

Building in Dify: from workflow to agent



Guide 1 for the AI Agents workshop

LeX Consultancy B.V. · 20 September 2026

This guide follows the cloud version of Dify as it looked on 20 September 2026. Dify changes fast: if your screen differs, follow what you see, not what is written here.

The screenshots show the Dutch build of this app: the same screens and the same clicks, with Dutch text in the fields and Dify's menus in Dutch. The button names inside the app are English on every account, and those are the names used below.

How to read this guide

Form	Meaning
1. 2. 3.	What you do: click, choose, open
Bold	A button, menu or field as it appears on your screen
Table <i>What you type</i>	Values you put in a form field
Grey box <i>Type in</i>	Text you type or paste literally
Orange box	A pitfall you would otherwise discover yourself
Green box <i>Your choice</i>	A place where you can replace our example with your own material, style or prompt

The design card for this app

Before we started clicking, this card was filled in (the blank card is in the Design card appendix). Fill in the same five boxes for your own app; the loose version of this filled-in card is in 00-design-card/examples.

1. Who is it for	2. What goes in
An English teacher with a reading text from the course book that is too hard for half of Year 9. Has ten minutes before the lesson and does not want to rewrite it by hand.	The text (pasted, a Paragraph field) and the level from a short list: year 6, year 7, year 9, year 12 (a Select field).

3. What comes out	4. What it must stick to
The text rewritten at the chosen level, with all the content kept and technical terms explained in a subordinate clause; below it three comprehension questions (fact, main idea, application) without answers.	Short sentences, leave nothing out, do not replace technical terms. No knowledge base needed: the rules live in the prompts of the two LLM steps.

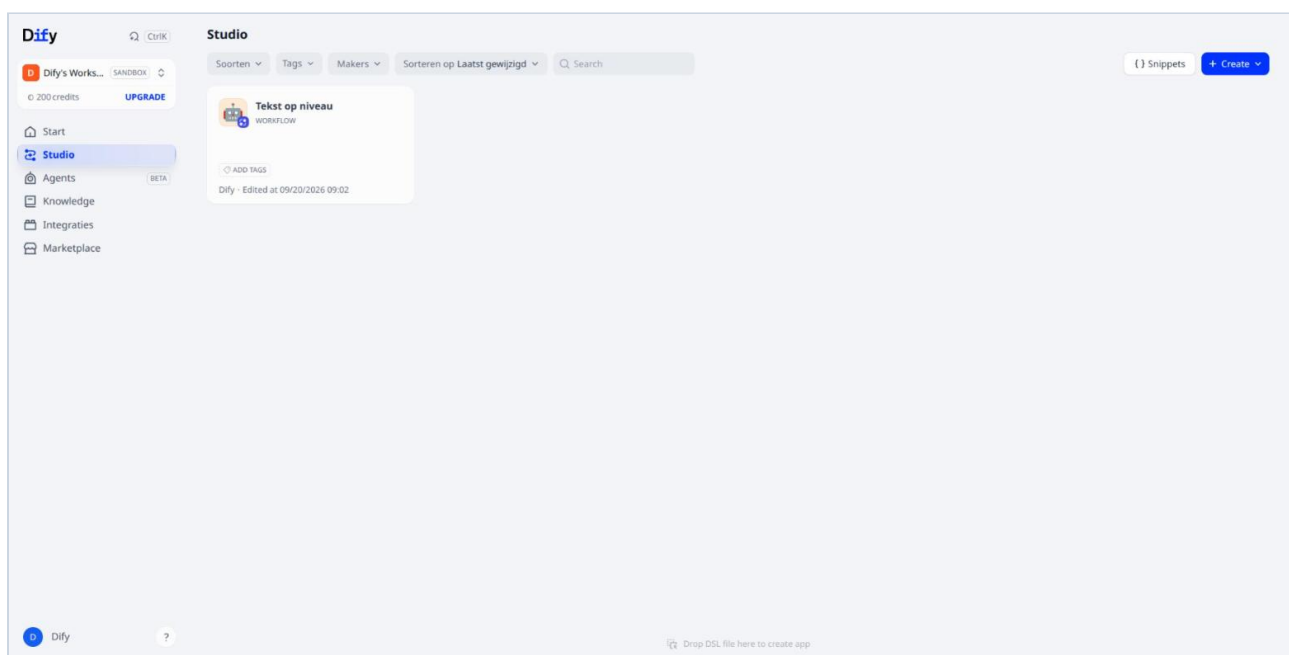
5. How you know it works

Test	What goes in	What must come out
1	A paragraph from a geography course book, level year 7	Shorter sentences, same facts, three questions that are about the text
2	The same paragraph, level year 12	Barely simplified, questions at analysis level
3 (the hard one)	A text with a table of figures, level year 6	The figures stay correct; the app invents no new numbers

