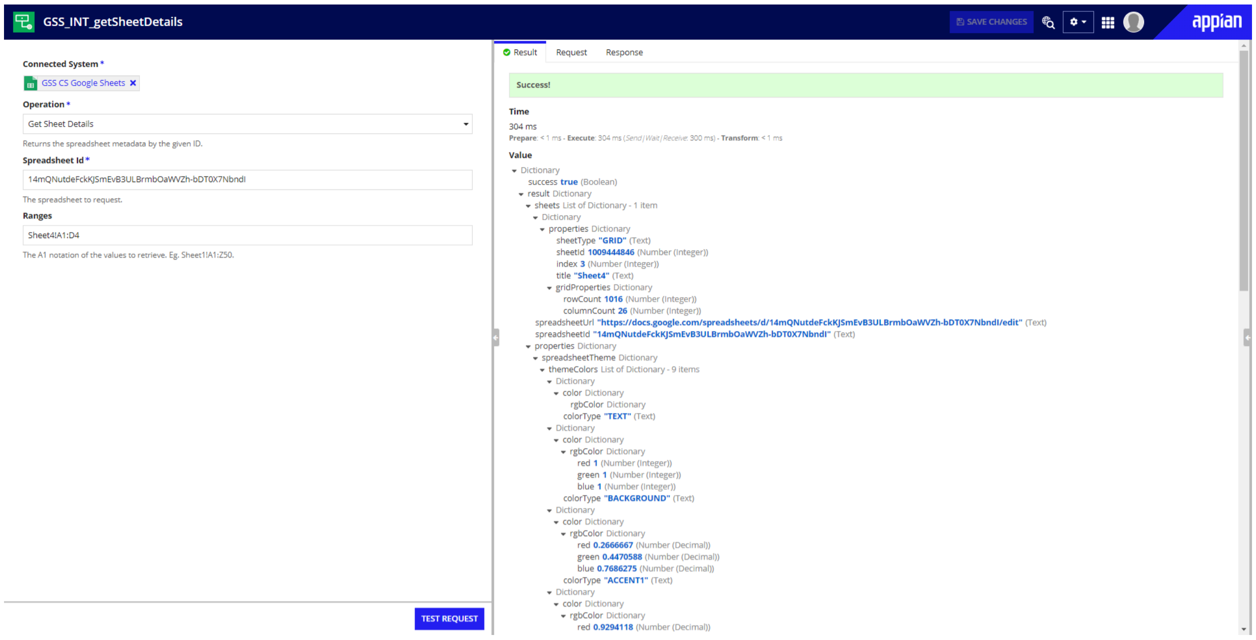
Value: Result
```
Dictionary
  replies List of Dictionary - 1 item
    Dictionary
  spreadsheetId "14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI" (Text)
```

Spreadsheet Id - Alphanumeric characters present in the sheet url
Sheet Id - number in the url which is after "gid=".
Start Column- Start Column number to be deleted. Provide integer instead of Column name. Example: 1 (for A), 26 (for Z).
End Column- End Column number to be deleted. If not provided, only the column provided in the start column field will be deleted.

Example: If you want to delete columns from C to E , then 3 should be provided in the start Column field and 5 should be provided in the End Column field.

Get Sheet Details

Returns the metadata about the spreadsheet.



GSS_INT_getSheetDetails

Connected System *

GSS CS Google Sheets

Operation *

Get Sheet Details

Returns the spreadsheet metadata by the given ID.

Spreadsheet Id *

14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI

The spreadsheet to request.

Ranges

Sheet4!A1:D4

The A1 notation of the values to retrieve. Eg. Sheet1!A1:Z50.

Result | Request | Response

Success!

Time

304 ms

Prepare: < 1 ms - **Execute**: 304 ms (*Send/Wait/Receive*: 300 ms) - **Transform**: < 1 ms

Value

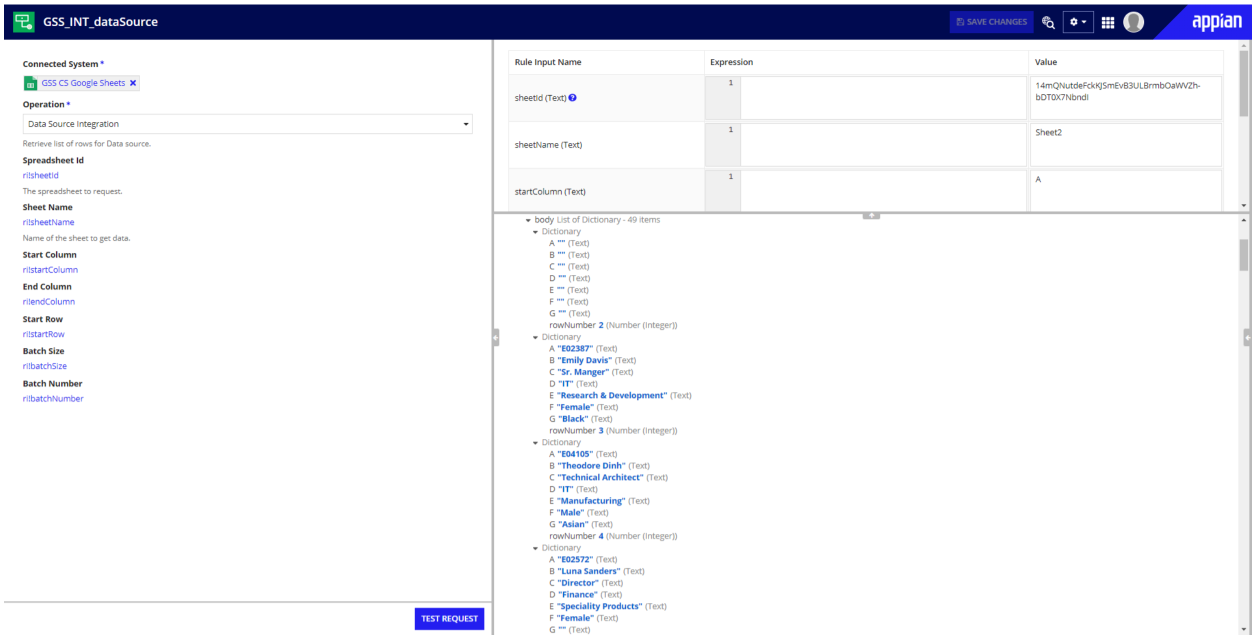
```
▾ Dictionary
    success true (Boolean)
  ▾ result Dictionary
    ▾ sheets List of Dictionary - 1 item
      ▾ Dictionary
        ▾ properties Dictionary
            sheetType "GRID" (Text)
            sheetId 1009444846 (Number (Integer))
            index 3 (Number (Integer))
            title "Sheet4" (Text)
          ▾ gridProperties Dictionary
              rowCount 1016 (Number (Integer))
              columnCount 26 (Number (Integer))
      spreadsheetUrl "https://docs.google.com/spreadsheets/d/14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI/edit" (Text)
      spreadsheetId "14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI" (Text)
    ▾ properties Dictionary
      ▾ spreadsheetTheme Dictionary
        ▾ themeColors List of Dictionary - 9 items
          ▾ Dictionary
            ▾ color Dictionary
                rgbColor Dictionary
              colorType "TEXT" (Text)
          ▾ Dictionary
            ▾ color Dictionary
              ▾ rgbColor Dictionary
                  red 1 (Number (Integer))
                  green 1 (Number (Integer))
                  blue 1 (Number (Integer))
              colorType "BACKGROUND" (Text)
          ▾ Dictionary
            ▾ color Dictionary
              ▾ rgbColor Dictionary
                  red 0.2666667 (Number (Decimal))
                  green 0.4470588 (Number (Decimal))
                  blue 0.7686275 (Number (Decimal))
              colorType "ACCENT1" (Text)
          ▾ Dictionary
            ▾ color Dictionary
              ▾ rgbColor Dictionary
                  red 0.9294118 (Number (Decimal))
```

Spreadsheet Id - Alphanumeric characters present in the sheet url

Ranges - Cell range in A1 notation

Data Source Integration

Returns cell values for each row in batches



GSS_INT_dataSource

Connected System *

GSS CS Google Sheets

Operation *

Data Source Integration

Retrieve list of rows for Data source.

Spreadsheet Id

ri!sheetId

The spreadsheet to request.

Sheet Name

ri!sheetName

Name of the sheet to get data.

Start Column

ri!startColumn

End Column

ri!endColumn

Start Row

ri!startRow

Batch Size

ri!batchSize

Batch Number

ri!batchNumber

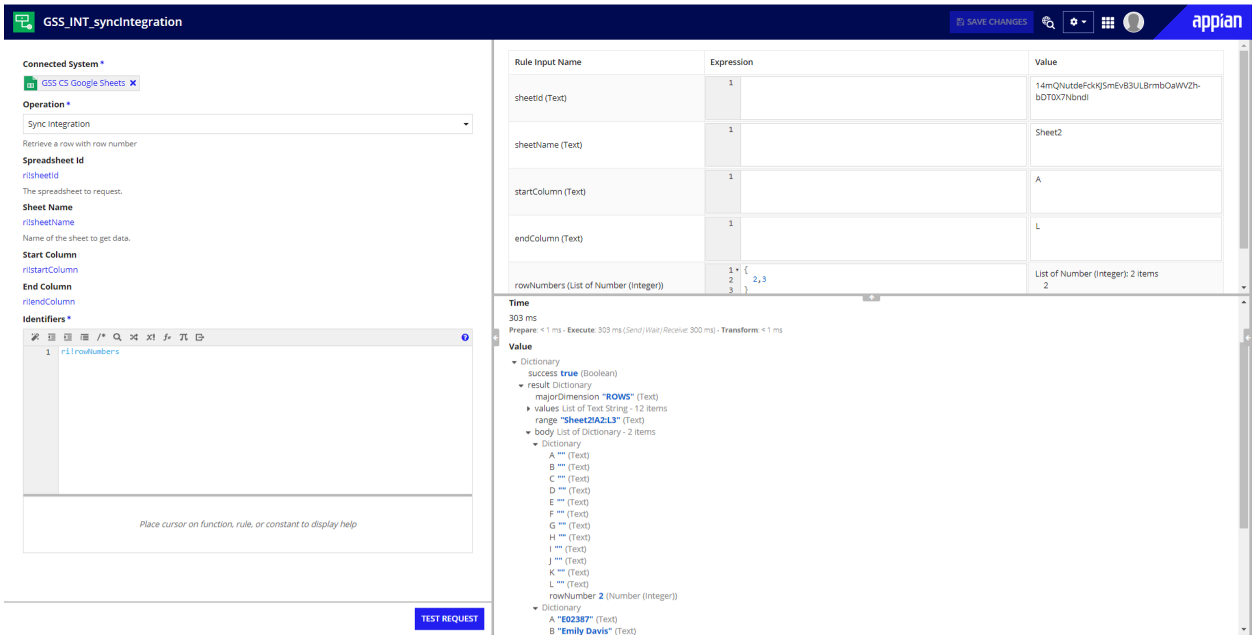
Rule Input Name	Expression	Value
sheetId (Text)	1	14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI
sheetName (Text)	1	Sheet2
startColumn (Text)	1	A

```
▾ body List of Dictionary - 49 items
  ▾ Dictionary
      A "" (Text)
      B "" (Text)
      C "" (Text)
      D "" (Text)
      E "" (Text)
      F "" (Text)
      G "" (Text)
      rowNumber 2 (Number (Integer))
  ▾ Dictionary
      A "E02387" (Text)
      B "Emily Davis" (Text)
      C "Sr. Manger" (Text)
      D "IT" (Text)
      E "Research & Development" (Text)
      F "Female" (Text)
      G "Black" (Text)
      rowNumber 3 (Number (Integer))
  ▾ Dictionary
      A "E04105" (Text)
      B "Theodore Dinh" (Text)
      C "Technical Architect" (Text)
      D "IT" (Text)
      E "Manufacturing" (Text)
      F "Male" (Text)
      G "Asian" (Text)
      rowNumber 4 (Number (Integer))
  ▾ Dictionary
      A "E02572" (Text)
      B "Luna Sanders" (Text)
      C "Director" (Text)
      D "Finance" (Text)
      E "Speciality Products" (Text)
      F "Female" (Text)
      G "" (Text)
```

Spreadsheet Id - Alphanumeric characters present in the sheet url
Sheet Name- Name of the current sheet. Example: Sheet2.
Start Column - Column name of the starting column for which data sourcing is performed. Example: F
End Column - Column name of the ending column for which data sourcing is performed.
Start Row - Row number of the starting row. Example: If you do not want row values up to 6th row, then 7 should be given in the start row field.
Batch Size - Number of row data that should be returned in a single call.
Batch Number - Number of batch that should be returned
Example: If we want a second batch of batch size 10, then 2 should be provided in the Batch Number field, 10 should be provided in the batch size field and the result will be rows from 11 to 20.

Sync Integration

Returns cell values for the specified rows.



Connected System * GSS CS Google Sheets
Operation * Sync Integration
Retrieve a row with row number
Spreadsheet Id
ri!sheetId
The spreadsheet to request.
Sheet Name
ri!sheetName
Name of the sheet to get data.
Start Column
ri!startColumn
End Column
ri!endColumn
Identifiers *
```
ri!rowNumbers
```
Place cursor on function, rule, or constant to display help

Rule Input Name	Expression	Value
sheetId (Text)		14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI
sheetName (Text)		Sheet2
startColumn (Text)		A
endColumn (Text)		L
rowNumbers (List of Number (Integer))	{ 2,3 }	List of Number (Integer): 2 items 2

Time
303 ms
Prepare: < 1 ms - **Execute**: 303 ms (*Send / Wait / Receive*: 300 ms) - **Transform**: < 1 ms
Value
```
▾ Dictionary
    success true (Boolean)
  ▾ result Dictionary
      majorDimension "ROWS" (Text)
    ▸ values List of Text String - 12 items
      range "Sheet2!A2:L3" (Text)
    ▾ body List of Dictionary - 2 items
      ▾ Dictionary
          A "" (Text)
          B "" (Text)
          C "" (Text)
          D "" (Text)
          E "" (Text)
          F "" (Text)
          G "" (Text)
          H "" (Text)
          I "" (Text)
          J "" (Text)
          K "" (Text)
          L "" (Text)
          rowNumber 2 (Number (Integer))
      ▾ Dictionary
          A "E02387" (Text)
          B "Emily Davis" (Text)
```

Spreadsheet Id - Alphanumeric characters present in the sheet url
Sheet Name- Name of the current sheet. Example: Sheet2.
Start Column - Column name of the starting column for which data sourcing is performed. Example: F
End Column - Column name of the ending column for which data sourcing is performed.
Row Numbers - List of Row numbers. Only the specified row data is returned in the results

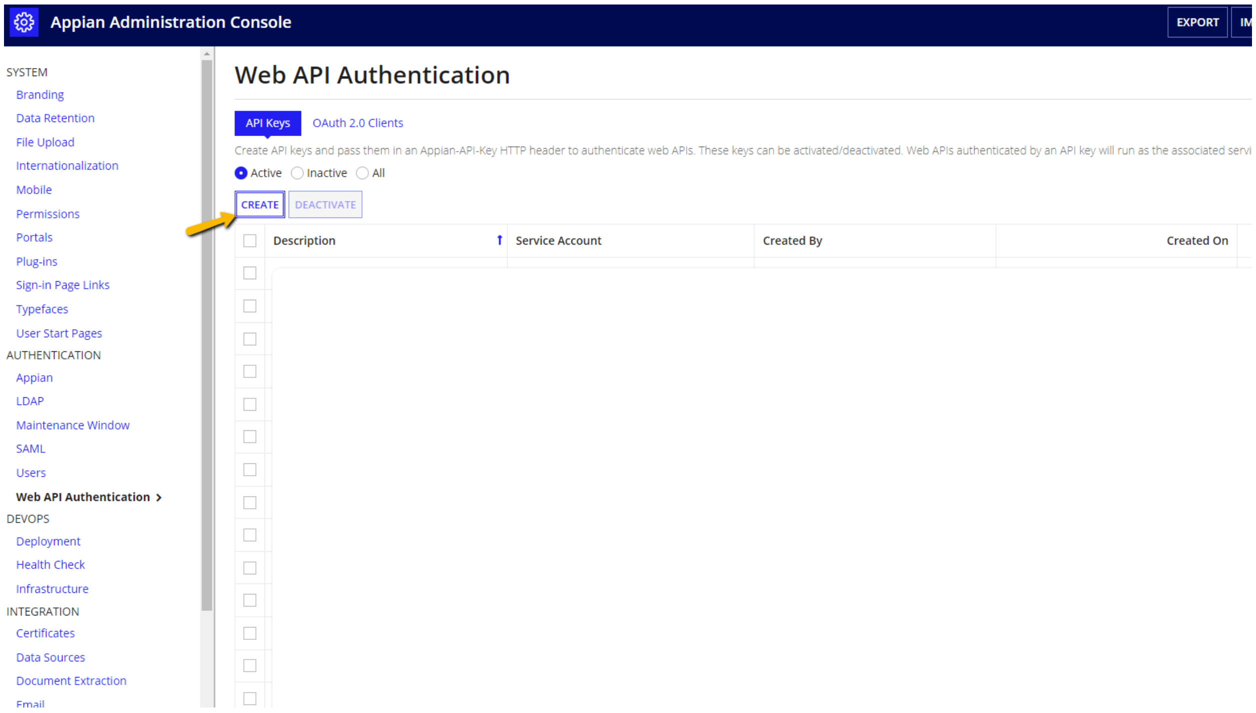
Record and Webhook Configuration

Appian API Key and Service Account Configuration

API keys are needed to be created from Appian which acts as authentication when Google sheets sends webhooks to Appian.

