



# SESSION **02**

Edge Analytics Online Training

## CREATE YOUR FIRST FLOW

Work with modules and flows





# Session 2

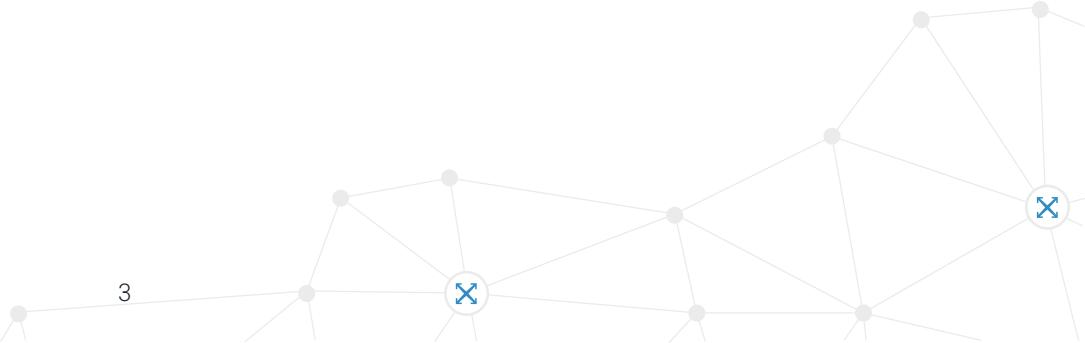
## Agenda

- UI introduction
  - Flows
  - The Flow Studio
  - Module settings
  - Testing and Deploying Flows
- Other functions
  - Message filtering
  - Managing Flows
- Modules covered:
  - Data Generator module
  - Aggregation module
  - Text Template module
  - MQTT Pub Client
- Exercise 1: Build your first flow – step by step



# FLows

This is what You create with Crosser



# Flows

- Created on the [Flows page](#) or from within the [Flow Studio](#):
  - Using the “+” button (empty flow)
  - From a FlowApp
  - From an existing flow (menu on version/tab)
- Flows must have a unique name and a description (non-unique)
- From the Flows page:
  - Organize flows by adding them to a Category
  - Control who can access your flow through the sharing settings

4

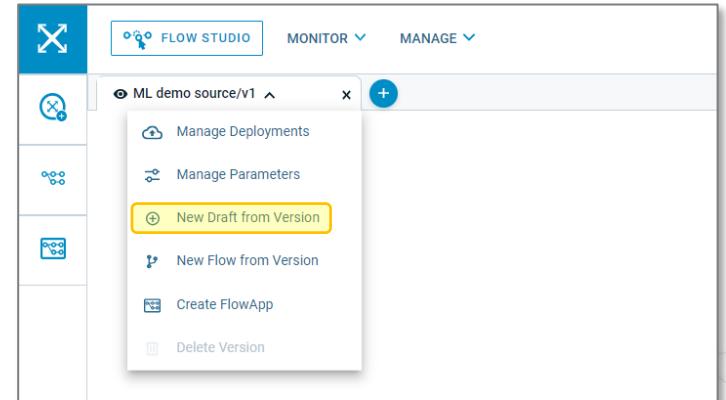
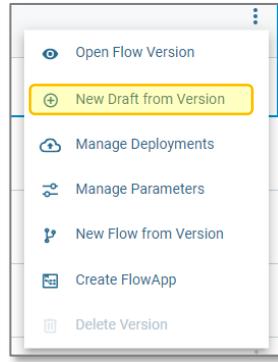
The image displays three screenshots of the Crosser Flow Studio interface. The top screenshot shows the 'Flows' page with a list of flows, including 'y weather forecast' and 'Machine to Cloud'. The middle screenshot shows the Flow Studio workspace with a flow diagram. The bottom screenshot shows the 'Create New Flow' dialog box, which includes fields for 'Basic Information' (name: 'My first Flow', description: 'Just a test'), 'Category' (selected: 'Development'), and 'Share' options (radio buttons for 'Private only to owner', 'Public to my organization', and 'Share with selected users', with 'Owner: Göran Appelquist' listed). The 'CREATE' button is visible at the bottom right of the dialog.

# Flow Versions

- Each flow can have any number of versions
- Versions are deployed to Nodes, not Flows
- Only the latest non-deployed version can be modified
- A version becomes read-only the first time it is deployed on a node
- Create new versions using the “New Draft from Version” action (only available on read-only flows)
- You can also create a new draft from a version inside the Flow Studio (tab menu)

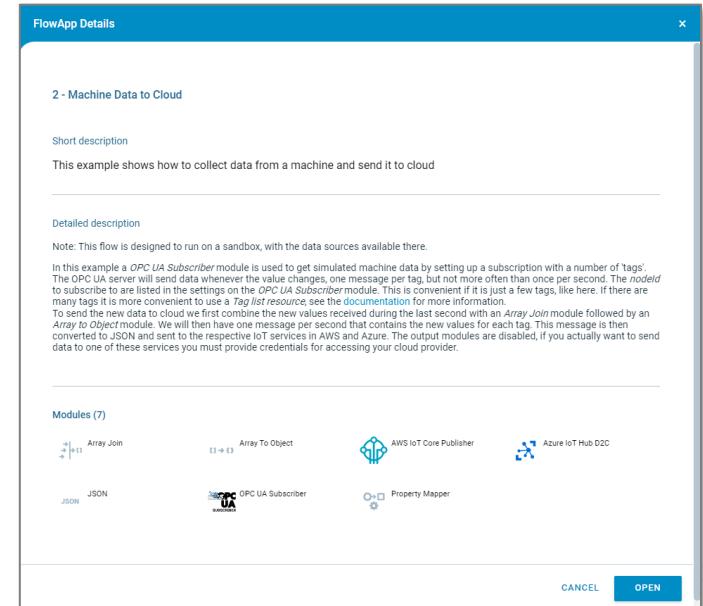
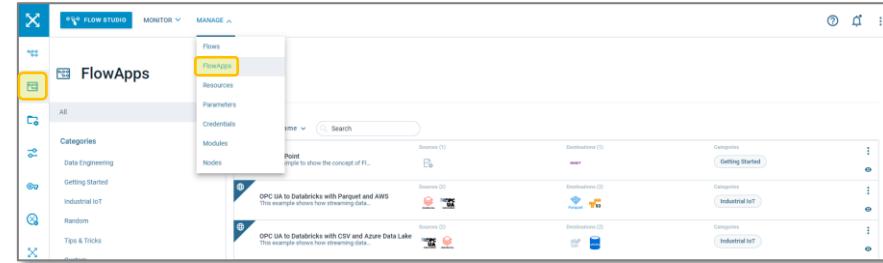
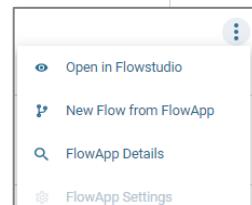
*Note: You will not be able to create new versions of your flows until you can deploy them on your own node in session 5*

Generic Demo 1	Generic Demo 1	2023-07-07 18:30	Mikael Isenberg	0	Category IoT
Blank	Demo	2023-07-07 18:30	Mikael Isenberg	Deployed Nodes	20 modules, 1 Parameter Overrides
Version 7	new draft	2023-08-22 19:27	David Nierhaus	Deployed Nodes	20 modules, 1 Parameter Overrides
Version 6	Generic Demo 1	2023-05-03 08:14	Mikael Isenberg	Deployed Nodes	20 modules, 1 Parameter Overrides



# FlowApps

- Pre-built templates to use as starting points for new flows
  - Created by Crosser
  - Created by your organisation
- Browse to find a FlowApp that is close to what you need
  - Read description
  - Open the flow in Flow Studio
- Create a flow from a FlowApp by:
  - Using the “New Flow from FlowApp” action in the menu
- FlowApps can also be opened from inside the Flow Studio





# THE FLOW STUDIO

This is where you design and test your flows



# The Flow Studio

## Design and Test Your Flows

The screenshot displays the Crosser Flow Studio interface, which is a visual tool for designing and testing data integration flows. The interface is divided into several sections:

- Header:** FLOW STUDIO, MONITOR, MANAGE.
- Left Sidebar:** Module Library with a search bar and categories like Most Used, Triggers, Sources (Cloud, Databases), and specific connectors for Couchbase Executer, Firebird Select, Influx Select, MariaDB Executer, MariaDB Select, MongoDB Select, MsSQL Executer, MsSQL Select, MsSQL Stored Procedure Read, MySQL Executer, MySQL Select, and Oracle Database Select.
- Middle Area:** A visual flow diagram showing the sequence of steps. The flow starts with "Every hour, 10 past" and "Time Now", followed by "Time one year back", "Get Date Range", and "Zoho Desk Search Tickets". This is followed by a loop: "LOOP OVER TICKETS", "Text Template(2)", "Check if Zoho case Number is set in SF", "Create search criteria for Contacts in SF", and "Remove meta data". The flow then continues with "Search Contacts", "Check if case missing the last day of the month", "Text Template", "Array JOIN", "Array Property Set", "JSON", and "Send missing users in SF to Mailbox". Finally, it includes "REMOVE ARRAY", "Dissolve", "Create Case", "Set SFDC ticket in Zoho", and "Zoho Desk Update Ticket".
- Right Sidebar:** External2, Test & Debug (Debug, Connect, SHOW ALL, FREEZE, CLEAR), and a code editor showing the JSON configuration for the "Zoho Desk Search Tickets - Out" step. The code is as follows:

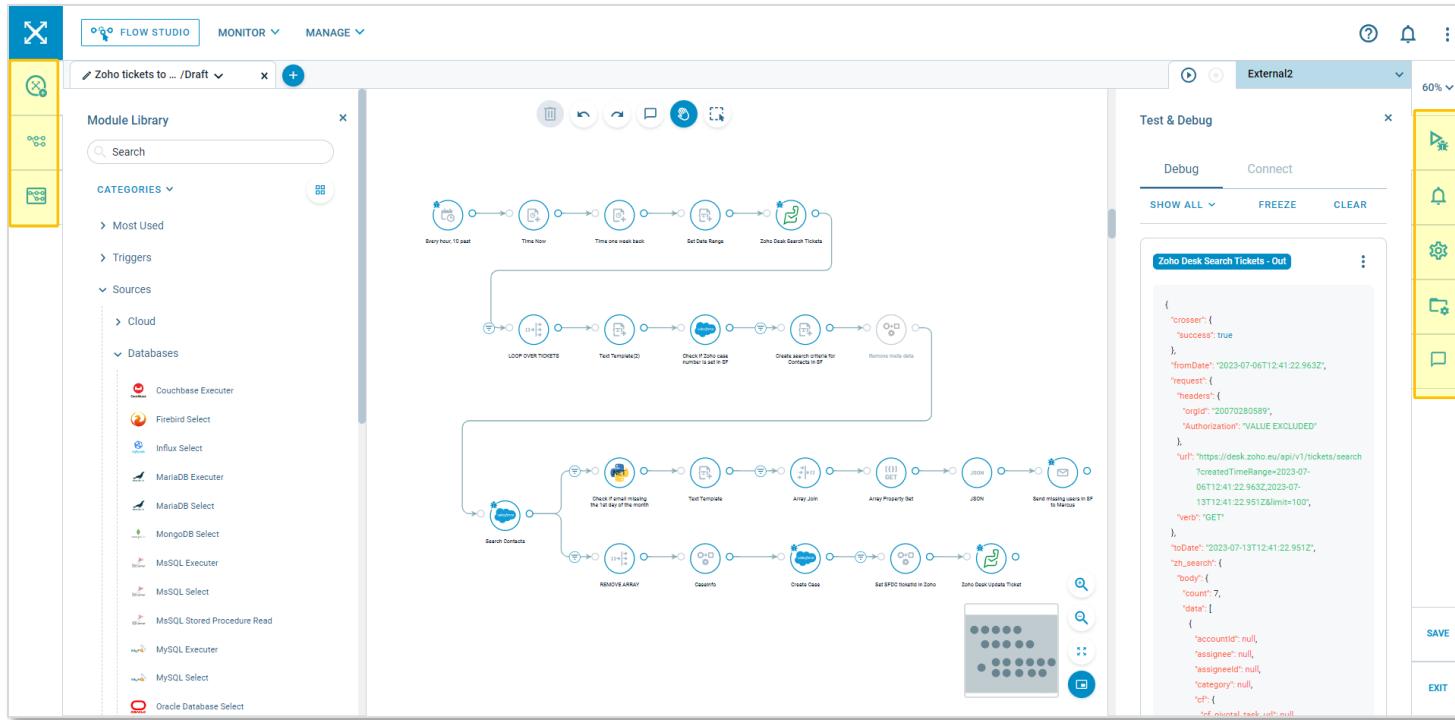
```
{
  "crosser": {
    "success": true
  },
  "fromDate": "2023-07-06T12:41:22.963Z",
  "request": {
    "headers": {
      "orgId": "2007280589",
      "Authorization": "VALUE EXCLUDED"
    },
    "url": "https://desk.zoho.eu/api/v1/tickets/search?createdTimeRange=2023-07-06T12:41:22.963Z,2023-07-13T12:41:22.951Z&limit=100",
    "verb": "GET"
  },
  "toDate": "2023-07-13T12:41:22.951Z",
  "zh_search": {
    "body": {
      "count": 7,
      "data": [
        {
          "accountid": null,
          "assignee": null,
          "assignedid": null,
          "category": null,
          "cf": {
            "id": null
          }
        }
      ]
    }
  }
}
```

Buttons for SAVE and EXIT are also present on the right.

# The Flow Studio

## Side Panels

Module library  
Flows  
FlowApps



Test & Debug  
Notifications  
Flow version settings  
Resources  
Annotations

# The Flow Studio

## Module Library

The screenshot shows the Zoho Flow Studio interface with the following components:

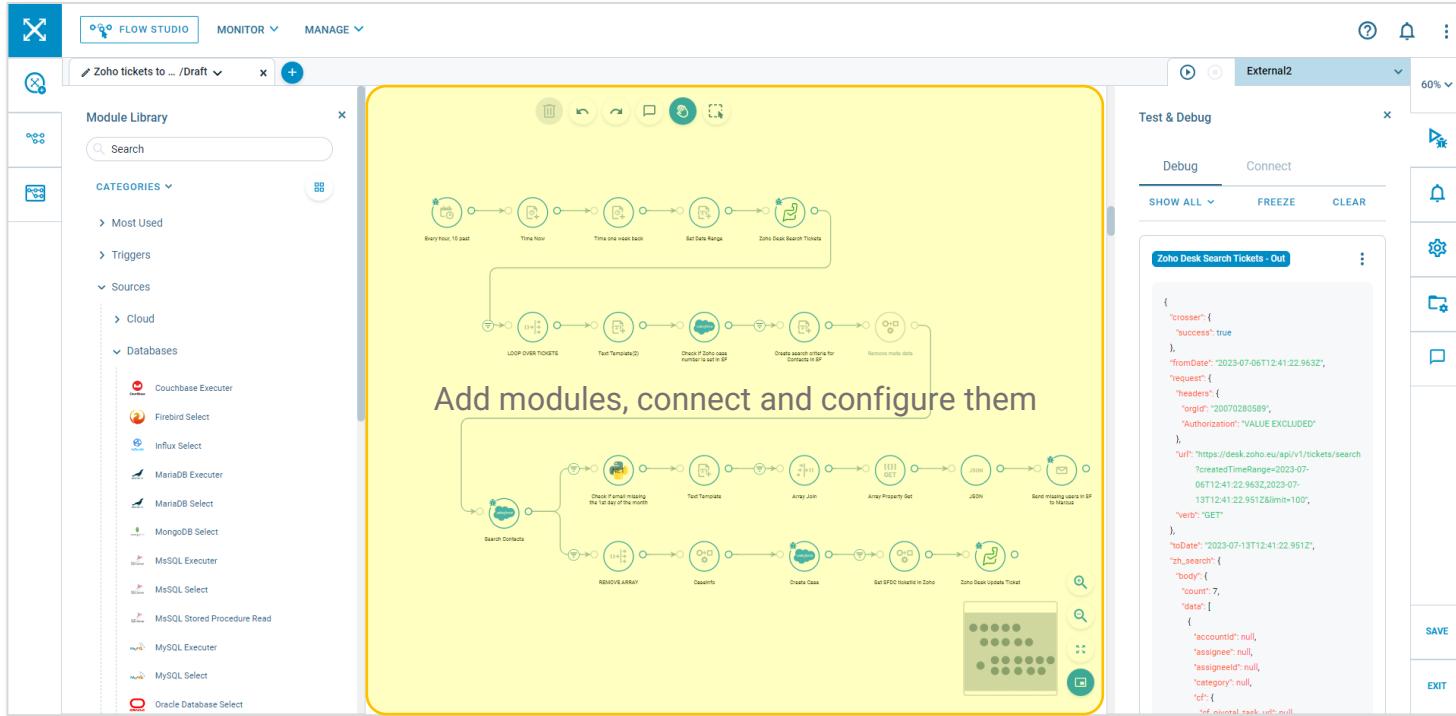
- Header:** FLOW STUDIO, MONITOR, MANAGE.
- Left Sidebar:** Module Library (highlighted with a yellow box), showing a search bar and categories: Most Used, Triggers, Sources (Cloud, Databases). A yellow callout box says "Find the modules you need". Below are lists of modules: Influx Select, MariaDB Executer, MariaDB Select, MongoDB Select, MsSQL Executer, MsSQL Select, MsSQL StoredProcedure Read, MySQL Executer, MySQL Select, Oracle Database Select.
- Workflow Area:** A complex workflow for "Zoho tickets to ... /Draft". It starts with a "Every hour 10 past" trigger, followed by "Time Now", "Time one hour back", "Get Date Range", and "Zoho Desk Search Tickets". This is followed by a "LOOP OVER TICKETS" loop. Inside the loop, it checks if the ticket number is set in ZT, creates search criteria for "Created in ZT", and removes meta data. After the loop, it performs a "Search Contacts" step, then "Check if and missing the 1st day of the month", "Text Template", "Array Join", "Array Priority Set", "JSON", and "Send missing users in EP to Mailbox". Finally, it uses "REMOVE ARRAY", "Cassandra", "Create Data", "Get EPIC Tickets in Zoho", and "Zoho Desk Update Ticket".
- Right Sidebar:** External2, Test & Debug (Debug, Connect, SHOW ALL, FREEZE, CLEAR), showing a log entry for "Zoho Desk Search Tickets - Out". The log entry is as follows:

```
{
  "crossover": {
    "success": true
  },
  "fromDate": "2023-07-06T12:41:22.963Z",
  "request": {
    "headers": {
      "origin": "20070280589",
      "Authorization": "VALUE EXCLUDED"
    },
    "url": "https://desk.zoho.eu/api/v1/tickets/search?createdTimeRange=2023-07-06T12:41:22.963Z,2023-07-13T12:41:22.951Z&limit=100",
    "verb": "GET"
  },
  "toDate": "2023-07-13T12:41:22.951Z",
  "zh_search": {
    "body": {
      "count": 7,
      "data": [
        {
          "accountid": null,
          "assignee": null,
          "assignedt": null,
          "category": null,
          "cf": {
            "id": null,
            "name": null
          }
        }
      ]
    }
  }
}
```

