**Make.com**

**Basic- Scenarios UI**

1. **What is Make.com? (In very simple words)**

* Make.com = Digital Automation tool
* It is a no-code / low-code automation platform that helps you connect different apps and make them work automatically without heavy programming.
* Real-life example:
  + When someone fills a Google Form → automatically save data in Google Sheet → send an email → send WhatsApp message
* All this without writing code.
* In simple words
* Make allows you to automate repetitive work by visually connecting apps and actions, so tasks run automatically instead of manually.

1. **What is Automation?**

* Automation means making a machine or software do work automatically, so humans don’t have to do the same task again and again.
* **Real-life examples of automation:**
* **Automatic washing machine:** 
  + You select a mode, and the machine washes, rinses, and dries clothes by itself.
* In short: Automation saves time, effort, and human work by letting systems work on their own.

1. **What can you do with Make.com?**

* You can automate tasks like:
* Google Forms → Google Sheets
* Website form → Email
* Shopify → Order → WhatsApp message
* Instagram leads → Excel/Sheet
* Django backend → API → Email / SMS
* CRM → Customer notification
* Since you are a Python + Django learner, Make.com is very useful for:
* API automation
* Backend workflows
* No-code integrations

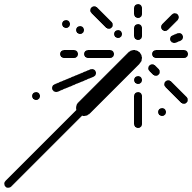
1. **First thing you should do after signup**

* Step 1: Login → Dashboard
* You will see: Create a new scenario
* Templates
* Left menu (Scenarios, Webhooks, Connections)
* Step 2: Click Create a new scenario
* You will see: Big ➕ button
* This is where automation starts

1. 

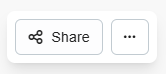
* **▶️ Run once:**
* When you are building a scenario, you need to test it to ensure that it works as intended. This is where you would use the Run once button.
* This will 'execute' your scenario.
* Once you have built a scenario, you can then select when to schedule it at intervals, depending on your need.
* This is where you would use scheduling.
* Example: Click Run once to check if data is moving from one app to another.
* **🔁 Immediately as data arrives (toggle, Auto or Scheduling(Test automation)):**
* This decides when your automation should run
* If ON → automation runs automatically whenever new data comes
* If OFF → automation runs based on schedule (every 5 min, 1 hour, etc.)
* Example: ON → New form submitted → automation runs instantly

OFF → Automation runs every 15 minutes

* **💾 Save scenario:**
* Saves your automation (scenario) in its current state.
* Always click Save after making changes
* ** Auto Align:**
* Used for advanced settings
* Auto-align- the more modules that you add to your scenario, the messier your scenario designer may become.
* Makes the scenario look neat and organized
* Helpful for large workflows (not mandatory for beginners)
* ✈️ **Explain Flow (Run scenario):**
* Another way to start or test or run the full automation or scenario
* Explain flow once you have your completed scenario, clicking this button will show you 'the journey' your data takes in the scenario.
* **🔄Scenario Inputs and Outputs:**
* Shows **what data comes in (input)** and **what data goes out (output)**
* Helps understand data flow between modules
* **⚙️ Scenario settings(Control behavior):**
* Control automation behavior
* Error handling, auto retries, Execution limits, etc.
* **📝 Notes:**
* Add notes or comments inside the scenario
* Useful for remembering logic or explaining steps
* **🕘 Previous Version:**
* If you have multiple saved versions of your scenario, you can switch back to these from here.
* Used after building the scenario
* Runs the scenario automatically at fixed intervals
* Example: Every 5 minutes, every hour, daily

1. 

* **🔀Flow Control:**
* Controls the flow of automation
* Used for: Conditions, Routers, Loops
* **🛠️ Tools:**
* Built-in helper tools / Helps you debug errors
* Tools are **helper modules** for:
  + Math operations
  + Date & time
  + Text handling
  + Data formatting
* **Debugging is done via logs & execution history,** not Tools.
* **Text Parser:**
* Extract or format text
* Example: Get email from text, Split name, number, or message
* **🟣 AI:**
* AI assistant for automation help
* Can suggest automation logic or data mapping
* **➕ Plus (+) (Add modules):**
* Add: New modules, Tools, Functions
* Add an app Clicking the + button will open the various apps that we can start to add to our scenario.
* Have a quick scroll through - there's a lot to work with!
* Note that you can search for your chosen application.
* **💡 Help**
* Tips, documentation, and support

1. 

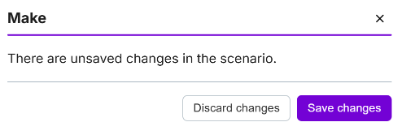
* This option lets you ***share your automation (scenario) with other people***
* It is useful if you want to:
* Show your Make skills
* Teach others how you created the scenario
* Other users can copy your scenario into their own Make account
* Important:
* Your connections, passwords, and login details are NOT shared
* Only the values and structure of the scenario are visible
* **Private backup (Export / Import)**
* You can export your scenario as a blueprint (backup file)
* You can later import it back if needed
* You can also copy the blueprint and share it safely with other Make users
* Blueprint = Backup of your scenario
* **In simple words**
* Share → Show your automation to others (safe, no passwords shared)
* Export → Save a backup of your scenario
* Import → Restore your scenario from backup

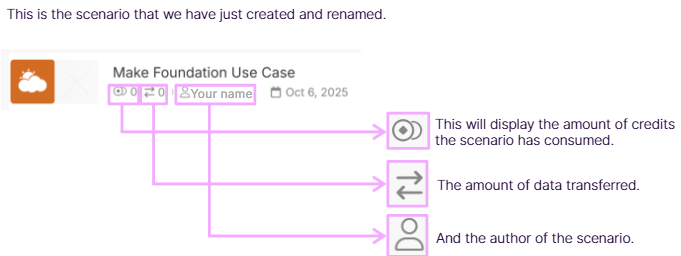
1. **Important words you must know (VERY IMPORTANT)**

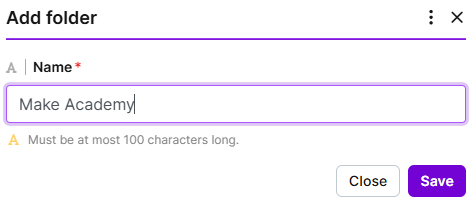
* Scenario: A workflow (automation flow)
* Example: Trigger → Action → Action
* Module:A single app step inside a scenario
* Example: Gmail module, Google Sheets module, Webhook module
* Trigger:Starting point of automation
* Examples: New form submitted, New order created, New webhook request
* Trigger runs the scenario automatically
* Action:What should happen after trigger
* Examples: Send email, Create row in Google Sheet, Send WhatsApp message

**Basic- Renaming / Saving scenarios**

1. **Creating and Saving a Scenarios**

* Step 1: Open Make and select Create a new scenario from the Organization / Team / Scenarios menu.
* Step 2: In the top left hand corner of the screen you will see text stating New scenario.
* Hover your mouse over this and select it with your mouse notice you can edit the text? Delete what is there, and rename this to Make Foundation Use Case.
* Step 3: Adding your first app
* Select the plus button. This will load our many usable apps.
* In the search box, type Weather.
* Select the Weather app.
* You will then be given a list of actions associated with this app.
* Select the Get current weather action module.
* Step 4: Configuring your first app
* This will bring up the configuration for this module. The only task required from you here is to put in a location.
* Type your current city in the City field.
* Click OK.
* A note on automatic naming.
* A note on general functions.
* Step 5: Saving the scenario
* At this stage it is important to save your scenario.
* Click the save icon on the control panel at the bottom of the scenario builder.
* Note that if you try to exit the scenario from within Make, you will be greeted with this prompt.
* This concludes this task. Click Next to continue.
* Step 6: Next you will organize your scenario by placing it in a folder.
* This represents the best way to keep your scenarios organized as you create them.
* To get started, select the scenario option from the menu.
* Step 7: This is the scenario that we have just created and renamed



* Step 8: Next you will add a new folder.
* In Make, click Scenarios in the left hand menu.
* Then select the + icon and type 'Make Academy’.
* Click Save when you are done.
* Step 9: Move to the right of your scenario and select the down arrow. Then select move to folder.
* You will receive this prompt on screen - select the drop down and then select Make Academy.
* Click the Move button to complete this task.
* You can then navigate to the Make Academy folder and view your first scenario there.

**Basic- What is an app / module**

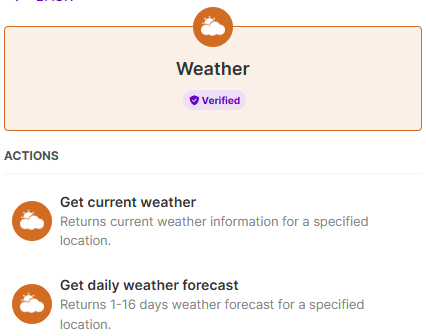
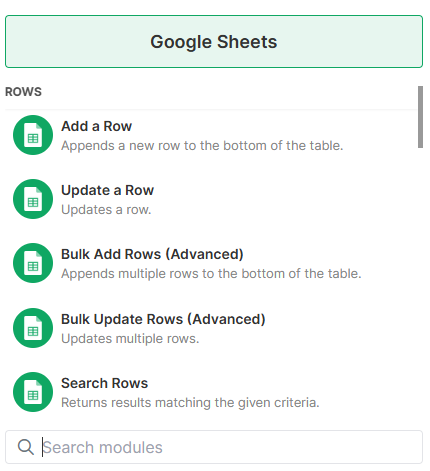
1. **What do we mean by module?**

* To keep it as simple as possible at this stage, a module is any element that you add within your scenario.
* Let's find out how to add another module. On your Make Foundation Use Case, hover over the Weather app - you see you will have the option to add another module.
* Alternatively, you can double click anywhere in your scenario editor to add another module.
* Next let's explore the different types of module that exist.

1. **The types of modules:-**

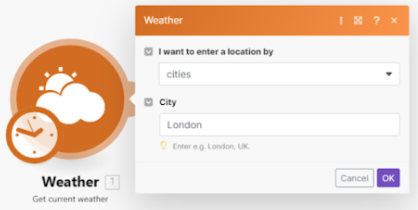
* In total you can use **five types of modules** in Make. Each of these will perform a separate function.
* Note that last two modules listed here (Iterators and Aggregators) are considered advanced, so don't worry about these right now.
* **ACTION:** We will use actions the most within our scenarios.
* Our Weather module is an example of this. It will get the data about the current weather for us.
* Terminology to learn: the data we have collected from this (or any) action will be referred to as a bundle.
* Other examples of an action module are:
  + Get the details about a post
  + Add a row to a spreadsheet
  + Send a push notification
  + Delete a spreadsheet
  + Create a report
* **Searches:** A search module will look for data that you specify and return results.
* Terminology to remember: Any data collected from Make is known as a bundle.
* For example, you may perform a search within a database / spreadsheet asking to show all users over the age of 30.
* The search module will find this information for you, and output it in to multiple bundles.
* Other examples:
  + Search a range of values users aged 30-35
  + Search by a specific location
* **Triggers:** A trigger module will usually be the first module at the beginning of your scenario.
* It's important to note that there are numerous types of triggers. The main two are known as polling and instant triggers. We will be exploring these in a later learning path.
* All you need to know for now, is that a polling trigger will watch a service, and provide updates each time a scenario is activated. And an instant trigger will watch an account (for example Facebook or Linked.in), and return any new information instantly.
* **Iterators:**  This is where it might start to sound complicated, but don't worry, we will be exploring this topic in much greater depth once we understand terminologies.
* An iterator will extract a list / group of items (known as an array) and split it up into many separate bits of information (known as bundles).
* For example, you receive an email containing three attachments (an array), an iterator could be used to separate each attachment into individual items (3 separate bundles).
* A real life example - it is like receiving a food order with a group of friends, and then handing out each person's order to them.
* **Aggregators:** Think of an aggregator as the opposite to an iterator; an aggregator will take multiple bundles of information, and combine them into a single bundle.
* To think of this in real life terms, it is like ordering several products from a store, and them being shipped into a single packaged order.

1. **What type of module I am using?**

* If you want to see what type of module you are using, you will see it on the list when you select a module to add.
* An **app is a group of modules**.
* For example, if you try to add another instance of the Weather app to your scenario, you will see that it has two action modules.
* This means that the Weather app can only perform these two actions: Get current weather and Get daily weather forecast.
* Soon you will be adding a Google Sheets module to your scenario.
* For now check the image to the left to see that there are different module types available to you in the Google Sheets app.
* When you search for an app like Slack in the module list, you may notice something:
* The module names look different
* You may not see common words like trigger, action, or search
* This is because:
* Each app has its own special functions
* The modules are designed specifically for that app
* Example: Slack
* Most Slack modules are actions (like sending a message)
* But instead of showing “action”, they are grouped under Slack-specific categories
* Example category: Messages

**Basic- Configure a model**

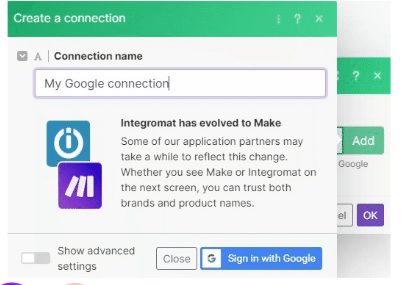
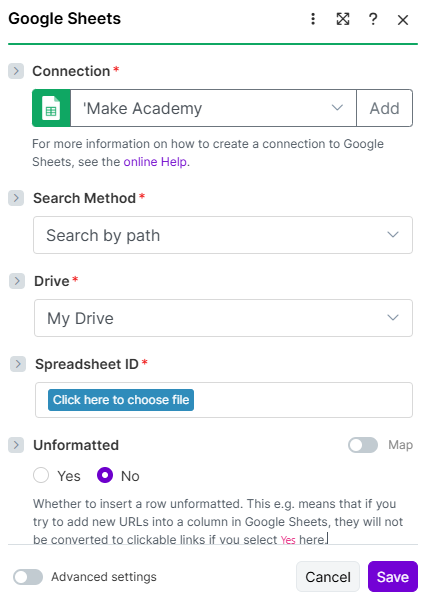
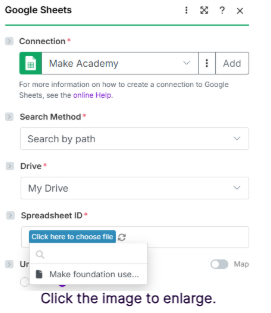
1. **What does “configure a models” means?**

* The best way to describe this is to look at what we have already configured: open your scenario and click the Weather module.
* Note that the action here is Get current weather.
* In order to configure the Weather module, we have two fields to choose from:
* I want to enter a location by and City.
* The only configuration we have made here is entering our location into the City field.