Below are the steps to create and configure API Key and Service account in the Sample Application.

1. Navigate to **Admin console** → **Web API Authentication**
2. Select **Create** under API Keys



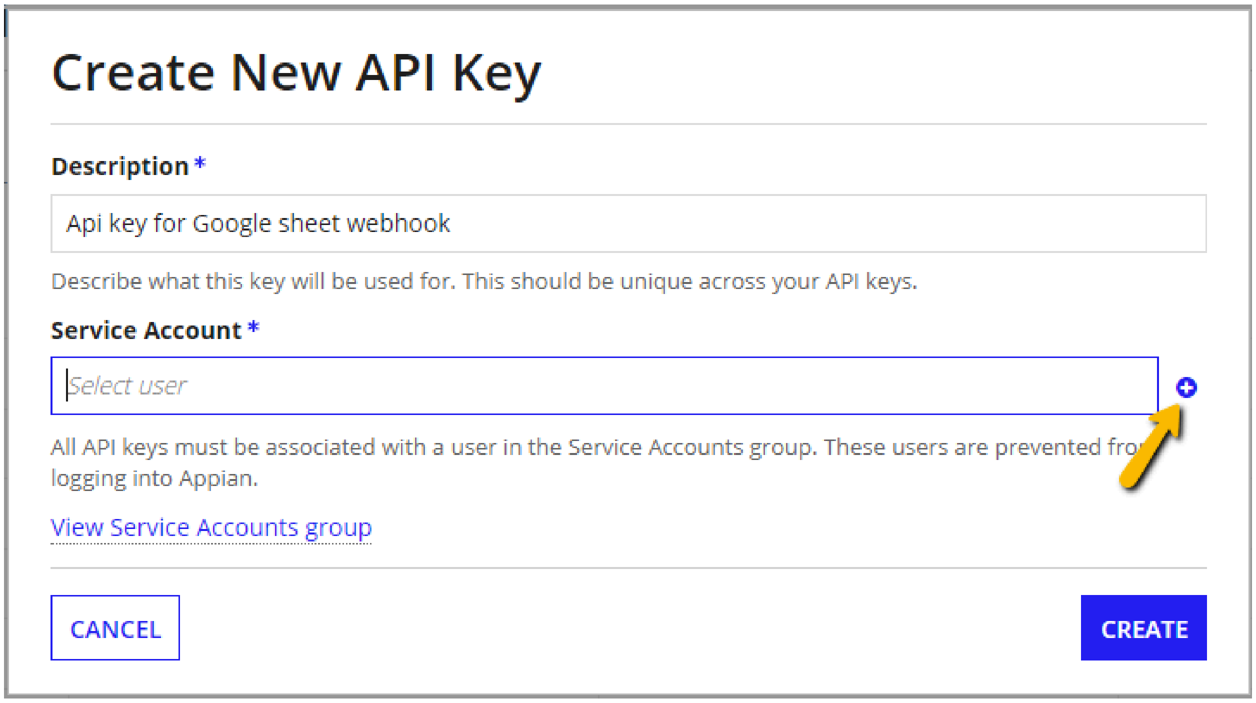
Web API Authentication

Create API keys and pass them in an Appian-API-Key HTTP header to authenticate web APIs. These keys can be activated/deactivated. Web APIs authenticated by an API key will run as the associated servi

[✓ Active] Inactive All

	Description	Service Account	Created By	Created On

3. Enter description for the API key under **Description** and select service account if you already have one. Otherwise, select "+" icon



Create New API Key

Description *

Api key for Google sheet webhook

Describe what this key will be used for. This should be unique across your API keys.

Service Account *

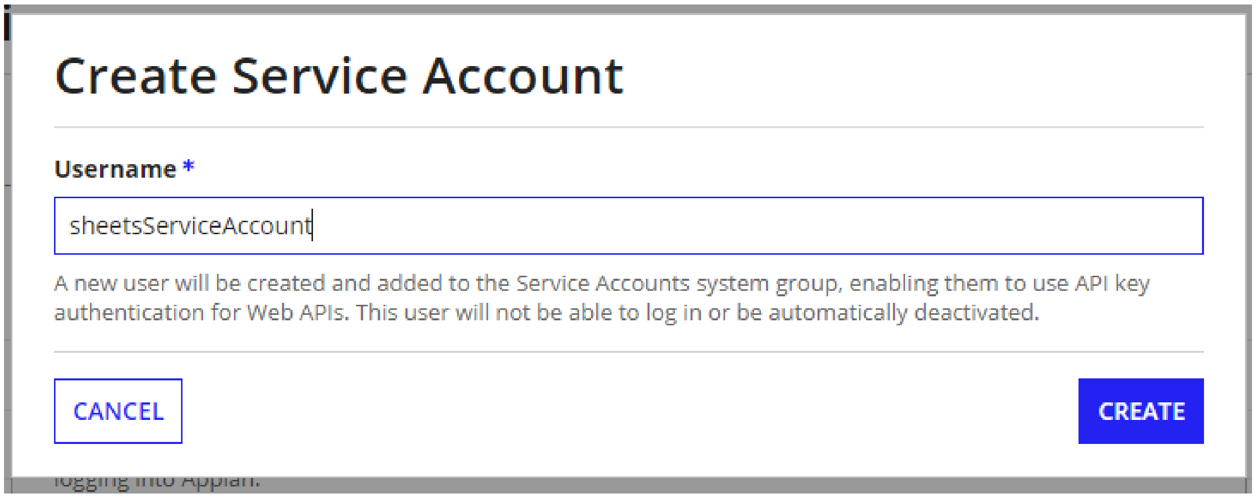
Select user

All API keys must be associated with a user in the Service Accounts group. These users are prevented fro
logging into Appian.

View Service Accounts group

CANCEL CREATE

4. Enter your username in the next screen and click **create.**



Create Service Account

Username *

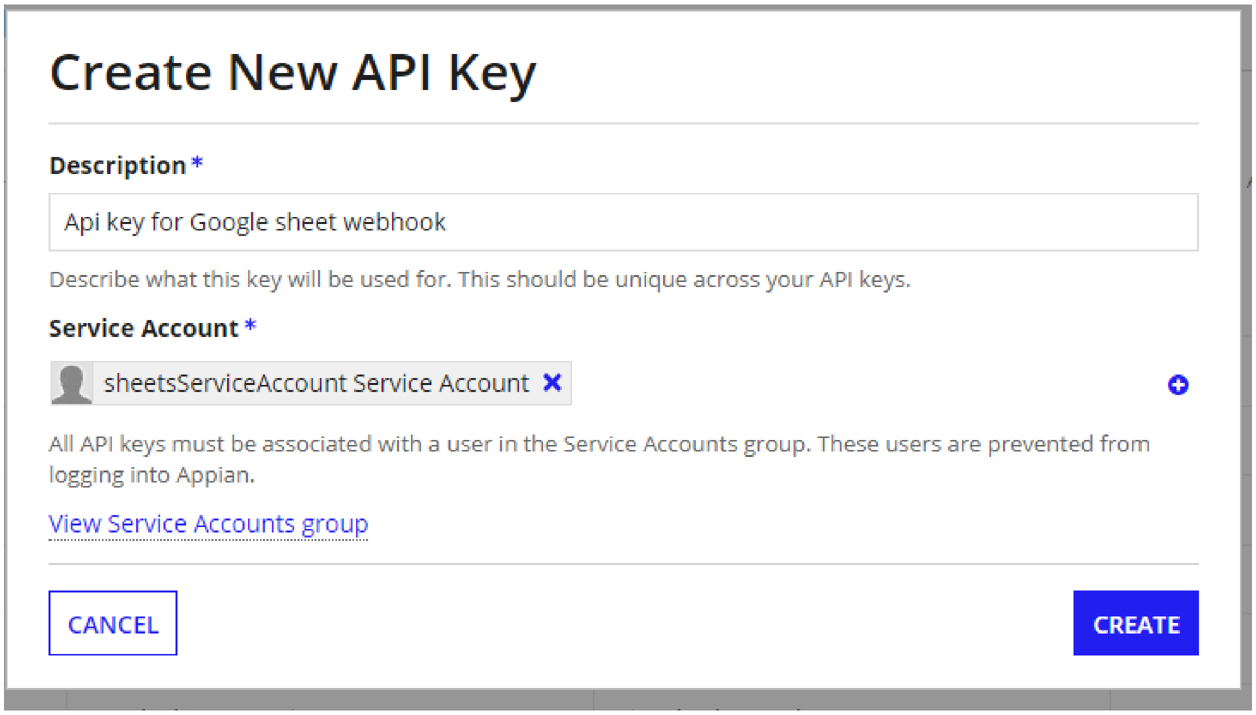
sheetsServiceAccount

A new user will be created and added to the Service Accounts system group, enabling them to use API key authentication for Web APIs. This user will not be able to log in or be automatically deactivated.

CANCEL CREATE

logging into Appian.

5. Select **create**



Create New API Key

Description *

Api key for Google sheet webhook

Describe what this key will be used for. This should be unique across your API keys.

Service Account *

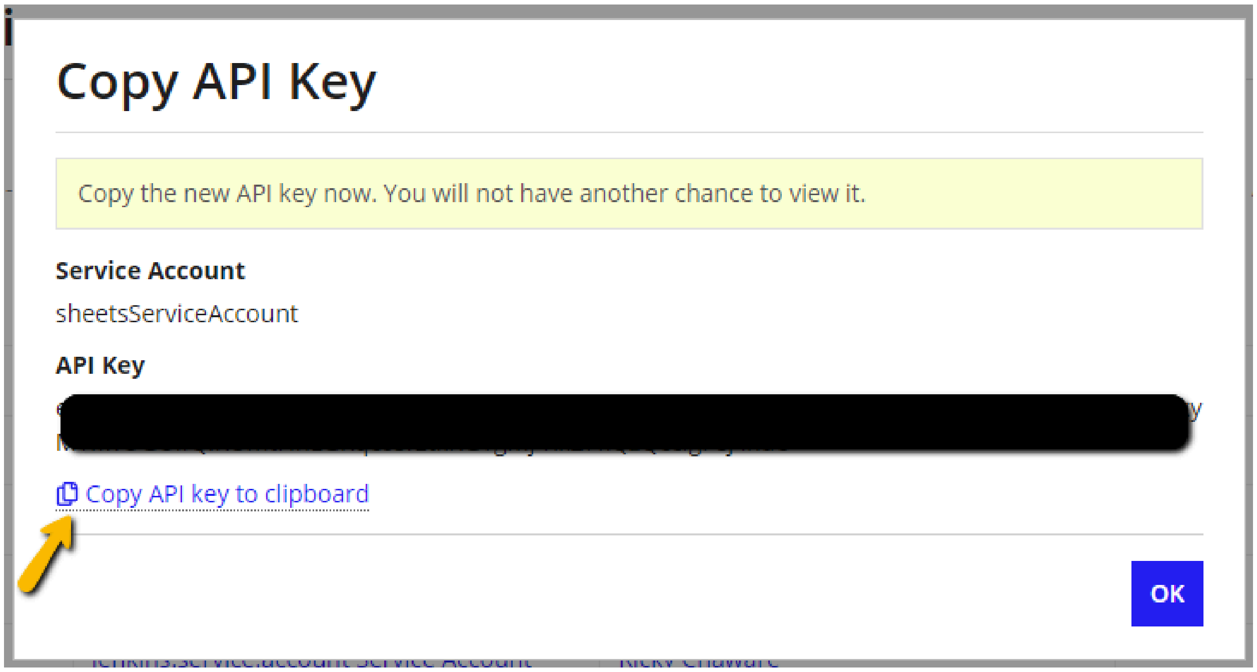
sheetsServiceAccount Service Account ✕

All API keys must be associated with a user in the Service Accounts group. These users are prevented from logging into Appian.

View Service Accounts group

CANCEL CREATE

6. Copy the API key from the next screen which will be used during webhook configuration



Copy API Key

Copy the new API key now. You will not have another chance to view it.

Service Account

sheetsServiceAccount

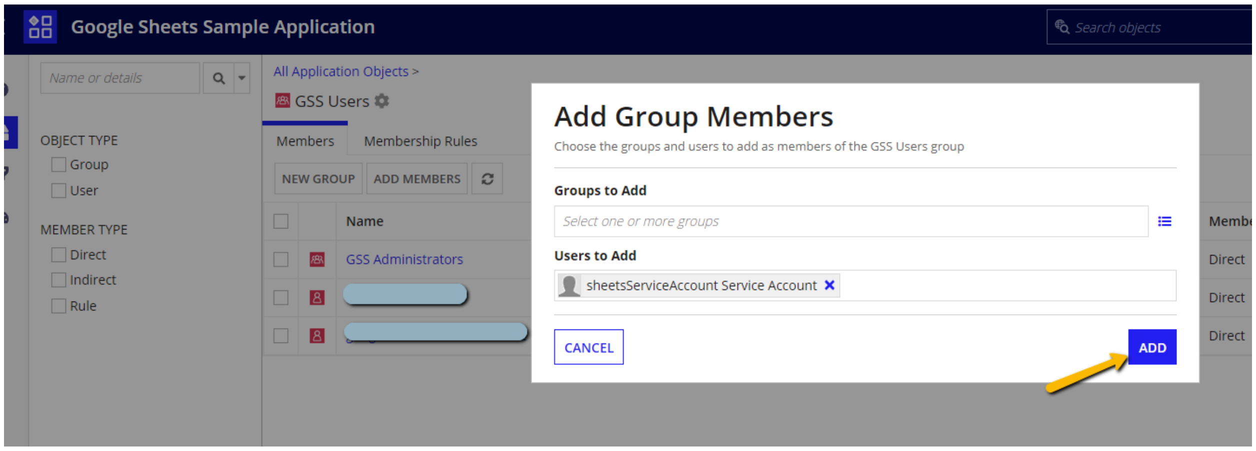
API Key

[redacted]

Copy API key to clipboard

OK

7. Add the service account to the users group after which the API key setup will be complete.



Add Group Members

Choose the groups and users to add as members of the GSS Users group

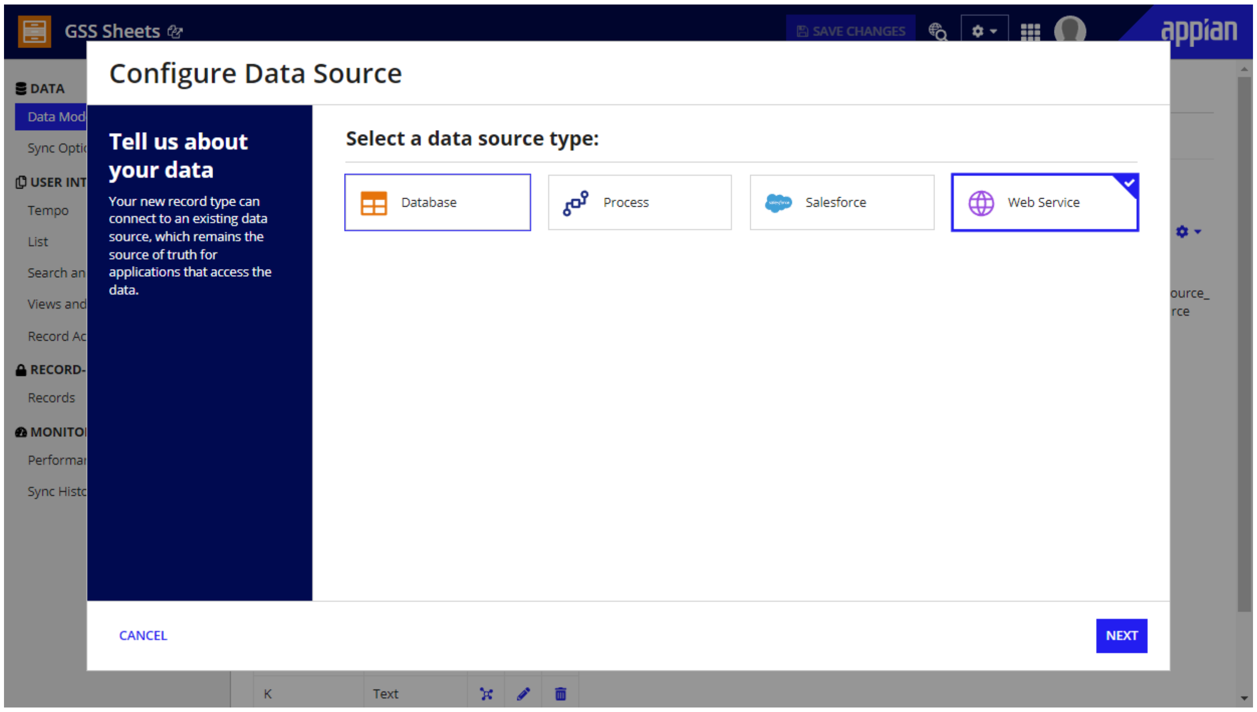
Groups to Add: Select one or more groups

Users to Add: sheetsServiceAccount Service Account

CANCEL ADD

Data Source Configuration

1. After creating a new Record Type in Appian, in the Data Model Tab, select **Tell us About your Data** button and select **Web service** and click Next.



