AppSheet Automation

Preview Release - Primer

Version 1.1

Updated: Nov 9, 2020

Disclaimer

This primer describes the basic capabilities of the AppSheet Automation Preview release. The Preview features described are not production ready and are intended for use in test environments only. We do not recommend using these Preview features in production or with production data sources. The Preview release is subject to change at any time and is provided to you for evaluation purposes only, without any SLAs. For more information, see the product launch stages.

Version History

Version 1.0	Initial version. Introduction, Contact Sync use case
Version 1.1	Added a new sheets eventing use case - Conference leads appointments generator.

Introducing AppSheet Automation

AppSheet is an intent-aware, no-code application development platform. AppSheet empowers businesses to create applications faster and with less money than a traditional, code-based approach. With AppSheet's true no-code platform, a business user does not need to know how to code in order to create a piece of software. The platform handles the technical heavy-lifting, allowing the business user to focus on the creation of effective apps.

This preview release introduces AppSheet Automation, a robust extension of AppSheet's platform offering new, integrated workflow automation capabilities.

Appsheet Automation:

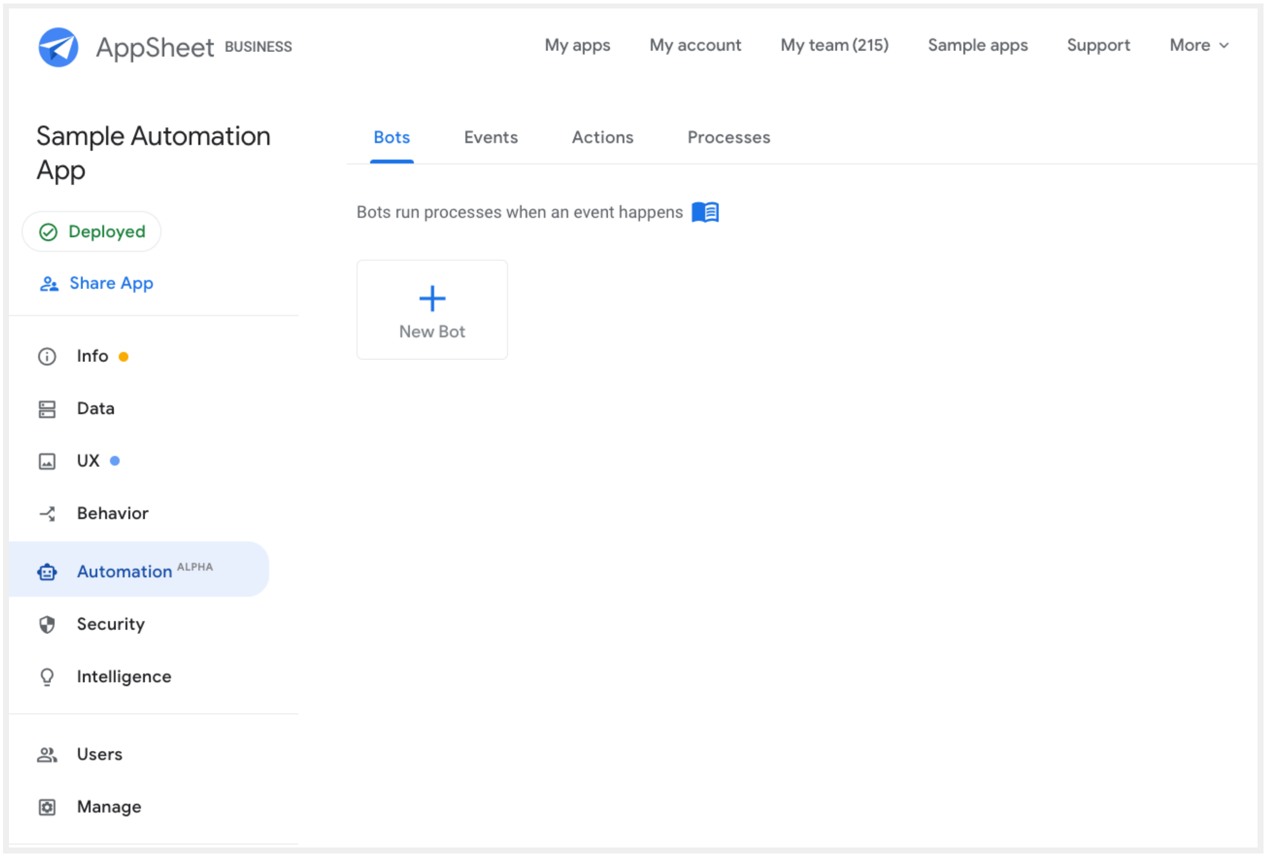
- Unifies AppSheet's intent-aware, no code-platform with process/workflow automation as a first class feature.
- Offers Intelligent Process Authoring & Runtime with rich connectivity, allowing users to author and execute their business processes using entities.
- Introduces new concepts (including Bots and Process) to enrich the core AppSheet platform.
- Addresses the long tail of human centric processes, document based workflows and application integration use cases.

Key AppSheet concepts

This section describes some of the basic concepts introduced with the preview release of AppSheet Automation.

Automation

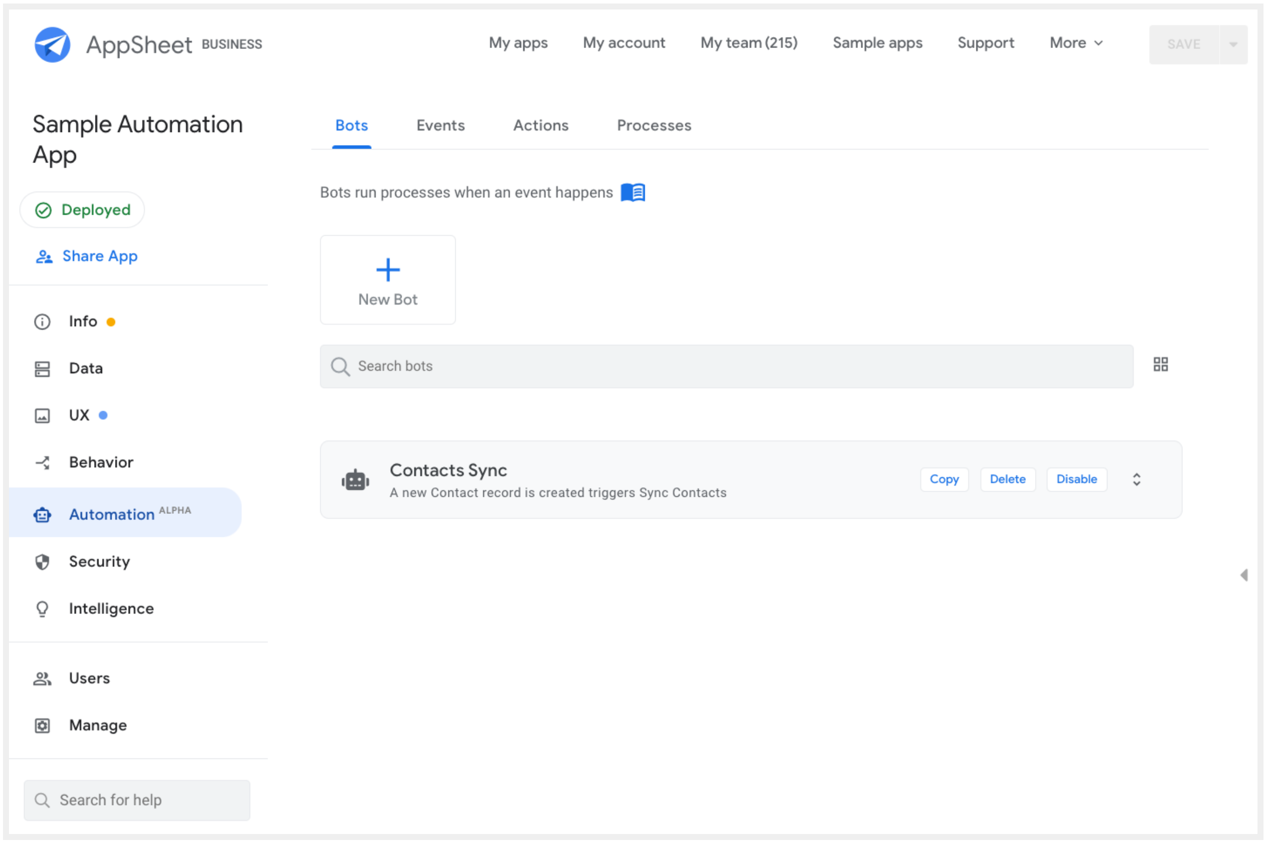
A new left-hand navigation item in the UI provides access to the new automation features of the AppSheet platform. AppSheet *automation* enables you to configure bots, events, actions and processes to carry out common business processes and workflows, as described below.



Bot

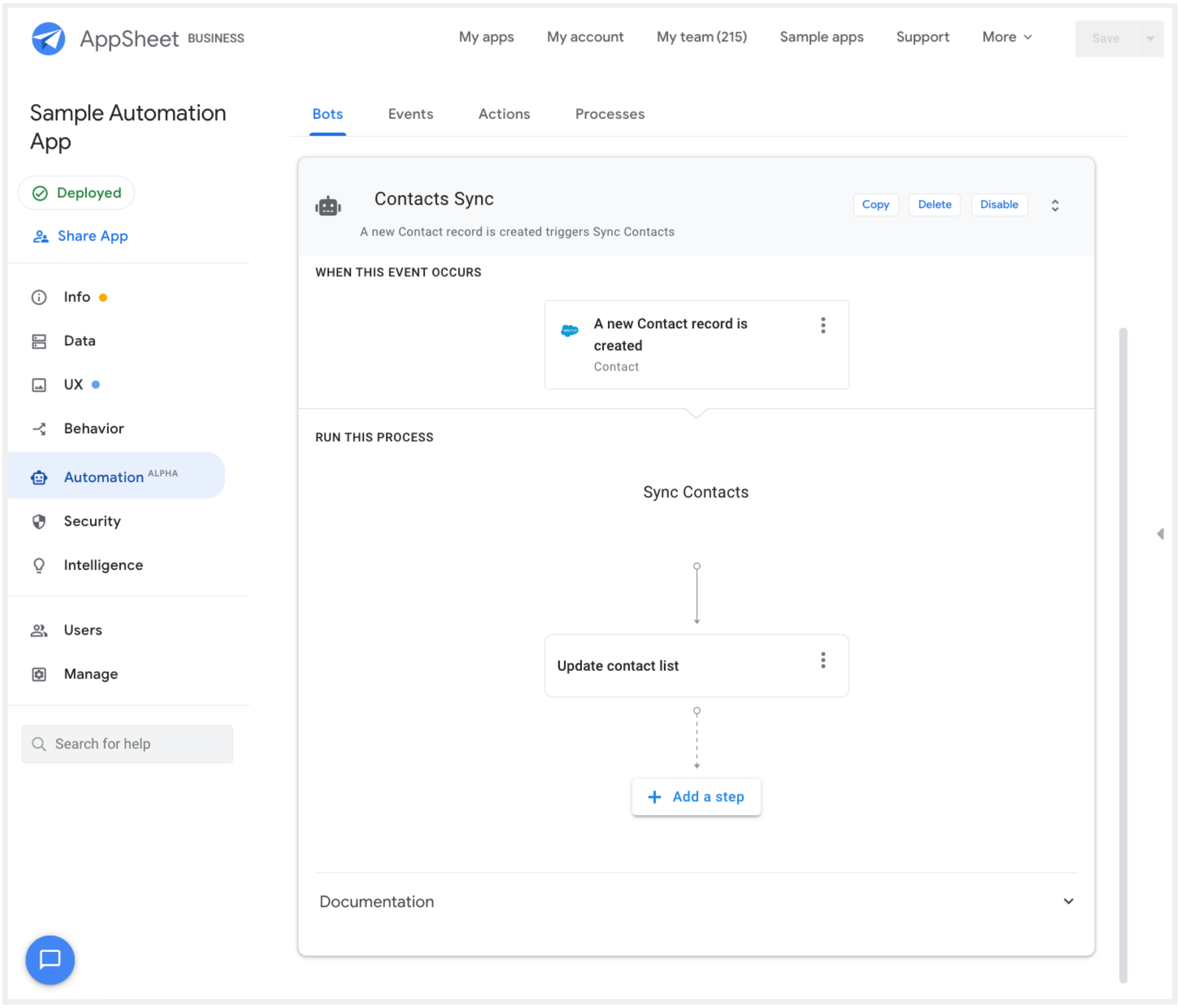
Bots are used to create automations by combining events with processes. Bots can be configured to trigger a process on detection of a specified event or according to a predetermined schedule. Once enabled, bots run in the background to listen for event triggers or trigger processes on a schedule. To terminate the process automation, you can disable the bot.

In the example shown below, the bot is configured to trigger the "Sync Contacts" process anytime a new "Contact" record is created in Salesforce. This bot will continue to listen for the order creation event and fire the associated process when it event occurs until it is disabled.



Bots run processes when an event happens

Contacts Sync
A new Contact record is created triggers Sync Contacts

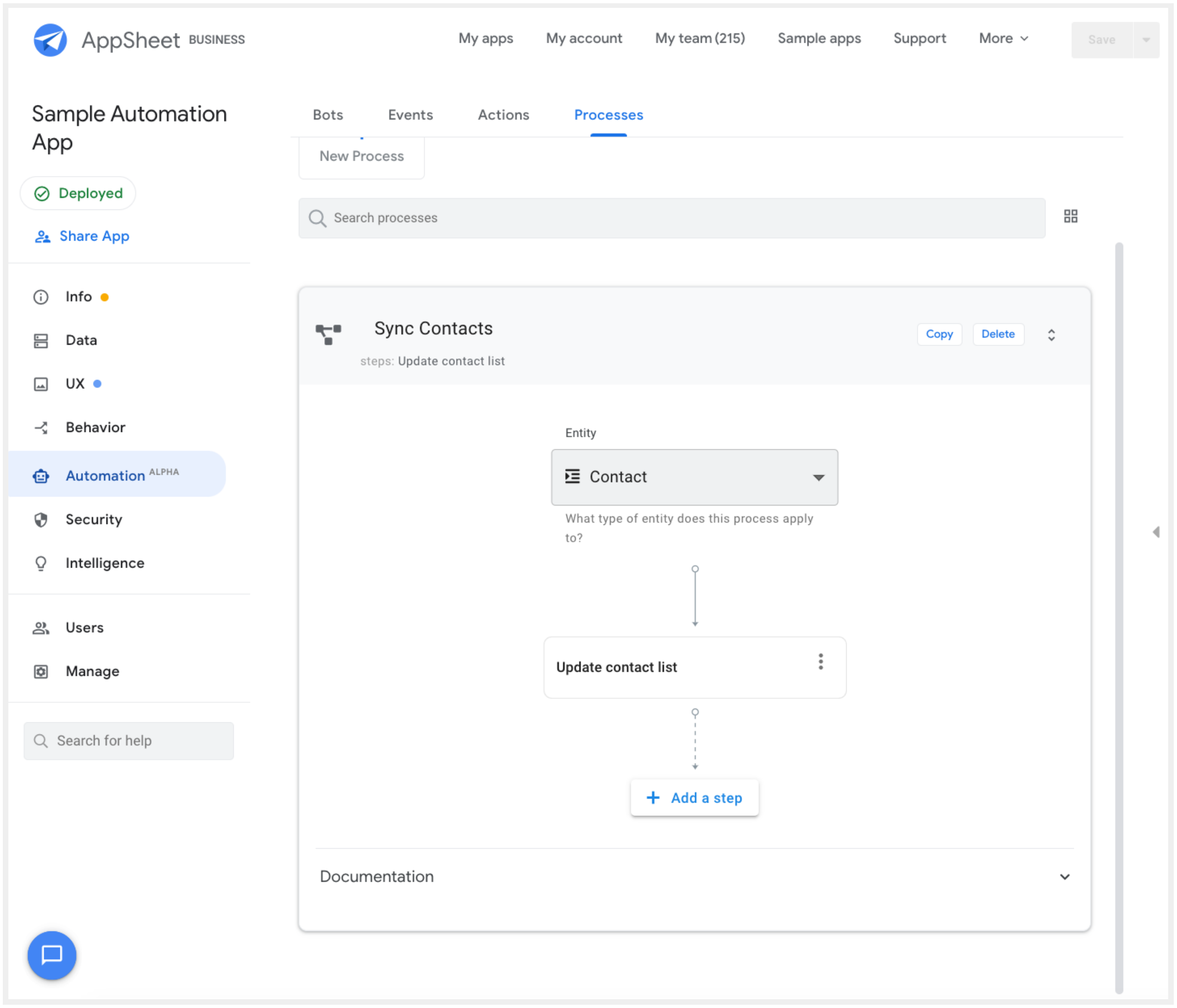


Process

A *process* represents a typical business process. For example, an "order approval" is a business process. A process contains the following elements:

- A sequence of steps
- An input
- An output (this is optional)
- An event that triggers the process

For example, in the screenshot below, you can see a sample "Sync Contacts" process. In this example, the *event* that triggers the process is the addition of a new contact record.

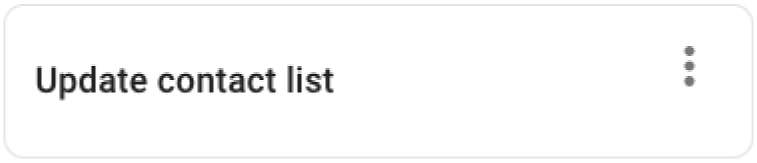


Step

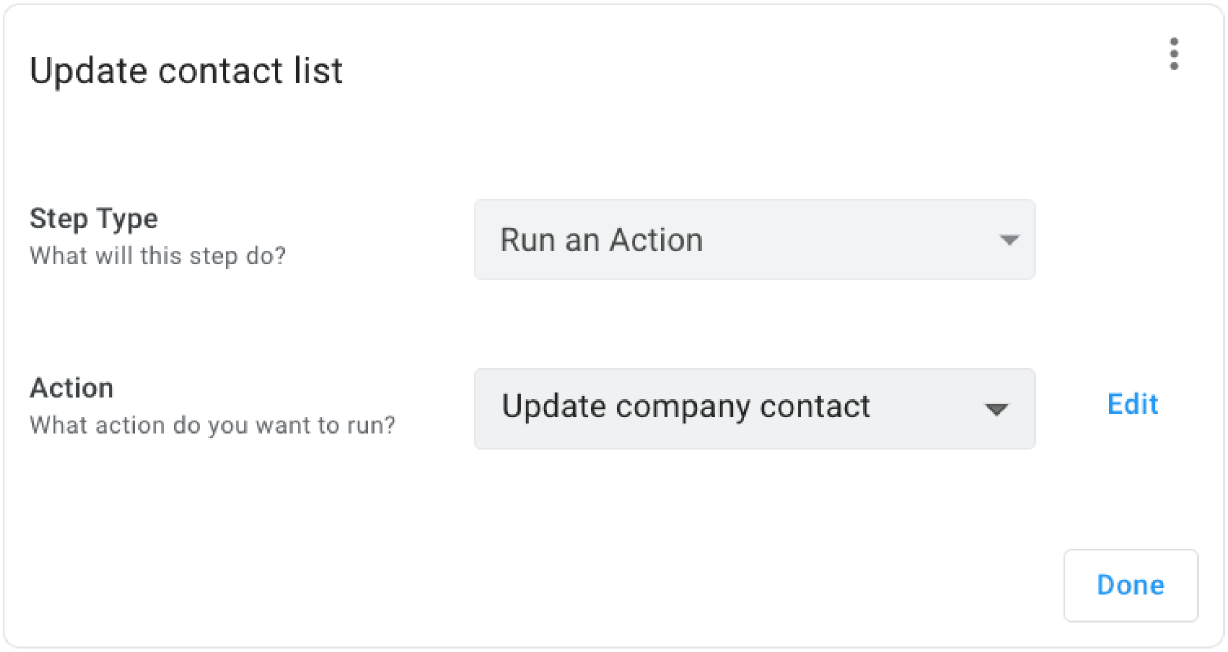
A *step* is a basic element in a process that defines an action to be carried out on the process input.

In our example, the *step* is “Update contact list.” The step will run an action based on the *input*. In this case, the input is the new contact record.

When collapsed, a step looks like this in the UI view of a process:



When expanded, a step looks like the image below:


Update contact list

Step Type
What will this step do?
Run an Action

Action
What action do you want to run?
Update company contact

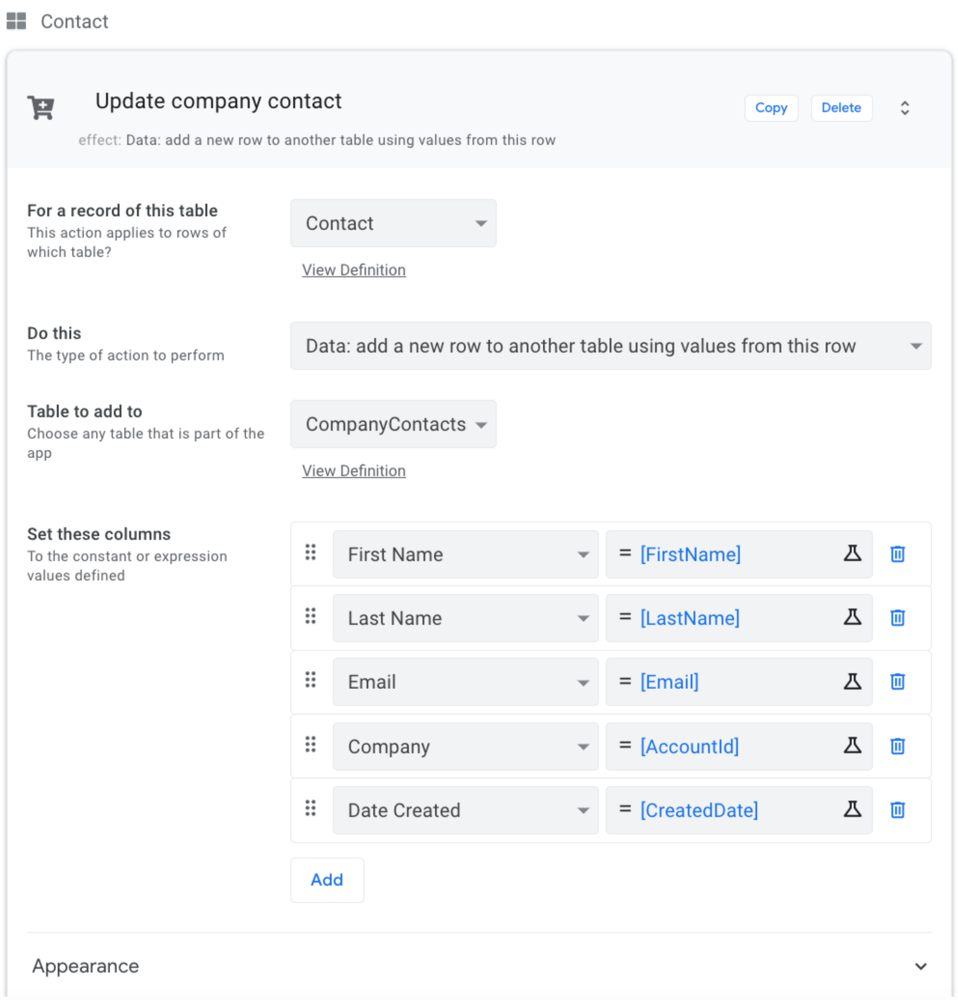
Action

As noted above, steps define *actions* to be carried out in the process. There are two types of actions:

- Data actions
- External communication actions

Data actions

Data actions perform changes to the underlying data of a process. In this example, you can use data actions to add, update, or delete information about the new contact. The image below indicates that upon detection of a new contact record, a new row should be added to the "CompanyContacts" table to add the new record.



Contact

Update company contact

effect: Data: add a new row to another table using values from this row

For a record of this table This action applies to rows of which table?	Contact (View Definition)
Do this The type of action to perform	Data: add a new row to another table using values from this row
Table to add to Choose any table that is part of the app	CompanyContacts (View Definition)

Set these columns To the constant or expression values defined

First Name	= [FirstName]
Last Name	= [LastName]
Email	= [Email]
Company	= [AccountId]
Date Created	= [CreatedDate]

External actions

External actions enable you to communicate information about your app or its data. External actions can use include the following:

- Customizing and sending email.
- Customizing and sending SMS messages.
- Customizing and sending Notification messages to Android and Apple phones.
- Invoking external web services via webhooks.

Currently “External actions” aren’t supported.