Preparation

1. Go to cloud.dify.ai and log in.
2. You arrive in **Studio**: this is where all your apps live.

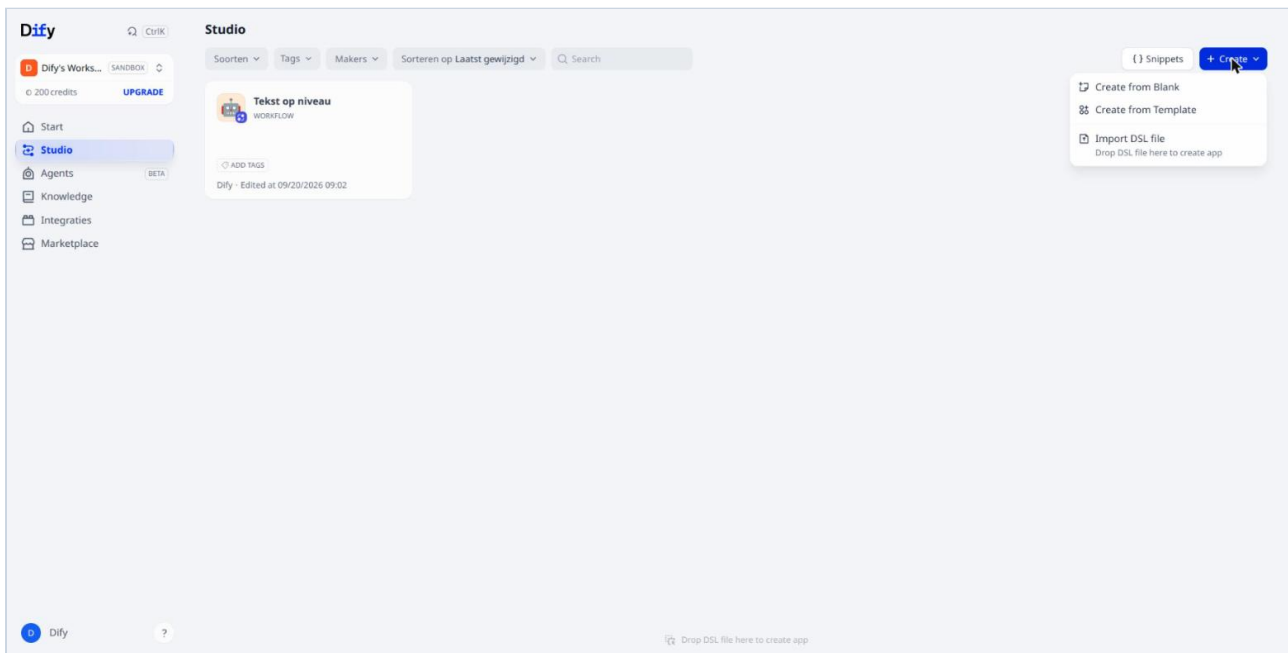
Top left you see your workspace with the label SANDBOX and your number of credits. That number is a one-off allowance, not per month.