Buttons on the right sidebar include: SAVE, EXIT, and a bell icon.

# The Flow Studio

## Drawing Canvas



# The Flow Studio

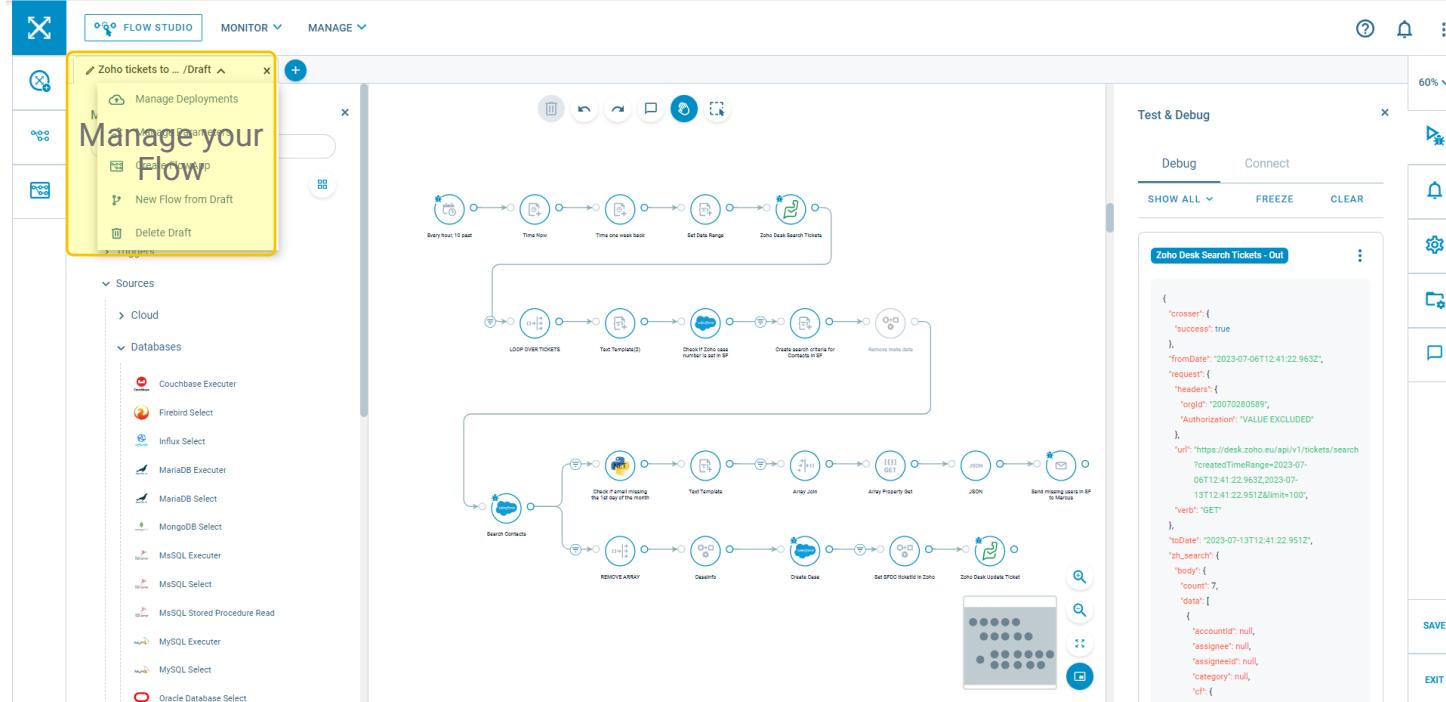
## Test & Debug

The screenshot displays the Crosser Flow Studio interface. On the left, the 'Module Library' sidebar is open, showing categories like 'Most Used' and 'Sources' (Cloud, Databases). The main workspace shows a complex workflow for 'Zoho tickets to ... /Draft'. The flow starts with 'Daily hour: 10 past' and 'Time Now' triggers, followed by 'Time one week back', 'Set Data Range', and 'Zoho Desk Search Tickets'. This leads to a 'LOOP OVER TICKETS' loop, which contains 'Text Terminal(2)', 'Check if 2nd user Purchaser is less than 2', 'Create search criteria for Contacts in ZF', and 'Remove media data'. The loop then branches to 'Search Contacts' (using 'Check if email missing the 1st day of the month' trigger), which leads to 'Text Template', 'Array JSON', 'Array Property Set', 'JSON', and 'Send missing users in ZF to Service'. The 'Text Terminal(2)' step in the loop also has a 'REMOVE ARRAY' action. The 'Zoho Desk Search Tickets' step has a 'Zoho Desk Update Ticket' action. The right side of the interface shows a 'Test & Debug' panel for the 'External2' connection, displaying the output of the 'Zoho Desk Search Tickets - Out' step, which is a JSON response from the Zoho Desk API.

