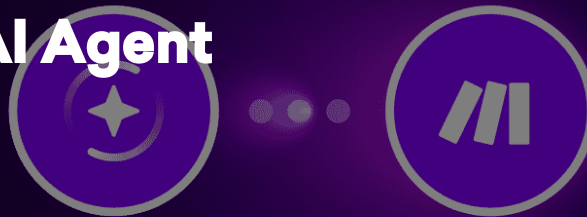


Unit 1 - Make AI Agent



☰ 1.1 Unit introduction

☰ 1.2 AI agents in Make

☰ 1.3 Scenario

☰ 1.4 Run an agent

☰ 1.5 Tools & knowledge

☰ 1.6 Testing and training

☰ 1.7 Wrap up



Unit 1 Make AI Agent

1.1 Unit Introduction

You are at the fourth unit of the course **AI agents for beginners**.

Now that you know everything about AI agents and automation, it's time to see how you can do this in Make.

In this unit you will learn:

how AI agents work in Make

what you need to build an AI agent in Make

very precious tips and tricks

Time to continue your investigation!

[Continue to 1.2: AI agents in Make](#)



1.2 AI agents in Make

In Make, you can create AI agents and include them in your automations.

If you want to use agentic automation in Make, the first step is to clearly decide **what you want your AI agent to do**. You need to **plan** carefully: the objective of your AI agent and what information and tools it needs to get it done.

Best practices



Start small. Begin with a **single, well-defined task**, like scheduling appointments or handling one type of customer request. This makes it easier to test and improve your AI agent. You can also make sure it works well before adding more functions.

Once you have a clear idea of what you want your AI agent to do, you can start building it.

You only need a Make scenario to build, configure and run your AI agent.

You will see how to build and run an AI agent in the next section. First, let's look at the different elements and characteristics of AI agents in more detail.

Continue to 1.3: Scenario



1.3 Scenario

You build your AI agent inside a Make scenario.

In this scenario, **you define**:

- How the scenario **starts** (manually, at regular intervals, or immediately when it receives data)
- If your AI agent is **part of a Make scenario** with modules that run before it
- Your **AI agent** setup: the LLM, instructions and input, plus any advanced settings
- The **tools** the AI agent can use
- Any **knowledge** the AI agent has access to

When building a scenario to run your AI agent, you have to:

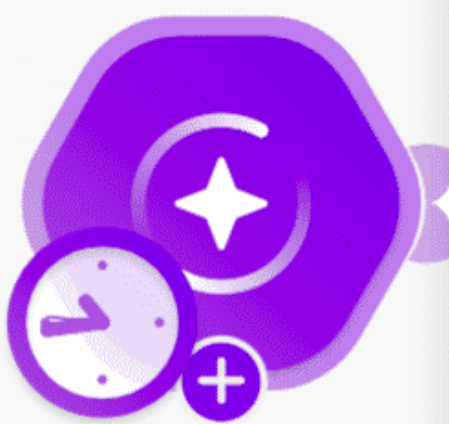
- Create a new **scenario**
- Add the **trigger** that defines how your scenario will start
- Add any relevant **modules** that provide information to the AI agent or perform a specific action before running the AI agent
- Add the **Run an agent** module and set it up
- Add any **tools** the agent needs
- Add the **knowledge** and upload or select any relevant files

Let's take a closer look at the **Run an agent** module and how to give your AI agent tools and knowledge.

[Continue to 1.4: Run an agent](#)



1.4 Run an agent




Make AI Agent

Run an agent

Make AI Agent

⋮ ✕ ? ×

> **Connection ***

 My Make's AI Provider ... ⌵ ⋮ Add

For more information on how to create a connection to Make AI Agent, see the [online Help](#).

> **Model *** ↻ Refresh Map

Recommended: Small ⌵

> **Instructions**

You are an agent specialized in...

Describe your agent's role and how it should behave. Define its goals, steps it should take, and any rules it should follow to do its job.

> **Input ***

YOUR PROMPT HERE

This is the incoming data or task your agent will work on. You can map values from previous modules here, such as new chat messages or emails.