Studio, the overview of your apps

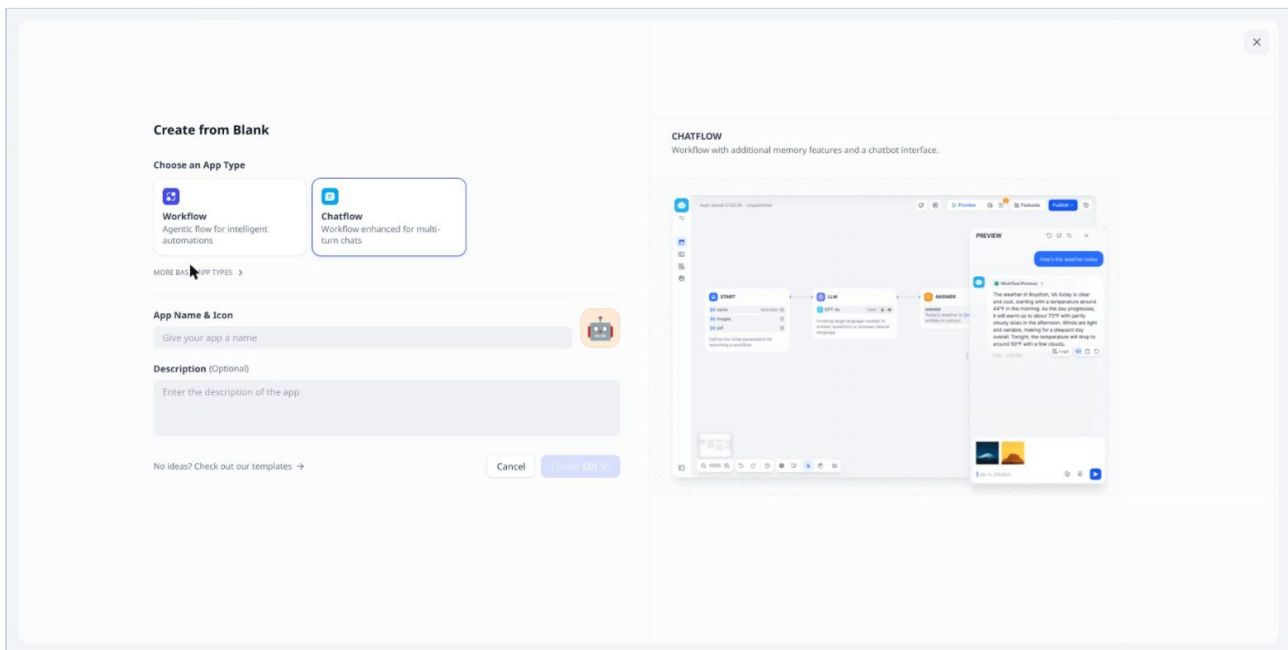
Step 1. Create a new app

1. Click **Create** at the top right.



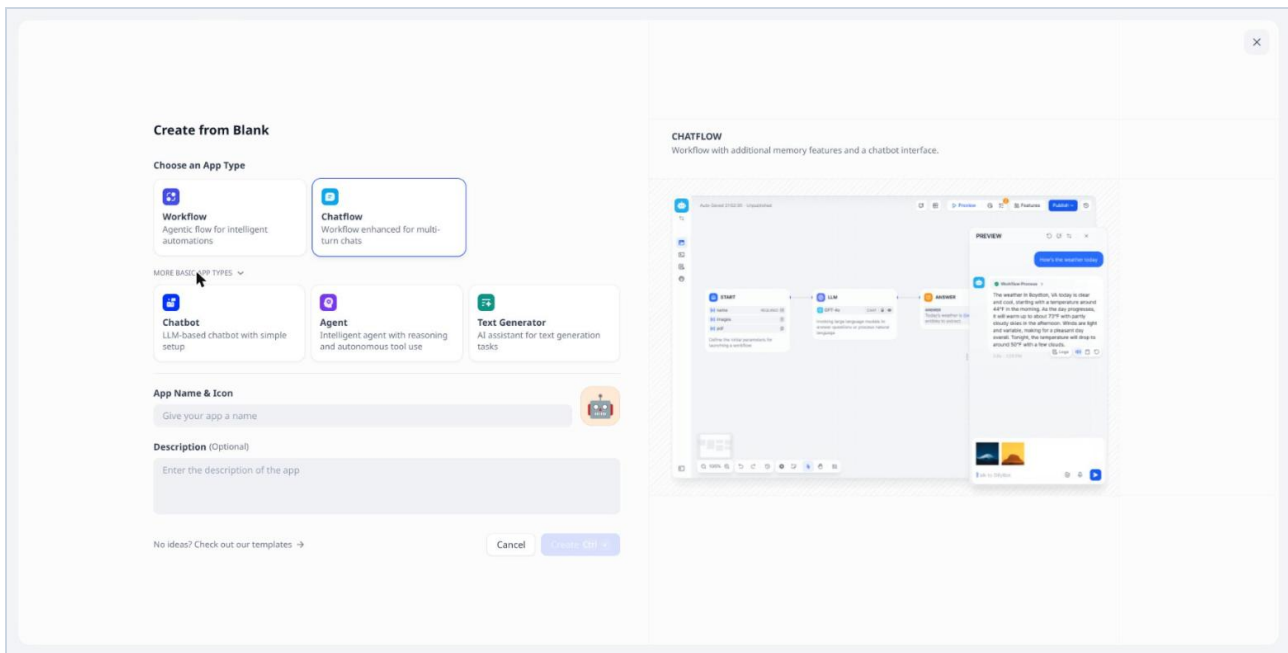
The Create menu

2. Choose **Create from Blank**. You see two kinds: Workflow and Chatflow.



Create from Blank, two visible kinds

3. Click **MORE BASIC APP TYPES**. Now Chatbot, Agent and Text Generator appear as well.
This is the button everyone misses when they want to build an agent.



