



SESSION 02

Edge Analytics Online Training

CREATE YOUR FIRST FLOW

Work with modules and flows

Session 2

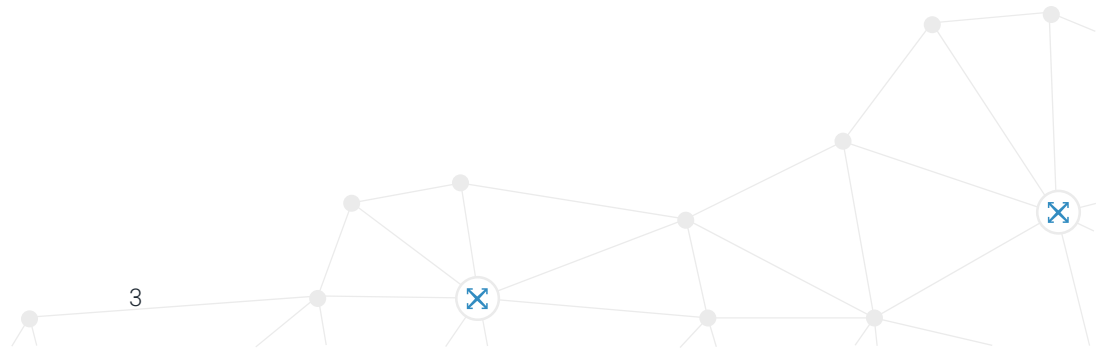
Agenda

- 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



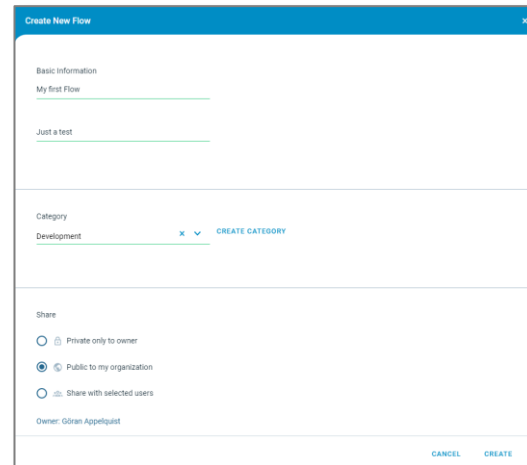
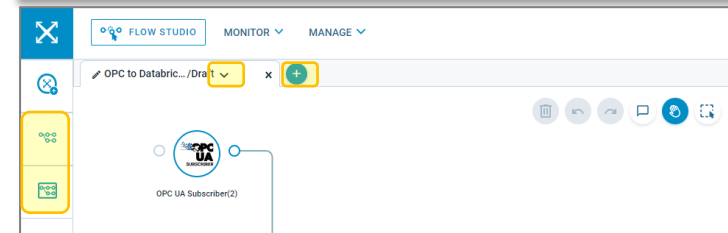
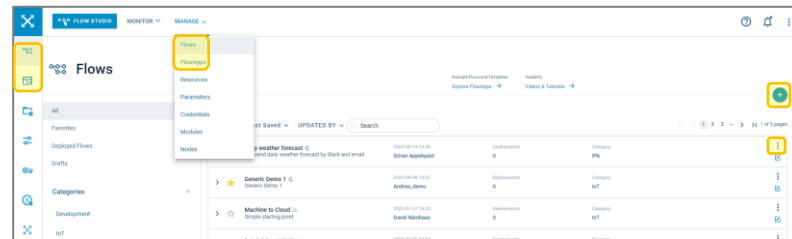
FLOWS

This is what You create with Crosser



Flows

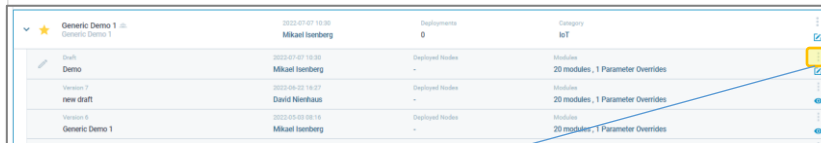
- Created on the [Flows page](#) or from within the [Flow Studio](#):
 - Using the “+” button (empty flow)
 - From a FlowApp
 - From an existing flow (menu on version/tab)
- Flows must have a unique name and a description (non-unique)
- From the Flows page:
 - Organize flows by adding them to a Category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



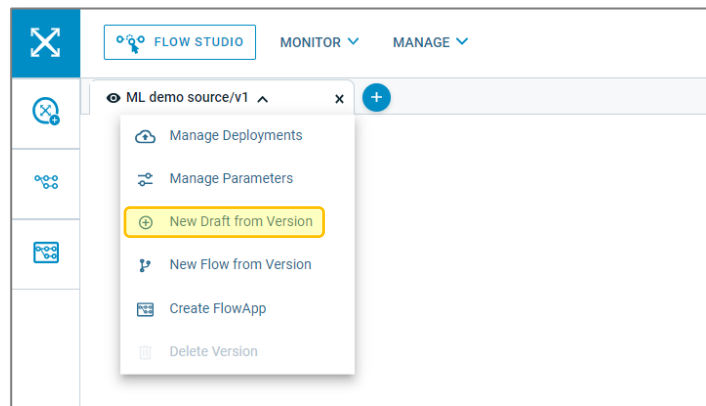
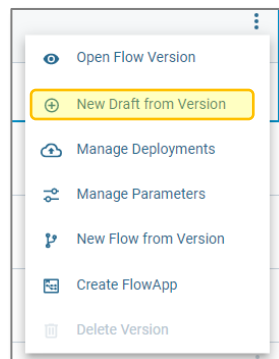
Flow Versions