> **Input files** Map

⋮ + Add file

Include a file for your agent to process with its incoming task. JPG, PNG, GIF, and PDF formats are supported. Large files may incur significant credit consumption.

> **Conversation ID**

This allows the agent to reference a conversation's history and reply in the same chain. The conversation ID is typically a Thread ID or Timestamp.

Advanced settings Cancel Save

Use the **Make AI Agent > Run an agent** module to set up and run your AI agent.

In this module you can:

- Connect the AI agent to an AI provider and select the **LLM** it will use
- Write the **instructions** and **input** that tell the AI agent what to do.
- Define any **AI agent settings**

Let's have a look at each of them.

[Continue to 1.4.1: Connection and model](#)

1.4.1 Connection and model

An AI agent needs an LLM to think, understand, and make decisions. LLMs come from AI providers, so you first need to create a connection to one of these providers to assign a model to your AI agent.

In Make, you can choose between two types of AI provider connections:

1

Make AI Provider:

Make includes a built-in AI provider, so you can give your AI agent access to an LLM without subscribing to a third-party service. Every type of organization has access to it.

Here you can find details about how it works, how many credits it costs, and which model you can choose from.

You will be using this option when building the use case.

CREDITS AND TOKENS

2

Your own connection:

You can also connect your AI agent to your own AI provider. This option is only available on the Pro plan and above.

Here you can find a list of AI providers you can choose from:

Make's AI Provider
OpenAI
Anthropic Claude
Gemini AI
Azure Foundry (OpenAI Models)
Azure Foundry (Models from other providers)
Mistral
Cohere
Groq
xAI
Amazon Bedrock
Other Providers (Compatible with OpenAI API)

Once you've set up a connection with an AI provider, you can choose the LLM model that your AI agent will use.

There are many LLMs you can use for your AI agent, and their costs depend on things like how fast they work and how well they can reason. When you're starting, pick one that's both quick and affordable. More advanced models can think in more depth but are slower and cost more.



Different models can produce different results and influence how your AI agent behaves. Check the AI provider's information to choose the model that best fits your needs.

Best practices



Begin with the most advanced model your provider offers. This shows you what's possible and gives you a baseline to compare against. Test your agent thoroughly and note how well it performs.

Once you know what great looks like, try smaller or faster models to see if they still meet your needs. Move down step by step until you find the sweet spot between performance and cost.

When picking a model, think about:

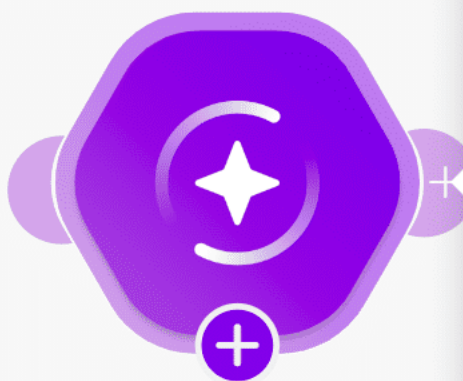
- **What your task needs:** Simple categorizing or routing works with smaller models. Complex decision-making or multi-step reasoning needs more advanced models.
- **Cost:** How much it costs to process data and generate results.

- **Speed:** Fast models work best for chat or real-time tasks. Slower models are fine when end users aren't waiting.

Check your provider's documentation to understand what each model does best. Some excel at writing, others at logic or problem-solving. The right model depends on your specific task.

Continue to 1.4.2 Instructions

1.4.2 Instructions



Make AI Agent

Run an agent

A screenshot of the Make AI Agent configuration interface. The window title is "Make AI Agent". It has a "Connection *" section with a dropdown menu showing "AA_C06" and an "Add" button. Below this is a link to "online Help". The "Model *" section has a "Refresh" button and a "Map" toggle, with a dropdown menu showing "Recommended: Large". The "Instructions" section has a text input field containing "You are a detective that investigates robberies." and a description: "Describe your agent's role and how it should behave. Define its goals, steps it should take, and any rules it should follow to do its job."

This is where you define the **instructions** that tell your AI agent its **role**.

It explains how the AI agent should act and what to focus on.