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "crossover": {
    "success": true
  },
  "from": "2023-07-13T12:41:22.951Z",
  "request": {
    "headers": {
      "Content-Type": "application/json",
      "Authorization": "VALUE EXCLUDED"
    },
    "url": "https://desk.zoho.eu/api/v1/tickets/search?createdTimeRange=2023-07-05T12:41:22.953Z,2023-07-13T12:41:22.951Z&limit=100",
    "verb": "GET"
  },
  "toDate": "2023-07-13T12:41:22.951Z",
  "z_deserch": {
    "body": {
      "count": 7,
      "data": [
        {
          "accountId": null,
          "assignee": null,
          "assignedId": null,
          "category": null,
          "cf": {
            "id": "10000000000000000000000000000000"
          }
        }
      ]
    }
  }
}
```

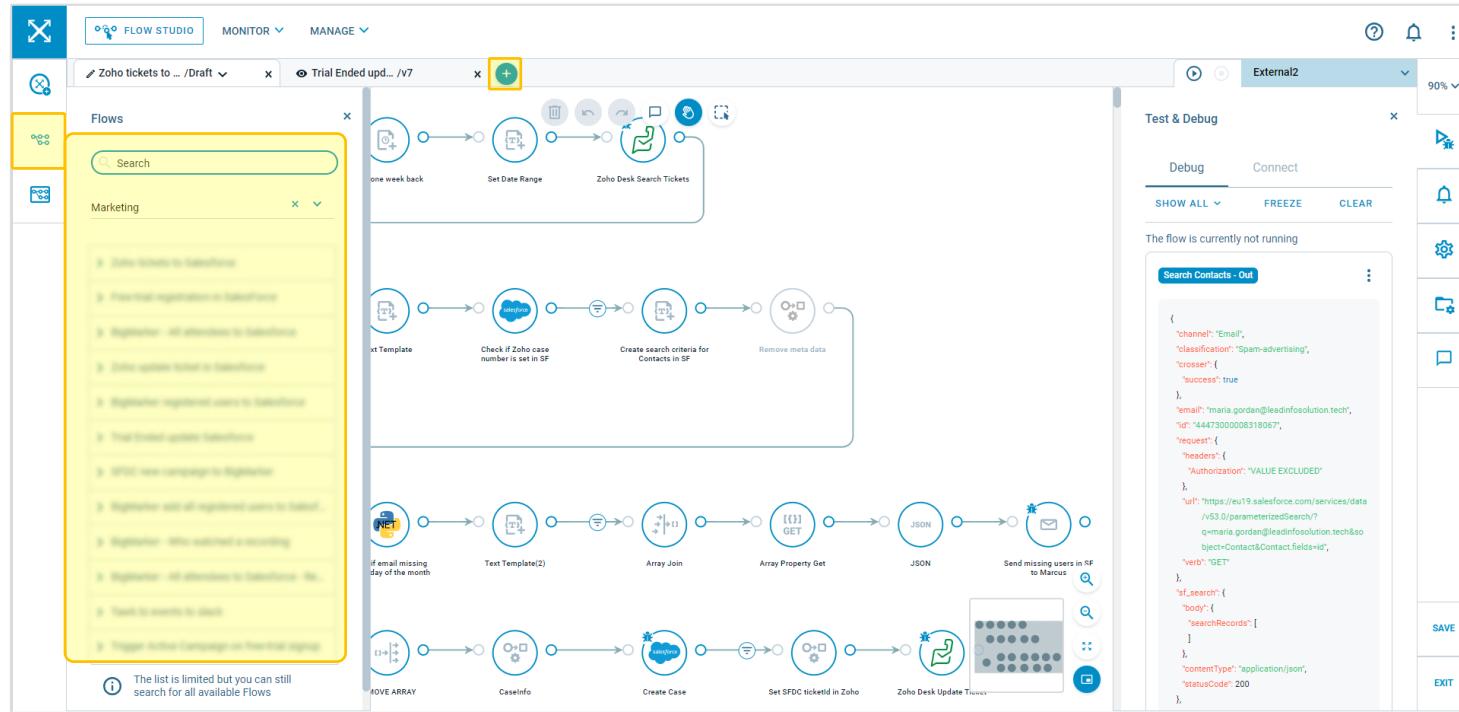
# The Flow Studio

## Manage Your Flows



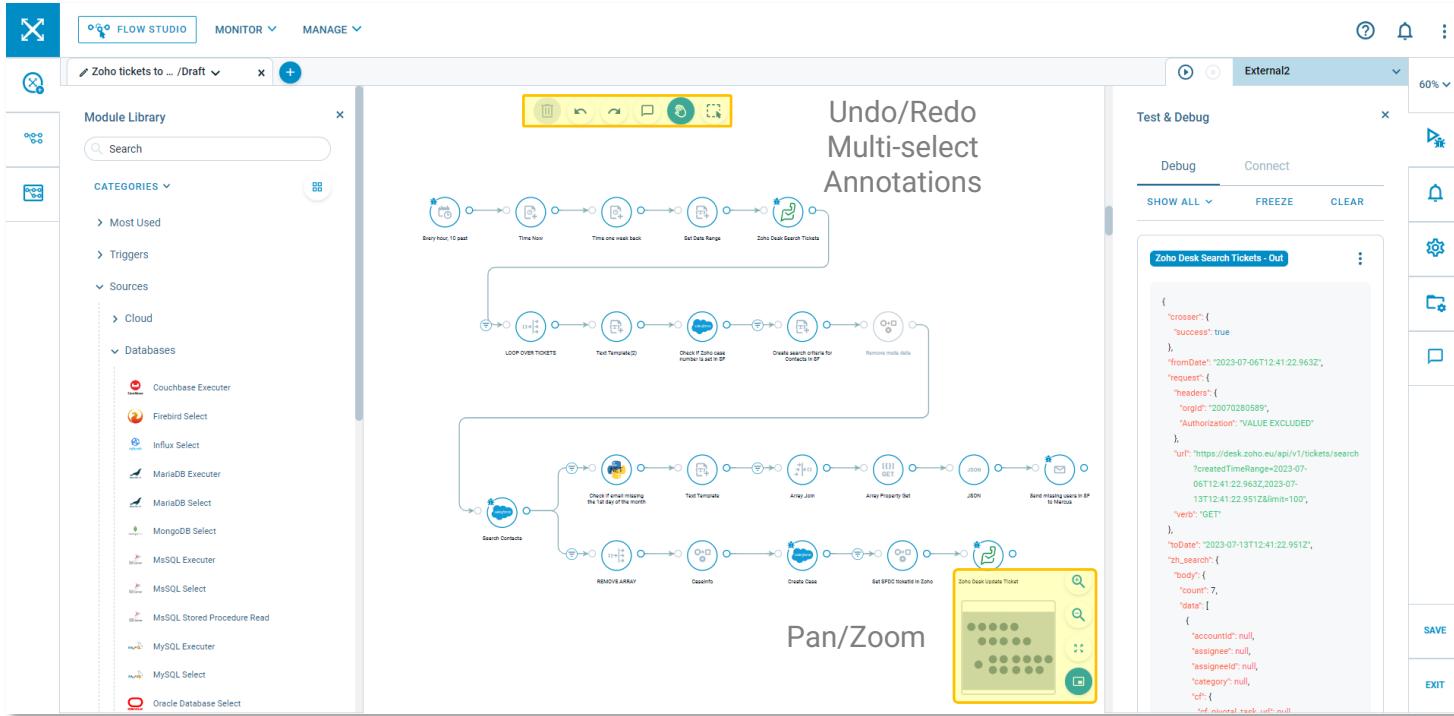
# The Flow Studio

## Create New Flows or Open Existing



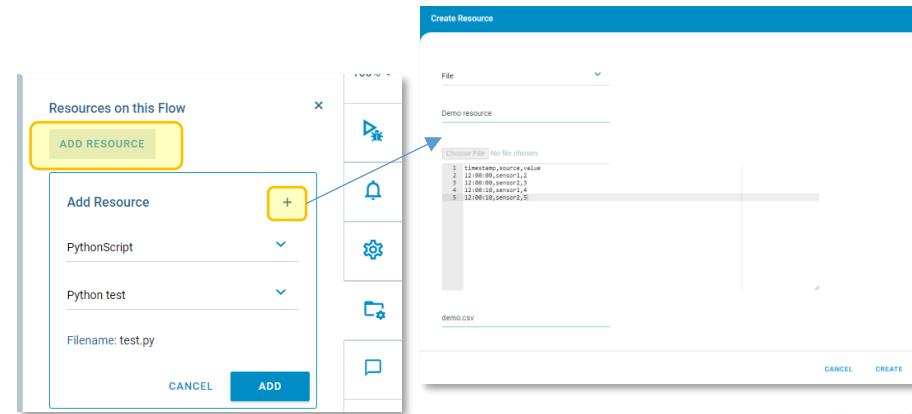
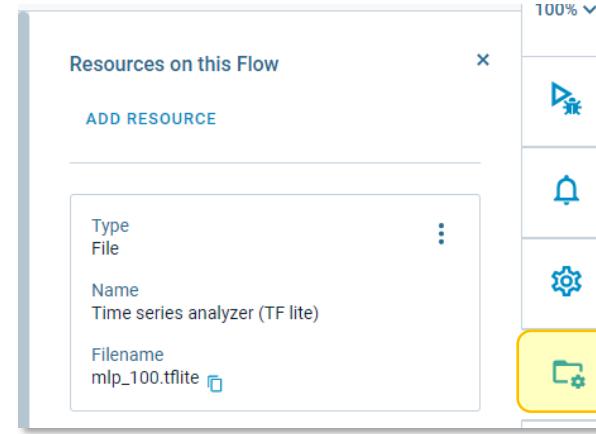
# The Flow Studio

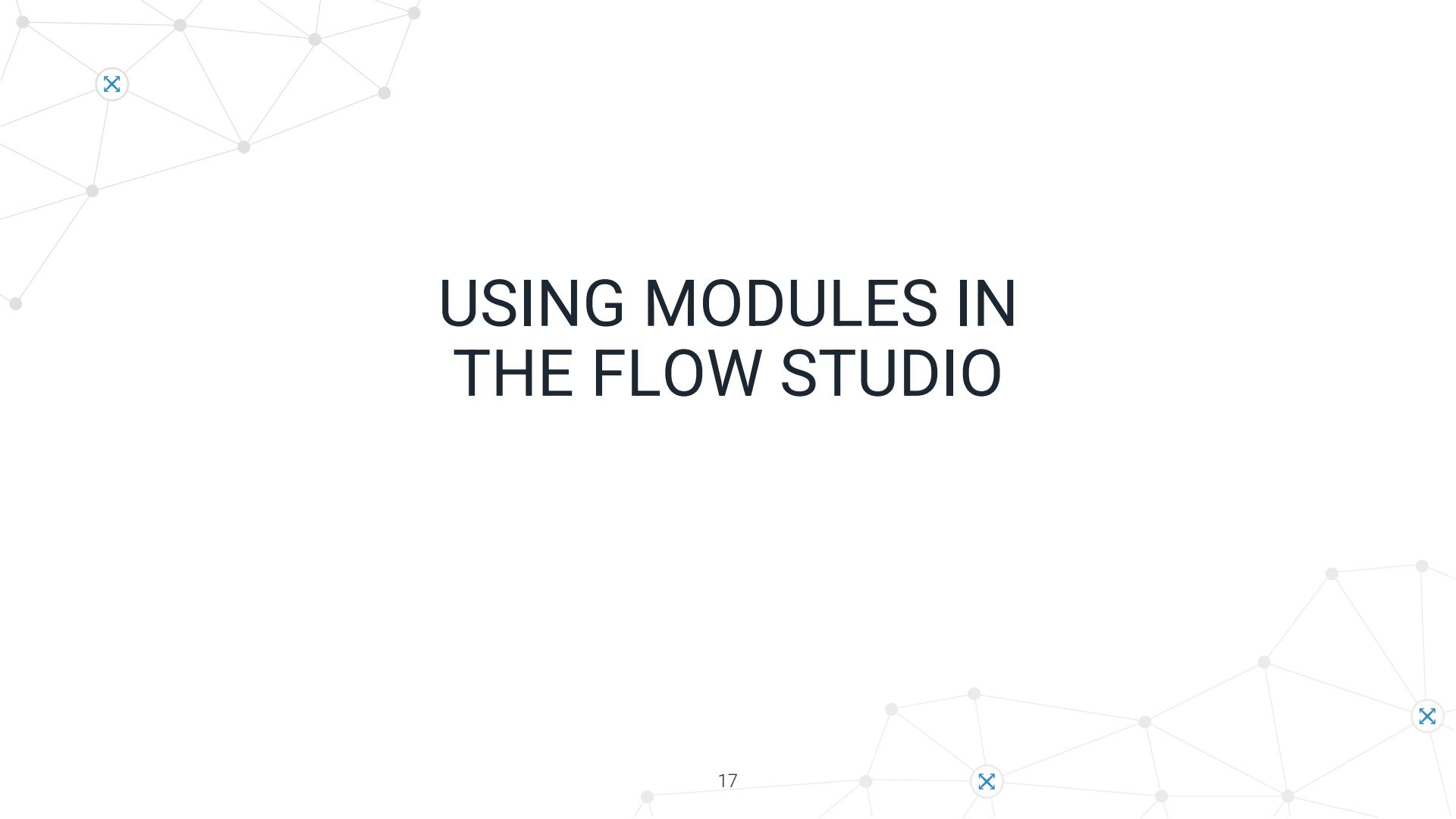
## Visual Appearance



# Resources

- Add resources from the library that your flow needs:
  - Files
  - ML models
- Create new resources by uploading files or enter data in the editor
- Resources added to a flow will be downloaded by nodes when the flow is deployed
- Note: PLC resources are added from the respective module settings



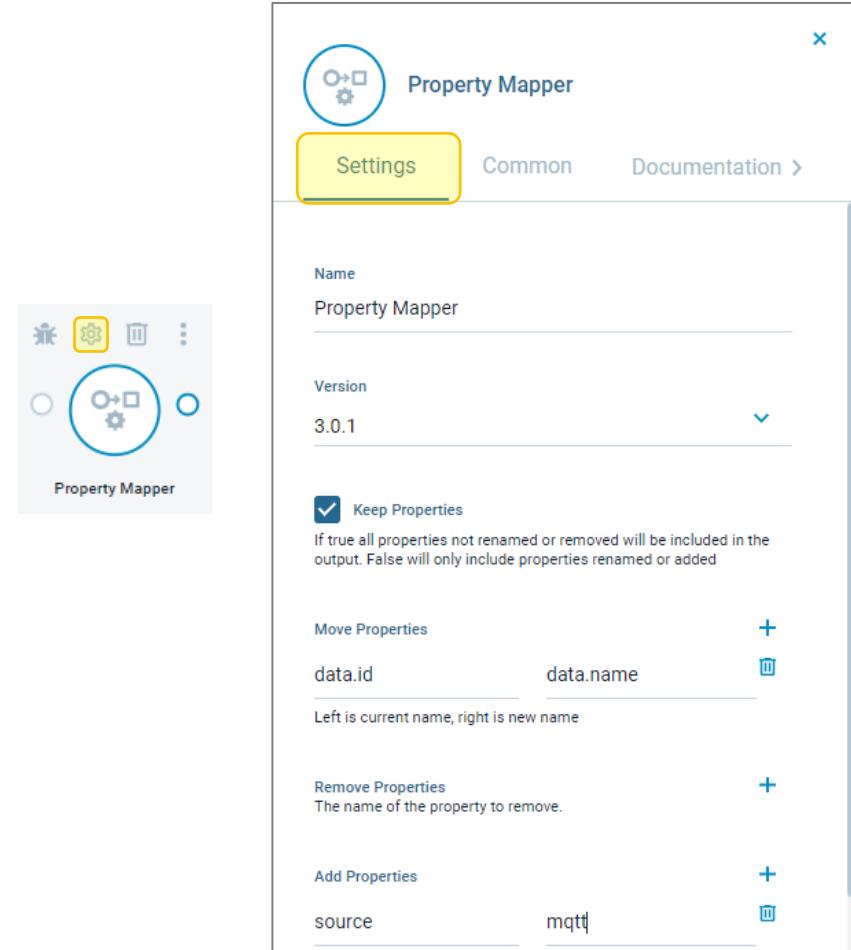


# USING MODULES IN THE FLOW STUDIO

# Modules

## Settings tab

- The *Settings* tab shows module specific settings
  - Any setting called something with *Property* references messages. Which data to use from the incoming message (typically called *Source* or *Value* properties) and where to add the result on the output message (*Target* property)
  - I/O modules have settings to specify how to access external systems
  - Analytics modules have settings to control the processing of the selected data
- Modules have versions:
  - When dragging a module from the library the latest version is used
  - Multiple instances of the same module in a flow must have the same version
  - Different flows can use different versions of a module
  - The module and versions available depends on the Node version you are targeting (advanced feature)



The screenshot shows the 'Property Mapper' settings tab within a configuration interface. The tab is highlighted with a yellow border. The interface includes a navigation bar with 'Settings' (highlighted), 'Common', and 'Documentation >'. Below the navigation is a 'Name' field containing 'Property Mapper'. A 'Version' field shows '3.0.1'. A checked checkbox labeled 'Keep Properties' has a tooltip: 'If true all properties not renamed or removed will be included in the output. False will only include properties renamed or added'. The 'Move Properties' section shows a mapping from 'data.id' to 'data.name' with a note: 'Left is current name, right is new name'. The 'Remove Properties' section has a placeholder: 'The name of the property to remove.' The 'Add Properties' section shows a mapping from 'source' to 'mqqt'.

# Modules

## Common tab

- The same on all modules
- Flow behavior settings
  - Halt On Error
    - If enabled the flow will stop on module errors
    - If disabled, the behavior is controlled by the flow setting
  - Debug
    - Enable debug output from this module when testing (same as debug action on the module icon)
  - Disabled
    - “Turn off” a module, e.g. to prevent data to be sent externally during testing
- Message filters
  - Without filters all messages will be processed