Configure Data Source

Tell us about your data

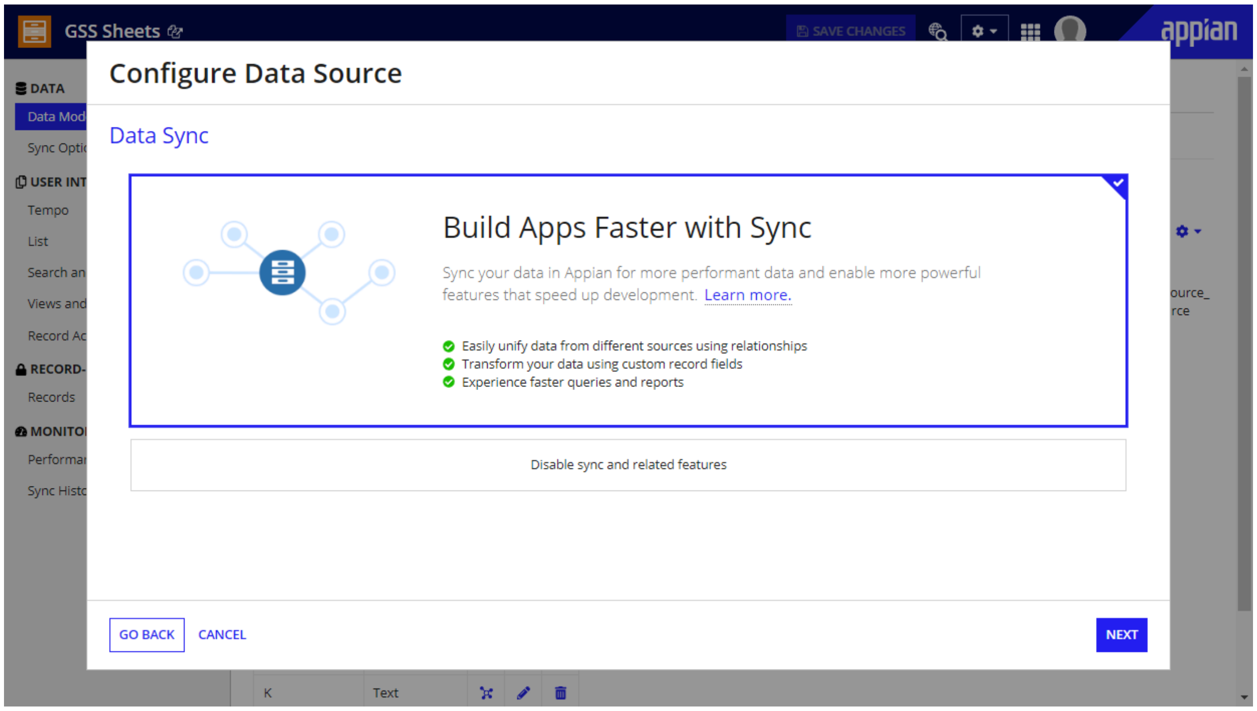
Your new record type can connect to an existing data source, which remains the source of truth for applications that access the data.

Select a data source type:

Database | Process | Salesforce | [✓ Web Service]

CANCEL NEXT

2. Turn on Sync Feature and select Next.



Configure Data Source

Data Sync

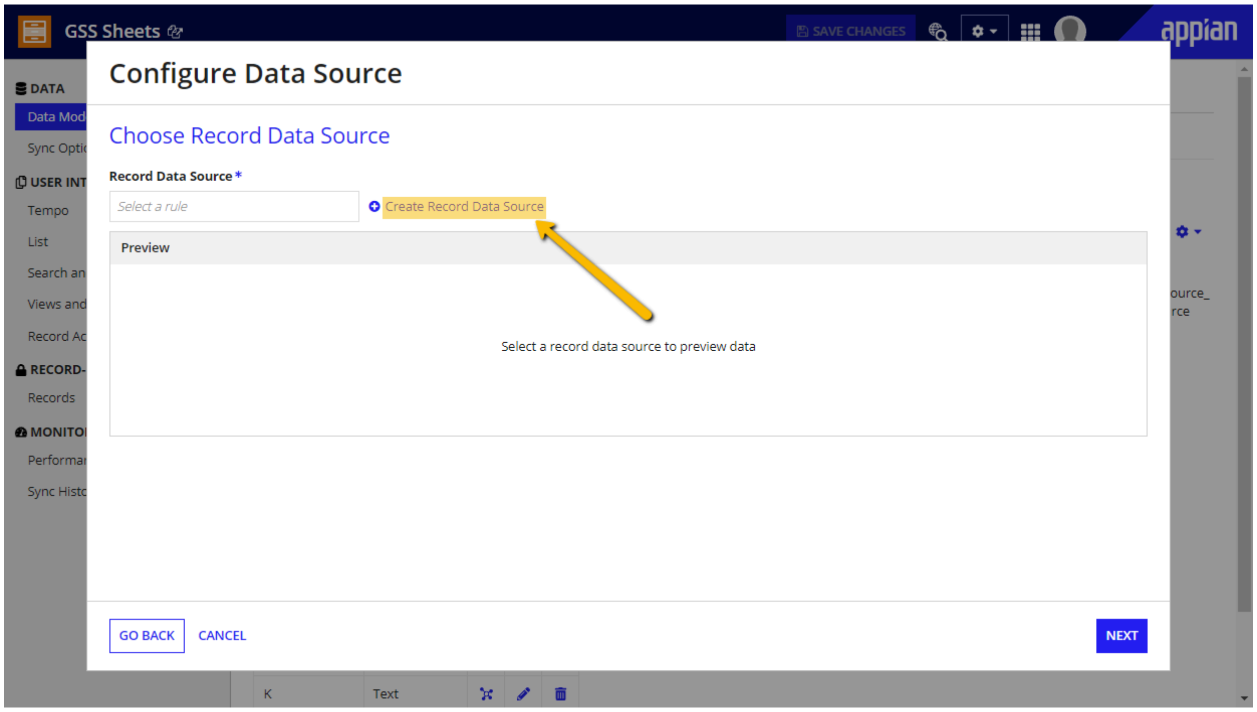
Build Apps Faster with Sync

Sync your data in Appian for more performant data and enable more powerful features that speed up development. Learn more.

- Easily unify data from different sources using relationships
- Transform your data using custom record fields
- Experience faster queries and reports

Disable sync and related features

3. If there is no Data Source Rule previously configured, select **Create Record Data Source.** Otherwise, enter the rule name in the search box.



Configure Data Source

Choose Record Data Source

Record Data Source *

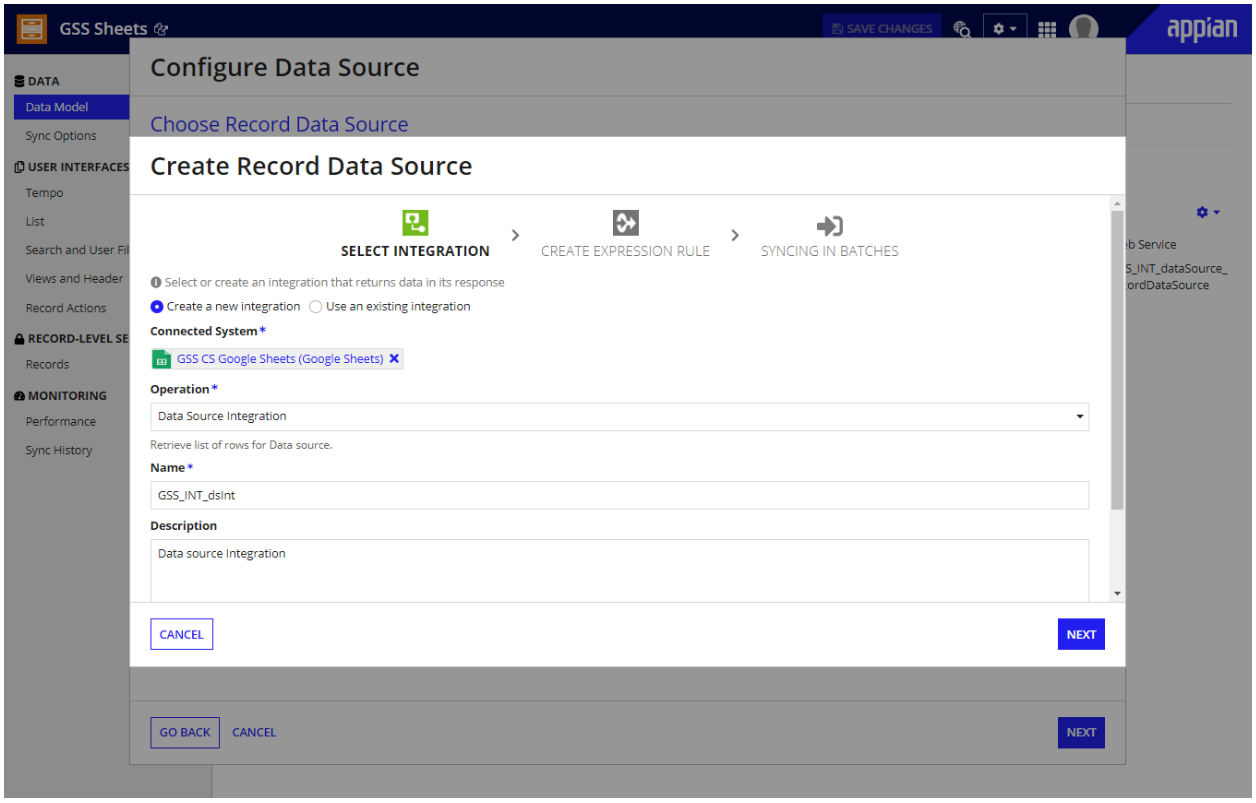
Select a rule

Create Record Data Source

Preview

Select a record data source to preview data

4. In **Create Record Data Source,** select the Data Source Integration from the connected system and enter the required details.



Select or create an integration that returns data in its response

[✓ Create a new integration] Use an existing integration

Connected System *
GSS CS Google Sheets (Google Sheets)

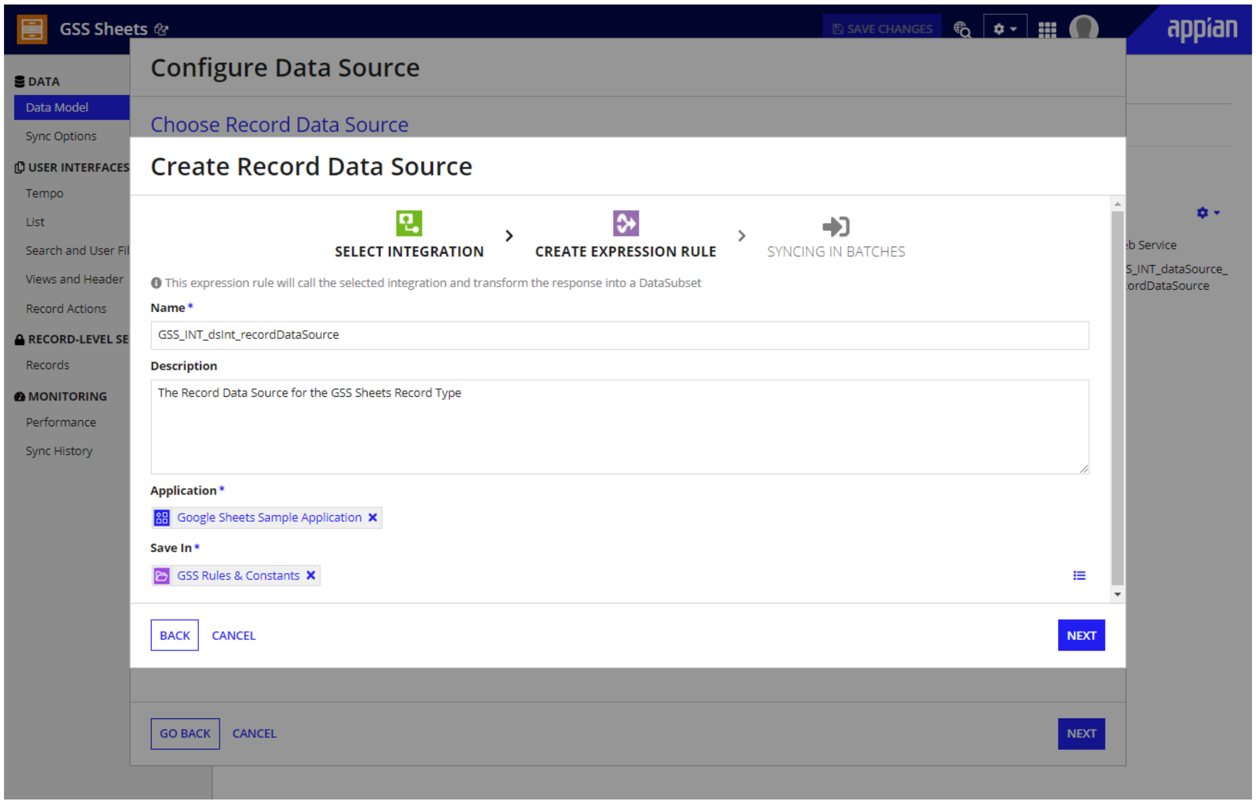
Operation *
Data Source Integration

Retrieve list of rows for Data source.

Name *
GSS_INT_dsInt

Description
Data source Integration

5. In the next page, the expression rule details will be automatically populated. Select **Next.**



This expression rule will call the selected integration and transform the response into a DataSubset