All five app types

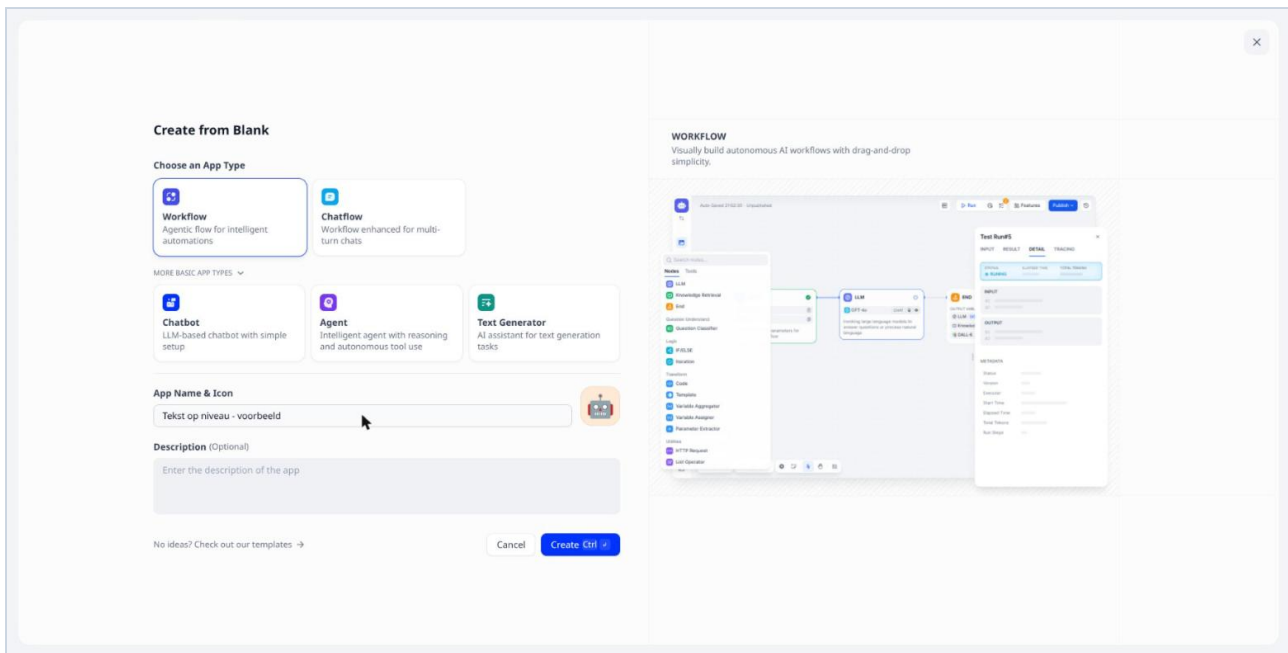
Type	What it is	What for
Workflow	Fixed steps from input to output	Producing material, always the same route
Chatflow	A workflow with a chat window and memory	A conversation with a fixed structure
Chatbot	A chat window with an instruction	Trying something out quickly
Agent	A goal plus tools; chooses its own steps	A conversation partner that thinks along
Text Generator	Text tasks	One-off text jobs

