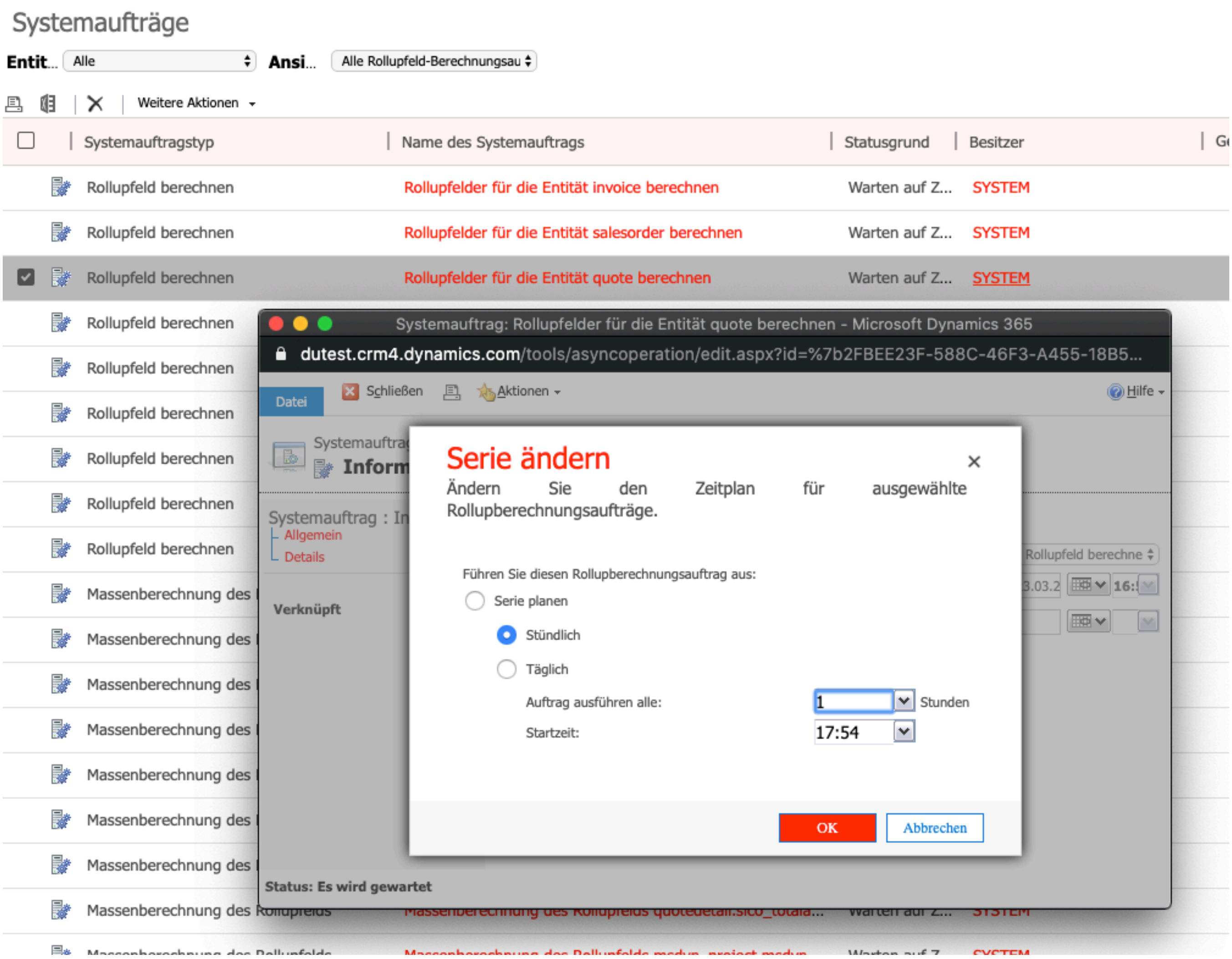
How a Rollup field works

Rollup fields can be a handy tool at your disposal if you need to calculate aggregate values of related records to give the user insights into data in the record relationships. One common scenario is using the COUNT() function to count the number of associated records. This can also be done using hierarchical relationships, i.e., aggregate values from related records and their related records.

There are two ways the rollup field calculates values. By default, there is a reoccurring system job created automatically for each field, which recalculates the values at a scheduled interval (every 12 hours by default). Another way values are calculated is when a user triggers the calculation manually for the record they are on by clicking the refresh button underneath the field.

