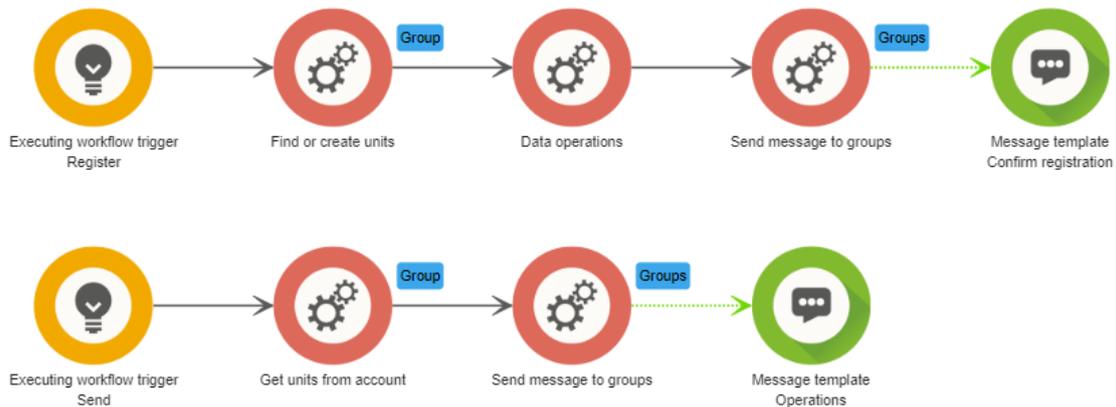# Creating a mailing list

Starting the workflow and inputting the required data will be done using the Bosbec REST-API. This guide requires some knowledge on how to use the REST-API. More information on how to get an authentication token and how to execute workflows using the API can be found here;
https://help.bosbec.io

This workflow will consist of two parts; one for registering new users and one for sending emails to the users of a specific group.

Register new users

This process will require one trigger, three jobs, and one message template.

This will allow us to:

1. Trigger the workflow by sending an API-call towards it
2. Find or create a unit for the user
3. Add it to a group of recipients
4. Send a confirmation to the user telling them they've been added to the group.

The API-call which will be used for this part will look like this;

```
{
        "workflowId": "00000000-0000-0000-0000-000000000000",
        "triggernames": "Register",
        "metadata": {
                "user-email": "user@bosbec.com",
                "group-code": "Test"
        }
}
```
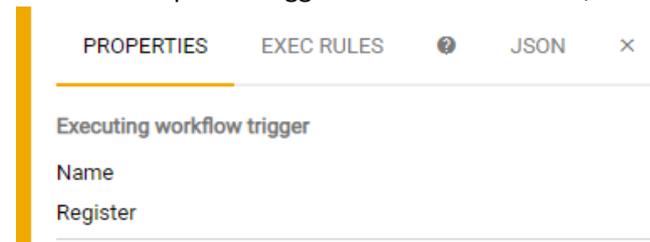
The workflow ID can be found by either right clicking the workflow in the workflow list and select "Details" or by opening the workflow and grabbing the "wfid" in the URL.

bec.io/&wfId=1d0ee906-03c1-4a47-853b-23d8858b6f18

Building the workflow

Start by adding an "**Executing workflow trigger**". We need to give the trigger a name in order to execute this specific trigger later on. In this case, we'll name the trigger "Register".

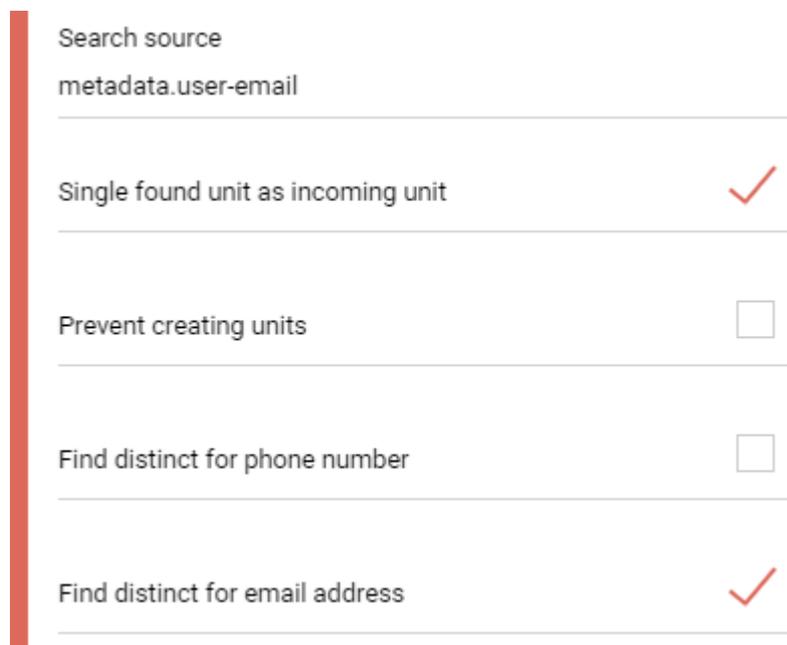| PROPERTIES | EXEC RULES | ❓ | JSON | ✕ |

Executing workflow trigger

Name

Register

Next up, we'll need to find a unit corresponding to the email entered in the property "user-email", or create a new one if no unit exists. This can be done using the job "**Find or create units**".

