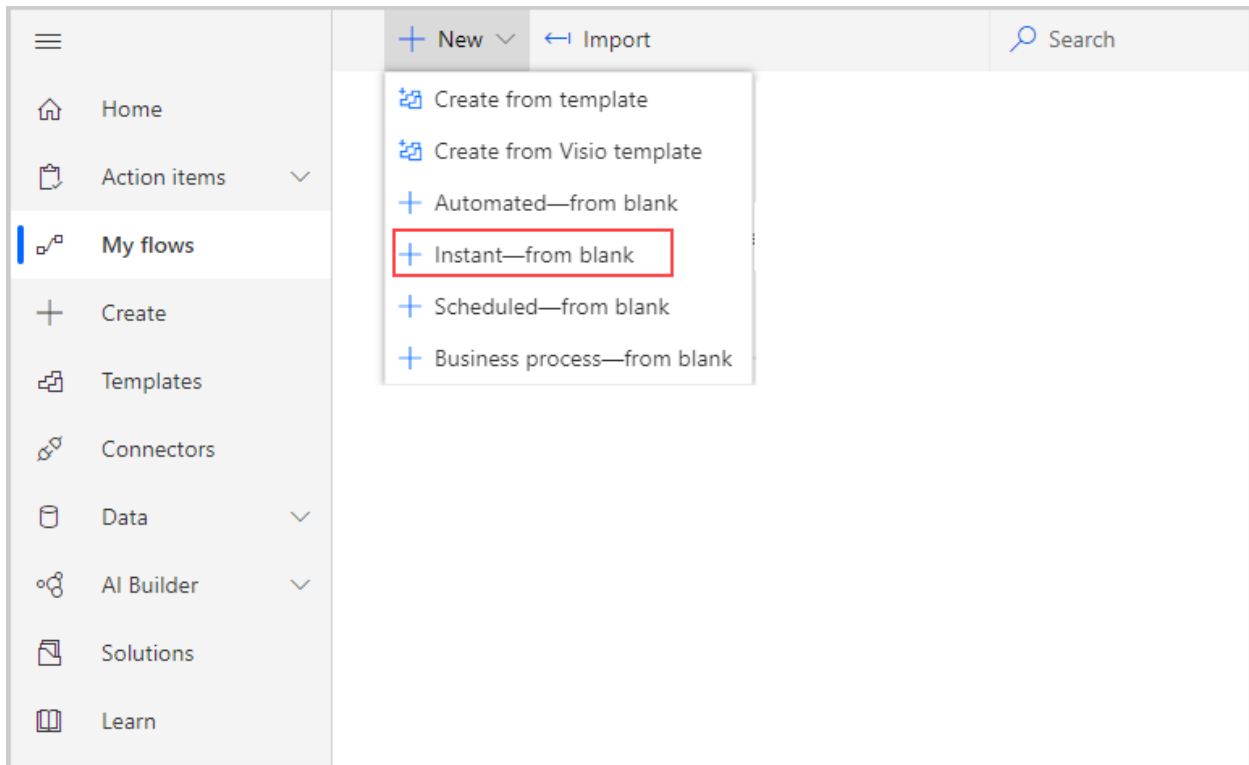


Using LMS365 Connectors in Microsoft Power Automate

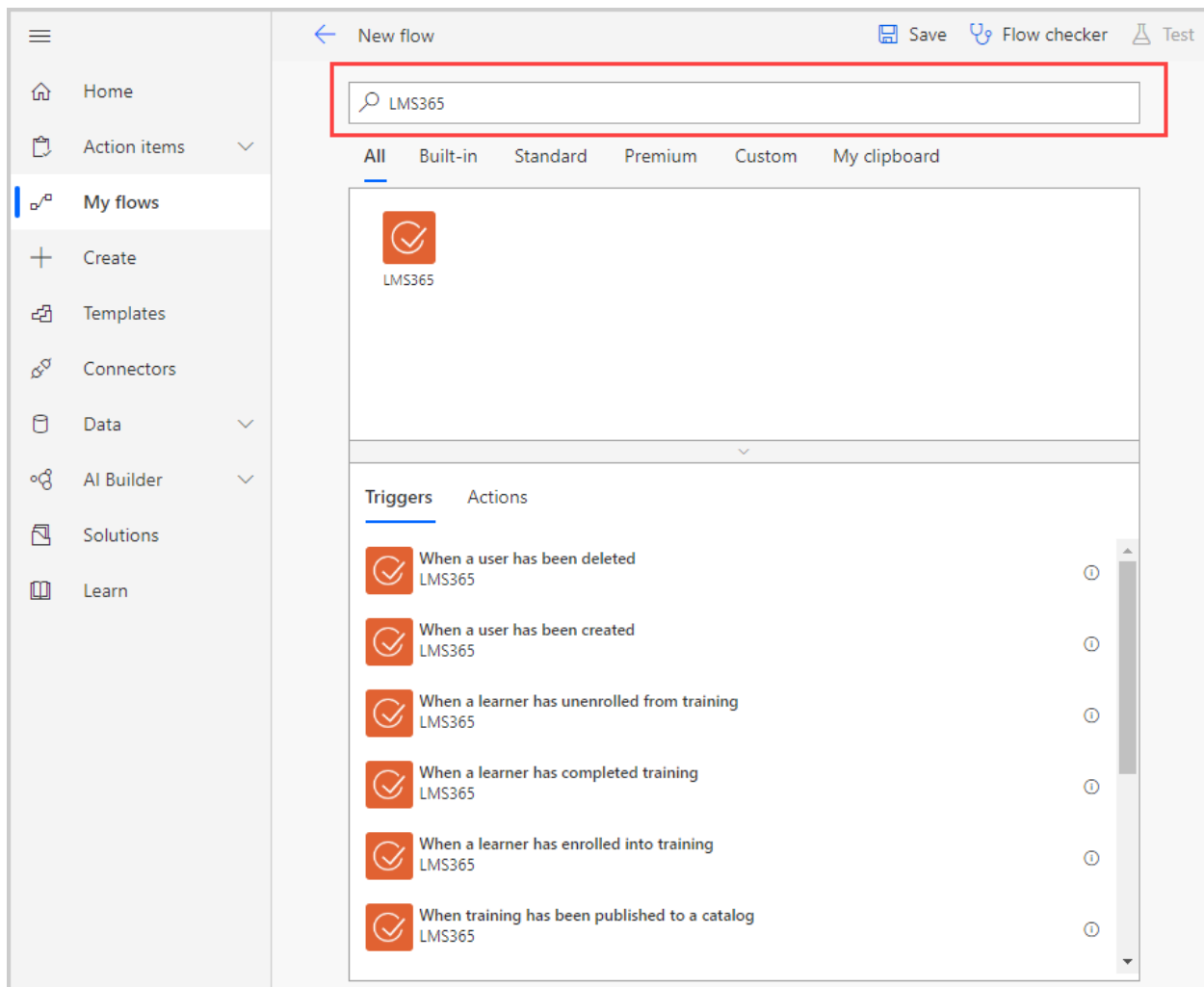
LMS365 give you an ability to automate some of the most common actions and manual processes in the LMS365 via Microsoft Power Automate. It means to react on the events happening LMS365 and send data to the LMS. You can use [the existing templates](#) for your flow or create your own one.

To create your own flow:

1. Go to the Microsoft Power Automate page and <https://flow.microsoft.com> and sign in.
2. Click My Flows section, and then click the New button in the top navigation bar and select Instant – from blank button:



3. On the creation page, you can see the full list of the MS Power Automate connectors. Search for the **LMS365** to see all actions related to the learning system:



The Connector consists of two groups of items:

Triggers – give you the ability to subscribe to some events in the LMS. Each trigger has a proper title that describes when the event is triggered. Usually, that is the starting point of the flow.

For example, the trigger “When a user has been deleted” means that the operation will be triggered when a user has been disabled/deleted from LMS365.

Actions – give you the ability to affect LMS. That means these are the set of predefined API calls to the LMS service.

For example, an action “Approve enrollment request” will approve the enrollment request.