An administrator of the Power Platform or Dynamics 365 organization may change the interval by going to **Settings > System Jobs > All Rollup Field Calculation Jobs** and locating the entity you want to alter the interval for, in my case ‚**Quote‘.**



Systemaufträge

Entit... Alle Ansi... Alle Rollupfeld-Berechnungsau

☐	Systemauftragstyp	Name des Systemauftrags	Statusgrund	Besitzer	Ge
☐	Rollupfeld berechnen	Rollupfelder für die Entität invoice berechnen	Warten auf Z...	SYSTEM	
☐	Rollupfeld berechnen	Rollupfelder für die Entität salesorder berechnen	Warten auf Z...	SYSTEM	
☑	Rollupfeld berechnen	Rollupfelder für die Entität quote berechnen	Warten auf Z...	SYSTEM	
☐	Rollupfeld berechnen				
☐	Rollupfeld berechnen				
☐	Rollupfeld berechnen				
☐	Rollupfeld berechnen				
☐	Rollupfeld berechnen				
☐	Rollupfeld berechnen				
☐	Massenberechnung des				
☐	Massenberechnung des				
☐	Massenberechnung des				
☐	Massenberechnung des				
☐	Massenberechnung des				
☐	Massenberechnung des				
☐	Massenberechnung des				
☐	Massenberechnung des Rollupfelds	Massenberechnung des Rollupfelds quotedetail.sico_totala...	Warten auf Z...	SYSTEM	

Systemauftrag: Rollupfelder für die Entität quote berechnen - Microsoft Dynamics 365

dutest.crm4.dynamics.com/tools/asyncoperation/edit.aspx?id=%7b2FBEE23F-588C-46F3-A455-18B5...

Status: Es wird gewartet

Serie ändern

Ändern Sie den Zeitplan für ausgewählte Rollupberechnungsaufträge.

Führen Sie diesen Rollupberechnungsauftrag aus:

○ Serie planen

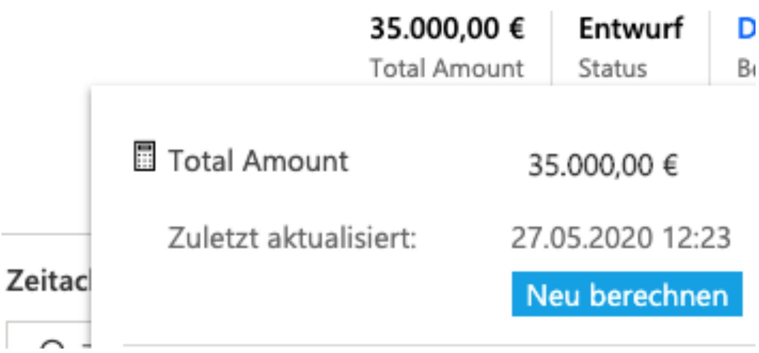
● Stündlich

○ Täglich

Auftrag ausführen alle: 1 Stunden

Startzeit: 17:54

So that means unless we manually force the recalculation, we can only change the schedule of the jobs and not the frequency.



35.000,00 € Total Amount | Entwurf Status | D Be

Total Amount 35.000,00 €

Zuletzt aktualisiert: 27.05.2020 12:23

Automating field recalculation

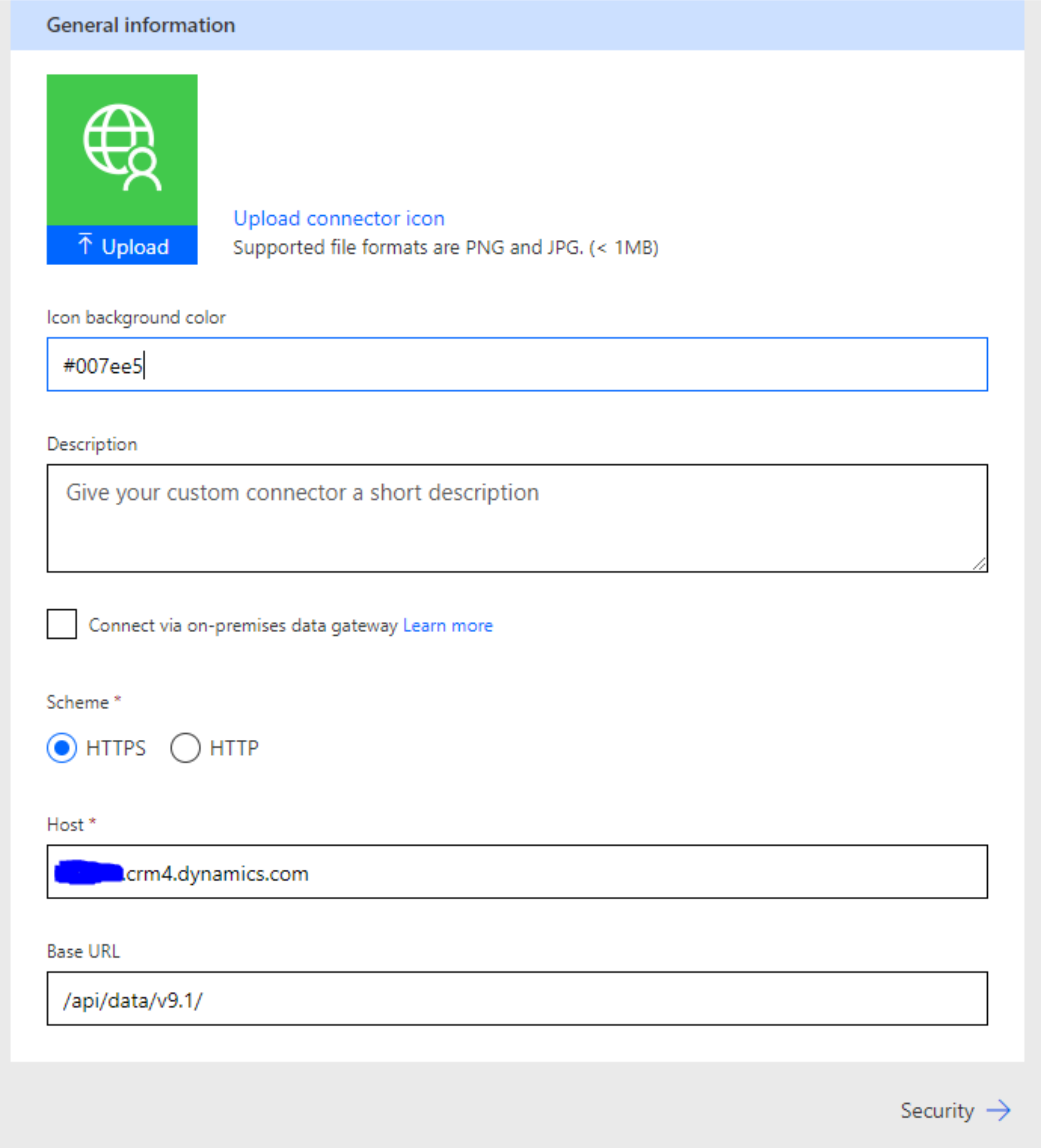
There is, fortunately, a way to make sure the fields stay updated using the Power Platform Web API. We can make use of the CalculateRollupField function. (**more info here:** https://docs.microsoft.com/en-us/dynamics365/customer-engagement/web-api/calculaterollupfield?view=dynamics-ce-odata-9). I will demonstrate how you can go about calling this function on demand when you need it. We will begin by creating a custom connector in Power Automate and then creating a Flow that calls it to update the fields we have selected.