4. Click **Workflow**.

5. Fill in the name.

Field on screen	What you type
App name	Text at level

6. Click **Create**.

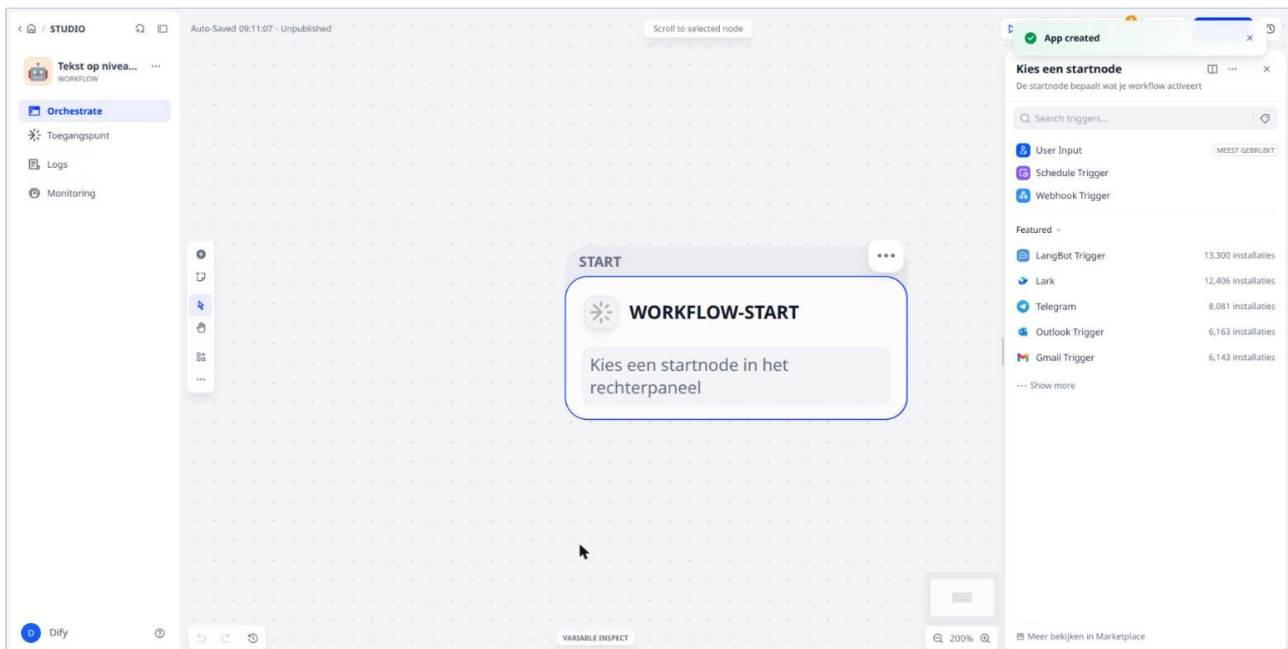


Workflow chosen, name filled in

Step 2. Choose a start node

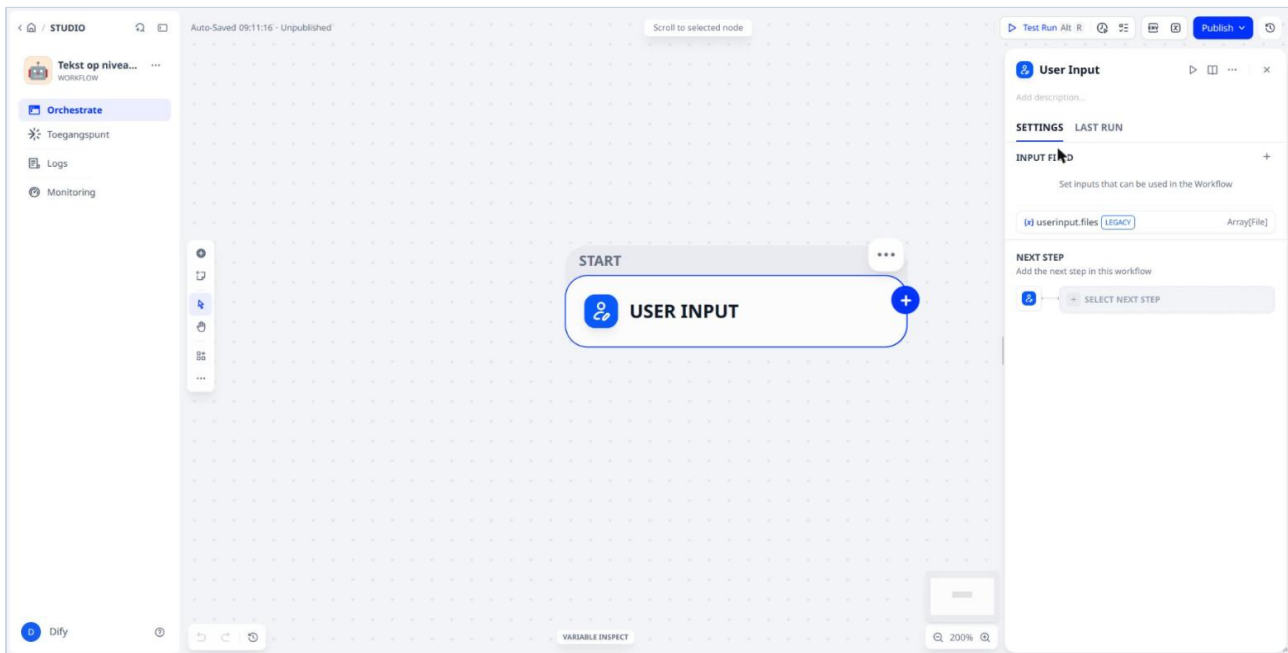
A workflow does not start by itself. In the right-hand panel you choose what triggers it: User Input, Schedule Trigger or Webhook Trigger.