Event

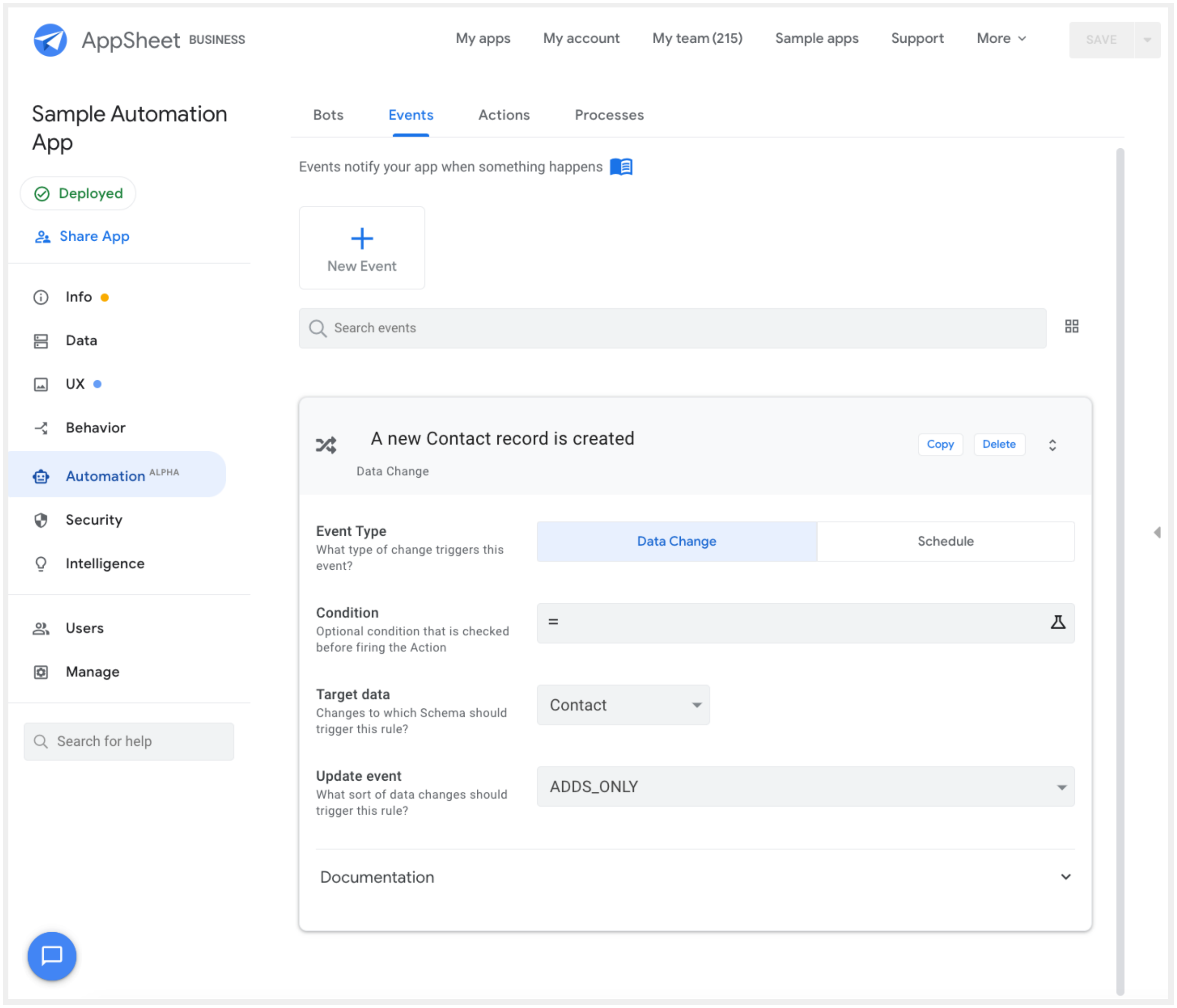
An *event* represents a change to an entity (table). Each event has the following attributes:

- **Event type.** This attribute indicates the kind of change that triggers the event.
- **Condition (optional)**. If specified, this condition is checked before firing the Action.

- **Target data**. Indicates which schema would trigger the event, if changed.
- **Update event**. Indicates the type of schema change that would trigger the event (i.e. additions, updates, deletions, etc.).

In our example, as shown in the image below, the New Event is a "Data Change" event only triggered by additions to the "Contact" data schema.

Currently only "Data Change" events are supported.



Events notify your app when something happens

A new Contact record is created
Data Change

Event Type
What type of change triggers this event?
Data Change

Condition
Optional condition that is checked before firing the Action
=

Target data
Changes to which Schema should trigger this rule?
Contact

Update event
What sort of data changes should trigger this rule?
ADDS_ONLY

Before you begin

Before testing the AppSheet Automation preview features, make sure you have completed the following steps:

- Created an account on appsheet.com.
- Obtained access to the Automation preview release.
- Created a copy of the Sample Automation App for testing your new automations.

Now you are ready to explore AppSheet Automation!

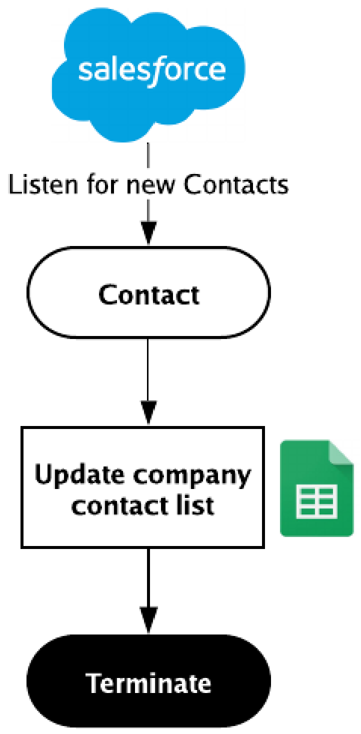
Contact Sync

In this brief tutorial, you will set up an automated process to sync contacts between two systems. This example implements a solution to a common business process scenario. When a new contact is created in Salesforce, AppSheet will automatically update a Google Sheet with the new contact.

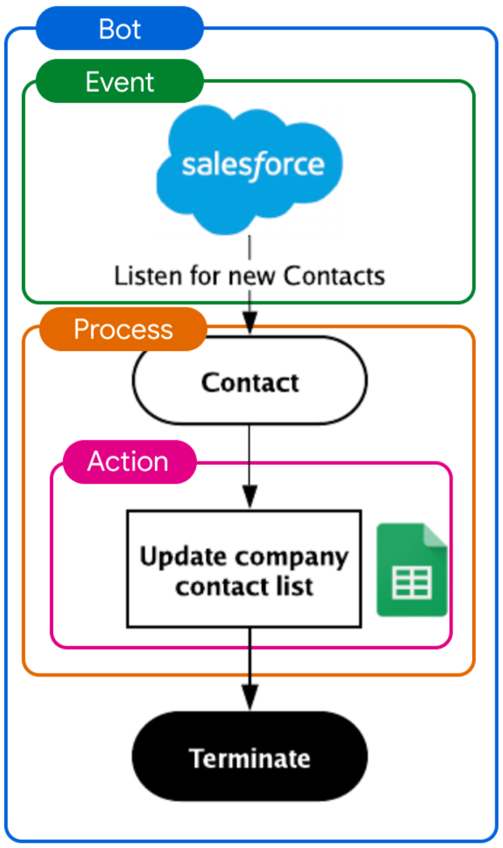
Implementing contact sync with AppSheet Automation involves the following steps:

1. Creating the Bot
2. Configuring the Event
3. Configuring the Process (steps and actions)

The following figure depicts the high level process flow:



The next figure depicts the high level process flow along with the corresponding AppSheet automation concepts:



Before you begin this tutorial:

- Make sure you can log into the AppSheet Editor UI and access the copy of the Sample Automation App you created as a prerequisite in Before you begin.
- Follow the steps in the Appendix to configure a Contact entity from a Salesforce data source and Company Contacts entity from a Google Sheets data source.

Automations begin with configuring the bot. A bot binds an *event* (ex: a new contact created in Salesforce) to a *process* (ex. a sequence of steps to update a contact sheet). The intuitive AppSheet editor enables you to create all the components of the automation/bot in-situ. That is, you can configure the event and process (including actions) without needing to switch between tabs to configure individual components.

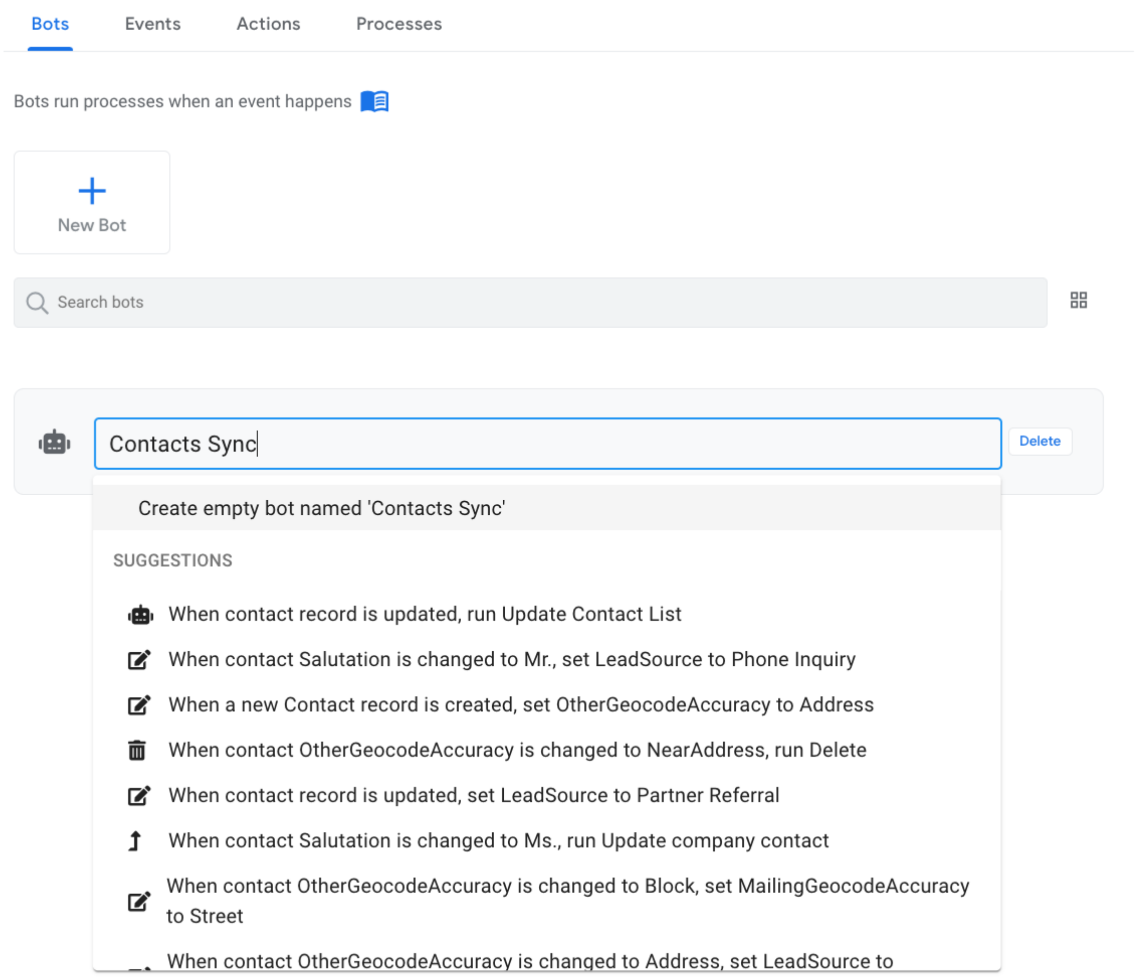
In addition, AppSheet Automation is an intent-aware platform. The platform understands user intent and will recommend configuration options that align with what you are trying to achieve.

The following sections guide you through configuring your end to end automation *in-situ* using a bot.

Creating the bot

Follow the steps below to create a new bot:

a. From the AppSheet UI, navigate to the **Bots** tab and click **New Bot**.
b. You will see a few suggested actions. Type“Contacts Sync” into the dialog and you will see a suggestion to “Create empty bot named ‘Contacts Sync.’”, as shown in the image below.
c. Select **Create empty bot named ‘Contacts Sync’** to create the bot.


Bots
Events
Actions
Processes
Bots run processes when an event happens
New Bot
Search bots
Contacts Sync
Delete
Create empty bot named 'Contacts Sync'
SUGGESTIONS

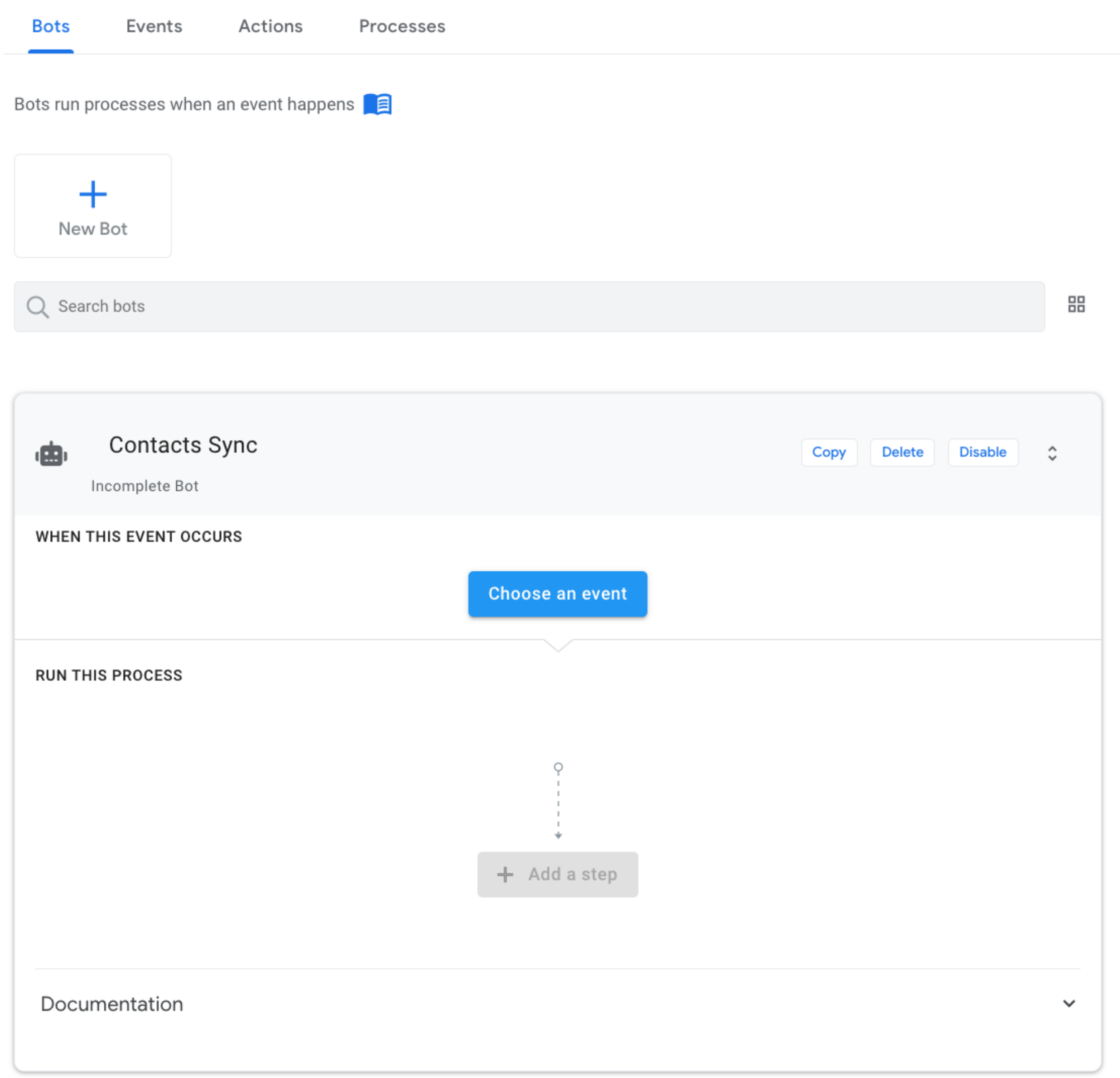
When contact record is updated, run Update Contact List
When contact Salutation is changed to Mr., set LeadSource to Phone Inquiry
When a new Contact record is created, set OtherGeocodeAccuracy to Address
When contact OtherGeocodeAccuracy is changed to NearAddress, run Delete
When contact record is updated, set LeadSource to Partner Referral
When contact Salutation is changed to Ms., run Update company contact
When contact OtherGeocodeAccuracy is changed to Block, set MailingGeocodeAccuracy to Street
When contact OtherGeocodeAccuracy is changed to Address, set LeadSource to

Configuring the event

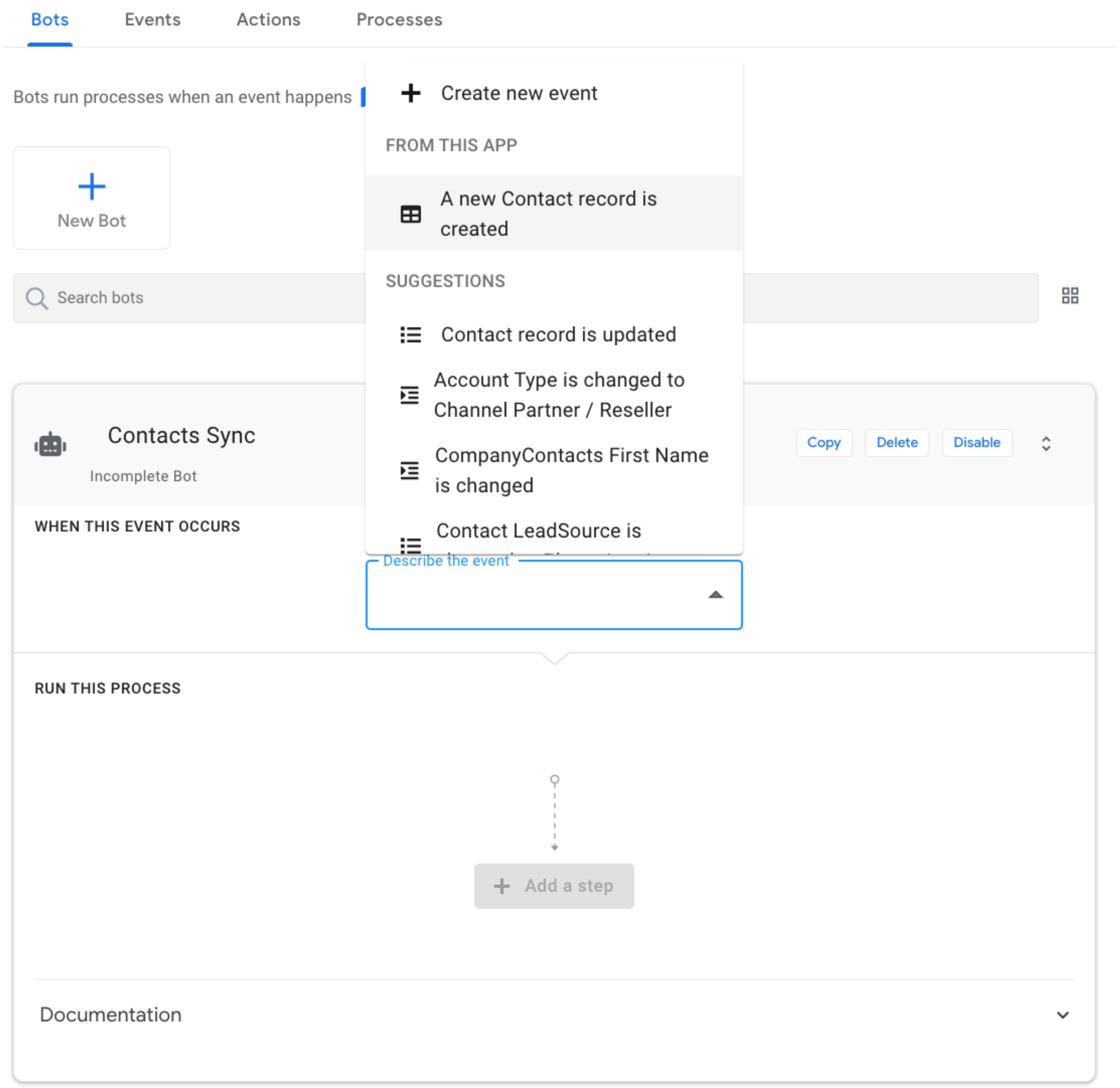
Once the bot is created, you can define the event for the bot. In this example, the event trigger is the addition of new contacts to Salesforce.

To create an event:

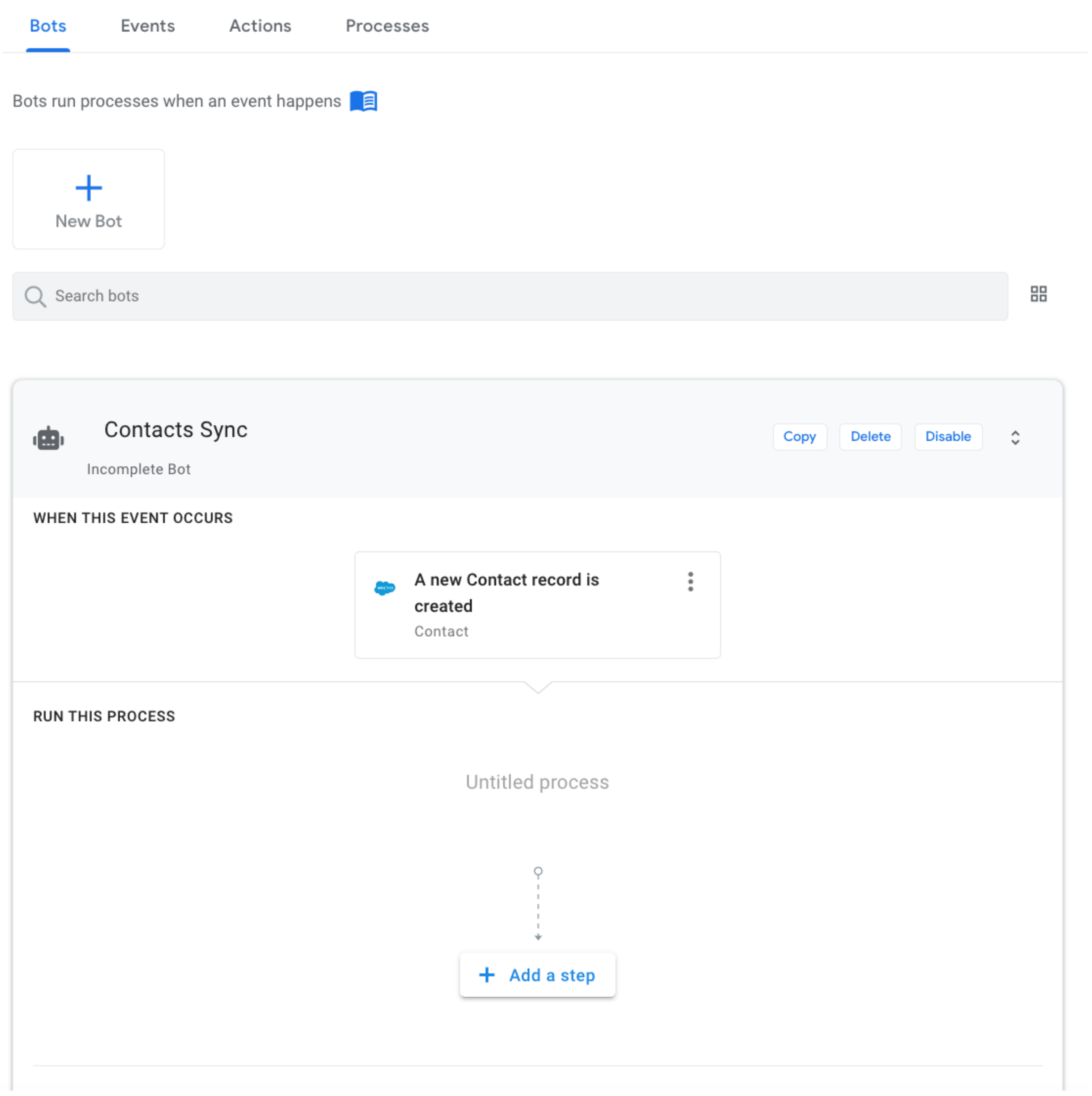
1. Select the bot you just created in the previous section.
2. Click **Choose an event**.
3. Suggested events will appear.
4. Select **A new Contact record is created** to have the event automatically created for you. The platform does this for you and saves you valuable time.