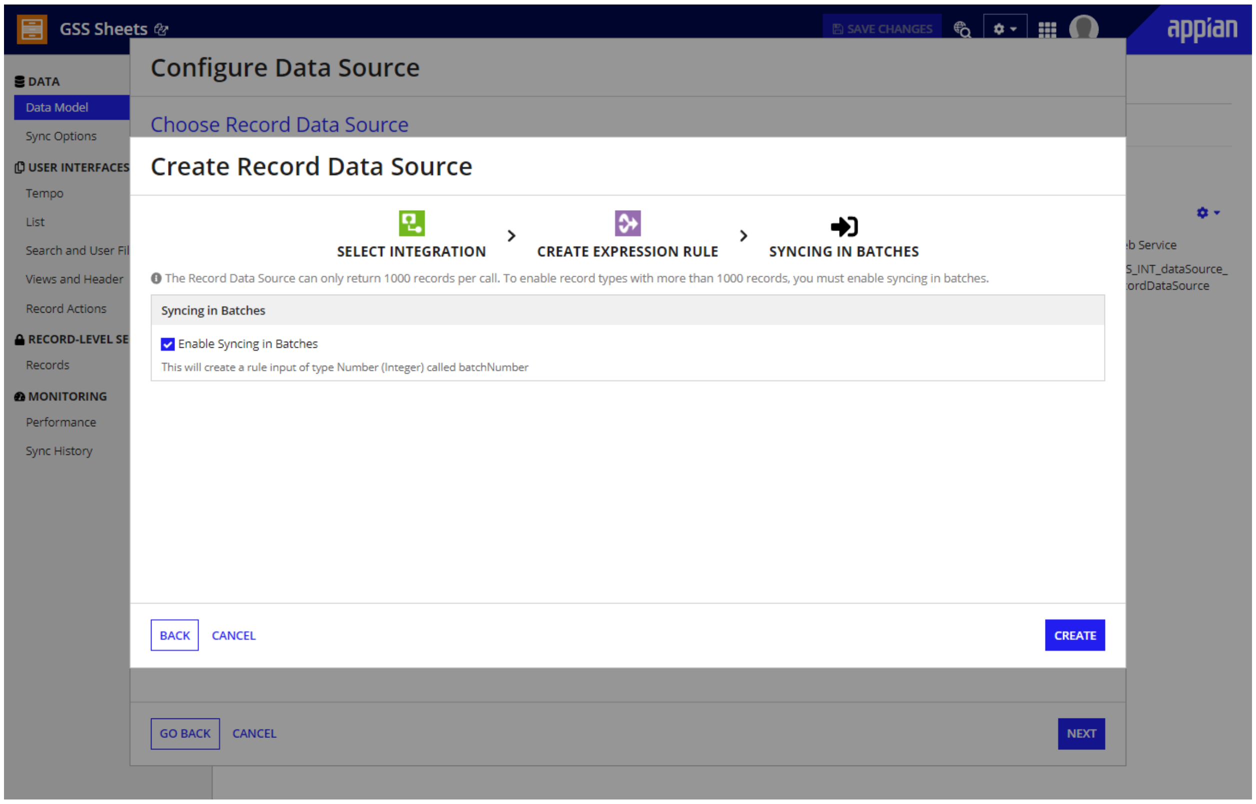
Name *
GSS_INT_dsInt_recordDataSource

Description
The Record Data Source for the GSS Sheets Record Type

Application *
Google Sheets Sample Application

Save In *
GSS Rules & Constants

6. Check the **Enable syncing in batches** checkbox. Select **Create.**


Create Record Data Source

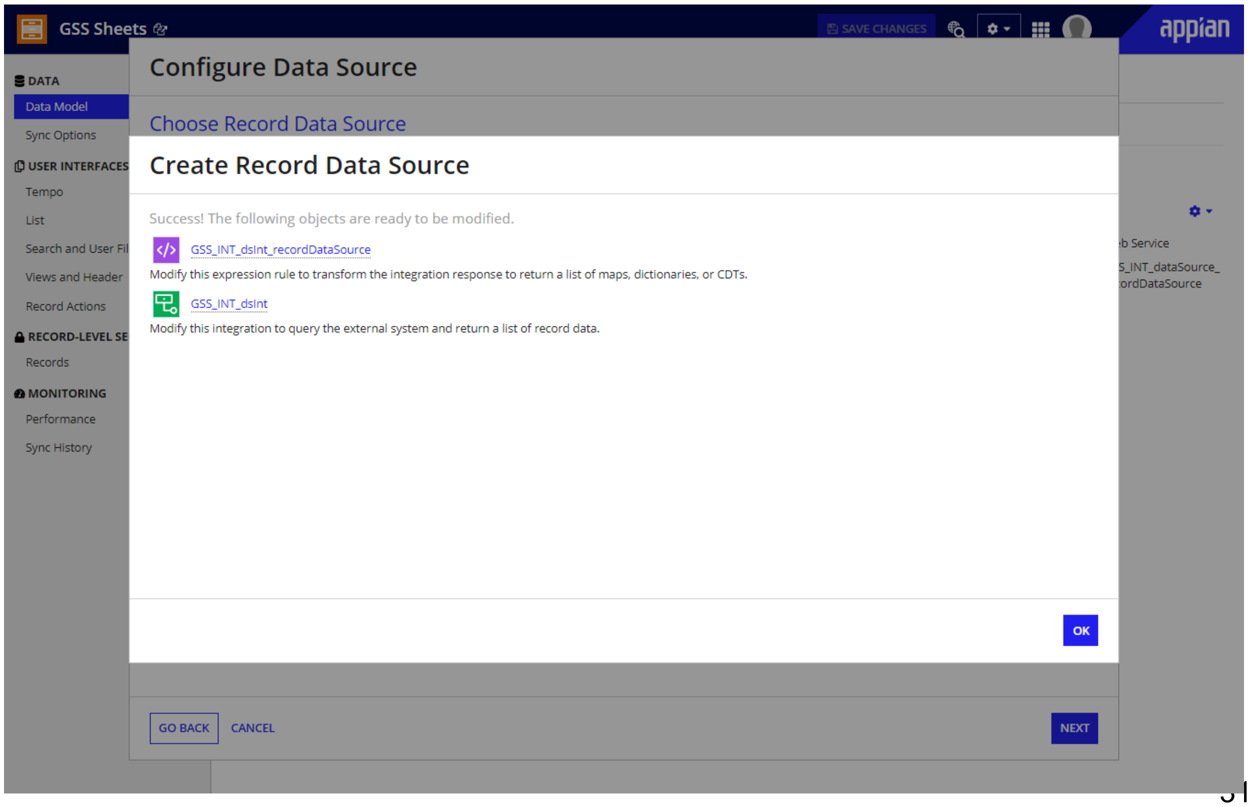
The Record Data Source can only return 1000 records per call. To enable record types with more than 1000 records, you must enable syncing in batches.

Syncing in Batches

☑ Enable Syncing in Batches

This will create a rule input of type Number (Integer) called batchNumber

7. Select the Expression rule and Integration to edit them.


Create Record Data Source

Success! The following objects are ready to be modified.

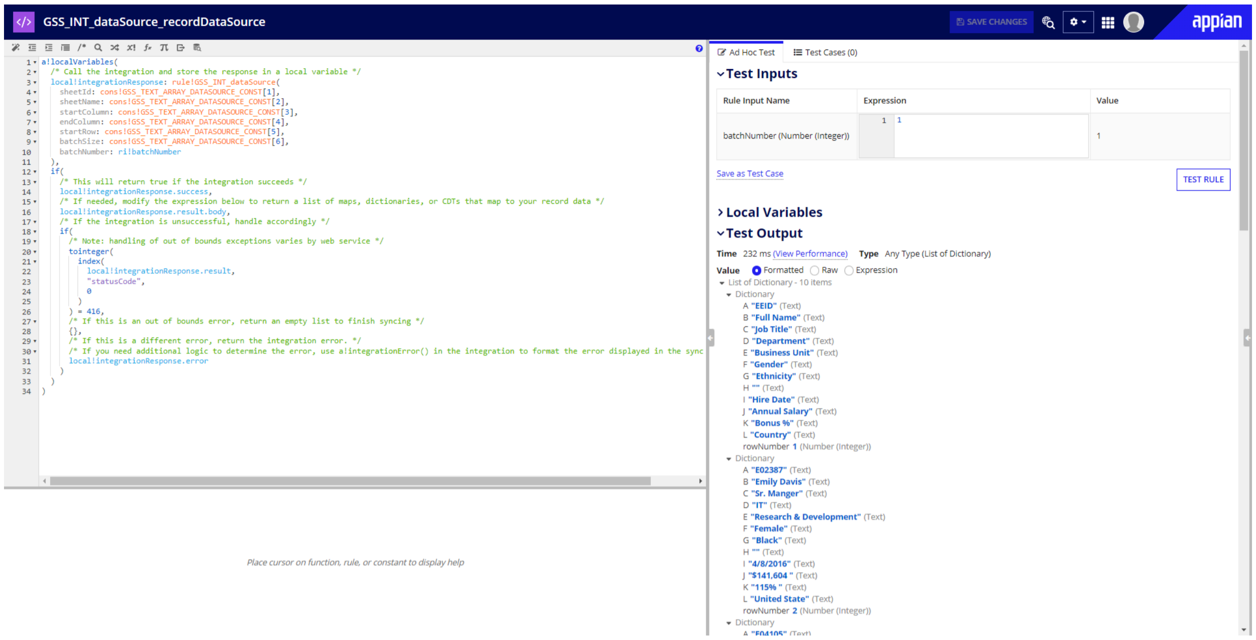
GSS_INT_dsInt_recordDataSource

Modify this expression rule to transform the integration response to return a list of maps, dictionaries, or CDTs.

GSS_INT_dsInt

Modify this integration to query the external system and return a list of record data.

8. Edit the Rule so that it looks like the code below.



GSS_INT_dataSource_recordDataSource

```
a!localVariables(
  /* Call the integration and store the response in a local variable */
  local!integrationResponse: rule!GSS_INT_dataSource(
    sheetId: cons!GSS_TEXT_ARRAY_DATASOURCE_CONST[1],
    sheetName: cons!GSS_TEXT_ARRAY_DATASOURCE_CONST[2],
    startColumn: cons!GSS_TEXT_ARRAY_DATASOURCE_CONST[3],
    endColumn: cons!GSS_TEXT_ARRAY_DATASOURCE_CONST[4],
    startRow: cons!GSS_TEXT_ARRAY_DATASOURCE_CONST[5],
    batchSize: cons!GSS_TEXT_ARRAY_DATASOURCE_CONST[6],
    batchNumber: ri!batchNumber
  ),
  if(
    /* This will return true if the integration succeeds */
    local!integrationResponse.success,
    /* If needed, modify the expression below to return a list of maps, dictionaries, or CDTs that map to your record data */
    local!integrationResponse.result.body,
    /* If the integration is unsuccessful, handle accordingly */
    if(
      /* Note: handling of out of bounds exceptions varies by web service */
      tointeger(
        index(
          local!integrationResponse.result,
          "statusCode",
          0
        )
      ) = 416,
      /* If this is an out of bounds error, return an empty list to finish syncing */
      {},
      /* If this is a different error, return the integration error. */
      /* If you need additional logic to determine the error, use a!integrationError() in the integration to format the error displayed in the sync
      local!integrationResponse.error
    )
  )
)
```

Place cursor on function, rule, or constant to display help

Ad Hoc Test | Test Cases (0)

Test Inputs

Rule Input Name	Expression	Value
batchNumber (Number (Integer))	1	1

Save as Test Case

Local Variables

Test Output

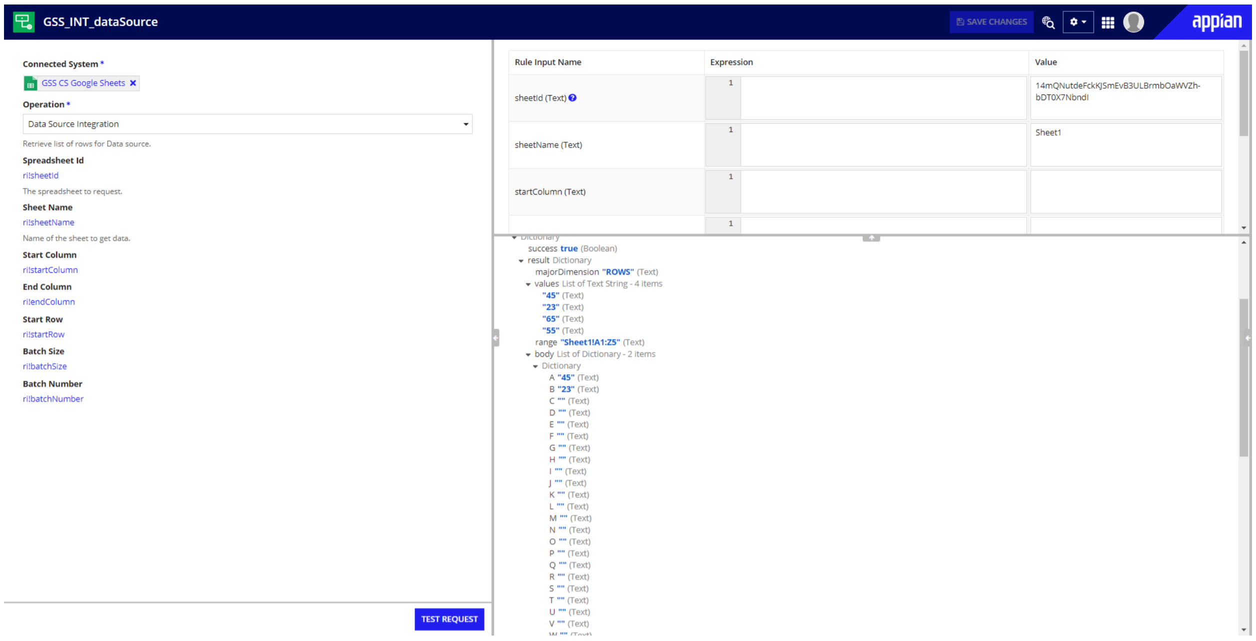
Time 232 ms (View Performance) Type Any Type (List of Dictionary)

Value [✓ Formatted] Raw Expression

```
List of Dictionary - 10 items
  Dictionary
    A "EEID" (Text)
    B "Full Name" (Text)
    C "Job Title" (Text)
    D "Department" (Text)
    E "Business Unit" (Text)
    F "Gender" (Text)
    G "Ethnicity" (Text)
    H "" (Text)
    I "Hire Date" (Text)
    J "Annual Salary" (Text)
    K "Bonus %" (Text)
    L "Country" (Text)
    rowNumber 1 (Number (Integer))
  Dictionary
    A "E02387" (Text)
    B "Emily Davis" (Text)
    C "Sr. Manger" (Text)
    D "IT" (Text)
    E "Research & Development" (Text)
    F "Female" (Text)
    G "Black" (Text)
    H "" (Text)
    I "4/8/2016" (Text)
    J "$141,604 " (Text)
    K "115% " (Text)
    L "United State" (Text)
    rowNumber 2 (Number (Integer))
  Dictionary
```

Important Note: Datasource constant is used in the sample application which contains sheet id, sheet name, start column, end column, start row and batch size. Parameters can also be manually configured according to requirements.

9. Check the Integration and whether it returns rows in batches.



GSS_INT_dataSource

Connected System * GSS CS Google Sheets

Operation * Data Source Integration

Retrieve list of rows for Data source.

Spreadsheet Id ri!sheetId

The spreadsheet to request.

Sheet Name ri!sheetName

Name of the sheet to get data.

Start Column ri!startColumn

End Column ri!endColumn

Start Row ri!startRow

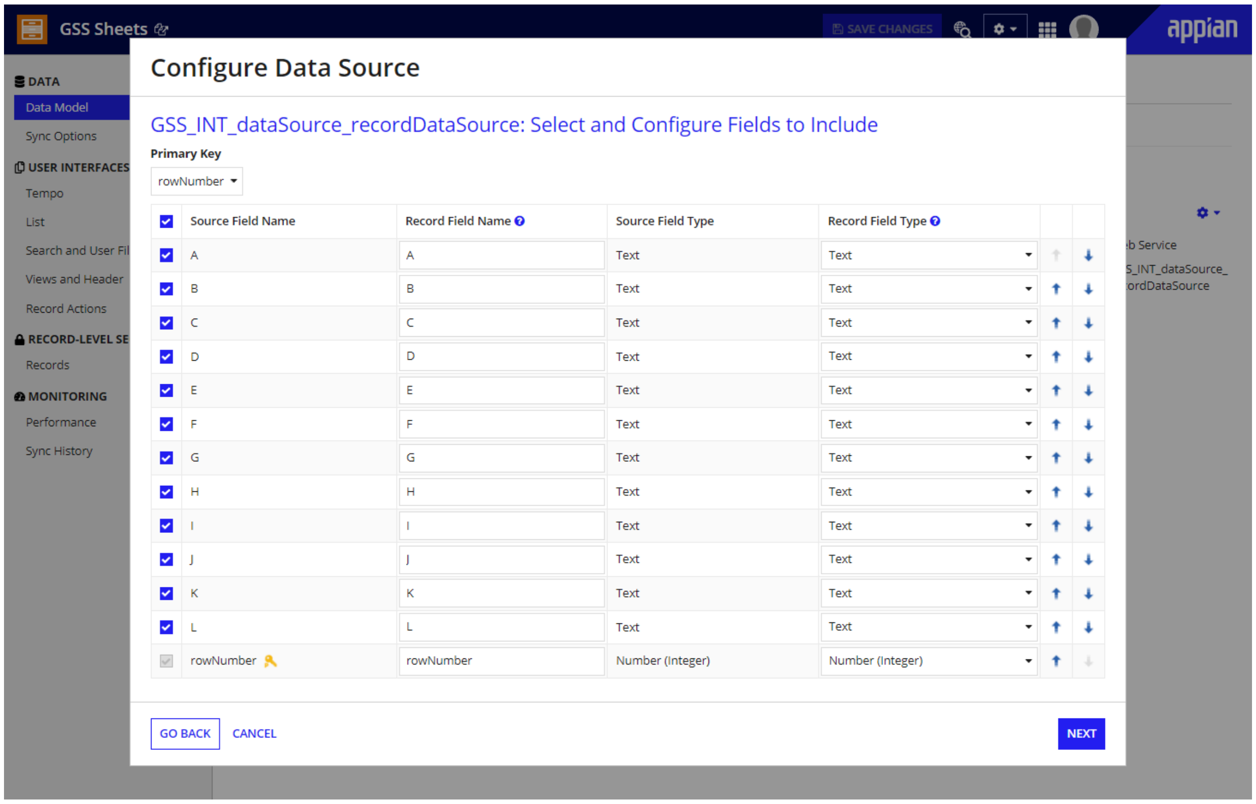
Batch Size ri!batchSize

Batch Number ri!batchNumber

Rule Input Name	Expression	Value
sheetId (Text)	1	14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI
sheetName (Text)	1	Sheet1
startColumn (Text)	1	
	1	

```
Dictionary
  success true (Boolean)
  result Dictionary
    majorDimension "ROWS" (Text)
    values List of Text String - 4 items
      "45" (Text)
      "23" (Text)
      "65" (Text)
      "55" (Text)
    range "Sheet1!A1:Z5" (Text)
    body List of Dictionary - 2 items
      Dictionary
        A "45" (Text)
        B "23" (Text)
        C "" (Text)
        D "" (Text)
        E "" (Text)
        F "" (Text)
        G "" (Text)
        H "" (Text)
        I "" (Text)
        J "" (Text)
        K "" (Text)
        L "" (Text)
        M "" (Text)
        N "" (Text)
        O "" (Text)
        P "" (Text)
        Q "" (Text)
        R "" (Text)
        S "" (Text)
        T "" (Text)
        U "" (Text)
        V "" (Text)
```