The instructions represents the system prompt of your AI agent.

Keep the instructions general so your AI agent can do many kinds of tasks, not just one. You want to be able to reuse your AI agent. Broad instructions let the AI agent handle many types of tasks, while very specific ones limit it. You can give more details later in the input.

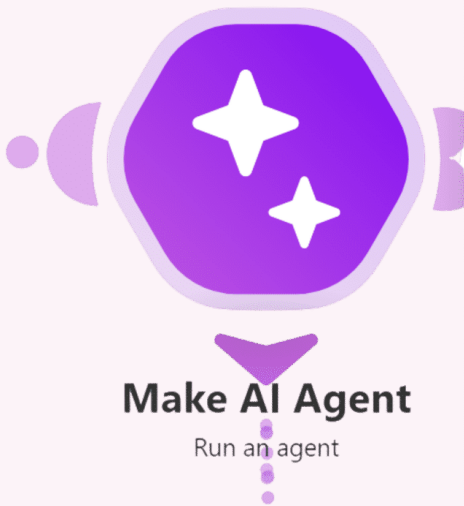
Best practices



Write a clear and direct description. The AI agent's description guides its decisions. Be clear about what it should do.

Continue to 1.4.3 Input

1.4.3 Input



The Very Precious Gem was stolen from the Every Bracelet You Take Hotel, near the lobby, between 4:00 p.m. and 5:00 p.m. There are three suspects: Crystal, Tyler, and Fred. Here is a pipe-separated list of the alibis for the three suspects: `"" text ""`. Try to determine who committed the theft.

- Proceed by eliminating suspects who have a valid alibi.
- If only one suspect doesn't have a good alibi, conclude that he or she could be the one that stole the Very Precious Gem, and send an email with your report. In this case, don't ask follow-up questions.
- If more than one suspect doesn't have a good alibi, ask follow-up questions. Do not send the email in this case.

Message that will be sent to the agent. Use this field to map values from previous modules.

In the **input**, you specify what you want your AI agent to do.



The input represents the user prompt of your AI agent.

Best practices



When writing the input:

- **Be explicit:** clearly state the questions or tasks you want the AI agent to address.
- **Specify the output:** define the format and the structure of the AI agent output.
- **Include fallback instructions:** guide the AI agent on what to do if information is missing.
- **Test and refine:** iterate prompts based on results to improve accuracy.

Continue to 1.4.4: Settings

1.4.4 Settings

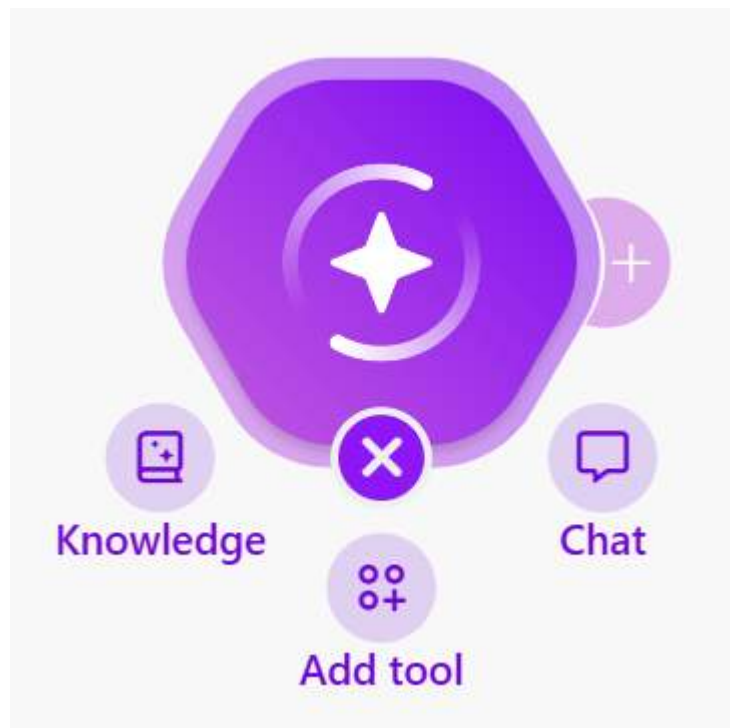
You can adjust some settings in the AI agent configuration to control how much your AI agent says, how many steps it takes to complete tasks, and how much past information it can remember.