***Note**: If you want to manually configure the event, you can navigate to the Events tab, "Create new event" and enter in the relevant details, as shown below.*



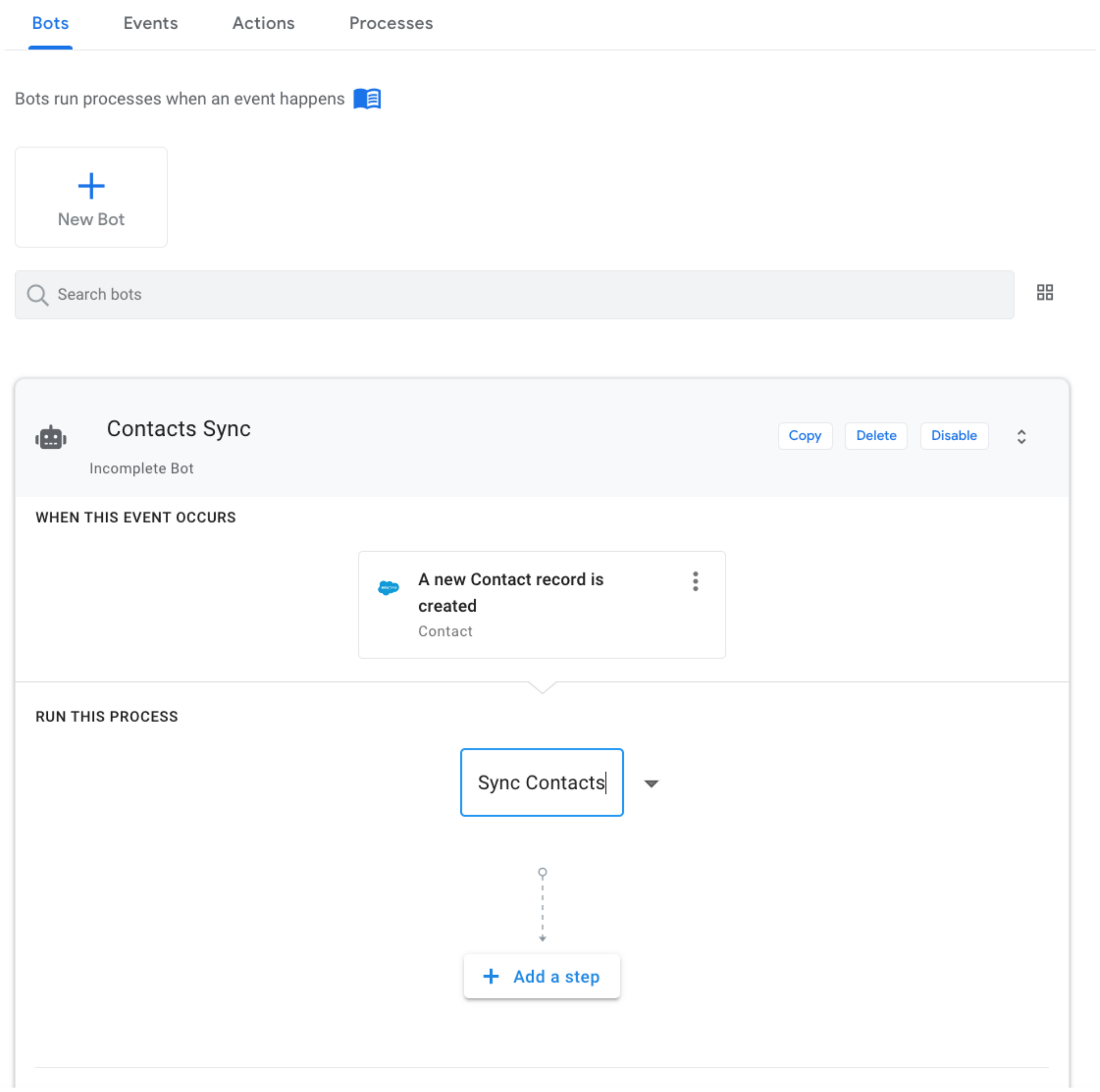
Once your event has been created, your bot editor should look like this:



Configuring the process

Configure the process your event should trigger:

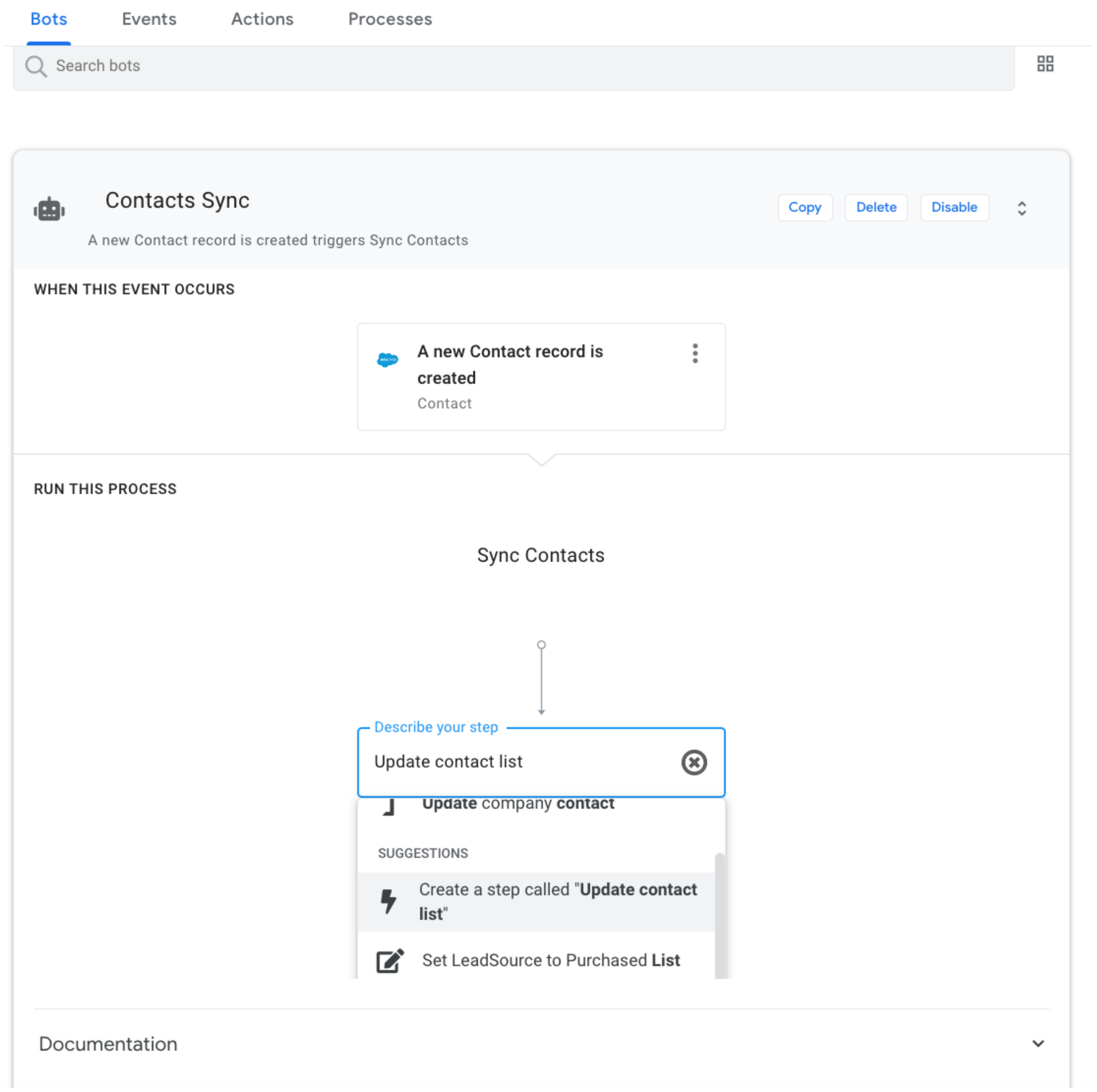
1. In the “Run this process” section of the editor, type “Sync Contacts” into the text box and hit enter to automatically create an empty process.



2. Add a step to the process.

    a. Type “Update contact list” into the text box, as shown below.

    b. Select **Create a step called “Update contact list”** from the suggested actions to automatically create an empty step.



Bots | Events | Actions | Processes

Search bots

Contacts Sync

A new Contact record is created triggers Sync Contacts

Copy | Delete | Disable

WHEN THIS EVENT OCCURS

A new Contact record is created
Contact

RUN THIS PROCESS

Sync Contacts

Describe your step
Update contact list

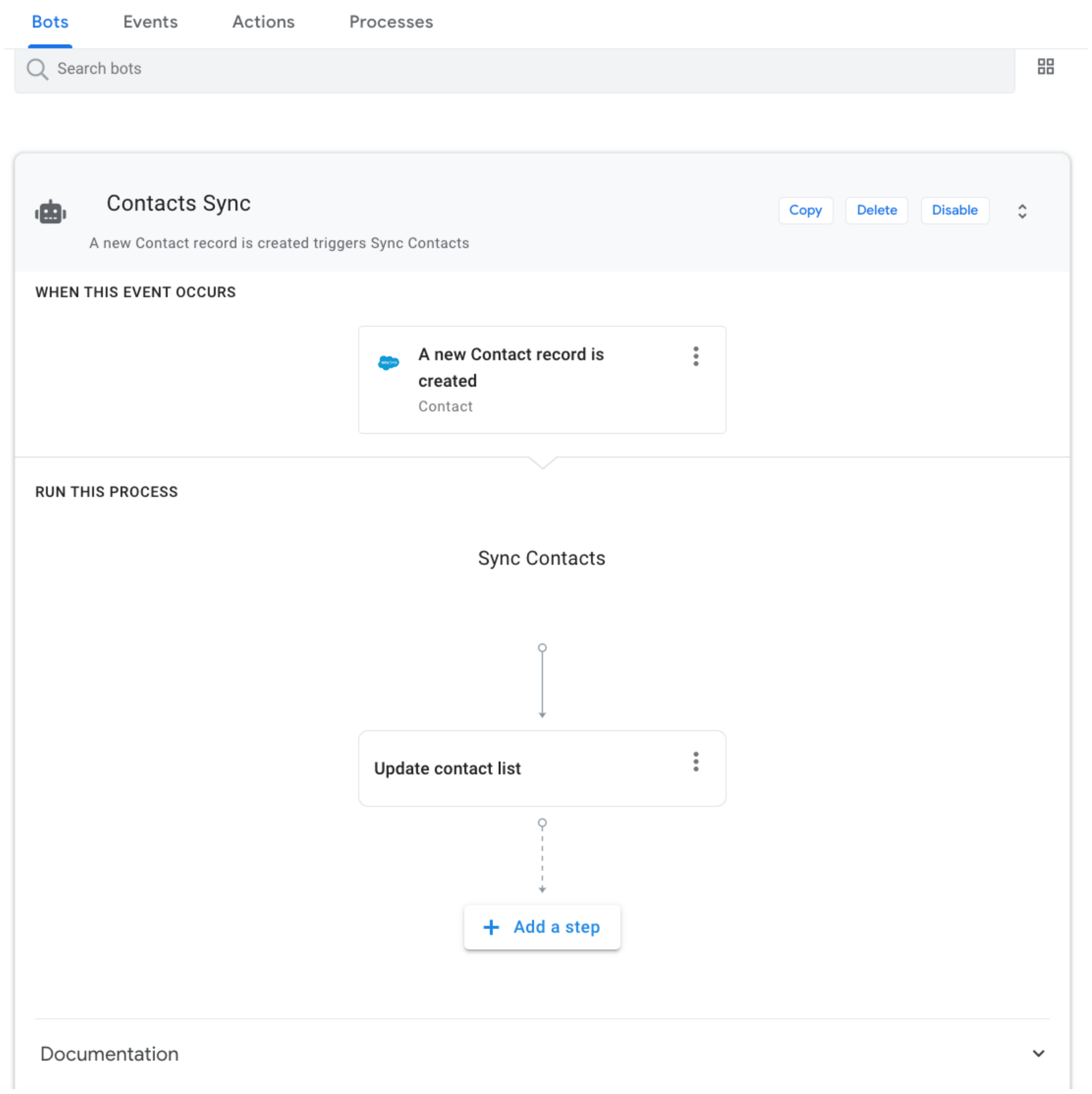
Update company contact

SUGGESTIONS

Create a step called "**Update contact list**"

Set LeadSource to Purchased **List**

Documentation



Contacts Sync

A new Contact record is created triggers Sync Contacts

WHEN THIS EVENT OCCURS

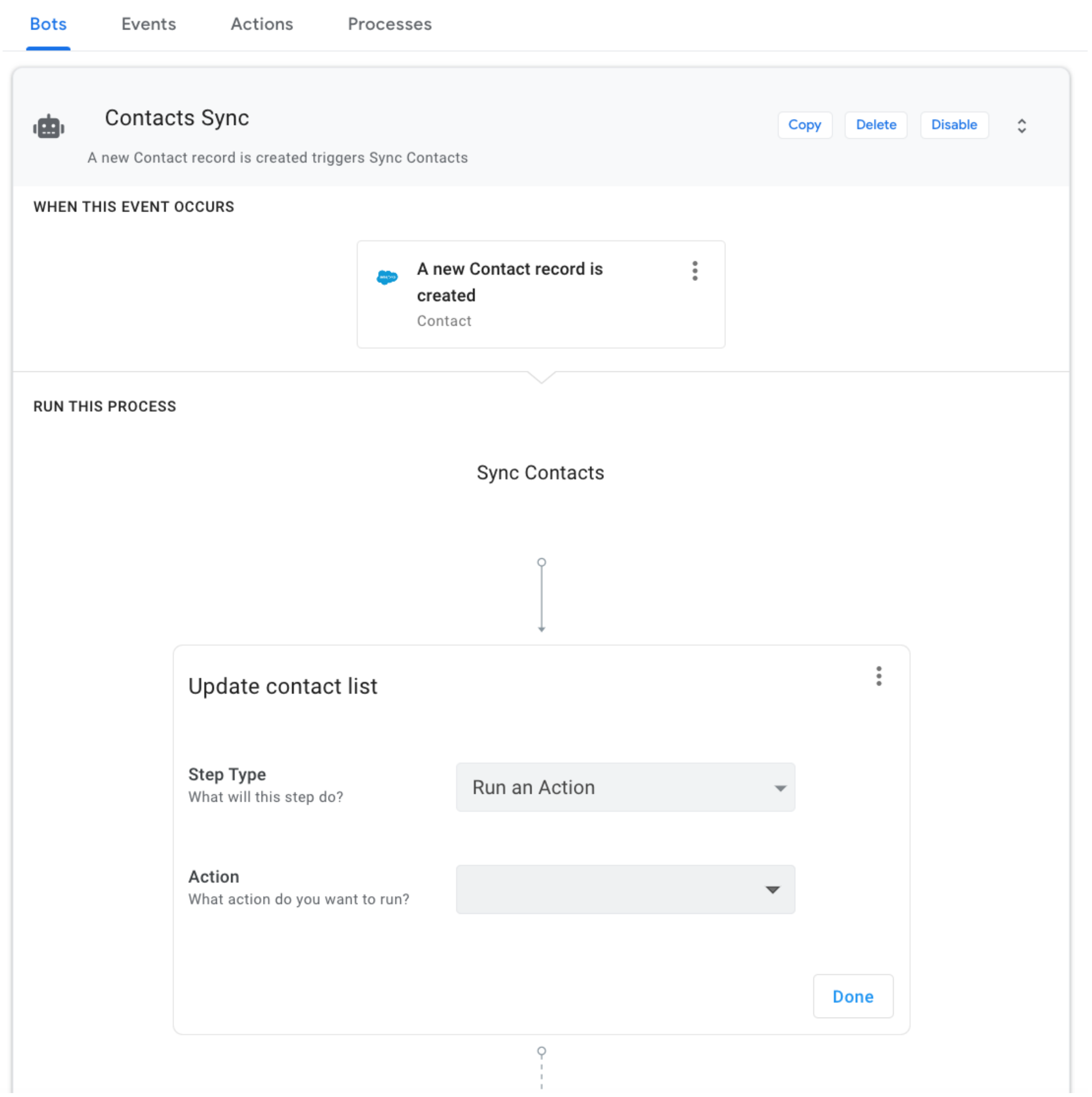
A new Contact record is created
Contact

RUN THIS PROCESS

Sync Contacts

Update contact list

3. Click on the **Update contact list** step to expand its definition, as shown below:



Contacts Sync

A new Contact record is created triggers Sync Contacts

WHEN THIS EVENT OCCURS

A new Contact record is created

Contact

RUN THIS PROCESS

Sync Contacts

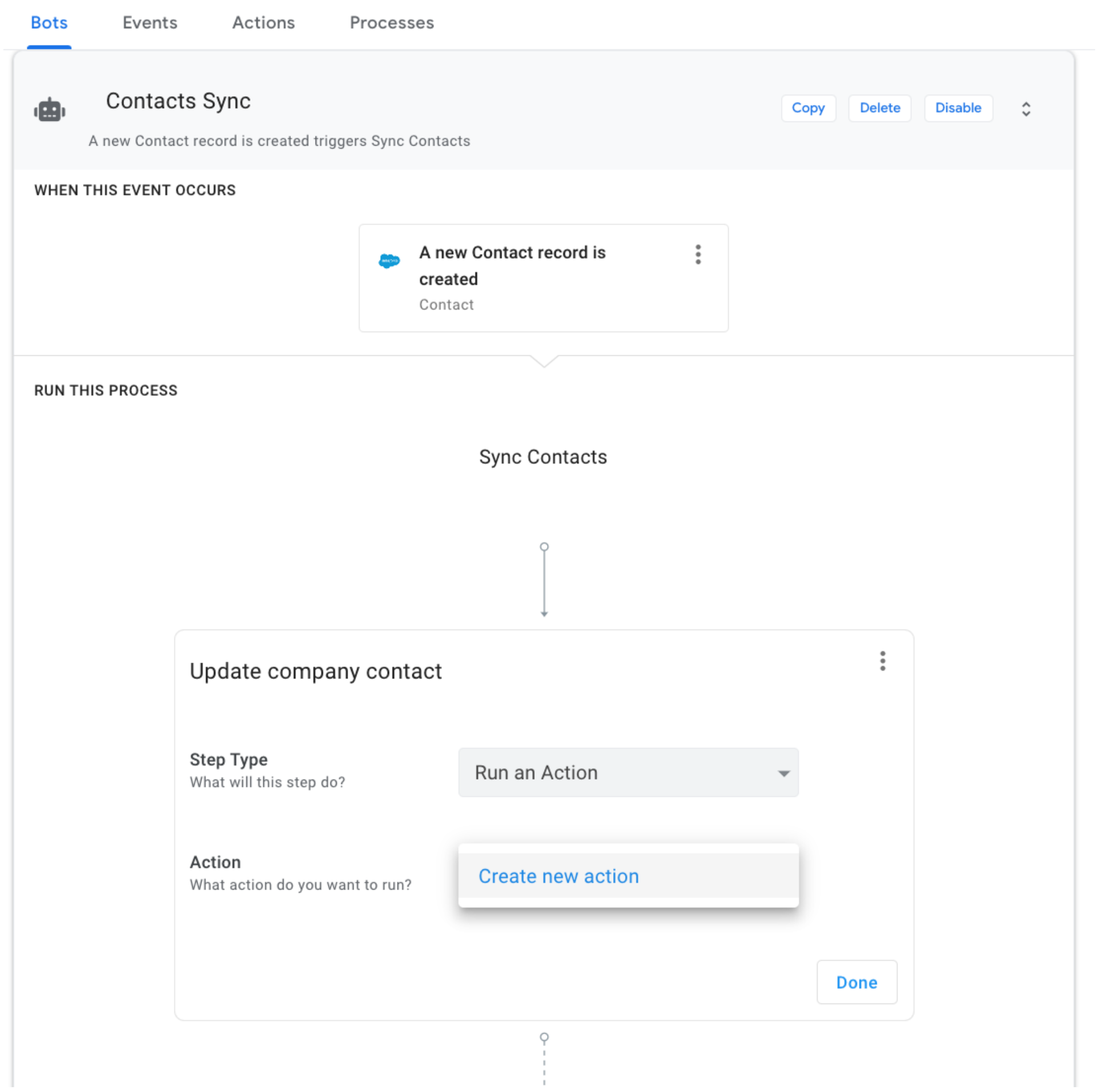
Update contact list

Step Type
What will this step do?

Run an Action

Action
What action do you want to run?

4. In the “Action” dropdown, select **Create new action**.



Contacts Sync

A new Contact record is created triggers Sync Contacts

WHEN THIS EVENT OCCURS

A new Contact record is created
Contact

RUN THIS PROCESS

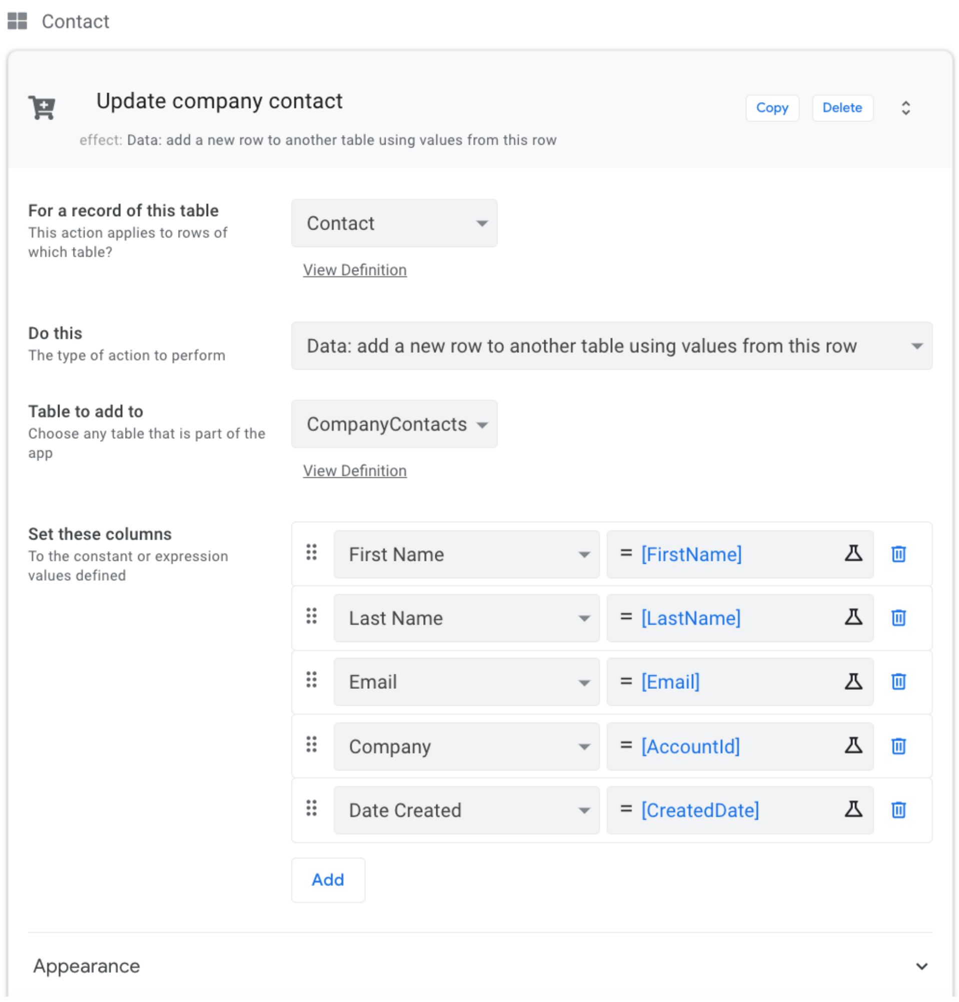
Sync Contacts

Update company contact

Step Type
What will this step do?
Run an Action

Action
What action do you want to run?
Create new action

5. Configure your action as follows:



Contact

Update company contact

effect: Data: add a new row to another table using values from this row

For a record of this table
This action applies to rows of which table?
Contact
View Definition

Do this
The type of action to perform
Data: add a new row to another table using values from this row

Table to add to
Choose any table that is part of the app
CompanyContacts
View Definition

Set these columns
To the constant or expression values defined

First Name	= [FirstName]
Last Name	= [LastName]
Email	= [Email]
Company	= [AccountId]
Date Created	= [CreatedDate]

6. Click **Save**.

You have created and enabled your bot. All the required components, including the event, process, and action, have been created. Each component can be accessed and edited from individual tabs if required.

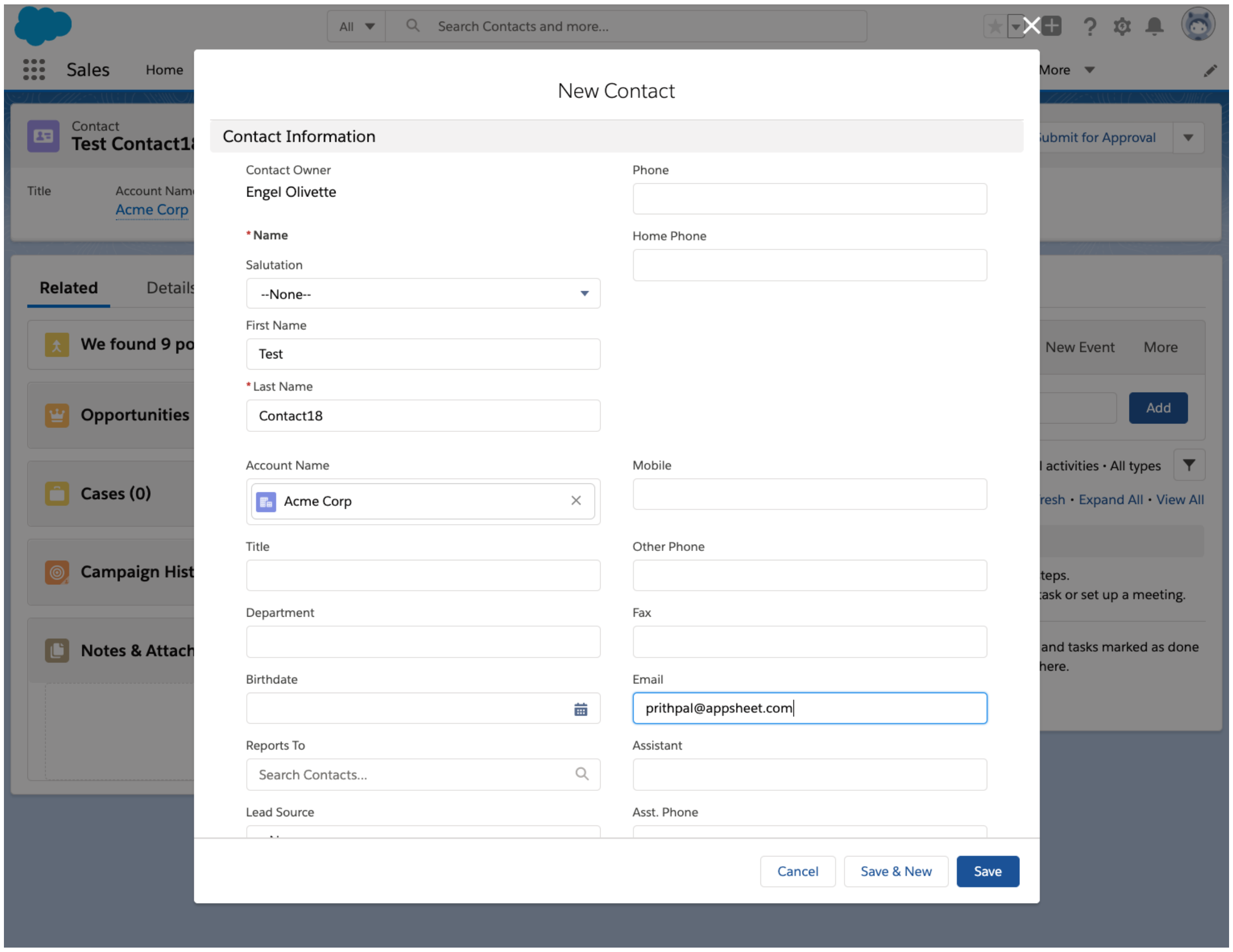
Note*: Make sure your app is deployed. If not, deploy it . The bots will listen for events and trigger processes only if the app is deployed and the specific bot is enabled.*

Now you are ready to test your bot.

Testing your automation

Follow these steps to test your automation:

1. Login to your Salesforce Instance and create a new contact.



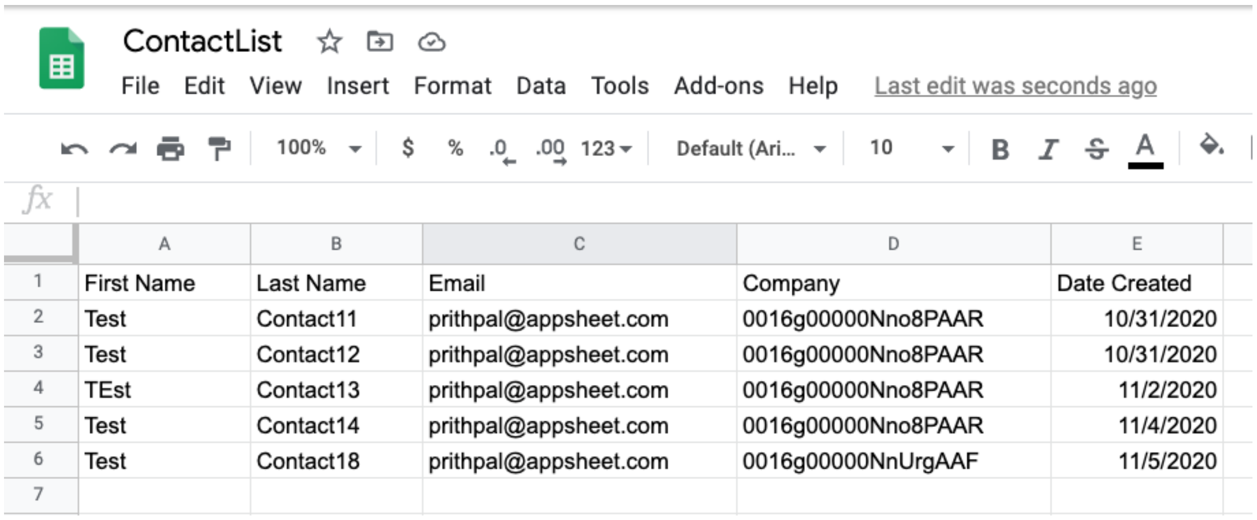
New Contact

Contact Information

Contact Owner: Engel Olivette	Phone:
*Name	Home Phone:
Salutation: --None--	
First Name: Test	
*Last Name: Contact18	
Account Name: Acme Corp	Mobile:
Title:	Other Phone:
Department:	Fax:
Birthdate:	Email: prithpal@appsheet.com
Reports To: Search Contacts...	Assistant:
Lead Source	Asst. Phone

The Bot listens for new Contact events every 60 seconds.