10. Once editing is done, select **OK** to preview data. Select **rowNumber** as **Primary Key** and Click **Finish.**



Configure Data Source

GSS_INT_dataSource_recordDataSource: Select and Configure Fields to Include

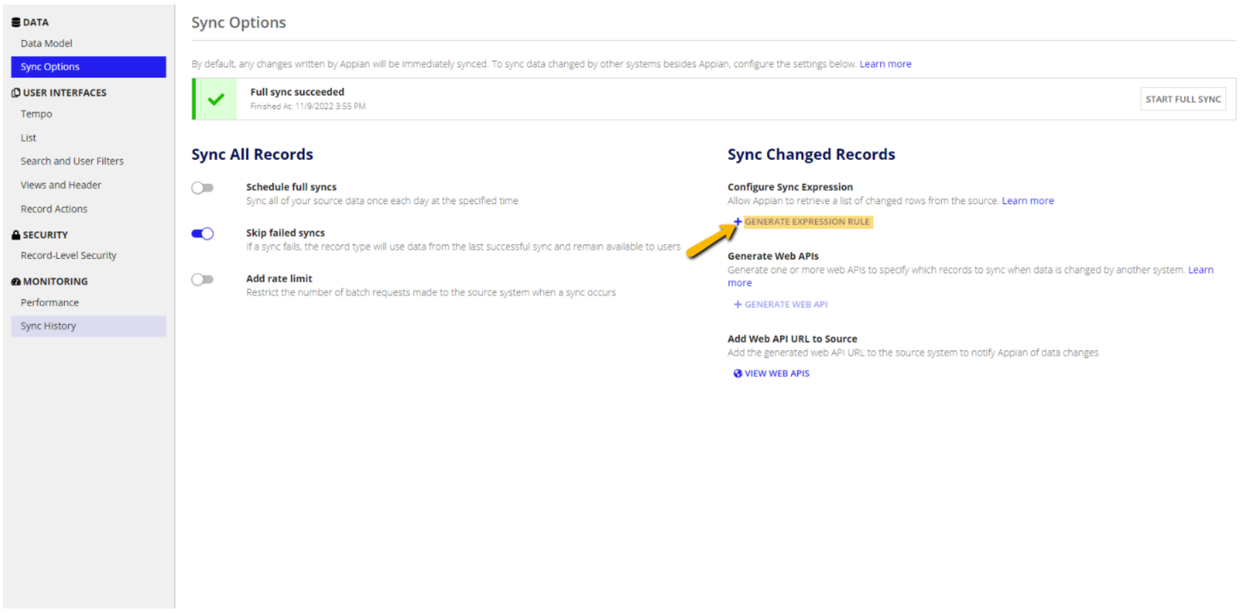
Primary Key

rowNumber

✓	Source Field Name	Record Field Name	Source Field Type	Record Field Type
✓	A	A	Text	Text
✓	B	B	Text	Text
✓	C	C	Text	Text
✓	D	D	Text	Text
✓	E	E	Text	Text
✓	F	F	Text	Text
✓	G	G	Text	Text
✓	H	H	Text	Text
✓	I	I	Text	Text
✓	J	J	Text	Text
✓	K	K	Text	Text
✓	L	L	Text	Text
✓	rowNumber	rowNumber	Number (Integer)	Number (Integer)

Sync Configuration

1. Once done, navigate to the **Sync Options Tab** and select the **Generate Expression rule** under **Configure Sync Expression.**



Sync Options

By default, any changes written by Appian will be immediately synced. To sync data changed by other systems besides Appian, configure the settings below. Learn more

Full sync succeeded
Finished At: 11/9/2022 3:55 PM

Sync All Records

Schedule full syncs
Sync all of your source data once each day at the specified time

Skip failed syncs
If a sync fails, the record type will use data from the last successful sync and remain available to users

Add rate limit
Restrict the number of batch requests made to the source system when a sync occurs

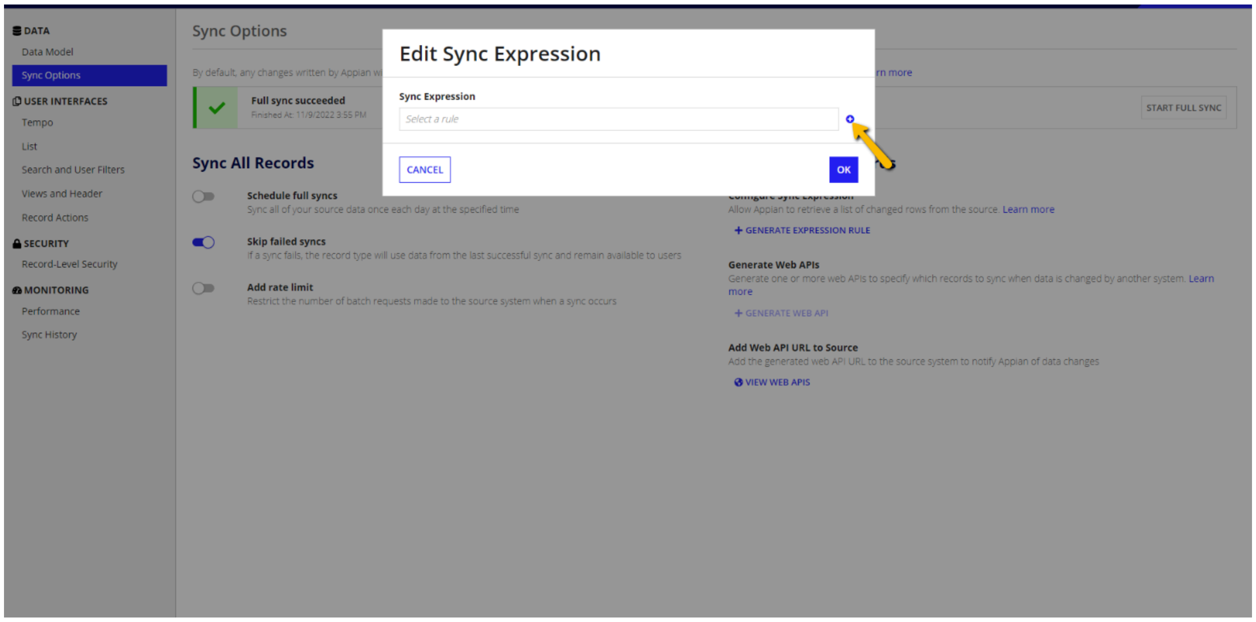
Sync Changed Records

Configure Sync Expression
Allow Appian to retrieve a list of changed rows from the source. Learn more

Generate Web APIs
Generate one or more web APIs to specify which records to sync when data is changed by another system. Learn more

Add Web API URL to Source
Add the generated web API URL to the source system to notify Appian of data changes

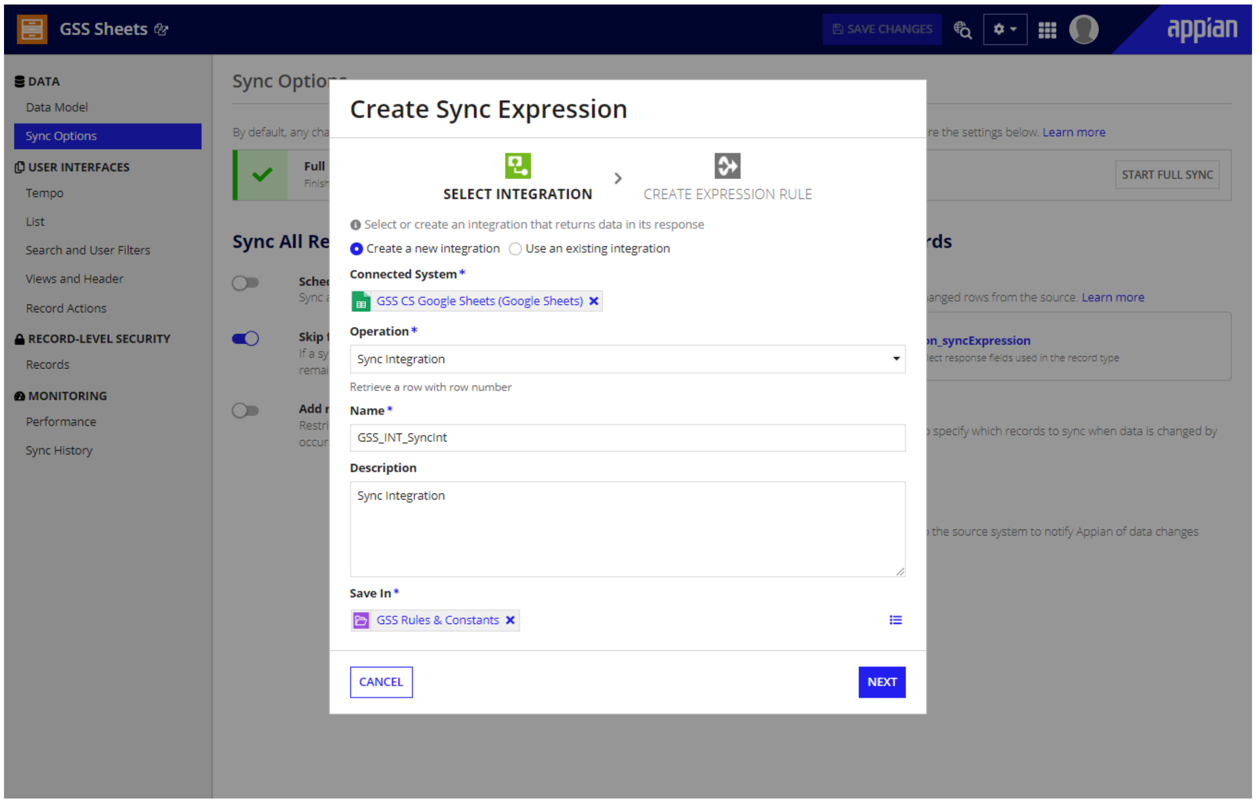
2. Select the **+** icon to create a new sync expression.



Edit Sync Expression

Sync Expression
Select a rule

3. You can either use existing integration or create new Integration. Select **Next.**


Create Sync Expression

SELECT INTEGRATION › CREATE EXPRESSION RULE

Select or create an integration that returns data in its response

[✓ Create a new integration] Use an existing integration

Connected System *
GSS CS Google Sheets (Google Sheets)

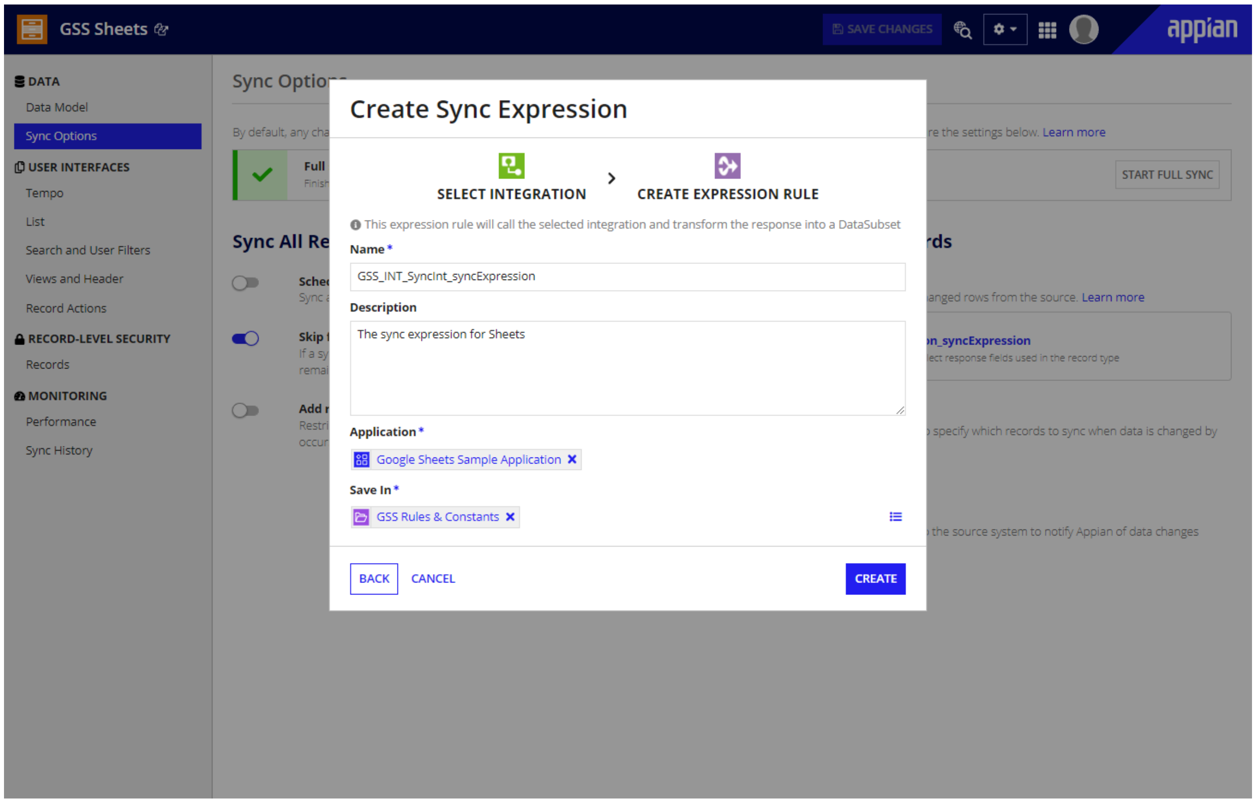
Operation *
Sync Integration
Retrieve a row with row number

Name *
GSS_INT_SyncInt

Description
Sync Integration

Save In *
GSS Rules & Constants

4. In the next page, the expression rule details will be automatically populated. Select **Create.**


GSS Sheets

Create Sync Expression

SELECT INTEGRATION > CREATE EXPRESSION RULE

This expression rule will call the selected integration and transform the response into a DataSubset

Name *
GSS_INT_SyncInt_syncExpression

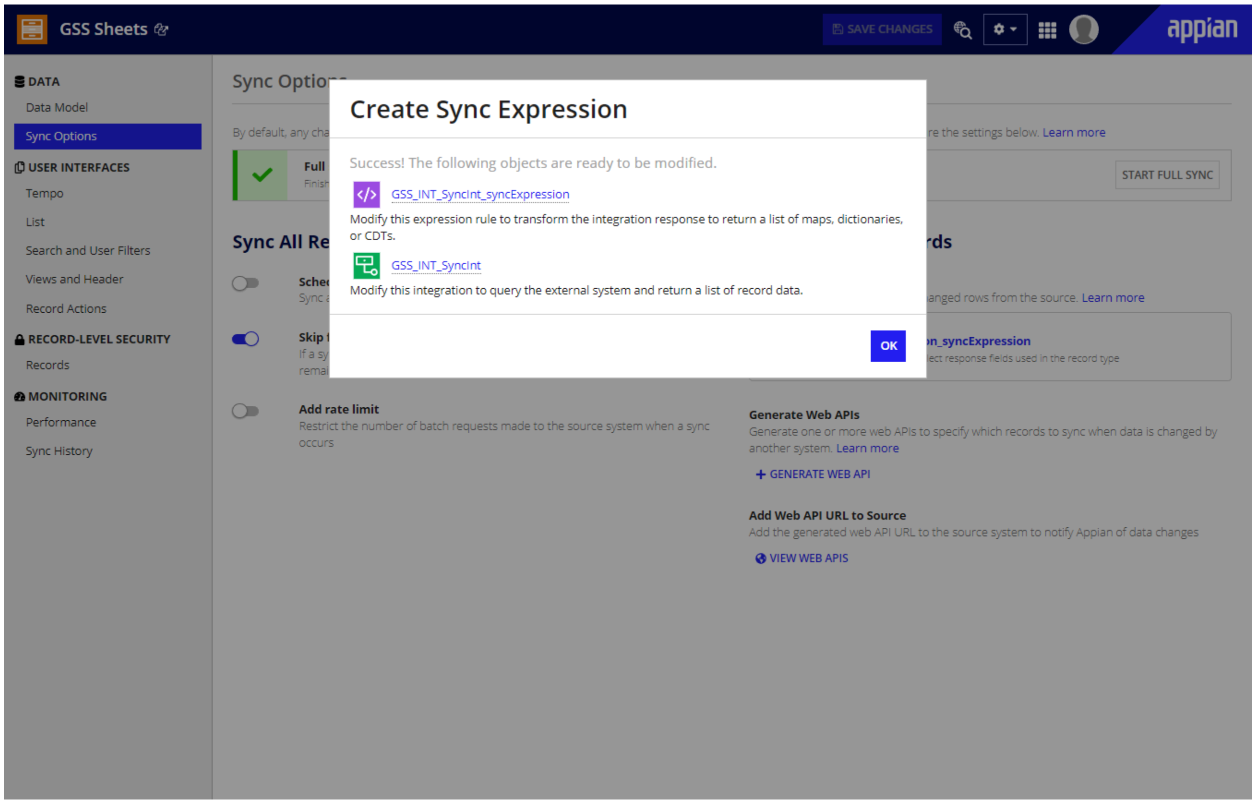
Description
The sync expression for Sheets

Application *
Google Sheets Sample Application

Save In *
GSS Rules & Constants

BACK CANCEL CREATE

5. Select Integration name and rule name to edit them.


GSS Sheets

Create Sync Expression

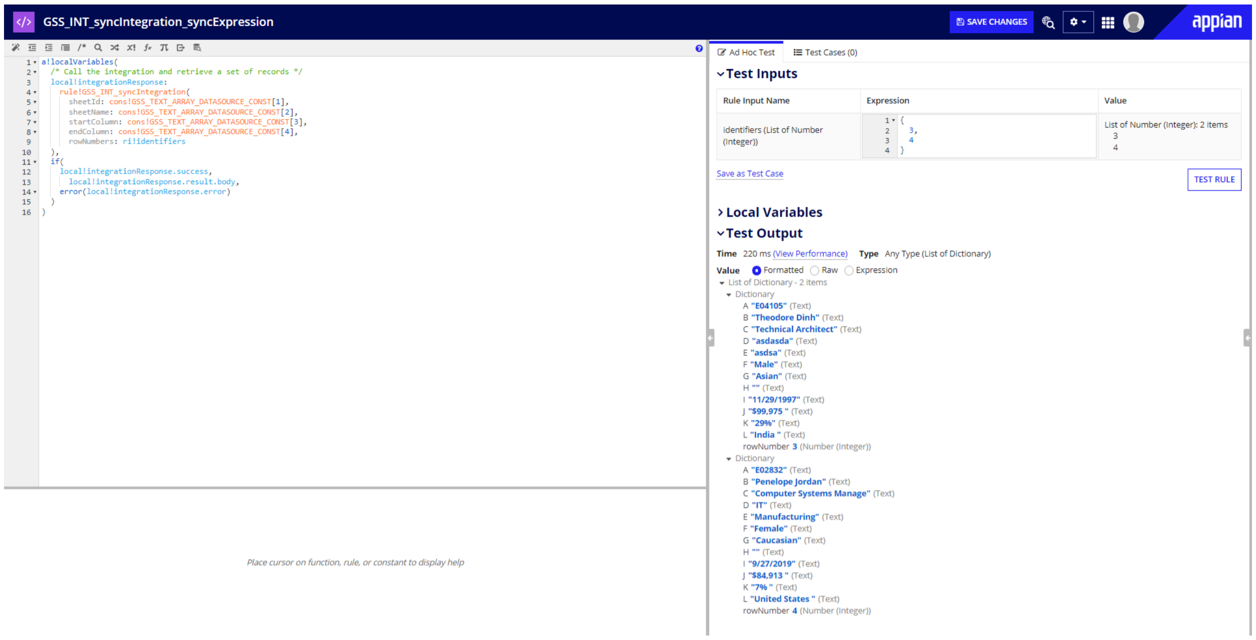
Success! The following objects are ready to be modified.