You will learn about these options in a more advanced course.

Now you've set up your AI agent. Before running it you need to give it all the tools and knowledge it needs to carry out its task.

[Continue to 1.5: Tools & knowledge](#)

1.5 Tools & knowledge



Hover over the + icon of the **Run an agent** module to give your agent the tools and the knowledge it needs.

Let's find out more about them.

Continue to 1.5.1: Tools

1.5.1 Tools



Tools let AI agents use services from third-party applications, so they can take specific actions and complete their tasks.

A **third-party application provides services** that people or other applications can use. To let others access these services, it offers an **API (Application Programming Interface)**. An API is a set of rules, tools, and instructions that define how to connect to the service. It specifies where to send the request to access a specific service, what inputs it expects, and what information it returns.



For example, **Google Maps** provides mapping services and you want to include them in your website or app. The **Google Maps API** lets

you send requests containing an address, and then returns outputs like maps, directions, or nearby points of interest.

Derek bought a surveillance kit from Spying in the Dark. Their delivery app uses Google Maps so the driver can find the fastest route to Derek's police station. Security cannot wait!

Make apps and modules let you automatically connect to and make requests to third-party APIs.

You can use three different types of tools to access third-party applications:

- 1 Modules
- 2 Scenarios
- 3 MCPs

Let's have a look at each of them in detail.

Modules

Determine what action your agent needs to perform. If a Make module exists for that action, you can add it directly to the agent's tools.



To do so, hover over the **+** icon of the **Run an agent** module and select **Add tool**. Then select the module you want to add. Check the tool name and description, update them if necessary, then set up the module as you normally would.

Voilà! Your agent can now use this tool.



Note that you can set certain parameters to **Let the AI agent decide**. The module then uses whatever values the AI agent provides based on the current task.

Click **Add details** to give your agent more information about what to include.

2: Scenarios

2

Scenarios

When you want your AI agent to perform more complex actions, or to use more than one third-party application, build a scenario.

You can create scenarios containing the modules that you need, and give them as tools to your AI agent. This allows the AI agent to perform specific actions that the third-party applications provide.

They are **normal scenarios**, similar to the ones that you usually create in Make. For example, a scenario that **sends an email to a specific contact**.

This is what you need to do when building your tool scenario:

- Add a **description** of what the scenario does. The AI agent uses this description, along with the **scenario name**, to understand what the tool does and when to use it.
- Specify the **scenario inputs and outputs**. Inputs are the data that the scenario needs to run and that the AI agent needs to provide. Outputs define which data your scenario returns to the AI agent, which it then processes and uses.
- [Optional, but highly recommended] Add **specific modules at the beginning and at the end** to manage the information that goes in and out of the scenario.

Best practices



Use a clear, short description that tells the AI agent what the tools does and what data it should use. Keep it general so you can use the same tool with different AI agents.

You have seen that **scenario inputs and outputs** let you choose what information goes in and comes out of your scenario.

Click each one to discover more.

Scenario inputs —

They define the information the scenario needs to run and that it requires from the AI agent.

Derek the Detective is using a fingerprint analyzer. To start the analysis, he provides fingerprint samples he collected at the crime scene. Hoping to catch the criminal!

Scenario outputs —

They represent the data the scenario shares with the AI agent when it finishes running.

The machine found a match! It prints a result with Tyler the Thief's name on it. Derek is one step closer to catching the person who stole the Very Important Gem!



AI agent



Scenario

How does an AI agent use scenario inputs and outputs?

The data flows bidirectionally.

When the AI agent wants to call a tool, it knows which data it needs to provide based on the scenario inputs that you defined for this scenario. So it calls it with all the required information. In this case, the **data goes from the AI agent to the scenario.**

The scenario runs and produces results. It sends them back to the AI agent. The **data goes from the scenario to the AI agent.**

Tools need clear inputs and outputs. You define them in your scenario. At the start, they usually don't have any value or information, you just define the parameters described below. They act as **placeholders you can map in the scenario.**

Click the markers to see what you need to set.