- 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

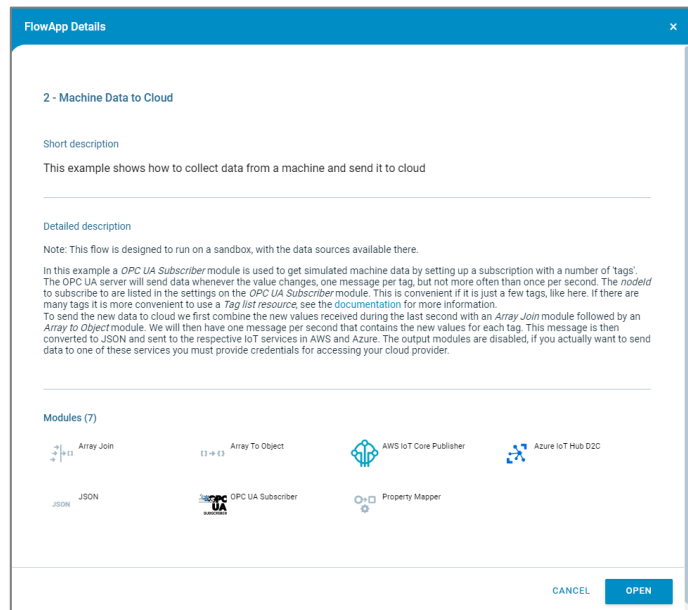
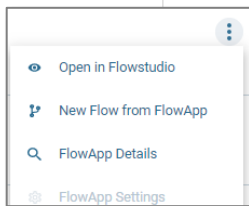
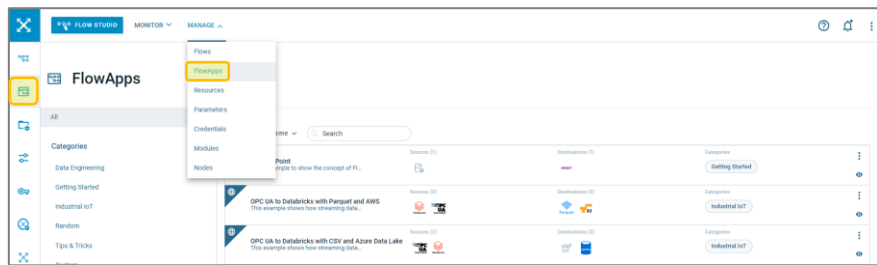


	Generic Demo 1	2023-07-07 10:36	Mikael Isenberg	Deployments	0	Category	IoT
Draft	Draft	2023-07-07 10:36	Mikael Isenberg	Deployed Nodes	-	Module	20 modules, 1 Parameter Overrides
Version 7	new draft	2023-06-25 16:27	David Nordhaus	Deployed Nodes	-	Module	20 modules, 1 Parameter Overrides
Version 6	Generic Demo 1	2023-07-03 10:18	Mikael Isenberg	Deployed Nodes	-	Module	20 modules, 1 Parameter Overrides



FlowApps

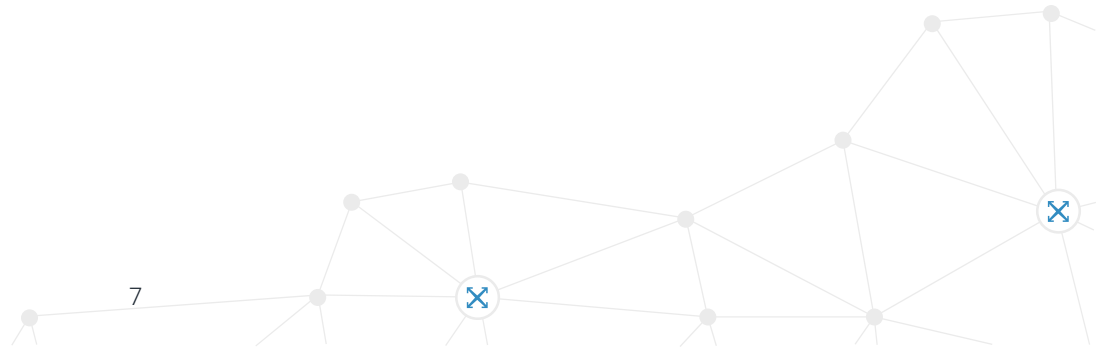
- Pre-built templates to use as starting points for new flows
 - Created by Crosser
 - Created by your organisation
- Browse to find a FlowApp that is close to what you need
 - Read description
 - Open the flow in Flow Studio
- Create a flow from a FlowApp by:
 - Using the “New Flow from FlowApp” action 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 Flow Studio interface with a flow titled "Zoho tickets to ... /Draft". The flow is designed to fetch Zoho Desk search tickets and update them in a database. The flow starts with a "Every hour, 15 past" trigger, followed by "Time Now", "Time one year back", "Set Date Range", and "Zoho Desk Search Tickets". The flow then branches into two parallel paths. The top path includes "LOOP OVER TICKETS", "Text Template()", "Check if Zoho case number is set in SP", "Create search criteria for Contacts in SP", and "Remove extra data". The bottom path includes "Search Contacts", "Check if email missing the 1st step of the month", "Text Template", "Array join", "Array Property Set", "JSON", and "Send missing users in SP to Manager". Both paths converge and lead to "REMOVE ARRAY", "Outputs", "Create Data", "Set SPDC tickets in Zoho", and finally "Zoho Desk updates Ticket".