GSS_INT_SyncInt_syncExpression
Modify this expression rule to transform the integration response to return a list of maps, dictionaries, or CDTs.

GSS_INT_SyncInt
Modify this integration to query the external system and return a list of record data.

OK

6. Edit the Expression Rule and Integrations. Once done, click **OK.**


GSS_INT_syncIntegration_syncExpression

```
a!localVariables(
  /* Call the integration and retrieve a set of records */
  local!integrationResponse:
    rule!GSS_INT_syncIntegration(
      sheetId: cons!GSS_TEXT_ARRAY_DATASOURCE_CONST[1],
      sheetName: cons!GSS_TEXT_ARRAY_DATASOURCE_CONST[2],
      startColumn: cons!GSS_TEXT_ARRAY_DATASOURCE_CONST[3],
      endColumn: cons!GSS_TEXT_ARRAY_DATASOURCE_CONST[4],
      rowNumbers: ri!identifiers
  ),
  if(
    local!integrationResponse.success,
      local!integrationResponse.result.body,
    error(local!integrationResponse.error)
  )
)
```

Place cursor on function, rule, or constant to display help

Ad Hoc Test | Test Cases (0)

Test Inputs

Rule Input Name	Expression	Value
identifiers (List of Number (Integer))	{ 3, 4 }	List of Number (Integer): 2 items 3 4

Save as Test Case

TEST RULE

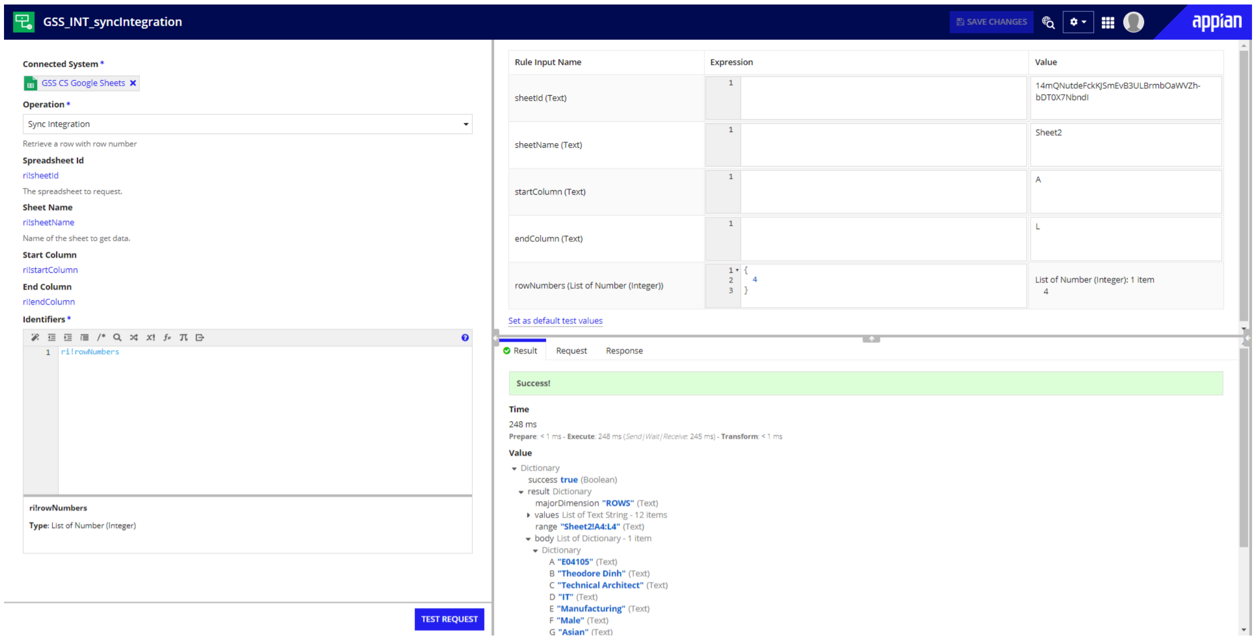
Local Variables

Test Output

Time 220 ms (View Performance) **Type** Any Type (List of Dictionary)

Value [✓ Formatted] Raw Expression

```
List of Dictionary - 2 items
  Dictionary
    A "E04105" (Text)
    B "Theodore Dinh" (Text)
    C "Technical Architect" (Text)
    D "asdasda" (Text)
    E "asdsa" (Text)
    F "Male" (Text)
    G "Asian" (Text)
    H "" (Text)
    I "11/29/1997" (Text)
    J "$99,975 " (Text)
    K "29%" (Text)
    L "India " (Text)
    rowNumber 3 (Number (Integer))
  Dictionary
    A "E02832" (Text)
    B "Penelope Jordan" (Text)
    C "Computer Systems Manage" (Text)
    D "IT" (Text)
    E "Manufacturing" (Text)
    F "Female" (Text)
    G "Caucasian" (Text)
    H "" (Text)
    I "9/27/2019" (Text)
    J "$84,913 " (Text)
    K "7% " (Text)
    L "United States " (Text)
    rowNumber 4 (Number (Integer))
```


GSS_INT_syncIntegration

Connected System *
GSS CS Google Sheets

Operation *
Sync Integration

Retrieve a row with row number

Spreadsheet Id
ri!sheetId
The spreadsheet to request.

Sheet Name
ri!sheetName
Name of the sheet to get data.

Start Column
ri!startColumn

End Column
ri!endColumn

Identifiers *

```
ri!rowNumbers
```

ri!rowNumbers
Type: List of Number (Integer)

TEST REQUEST

Rule Input Name	Expression	Value
sheetId (Text)		14mQNutdeFckKJSmEvB3ULBrmbOaWVZh-bDT0X7NbndI
sheetName (Text)		Sheet2
startColumn (Text)		A
endColumn (Text)		L
rowNumbers (List of Number (Integer))	{ 4 }	List of Number (Integer): 1 item 4

Set as default test values

Result | Request | Response

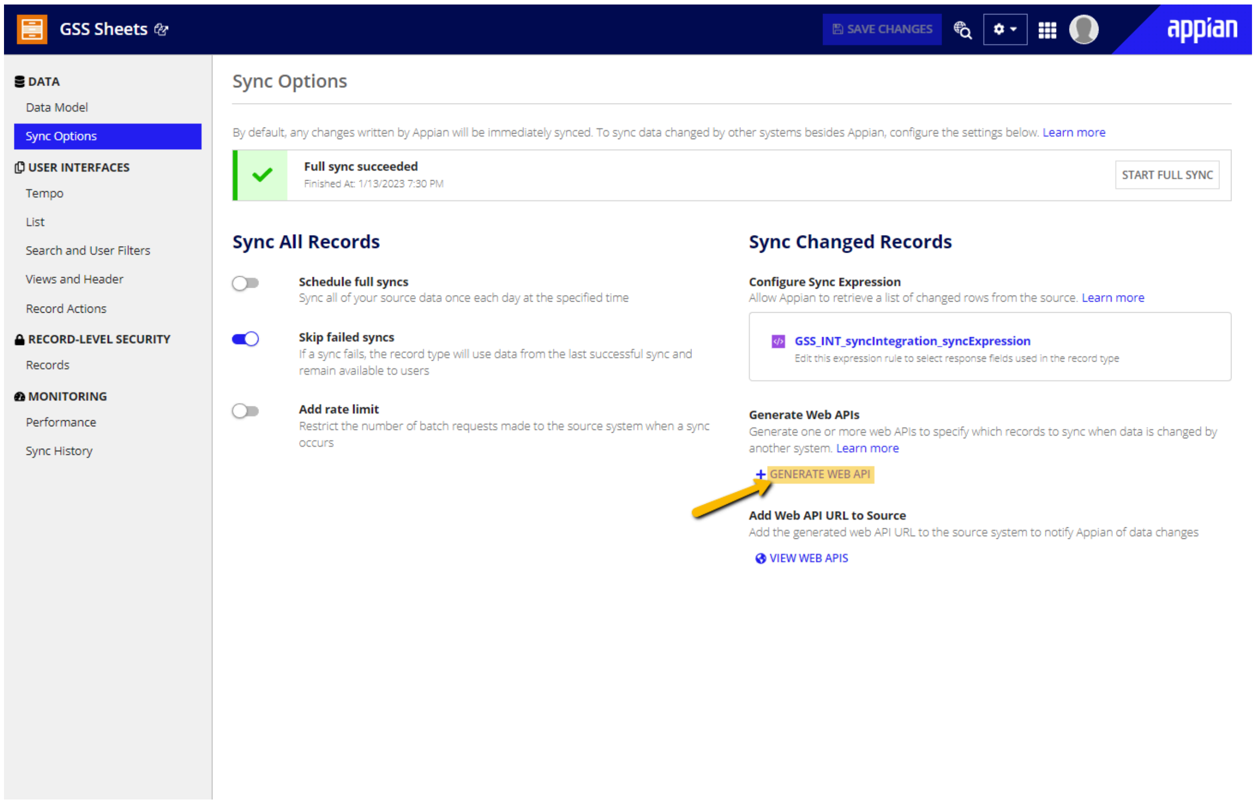
Success!

Time
248 ms
Prepare: < 1 ms - **Execute**: 248 ms (*Send / Wait / Receive*: 245 ms) - **Transform**: < 1 ms

Value

```
Dictionary
  success true (Boolean)
  result Dictionary
    majorDimension "ROWS" (Text)
    values List of Text String - 12 items
    range "Sheet2!A4:L4" (Text)
  body List of Dictionary - 1 item
    Dictionary
      A "E04105" (Text)
      B "Theodore Dinh" (Text)
      C "Technical Architect" (Text)
      D "IT" (Text)
      E "Manufacturing" (Text)
      F "Male" (Text)
      G "Asian" (Text)
```

7. Once Sync Expression is configured, select the **Generate Web API** button under **Generate Web APIs**



GSS Sheets

SAVE CHANGES

DATA
- Data Model
- Sync Options

USER INTERFACES
- Tempo
- List
- Search and User Filters
- Views and Header
- Record Actions

RECORD-LEVEL SECURITY
- Records

MONITORING
- Performance
- Sync History

Sync Options

By default, any changes written by Appian will be immediately synced. To sync data changed by other systems besides Appian, configure the settings below. Learn more

Full sync succeeded
Finished At: 1/13/2023 7:30 PM

START FULL SYNC

Sync All Records

Schedule full syncs
Sync all of your source data once each day at the specified time

Skip failed syncs
If a sync fails, the record type will use data from the last successful sync and remain available to users

Add rate limit
Restrict the number of batch requests made to the source system when a sync occurs

Sync Changed Records

Configure Sync Expression
Allow Appian to retrieve a list of changed rows from the source. Learn more

GSS_INT_syncIntegration_syncExpression
Edit this expression rule to select response fields used in the record type

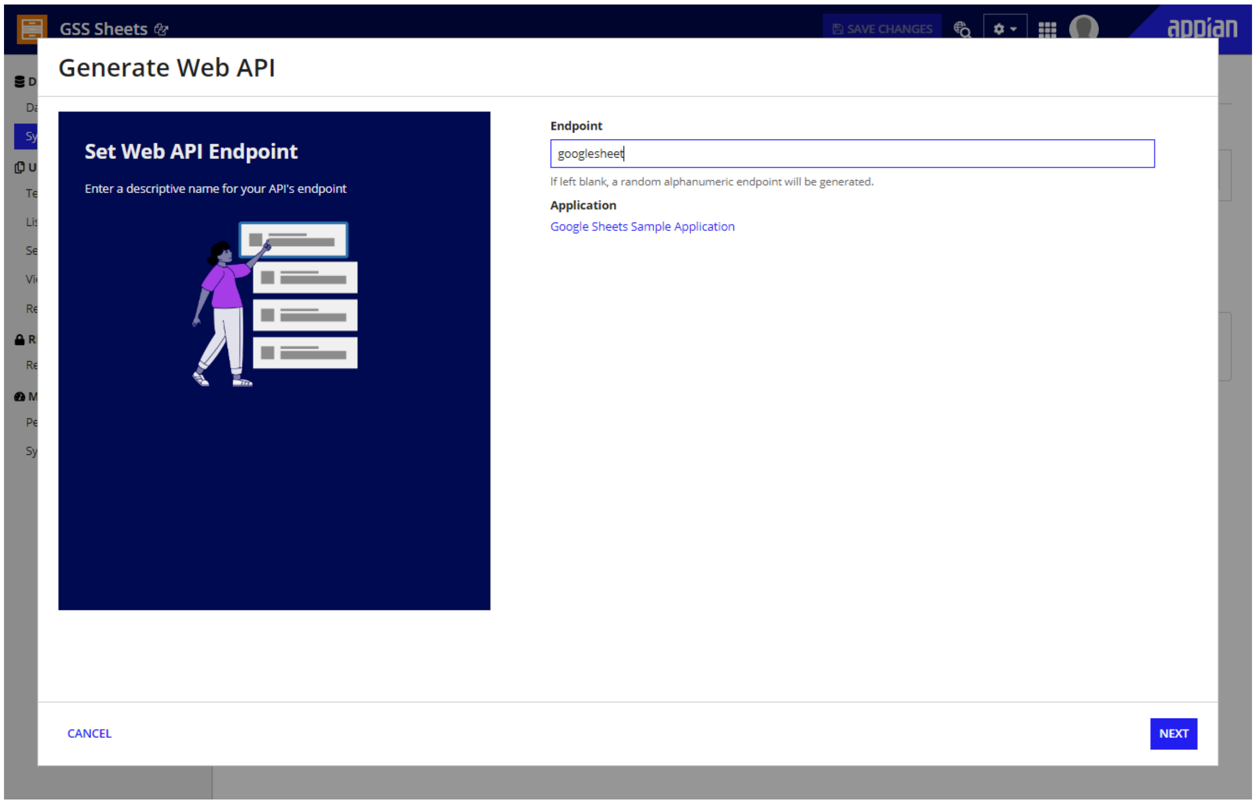
Generate Web APIs
Generate one or more web APIs to specify which records to sync when data is changed by another system. Learn more

+ GENERATE WEB API

Add Web API URL to Source
Add the generated web API URL to the source system to notify Appian of data changes

VIEW WEB APIS

8. Specify the endpoint in the next page and select **Next.** Select **Generate Web API** in the next page.



Generate Web API

Set Web API Endpoint

Enter a descriptive name for your API's endpoint

Endpoint
googlesheet

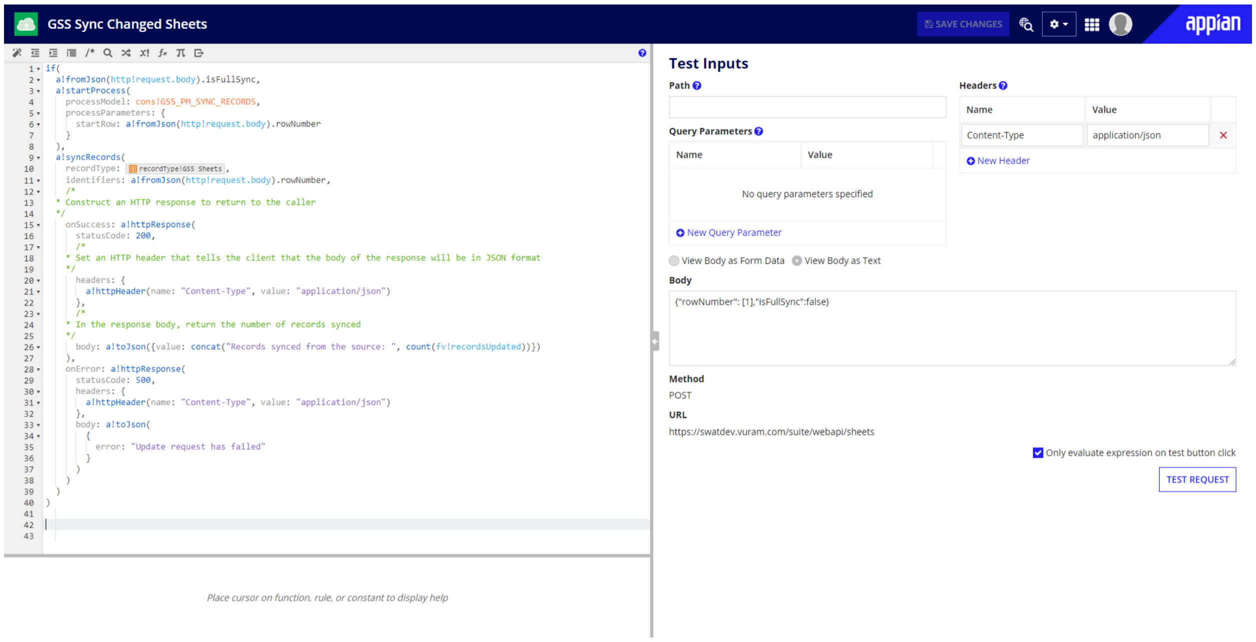
If left blank, a random alphanumeric endpoint will be generated.