Name  *

subject

✓ Use only letters, numbers, and underscores (_). Do not start with a number or _ followed by a number.

Description 

Subject of the email

Type  *

Text 

Name 

subject

✓ Use only letters, numbers, and underscores (_). Do not start with a number or _ followed by a number.

Description

Subject of the email

Type *

Text

Name

The **name** of the data (e.g. subject).

Name *

subject

✓ Use only letters, numbers, and underscores (_). Do not start with a number or _ followed by a number.

Description 

Subject of the email

Type *

Text

Description

The **description** of what the data contains (e.g. Subject of the email)

Name *

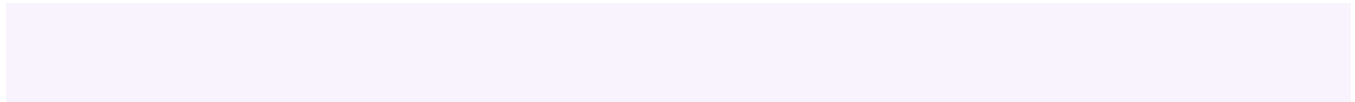
✓ Use only letters, numbers, and underscores (_). Do not start with a number or _ followed by a number.

Description

Type *

Type

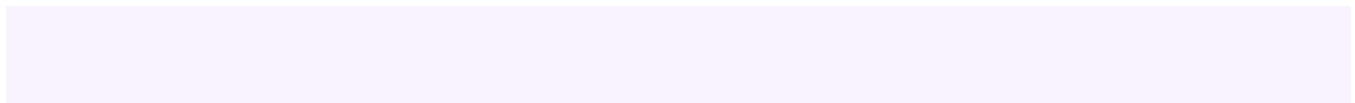
The **type** of data (e.g. Text)



Best practices

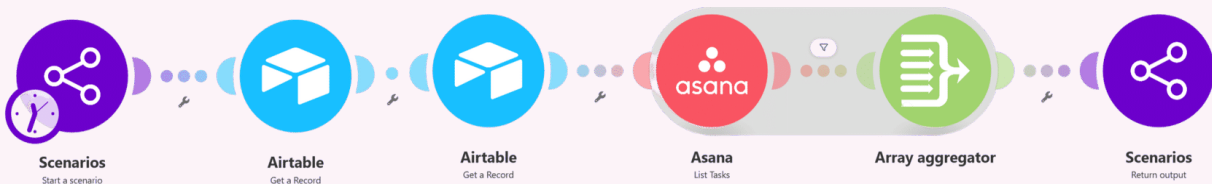


Use clear, descriptive names for scenario inputs and outputs so the AI agent knows exactly what data to receive and send.



You will learn how to set up scenario inputs and outputs when building the AI agent in the next section.

Now let's have a look at the **specific modules** that you can add at the beginning and at the end of your scenario to **help you manage the data**.



Every scenario used as a tool should have:

- **Scenarios> Run a scenario** module at the beginning
- **Scenarios> Return output** at the end