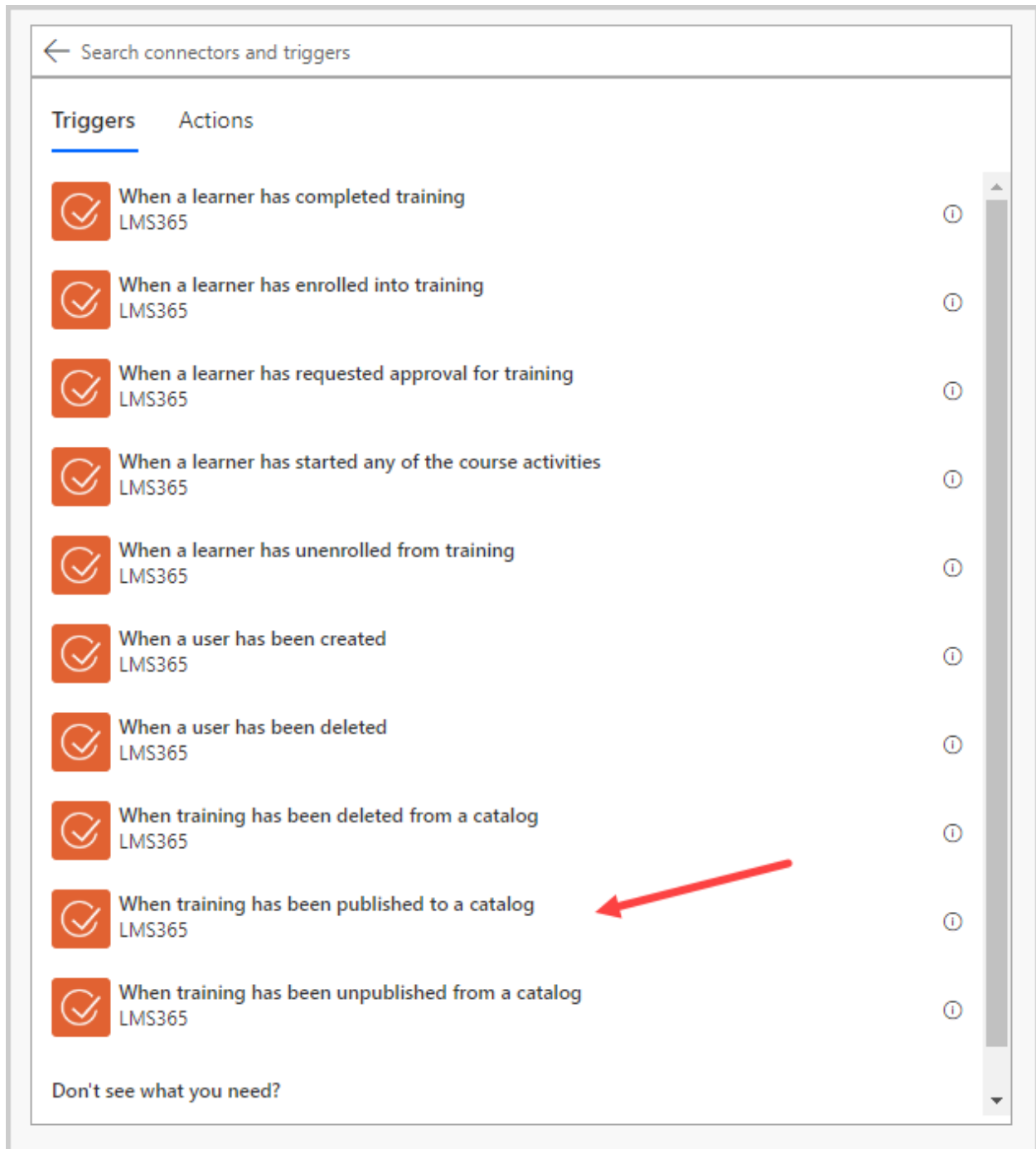
For more information about the actions, visit [the connector reference for the LMS365](#).

4. Select the desired trigger from the list.
5. Add the steps.
6. Search for the required action that can be performed within the LMS365.
7. Fill in the required fields.

