

## Unique Variable

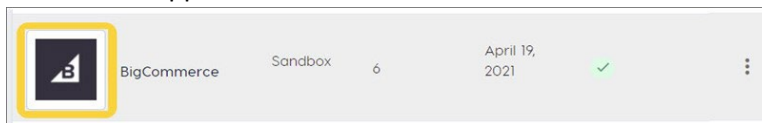
A unique variable filters API responses so each new payload contains only objects, or records, you want and not all in the database. The initial workflow instance moves every object and results in a large payload because all objects in the source are new to the destination, but subsequent workflow instances only move new or updated objects. For example, to maintain employee files from a local office in the cloud-based HR database, only new and updated files need to migrate at a set interval, such as once a day. That's where the unique variable comes in. A unique variable

- Creates a smaller payload, so it saves time and money.
- Eliminates duplicates that could be created if you loaded every employee record with each new workflow instance.

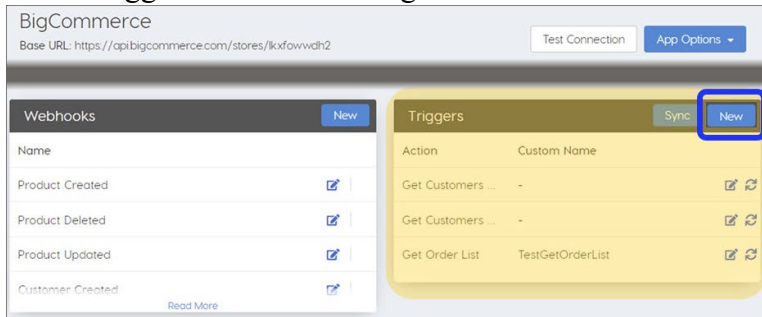
### Set up a unique variable

Set the unique variable for an Application Trigger, so it becomes part of the trigger that initiates a new workflow instance.

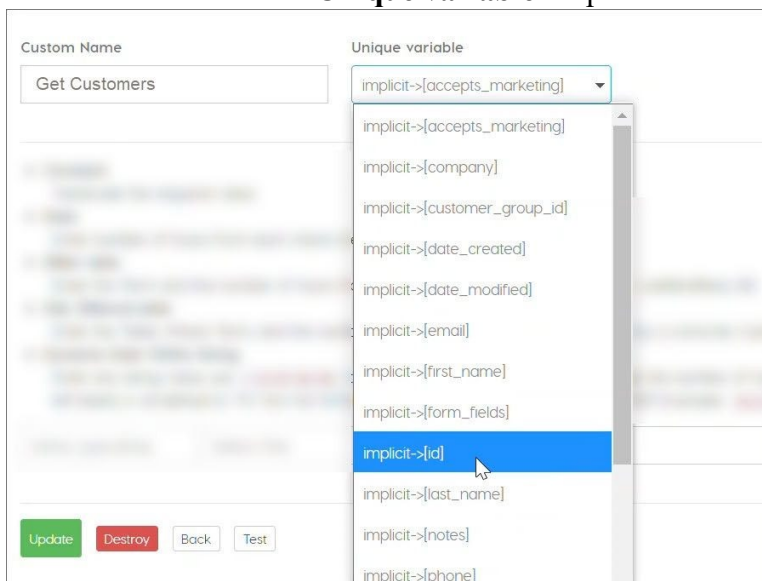
1. Click on the App on **MYAPPS**.



2. Go to Triggers in the bottom right and click the **New** button.



3. Choose an Endpoint and click **Next** to open the Edit Trigger page.
4. Choose a field from the **Unique variable** drop-down list.



**Note:** You cannot choose a field that does not exist within a workflow application, but you can select a field that may not return the data you need.

5. Click the **Update** button.

Each workflow instance then uses the unique variable.

### How it works

A unique variable can only be a field in an endpoint for one of the workflow's applications. Enter the endpoint fields on the APP page, They then appear in the drop-down list for a trigger's **Unique Variable** field.

**Warning:** The trigger only checks for a change in the unique variable. If any field other than the unique variable has changed since the last workflow instance, the object is NOT part of the new payload.

With this in mind you want to use a field that rarely, if ever, changes, for example the **Employee Number**, or a field that changes whenever an employee record is updated, such as the **Date\_Modified**.

The two options produce different outcomes.

- **Employee Number**

The payload will contain only NEW employee records.

**PRO:** This minimizes duplicates.

**CON:** Each new workflow instance only pulls new employees but misses those that may have other updated fields. For example, if a customer's name was updated but the record keeps the same ID number.

- **Date\_Modified**

The payload will contain the new employees as well as those existing employees whose records were updated since the last workflow instance.

**PRO:** The destination database is fully updated, without duplicates.

**CON:** You may not be able to determine which records are new.