Creating a Custom Connector

You will first need to make sure that you already have a valid app registered in Azure Active Directory. If you do not, you can create one for your Power Platform instance by following the guide found here: https://docs.microsoft.com/en-us/powerapps/developer/common-data-service/walkthrough-register-app-azure-active-directory

Here are the steps to creating the CalculateRollupField Custom Connector:

1. Sign in to Power Automate and navigate to **Data > Custom Connectors**
2. Click **+ New custom connector** and **‚Create from blank‘**.
3. Enter a name and then you will be directed to the General Information tab where you will fill in the Host and Base URL fields as shown below.
4. Once you are done, click the **Security** button to navigate to the next screen.



Icon background color: #007ee5

Description: Give your custom connector a short description

□ Connect via on-premises data gateway Learn more

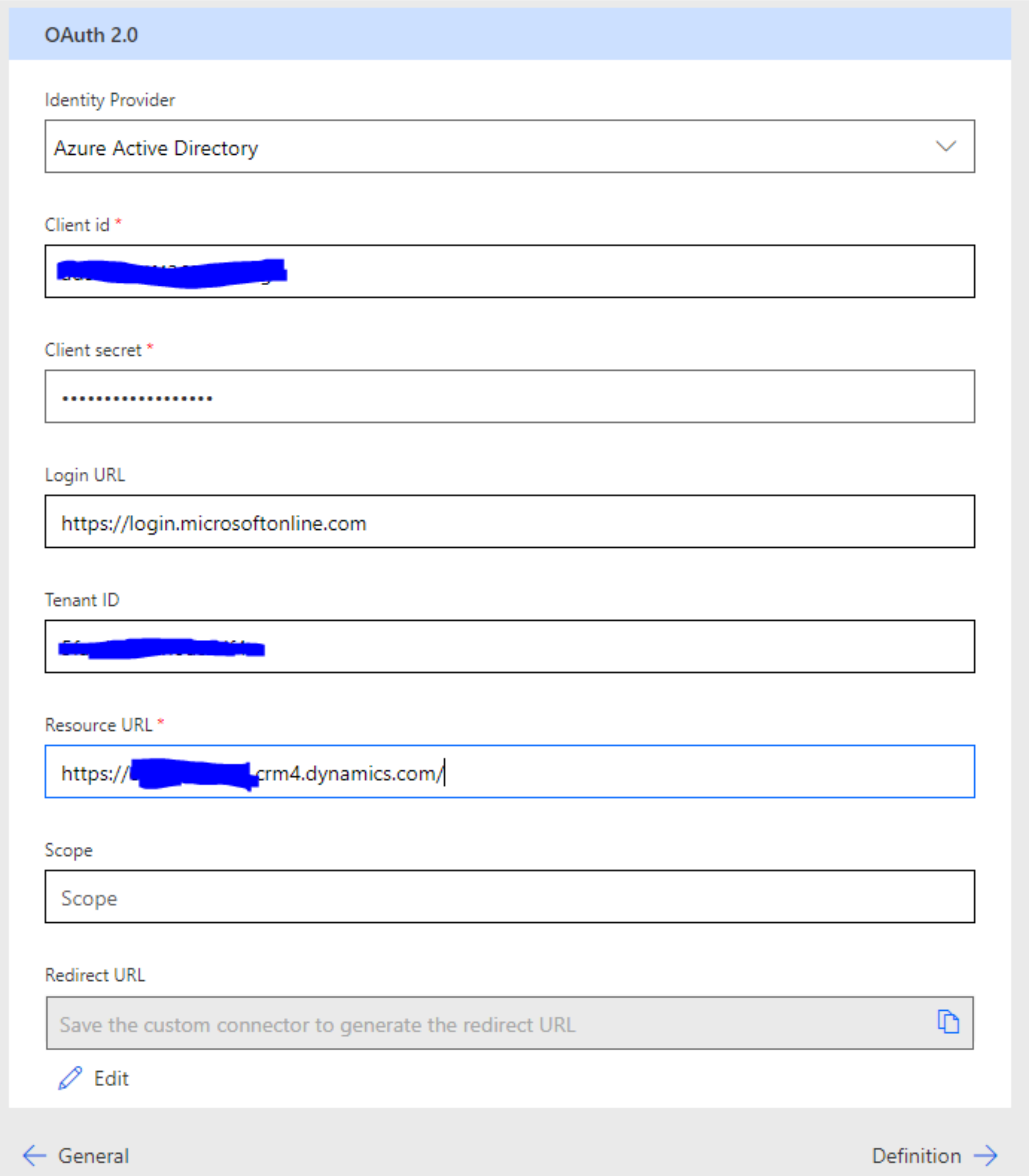
Scheme *: ● HTTPS ○ HTTP

Host *: [redacted].crm4.dynamics.com

Base URL: /api/data/v9.1/

5. Select OAuth 2.0 as the Authentication type.

6. For Identity Provider select Azure Active Directory and fill in the details from your registered app as shown below:



OAuth 2.0

Identity Provider
Azure Active Directory

Client id *
[redacted]

Client secret *
•••••••••••••••••

Login URL
https://login.microsoftonline.com

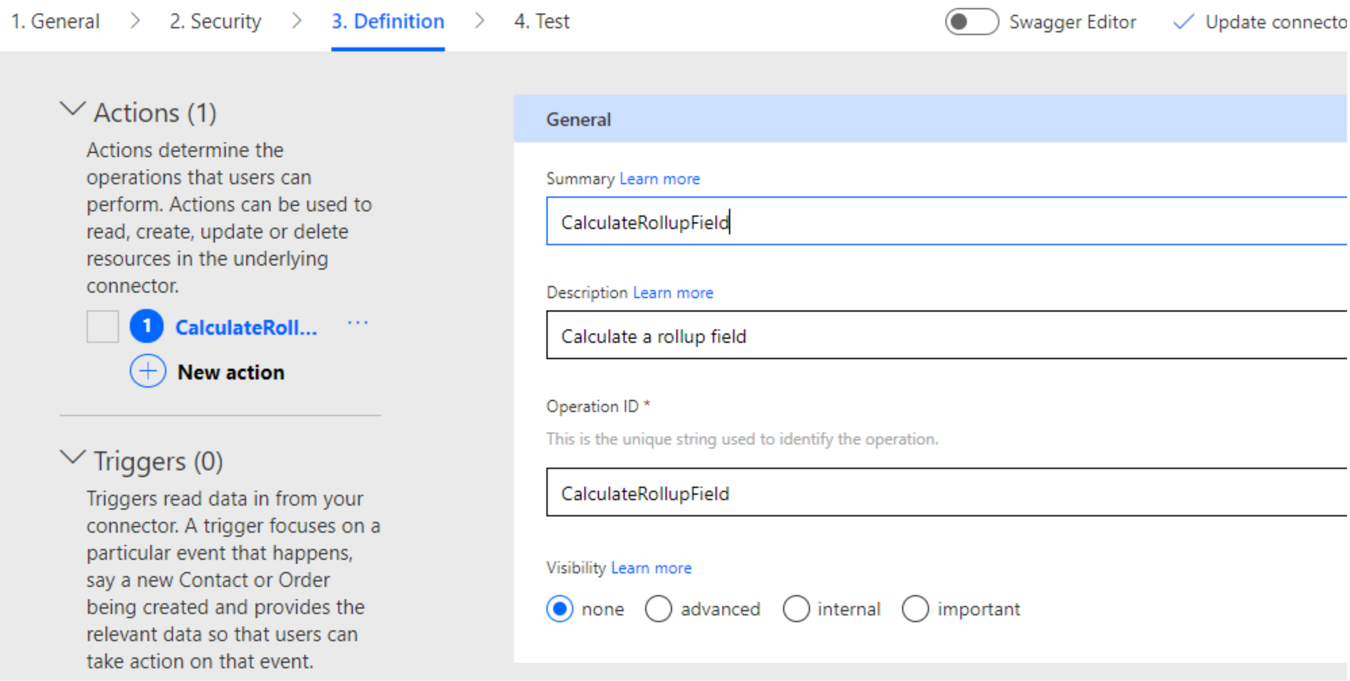
Tenant ID
[redacted]

Resource URL *
https://[redacted]crm4.dynamics.com/

Scope
Scope

Redirect URL
Save the custom connector to generate the redirect URL

7. Add an action by clicking ‚New Action‘ and enter all required information.



Actions determine the operations that users can perform. Actions can be used to read, create, update or delete resources in the underlying connector.