Module Library

- CATEGORIES
- Most Used
- Triggers
- Sources
- Cloud
- Databases
 - Couchbase Executor
 - Firebird Select
 - Influx Select
 - MariaDB Executor
 - MariaDB Select
 - MongoDB Select
 - MySQL Executor
 - MySQL Select
 - MySQL Stored Procedure Read
 - MySQL Executor
 - MySQL Select
 - Oracle Database Select

Test & Debug

Debug Connect

SHOW ALL FREEZE CLEAR

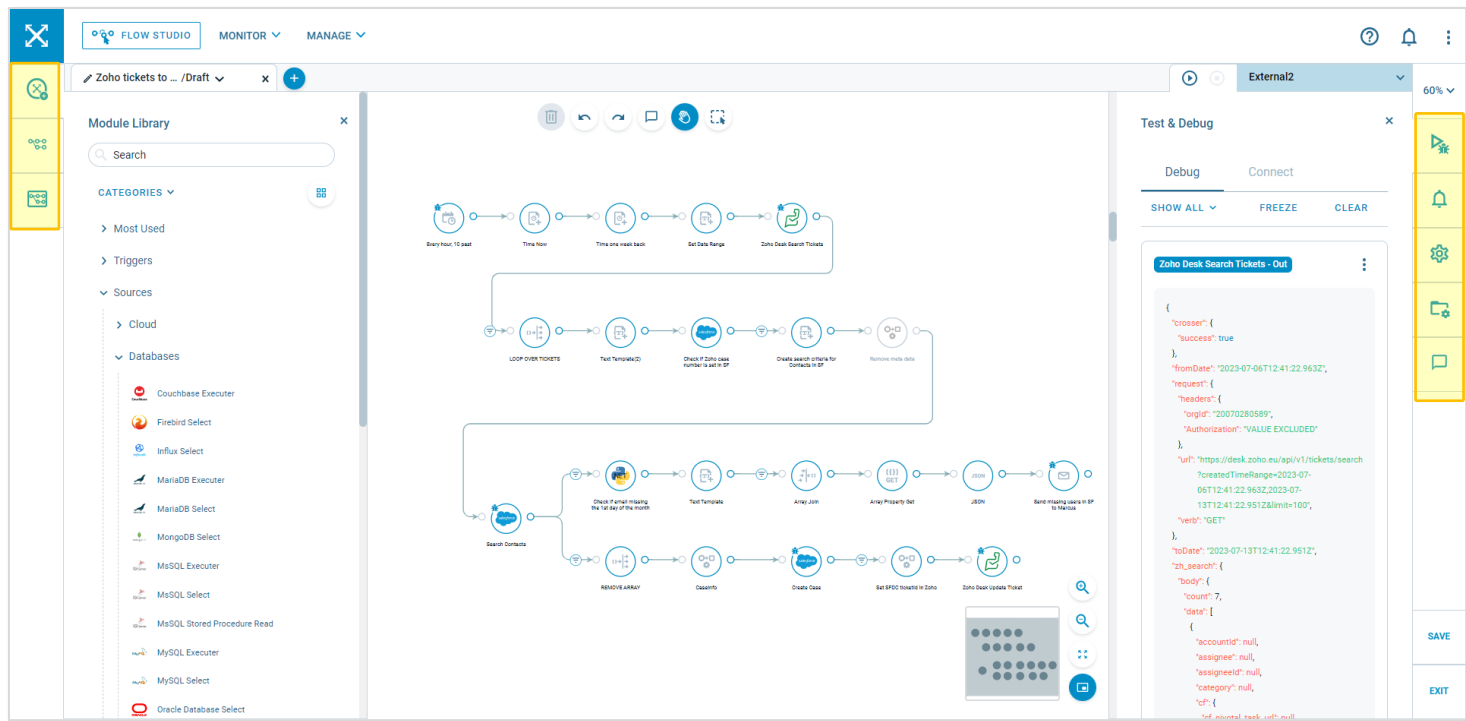
Zoho Desk Search Tickets - Out

```
{
  "crosser": {
    "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,
          "assigneeid": null,
          "category": null,
          "cf": {
            "cf_milestone_name": null
          }
        }
      ]
    }
  }
}
```


The Flow Studio

Side Panels

Module library
Flows
FlowApps



Test & Debug
Notifications
Flow version
settings
Resources
Annotations

The Flow Studio

Module Library

The screenshot displays the Flow Studio interface with a workflow diagram in the center. The workflow starts with a 'Every hour, 10 past' trigger, followed by 'Time Now', 'Time one week back', 'Get Date Range', and 'Zoho Desk Search Tickets'. It then branches into two paths. The top path includes 'LOOP OVER TICKETS', 'Text Template', 'Check if Zoho case number is set in SP', 'Check search settings for contacts in SP', and 'Remove extra data'. The bottom path starts with 'Search Contacts', followed by 'Check if email mapping the 1st day of the month', 'Text Template', 'Array join', 'Array Property Set', 'JSON', 'Send missing data in SP to Slack', 'RECURSE ARRAY', 'Classify', 'Create Case', 'Set SPID tickets in Zoho', and 'Zoho Desk Update Ticket'.

Module Library

Search

CATEGORIES

- > Most Used
- > Triggers
- > Sources
 - > Cloud
 - > Databases

Find the modules you need

- Influx Select
- MariaDB Executor
- MariaDB Select
- MongoDB Select
- MySQL Executor
- MySQL Select
- MySQL Stored Procedure Read
- MySQL Select
- Oracle Database Select