8. Add other actions if needed.

Let us create a **simple flow** which reacts to the Course Published event, gets the course information, and sends an email with the course information to the specified users. To perform this action:

- Choose the “When training has been published to a catalog” trigger:

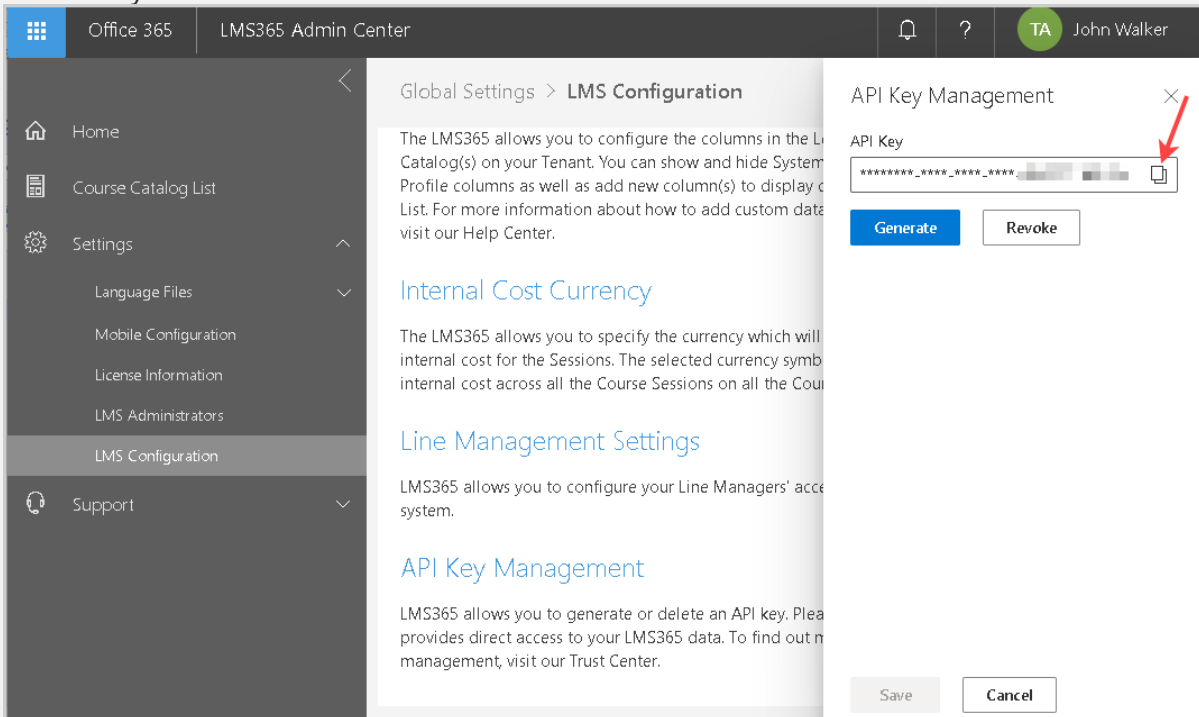


The screenshot shows a user interface for selecting triggers. At the top, there is a search bar with the text "Search connectors and triggers". Below the search bar, there are two tabs: "Triggers" (which is selected and underlined) and "Actions". A list of triggers is displayed, each with a red square icon containing a white checkmark, the trigger name, the connector name "LMS365", and a circular information icon on the right. A red arrow points to the trigger "When training has been published to a catalog". At the bottom of the list, there is a link that says "Don't see what you need?".