Application
Google Sheets Sample Application

CANCEL

NEXT

9. Edit the Web service like the code below


GSS Sync Changed Sheets

```
if(
  a!fromJson(http!request.body).isFullSync,
  a!startProcess(
    processModel: cons!GSS_PM_SYNC_RECORDS,
    processParameters: {
      startRow: a!fromJson(http!request.body).rowNumber
    }
  ),
  a!syncRecords(
    recordType: recordType!GSS Sheets,
    identifiers: a!fromJson(http!request.body).rowNumber,
    /*
  * Construct an HTTP response to return to the caller
  */
    onSuccess: a!httpResponse(
      statusCode: 200,
      /*
    * Set an HTTP header that tells the client that the body of the response will be in JSON format
    */
      headers: {
        a!httpHeader(name: "Content-Type", value: "application/json")
      },
      /*
    * In the response body, return the number of records synced
    */
      body: a!toJson({value: concat("Records synced from the source: ", count(fv!recordsUpdated))})
    ),
    onError: a!httpResponse(
      statusCode: 500,
      headers: {
        a!httpHeader(name: "Content-Type", value: "application/json")
      },
      body: a!toJson(
        {
          error: "Update request has failed"
        }
      )
    )
  )
)
```

Place cursor on function, rule, or constant to display help

Test Inputs

Path

Query Parameters

Name	Value
No query parameters specified	

New Query Parameter

Headers

Name	Value
Content-Type	application/json

New Header

View Body as Form Data   View Body as Text

Body

```
{"rowNumber": [1],"isFullSync":false}
```

Method

POST

URL

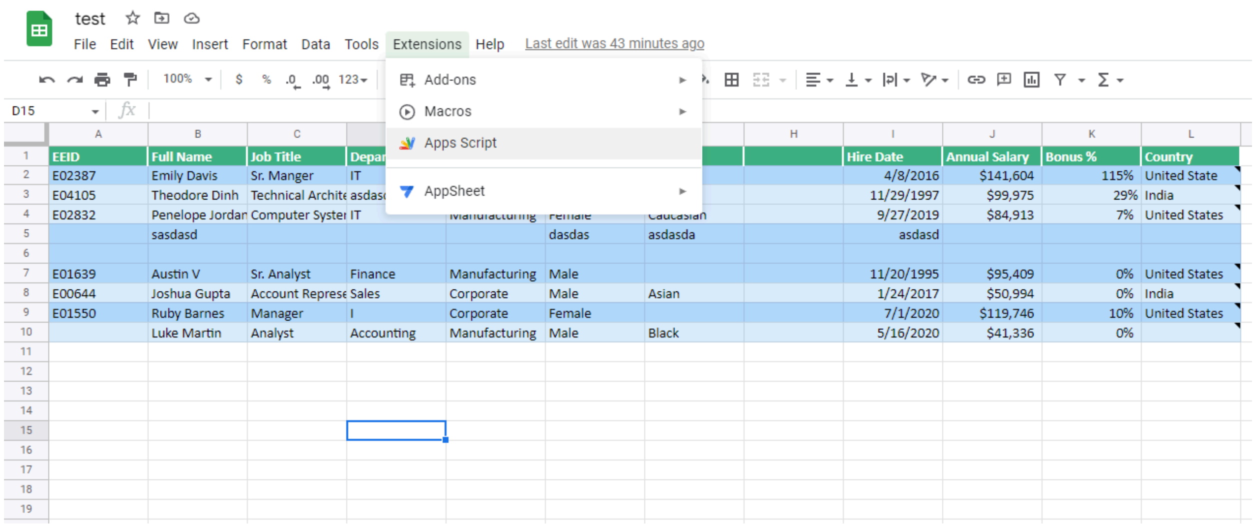
https://swatdev.vuram.com/suite/webapi/sheets

[✓] Only evaluate expression on test button click

TEST REQUEST

Webhook Configuration

1. Open the Google spreadsheet that is configured in the datasource and navigate to **Extensions → Apps Script.**


test
File Edit View Insert Format Data Tools Extensions Help
Last edit was 43 minutes ago
Add-ons
Macros
Apps Script
AppSheet
D15

	A	B	C	D	E	F	G	H	I	J	K	L
1	EEID	Full Name	Job Title	Depar					Hire Date	Annual Salary	Bonus %	Country
2	E02387	Emily Davis	Sr. Manger	IT					4/8/2016	$141,604	115%	United State
3	E04105	Theodore Dinh	Technical Archite	asdas					11/29/1997	$99,975	29%	India
4	E02832	Penelope Jordan	Computer Syster	IT	Manufacturing	Female	Caucasian		9/27/2019	$84,913	7%	United States
5		sasdasd				dasdas	asdasda		asdasd			
6												
7	E01639	Austin V	Sr. Analyst	Finance	Manufacturing	Male			11/20/1995	$95,409	0%	United States
8	E00644	Joshua Gupta	Account Represe	Sales	Corporate	Male	Asian		1/24/2017	$50,994	0%	India
9	E01550	Ruby Barnes	Manager	I	Corporate	Female			7/1/2020	$119,746	10%	United States
10		Luke Martin	Analyst	Accounting	Manufacturing	Male	Black		5/16/2020	$41,336	0%	

2. Paste the following code as a **code.gs** file

```
function webhookOnChange(e){
  var rowNumber = null;
  const ss=SpreadsheetApp.getActive();
```

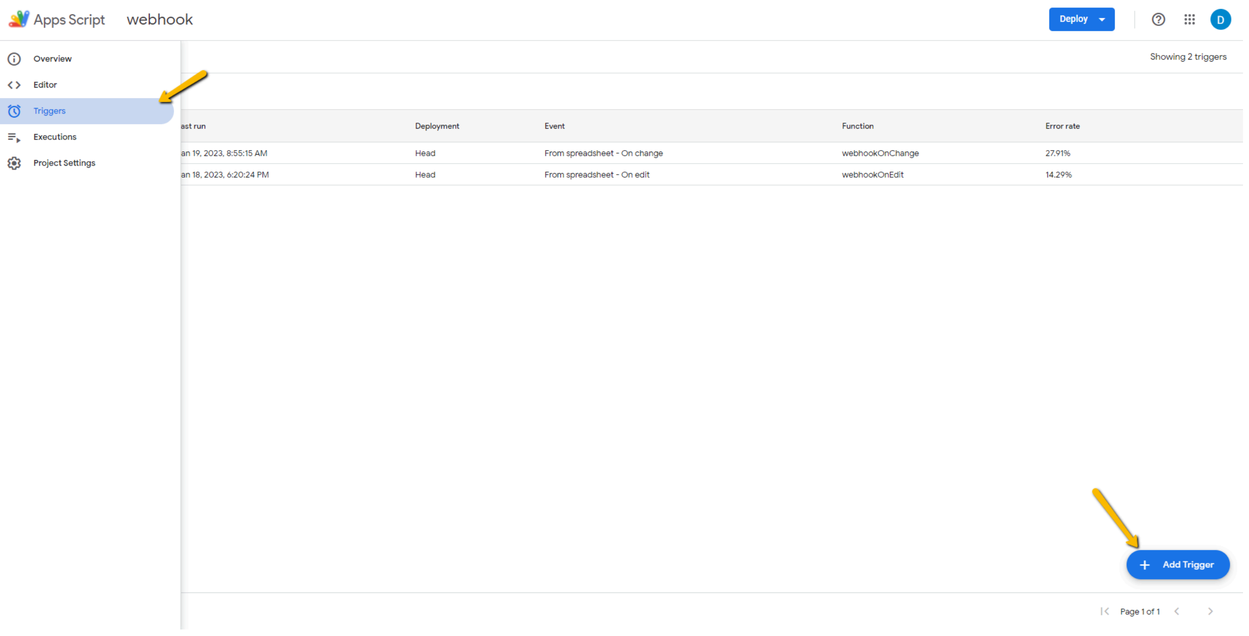
```
    const sh=ss.getActiveSheet();
    const rg=sh.getActiveRange();
  if(e.changeType == "INSERT_ROW" || e.changeType == "INSERT_COLUMN" || e.changeType ==
"REMOVE_ROW" || e.changeType == "REMOVE_COLUMN"){
    rowNumber = rg.getRow();
    if(e.changeType == "INSERT_COLUMN" || e.changeType == "REMOVE_COLUMN"){
      rowNumber = 1;
    }
    isFullSync = true;
    var headers = {
    "Appian-API-Key" : "<Your serviceAccount API key>"
  };
  var payload = {
    "rowNumber": rowNumber,
    "isFullSync": true
  };
 var request = {
 'method' : 'post',
 'payload' : JSON.stringify(payload),
 'headers' : headers
 };
 UrlFetchApp.fetch('<Your Sync Web API endpoint>', request);
  }
 return;
}

function webhookOnEdit(e){

  const range = e.range;
  var headers = {
    "Appian-API-Key" : "<Your serviceAccount API key>"
  };
  var payload = {
    "rowNumber": range.getRow(),
    "isFullSync": false
  };
 var request = {
 'method' : 'post',
 'payload' : JSON.stringify(payload),
 'headers' : headers
 };
 console.log("request body ",request);
 UrlFetchApp.fetch('<Your Sync Web API endpoint>', request);
 return;
}
```

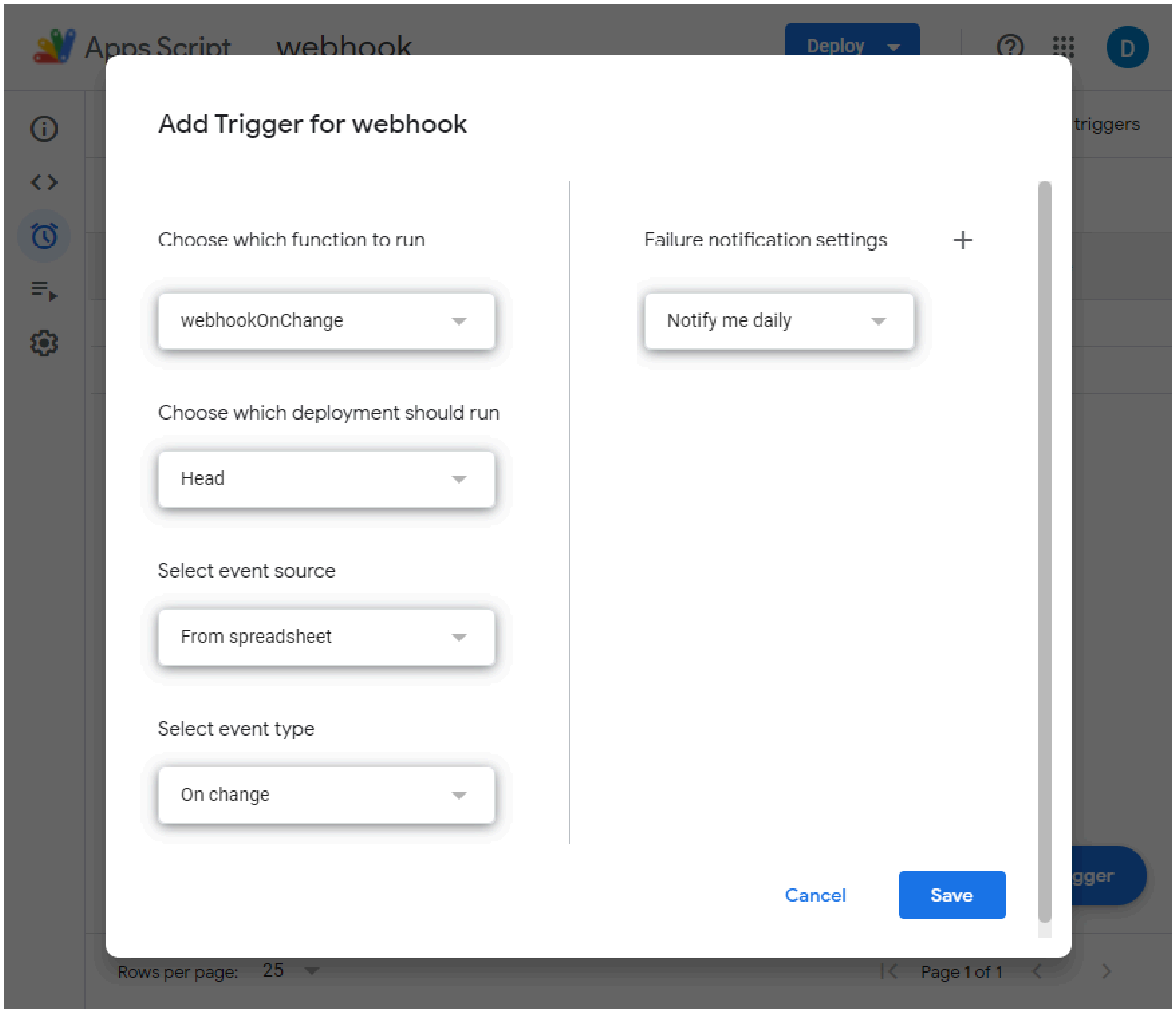
Important Note:
Add your Appian WebApi URL in the marked fields.
Add your Appian Service account API key in the marked fields.

3. Once done, select **Triggers** on left pane and select **Add Trigger** button on bottom right



ast run	Deployment	Event	Function	Error rate
an 19, 2023, 8:55:15 AM	Head	From spreadsheet - On change	webhookOnChange	27.91%
an 18, 2023, 6:20:24 PM	Head	From spreadsheet - On edit	webhookOnEdit	14.29%

4. Add the first trigger and configure the on change trigger as configured below. Select **Save** to add the trigger.



Add Trigger for webhook

Choose which function to run: webhookOnChange

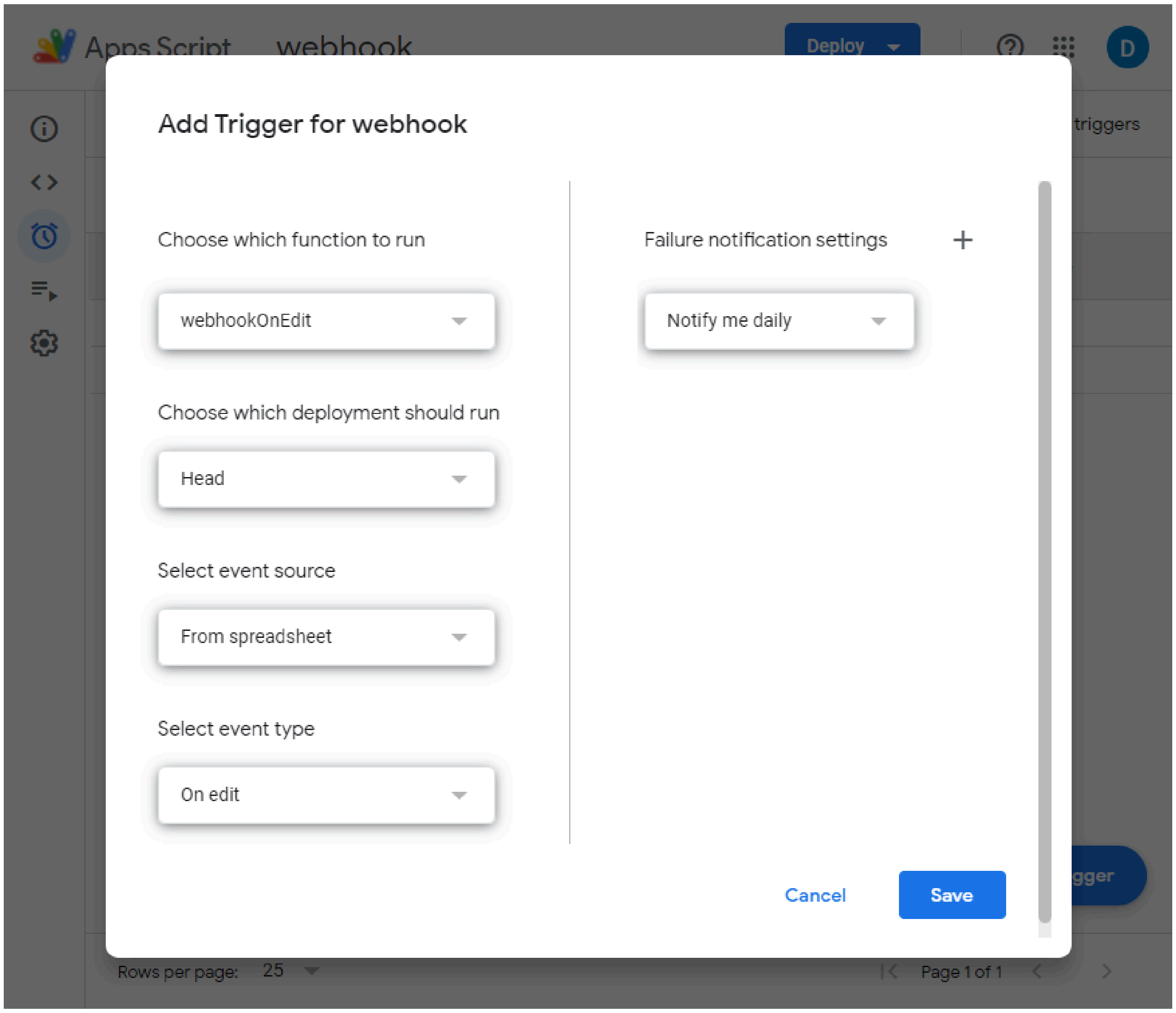
Choose which deployment should run: Head

Select event source: From spreadsheet

Select event type: On change

Failure notification settings: Notify me daily

5. Add another trigger for **On Edit** and configure as below. Select **Save** to add a trigger.



Add Trigger for webhook

Choose which function to run: webhookOnEdit

Choose which deployment should run: Head

Select event source: From spreadsheet

Select event type: On edit

Failure notification settings: Notify me daily

The webhook configuration is now complete and Appian will be notified whenever a cell is edited or a new row or column is added.