Test & Debug

Debug Connect

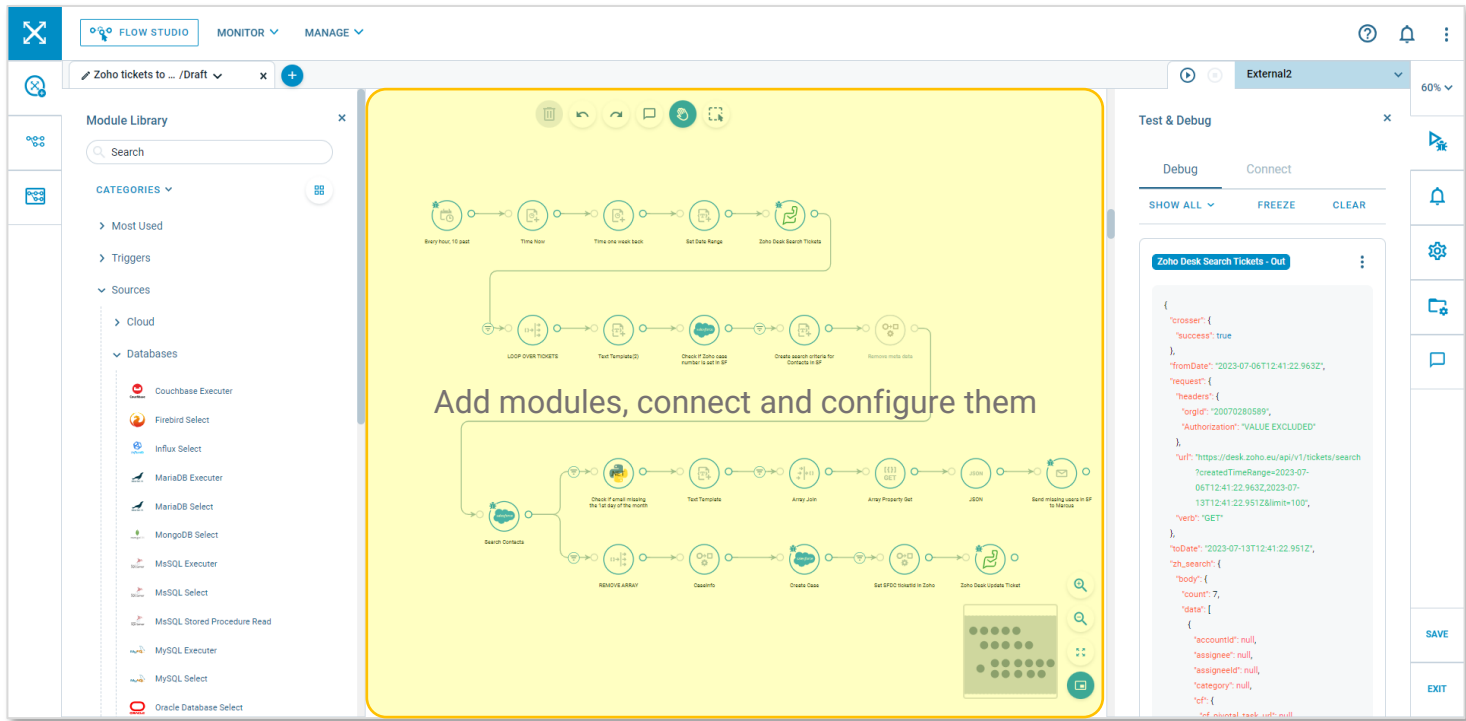
SHOW ALL FREEZE CLEAR

Zoho Desk Search Tickets - Out

```
{
  "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",
  "zn_search": {
    "count": 7,
    "data": [
      {
        "accountId": null,
        "assignee": null,
        "assigneeid": null,
        "category": null,
        "cf": {
          "cf_account_email_conf": null
        }
      }
    ]
  }
}
```



The Drawing



The Flow Studio

Test & Debug

The screenshot displays the Flow Studio interface. On the left is the Module Library with categories like Most Used, Triggers, Sources, Cloud, and Databases. The main workspace shows a complex workflow diagram with various modules connected by arrows. On the right, the Test & Debug panel is open, showing a JSON output from a 'Zoho Desk Search Tickets - Out' module. The output is a JSON object containing search results for Zoho Desk tickets.

Test & Debug Panel Output:

```
{
  "crosser": {
    "success": true
  },
  "request": {
    "headers": {
      "Content-Type": "application/json",
      "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,
          "assigned": null,
          "assigneeId": null,
          "category": null,
          "ch": {
            "ch_created": null,
            "ch_created": null
          }
        }
      ]
    }
  }
}
```

The Flow Studio

Manage Your Flows

The screenshot displays the Flow Studio interface. On the left, a sidebar contains a 'Manage your Flow' panel with options: 'Manage Deployments', 'New Flow from Draft', and 'Delete Draft'. Below this is a 'Sources' section with a list of databases: Couchbase Executor, Firebird Select, Influx Select, MariaDB Executor, MariaDB Select, MongoDB Select, MySQL Executor, MySQL Select, MySQL Stored Procedure Read, MySQL Executor, MySQL Select, and Oracle Database Select. The main workspace shows a flow diagram with nodes like 'Every Hour: 10 past', 'Time Now', 'Time one week back', 'Get Date Range', 'Zoho Desk Search Tickets', 'LOOP OVER TICKETS', 'Test Template(2)', 'Check if Zoho desk number is set in SF', 'Create search criteria for Contacts in SF', 'Remove node data', 'Search Contacts', 'Check if email matching the 1st day of the month', 'Test Template', 'Array join', 'Array Property Set', 'JOIN', 'Send missing cases in SF to MariaDB', 'REMOVE ARRAY', 'Outputs', 'Create Case', 'Get EPDC tickets in Zoho', and 'Zoho Desk Update Ticket'. On the right, a 'Test & Debug' panel is active, showing a 'Debug' tab with a 'SHOW ALL' dropdown, 'FREEZE', and 'CLEAR' buttons. Below this is a 'Zoho Desk Search Tickets - Out' section displaying a JSON response.

```
{
  "crosser": {
    "success": true
  },
  "fromDate": "2023-07-06T12:41:22.963Z",
  "request": {
    "headers": {
      "orgId": "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",
  "pl_search": {
    "body": {
      "count": 7,
      "data": [
        {
          "accountId": null,
          "assignee": null,
          "assigneeId": null,
          "category": null,
          "cf": {

```

At the bottom right of the interface, there are 'SAVE' and 'EXIT' buttons.