1. **Are all modules configured the same?**

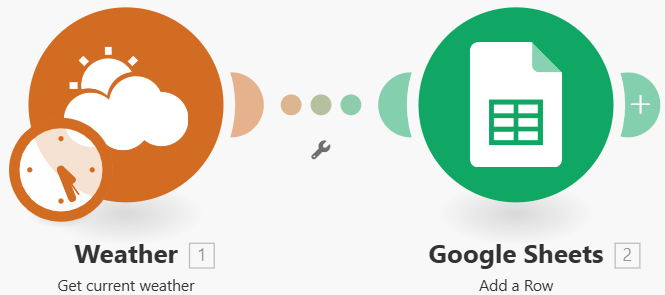
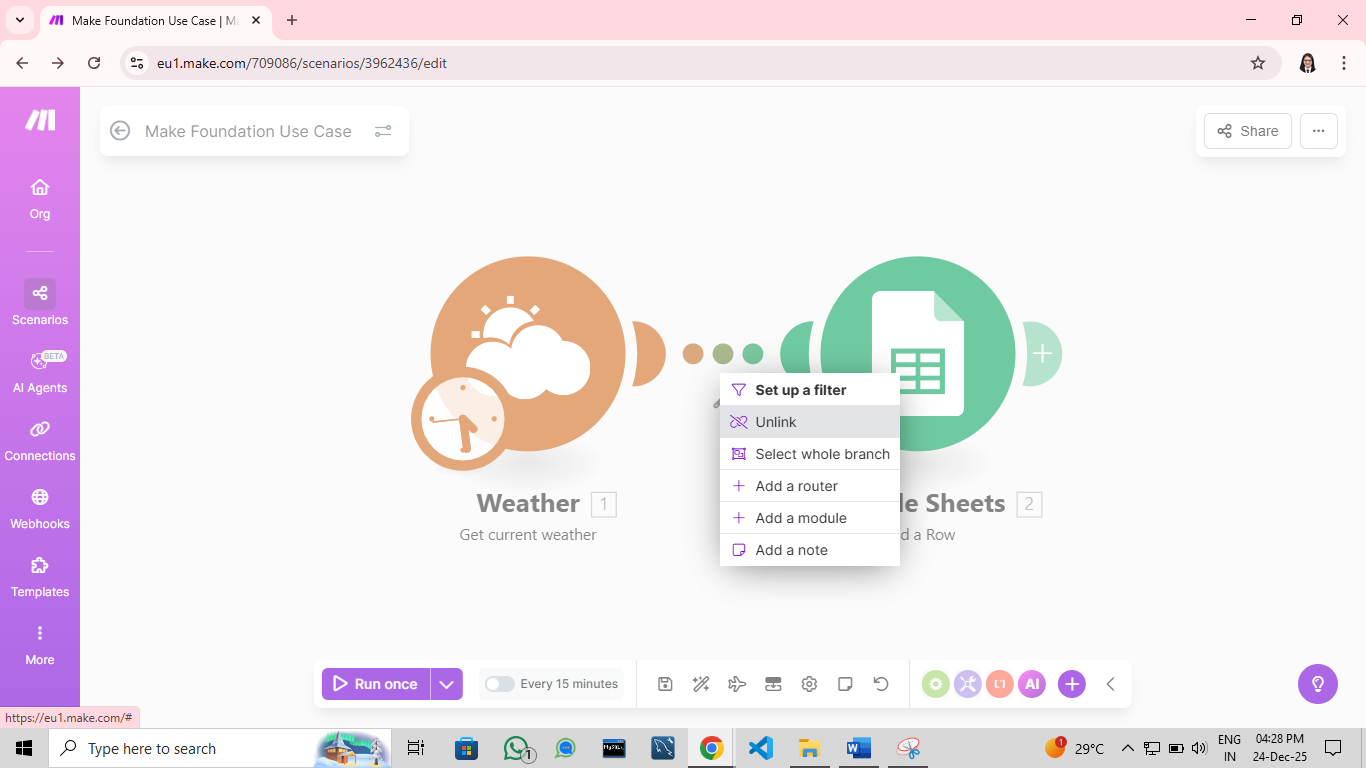
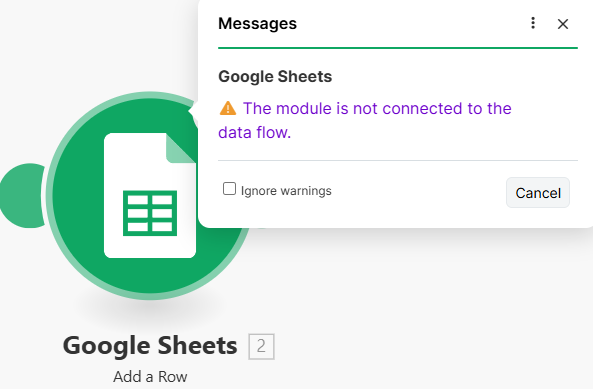
* No they are not - different modules require different levels of configuration. A lot of this will depend on how complex the application you are using is.
* We will start by configuring a simpler module from the Google Sheets app.
* For this you will need a Google account. If you don't have one, sign up here (it's free!).
* Once you are signed up, click Next to proceed.

1. **Let’s configured the Google sheets module.**

* **Step 1:** Adding a Google Sheet for this exercise
* Once you have signed up for a Google account, open Google Drive.
* Navigate to New > Google Sheets > Blank spreadsheet.
* Name the new spreadsheet Make Foundation Use Case.
* You are doing this now so the next step will run smoother.
* That is all you need to do outside of Make
* **Step 2:** Adding the Google Sheet module
* Within your scenario editor in Make, add a new module. Search for "sheets" and select Google Sheets from the options returned.
* You will then be presented with a list of module types this was briefly viewed in the last unit.
* Click the Add a Row module, located under Actions.
* Once you have clicked this, you will be greeted with the configuration prompt.
* **Step 3:** Connecting your Google account to Make
* The first step that you should do is click Add next to the Connection.
* Name your connection Make Academy and click Save.
* As you're already signed in to Google, the only action here is to 'Allow' Make access.
* This will allow Make to access Google Sheets.
* You will see why this is important on the next step.
* **Step 4:** First items to configure
* Now that the connection has been made, you can start to configure the module.
* Let's stick to the very basic options:
* Search Method: leave this as Search by path.
* Drive: as this is a file within your own Google Drive, leave this option as My Drive.
* Next, you'll see why we had to create the spreadsheet ahead of creating the module.
* **Step 5:** Selecting your destination sheet
* Spreadsheet ID: Select Click here to choose file you'll notice that the contents of your Google Drive will load.
* Select Make Foundation Use Case.
* Sheet Name will then appear. By default you will only have one sheet set up. Select 'Sheet1’.
* You'll notice that more options will appear now below sheet name - leave these alone as they are for your next unit. For the moment, click Save to finish.
* ...and don't forget to save! Once you have, you have successfully configured this task. Great job! Click Next.
* **A final note on connections and authorizations**
* You may need to reauthorize connections after a certain period of time.
* For example, a free / regular Gmail account will need to be reauthorized every 7 days, whilst a Gmail Business account does not need to be reauthorized.
* If you change the password on any of your linked accounts, you will also need to reauthorize.
* This can be done from the Connections option in the menu.
* Clicking Reauthorize will reset the connections and allow your scenarios to function as intended.

**Basic- Connecting Module**

1. **Connecting / disconnecting module.**

* **Step 1:** Reviewing your scenario
* In the previous unit we added and configured the Google Sheets > Add a row module.
* Your scenario should look like the image, two modules connected to each other.
* The next step is to disconnect them.
* **Step 2:** Disconnecting a module
* Right click the dotted line connecting the two modules, and select Unlink.
* This will disconnect the two modules from each other.
* **Step 3:** Module warnings
* You will now have two standalone modules - your Google Sheets will present a warning in a yellow circle.
* Select the yellow alert: it will give you the details of the warning or error your module has encountered.
* Your Google Sheets module has an action of Add a row but there is no source of where that information is coming from.
* Therefore Make will provide you with the warning prompt.

1. **Different ways to connect modules.**

* **Method 1:** Your first method is to add the module linked directly from the previous module.
* For example if we hover over the Weather module and then select Add another module, the next module that we select will be automatically connected.
* **Method 2:** If you have not automatically connected a module, you can drag the connector on a module, to the one that you would like to connect it to.
* You can do this from the starting module, or any additional module.
* **Method 3:** Another option is to drag your module towards the module you would like to connect to; this will automatically connect it.
* If you accidentally connect the wrong module, and you are still holding down your mouse, dragging the module away will disconnect it.
* However once you let go of your mouse, you will need to right-click > Unlink if you want to disconnect the module.

1. **Why do we connect module?**

* To put it simply, you are passing information from one module to another.
* From the Weather app, you are sending data about the weather from a city (first module), to a spreadsheet (second module).
* Without connecting these modules, the information doesn't go anywhere from the Weather app, and the spreadsheet has nothing to receive. Think of it as a taxi transporting passengers to their destination.
* If you select the Explain flow icon under controls, it will illustrate the journey of your data.

1. **Have go.**

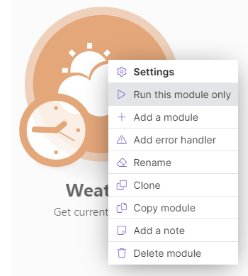
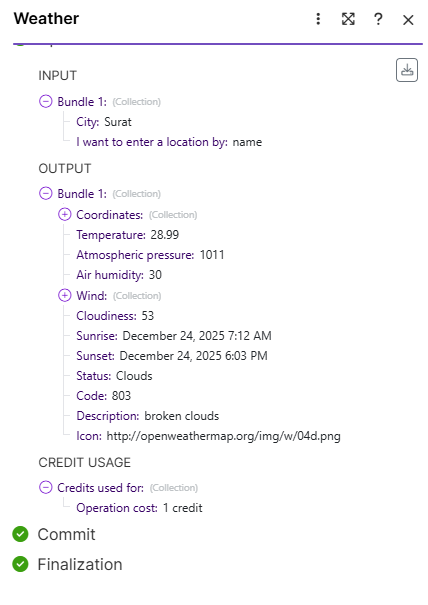
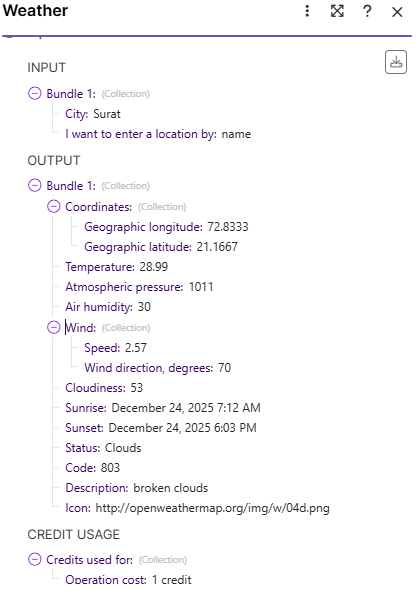
* Spend a few minutes now exploring the different ways that we can connect modules together.
* TIP: a handy function within Make is the use of clone / copy and paste function. You can do this by right-clicking your chosen module, and selecting one of the options. You can also hold shift and drag your mouse to select multiple modules at once.
* This is a useful tool to know about, as it will copy the existing configurations of the module.
* Copy the modules and link them together in a straight line, as shown below.

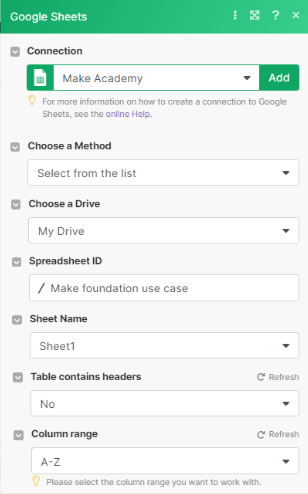
**Basic- Introduction to Mapping**

1. **What is Mapping?**

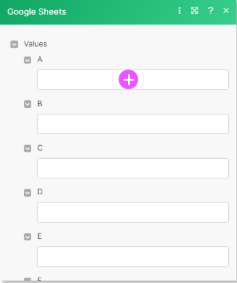
* To keep this question relevant to what you have built in your scenario so far, mapping involves generating items of information from one module, and telling Make where you want the information to appear in the next module(s).
* For example:
  + In the Weather app you want to take the description of the weather
  + You would like this item of information to appear within the first row on your spreadsheet
* In order to do this, you need to map the data from the first module (source) into the second one (destination).
* This is the equivalent of a map that plots a journey, with a destination at the end of it.

1. **Knowing what we can map(task 1)**

* **Step 1:** Generating a bundle / consume a credit
* We're going to split this task into two halves. Make sure that you follow this task at your own pace.
* The first part is identifying the items that you can map.
* On the Weather app, right-click and click Run this module only.
* As this is your source module, it will generate a bundle of information for you.
* **Step 2:** Generating a bundle and performing an operation
* When you have clicked Run this module once, there are two things to review here:
* A thought bubble will appear next to Weather containing a 1.
* This means that:
* You have performed an operation and consumed a credit
* You have generated at least one bundle.
* Click **➕**, and you will be able to see the bundle, and the items it contains.
* **Step 3:** What's in the bundle?
* There are two sets of information here.
* Number 1 - your input bundle might look familiar it shows the configurations you entered into the Weather module, such as the city name.
* Number 2 - this shows your output bundle. Every item here can be mapped in the next module(s).
* Note that some items contain a +, this is known as a collection/array, and it means that there are multiple items that can be mapped here. You will learn more about collections and arrays later on.
* Click the + next to Coordinates / Wind to see what other items you are able to map.
* This concludes the first task click next.