1. Click **User Input** in the right-hand panel.



Choosing the start node

The node is now on the canvas. On the right you see **INPUT FIELD** with a plus sign: that is where you create the fields the user will fill in.

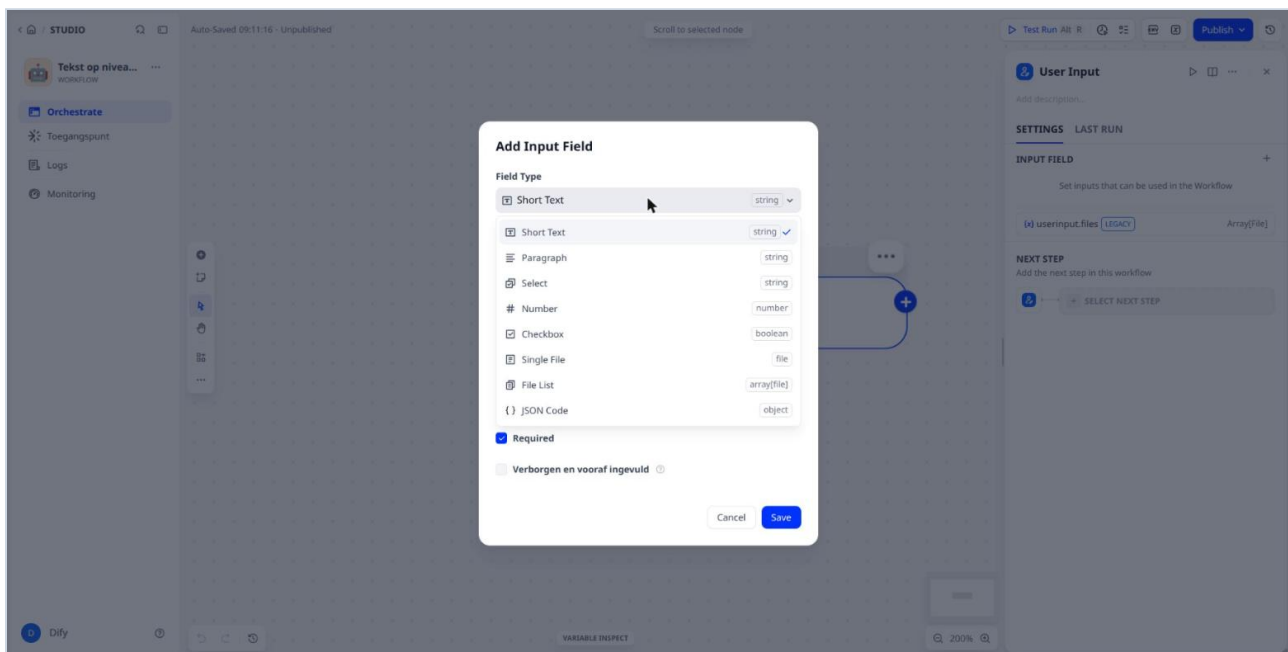


The User Input node

Step 3. Create two input fields

Field 1: the text

1. Click the plus sign next to **INPUT FIELD**.
2. Open **Field Type** at the top.

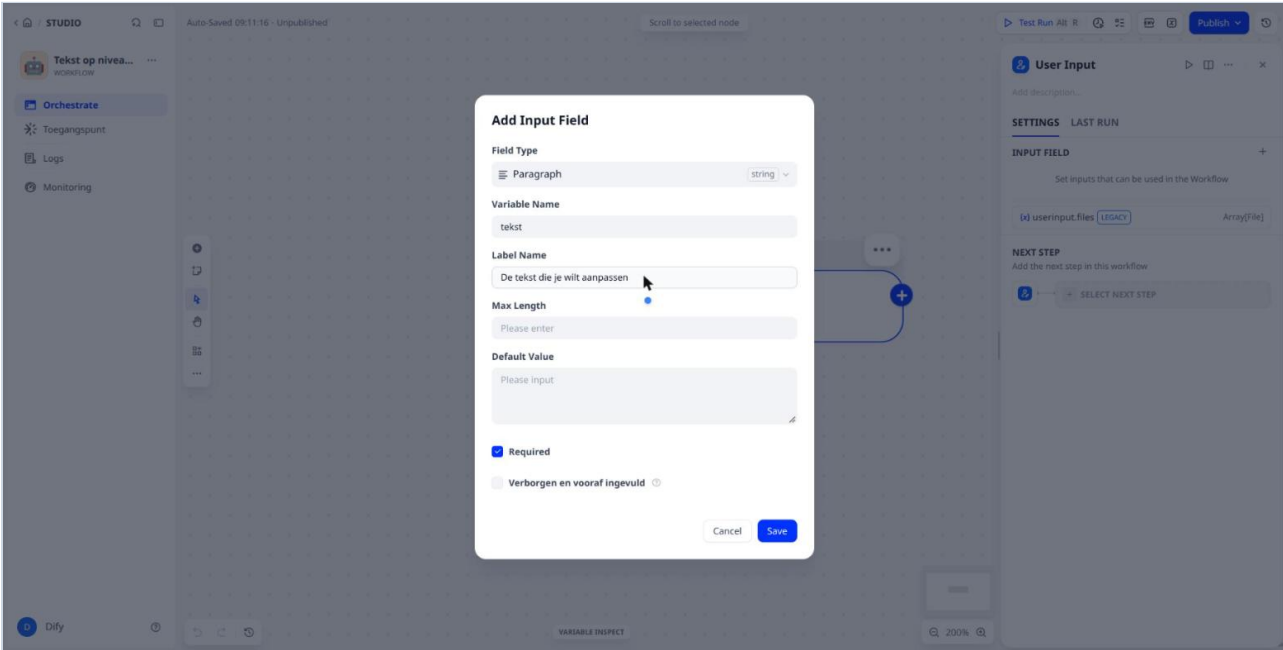


The field types

- 3. Choose **Paragraph**.
- 4. Fill in the two fields.