Triggers read data in from your connector. A trigger focuses on a particular event that happens, say a new Contact or Order being created and provides the relevant data so that users can take action on that event.

Summary Learn more: CalculateRollupField

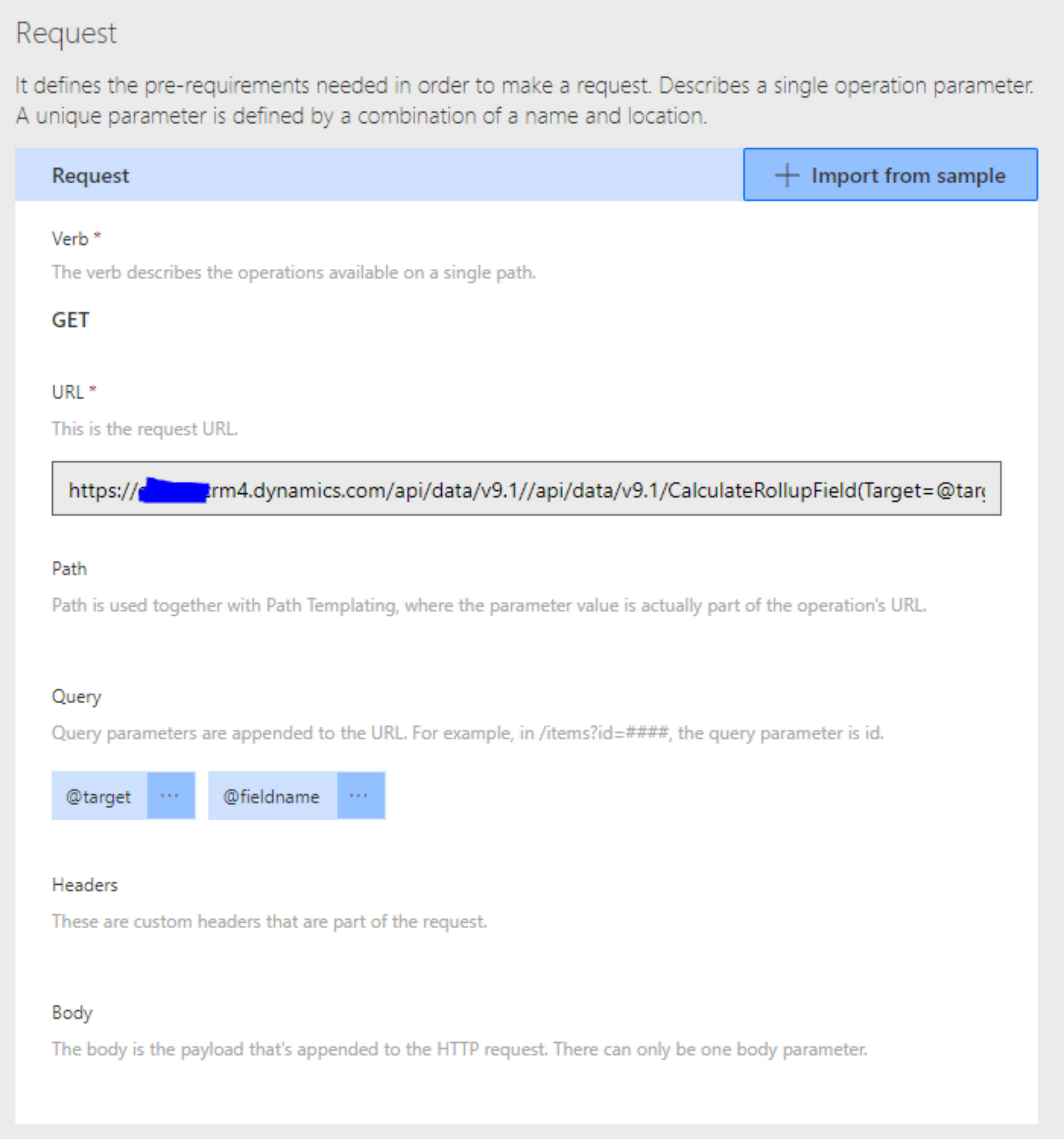
Description Learn more: Calculate a rollup field

Operation ID *
This is the unique string used to identify the operation.
CalculateRollupField

Visibility Learn more
● none ○ advanced ○ internal ○ important

8. To define the requirements of the request you've created click ‚Import from sample'

9. In the Import from sample form select the option ‚GET' and paste in the following URL pointing to your own environment: https://[YOUR_ORG].crm4.dynamics.com/api/data/v9.1/CalculateRollupField(Target=@target,FieldName=@fieldname)?@target={‚@odata.id':'[your_entity]([your_recordid])'}&@fieldname='[your_fieldname]'



Request

It defines the pre-requirements needed in order to make a request. Describes a single operation parameter. A unique parameter is defined by a combination of a name and location.