1. **Mapping items into our target module(task 2)**

* **Step 1:** Where to map your items
* You now know from the previous page which items you can map from the Weather module. Now let's look at where you are going to map them to in your target module.
* NOTE: it would be useful if you opened the Make Foundation scenario Google Sheet that you created earlier for this exercise. Ensure that you have selected your Make Foundation Use Case from the Google Sheets module.
* Within Make, open the module Google Sheets > Add a Row. This will load the configuration page.

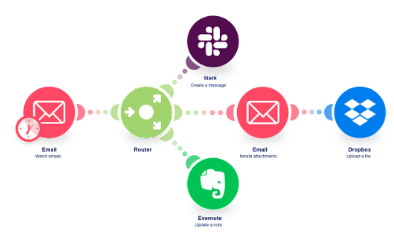


* **Step 2:** Where to map your items
* Scroll down on the configuration panel to Values - you will see several fields that say A, B, C, etc. These correspond with the columns in your Google Sheet.
* Click + in the field labeled A you will be greeted with the fields that you can map. Does this look familiar? It should; these are the items from your bundle. This is real data that you have generated.
* You can select one or many of the items here and it will process that information into row A on your Google Sheet.
* **Step 3:** Mapping items
* To complete this exercise, you are going to map four different items to your Google Sheet.
* For Column A, select Sunrise
* For Column B, select Temperature
* For Column C, select Status
* For Column D, select Description
* Click OK. And then save your scenario.

1. **Running you scenario for the first time**

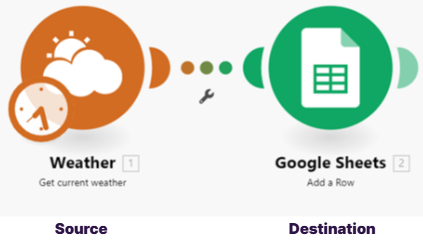
* The next course will cover running your scenario in more depth. However, you're ready to run your first automation now.
* It would be beneficial to have your Google Sheet open in another tab / screen, to see what happens.
* Click the Run once button, and look at your Google Sheet, you will see that it now has content.
* This is what you have mapped from the Weather app.
* Congratulations - you have just run your first scenario!

**Foundation - Adding a router**

1. **What is a router?**

* A router in Make is used to split one automation into multiple paths so different actions can be performed simultaneously or based on conditions.
* A router is a native Make tool that allows you to branch your workflows onto separate paths (or routes).
* Routers are useful for processing data towards separate outputs, and process data differently depending on your needs.

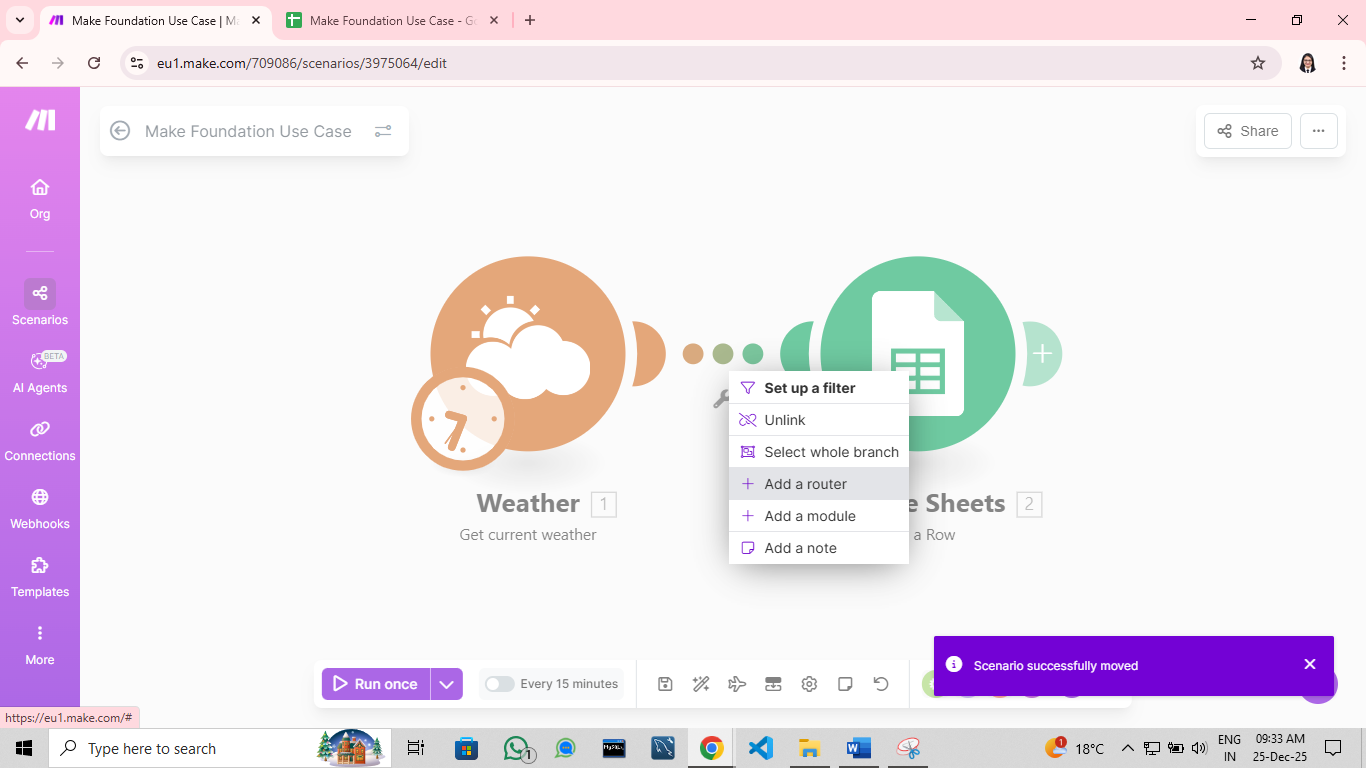
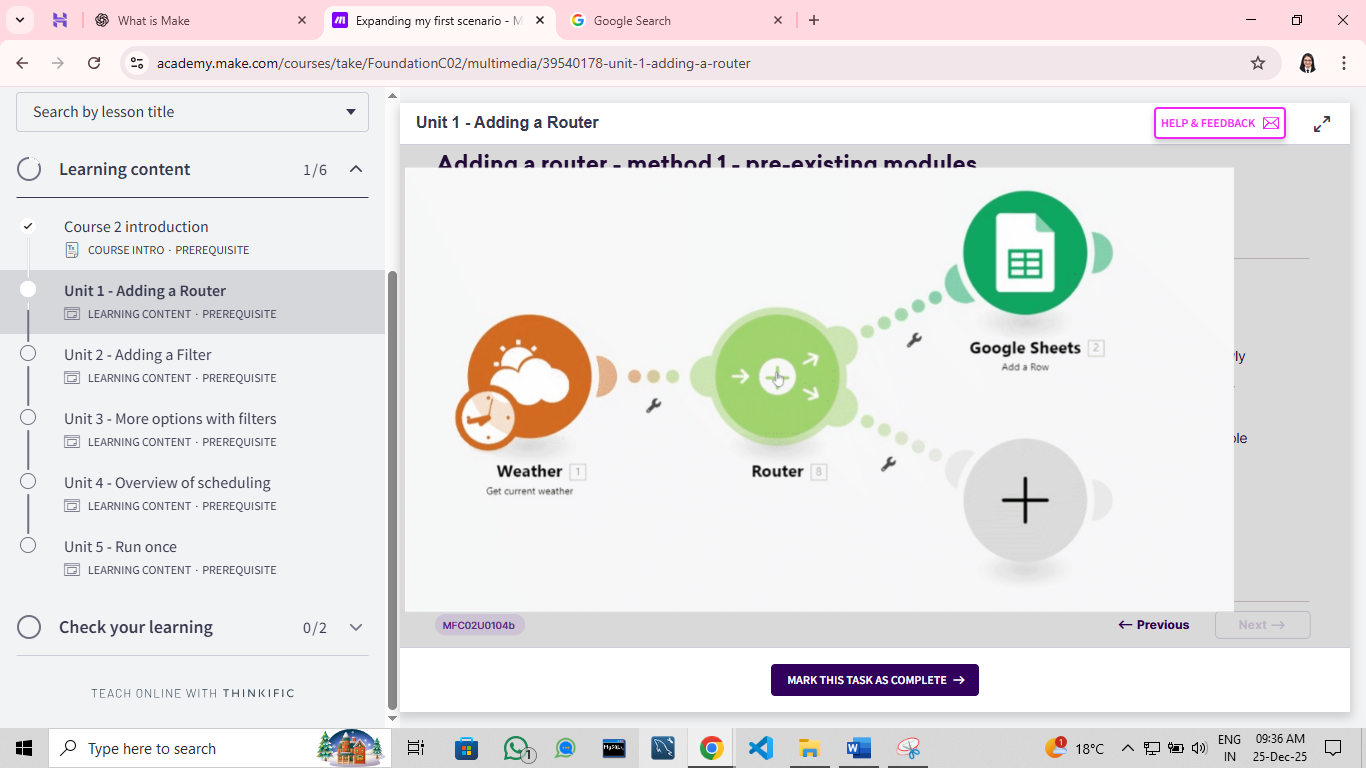
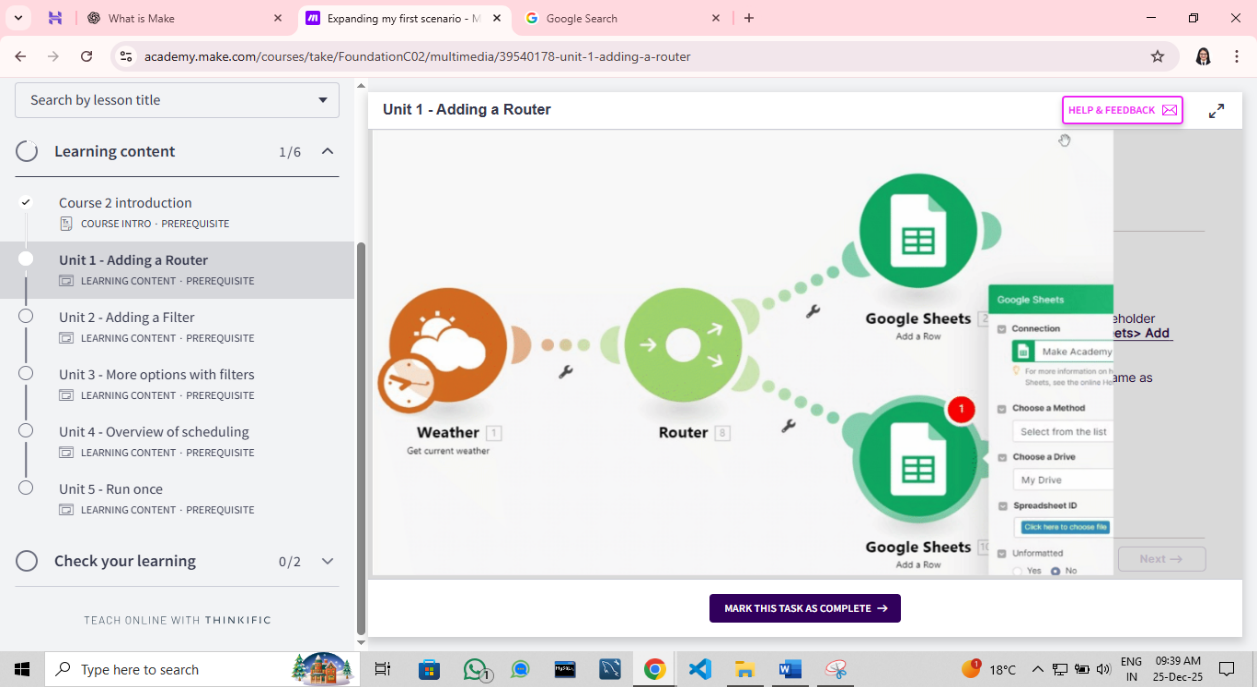
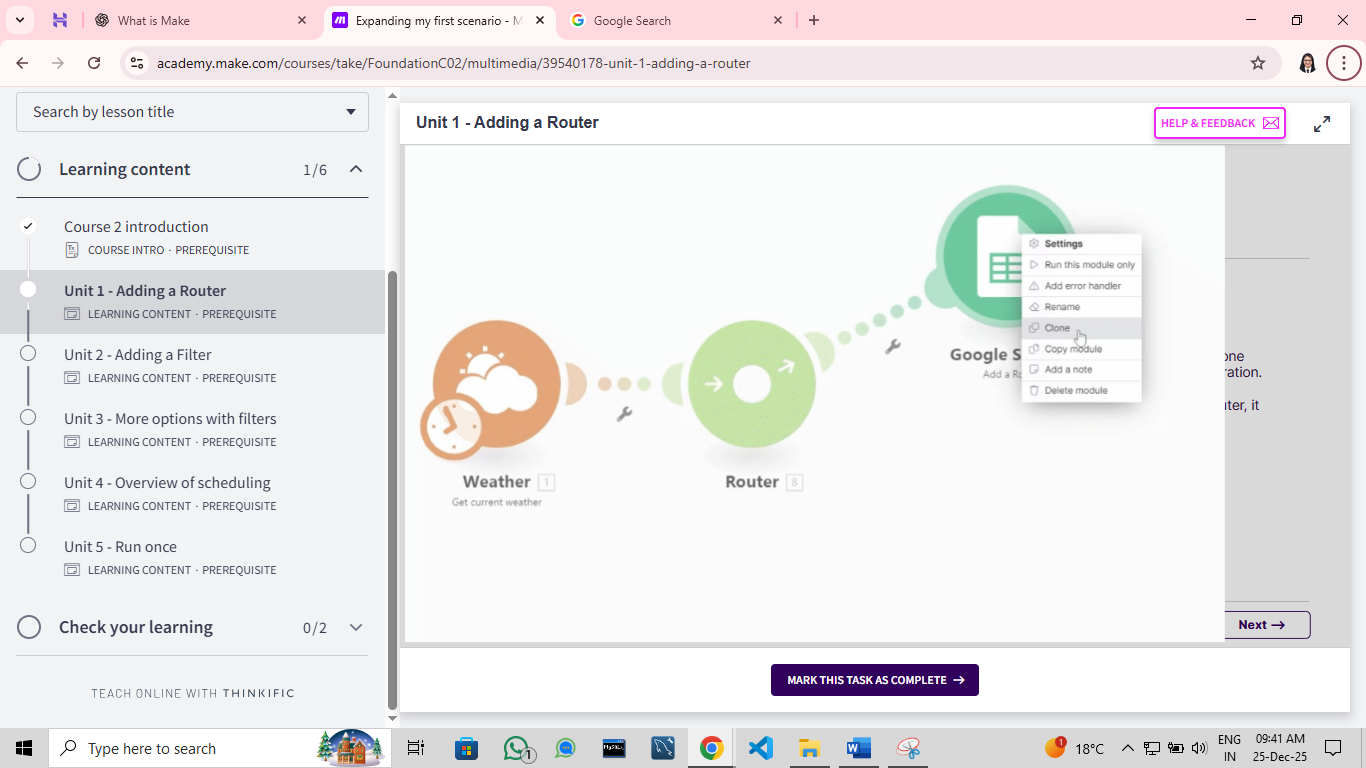
1. **Why would you use a router?**

* In the scenario you created before, the Weather app passes information to a Google Sheet.
* Simultaneously, you could use a router to process the information in a different format for example, to a messaging service like Slack or Microsoft Teams.
* In this example, we'll use the Weather app as the source module, as it contains data about the weather.
* The destination app for the weather data is Google Sheets, but thanks to the router we can add other destination apps, such as a messaging service app.