Field on screen	What you type
Variable Name	text
Label Name	The text you want to adapt

Watch out: as soon as you type the variable name, Dify fills the label field with that same name. Clear that field before typing your own label, otherwise you end up with `textThe text you...`



Field text, type Paragraph

- 5. Click **Save**.

Field 2: the level

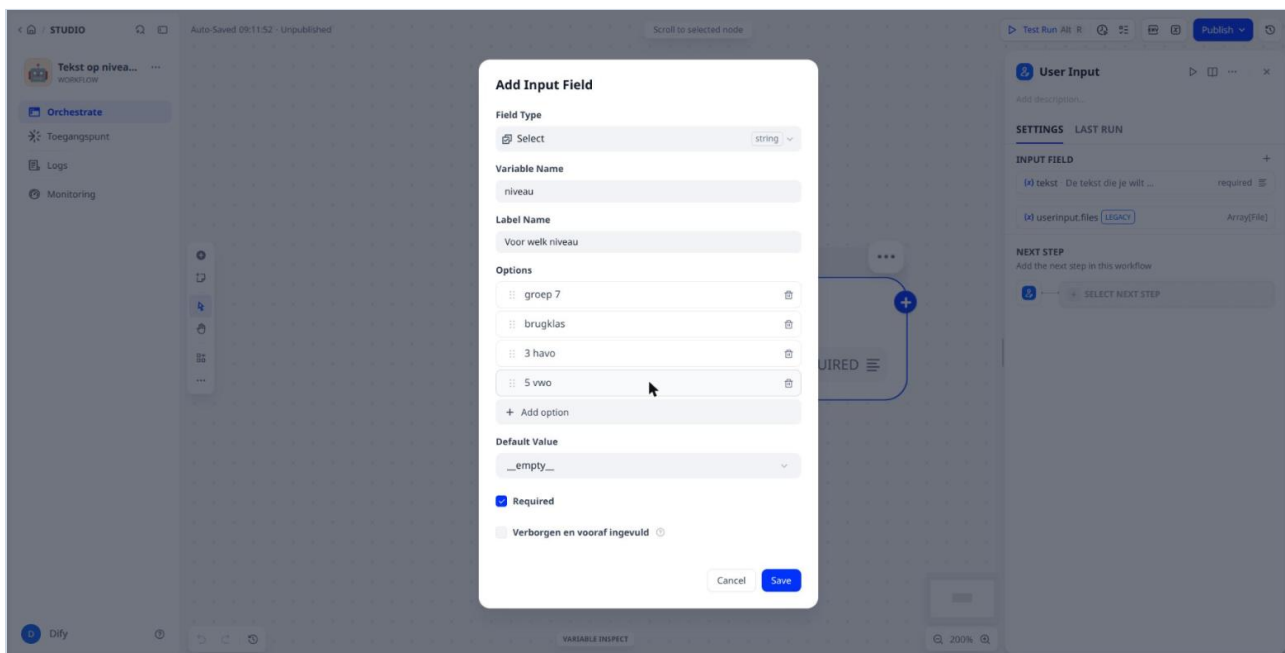
- 1. Click the plus sign next to **INPUT FIELD** again.
- 2. Set **Field Type** to **Select**.
- 3. Fill in the two fields.