The Flow Studio

Create New Flows or Open Existing

The screenshot displays the Flow Studio interface. On the left, a sidebar shows a list of flows under the 'Marketing' category. The main canvas displays three flow diagrams: 1. A flow starting with 'one week back', followed by 'Set Date Range', and 'Zoho Desk Search Tickets'. 2. A flow starting with 'Text Template', followed by 'Check if Zoho case number is set in SF', 'Create search criteria for Contacts in SF', and 'Remove meta data'. 3. A flow starting with 'if email missing day of the month', followed by 'Text Template(2)', 'Array Join', 'Array Property Get', 'JSON', and 'Send missing users in RF to Marcus'. The right panel, titled 'Test & Debug', shows a JSON payload for a 'Search Contacts - Out' action.

```
{
  "channel": "Email",
  "classification": "Spam-advertising",
  "crosser": {
    "success": true
  },
  "email": "maria.gordan@leadinfoolution.tech",
  "id": "44473000008318067",
  "request": {
    "headers": {
      "Authorization": "VALUE EXCLUDED"
    },
    "url": "https://eu19.salesforce.com/services/data/v53.0/parameterizedSearch?q=maria.gordan@leadinfoolution.tech&object=Contacts&contact.fields=id",
    "verb": "GET"
  },
  "wf_search": {
    "body": {
      "searchRecords": [
        :
      ]
    },
    "contentType": "application/json",
    "statusCode": 200
  }
}
```

The Flow Studio

Visual Appearance

The screenshot displays the Flow Studio interface with a workflow diagram in the center. The workflow starts with a 'Every hour, 10 past' trigger, followed by 'Time Now', 'Time one week back', 'Get Data Range', and 'Zoho Desk Search Tickets'. It then branches into two parallel paths. The top path includes 'LOOP OVER TICKETS', 'Text Template', 'Check if Zoho case number is set in SP', 'Create search criteria for contacts in SP', and 'Remove extra data'. The bottom path includes 'Search Contacts', 'Check if email missing the 1st day of the month', 'Text Template', 'Array Join', 'Array Property Out', 'JSON', and 'Send missing cases in SP to Redis'. Both paths converge at a 'Zoho Desk Create Ticket' module. A yellow box highlights the 'Undo/Redo Multi-select Annotations' toolbar at the top of the canvas, and another yellow box highlights the 'Pan/Zoom' toolbar at the bottom right. The left sidebar shows the 'Module Library' with categories like 'Most Used', 'Triggers', 'Sources', 'Cloud', and 'Databases'. The right sidebar shows the 'Test & Debug' panel with a 'Debug' tab and a 'Zoho Desk Search Tickets - Out' output window displaying a JSON response.

Module Library

Search

CATEGORIES

Most Used

Triggers

Sources

Cloud

Databases

Couchbase Executor

Firebase Select

Influx Select

MariaDB Executor

MariaDB Select

MongoDB Select

MySQL Executor

MySQL Select

MySQL Stored Procedure Read

MySQL Executor

MySQL Select

Oracle Database Select

Undo/Redo Multi-select Annotations

Pan/Zoom

Test & Debug

Debug Connect

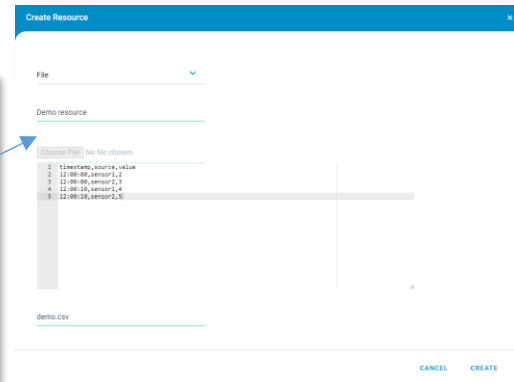
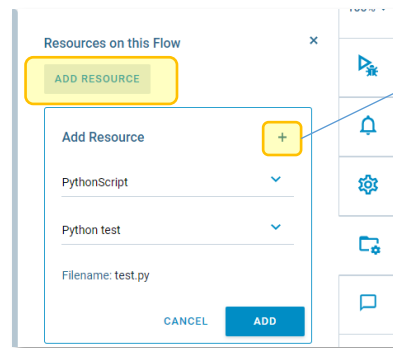
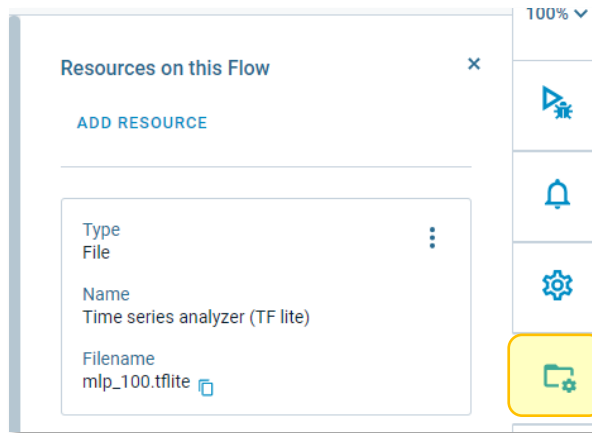
SHOW ALL FREEZE CLEAR

Zoho Desk Search Tickets - Out

```
{
  "crosser": {
    "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,
          "assigneeid": null,
          "category": null,
          "cf": {
            "cf_mismatch_ticket_with": null
          }
        }
      ]
    }
  }
}
```

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*





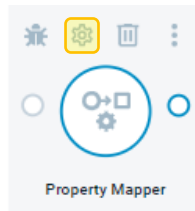
USING MODULES IN THE FLOW STUDIO




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)





Property Mapper

Settings

Common

Documentation >

Name

Property Mapper

Version

3.0.1

▼

☒ Keep Properties

If true all properties not renamed or removed will be included in the output. False will only include properties renamed or added

Move Properties

data.id

data.name

Left is current name, right is new name

Remove Properties

The name of the property to remove.