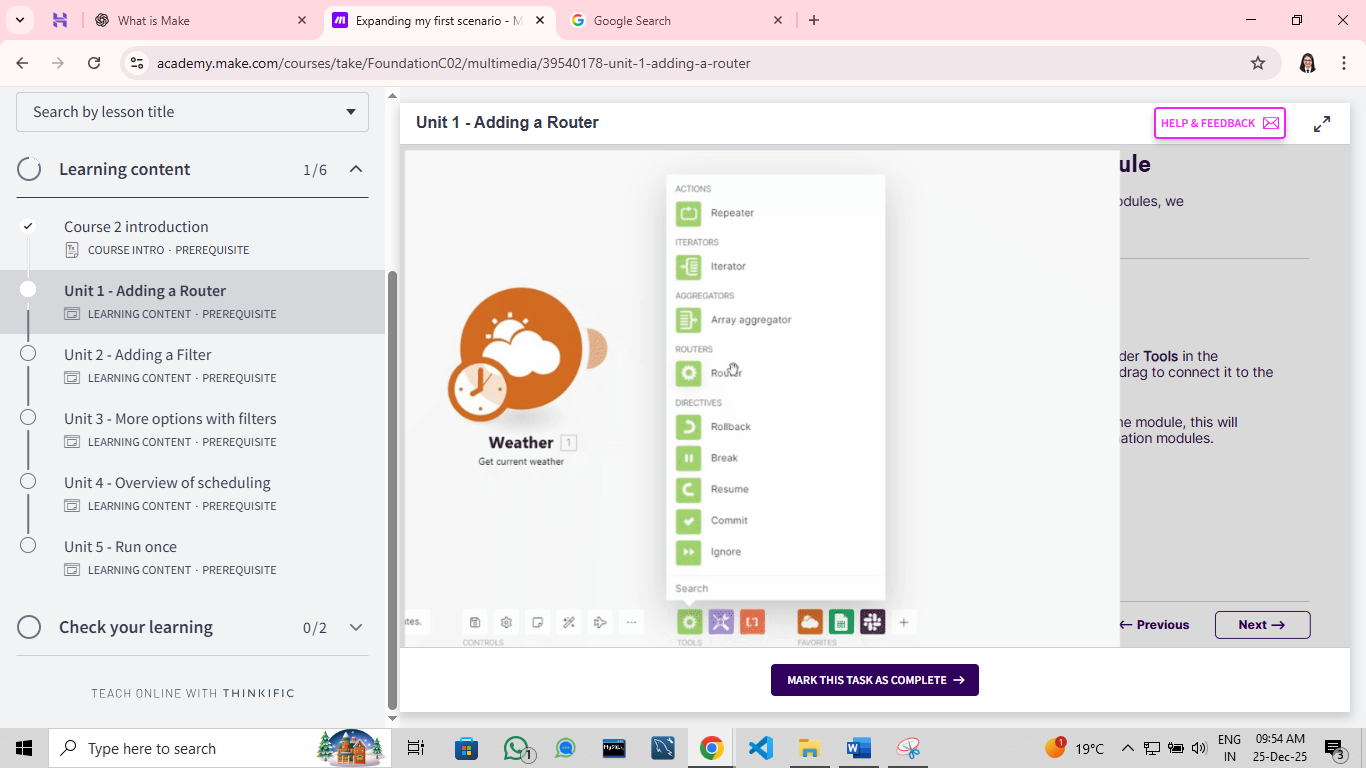
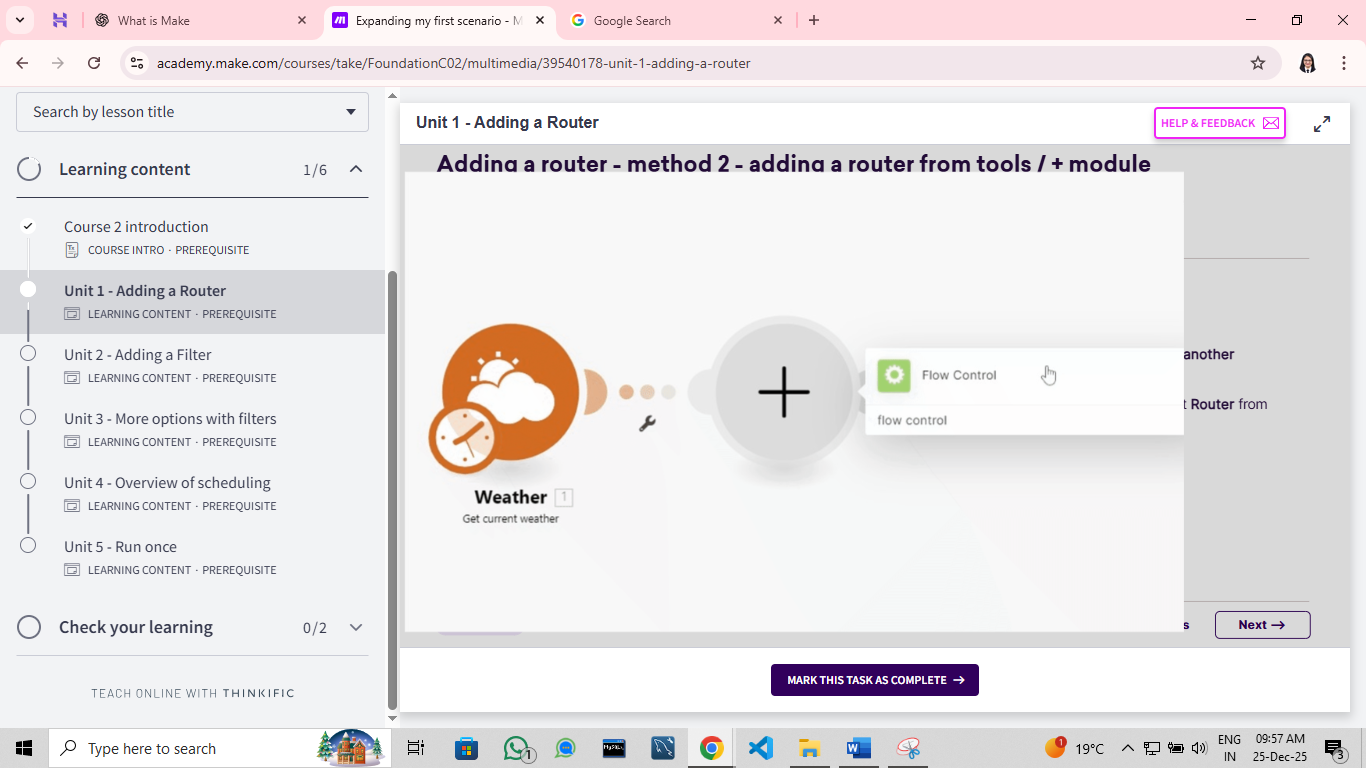
1. **Routers in theory:**

* Let's look at how the previous example would work visually:
* The Weather data is processed, and creates a bundle
* This is sent to the Router, which processes it towards the two destination modules
* First, the data is passed to the Google Sheets module where it will perform the action Add a row
* Finally, the data is passed to the Slack module, where it will perform the action Create a message
* Now that we have explored the concept, let's implement something similar.
* NOTE: whilst we have not yet explored credit usage in Make, it's worth knowing that a router will not consume any credits.

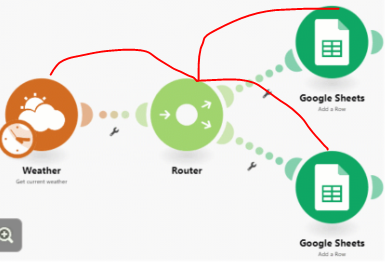
1. **Adding a router - method 1 - pre-existing modules.**

* **Step 1:** Right-click the path between the two modules and select Add a router.
* This will insert the router tool between the two modules.
* **Step 2:** Now that your router is in place, hover over your newly placed router module. You will notice that a **+** state appears. Select this - it will add a placeholder for our new module.
* Note: you can select this multiple times to add multiple placeholders.
* **Step 3:** For the purpose of your scenario, select the placeholder for the new module and add another Google Sheets> Add a Row module to your scenario.
* Configure your new Google Sheets module the same as the previous module, and click OK.
* Save your scenario.
* **Step 4:** An alternative approach here is to right click and clone your Add a Row module - this will save the configuration.
* By dragging your newly cloned module near the router, it will automatically connect to it.

1. **Adding a router - method 2 – adding a router from tools /+ module.**

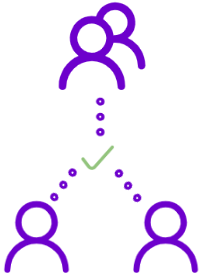
* **Option 1:** From your Flow control located under Tools in the scenario editor, select Router and drag to connect it to the source module.
* When selecting the + icon within the module, this will create placeholders for your destination modules.
* **Option 2:**  Hover over any module and select the +Add another module option.
* Search for routers or flow control, and select Router from the list.

1. **In which order will the data be processed?**

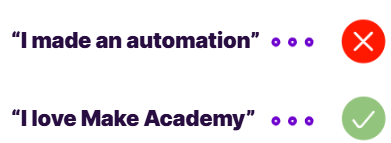
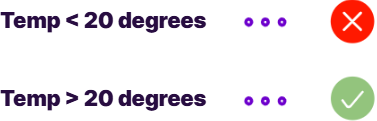
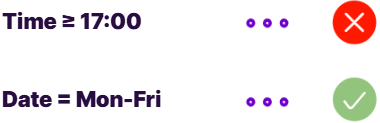
* To visualize the order of the data flow, select the Explain flow item under Controls.
* This will visually illustrate the route the data in your scenario will follow.
* You can set the order of routes in which Make processes them in the scenario.

**Foundation - Adding a filter**

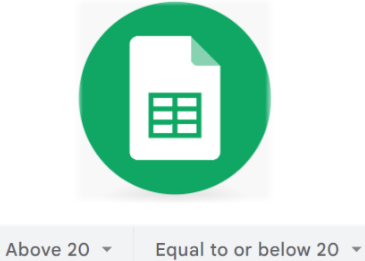
1. **What are filters?**

* A filter is a tool that allows specific data to be passed or restricted, within the flow of a scenario.
* Filters will provide you with an idea of how to start controlling and manipulating data. For example:
* A social media account could generate and save posts if they contained the words make.com.
* To describe this in real-world terms, is to think of attending a concert: only those with a ticket will be allowed in. Those without a ticket cannot enter.
* Your filter here would be: if person has a ticket > let them in.