Verb *

The verb describes the operations available on a single path.

GET

URL *

This is the request URL.

https://[redacted]rm4.dynamics.com/api/data/v9.1//api/data/v9.1/CalculateRollupField(Target=@tar

Path

Path is used together with Path Templating, where the parameter value is actually part of the operation's URL.

Query

Query parameters are appended to the URL. For example, in /items?id=####, the query parameter is id.

Headers

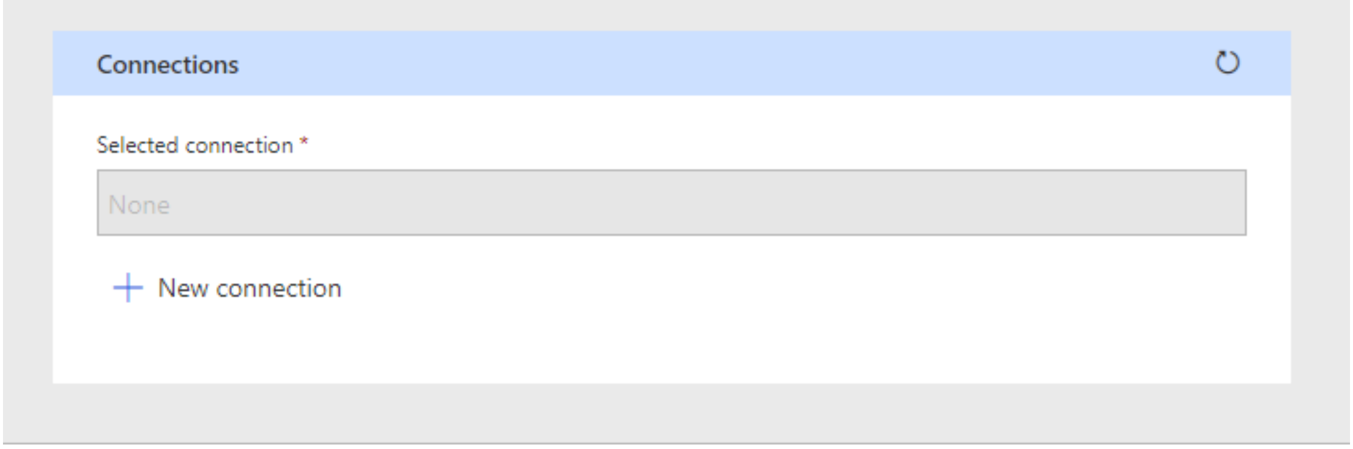
These are custom headers that are part of the request.

Body

The body is the payload that's appended to the HTTP request. There can only be one body parameter.

10. Once you are done, click ‚Import' to create the request and click on the ‚Test' button to proceed. NOTE: make sure you click ‚Create connector' before testing.