Add Properties

source

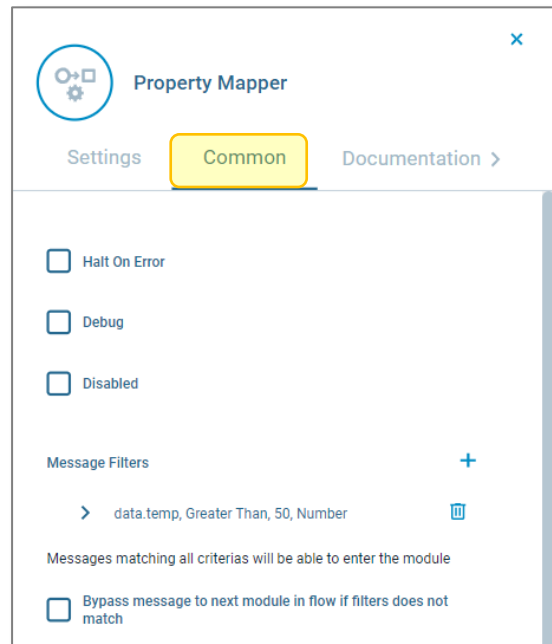
mqtt

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



Modules

Documentation tab

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

The screenshot shows a web application interface for the 'Property Mapper' module. On the left is a sidebar with a 'Documentation <' button. The main content area has two tabs: 'Documentation' (selected) and 'Release Notes'. Below the tabs, the 'Property Mapper' section contains a general description and a settings table.

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.

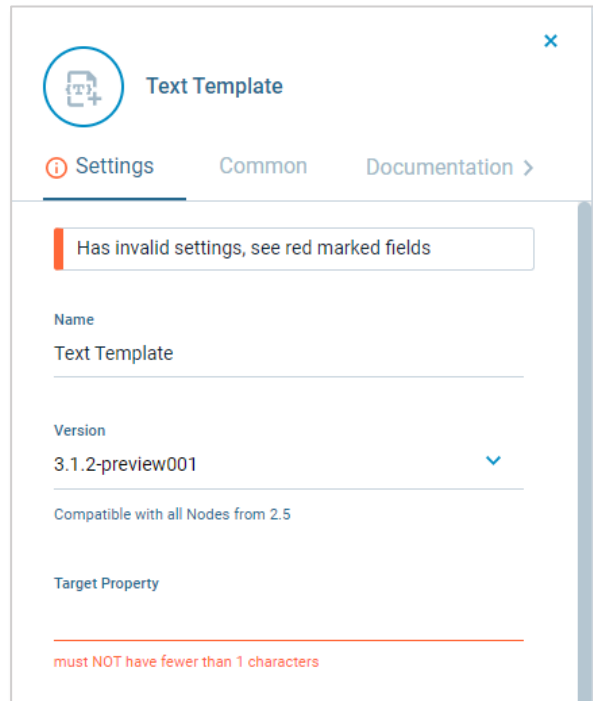
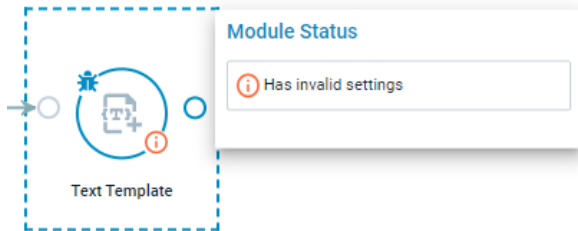
Settings

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





TESTING YOUR FLOW

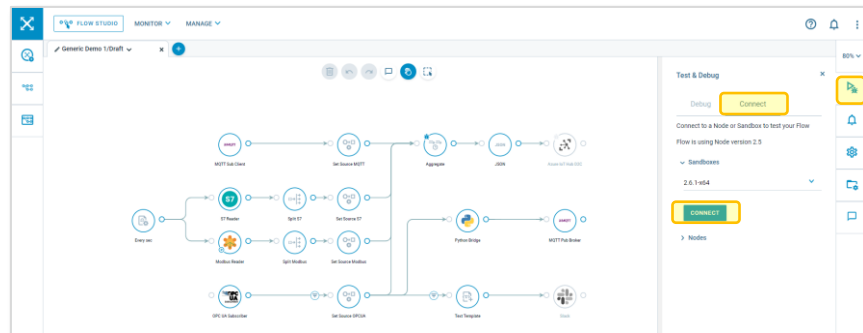
In the Flow Studio
Deploy to a Node

Connect to a Node

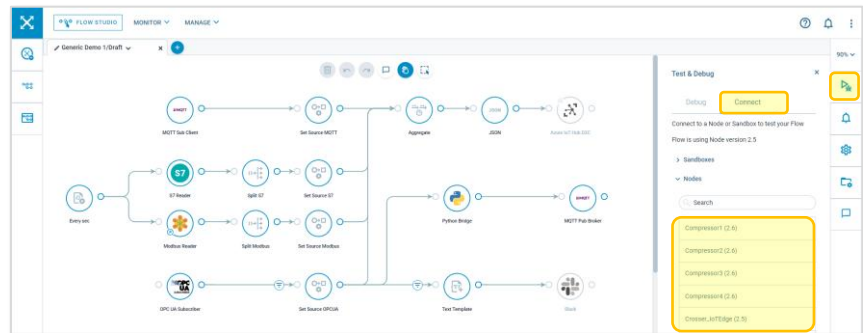
- Connect to a node to use for testing
 - **Sandbox**: Node hosted by Crosser. Can be used with flows that don't need access to local data sources. Only available in the Flow Studio
 - **Local nodes**: Nodes running on your infrastructure