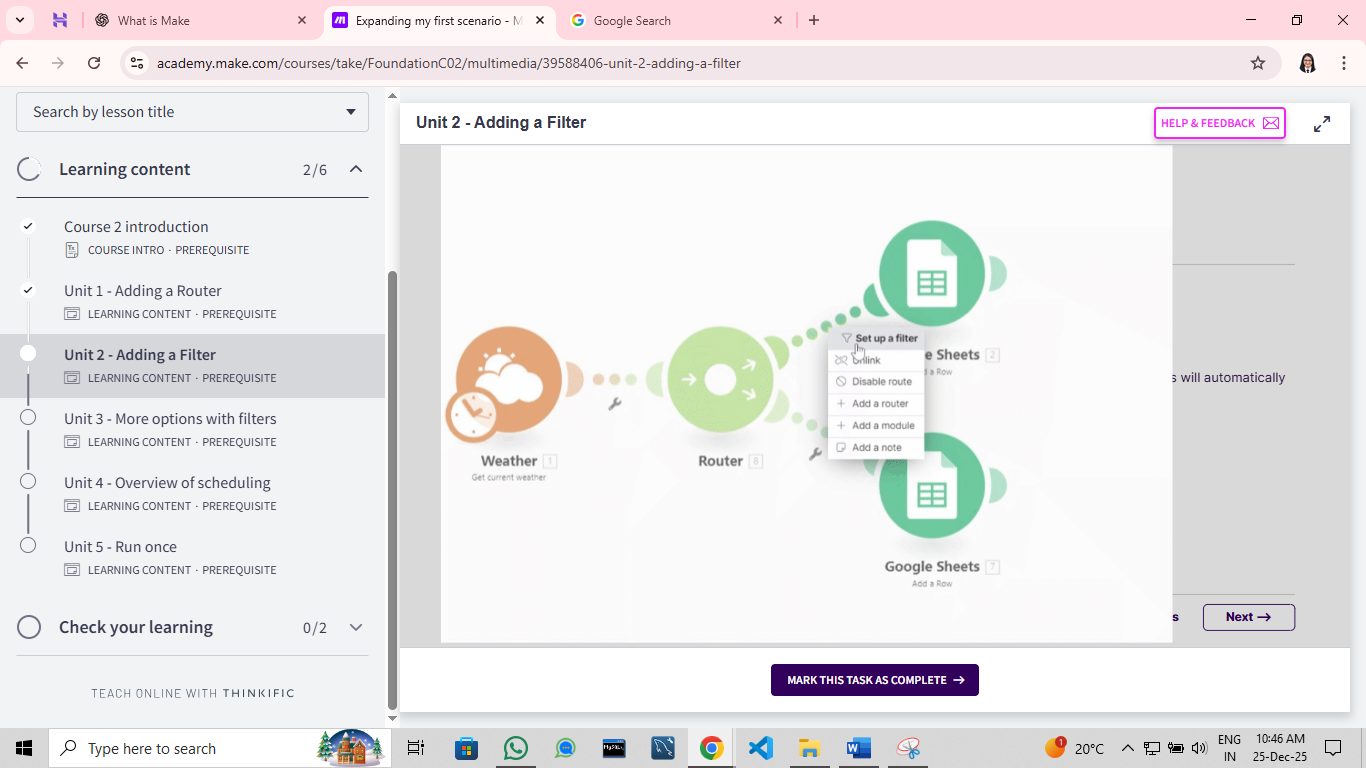
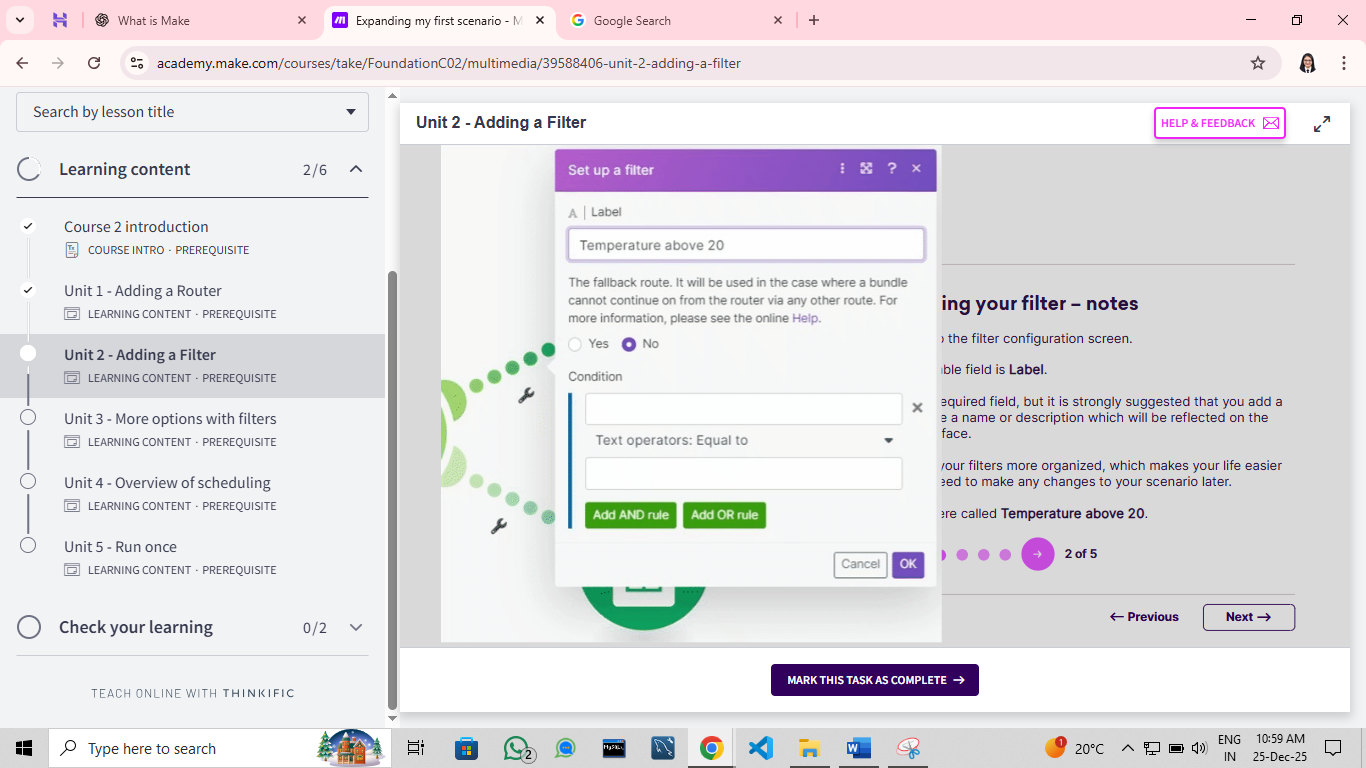
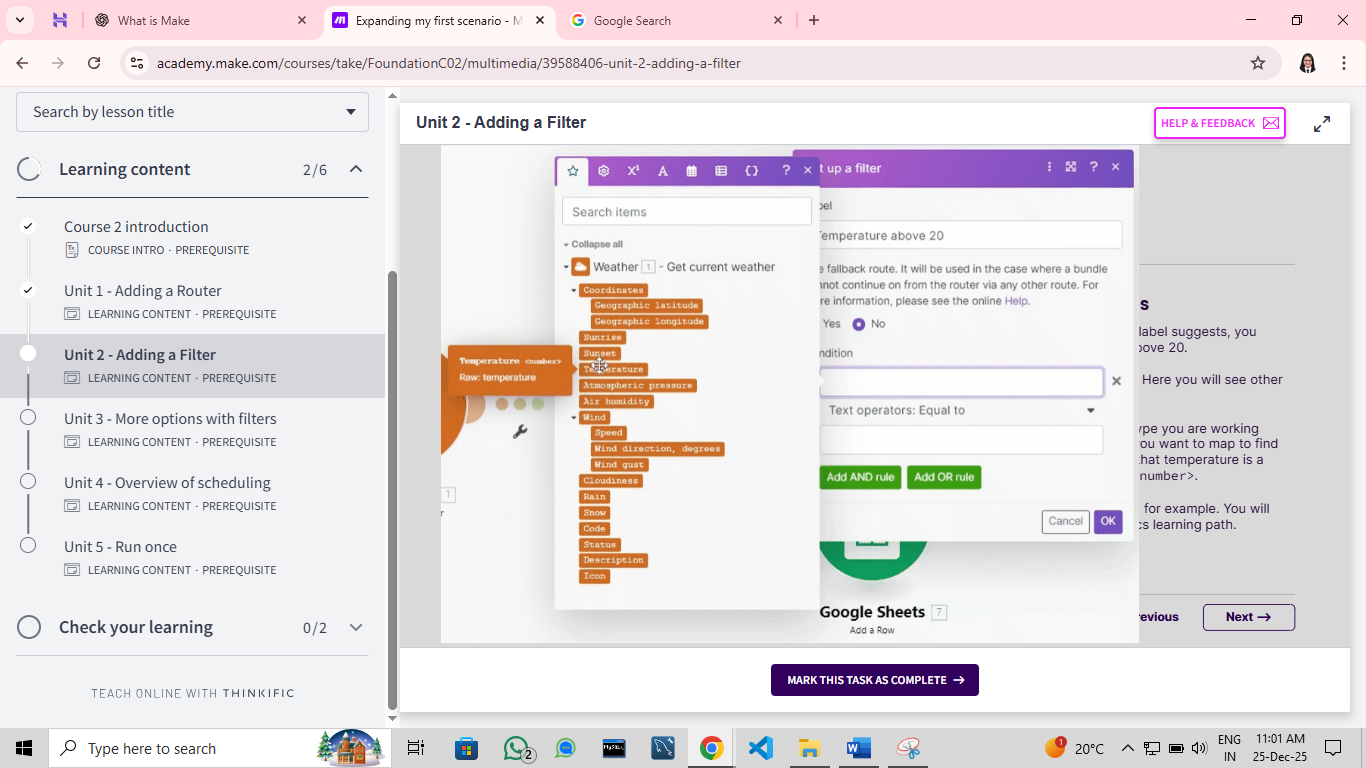
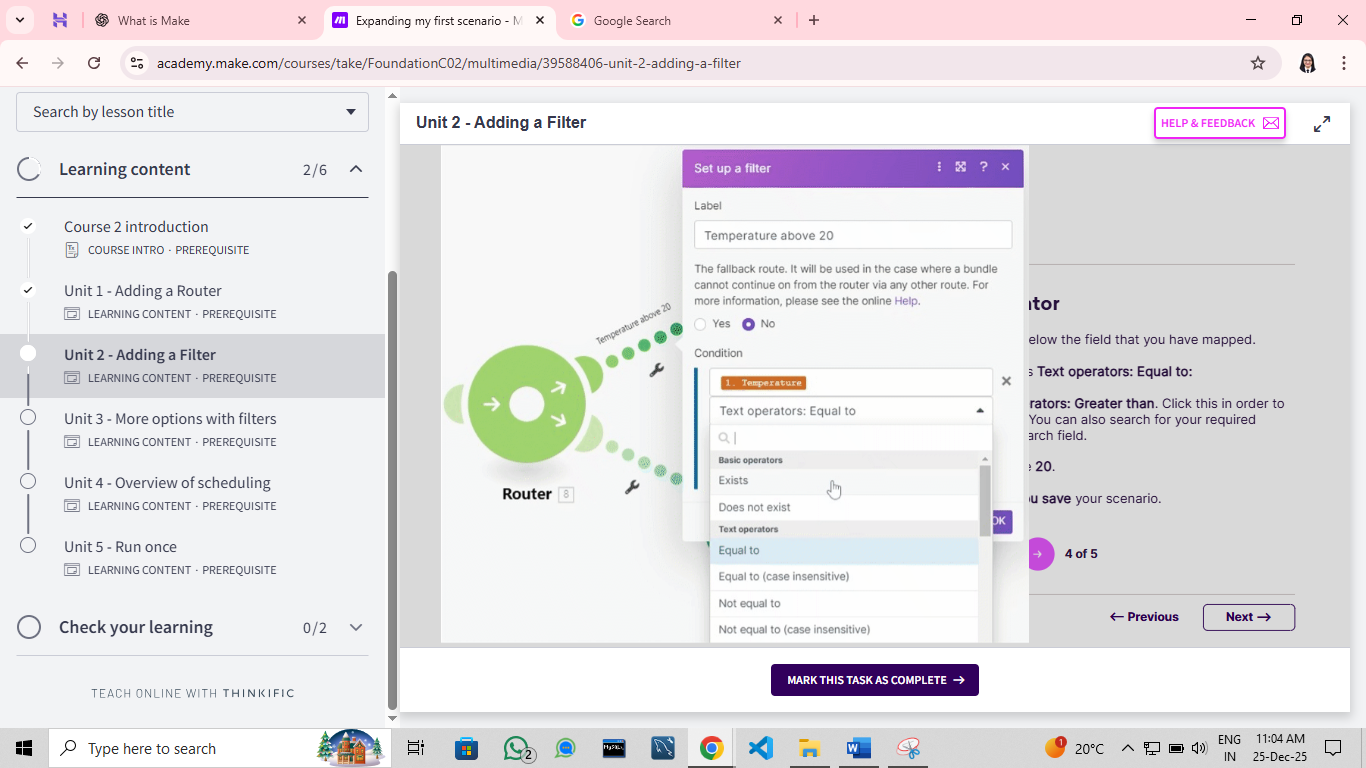
1. **What kind of information can be filtered?**

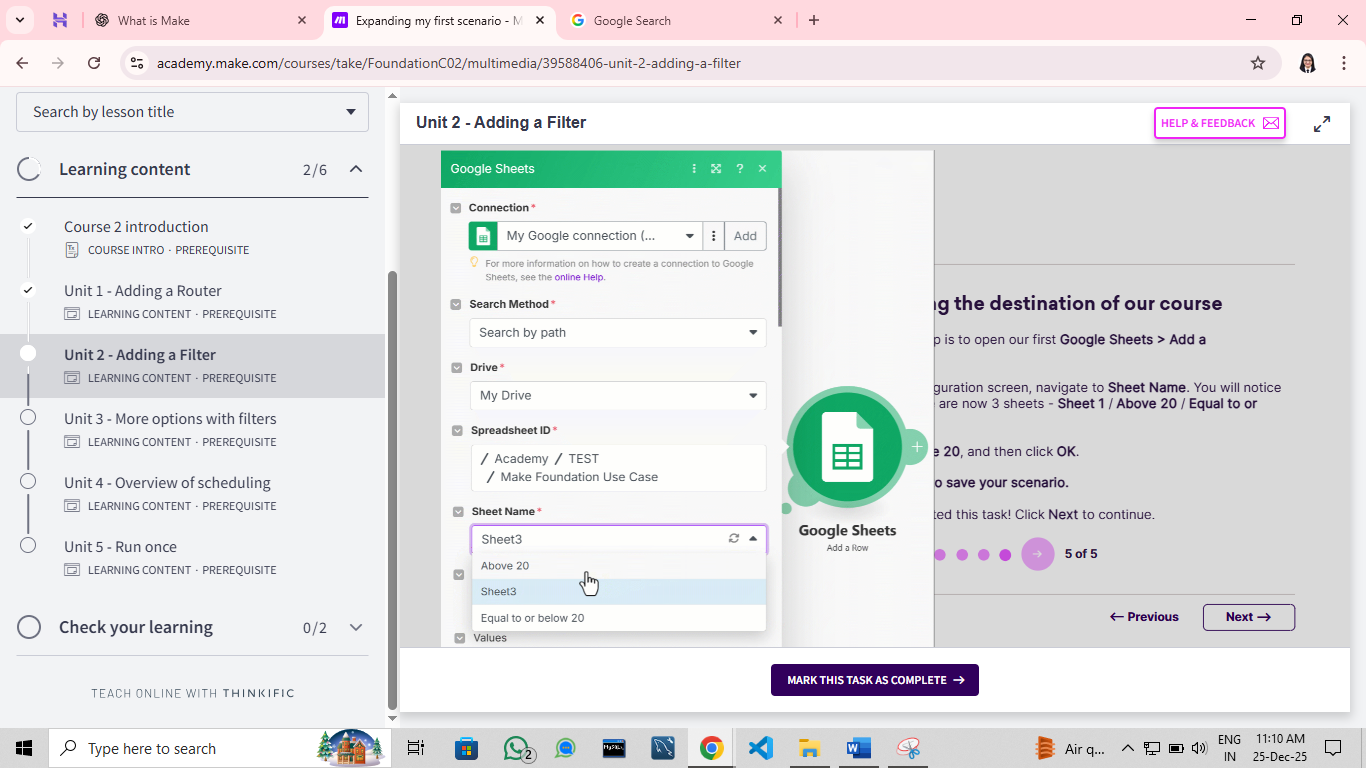
* **Basic Operators:** Data will be passed through the filter if it does / does not exist.
* Example: pass data in your scenario if the word Make exists in an email address.
* **Text Operators:** Text operators give us many options to work with. Let's use the example of a social media post:
* A scenario watches a social media account for posts, and can filter data that contains text:
* Equal to: "Make Academy", or
* Contains the word: "Make Academy" or
* Does not contain the word: "Make Academy" or
* Starts with the word: "Hello"
* **Numeric Operator:** Numeric operators are useful for returning numeric values. Let's consider the use of your Make Foundation scenario:
* Your scenario watches the weather, and can filter data based on the temperature if it is:
* Equal to: 20 degrees, or
* Greater than: 20 degrees, or
* Less than: 15 degrees, or
* Greater than or equal to: 25 degrees, or
* Less than or equal to: 10 degrees
* **Datetime Operator:** Datetime operators will filter results based on a defined time. As an example, imagine you have an e-commerce store:
* You want to filter orders made outside of your working hours, so you can prioritize them the next day.
* You could filter:
* Later than: 5PM, or
* Earlier than: 7AM, or
* Later than or equal to: 5PM (this would be useful for any orders made at exactly 5PM)

1. **Before you add first filter:**

* For the purpose of this next exercise, you will need to add two more sheets to Google Sheets.
* Open your Google Sheet named Make Foundation Use Case, and add two additional sheets.
* Name these Above 20.
* You will find out why shortly.