2. In about 60 seconds, check to confirm that a new contact is added to your CompanyList Sheet.

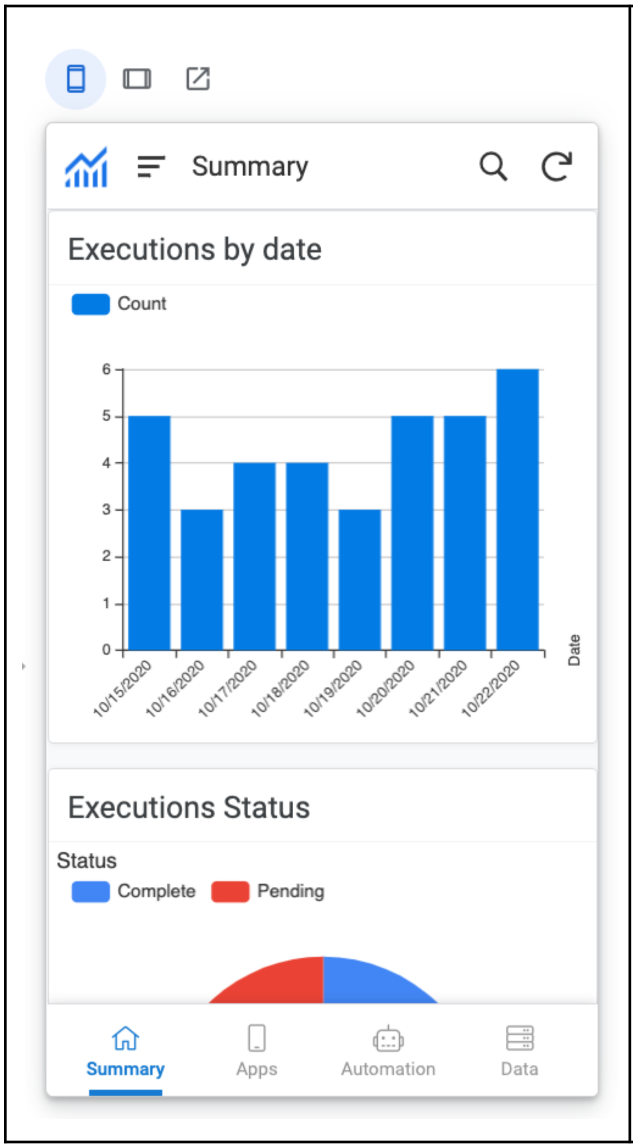

ContactList
File Edit View Insert Format Data Tools Add-ons Help Last edit was seconds ago
100%
Default (Ari...
10

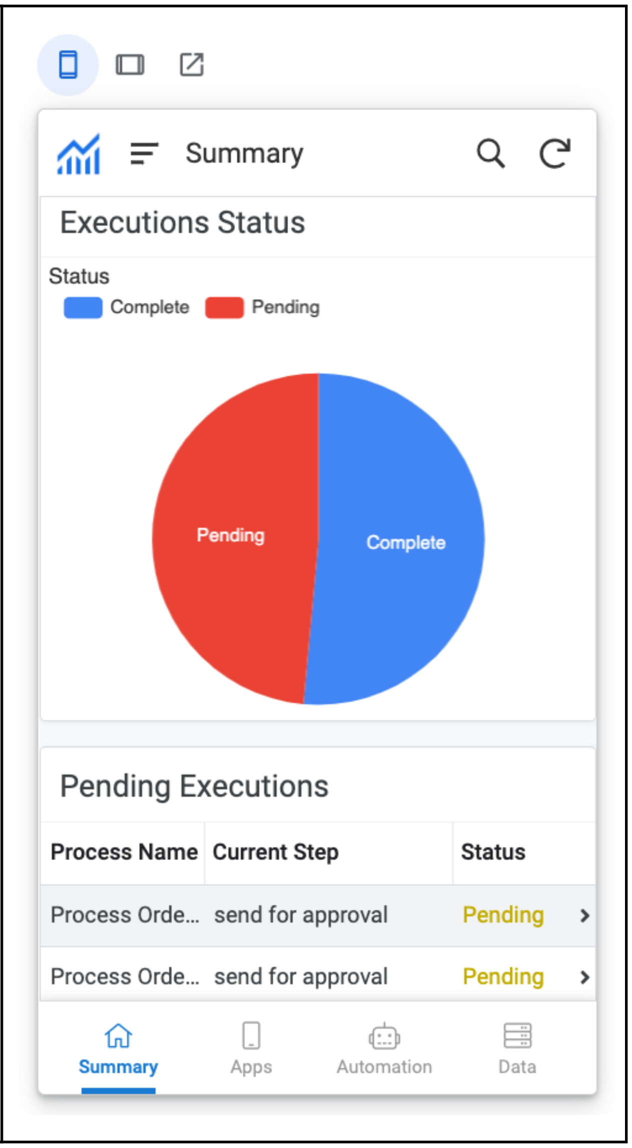
	A	B	C	D	E
1	First Name	Last Name	Email	Company	Date Created
2	Test	Contact11	prithpal@appsheet.com	0016g00000Nno8PAAR	10/31/2020
3	Test	Contact12	prithpal@appsheet.com	0016g00000Nno8PAAR	10/31/2020
4	TEst	Contact13	prithpal@appsheet.com	0016g00000Nno8PAAR	11/2/2020
5	Test	Contact14	prithpal@appsheet.com	0016g00000Nno8PAAR	11/4/2020
6	Test	Contact18	prithpal@appsheet.com	0016g00000NnUrgAAF	11/5/2020
7					

Automation Monitoring

Basic monitoring views have already been added to the Sample Automation App. As you test your bot by adding records in Salesforce, the monitoring views render bot executions data.

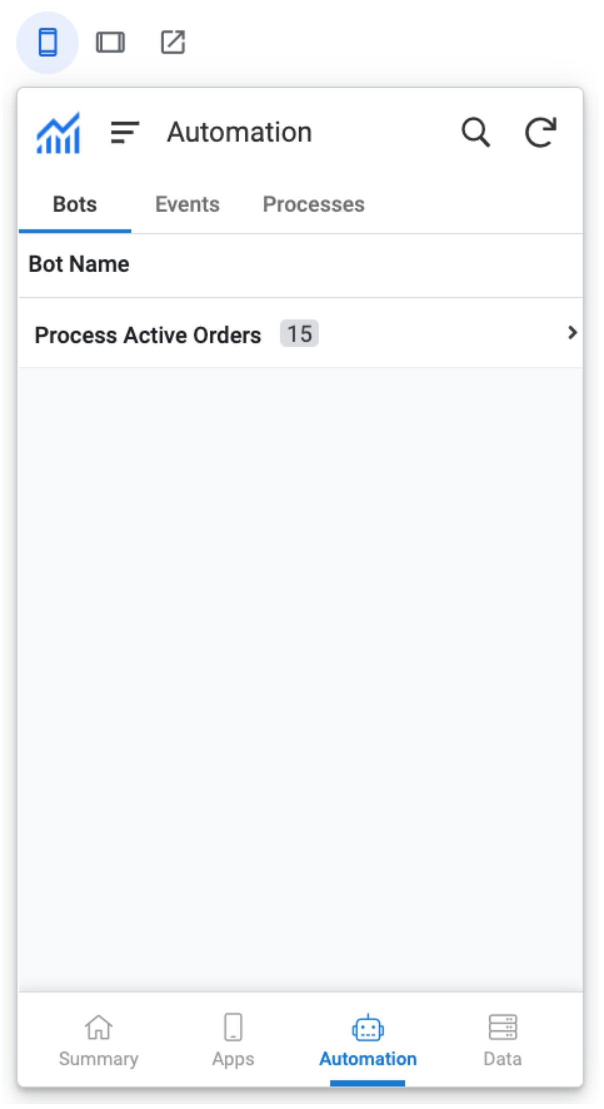
To view the bot executions data, click on the “Summary” and “Automations” views in your app. *Your view may look different from the screen shots presented below.*





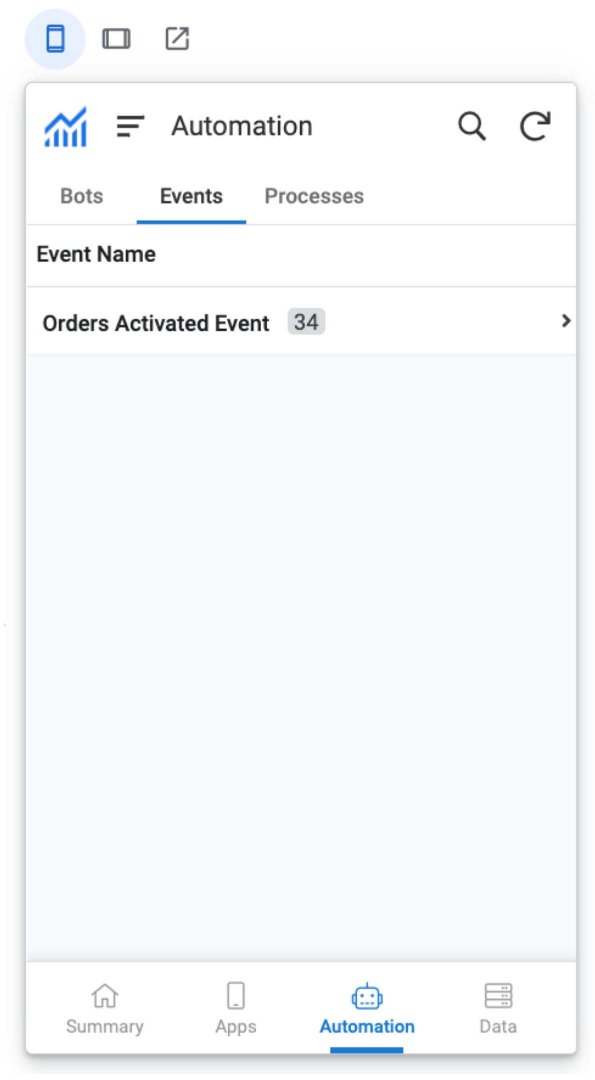
Pending Executions

Process Name	Current Step	Status
Process Orde...	send for approval	Pending
Process Orde...	send for approval	Pending



Bot Name

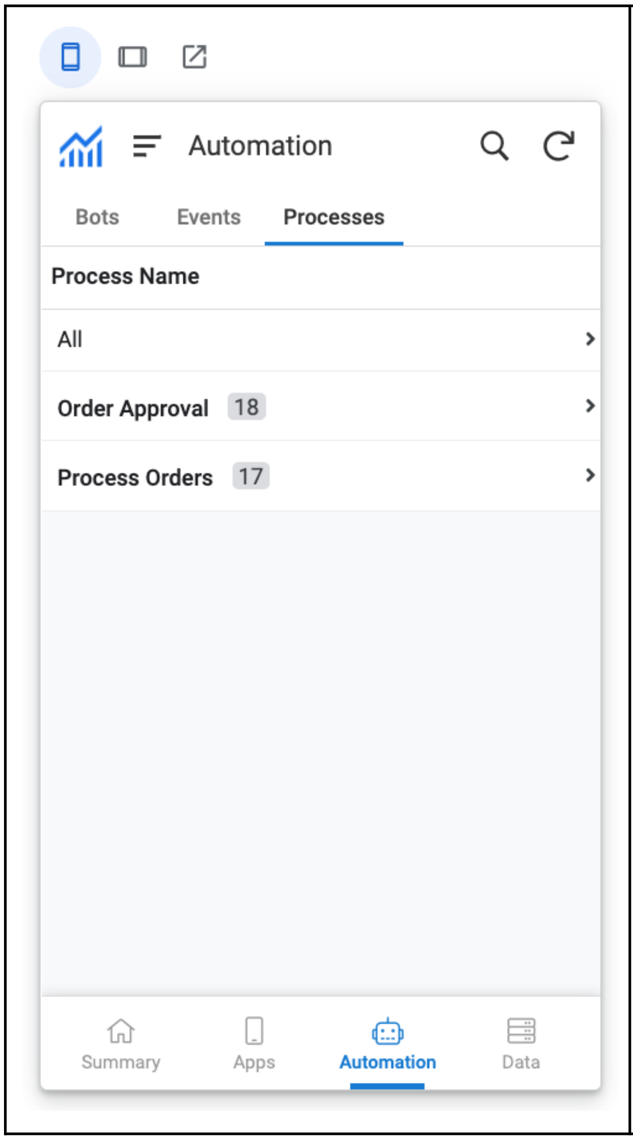
Process Active Orders 15



Event Name

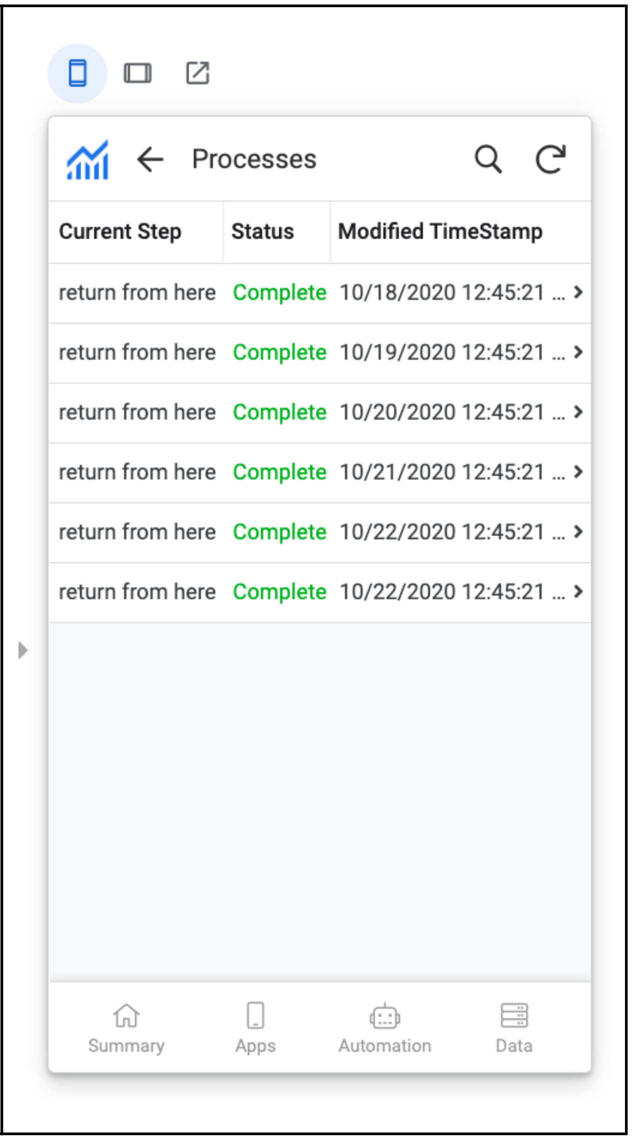
Orders Activated Event 34

Google Confidential - Internal Use Only (Do not print, distribute or share without explicit permission)

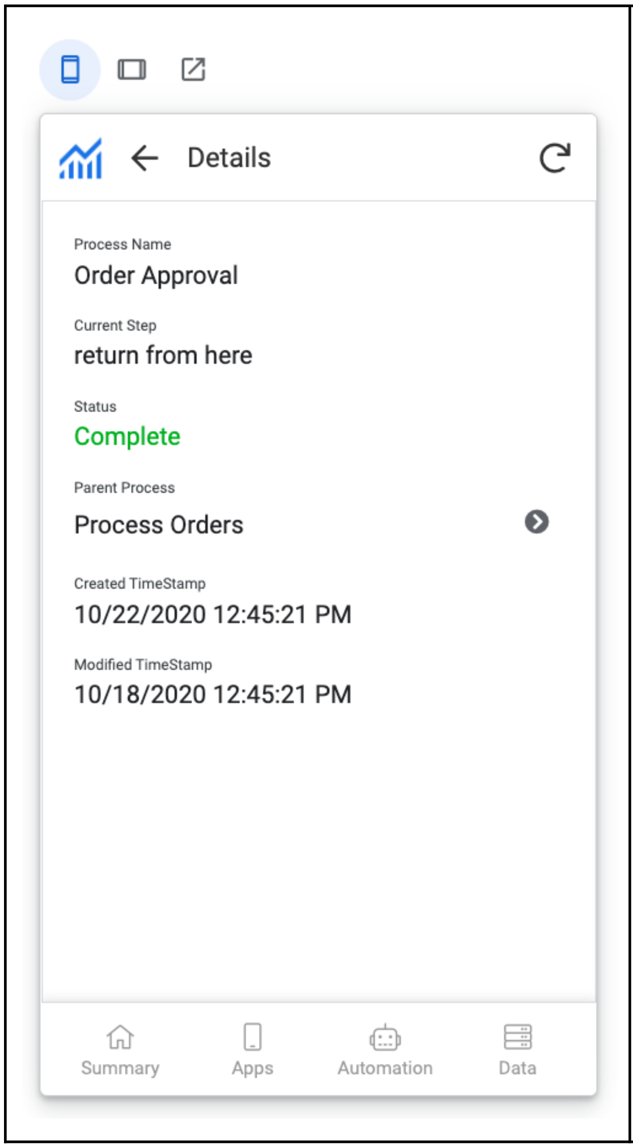

Automation

Bots Events Processes

Process Name
All
Order Approval 18
Process Orders 17


Processes

Current Step	Status	Modified TimeStamp
return from here	Complete	10/18/2020 12:45:21 ...
return from here	Complete	10/19/2020 12:45:21 ...
return from here	Complete	10/20/2020 12:45:21 ...
return from here	Complete	10/21/2020 12:45:21 ...
return from here	Complete	10/22/2020 12:45:21 ...
return from here	Complete	10/22/2020 12:45:21 ...



Process Name
Order Approval
Current Step
return from here
Status
Complete
Parent Process
Process Orders
Created TimeStamp
10/22/2020 12:45:21 PM
Modified TimeStamp
10/18/2020 12:45:21 PM

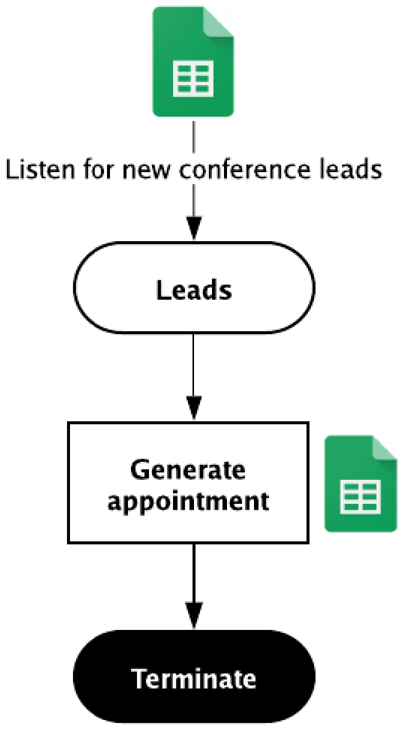
Conference leads appointment generator

In this brief tutorial, you will set up an automated process to create appointments for leads attending a conference that have requested a follow up. When a new lead is created in Google Sheets, AppSheet will determine if a follow up has been requested by the individual and if so automatically create an appointment entry in another Google sheet.

Implementing contact sync with AppSheet Automation involves the following steps:

1. Creating the Bot
2. Configuring the Event
3. Configuring the Process (steps and actions)

The following figure depicts the high level process flow:



Before you begin this tutorial:

- Install the AppSheet Events add-on for Google Sheets.
- Configure the leads and appointments entities (tables) in your new copy of the Sample Automation App.

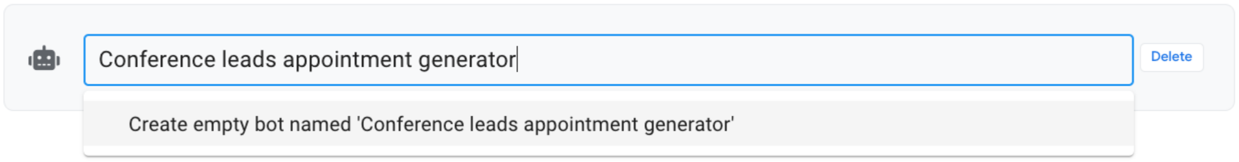
The following sections guide you through configuring your end to end automation *in-situ* using a bot.

Creating the bot

Follow the steps below to create a new bot:

1. From the AppSheet UI, navigate to the **Bots** tab and click **New Bot**.
2. You will see a few suggested actions. Type “Conference leads appointment generator” into the text box and you will see a suggestion to “Create an empty bot named ‘Conference leads appointment generator’”, as shown in the image below.
3. Select **Create empty bot named ‘Conference leads appointment generator’**

to create the bot.


Conference leads appointment generator

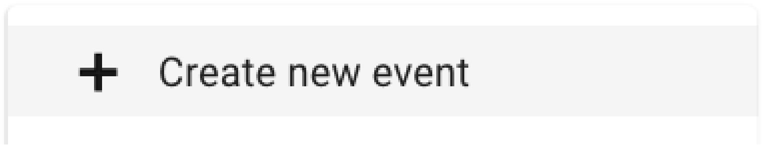
Create empty bot named 'Conference leads appointment generator'

Configuring the event

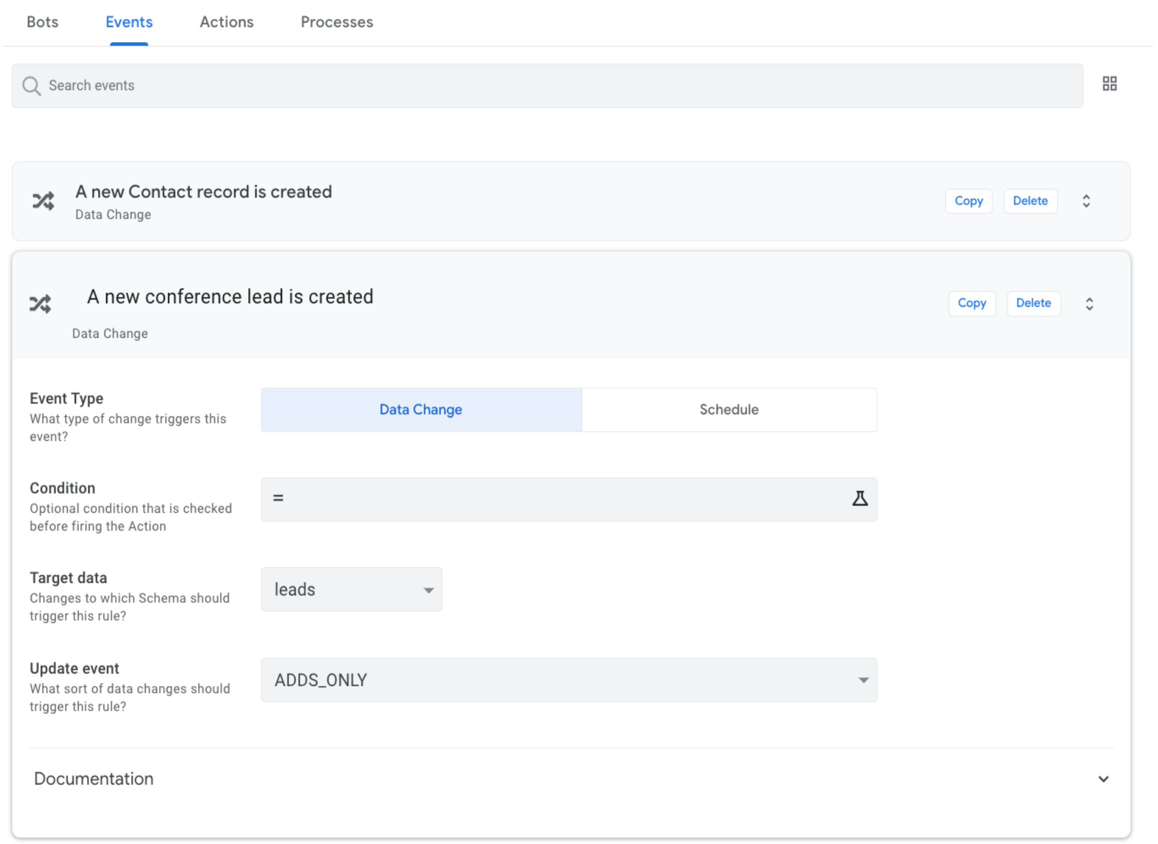
Once the bot is created, you can define the event for the bot. In this example, the event trigger is the addition of new leads to the Leads Google Sheet.