11. Create a connection and login to your account for testing.



Connections

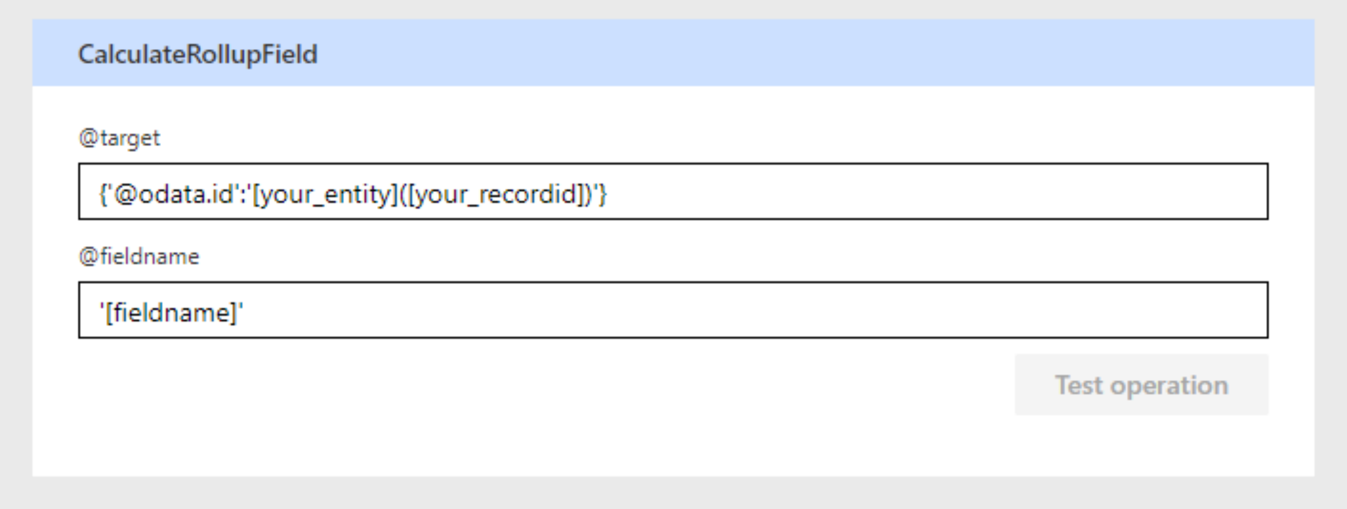
Selected connection *

None

12. In the @target parameter input the following
{‚@odata.id‘:'[your_entity]([your_recordid])‘}

13. In the @fieldname parameter input the following including the single quotation marks (‚)
‚[fieldname]‘

NOTE: Be sure to have a look at the Web API EntityType Reference to ensure you are using the correct name for the Entity Set Path.



CalculateRollupField

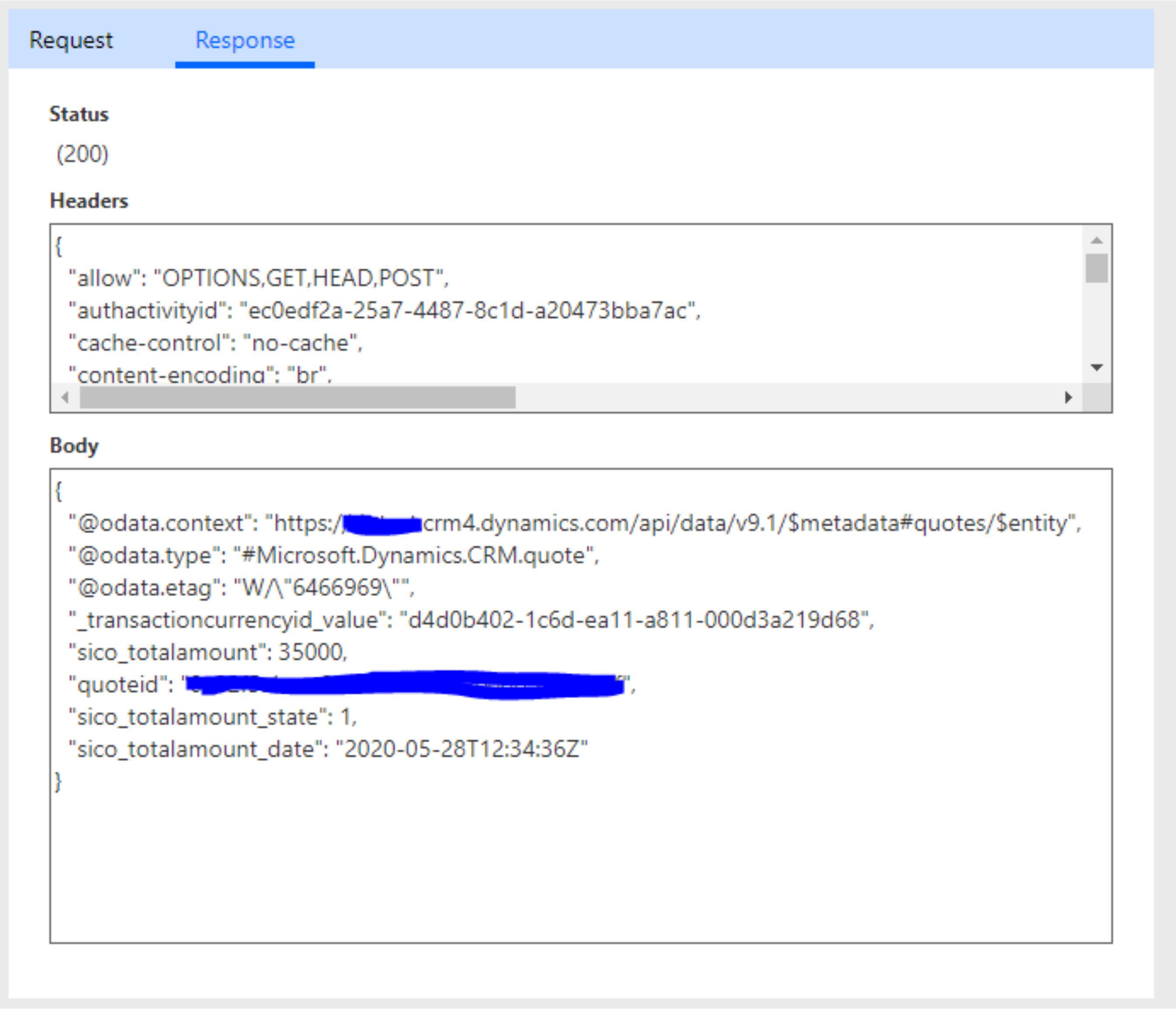
@target

{'@odata.id':'[your_entity]([your_recordid])'}

@fieldname

'[fieldname]'