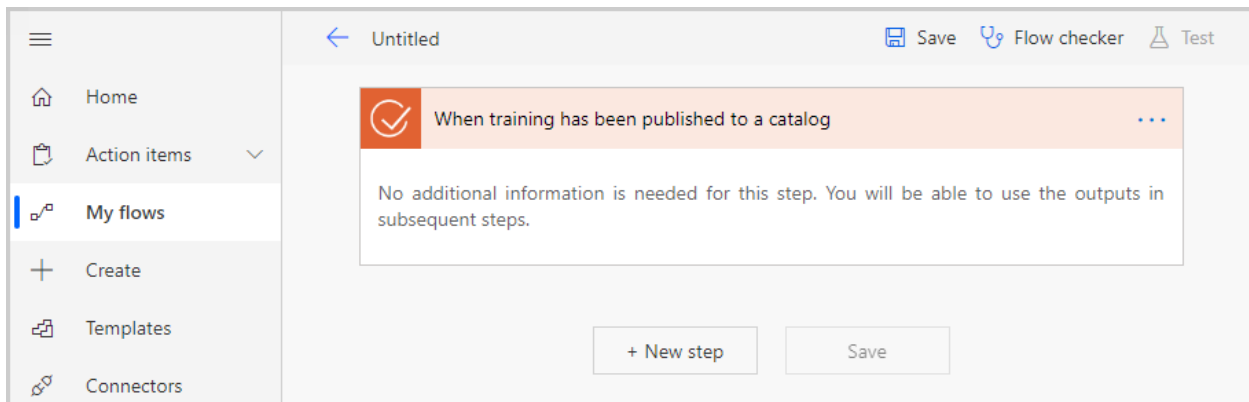
Trigger Name	Connector
When a learner has completed training	LMS365
When a learner has enrolled into training	LMS365
When a learner has requested approval for training	LMS365
When a learner has started any of the course activities	LMS365
When a learner has unenrolled from training	LMS365
When a user has been created	LMS365
When a user has been deleted	LMS365
When training has been deleted from a catalog	LMS365
When training has been published to a catalog	LMS365
When training has been unpublished from a catalog	LMS365

If you use the LMS365 Connector for the first time, enter connection name (LMS365) and credentials for the LMS API service. Please use the API key for both **username** and **password** fields.

- ❗ The API key is a password for your LMS365 data. Keep it safe.
- ❗ To get the API key for your tenant domain e.g. *elearningforce.sharepoint.com*, the Office 365 Global Administrator should navigate to LMS365 Admin Center – Global Settings – Settings – LMS Configuration – API key management and copy the API key:



- After the connection is set, you will see a message that your flow has only a starting point and you need to add new steps:



- Click the New step button.
- Enter LMS365 in the search box to find the available LMS connector action.

In our example, we want to get information about the course that has been published – we select “Get course details” action:

When training has been published to a catalog

Choose an action

LMS365

All Built-in Standard Premium Custom My clipboard

LMS365

Triggers Actions

- Upload File LMS365
- Send an HTTP request LMS365
- Create course category LMS365
- Get course details LMS365
- Create course tag LMS365
- Get course tags LMS365

- Specify the Course Id value in order to get the information about the specific course. Click the field to see all available data you can operate with. Select the Course Id from the list to proceed:

The screenshot displays a workflow editor interface. On the left, a step titled 'Get course details' is highlighted with a red border. It contains a text field for '* Course Id' with the placeholder text 'Specify the internal course id'. Below this field is an 'Expand Query' section listing various data fields: DueDate, Publishing, CertificateExpiry, SharepointWeb(\$select=Url), Categories, Tags, CourseSessions, ProvisioningProgress, and Trainers. At the bottom of this step are '+ New step' and 'Save' buttons. On the right side of the interface, a panel titled 'Add dynamic content from the apps and connectors' is visible. It features a search bar and a list of dynamic content options. A red arrow points to the 'Course Id' option in this list. The list includes: Action, Attempt, Course Id, CourseType, Published, Siteld, TimeStamp, and Training Title.

- After the Course Id value is selected, you can choose what to do with the data returned by the 'Get course details' action (by adding new action). In our example, we will select the Office 365 Outlook Connector and search for the 'Send an email' action:

✓ When training has been published to a catalog ...



↻ Apply to each ...

*Select an output from previous steps

✓ Notifications x

✓ Get course details ...



Office 365 Outlook X

← Search connectors and actions

Triggers Actions

- Office 365 Outlook Create contact (V2)
- Office 365 Outlook Create event (V4)
- Office 365 Outlook Send an email (V2)
- Office 365 Outlook Delete contact (V2)
- Office 365 Outlook Delete email (V2)
- Office 365 Outlook Delete event (V2)
- Office 365 Outlook Export email (V2)
- Office 365 Outlook Find meeting times (V2)
- Office 365 Outlook Flag email (V2)



❗ Once you have set up a specific connection (Outlook, LMS365, SharePoint, etc.), you can reuse that connection in all flows.

and fill in some course data (e.g. title and description):

The screenshot shows a Microsoft Power Automate flow configuration. The flow starts with a trigger "When training has been published to a catalog". This is followed by an "Apply to each" loop containing a "Get course details" action. Inside the loop, there is a "Send an email (v2)" action. The email action has fields for "To", "Subject", and "Body". The "Body" field is currently empty, and a "Dynamic content" panel is open on the right, showing a list of available fields: ShowTimer, StartDate, TimeLimit, Title, Trainers Display Name, Training Title, and Uri. The "Training Title" field is selected. At the bottom of the flow editor, there are buttons for "+ New step" and "Save".