  - With filters only selected messages will be processed
  - Messages not matching the filter can either be ignored or bypassed to the next module (“Bypass...” setting)

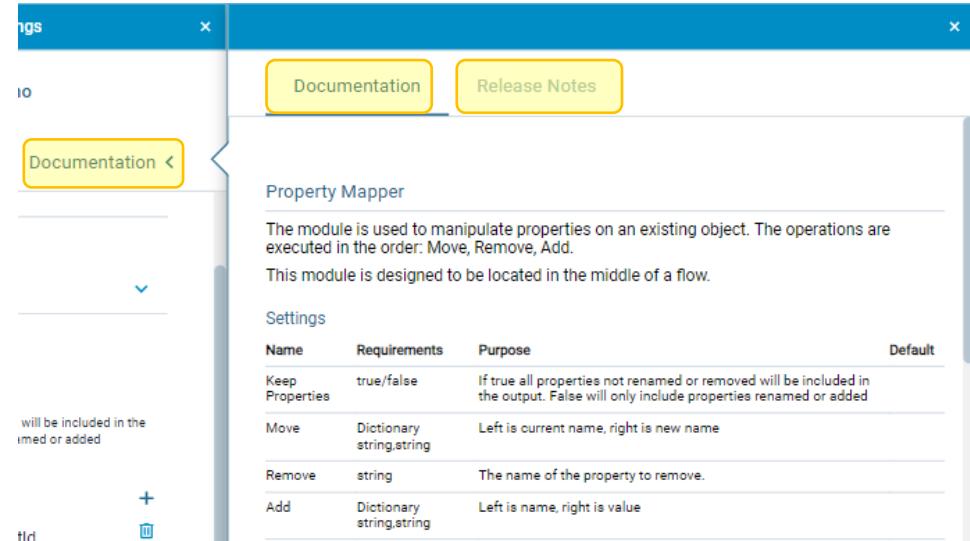
*Note: Queue/Retry settings will not be covered in the Fundamentals course*

The screenshot shows the 'Property Mapper' interface with the 'Common' tab selected. The 'Common' tab is highlighted with a yellow background. Below the tabs, there are three settings with checkboxes: 'Halt On Error', 'Debug', and 'Disabled'. Under 'Message Filters', there is a list with a single entry: 'data.temp, Greater Than, 50, Number'. A note below the filter says 'Messages matching all criterias will be able to enter the module'. At the bottom, there is a checkbox for 'Bypass message to next module in flow if filters does not match'.

# Modules

## Documentation tab

- Module documentation
  - General description
  - Settings
  - Message requirements
  - Examples
- Release Notes
  - Describes changes in module versions



Documentation

Release Notes

Property Mapper

The module is used to manipulate properties on an existing object. The operations are executed in the order: Move, Remove, Add.

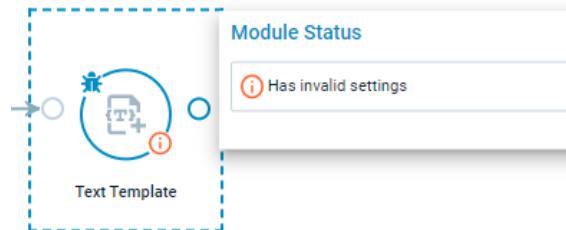
This module is designed to be located in the middle of a flow.

Name	Requirements	Purpose	Default
Keep Properties	true/false	If true all properties not renamed or removed will be included in the output. False will only include properties renamed or added	
Move	Dictionary<string, string>	Left is current name, right is new name	
Remove	string	The name of the property to remove.	
Add	Dictionary<string, string>	Left is name, right is value	

# Modules

## Settings validations

- Settings entered in modules are validated
- Invalid or incomplete settings:
  - Are indicated on the module in the canvas
  - In the Settings panel for the module



A screenshot of a module settings panel for 'Text Template'. The 'Settings' tab is selected. A red box highlights the message 'Has invalid settings, see red marked fields'. The 'Name' field contains 'Text Template'. The 'Version' dropdown is set to '3.1.2-preview001'. The 'Target Property' field is empty and has a red validation message: 'must NOT have fewer than 1 characters'.



# TESTING YOUR FLOW

In the Flow Studio  
Deploy to a Node



22

# Test Flow

## Connect to a Node

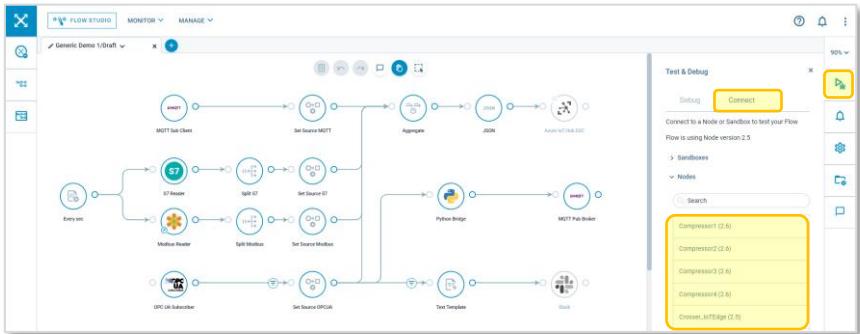
- Connect to a node to use for testing
  - **Sandbox**: Node hosted by Crosser. Can be used with flows that don't need access to local data sources. Only available in the Flow Studio
  - **Local nodes**: Nodes running on your infrastructure

*Note: Exercises can be tested on a Sandbox, if not explicitly referring to your local node*

### Sandbox



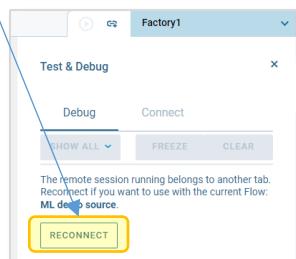
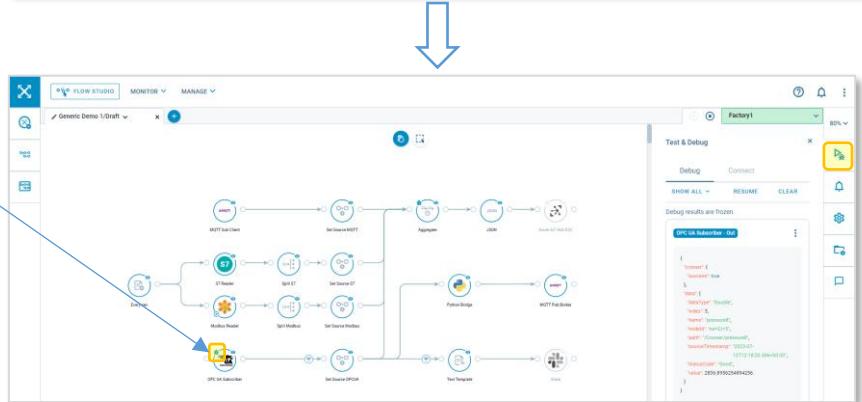
### Local Nodes



# Test Flow

## Run Your Flow and Check Results

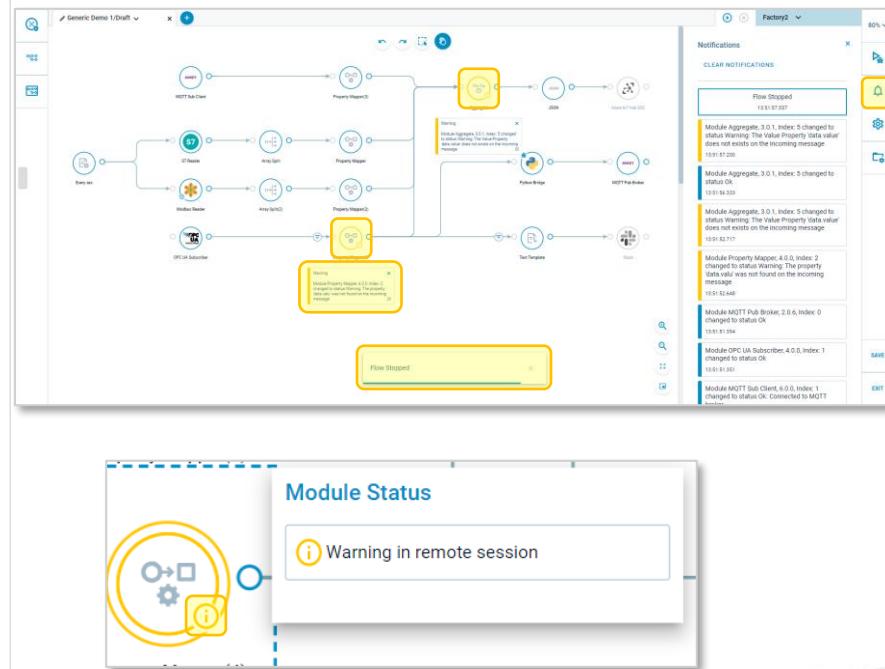
- Start (and Stop) the flow
  - The flow must be saved before a remote session can be started. This can be done automatically, if you chose to enable that in the Save dialog
- Enable debugging on modules to see output in the Debug window (hover and use the debug action)
- A remote session is connected to a specific Flow version. You need to *Reconnect* to test another Flow version.
- The Flow is removed from the Node when you leave the Flow Studio



# Test Flows

## Notifications

- Notifications from the Flow will show up as toasters (bottom of screen):
  - Blue: Information and status
  - Yellow: Warnings
  - Red: Errors (flow will stop)