To create an event:

1. From the AppSheet UI, select the bot you created in the previous section.
2. Click **Choose an event**.
3. Suggested events will appear.
4. Select **+ create new event**



5. Once on the event page, configure the event as shown below:



A new Contact record is created
Data Change

A new conference lead is created
Data Change

Event Type
What type of change triggers this event?
Data Change | Schedule

Condition
Optional condition that is checked before firing the Action
=

Target data
Changes to which Schema should trigger this rule?
leads

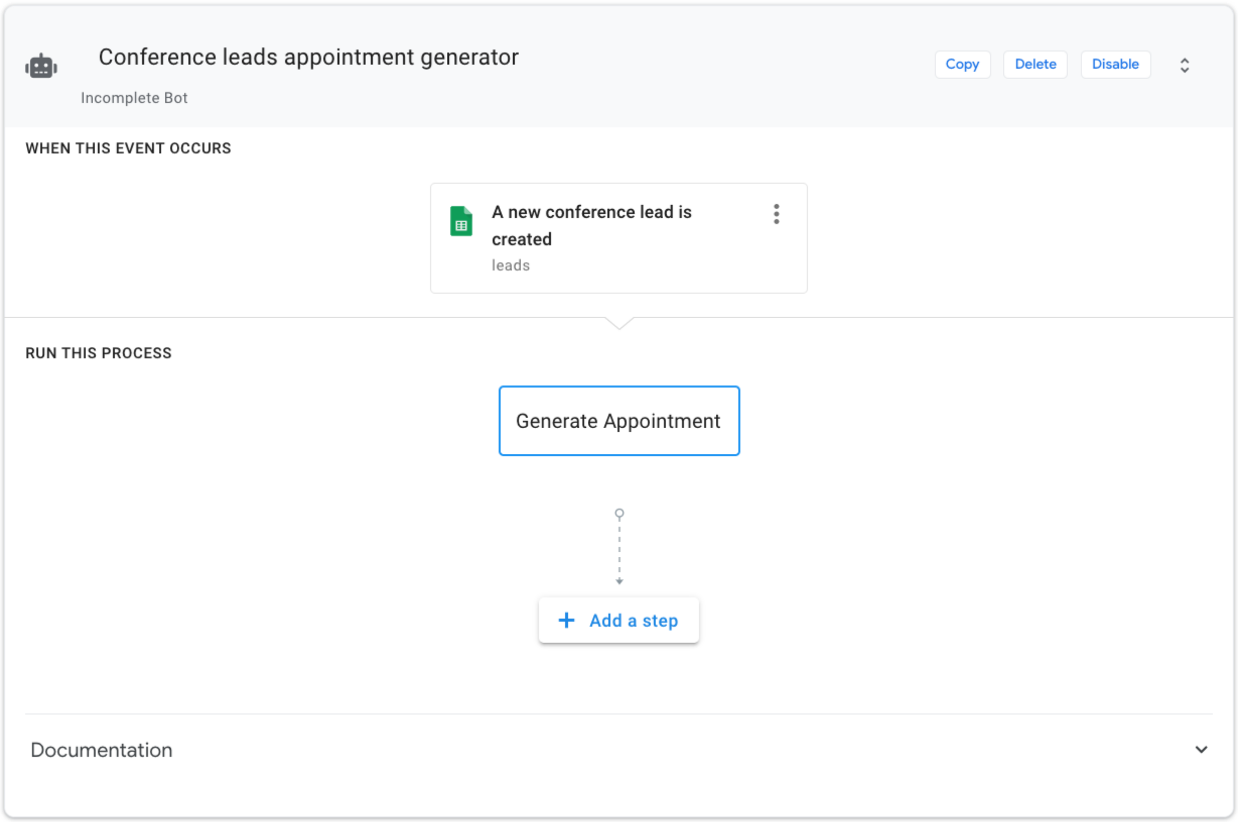
Update event
What sort of data changes should trigger this rule?
ADDS_ONLY

Documentation

Configuring the process

Configure the process your event should trigger:

1. In the "Run this process" section of the editor, type "Generate Appointment" into the text box and hit enter to automatically create an empty process.


Conference leads appointment generator

Incomplete Bot

WHEN THIS EVENT OCCURS

A new conference lead is created

leads

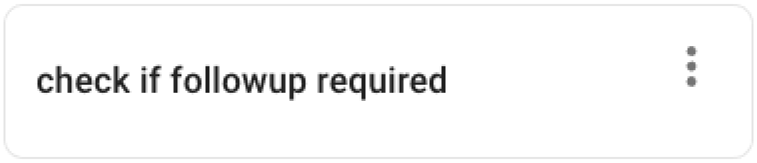
RUN THIS PROCESS

Generate Appointment

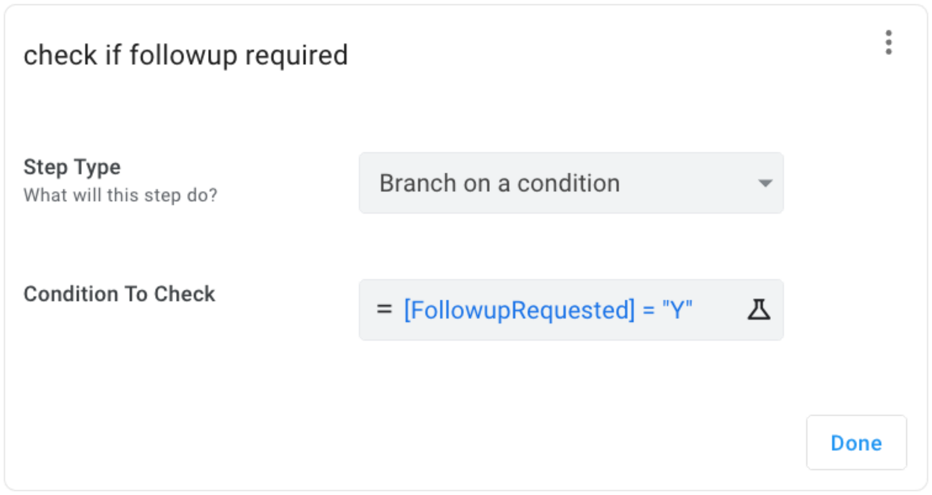
2. Add a step to the process.

    a. Type “check if follow up required” into the text box, as shown below.

    b. Select **Create a step called “check if followup required”** from the suggested actions to automatically create an empty step.


check if followup required

    c. Configure the step to look like this:


check if followup required

Step Type
What will this step do?
Branch on a condition

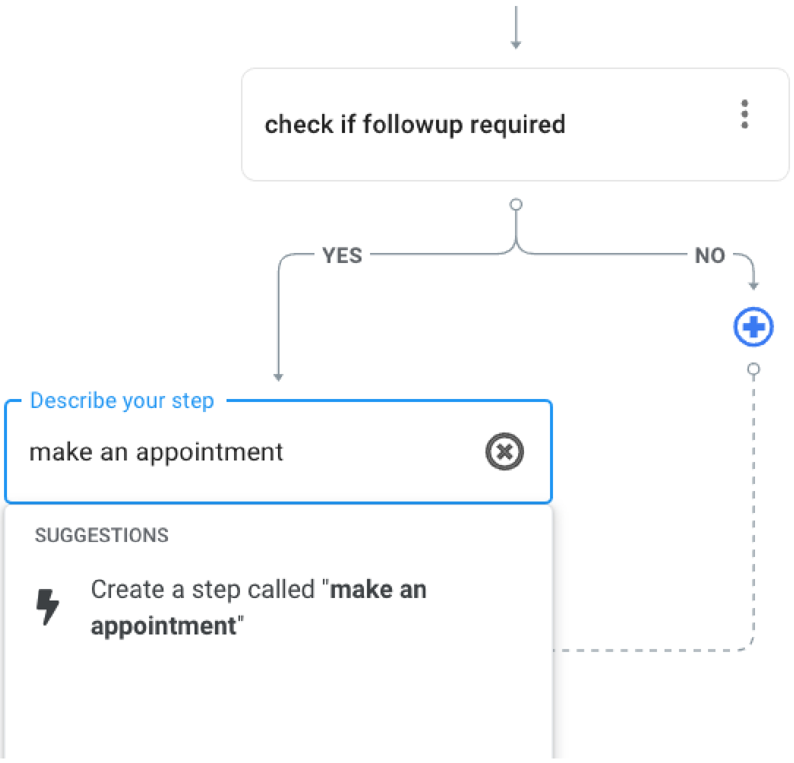
Condition To Check
= [FollowupRequested] = "Y"

Done

3. Add another step to the process.

    a. The previous step is a branch step. Let's add the next step in the left (Yes) branch.

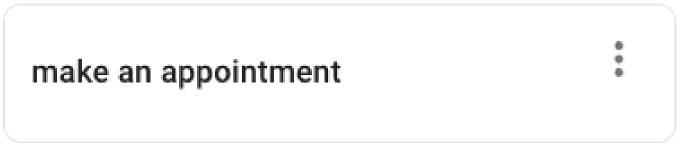
    b. Type "make an appointment" into the text box, as shown below.


check if followup required

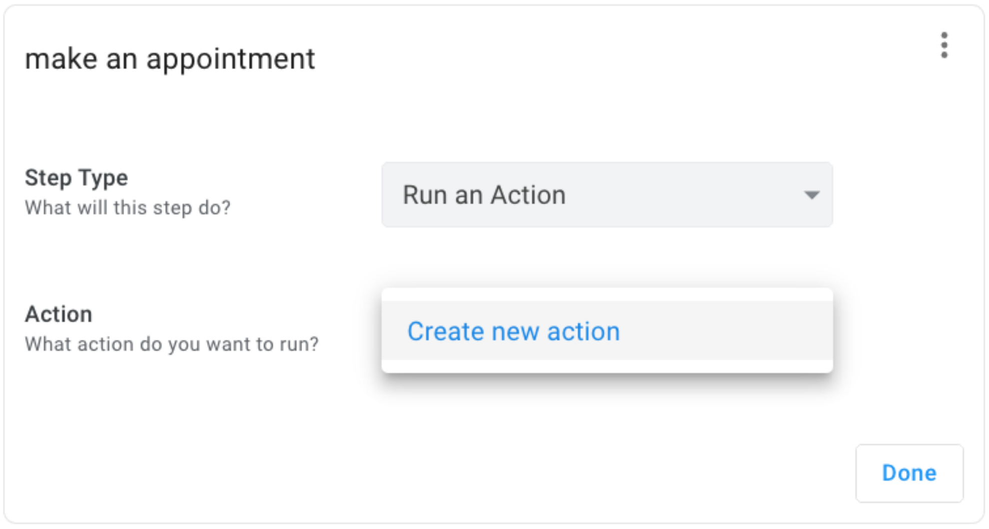
Describe your step
make an appointment

SUGGESTIONS
Create a step called "**make an appointment**"

    c. Select **Create a step called "make an appointment"** from the suggested actions to automatically create an empty step.


make an appointment

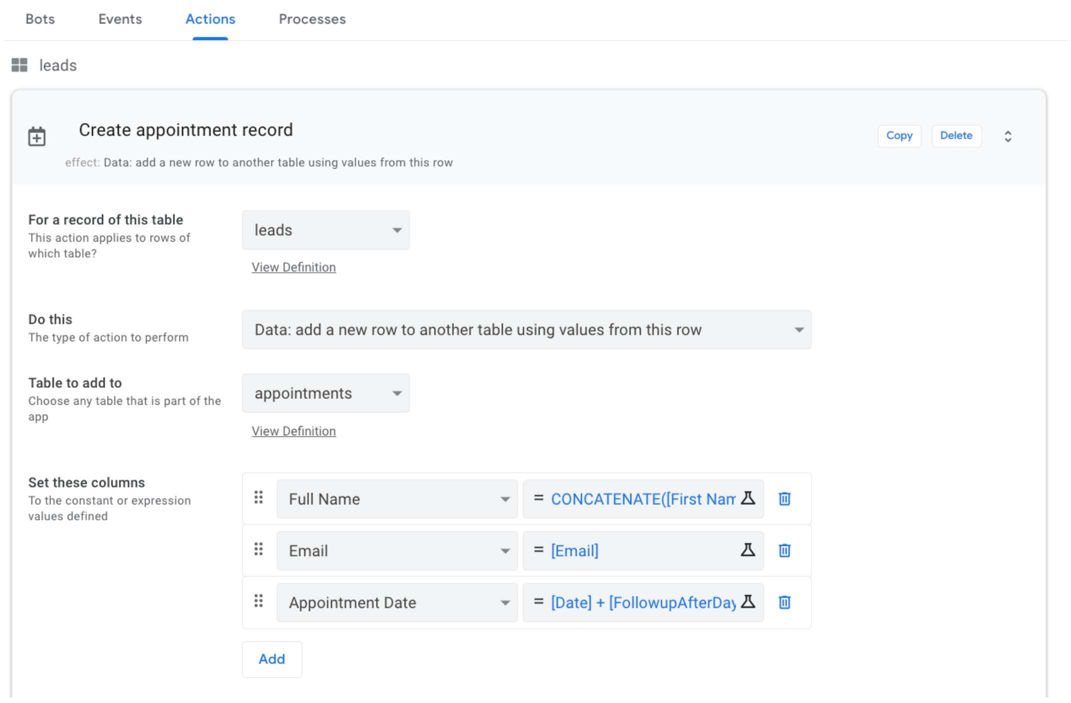
d. Click **Create new action**:


make an appointment

Step Type
What will this step do?
Run an Action

Action
What action do you want to run?
Create new action

e. Configure the action as shown below:



Create appointment record
effect: Data: add a new row to another table using values from this row

For a record of this table
This action applies to rows of which table?
leads

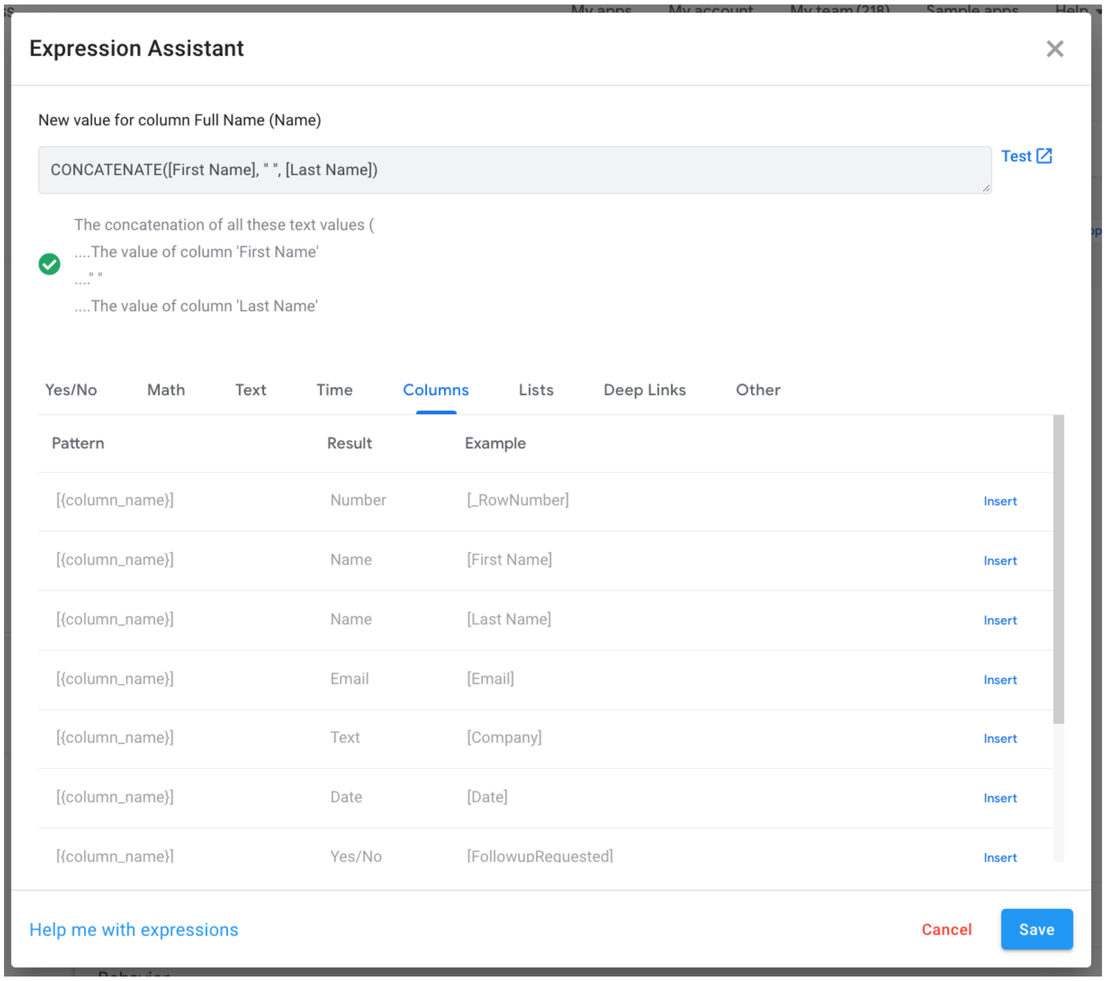
Do this
The type of action to perform
Data: add a new row to another table using values from this row

Table to add to
Choose any table that is part of the app
appointments

Set these columns
To the constant or expression values defined

Full Name	= CONCATENATE([First Nam
Email	= [Email]
Appointment Date	= [Date] + [FollowupAfterDay

f. Ensure the "Full Name" expression has the value *CONCATENATE([First Name], " ", [Last Name])*.



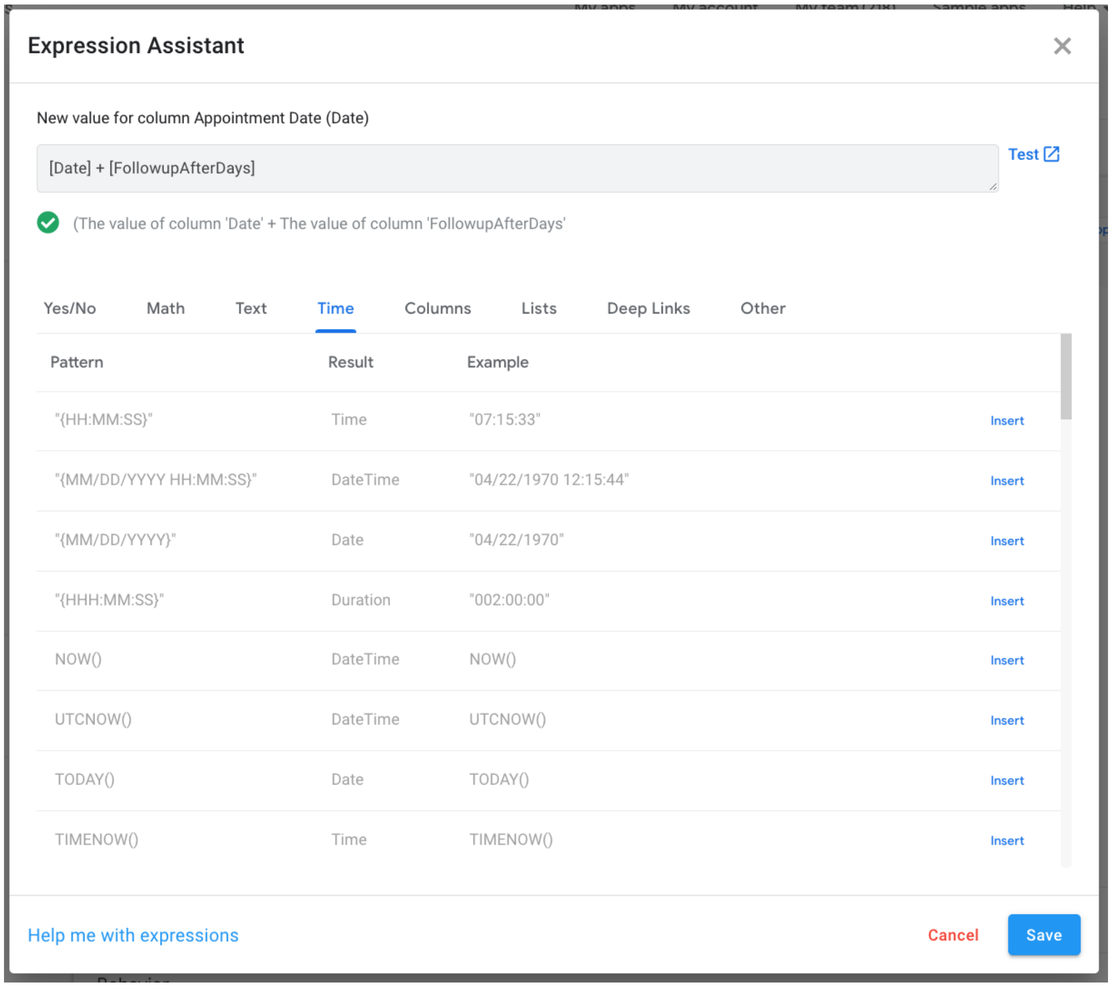
New value for column Full Name (Name)

```
CONCATENATE([First Name], " ", [Last Name])
```

The concatenation of all these text values (
....The value of column 'First Name'
....." "
....The value of column 'Last Name'

Pattern	Result	Example	
[{column_name}]	Number	[_RowNumber]	Insert
[{column_name}]	Name	[First Name]	Insert
[{column_name}]	Name	[Last Name]	Insert
[{column_name}]	Email	[Email]	Insert
[{column_name}]	Text	[Company]	Insert
[{column_name}]	Date	[Date]	Insert
[{column_name}]	Yes/No	[FollowupRequested]	Insert

g. Ensure the "Appointment Date" expression has the value *[Date] + [FollowupAfterDays]*.


Expression Assistant

New value for column Appointment Date (Date)

[Date] + [FollowupAfterDays]

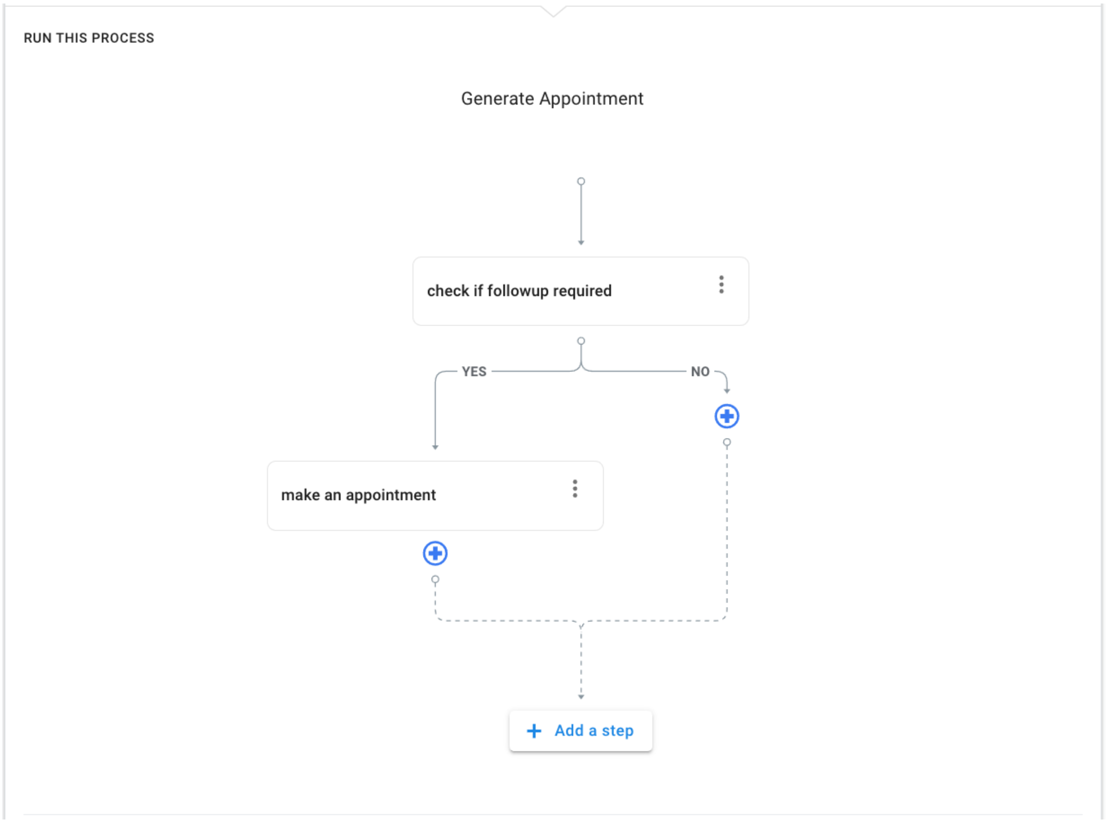
(The value of column 'Date' + The value of column 'FollowupAfterDays'

Yes/No | Math | Text | Time | Columns | Lists | Deep Links | Other

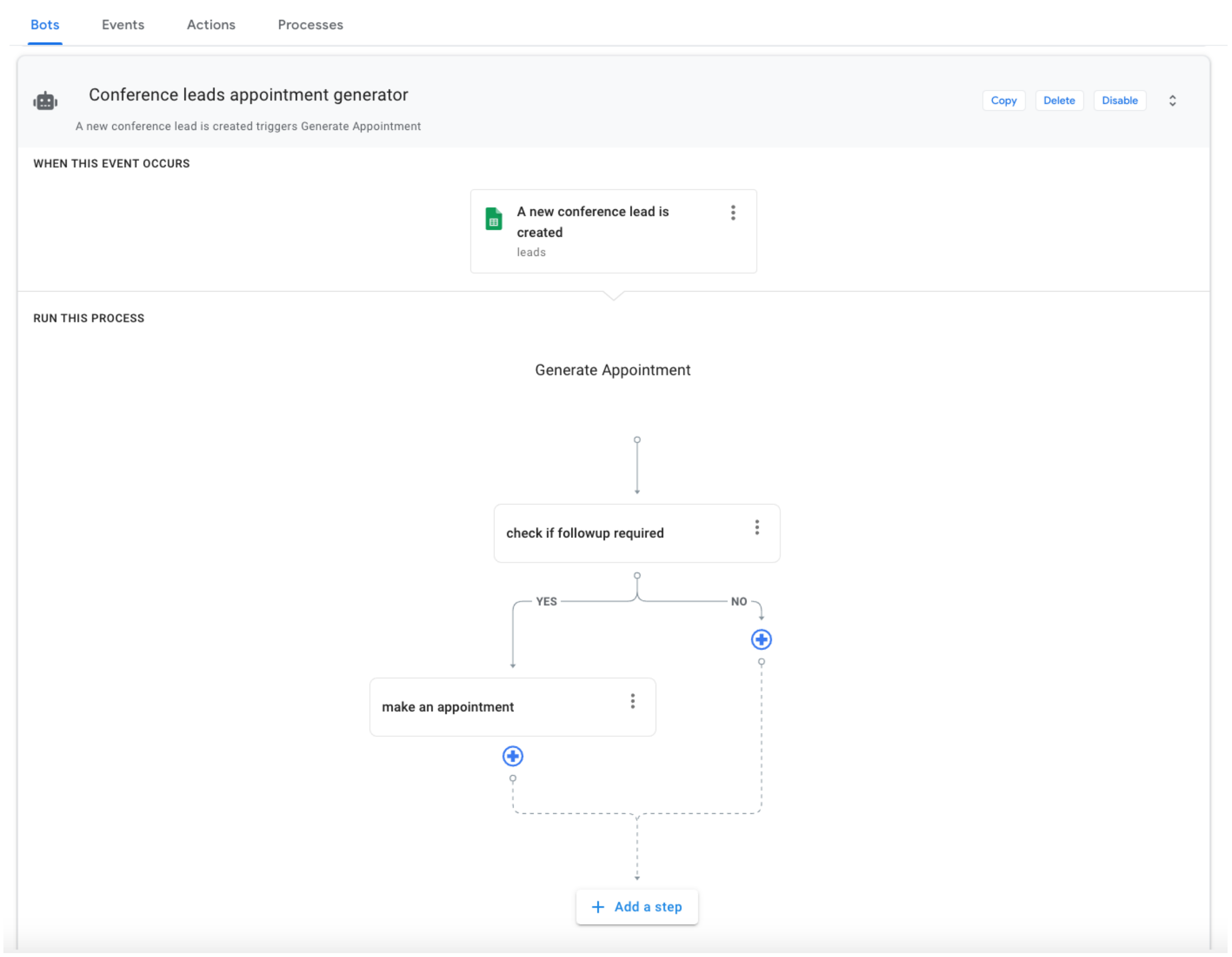
Pattern	Result	Example	
"{HH:MM:SS}"	Time	"07:15:33"	Insert
"{MM/DD/YYYY HH:MM:SS}"	DateTime	"04/22/1970 12:15:44"	Insert
"{MM/DD/YYYY}"	Date	"04/22/1970"	Insert
"{HHH:MM:SS}"	Duration	"002:00:00"	Insert
NOW()	DateTime	NOW()	Insert
UTCNOW()	DateTime	UTCNOW()	Insert
TODAY()	Date	TODAY()	Insert
TIMENOW()	Time	TIMENOW()	Insert