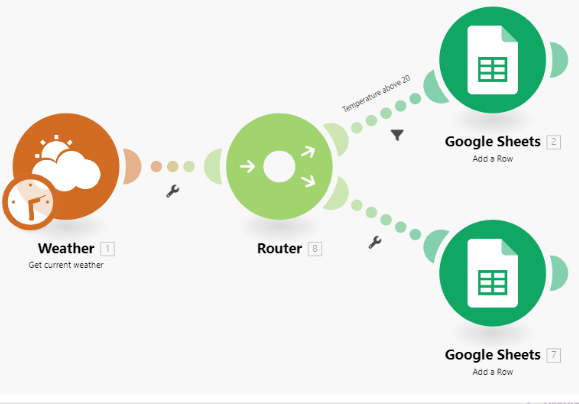
1. **Adding your first filter:**

* **Ways to add a filter:** Filters can be added in two ways.
* 1: left-click on any route between two modules. This will automatically load the filter prompt.
* 2: right-click on any route between two modules. Select Set up a filter.
* **Configuring your filter – notes:** This brings up the filter configuration screen.
* The first editable field is Label.
* This is not a required field, but it is strongly suggested that you add a label. It can be a name or description which will be reflected on the scenario interface.
* Labels make your filters more organized, which makes your life easier in case you need to make any changes to your scenario later.
* Add a label here called Temperature above 20.
* **Adding conditions + data types:** Next you are going to add a condition. As the label suggests, you want to add a condition that temperature is above 20.
* In the first condition field, map Temperature. Here you will see other items that can be mapped as part of the filter.
* NOTE: It is useful to know what kind of data type you are working with for your filters. Hover over the field that you want to map to find out the data type. In the case here you know that temperature is a numeric value, this is shown in the format of <number>.
* Other values here can be <date> and <text> for example. You will learn more about data types in the Make Basics learning path.
* **Selecting an operator:** Next select the dropdown below the field that you have mapped.
* By default this will display as Text operators: Equal to:
* Scroll down to Numeric operators: Greater than. Click this in order to populate the operator field. You can also search for your required operator by typing in the search field.
* Finally in your last field, type 20.
* Click OK. Make sure that you save your scenario.



* **Changing the destination of our course:** The final step is to open our first Google Sheets > Add a row module.
* On the configuration screen, navigate to Sheet Name. You will notice in here there are now 3 sheets Sheet 1/ Above 20.
* Select Above 20, and then click OK.
* Remember to save your scenario.

1. **Here is how your scenario should look:**

* Here is how your scenario should now look: you'll notice that the settings icon has changed from a wrench to a filter.
* **For your next task,** add a filter to the second module that adds temperature less than or equal to 20.
* You have now enhanced your scenario by adding filters it will process information to two different sheets based on the values that you have specified - click Run once and see it in action! Go to your Google Sheet to see what happened.
* Note: here would be a great opportunity to try using the Auto-align / Explain flow tool in your scenario.
* This will give you an idea of the order in which your automation will be processed.

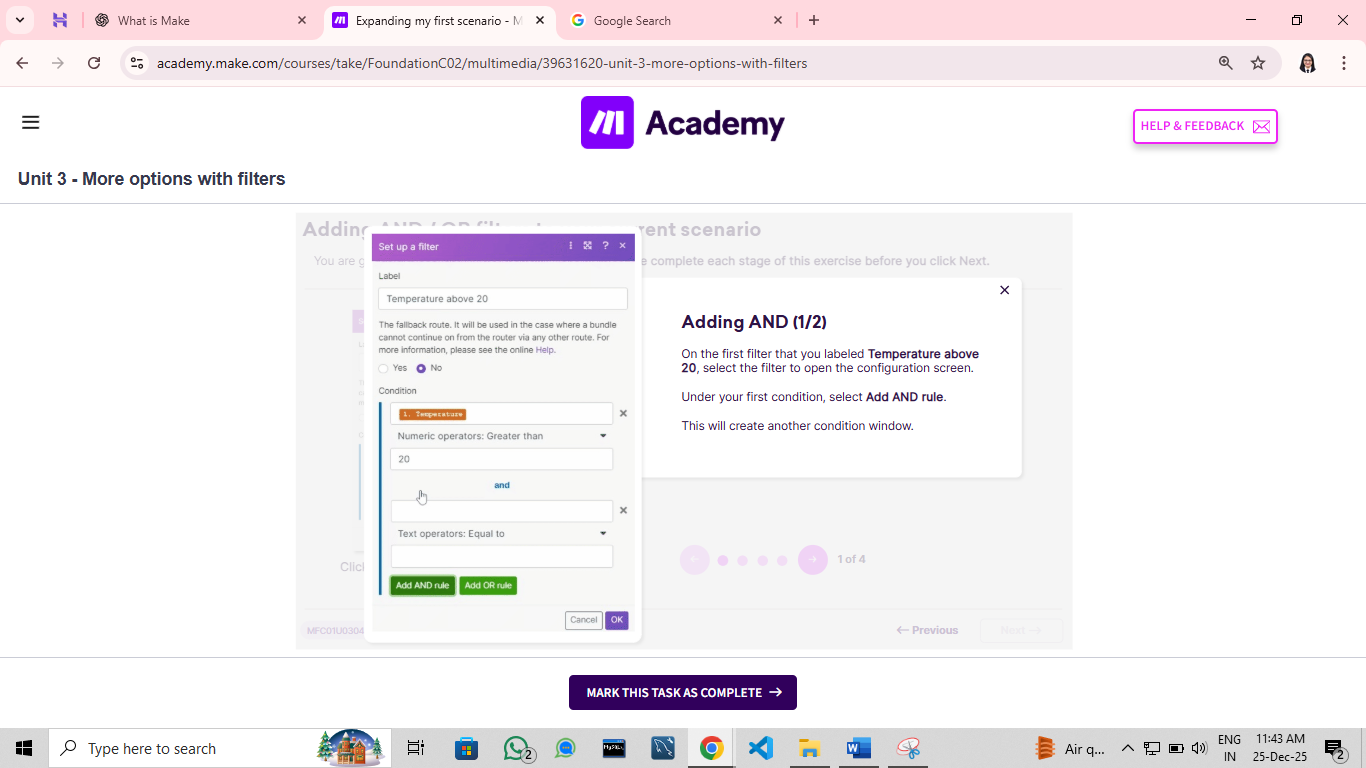
**Foundation - More options with filters**

1. **Quick recap on your current filters.**

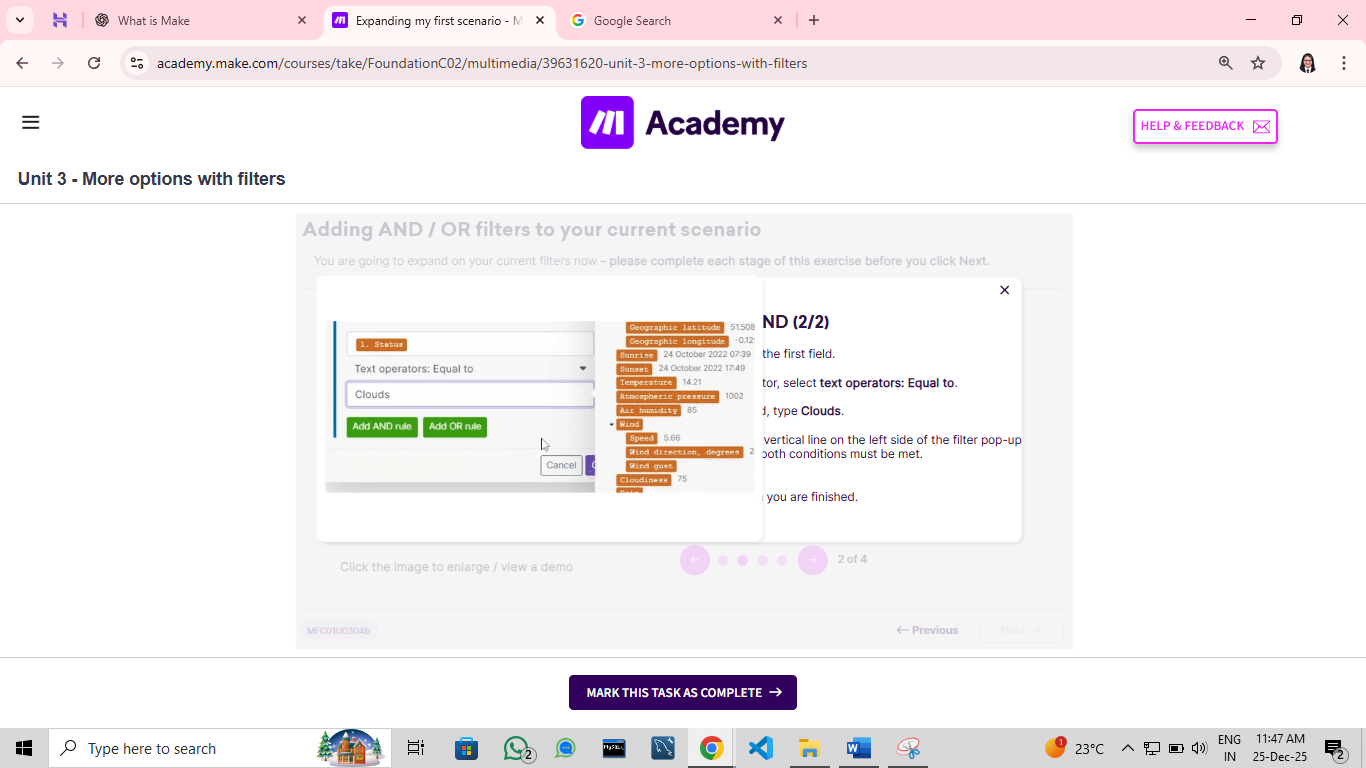
* The current scenario you have built will filter data from the Weather app on to two separate paths:
* If the temperature is above 20, it will pass to a sheet named Above 20.
* If the temperature is equal to or below 20, it will filter to a different sheet named Equal to or below 20.
* When you were configuring your filters previously, there were two options under condition that were purposely ignored. These are:
* Add AND rule
* Add OR rule
* You are now going to implement both of these filters shortly.