- Errors and Warnings from modules will be added on the module causing the event, as a colored ring, with the message below (large messages can be opened in a separate window)
- Notifications are also added to the **Notifications** tab and to the notification list on modules
- When the flow is deployed notifications will be sent to the Event log in the Control Center

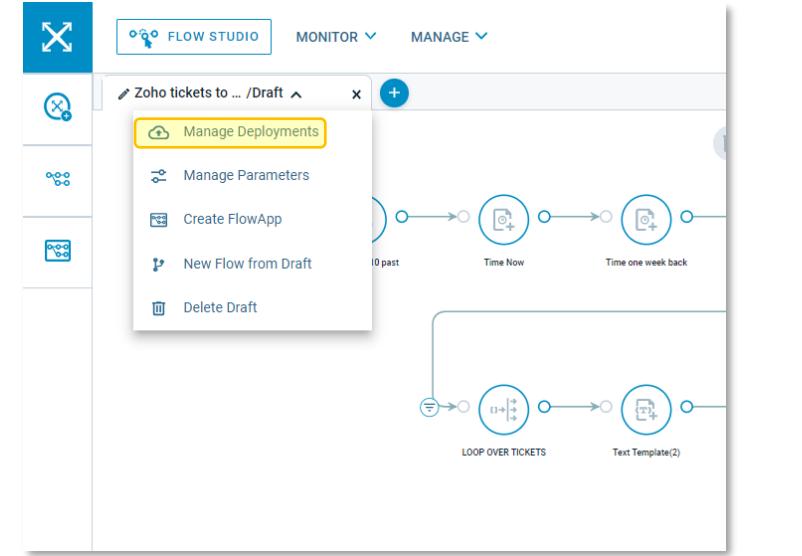


# Deploy Flows

## Install permanently

Open the [Deployment tool](#)

- In the Flow Studio:
  - Select the [Manage Deployments](#) action in the tab menu
- On the Flows page:
  - Expand a flow to see the versions
  - Select the [Manage Deployments](#) action in the menu, or click in the [Deployments](#) column



The screenshot shows the Zoho Flows page. On the left, there's a sidebar with categories like 'All', 'Favorites', 'Deployed Flows', 'Drafts', 'Categories', 'Development', and 'INT'. The 'INT' category is selected. The main area shows a table of flows, with one row expanded to show its details. The expanded row for 'Generic Demo 1' shows several versions, with the first one being the 'Latest' version. On the right side of the table, there's a context menu for the 'Latest' version, with the 'Manage Deployments' option highlighted in yellow. Other options in the menu include 'Edit Draft', 'Edit Description', 'Manage Parameters', 'New Flow from Draft', 'Create FlowApp', and 'Delete Draft'.

Flow	Deployment	Category
Generic Demo 1	Latest	Webhooks
Copy of 33 - Copy of 32 - Copy of 30 - identity interval	2020-01-13 16:46	Webhooks
Copy of 32 - Copy of 30 - identity interval	2020-01-13 16:46	Webhooks
Copy of 30 - identity interval	2020-01-13 16:46	Webhooks
identity interval	2020-01-13 16:46	Webhooks
interval change	2020-01-13 16:46	Webhooks

# Deploy Flows

## Select Node(s)

In the Deployment tool

- In the list of available Nodes, selects the ones where you want to install the Flow
- If the Node is currently offline, it will install the Flow the next time it comes online
- Click on Deploy This Version

*Note 1: After a Flow has been deployed to Nodes, processing will run locally on those Nodes without any dependencies on the Crosser Control Center service*

*Note 2: Deploying a Flow will make it read-only. To update, create a new version*

*Note 3: You will not be able to deploy a flow until session 5, where you will install your own node*

Node	Node Status	Labels	Flow Version deployed
Compressor1	online	Stockholm, Demo, All, Compressor	-
Compressor2	online	Demo, Sundsvall, All, Compressor	-
Compressor3	online	Azure, Demo, Sundsvall, All, Compressor	-
Compressor4	online	Demo, Sundsvall, All, Compressor	-
Crosser_IoTEdge	online	mynewlabel	-



# MODULES USED IN EXERCISES



Data Generator

Aggregate

Text Template

SMTP Send

# MODULE

## Data generator

- Create test data (from JSON template)
- Useful for testing a flow without using external data sources
- Multiple “Samples” can be generated, e.g. to simulate multiple sensors or sources
- Enter a JSON template with the message structure you want to generate → Click “Update”
- In the “Data Rules” section, specify how you want to generate data for each property in the template

Integers/GUIDs can be:

- Static → Value from template
- Key → Value unique per sample
- Random

Text can be:

- Static → Value from template
- Identifier → The template value plus a key added per sample

Double/Booleans can be Static or Random

- Change “Output Strategy” to generate an ordered sequence of IDs, or an array of samples

Sample interval: 1 Seconds

Number of Samples: 10

Add timestamp to each sample:

Output Strategy: Random

JSON:

```
1 [ { 2 "id": 1, 3 "name": "machine", 4 "data": { 5 "temp": 12, 6 "pressure": 489 7 } 8 } ]
```

ADD EXAMPLE FORMAT JSON UPDATE

Data Rules:

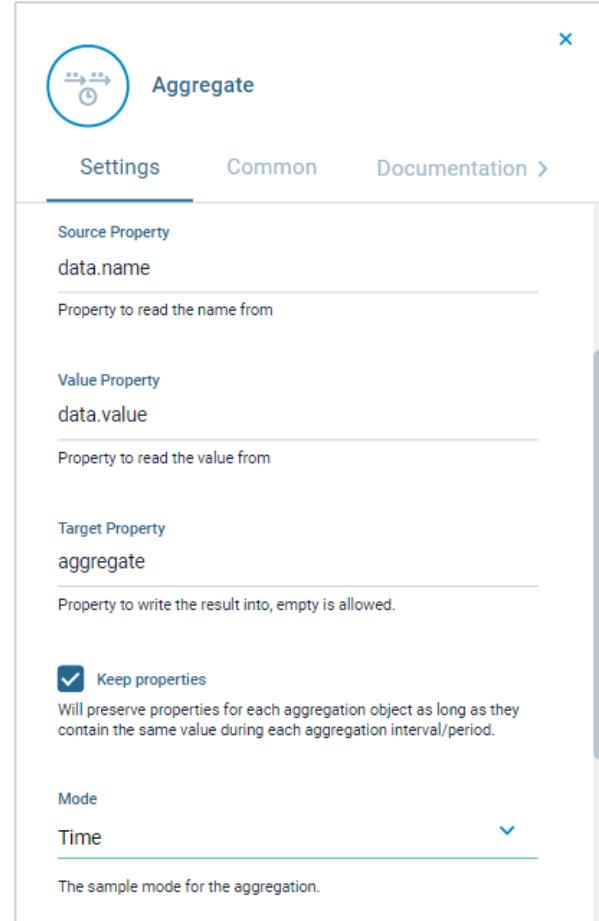
- > id, Key, Integer
- > name, Static, Text
- > data.temp, Random, Integer, 11, 100
- > data.pressure, Random, Integer, 101, 1000

# Module

## Aggregate

- The Aggregate module is used to aggregate values:
  - Over a time window
  - Over a certain number of messages
  - An output is generated when the end of the aggregation window is reached
- Calculates Average, Min and Max values
- Can group messages by source, i.e. aggregate data from multiple sources

*Note: Use this module with streaming data. If you want to operate on data in an array, use the Array Statistics module.*



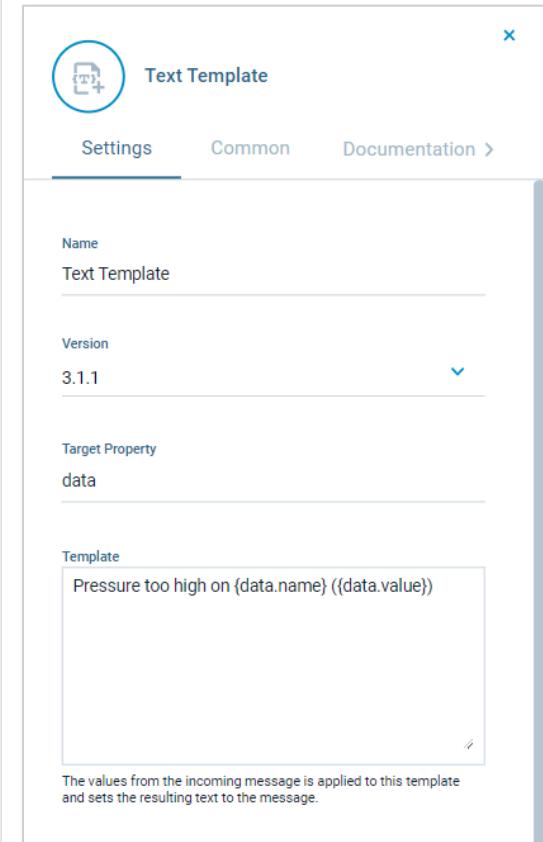
The screenshot shows the configuration interface for the Aggregate module. At the top, there is a circular icon with two arrows pointing in opposite directions and a clock, labeled 'Aggregate'. Below the icon are three tabs: 'Settings' (which is selected), 'Common', and 'Documentation >'. The 'Settings' tab contains the following configuration options:

- Source Property:** `data.name`  
Property to read the name from
- Value Property:** `data.value`  
Property to read the value from
- Target Property:** `aggregate`  
Property to write the result into, empty is allowed.
- Keep properties:**  A checked checkbox with the label 'Keep properties' and a descriptive text: 'Will preserve properties for each aggregation object as long as they contain the same value during each aggregation interval/period.'
- Mode:** `Time`  
The sample mode for the aggregation.

# Module

## Text Template

- Create text strings based on a template where data from the flow message can be inserted
- Anything inside {} is replaced with the value from the corresponding property on the incoming message
- Use it to:
  - Add/append text to messages
  - Convert numeric values to strings

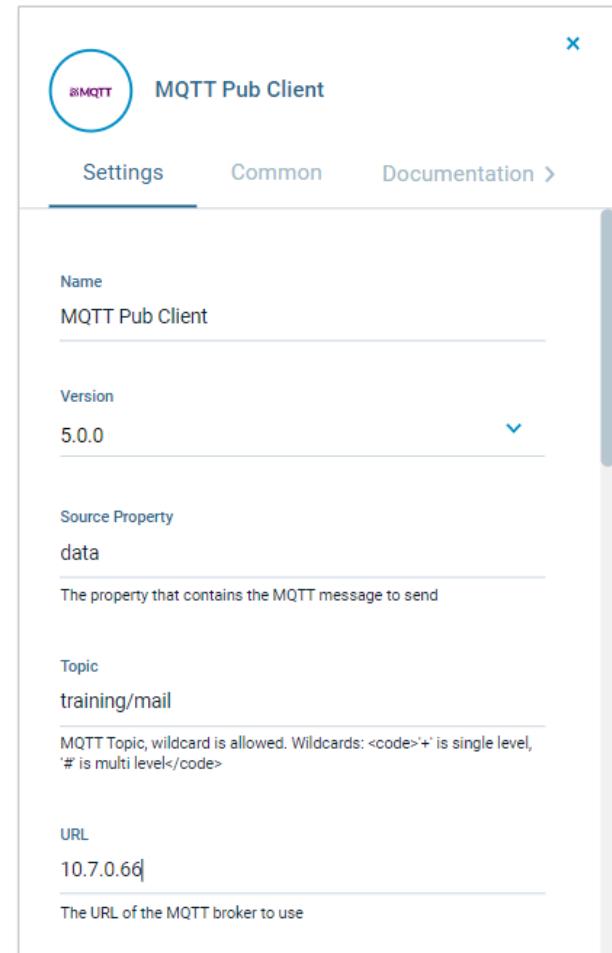


The screenshot shows the 'Text Template' configuration interface. At the top, there is a header with a 'Text Template' icon, the title 'Text Template', and tabs for 'Settings', 'Common', and 'Documentation >'. The 'Settings' tab is selected. Below the tabs, there are three sections: 'Name' (set to 'Text Template'), 'Version' (set to '3.1.1'), and 'Target Property' (set to 'data'). In the 'Template' section, the text 'Pressure too high on {data.name} {{data.value}}' is displayed in a code editor-like area. A note at the bottom states: 'The values from the incoming message is applied to this template and sets the resulting text to the message.'

# Module

## MQTT Pub Client

- Publish message data to an external MQTT broker
- Default settings will convert message to JSON
  - Allows complete message structures to be transferred
  - XML and Raw formats also available
- Required settings:
  - Source Property: Which part of the message to send
  - Topic: Which topic to use when sending the data
  - URL: The IP address or hostname of the external broker



The screenshot shows a configuration interface for the 'MQTT Pub Client' module. The top navigation bar includes a MQTT icon, the module name 'MQTT Pub Client', and tabs for 'Settings' (which is selected), 'Common', and 'Documentation >'. The 'Settings' tab contains the following configuration fields:

Name	MQTT Pub Client
Version	5.0.0
Source Property	data
Topic	training/mail
URL	10.7.0.66

Below the configuration fields, there is a note: 'MQTT Topic, wildcard is allowed. Wildcards: <code>+</code> is single level, <code>#</code> is multi level'.

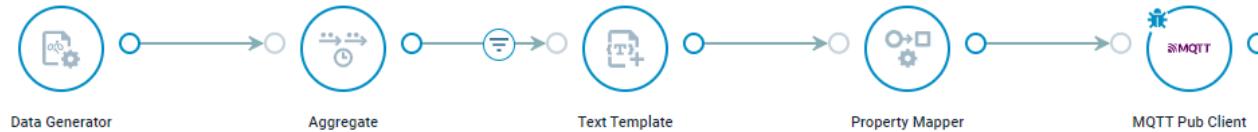


# EXERCISES



# Exercise 1

## Overview



- You will build a flow step by step, starting with an internal data source
- You will see how we can process this data (aggregate)
- Use message filters
- Finally, you should get an email from your flow

# Exercise 1A

## Your very first flow!



Data Generator