Additional Configuration Information

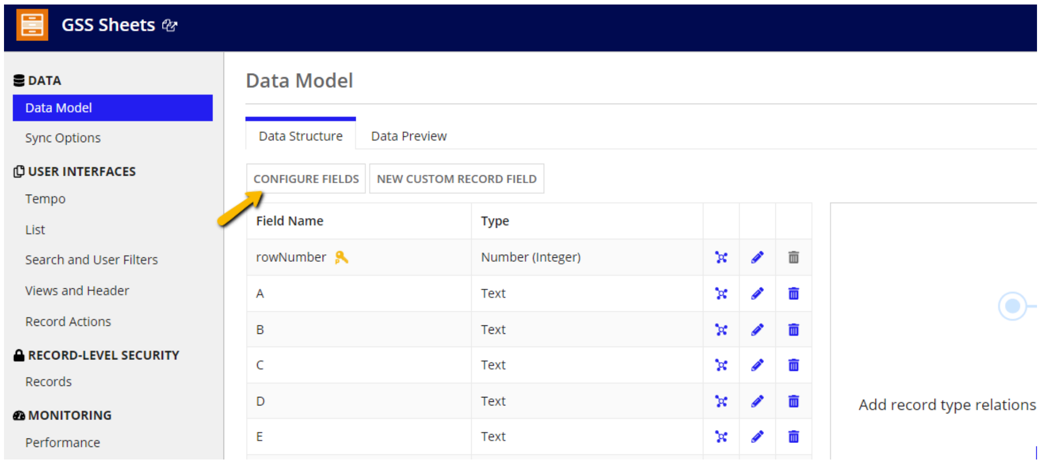
- The webhook shown here is configured such that it triggers a Data Sync Process Model when Insertion or Deletion happens in Rows or Columns. It may take some time to reflect on the record due to this.
- Editing multiple cells (either continuous or discontinuous) at the same time might not reflect in the record as expected.
- A new column added or inserted which exceeds the column range in record will not be reflected automatically in records.

- The User needs to:

a. Change start column and end column value on both data source Integration and sync Integration

b. Configure fields in record. This can be done by following the below steps:

- Select the **configure fields** button under **Data Model**



GSS Sheets

DATA
Data Model
Sync Options
USER INTERFACES
Tempo
List
Search and User Filters
Views and Header
Record Actions
RECORD-LEVEL SECURITY
Records
MONITORING
Performance

Data Model

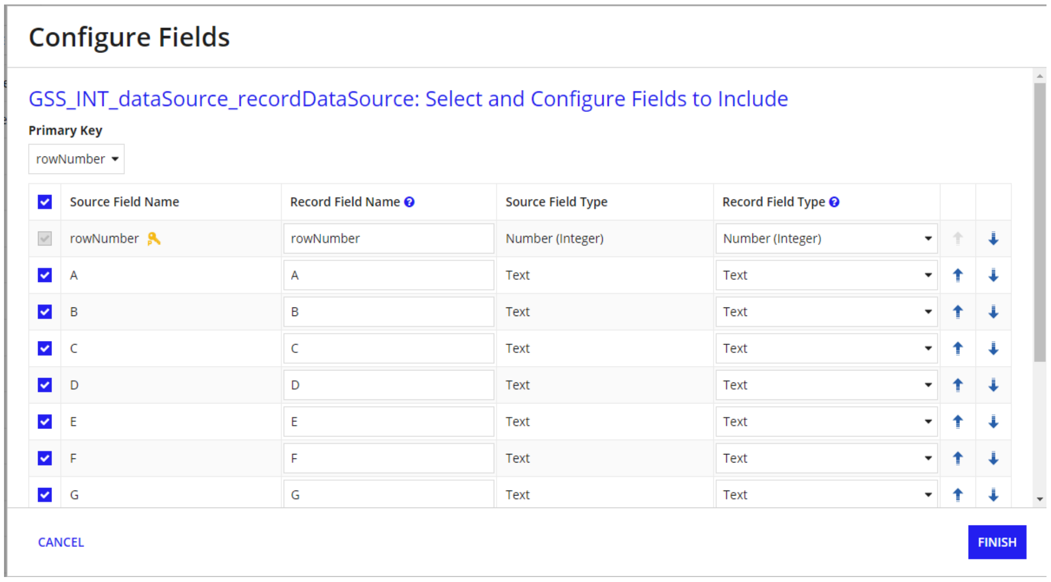
Data Structure | Data Preview

CONFIGURE FIELDS | NEW CUSTOM RECORD FIELD

Field Name	Type
rowNumber	Number (Integer)
A	Text
B	Text
C	Text
D	Text
E	Text

Add record type relations

- Select the updated rows and click **Finish.** Now, the updated columns will be reflected in the record view.



Configure Fields

GSS_INT_dataSource_recordDataSource: Select and Configure Fields to Include

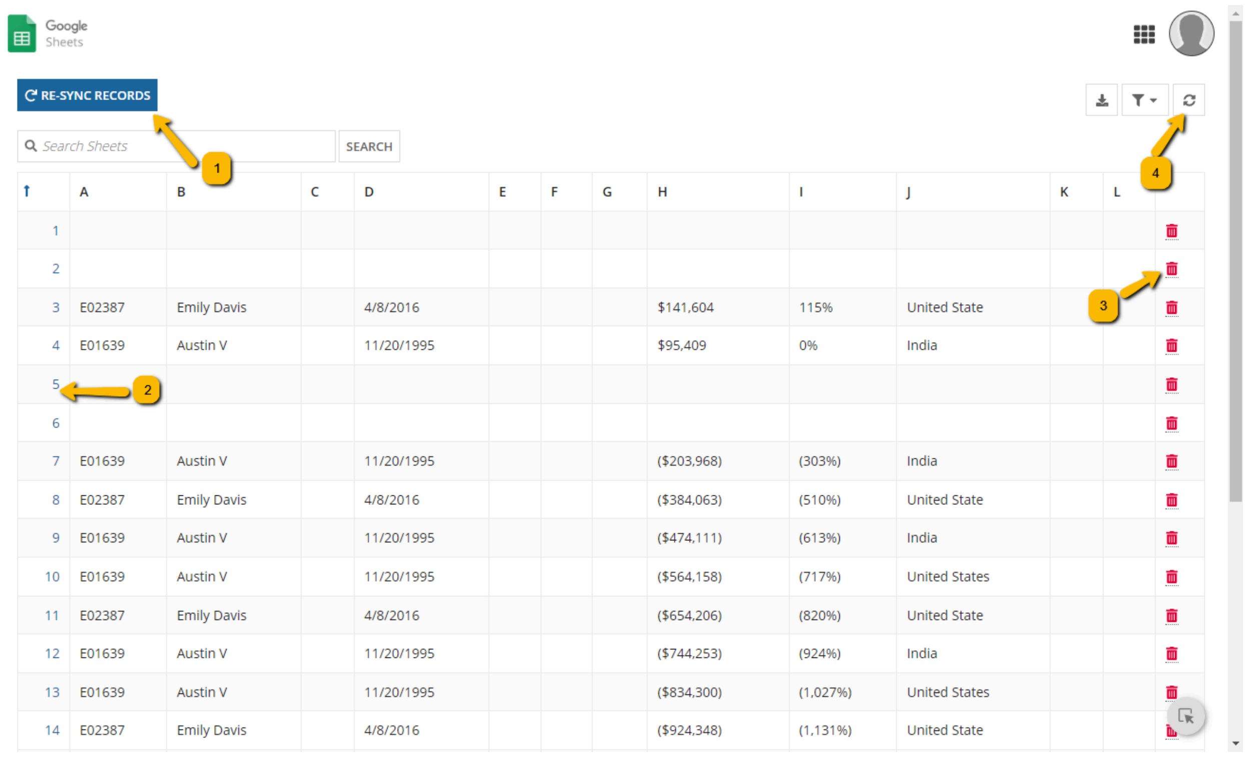
Primary Key
rowNumber

✓	Source Field Name	Record Field Name	Source Field Type	Record Field Type
✓	rowNumber	rowNumber	Number (Integer)	Number (Integer)
✓	A	A	Text	Text
✓	B	B	Text	Text
✓	C	C	Text	Text
✓	D	D	Text	Text
✓	E	E	Text	Text
✓	F	F	Text	Text
✓	G	G	Text	Text

CANCEL FINISH

Record Type Functionalities

This section outlines key functionalities of the record types.

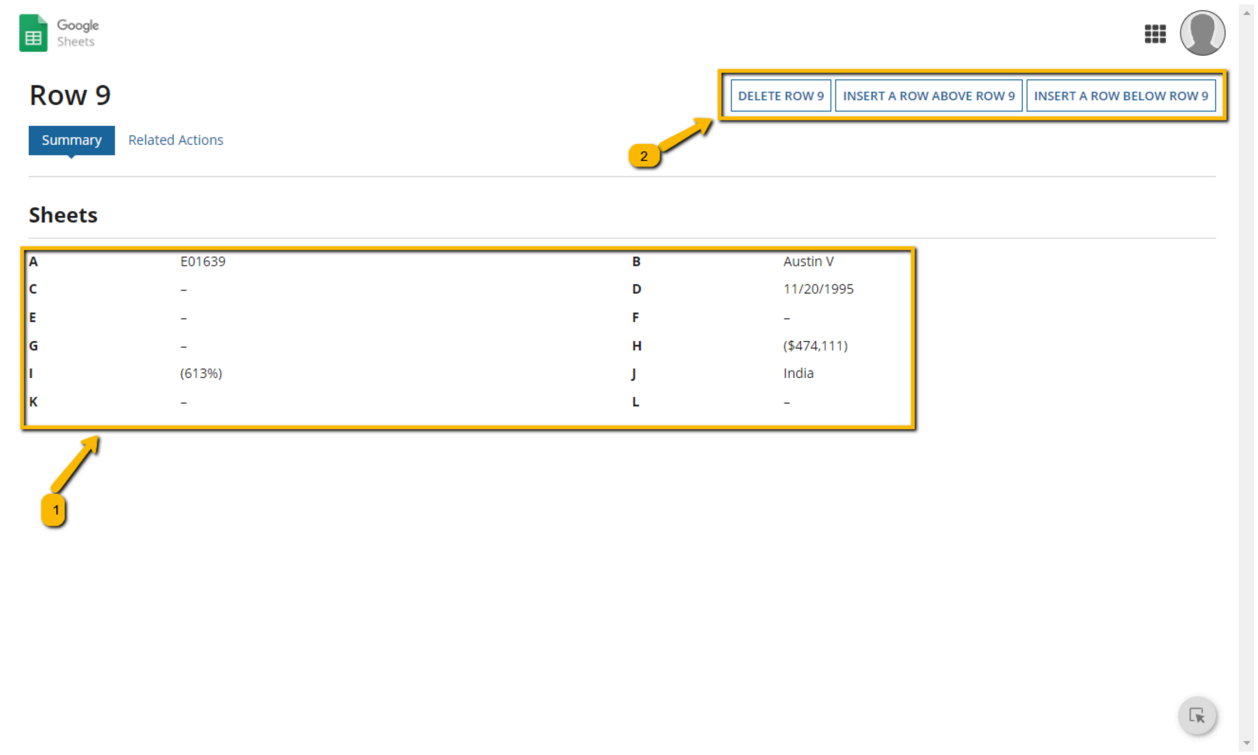


↑	A	B	C	D	E	F	G	H	I	J	K	L
1												
2												
3	E02387	Emily Davis		4/8/2016				$141,604	115%	United State		
4	E01639	Austin V		11/20/1995				$95,409	0%	India		
5												
6												
7	E01639	Austin V		11/20/1995				($203,968)	(303%)	India		
8	E02387	Emily Davis		4/8/2016				($384,063)	(510%)	United State		
9	E01639	Austin V		11/20/1995				($474,111)	(613%)	India		
10	E01639	Austin V		11/20/1995				($564,158)	(717%)	United States		
11	E02387	Emily Davis		4/8/2016				($654,206)	(820%)	United State		
12	E01639	Austin V		11/20/1995				($744,253)	(924%)	India		
13	E01639	Austin V		11/20/1995				($834,300)	(1,027%)	United States		
14	E02387	Emily Davis		4/8/2016				($924,348)	(1,131%)	United State		

1. Record Action which triggers the Process Model which to Syncs all the records from the sheet. It's functionality is similar to *Start Full Sync* Function
2. Record link which opens *Summary* page and *Related Actions* view
3. Delete Row link which will delete the current row in the Google sheet. The changes will be reflected in the record after a few seconds and it depends on the record refresh interval.
4. Refresh record button which will refresh the record and the changes will be reflected when the Data in the record is not the same as the Data in the current screen.
   Note: Refresh record will not trigger Full sync function.

Summary Page

This section outlines key parts of the summary page.

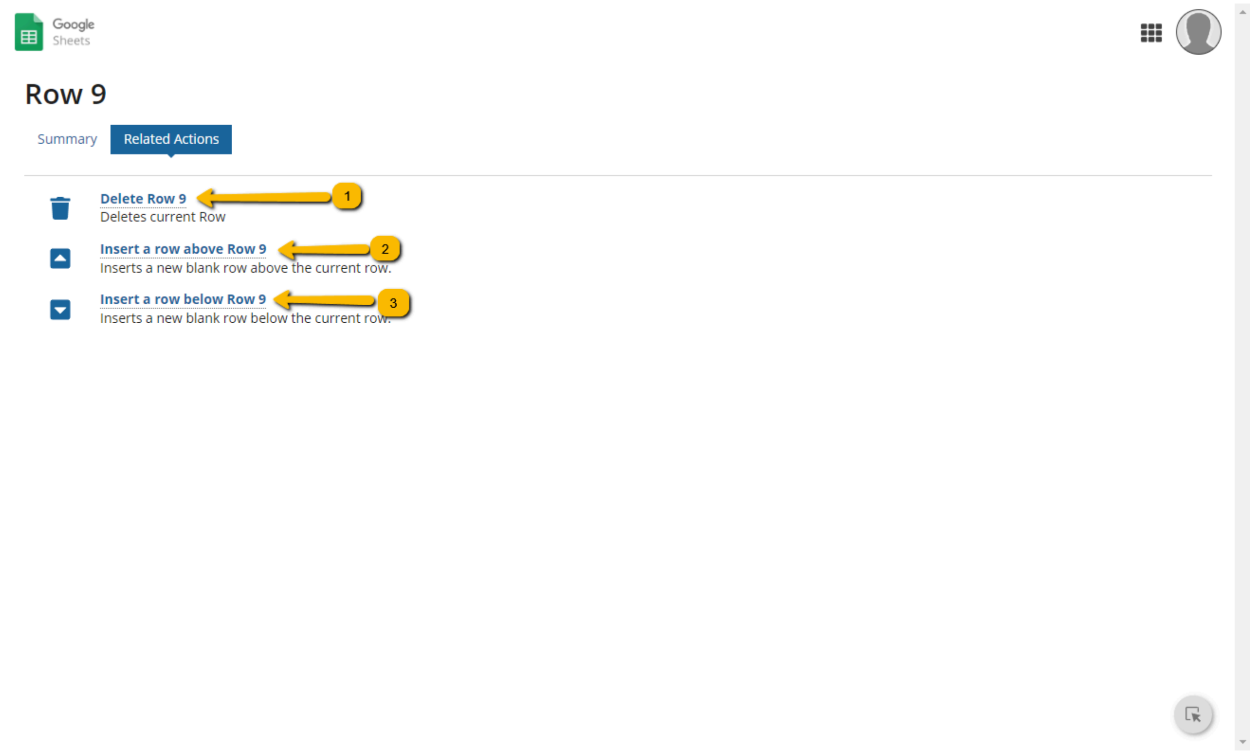


Sheets

A	E01639	B	Austin V
C	–	D	11/20/1995
E	–	F	–
G	–	H	($474,111)
I	(613%)	J	India
K	–	L	–

1. Row Data
2. Related Action Shortcuts

Related Actions



Google Sheets

Row 9

Summary Related Actions

Delete Row 9
Deletes current Row

Insert a row above Row 9
Inserts a new blank row above the current row.

Insert a row below Row 9
Inserts a new blank row below the current row.

1. Deletes the current row and shifts the below cells up.
2. Inserts a blank row above the current row.
3. Inserts a blank row below the current row.

Important Note: It might take few seconds to reflect the changes after related action in record