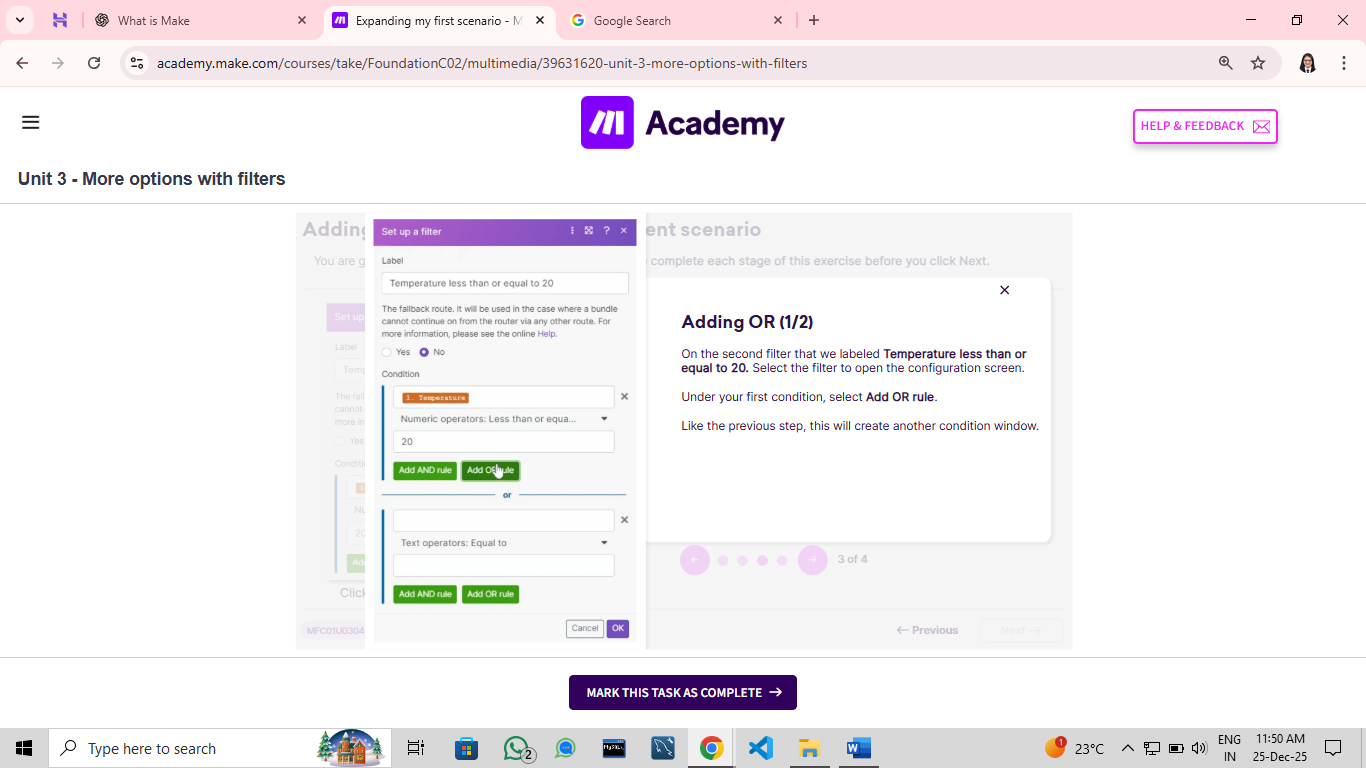
1. **Adding some AND/OR filters to your current scenario**

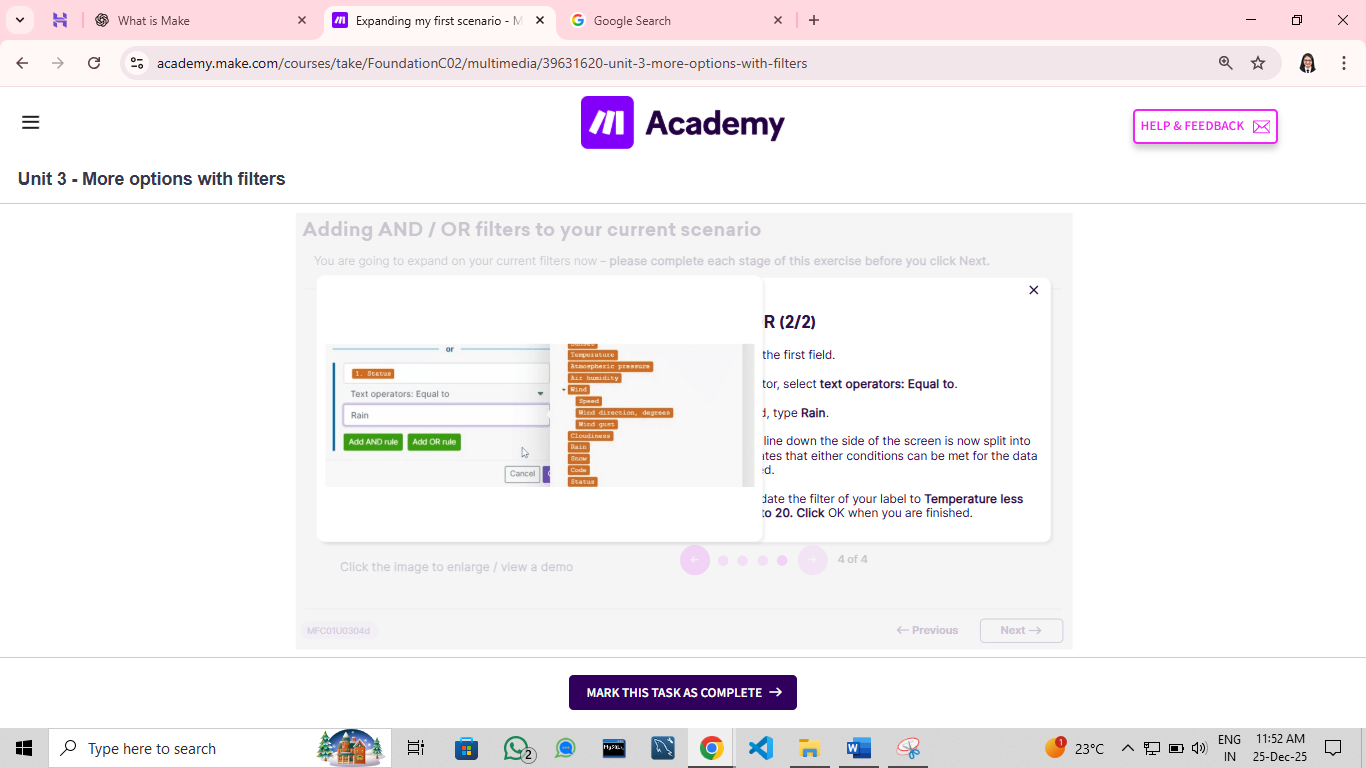
* Using **AND** will add an additional level of passing criteria to your filters.
* For your current scenario, your top filter will pass the data to the next module if the temperature is above 20.
* Using AND you could expand this to:
* Temperature is above 20
* AND Status = clouds
* AND Air humidity is greater than 50
* The data will only be processed to the next module if it fulfils all three of the criteria.
* Using **OR** will add an additional option of filtering data.
* For your current scenario, your bottom filter will pass the data to the next module if the temperature is less than or equal to 20. Using OR you could have additional options such as:
* Temperature is equal to or less than 20
* OR Status = rain
* OR Air humidity is less than 50
* If the required data meets any of the criteria, it will be processed to the next module.

1. **Adding AND/OR filters to your current scenario**

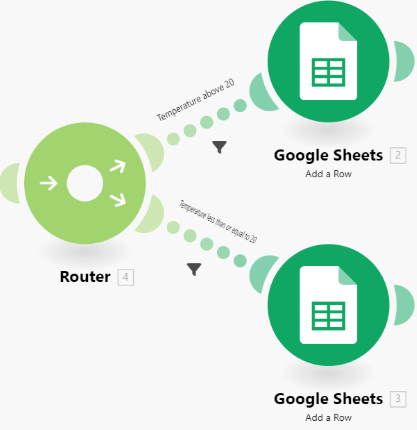
* **Adding AND (1/2):** On the first filter that you labeled Temperature above 20, select the filter to open the configuration screen.
* Under your first condition, select Add AND rule.
* This will create another condition window.



* **Adding AND (2/2):** Map Status in the first field.
* For your operator, select Text operators: Equal to.
* In the final field, type Clouds.
* Note: the blue vertical line on the left side of the filter pop-up indicates that both conditions must be met.
* Click OK when you are finished.
* **Adding OR (1/2):** On the second filter that we labeled Temperature less than or equal to 20. Select the filter to open the configuration screen.
* Under your first condition, select Add OR rule.
* Like the previous step, this will create another condition window.

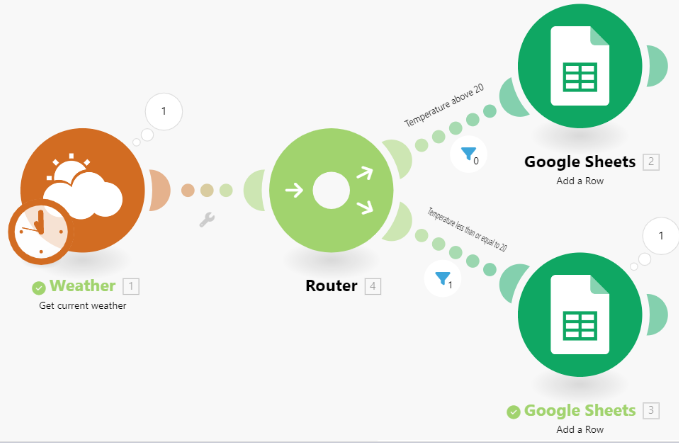


* **Adding OR (2/2):** Map Status in the first field.
* For your operator, select Text operators: Equal to.
* In the final field, type Rain.
* Note: the blue line down the side of the screen is now split into two; this indicates that either conditions can be met for the data to be processed.
* Ensure you update the label of your filter to Temperature less than or equal to 20. Click OK when you are finished.

1. **Summarizing your module additions:**

* You have now added another level of criteria to your filters within your scenario.
* The top module will:
* Process data if the temperature is above 20 AND equal to clouds.
* The bottom module will:
* Process data if the temperature is less than 20, OR equal to rain.
* Following up on our previous concert analogy, this is like saying "You can enter if you have a ticket, and you are wearing shoes. If you are not wearing shoes but you have a ticket, you can't come in!".
* The next step is to Run once the output will vary based on what the weather is like where you are.

1. **Look at the result:**

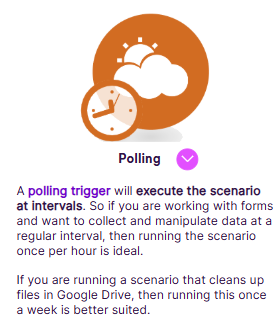
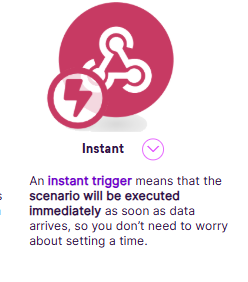
* The last thing to do is to run the scenario by clicking Run once.
* Depending on where you are in the world or at what time, you will get a different result.
* The screenshot here reflects what happened when we ran our module.
* Because the temperature was less than 20, it filtered to .
* Look closely at the filter icons; the bottom one reflects that one bundle of data was passed through this filter. The top filter did not meet the criteria, and therefore was not processed.
* At this point we recommend duplicating your scenario and playing with other outputs; for example setting up additional tabs in your Google Sheet and adding criteria for different types of weather/temperature combinations.

**Foundation - Overview of scheduling**

1. **What is scheduling ?**

* Scheduling allows you to define the specific time and frequency at which you run your scenarios.
* There are many options available here, and these will be based on the needs, type, and purpose of your scenario.
* You can run your scenario:
  + At regular intervals such as 15/30/45 minutes
  + Once - a one-time execution on a specified date / time
  + Every day at a set time
  + Days of the week at a specific time
  + Days of the month at a specific time
  + Specified dates / time
  + Immediately when data arrives
  + On demand via the Run once button or an API call

1. **How often should I schedule my scenario?**

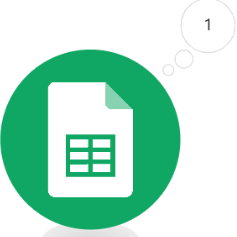
* It all depends on the types of scenario that you are running:
* A polling trigger will execute the scenario at intervals. So if you are working with forms and want to collect and manipulate data at a regular interval, then running the scenario once per hour is ideal.
* If you are running a scenario that cleans up files in Google Drive, then running this once a week is better suited.
* An instant trigger means that the scenario will be executed immediately as soon as data arrives, so you don't need to worry about setting a time.

1. **Credits count**

* The key factor to remember is that every time your scenario runs, it will perform operations and consume credits towards your total credits count.
* A quick breakdown of your credits count on your foundation use case:
* The Weather app retrieves the current weather = 1 credit.
* The Google Sheets app adds a row = 1 credit
* = 2 credits total
* Therefore if you schedule the scenario to run every 15 minutes, this would total:

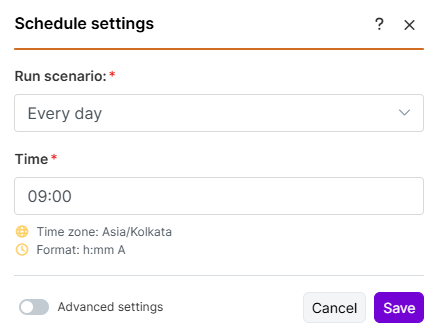
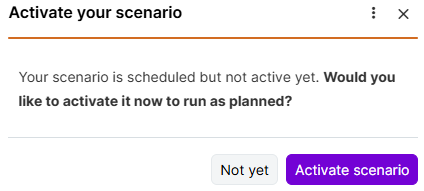
8 credits per hour

192 credits per day

1,344 credits per week.

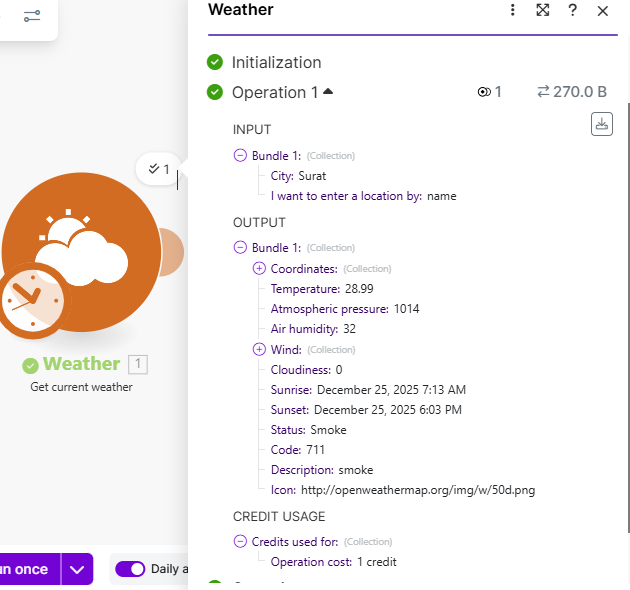
* We will cover operations and credits in greater detail in the next course.