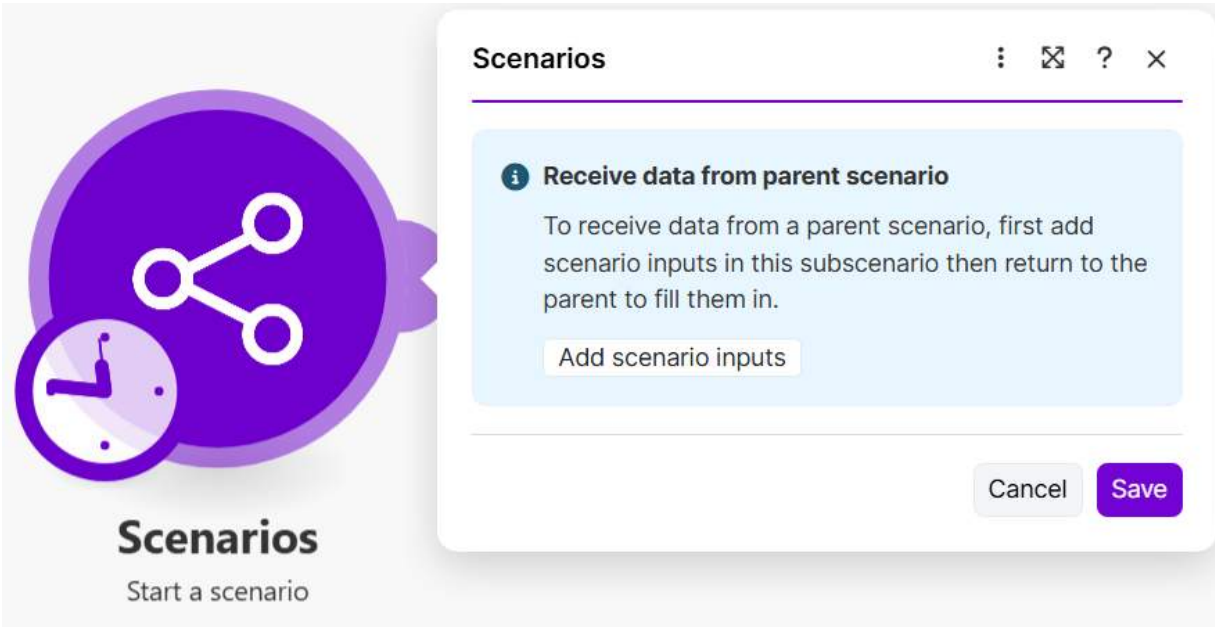
Click each tab to learn more about these two modules.

START A SCENARIO

RETURN OUTPUT

The **Start a scenario** module is a trigger module, so it goes at the start of the scenario. It's best practice to use it in an AI agent tool because it:

- Sets the scenario scheduler to **on demand**, which is required for a subscenario or a scenario used as a tool. This lets the AI agent run the scenario whenever it needs it.
- Makes the **scenario inputs available in the scenario**, so you can easily map them in the next modules.



The screenshot shows a user interface for managing scenarios. On the left, there is a purple circular icon with a white network diagram and a clock, labeled "Scenarios" with the subtext "Start a scenario". To the right, a modal dialog titled "Scenarios" is open. The dialog has a light blue background and contains the following text:

Receive data from parent scenario

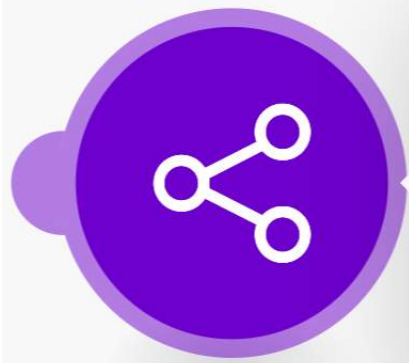
To receive data from a parent scenario, first add scenario inputs in this subscenario then return to the parent to fill them in.

Below the text is a button labeled "Add scenario inputs". At the bottom right of the dialog are two buttons: "Cancel" and "Save".

START A SCENARIO

RETURN OUTPUT

When you define scenario outputs, they start out empty. The **Return output** module is where you **decide what values they will have**. You can do this by either by mapping data from earlier modules or by typing a value directly. This is the data the AI agent will receive as a response. It always ends the scenario run, so whenever you use it, place it at the very end of your scenario.



Scenarios

Return output

Scenarios

status

Status of the email sending

Cancel Save



Both modules don't perform any operations or consume any credits, and make your AI agent tool scenario more organized. Use them!



You need to set your scenario as a tool to **on demand** so your AI agent can trigger it when needed.



To add a scenario tool to your AI agent, hover over the **+** icon of the **Run an agent** module. Then select **Add tool**. Then select the **Scenarios > Call a scenario** module.

After you add the **Scenarios > Call a scenario** module, you need to write a tool name and description. This helps the agent understand what the module does and when to use it.

You can give a scenario to your agent in two ways:

- **Select an existing scenario.** It needs to be on demand with defined input and output, plus the extra modules at the beginning and end.