Help me with expressions   Cancel   Save

4. Once configured, your process should look like this:



5. Your completed bot should look like this:



Conference leads appointment generator

A new conference lead is created triggers Generate Appointment

WHEN THIS EVENT OCCURS

A new conference lead is created
leads

RUN THIS PROCESS

Generate Appointment

check if followup required

make an appointment

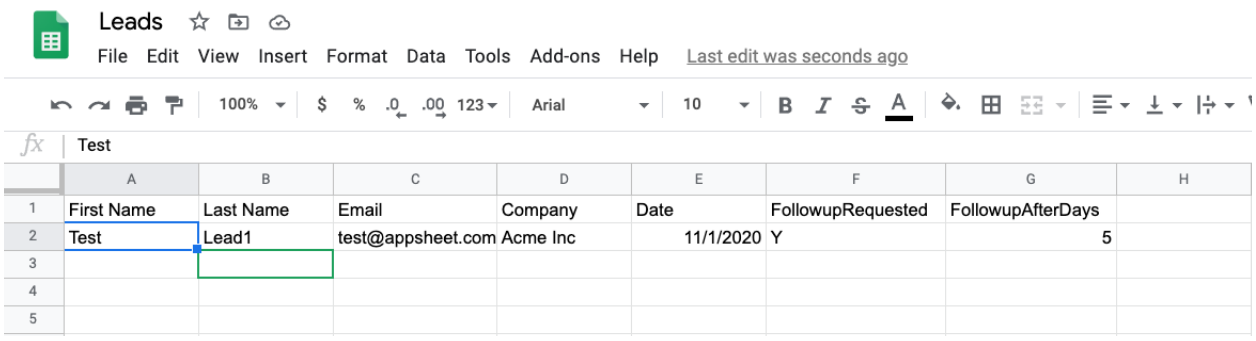
Testing your automation

Follow these steps to test your automation:

1. Go to your “Leads” sheet and enter an entire leads record into a new row.

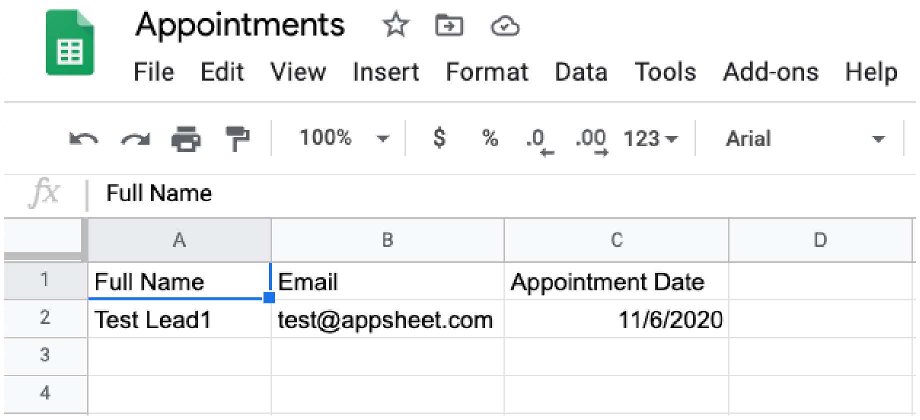
   Tip: Create a temporary sheet and add your row of data there, copy it and then paste it into the “Leads” Sheet)

   Note: *Automation only supports adding a full record at this point. Do not add a partial lead record.*


Leads
File Edit View Insert Format Data Tools Add-ons Help Last edit was seconds ago
100% $ % .0 .00 123 Arial 10
fx Test

	A	B	C	D	E	F	G	H
1	First Name	Last Name	Email	Company	Date	FollowupRequested	FollowupAfterDays	
2	Test	Lead1	test@appsheet.com	Acme Inc	11/1/2020	Y	5	
3								
4								
5								

2. Within a few seconds, an appointment record is added to your Appointments Sheet.


Appointments
File Edit View Insert Format Data Tools Add-ons Help
100% $ % .0 .00 123 Arial
fx Full Name

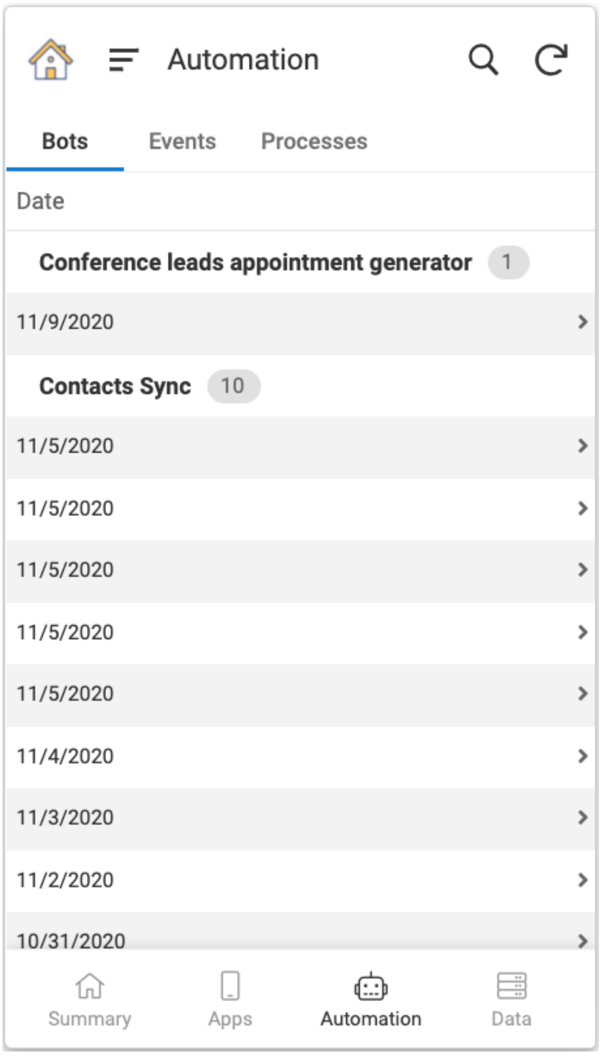
	A	B	C	D
1	Full Name	Email	Appointment Date	
2	Test Lead1	test@appsheet.com	11/6/2020	
3				
4				

Based upon the process created in previous steps, a new appointment record is only created when the value of the “FollowupRequested” column is “Y” in the Leads Sheet. The “Full Name” field in the Appointment Sheet is a concatenation of the “First Name” and “Last Name” values from the Leads Sheet. The “Appointment Date” in the Appointments Sheet should reflect the date value calculated from the Expression Assistant formula.

Automation Monitoring

As noted earlier, basic monitoring views have already been added to the Sample Automation App. As you test your bot by adding records in Salesforce, the monitoring views render bot executions data.

To view the bot executions data, click on the “Summary” and “Automations” views in your app. *Your view may look different from the screen shots presented below.*



Automation

Bots | Events | Processes

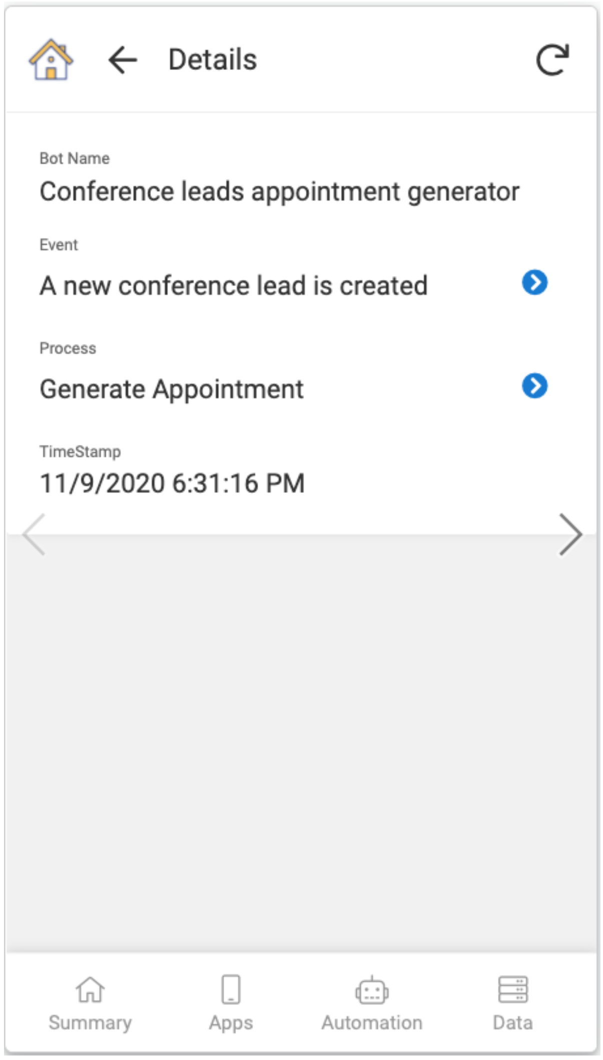
Date

Conference leads appointment generator 1

11/9/2020

Contacts Sync 10

11/5/2020
11/5/2020
11/5/2020
11/5/2020
11/5/2020
11/4/2020
11/3/2020
11/2/2020
10/31/2020



Details

Bot Name
Conference leads appointment generator

Event
A new conference lead is created

Process
Generate Appointment

TimeStamp
11/9/2020 6:31:16 PM

Appendix

Configuring the Sample Automation app

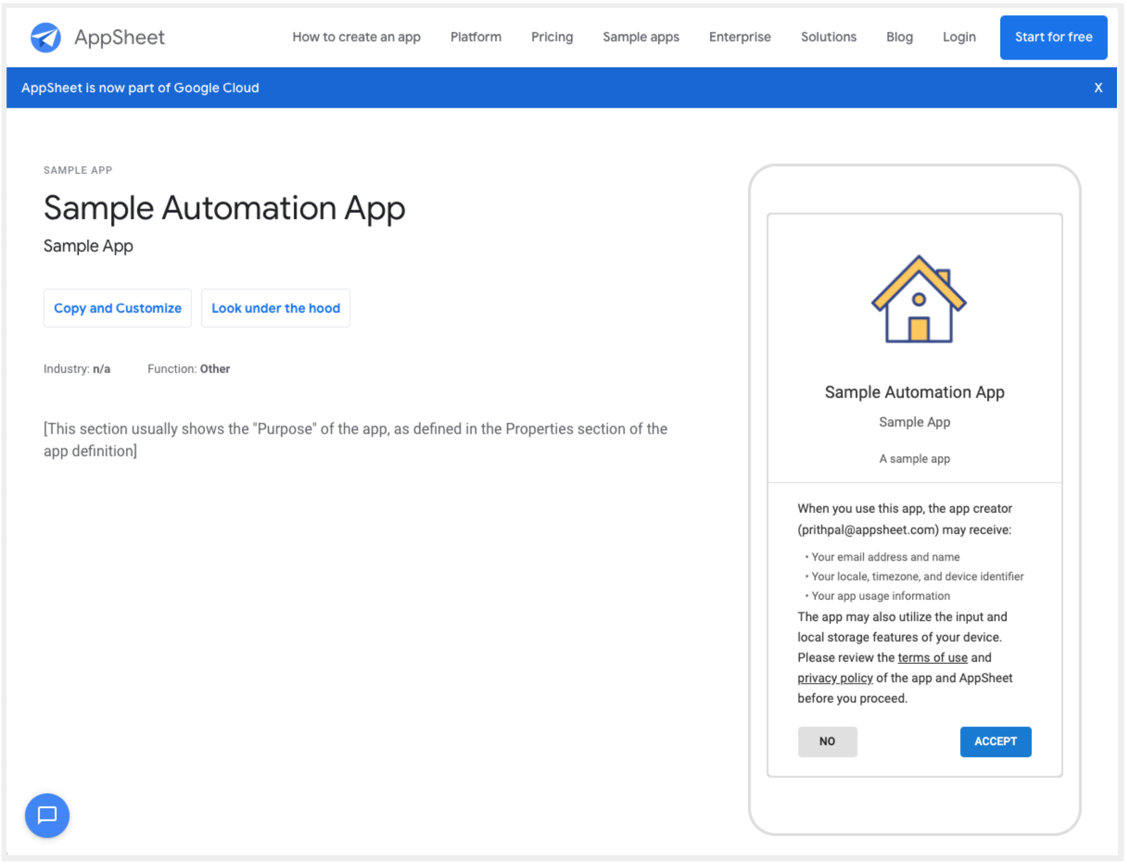
AppSheet Automation provides monitoring of automations through a Monitoring App. The app can be used by app creators and operators alike to monitor the status of bots, events and processes. The Monitoring App has the necessary views and appropriate configuration to the native data sources (contained inside your app) to render monitoring data.

Note: *In the future, the integrated monitoring capabilities of the sample app will be migrated to a standalone app.*

For early access feedback and testing, we recommend that you make a copy of the sample automation app and add your test automations to it. Do not make any changes to the pre-existing view/data sources that have been configured for providing monitoring capabilities.

To copy and configure the Sample Automation App with monitoring:

1. Visit the Sample Automation App page and click **Copy and Customize**.



Sample Automation App

Sample App

Industry: n/a Function: Other

[This section usually shows the "Purpose" of the app, as defined in the Properties section of the app definition]

Sample Automation App

Sample App

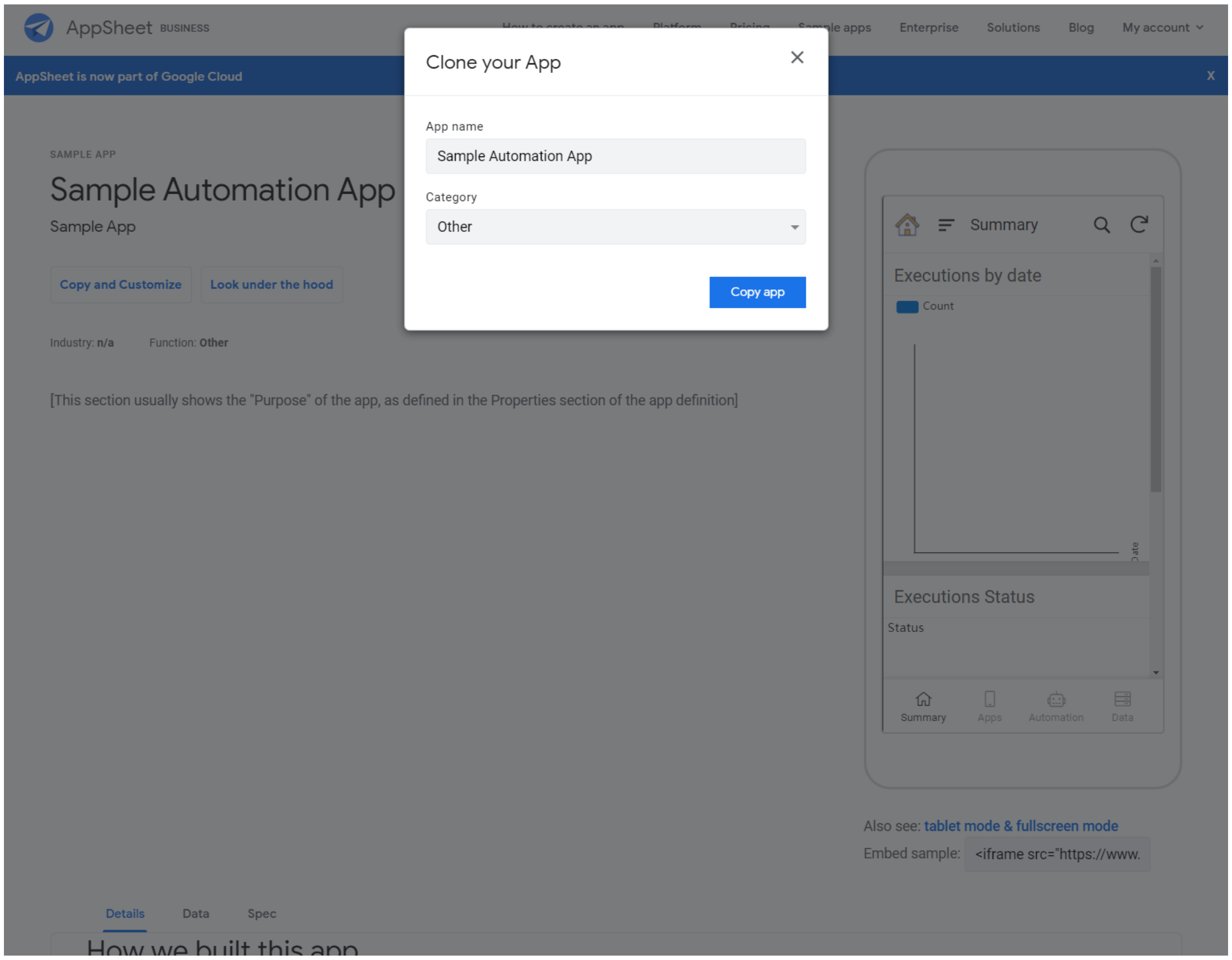
A sample app

When you use this app, the app creator (prithpal@appsheet.com) may receive:

- Your email address and name
- Your locale, timezone, and device identifier
- Your app usage information

The app may also utilize the input and local storage features of your device. Please review the terms of use and privacy policy of the app and AppSheet before you proceed.

2. When the “Clone your App” dialog appears, click **Copy app**.



Clone your App

App name: Sample Automation App

Category: Other

Copy app

[This section usually shows the "Purpose" of the app, as defined in the Properties section of the app definition]

3. The application will be copied into your account, and then opened in the AppSheet Editor UI.
4. Deploy the Sample Automation App from your account..

Your sample app is now ready to test with the addition of automations.

Configuring Contact entity from Salesforce datasource

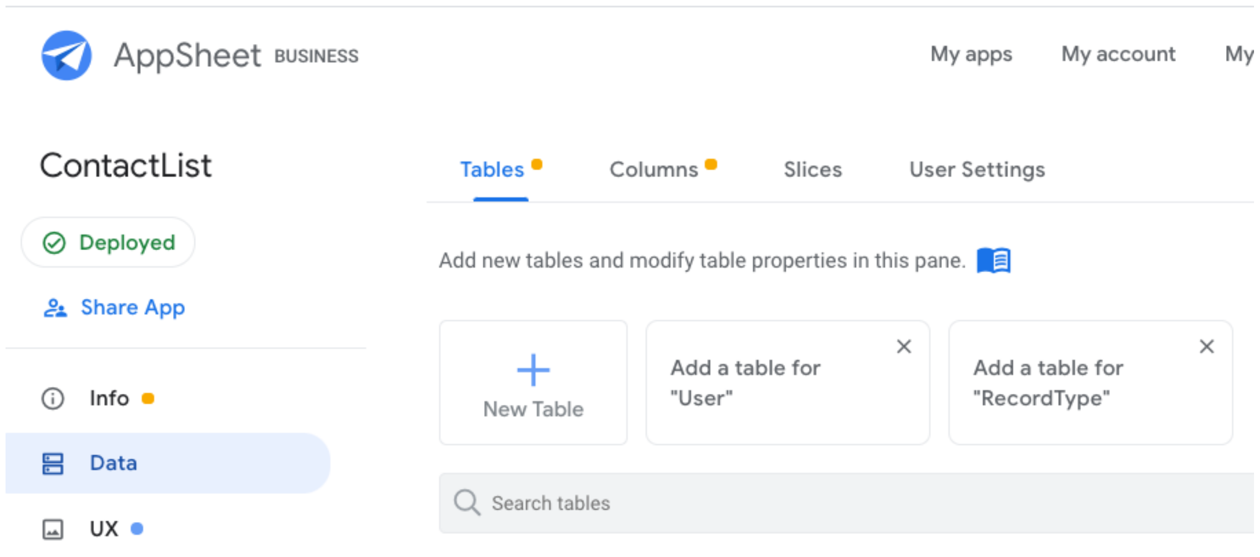
Please refer to this to ensure that you have correctly configured a “Contact” entity in Salesforce.

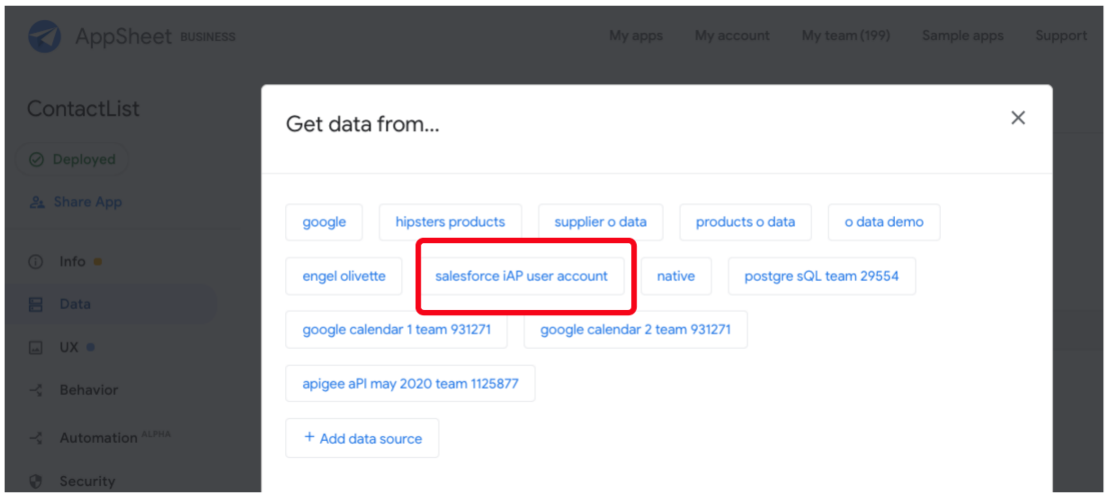
Before you begin:

- Confirm that you have access to a Salesforce account.
- Confirm that the appropriate AppSheet package has been deployed in Salesforce. (Refer to this help article for further detail).

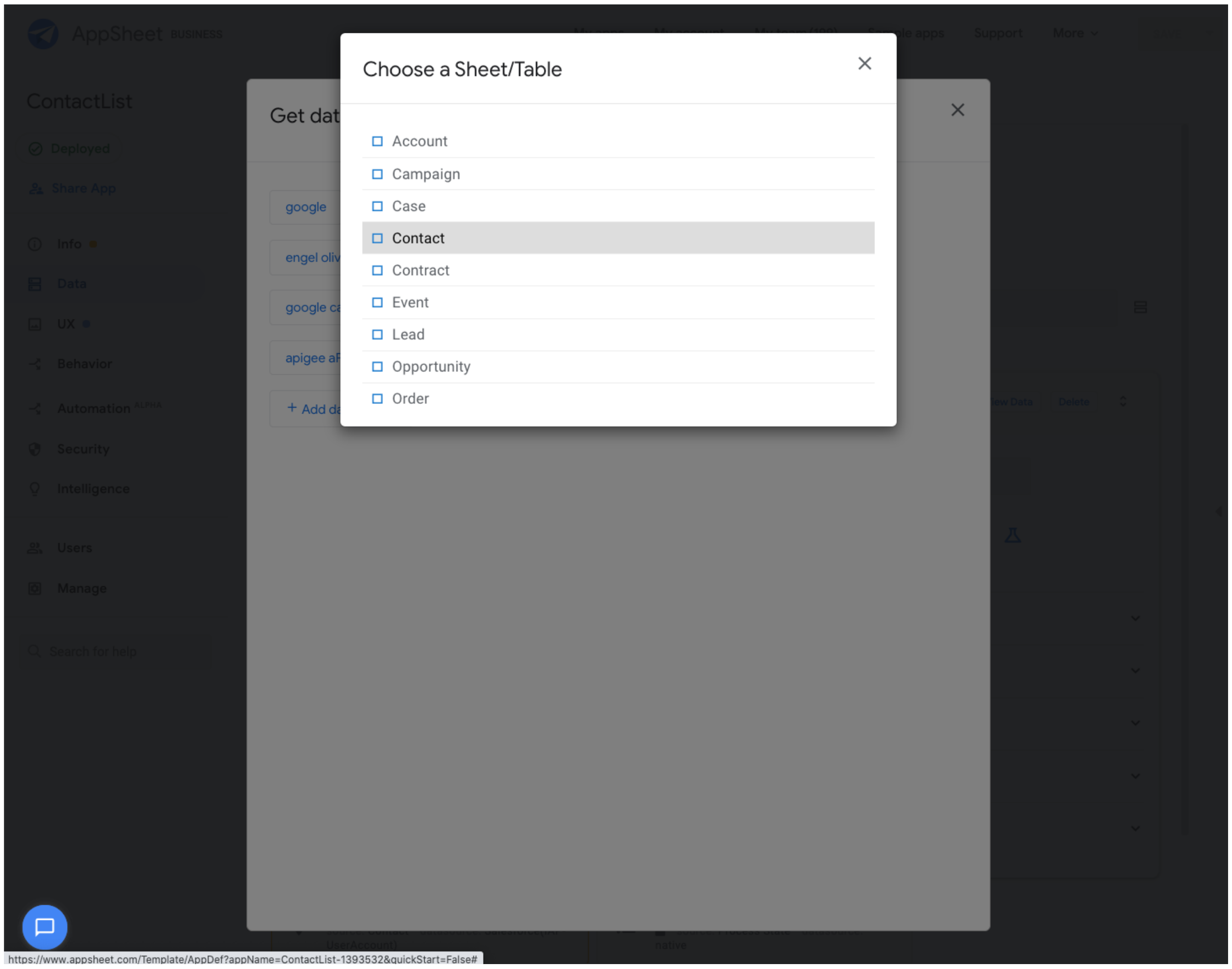
To configure a “Contact” entity in AppSheet using your Salesforce account data:

1. From the AppSheet UI, click **Data** in the left navigation bar.
2. Click **New Table**.
3. Select your configured Salesforce Data Source, as shown in the image below.