That is all, we are ready with the Flow that is triggered when a Course has been published, gets the detailed information about it, and sends Course's description via email to the specified recipients.

The logic and approach are the same for all other cases.

You can make a request to get the desired data from API via Microsoft Power Automate as well. Let us create one more flow to get the list of all Courses and the enrollments withing the Course Catalog:

- Choose the "Send an HTTP request" trigger and define the methods for the HTTP request. In our case use the GET as method as we need to receive the information.

There are 5 HTTP methods: GET, PUT, POST, PATCH, DELETE.

- the GET method is used for retrieving information,
- the PUT method is used for updating the existing resource,
- the POST method is used for creating a new resource,
- the PATCH method is used for making a partial update on a resource,
- the DELETE method is used for deleting resources.

- Specify the path in the Uri field. You can get these data from [Swagger](#). Navigate to the desired endpoint. In our case we use the endpoint GET /odata/v2/Courses that returns the list of all Courses. As we want to have all enrollments, we need to specify the "Enrollments" as the parameter. So the path will be Courses?\$expand=Enrollments:

The screenshot shows a configuration interface for a flow. At the top, there is a blue button labeled "Manually trigger a flow" with a hand icon and a three-dot menu. Below it, a downward arrow points to an orange step labeled "Send an HTTP request" with a checkmark icon and a three-dot menu. The configuration for this step is as follows:

* Method	GET
* Uri	Courses?\$expand=Enrollments
Headers	Enter key Enter value
Body	Enter request content in JSON

Once you have tried out the endpoint you can find the Uri in the Request URL field as well.

i You can find all available parameters in the Responses field – Model.

When you run the flow, you will find required information in the Body field. The API will response in the JSON format:

```
"Enrollments": [
```

```

{
  "Id": "407de1a8-e9a0-40a6-94ed-e70d7a1992b4",
  "CourseId": "144e2fc1-76d1-4387-ae26-27df7f77ef26",
  "UserId": "ae5119a5-3426-4402-982a-f071b54f0545",
  "UserLoginName": "c:0t.c|tenant|e6e6f293-8809-4c64-a888-71f824be52df",
  "Roles": [
    "Teacher"
  ]
}