1. **How to schedule scenario:**

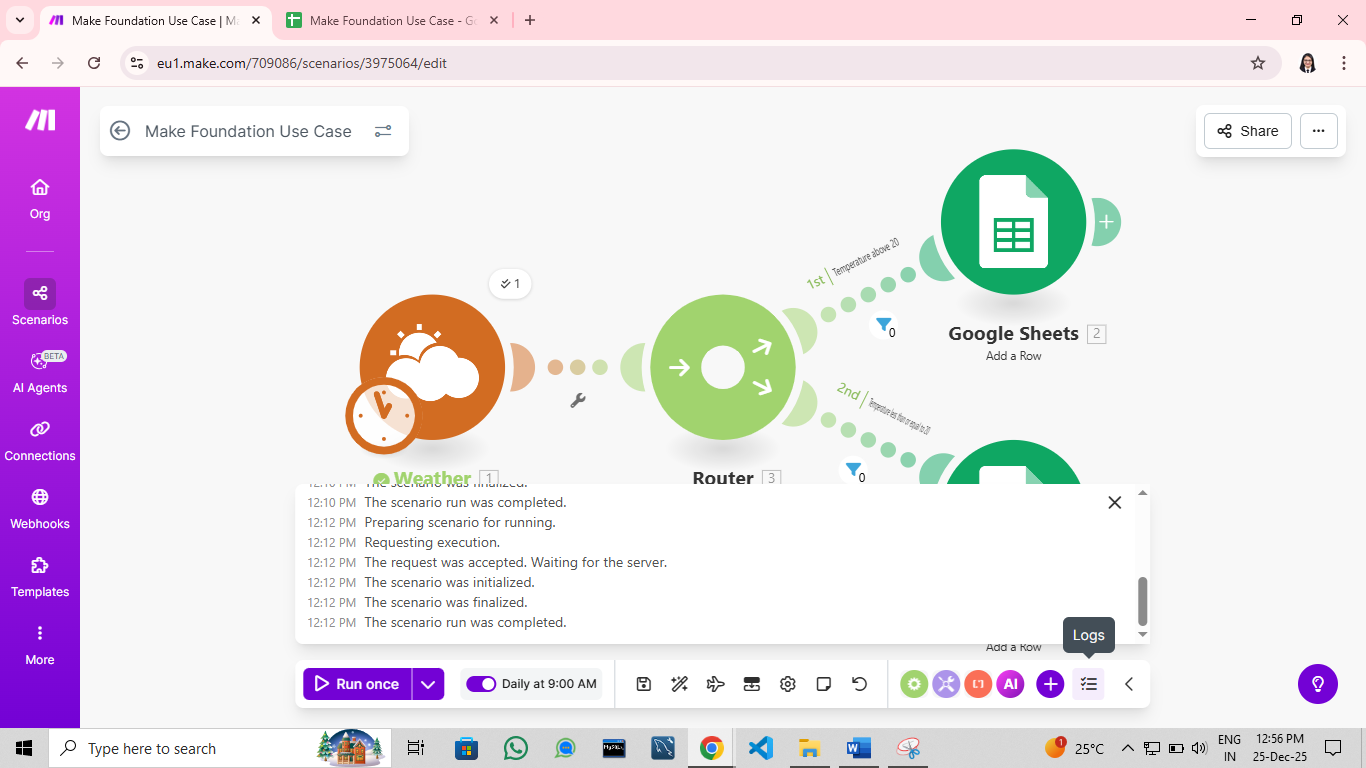
* Scheduling your scenario is a straightforward process, let’s explore this now:
* **Before you decide to schedule:** The most crucial thing with scheduling your scenario is ensuring that it is working properly.
* This means ensuring there are no errors when you select Run once, and that the data that you are working with is going where it needs to go.
* Once you are happy with your scenario, let's schedule it.
* **Setting the interval:** You can access scheduling in the bottom left hand corner of the scenario builder. Clicking Scheduling will bring up the configuration pop-up.
* By default At regular intervals is selected. In the dropdown here you can select your various options as previously mentioned.
* For your scenario, let's select Every day, and then type 9:00. Click Save.
* NOTE: this will consume 14 credits in a week.
* **Toggle ON/OFF:** The last step to activate the scenario is in the bottom left hand corner of the scenario builder.
* After you save your interval settings, click Activate Scenario in the pop-up window. Then save your scenario.
* You will know that your scenario is active when the ON/OFF toggle is purple.
* Note: Scenarios can also be toggled ON/OFF from the scenario menu.

**Foundation - Run once**

1. **Reviewing what happens when you select ‘Run Once’:**

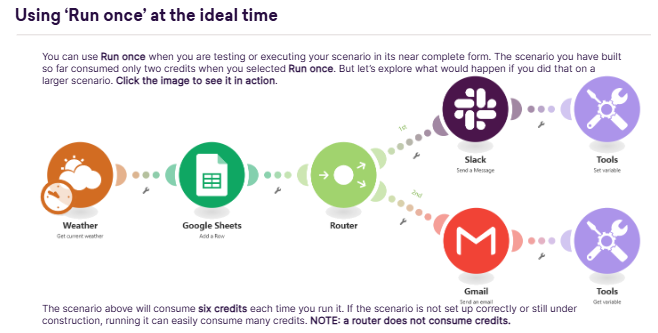
* **Run Once:** Selecting the Run once button will execute your entire scenario - this will retrieve the weather, and route the data to one of two paths.
* Note that the number of credits depends on the filters you may have set up. For example, running this scenario once would normally consume two credits but depending on the filter, it could possibly consume only one credit.
* Data is generated from your source module. This will get the weather from London (or the location you have specified), and generate an output bundle: temperature, sunrise, sunset, status, etc.



* The data you have mapped is passed along the scenario to the router.
* It will first be passed to the top module and is met by a filter.
* If it does not meet the filter criteria, it will not be passed any further.
* It will then be passed to the bottom module. In the example here, it meets the criteria and is passed to the module.
* Parts of the data that you have generated in the source bundle is mapped to the Google Sheets module.
* Out of all items you generated in the source bundle, you are only mapping four of them here.
* They will be reflected in the Make Foundation Use Case Google Sheet.
* After selecting Run once, the scenario will consume a total of two credits only if the filter criteria is met. Otherwise, it will consume one credit.
* Note that the scenario log will reflect the running order of a successful/ failed scenario run.
* Any errors encountered in the scenario will appear in red.

1. **Using 'Run once' at the ideal time**

* You can use Run once when you are testing or executing your scenario in its near complete form. The scenario you have built so far consumed only two credits when you selected Run once. But let's explore what would happen if you did that on a larger scenario.

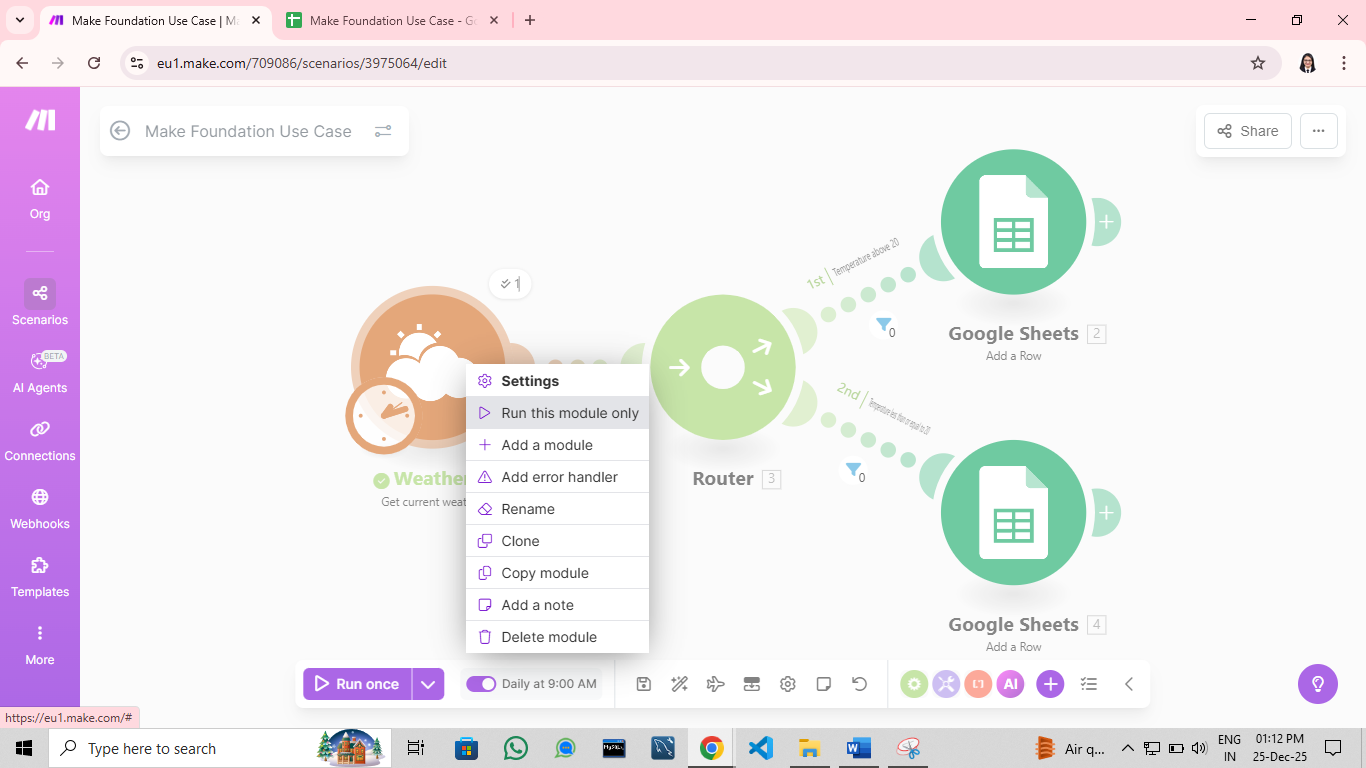


* The scenario above will consume six credits each time you run it. If the scenario is not set up correctly or still under construction, running it can easily consume many credits. NOTE: a router does not consume credits.

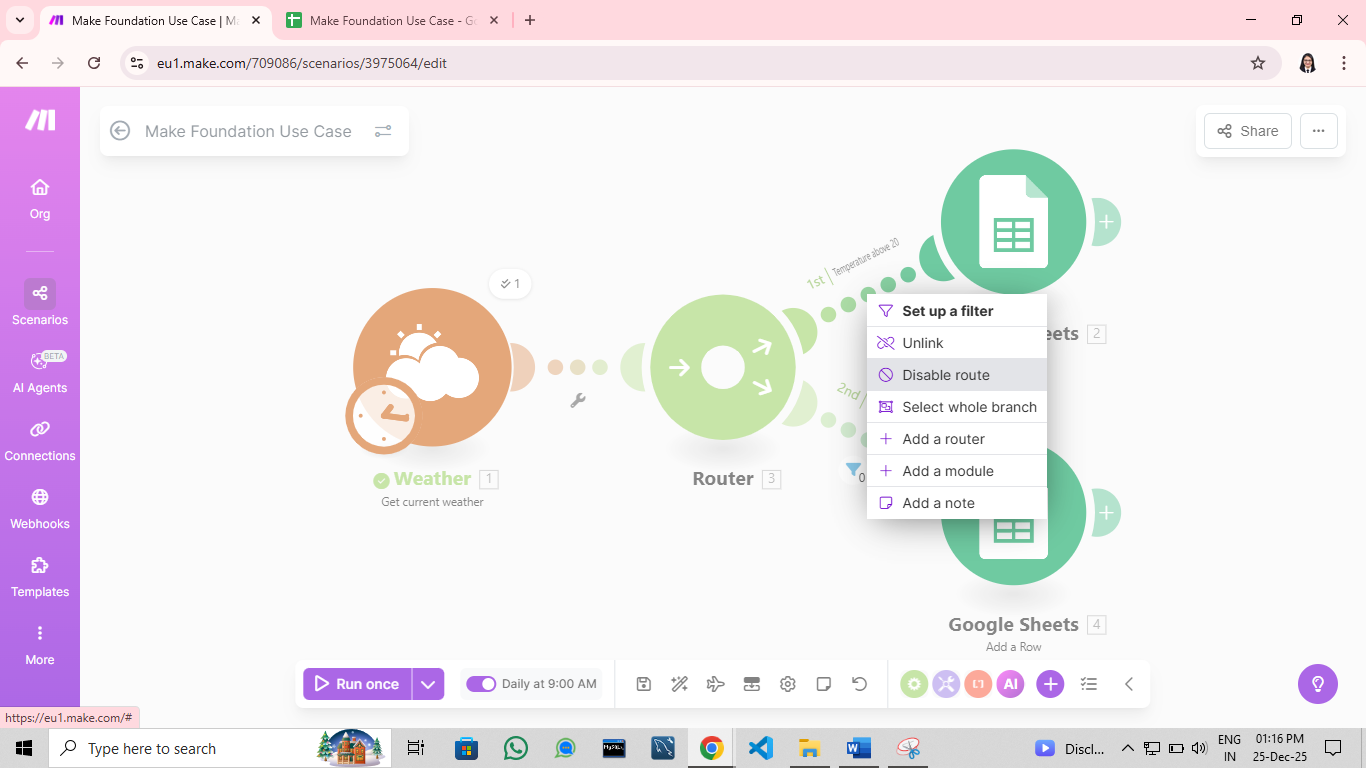
1. **Using ‘Run Once’ at the ideal time**

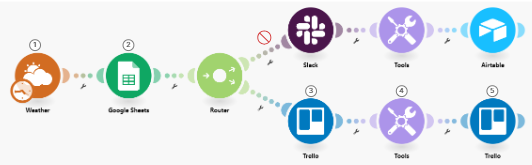
* Another ideal use of the Run once button could be in a scenario where you need to transfer data one time only.
* For example, the modules listed here will return all specified records from Airtable and put them into a Google Sheet.
* In this instance you would not need to schedule this to run at set intervals, you just need to run the scenario one time.

1. **How can I save on credits?**

* The safest way to ensure that you do not consume unnecessary credits is to be aware of this helpful function:
* By right-clicking any of your modules you can opt to Run this module only.
* This is especially useful on your source module, as it will show you the bundle of information that you are working with going forward.
* It is also incredibly handy to see what your output bundle looks like.

1. **Disabling routes**

* The bigger a scenario is, the more operations it will perform, and the more credits it will consume when testing.
* A way to navigate this is by disabling the routes. This will allow you to isolate part of your scenario and consume less credits when testing.
* To do this, right-click between any route between the modules and select Disable route.
* The status of the icon will change from to 
* This will stop any further data from passing through.
* With the scenario displayed here, the Run once would only consume 5 credits; you can disable multiple routes to further save on your credits just remember to enable the route when you are ready to fully test (Right click > Enable route).



**Introduction - Introduction to operations, credits**

**Introduction - What are bundles**

**Introduction - How to schedule your scenario**