"**Search Source**" needs to be set to the metadata property we'll be using to specify the user's email.

An incoming unit can be a unit representing the sender that triggered the workflow but can also be a unit that you've picked up from your account to work actively with. Ticking the checkbox for "**Single found unit as incoming unit**" allows us to work with this unit in the next steps of the workflow.

We also need to let the job know that we are looking for a unit matching an email address rather than a phone number by ticking the checkbox for "**Find distinct for email address**".

Search source

metadata.user-email

Single found unit as incoming unit ✓

Prevent creating units ☐

Find distinct for phone number ☐

Find distinct for email address ✓

Now that we've found or created a unit for the user we can tag it. This tag is what we'll use to identify the different groups of recipients.

We'll use the job "**Data operations**" to tag the unit. Add a "set data-operation" to the job by clicking **New Operation**.

We used "group-code" in the example API-call so that's what we'll put in the source. Since we found the unit earlier and made it the incoming unit, we can now write incomingunit.tags as destination. This operation will now add the tag "Test" to the unit with email address "user@bosbec.com".

Set data ^

Source
metadata.group-code

Destination
incomingunit.tags

Order
0

Remove

Since we already have the relevant user as incoming unit we can simply add the "**Send message to groups**" job but not connect any groups to it. As no group is connected, the message will be sent to any units currently in the workflow context.

Add a message template to the workflow and give it an email template. Give the email a subject and a sender. When filling in the message body, you can add the metadata "group-code" within brackets. This will replace the bracketed string with whatever the metadata reads. In this case it will be replace by "Test".

Email message ^

Subject
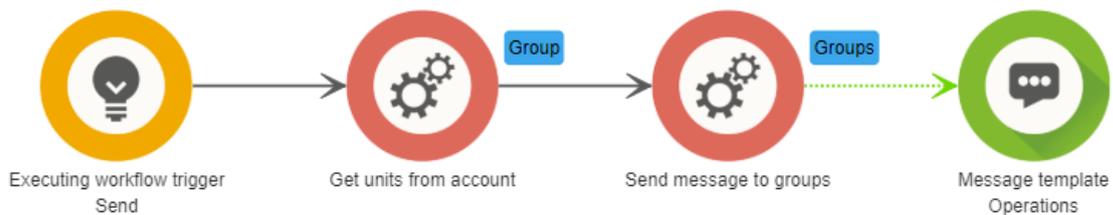Confirmation on new registration

From
noreply@bosbec.io

Body
Hi,

You have now been registered for group [group-code].

Send message

Sending the actual message will require one trigger, two jobs, and a message template.

The workflow will:

1. Be executed using an API-call
2. Pick up the users with a specific tag
3. Send a message template to the users.

The API-call which we will be using for this part will look like this;

```
{
        "workflowId": "00000000-0000-0000-0000-000000000000",
        "triggernames": "Send",
        "metadata": {
                "group-code": "Test",
                "subject": "Test Subject"
                "message-text": "Test message"
        }
}
```

Building the workflow

Start by adding an "**Executing workflow trigger**" and naming it "Send".

First thing we should do is finding all units that have been registered to the group-code provided in the API-call. We'll do this by adding a "**Get units from account**" job. This allows us to specify that we are looking for a specific tag and to add all units that match this tag to the workflow context.

Note that "**Use workflow context meta data value**" need to be set to true. If not, we will be searching for the string "metadata.group-code" instead of the actual value of that metadata.

Use workflow context meta data value ✓

Metadata

Key

Value

key

value

Phone criteria

Email criteria

With tags

metadata.group-code

Without tags

Single found unit as incoming unit ☐

Once the correct units have been added to the workflow context, all that is left to do is to send the message. Once again, add a "**Send message to groups**" job without connecting a group. The message will be sent to the units currently in the workflow context.

Add a message template to the workflow and give it an email template. Give the email a sender address. When filling in the subject, add the metadata "subject" within brackets. Do the same for the body but using metadata "message-text".

Email message ∧

Subject

[subject]

From

noreply@bosbec.io

Body

[message-text]

Finished workflow

The workflow should now look something like the screenshot below and you are ready to register users and send them emails.