```

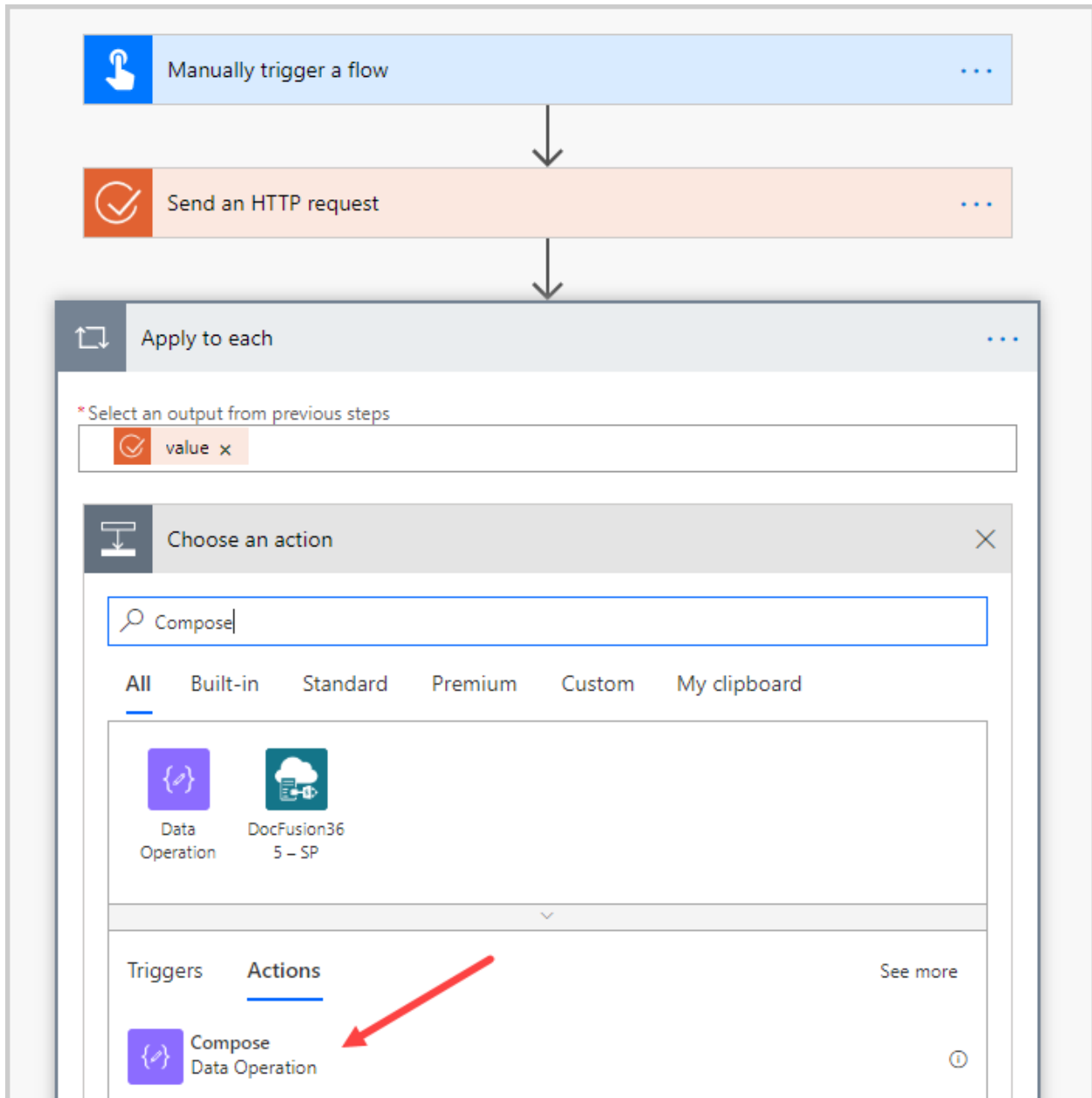
❗ You can find more about JSON format in the [JSON Tutorial](#).

To apply it for each course you need to add the new "Apply to each" action, click Expression and specify the following expression, which is the output from the previous step and looks like:

```
body('Send_an_HTTP_request')['value']
```

The screenshot shows a workflow editor with three steps: "Manually trigger a flow", "Send an HTTP request", and "Apply to each". The "Apply to each" step is currently empty, and a modal is open to add an expression. The expression "body('Send_an_HTTP_request')['value']" is entered in the "Expression" field. Below the modal, a list of string functions is visible, including "concat(text_1, text_2?, ...)" which combines any number of strings together.

- To get all enrollments add one more "Compose" action.



In the Input field add the desired expression. As we want to get the enrollments for each of the courses, we need to specify the following expression:

```
items('Apply_to_each')['Enrollments'].
```

❗ For the detailed information about functions see [the guide to using functions in expressions](#).

That is all, we are ready with this flow and get all enrollments:

✓ Your flow ran successfully. ✕

👉 Manually trigger a flow 0s ✓

✓ Send an HTTP request 1s ✓

↻ Apply to each 22s ✓

< Previous < Previous failed Show 1 of 101 Next failed > Next >

{ } Compose 0s ✓

INPUTS

[Click to download](#)

OUTPUTS

Outputs

```
[
  {
    "Id": "a0241891-a284-4fef-a450-35d0ff5f2670",
    "CourseId": "915b0890-91d4-4fb3-b9fc-1684303d8da3",
    "UserId": "ae5119a5-3426-4402-982a-f071b54f0545",
    "UserLoginName": "c:0t.c|tenant|e6e6f293-8809-4c64-a888-71f824b",
    "Roles": [
      "Learner"
    ]
  }
]
```

Useful links:

[Swagger](#);

[The JSON tutorial](#);

[The API fast track learning](#).