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

Sandbox



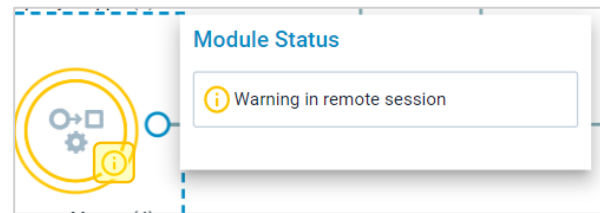
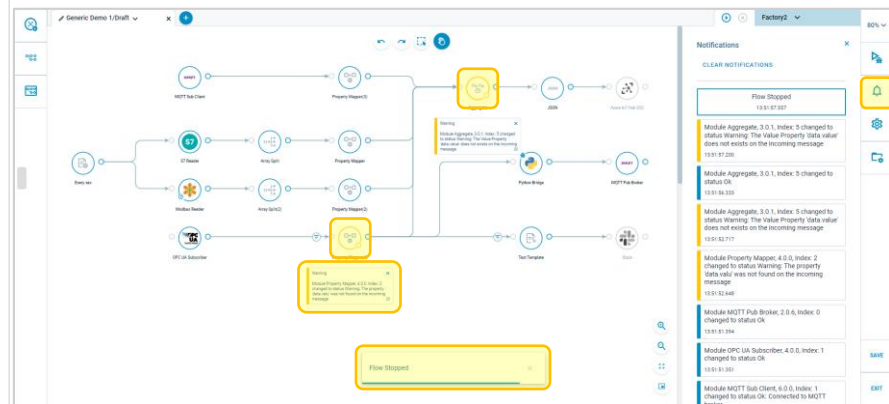
Local Nodes



Test Flows

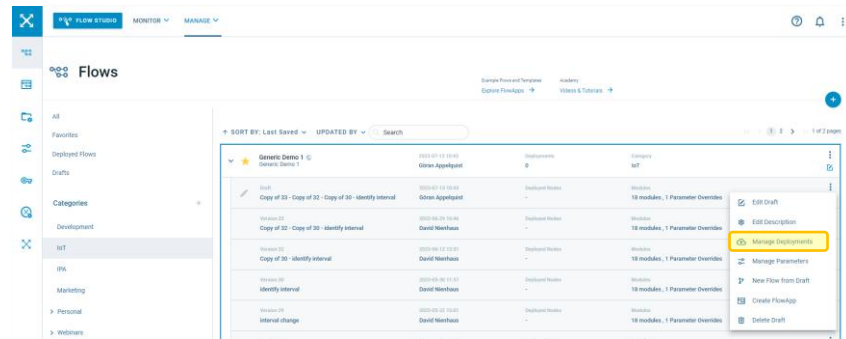
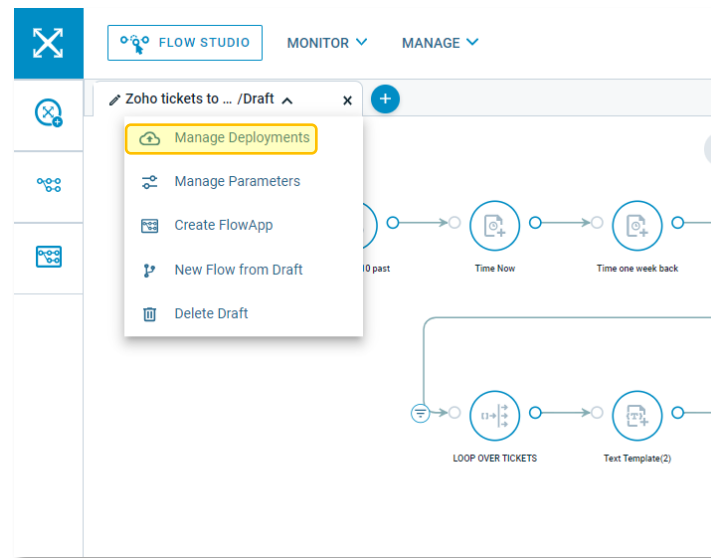
Notifications

- 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



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



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

The screenshot shows the 'Manage Deployments' interface. At the top, it says 'Generic Demo 1 - Version: 18' and 'Draft for demo'. Below this, there are tabs for 'DEPLOY' and 'DEPLOYMENTS'. The 'DEPLOY' tab is active, and a 'DEPLOY THIS VERSION' button is highlighted. Below the tabs, there is a table of nodes with columns for Node, Node Status, Labels, and Flow Version deployed.

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

The screenshot shows the 'Data Generator' settings window. It has tabs for 'Settings', 'Common', and 'Documentation'. The 'Settings' tab is active. It contains the following fields and options:

- Sample interval:** A text input with the value '1' and a dropdown menu set to 'Seconds'. Below it, a note says 'Set to 0 if no interval is needed'.
- Run on start:** A checked checkbox.
- Number of Samples:** A text input with the value '10'. Below it, a note says 'The number of samples to use when producing data'.
- Add timestamp to each sample:** An unchecked checkbox.
- Output Strategy:** A dropdown menu set to 'Random'.
- JSON Template:** A text area containing a JSON object:


```
{  "id": 1,  "name": "machine",  "data": {    "temp": 12,    "pressure": 489  } }
```
- Data Rules:** A list of rules for generating data for each property in the template. The rules are:
 - 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.

Aggregate

SettingsCommonDocumentation >

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

Will preserve properties for each aggregation object as long as they contain the same value during each aggregation interval/period.

Mode

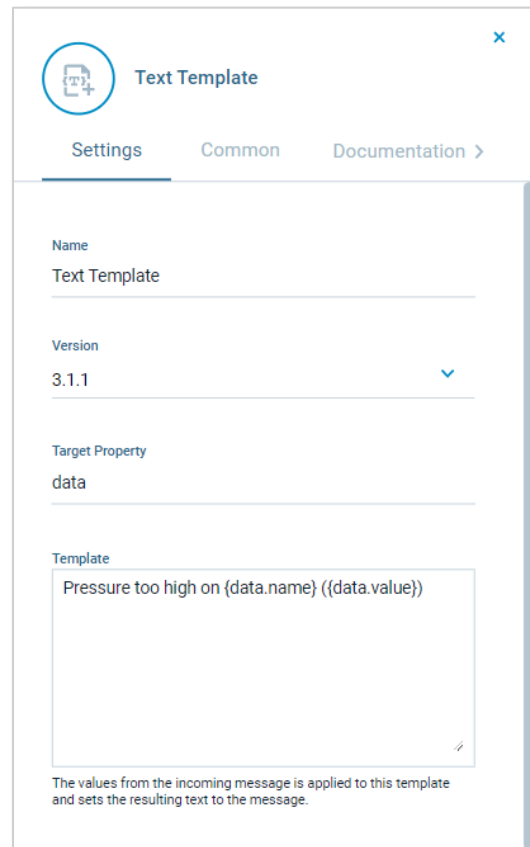
Time

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 window in the Crosser application. The window has a title bar with a close button (X) and a tab bar with three tabs: 'Settings' (selected), 'Common', and 'Documentation >'. The 'Settings' tab contains the following fields:

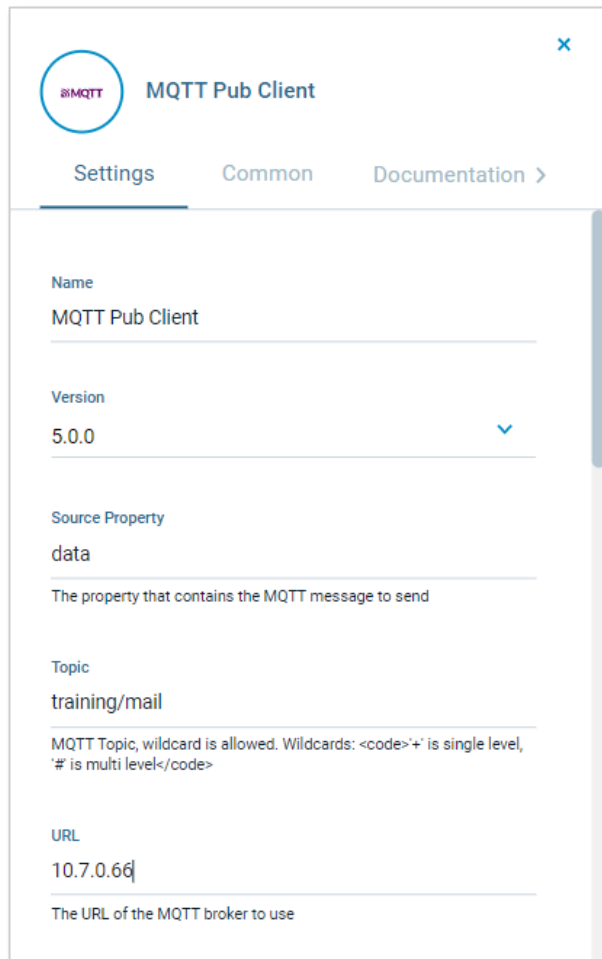
- Name:** Text Template
- Version:** 3.1.1 (with a dropdown arrow)
- Target Property:** data
- Template:** Pressure too high on {data.name} {(data.value)}

Below the template field, there is a note: '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 the 'MQTT Pub Client' settings window. It has a title bar with a close button (X) in the top right corner. Below the title bar is a navigation bar with three tabs: 'Settings' (selected), 'Common', and 'Documentation >'. The main content area is divided into sections for different settings:

- Name:** A text field containing 'MQTT Pub Client'.
- Version:** A dropdown menu showing '5.0.0' with a blue downward arrow.
- Source Property:** A text field containing 'data'. Below it is a description: 'The property that contains the MQTT message to send'.
- Topic:** A text field containing 'training/mail'. Below it is a description: 'MQTT Topic, wildcard is allowed. Wildcards: <code>+</code> is single level, <code>#</code> is multi level</code>'.
- URL:** A text field containing '10.7.0.66'. Below it is a description: 'The URL of the MQTT broker to use'.



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

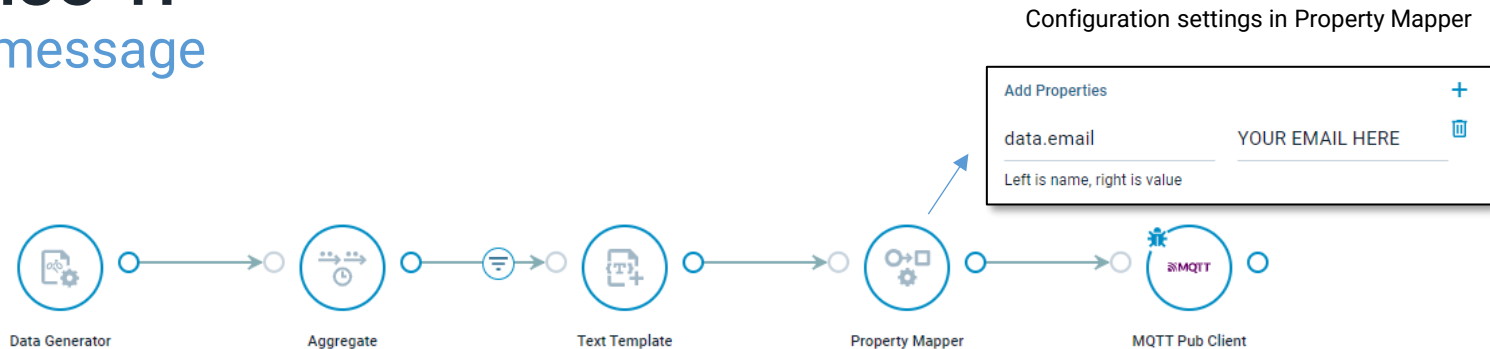
Message filters



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

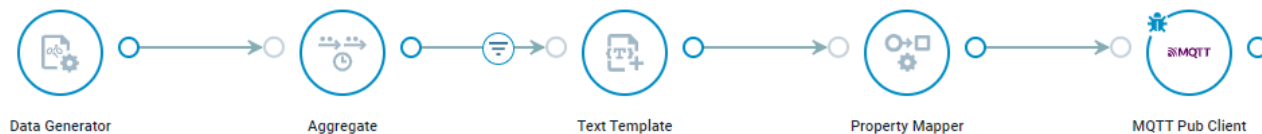
Send a message



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