14. The response should return with a status (200) and it should look something similar to the image below:



Status

(200)

Headers

```
{
  "allow": "OPTIONS,GET,HEAD,POST",
  "authactivityid": "ec0edf2a-25a7-4487-8c1d-a20473bba7ac",
  "cache-control": "no-cache",
  "content-encoding": "br",
```

Body

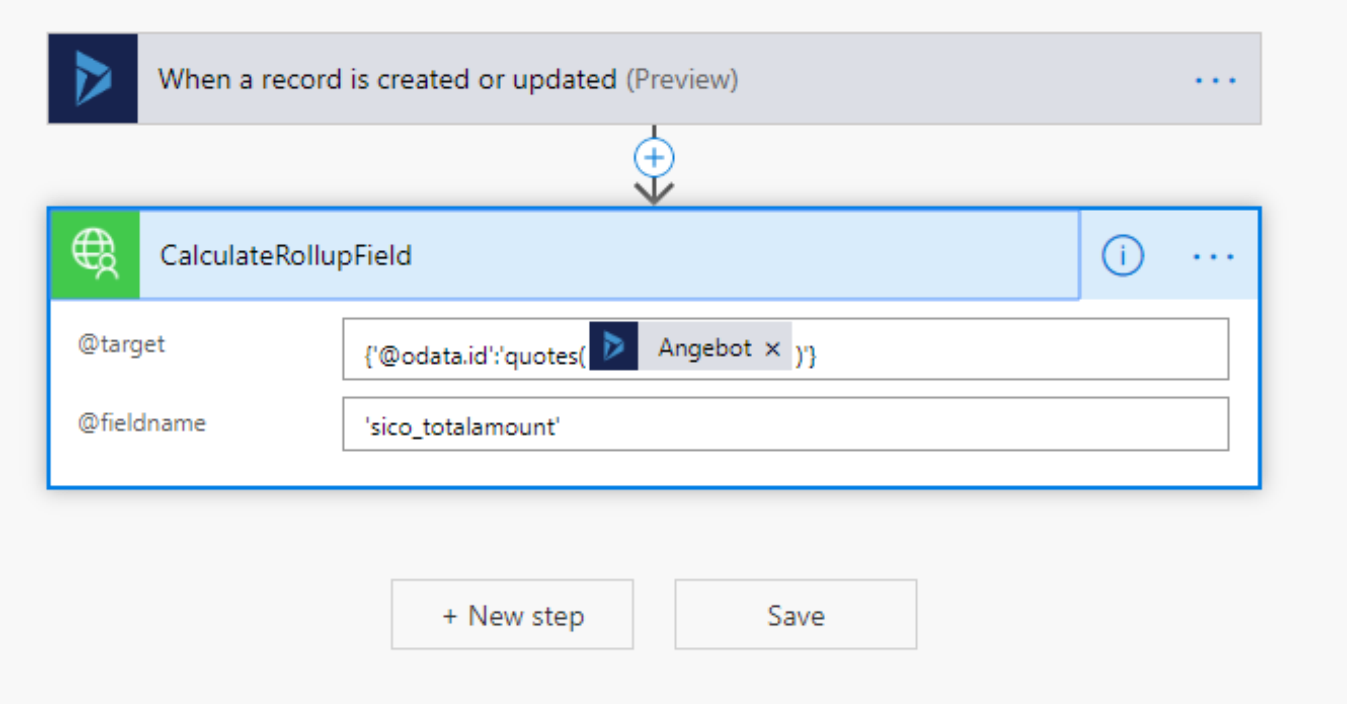
```
{
  "@odata.context": "https:/[redacted]crm4.dynamics.com/api/data/v9.1/$metadata#quotes/$entity",
  "@odata.type": "#Microsoft.Dynamics.CRM.quote",
  "@odata.etag": "W/\"6466969\"",
  "_transactioncurrencyid_value": "d4d0b402-1c6d-ea11-a811-000d3a219d68",
  "sico_totalamount": 35000,
  "quoteid": "[redacted]",
  "sico_totalamount_state": 1,
  "sico_totalamount_date": "2020-05-28T12:34:36Z"
}
```

Create an On-Demand Flow to automatically calculate your Rollup fields

Now that you have successfully created your custom connector, lets put it into action!

For the purposes of this quick demo I created a simple Model-Driven App and included two entities, Quote and Quotedetail.

I have a rollup field on my Quote main form that calculates the sum of the amounts for all positions and displays a ‚Total Amount‘. Basically, every time I update the Quote positions I want my ‚Total Amounts‘ rollup field to automatically be recalculated to reflect the correct amount.



When a record is created or updated (Preview)

CalculateRollupField

@target {'@odata.id':'quotes(Angebot)'}

@fieldname 'sico_totalamount'

Once you have successfully tested your Flow then you are good to go and now have a rollup field that will automatically be recalculated based on the trigger of your choice!