4. Select “Contact” as the table for your entity.



5. For ‘Are updates allowed?’ below, change the selection to **Read-Only**.

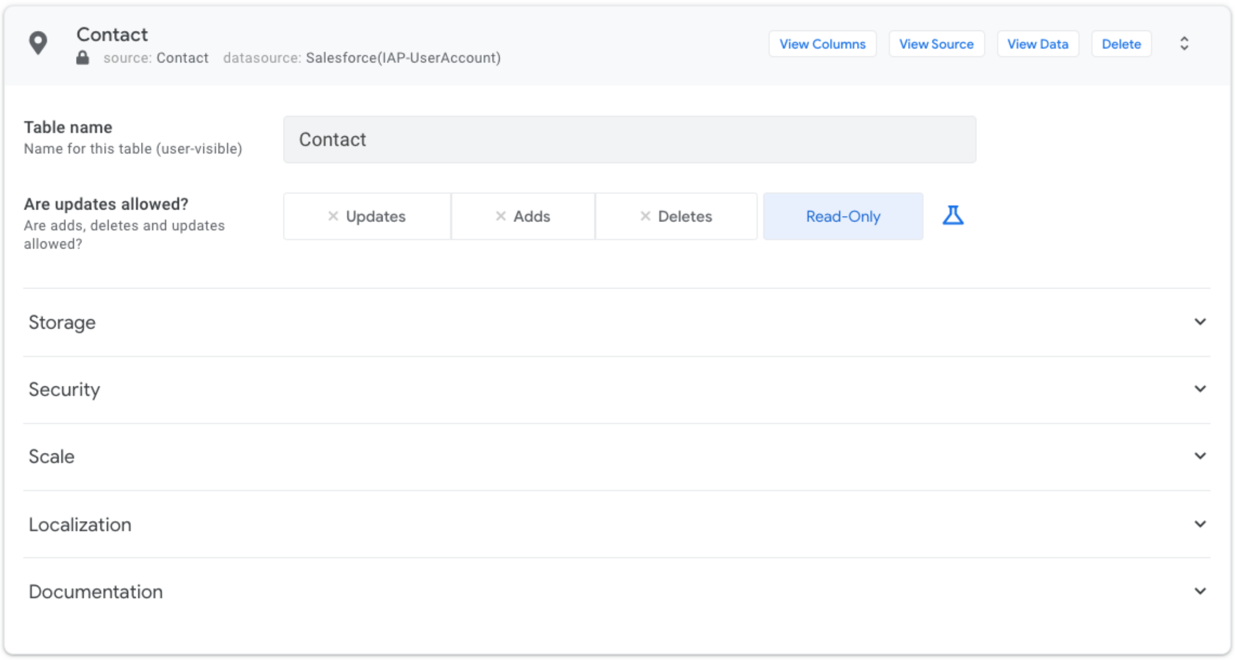
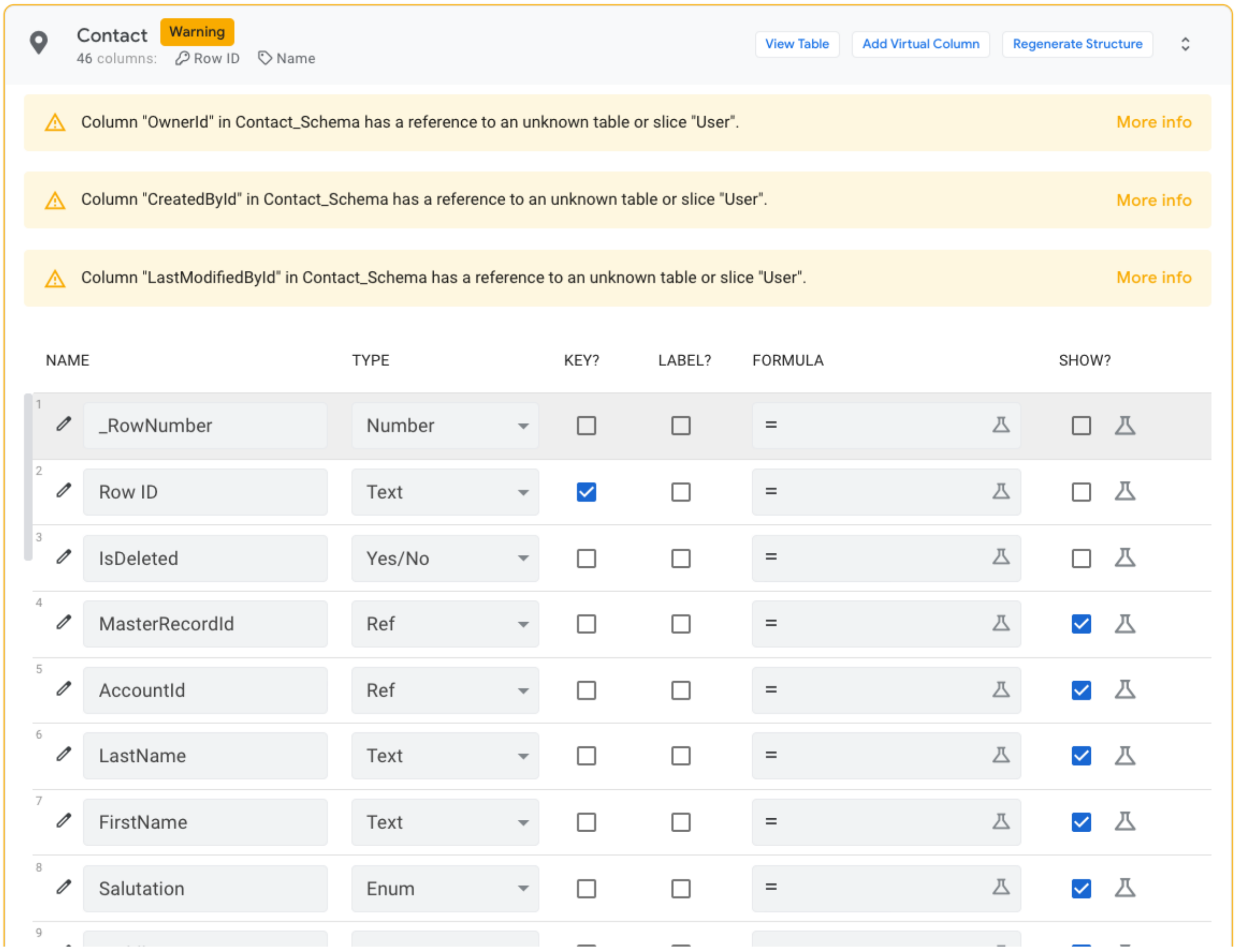

Contact
source: Contact datasource: Salesforce(IAP-UserAccount)

Table name
Name for this table (user-visible)
Contact

Are updates allowed?
Are adds, deletes and updates allowed?
× Updates | × Adds | × Deletes | Read-Only

6. Click **View Table**, as shown below:


Contact Warning
46 columns: Row ID Name
View Table Add Virtual Column Regenerate Structure

Column "OwnerId" in Contact_Schema has a reference to an unknown table or slice "User". More info

Column "CreatedById" in Contact_Schema has a reference to an unknown table or slice "User". More info

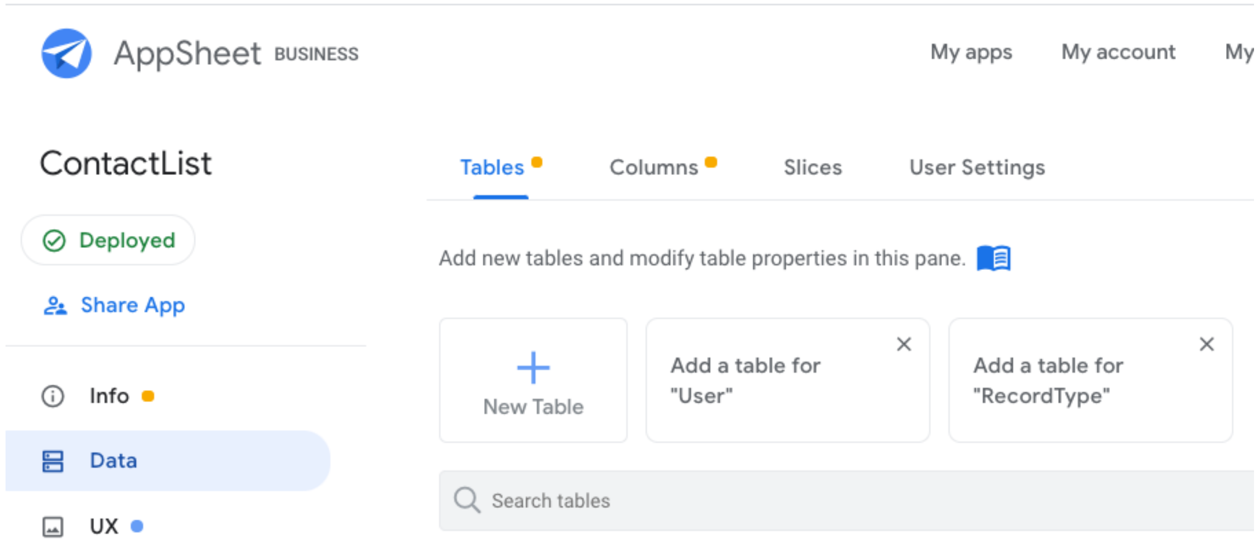
Column "LastModifiedById" in Contact_Schema has a reference to an unknown table or slice "User". More info

	NAME	TYPE	KEY?	LABEL?	FORMULA	SHOW?
1	_RowNumber	Number	☐	☐	=	☐
2	Row ID	Text	✓	☐	=	☐
3	IsDeleted	Yes/No	☐	☐	=	☐
4	MasterRecordId	Ref	☐	☐	=	✓
5	AccountId	Ref	☐	☐	=	✓
6	LastName	Text	☐	☐	=	✓
7	FirstName	Text	☐	☐	=	✓
8	Salutation	Enum	☐	☐	=	✓
9						

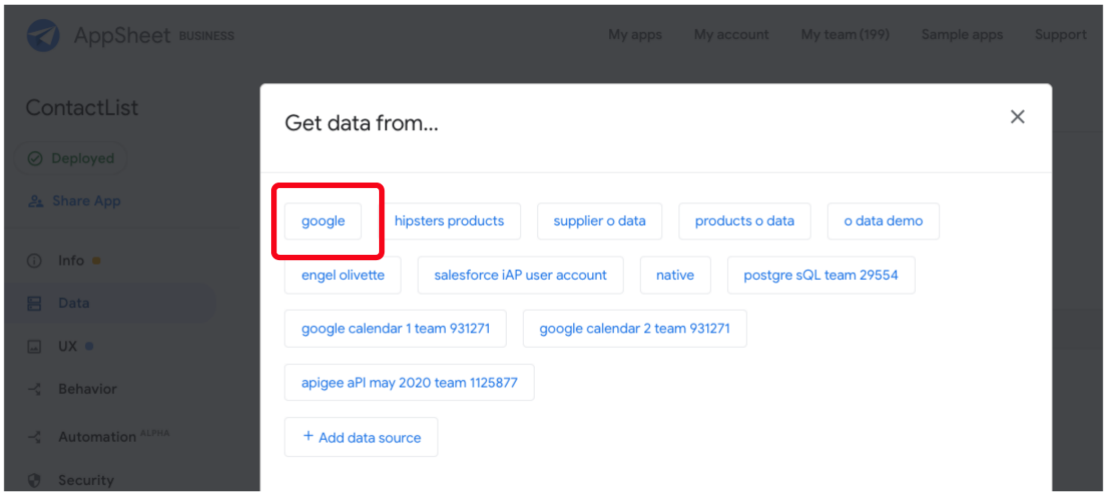
Configuring Company Contacts entity from Google Sheets datasource

To configure your “Company Contacts” entity from Google Sheets:

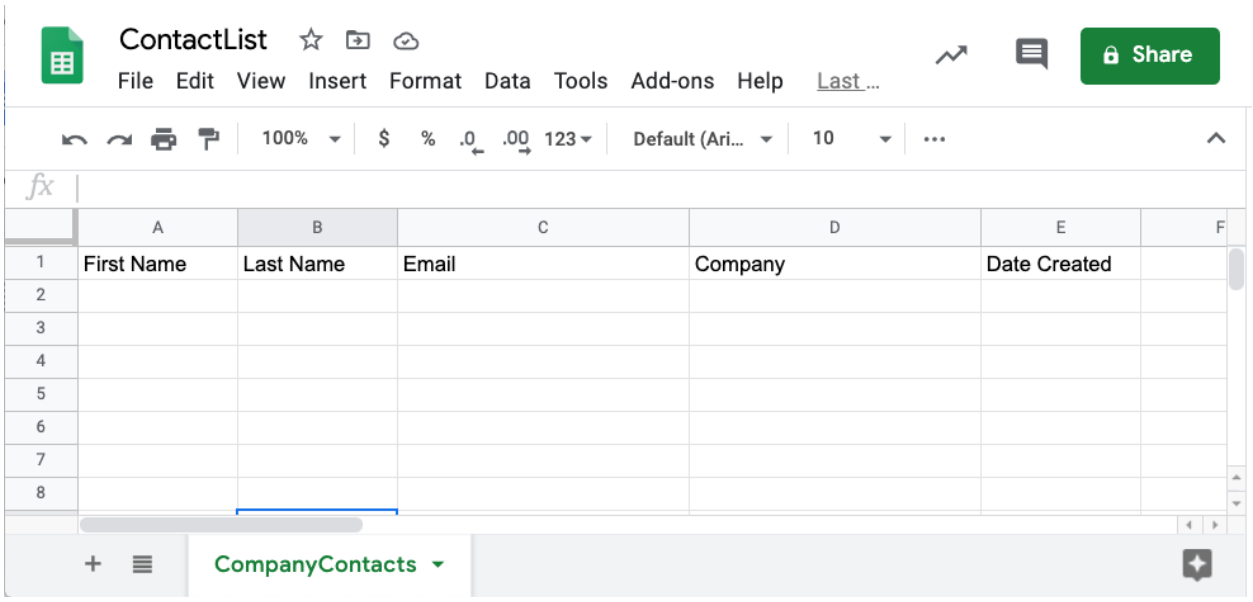
1. From the AppSheet UI, click **Data** in the left navigation bar.
2. Click **New Table**.



3. Select your configured Google Data Source.

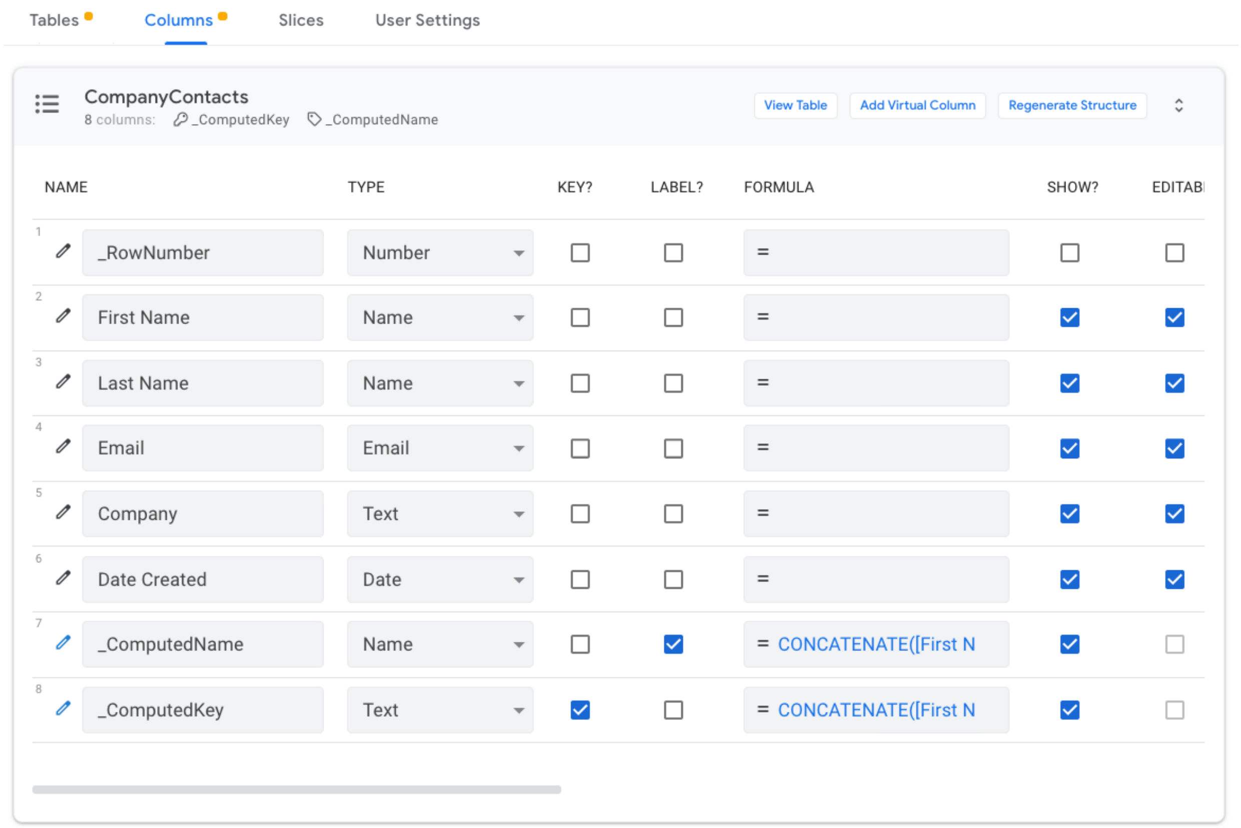


4. Use the file browser to select the appropriate **ContactList** Sheet as the data source.


ContactList
File Edit View Insert Format Data Tools Add-ons Help Last ...
Share
100% $ % .0 .00 123 Default (Ari... 10
CompanyContacts

	A	B	C	D	E	F
1	First Name	Last Name	Email	Company	Date Created	
2						
3						
4						
5						
6						
7						
8						

5. Select **Company Contacts** as the table for your entity.

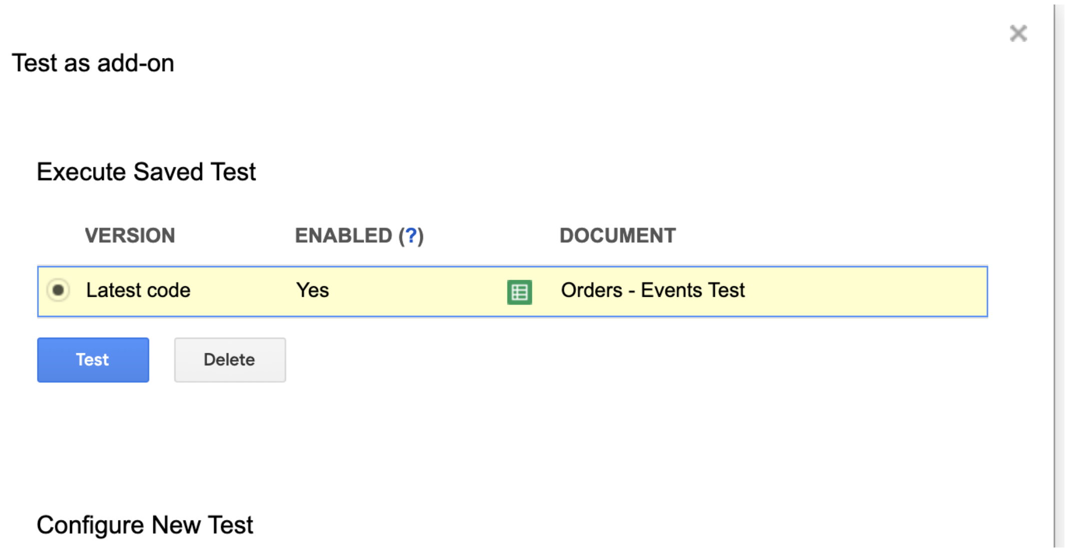


	NAME	TYPE	KEY?	LABEL?	FORMULA	SHOW?	EDITABI
1	_RowNumber	Number	☐	☐	=	☐	☐
2	First Name	Name	☐	☐	=	✓	✓
3	Last Name	Name	☐	☐	=	✓	✓
4	Email	Email	☐	☐	=	✓	✓
5	Company	Text	☐	☐	=	✓	✓
6	Date Created	Date	☐	☐	=	✓	✓
7	_ComputedName	Name	☐	✓	= CONCATENATE([First N	✓	☐
8	_ComputedKey	Text	✓	☐	= CONCATENATE([First N	✓	☐

Configuring the AppSheet Events add-on for Google Sheets

Follow the steps below to configure the AppSheet Events add-on for Google Sheets:

1. Open the Sheet that you are using as your primary data source.
2. From within the Sheets app, click on **Tools > Script Editor** in the top navigation.
3. Name the project.
    - Click on the *Untitled Project* placeholder in the upper left corner of the Script Editor. For the purposes of this tutorial, name it "Appsheet Events."
    - Click **OK**.
    - **Note:** I*t may take a moment for the operation to complete. Before moving on to the next step, you should see the 'Saving project 'Appsheet Events'...' message at the top of the screen disappear.*
4. Copy and paste the AppSheet Events add-on code into the Script Editor's code.gs file, replacing any placeholder code.
5. From the Script Editor's top navigation, click **Run** -> **Test as add-on.**
6. In the "Test as add-on" dialog window that appears, use the drop-down selector under **INSTALLATION CONFIG** to select **Installed And Enabled.**
7. Click on **Select Doc.**
8. In the "Select an item" dialog window that appears, navigate to the **Spreadsheets** tab.
    - Select the Sheet you want to use as your data source.
    - Click **Select**.
9. Click **Save**.
10. Click the radio button in the "Execute saved test" section that contains your Sheet selection, as shown in the figure below:


Test as add-on

Execute Saved Test

	VERSION	ENABLED (?)	DOCUMENT
●	Latest code	Yes	Orders - Events Test

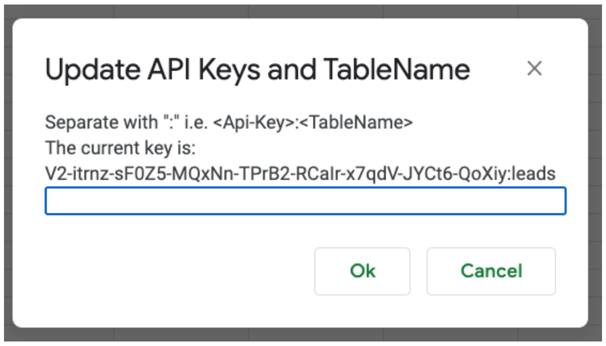
Configure New Test

11. Click on **Test.**
12. A new tab will be opened with the same doc. **Close this tab**.

Follow the steps below to configure the permissions and authorization required for your Sheet to communicate with AppSheet:

1. Return to the Sheet you opened in Step 1 of the previous section and **refresh** the page.
2. Configure permissions for your Sheet:
    a. From the top navigation, select **Add-ons > Appsheet Events > Update App Keys.** If you chose another name for your project in the Script Editor, select it from the **Add-ons** menu and then navigate to **Update App Keys**.
    b. An "Authorization Required" dialog window will appear. Click **Continue.**
    c. Sign-in using the email address associated with your Sheet.
    d. Click **Allow** to give the appropriate permissions to your project.
3. Obtain an API Key:
    a. Log in and open your app in AppSheet UI.
    b. Click **Manage** in the side navigation.
    c. Select the **Integrations** tab.
    d. Click on the **IN: from cloud services to your app** to enable your Sheet to communicate with your app.
        i. Click the **Enable** toggle to enable the integration, if it is not enabled.
        ii. In the "Application Access Keys" section, check to ensure the access key is not expired.
        Note: *If the key is valid, you should see a message like this "This access key will stop working on <YYYY-MM-DD>" containing a future date.*
        iii. If the key is expired, click on **Create Application Access Key** to generate a new key.
        iv. Click **Show Access Key** to display the valid key string. For example, "V2-itrnz-sF0Z5-MQxNn-TPrB2-RCalr-x7qdV-JYCt6-QoXiy"
        v. Copy the access key into a notepad. The key string will be used in a later step.
4. Confirm the appropriate app table name:
    a. From within your app, click **Data** in the side navigation.
    b. Click on the table you want to use for events. The **Table name** of the table should correspond to the Sheet you are using. For example, if your Sheet is titled "Appsheet Events," select the "Appsheet Events" table.
    c. Copy the table name into a notepad. The table name will be used in a later step.
5. Update the App Key in Sheets.
    a. Return to your original Sheet data source from Step 1.
    b. Navigate to **Add-on > AppSheet Events > Update App Keys**.

c. Using the values from Steps 3 and 3, enter the new App Key value in the dialog window, as shown in the figure below. The format of the key is <appkey>:<tablename>. The values are separated by a colon. For example: V2-itrnz-sF0Z5-MQxNn-TPrB2-RCalr-x7qdV-JYCt6-QoXiy:leads


Update API Keys and TableName

Separate with ":" i.e. <Api-Key>:<TableName>
The current key is:
V2-itrnz-sF0Z5-MQxNn-TPrB2-RCalr-x7qdV-JYCt6-QoXiy:leads

d. Click **Ok**.