- **Create a new scenario directly from this menu.** You still provide the name, description, input, and output, but Make adds the extra modules for you automatically at the beginning and at the end.

You'll see how this works in the next section.

3: MCPs

3

MCPs

Model Context Protocol (MCP) is a set of rules that lets AI agents and LLMs connect to and request services from third-party applications in an easier and more automatic way.

APIs do not follow a single standard, so every third-party application has its own way of handling requests. Because of this, AI agents cannot automatically connect to and use these services. You need to set up the

connection manually, for example with Make modules. **MCPs** can also help by providing a standard format that makes this process automatic.



MCPs **describe a third-party service in a standard format** that AI agents can automatically understand.

This includes information such as:

- What the tool is called
- What it does
- What input it needs
- What it will return
- How to call it (how to access the service)

If a third-party application provides an MCP, you can give it directly to your AI agent. The AI agent will then know on its own which services it

can use, what details it needs to send, and what information it will get back. It can **access the application's services automatically**, without extra setup.



To add a tool to your AI agent using MCP, hover over the + icon of the **Run an agent** module. Then select **Add tool**. Then select the **MCP> Call a tool** module.

You will learn more about this in a more advanced course.

[Continue to 1.5.2: Knowledge](#)

1.5.2 Knowledge

You can upload files with additional information that helps your AI agent make better decisions.



To do so, hover over the + icon of the **Run an agent** module and select **Knowledge**.

You will learn more about this in a more advanced course. And remember that this part is **optional**. If you don't have specific information your AI agent should know, skip it!

Best practices



Always provide your AI agent with high-quality information. The AI agent learns from what you give it, so only upload clear, accurate data you want it to follow to get the best results.

Your agent now has everything it needs to complete its tasks. You just need to run the scenario to use it. You'll see a demonstration in the next section.

[Continue to 1.6: Testing and training](#)



1.6 Testing and training

Before running your AI agent, you can **test it directly in Make** by chatting with it before running it. This lets you check if everything works as expected before running it for real.



Hover over the + icon of the **Run an agent** module and select **Chat**.

Here you can **try out your instructions and input** and see how the AI agent responds. You can also ask the AI agent to **run specific tools** to see what it returns.

If the answer isn't quite right, you can adjust the input, the description of the tools, or any configuration, and test again until you're happy with the result.

Before Derek goes out to investigate, his boss calls him in for a quick chat. They go over the case, check that he has all the right tools and that they're working properly. It's a way to make sure Derek is prepared, has the right tools, and is heading in the right direction, before hitting the field.

This helps you **refine the input or improve the tools' description** based on real feedback.

Best practices



Start simple

Begin with clear, direct requests to verify the AI agent triggers the correct tools and responds in-thread.

Test special cases

Try various or indirect inputs to ensure the AI agent handles them correctly or responds appropriately without guessing.

Adjust if needed

If the AI agent misuses a tool or provides incomplete answers, update its description and the tool instructions, clarifying their

purpose and proper use.

You will practice all of this in the more advanced course!

[Continue to the wrap up for this unit](#)



1.7 Wrap up

1

To use an AI agent in Make, you **build it inside a scenario**. In this scenario, you add the **Run an agent** module to configure it. You then need to provide it with the tools and knowledge it needs. The AI agent configuration includes: the LLM it uses, instructions describing its role, an input specifying what you want it to do, and any optional settings that control its behavior.

2

Tools let AI agents use services from third-party applications. In Make, you can add tools in three ways: using **Make modules** directly, building

scenarios that act as tools, or connecting **MCP** services. When building a scenario to use as a tool, you must set it to on demand, define scenario inputs and outputs, and add the Start a scenario module at the beginning and the Return output module at the end to manage the data.

3

You only need a Make scenario to build, configure, and run your AI agent. In this scenario, you decide how it starts, set up your AI agent using the Run an agent module, and give it the tools and knowledge it needs. Once everything is configured, you just need to **run the scenario to use your AI agent.**

Well done! You now know what you need to build your AI agents in Make.

In the next unit, you'll see a demo of how to build your first AI agent.



 **make | academy**



Mark this task complete to continue to the next unit.