```
{  
  "name": "machine",  
  "data": {  
    "temp": 12,  
    "pressure": 489  
  }  
}
```

1. Login to Crosser Cloud (<https://cloud.crosser.io>)
2. On the Flows page, create a new flow called **Exercise 1**
3. Add a **Data Generator** module with default settings
  - Click on 'ADD EXAMPLE' to add the default template
4. Start an interactive debug session on the sandbox node
  - Click on 'Connect Node' and go to the 'Sandboxes' tab and click on 'Connect'
5. Start the flow, turn on debugging on the module and watch the output in the debug window
6. Change settings on the 'Data Generator':
  - Change *Number of Samples* to **3**
  - In the 'Data Rules' section at the bottom, set the behaviour on the *name* property to **Identifier**
7. Run the flow again and check the output

# Exercise 1B

## Aggregation

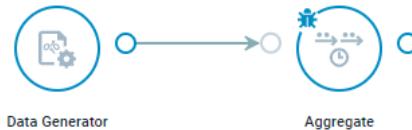


---

1. Add an **Aggregate** module
  - Value Property: `data.temp`
  - Target Property: `aggregate`
  - Interval: `10 seconds`
2. Run the flow and check the output from the 'Aggregate' module

# Exercise 1C

## Grouping



---

1. In the 'Aggregate' module, add `name` as the *Source Property*
2. Run the flow and check the output from the 'Aggregate' module
3. What is the difference?

# Exercise 1D

## Creating a text message



1. Add a **Text Template** module after the 'Aggregate' module:
  - Target Property: `data.content`
  - Template: `The max value is {aggregate.max}`
2. Run the flow on your node and check the output

# Exercise 1E

## Message filters

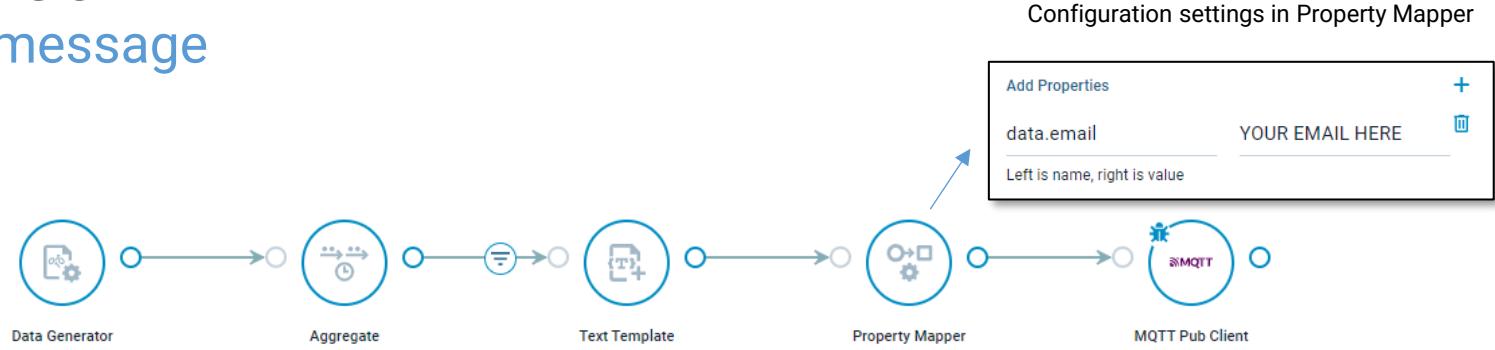


---

1. In the 'Text Template' module open the *Common* tab and specify a filter that selects messages where the *name* property is **machine-1**
2. Run the flow on your edge node and notify the difference

# Exercise 1F

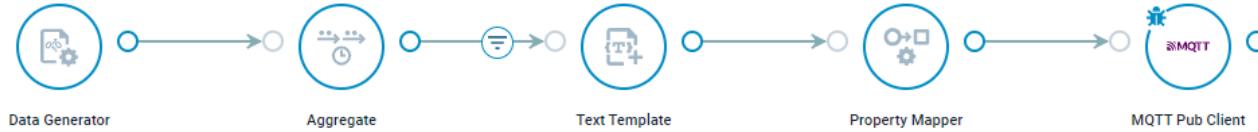
## Send a message



1. Add a **Property Mapper** module and add the settings shown above
  - The Property Mapper will be covered in the next session
2. Add a **MQTT Pub Client** module:
  - Topic: [training/mail](#)
  - MQTT Brokers: [10.0.48.117:1883](#)
3. Run the flow and you should receive an email!
  - You cannot send more than one email per minute

# Exercise 1

## Wrap-up



### Things to test/consider:

- Try changing some settings in the Data Generator and notify the changes in the output
- Why did the number of output messages change when you added the 'Source Property' in the Aggregate module?
- Try some other filters in the Text Template module, like only letting through values in a range
- Try creating some other messages in the Text Template module, e.g. specifying the range of values (using min/max from the Aggregate module)



# SESSION – 02 END

How to build a new flow

How to work with modules

Testing flows in the FlowStudio