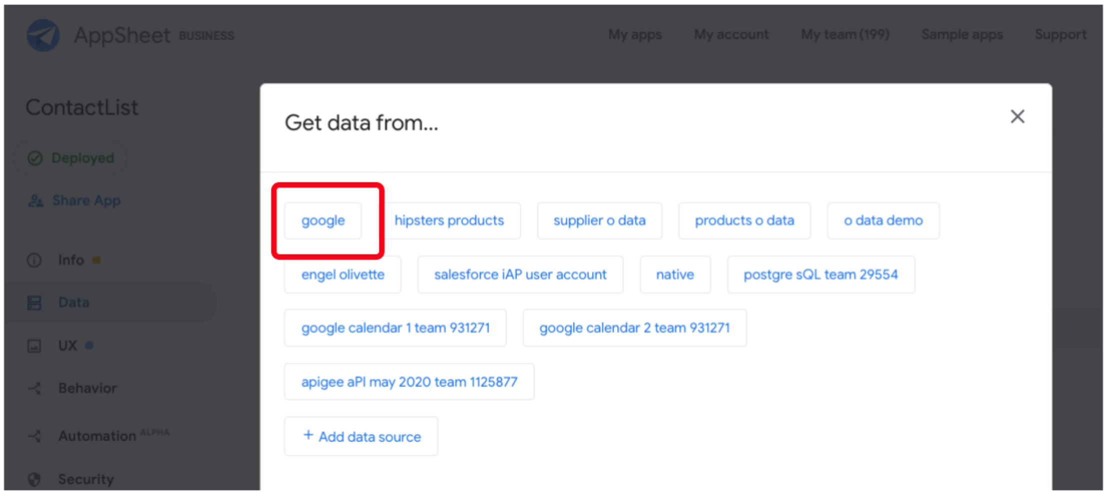
Follow the steps below to manually create a trigger:

1. Open the Script Editor from the Sheet: **Tools > Script editor.**
2. In the Script Editor, click the trigger icon in the top navigation bar:
3. Click **Add Trigger** (located at the bottom right of the screen)
4. Under "Select event type, " use the dropdown to select **On edit.**
5. Click **Save**.

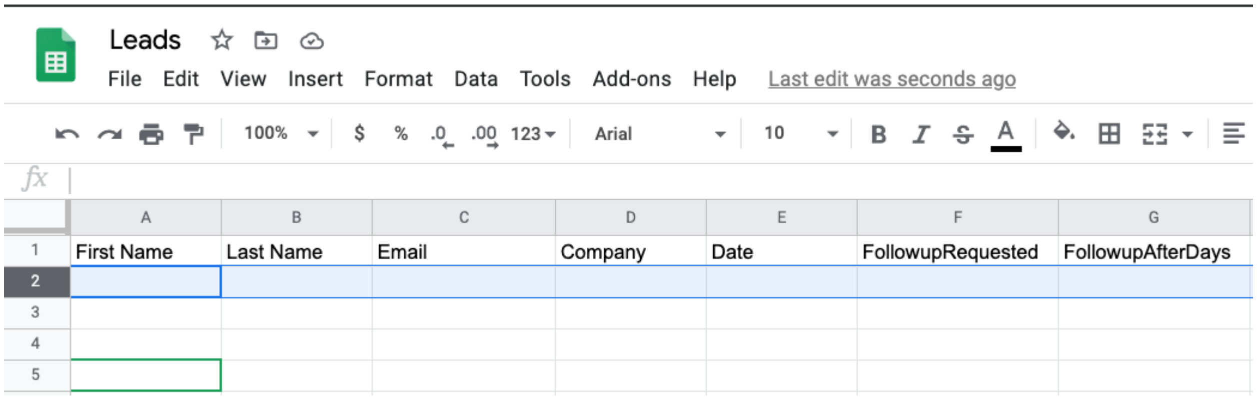
Configuring Leads entity from Google Sheets datasource

To configure your "Leads" entity from Google Sheets:

1. From the AppSheet UI, click **Data** in the left navigation bar.
2. Click **New Table**.
3. Select your configured Google Data Source.

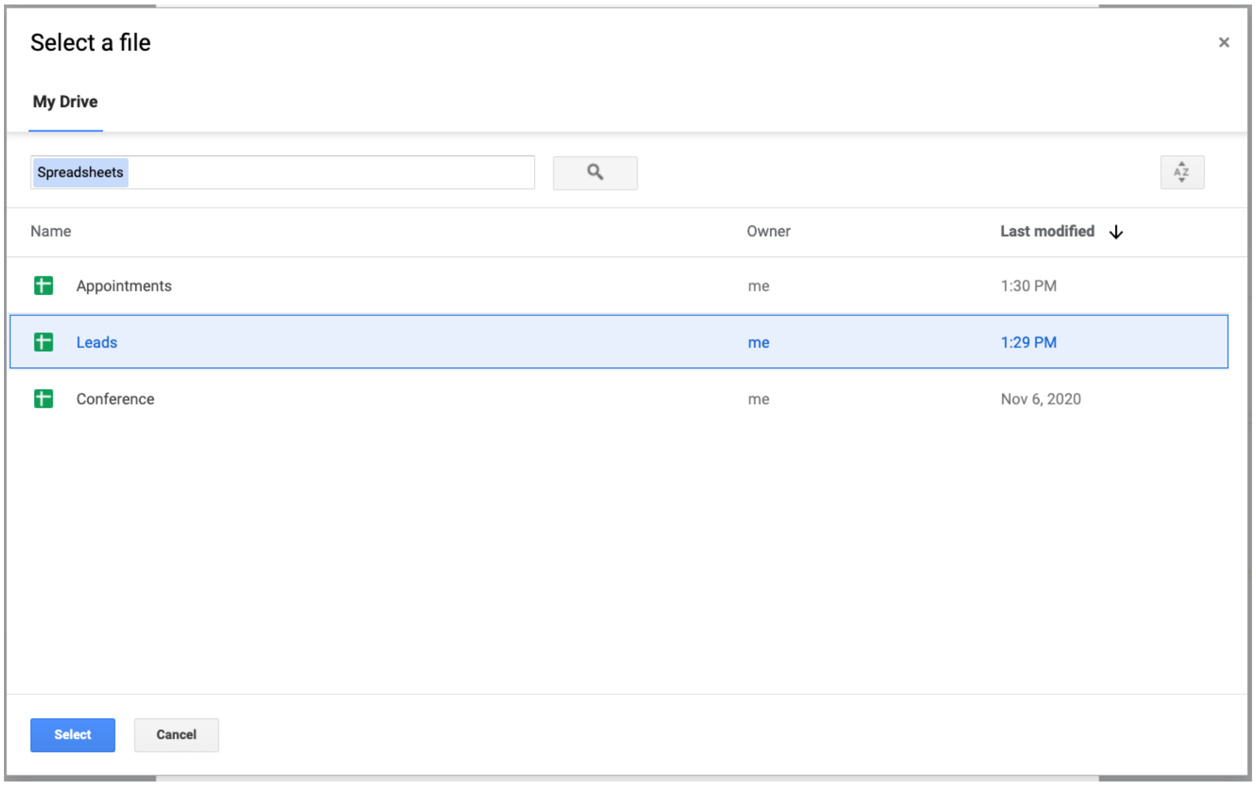


4. Create a new Google Sheet named "Leads." Use the column names shown below and name the sheet (tab) "leads". The Sheet should be in the Google Drive accessible to your AppSheet account.


Leads
File Edit View Insert Format Data Tools Add-ons Help
Last edit was seconds ago
100%
Arial
10

	A	B	C	D	E	F	G
1	First Name	Last Name	Email	Company	Date	FollowupRequested	FollowupAfterDays
2							
3							
4							
5							

5. Use the file browser to select the appropriate **Leads** Sheet as the data source.

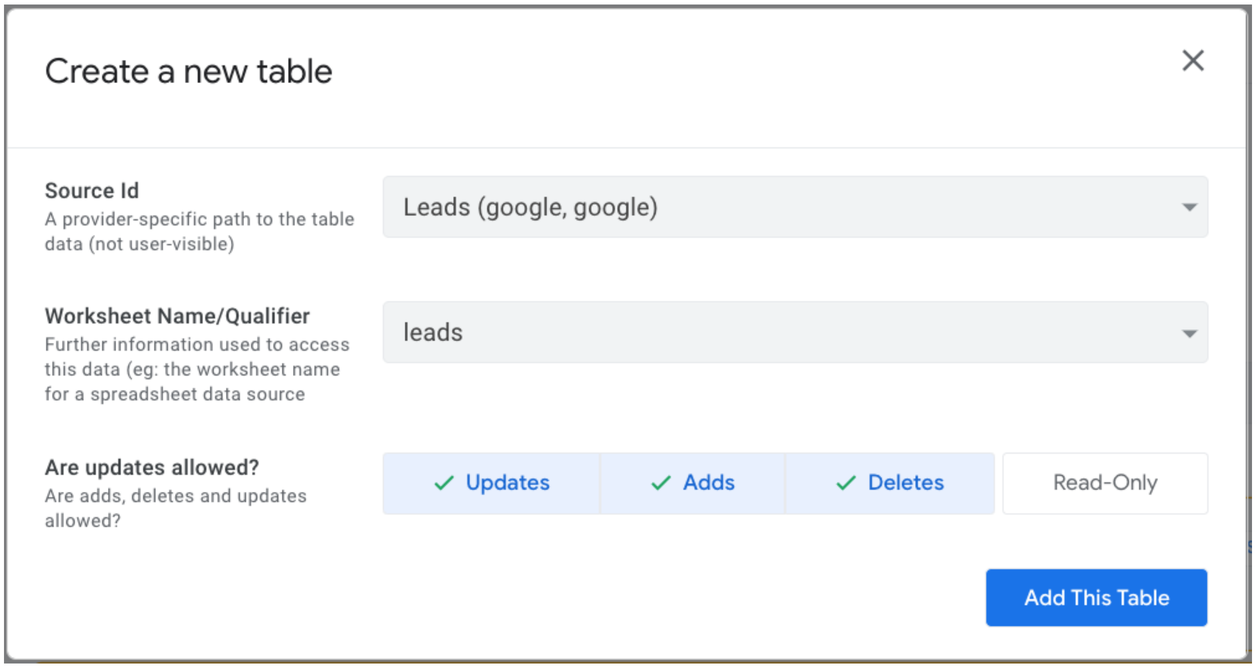

Select a file

My Drive

Spreadsheets

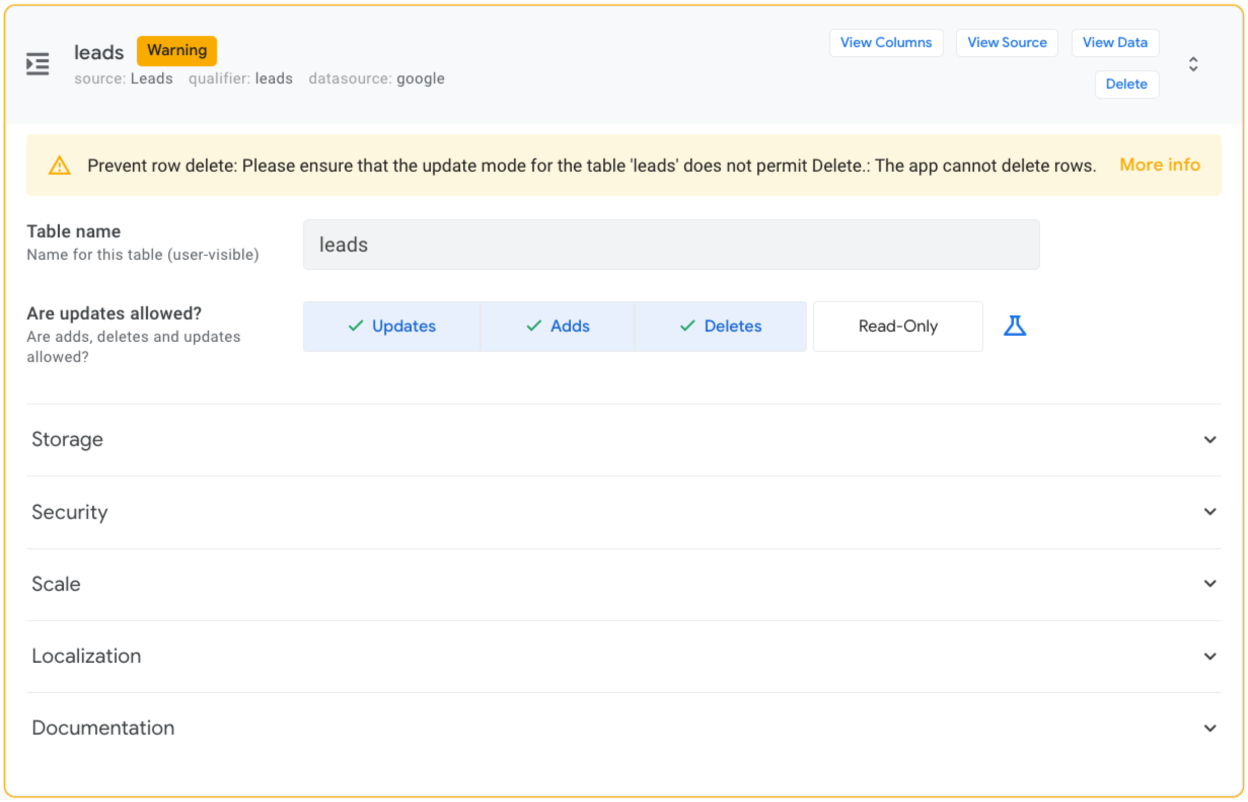
Name	Owner	Last modified
Appointments	me	1:30 PM
Leads	me	1:29 PM
Conference	me	Nov 6, 2020

6. Select **leads** as the table for your entity.


Create a new table

Source Id A provider-specific path to the table data (not user-visible)	Leads (google, google)
Worksheet Name/Qualifier Further information used to access this data (eg: the worksheet name for a spreadsheet data source	leads
Are updates allowed? Are adds, deletes and updates allowed?	✓ Updates ✓ Adds ✓ Deletes Read-Only

7. Click **Add This Table**.



source: Leads qualifier: leads datasource: google

Prevent row delete: Please ensure that the update mode for the table 'leads' does not permit Delete.: The app cannot delete rows. More info

Table name
Name for this table (user-visible)
leads

Are updates allowed?
Are adds, deletes and updates allowed?
✓ Updates ✓ Adds ✓ Deletes Read-Only

Storage

Security

Scale

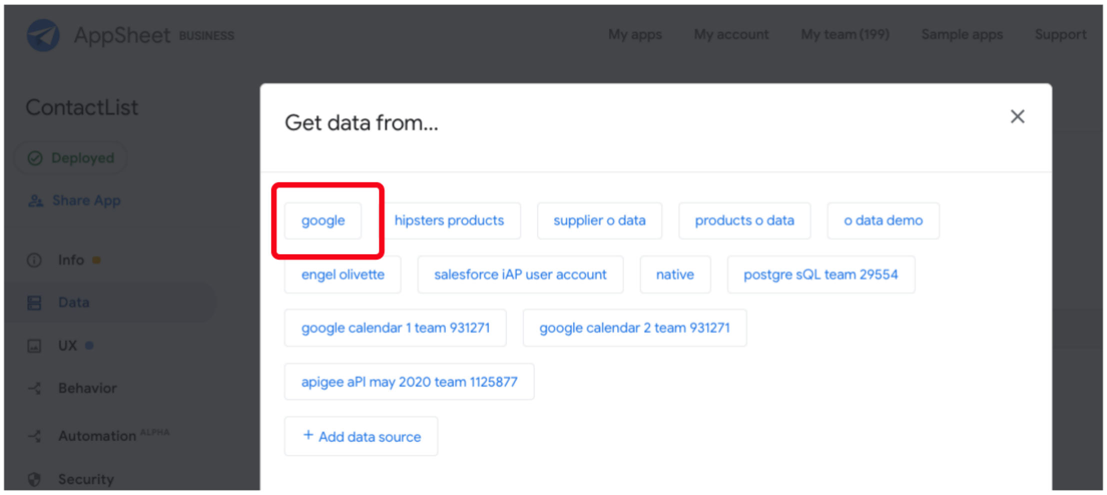
Localization

Documentation

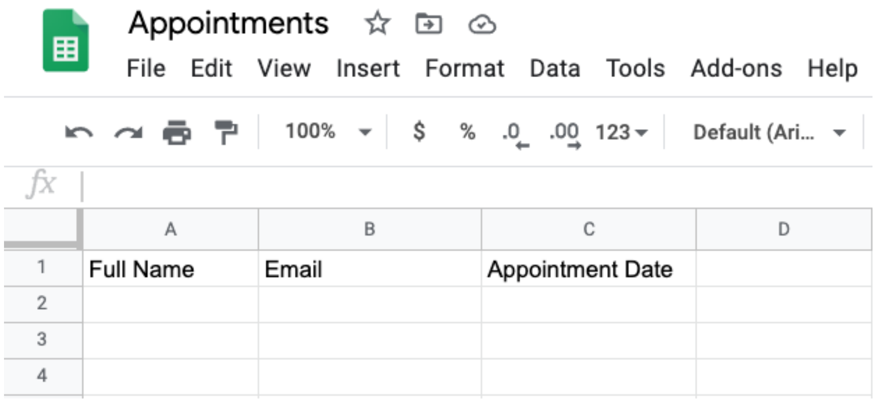
Configuring appointments entity from Google Sheets datasource

To configure your "appointments" entity from Google Sheets:

1. From the AppSheet UI, click **Data** in the left navigation bar.
2. Click **New Table**.
3. Select your configured Google Data Source.

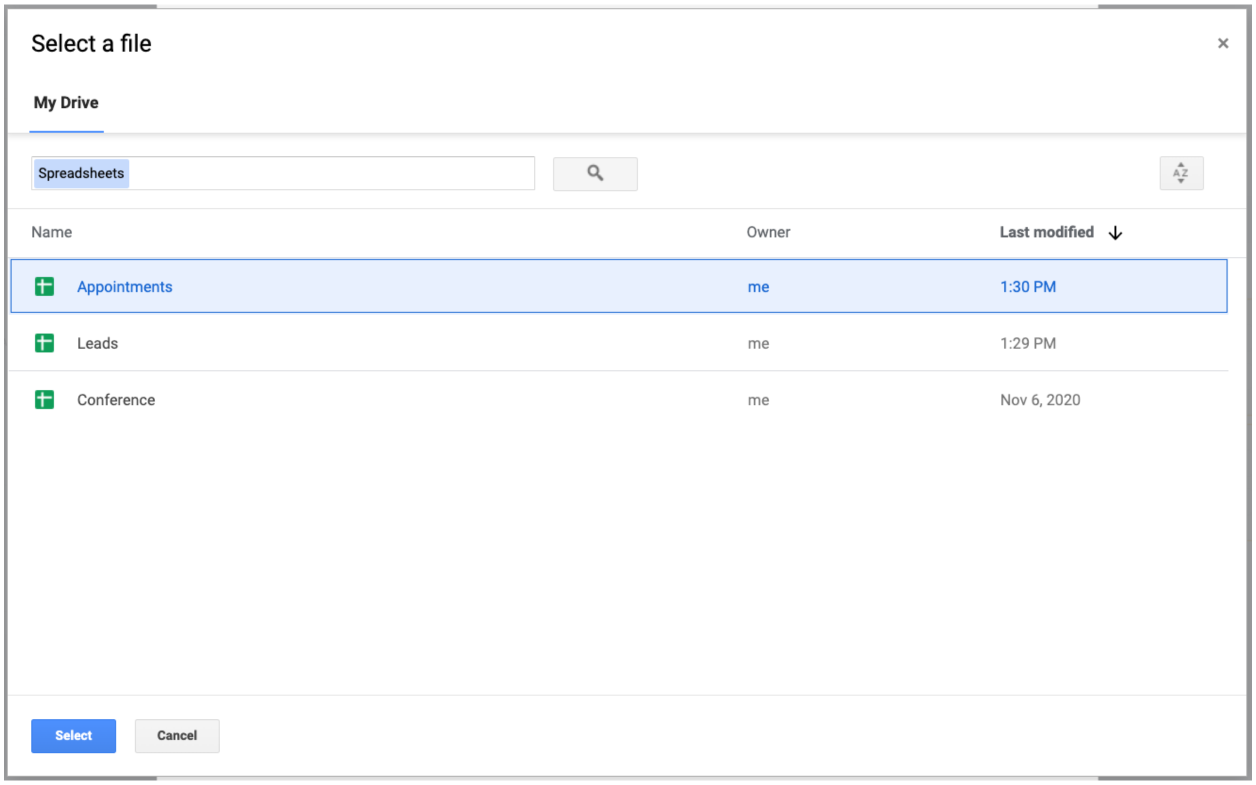


4. Ensure you create a new Google Sheet named "Appointments" (with the column names as below, the sheet (tab) name should be "appointments") in the Google Drive accessible to your AppSheet account



	A	B	C	D
1	Full Name	Email	Appointment Date	
2				
3				
4				

5. Use the file browser to select the appropriate **Appointments** Sheet as the data source.



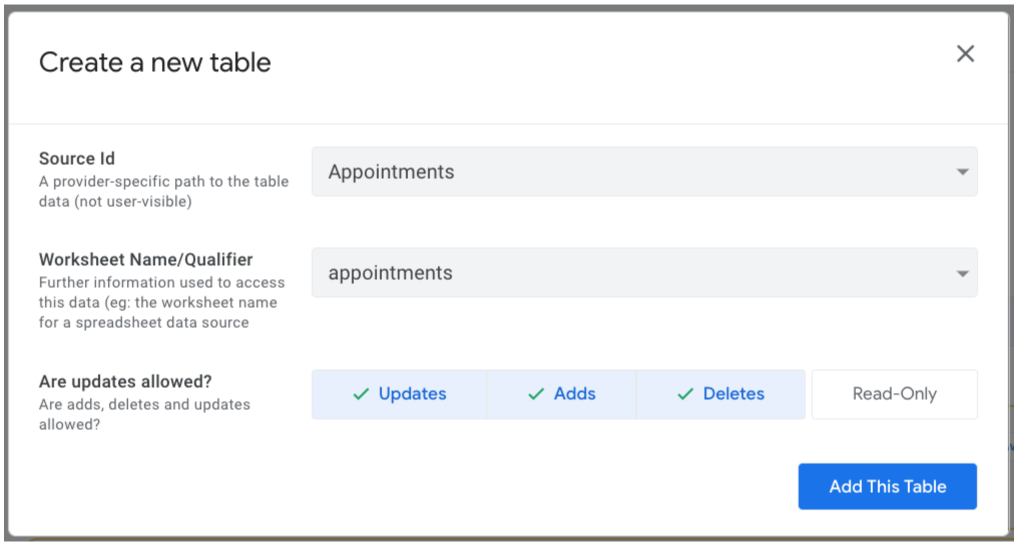
Select a file

My Drive

Spreadsheets

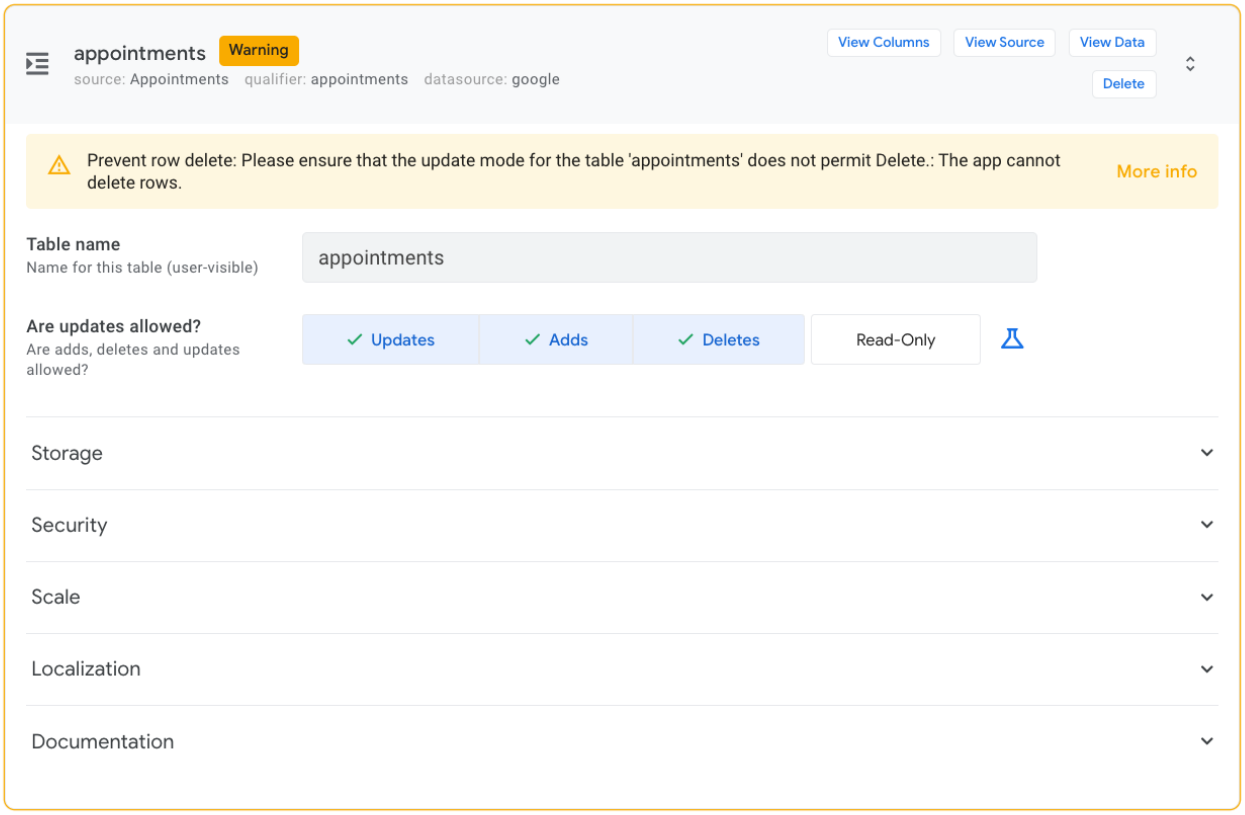
Name	Owner	Last modified
Appointments	me	1:30 PM
Leads	me	1:29 PM
Conference	me	Nov 6, 2020

6. Select **appointments** as the table for your entity.



Create a new table

Source Id A provider-specific path to the table data (not user-visible)	Appointments
Worksheet Name/Qualifier Further information used to access this data (eg: the worksheet name for a spreadsheet data source	appointments
Are updates allowed? Are adds, deletes and updates allowed?	✓ Updates ✓ Adds ✓ Deletes Read-Only



appointments Warning

source: Appointments qualifier: appointments datasource: google

View Columns View Source View Data Delete

Prevent row delete: Please ensure that the update mode for the table 'appointments' does not permit Delete.: The app cannot delete rows.

More info

Table name
Name for this table (user-visible)

appointments

Are updates allowed?
Are adds, deletes and updates allowed?

✓ Updates ✓ Adds ✓ Deletes Read-Only

Storage

Security

Scale

Localization

Documentation