Field on screen	What you type
Variable Name	level
Label Name	For which level

- 4. Click **Add option** four times.
- 5. Click in each empty box and type one option per line.

Option	What you type
1	year 6
2	year 7
3	year 9
4	year 12

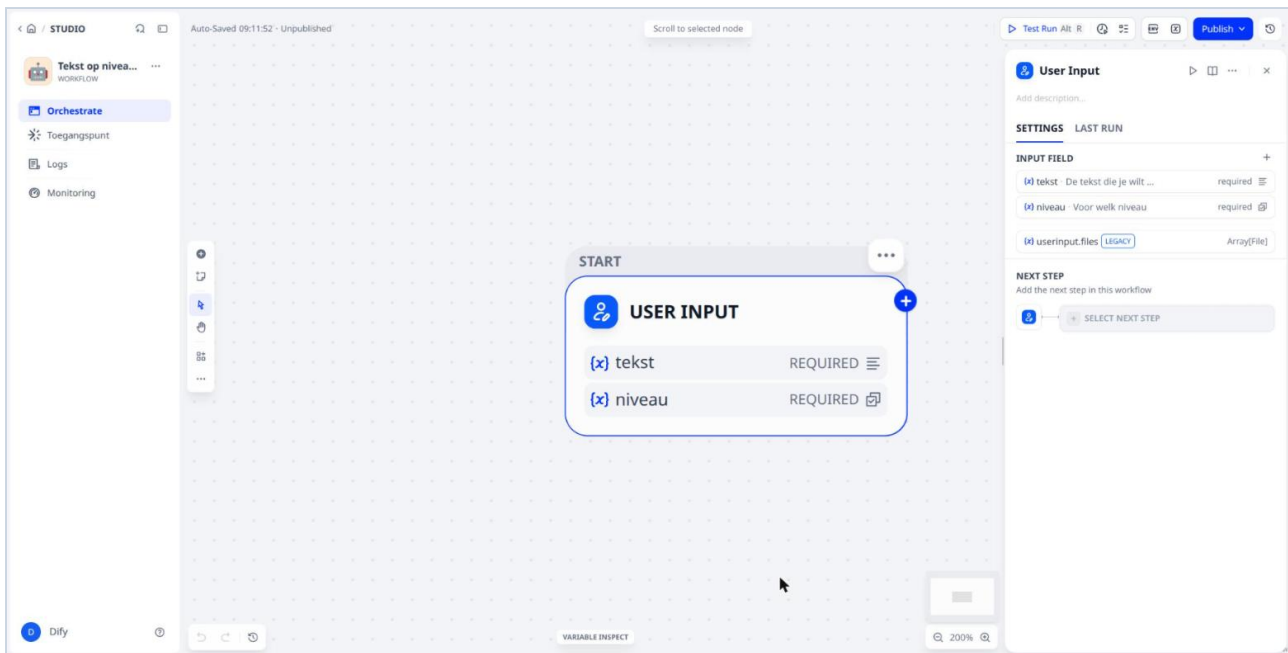
After each new line, click in the empty box first before you type. If you do not, your text ends up nowhere.



Field level, type Select with four options

Your choice: the levels. These four options are an example. Put in the levels you actually teach: “year 3”, “vocational year 2”, “adult learners”, “ESL A2”, or something else entirely such as “colleague”, “parent”, “governor”. The option later becomes the literal word that goes into the prompt, so choose words the model can work with.

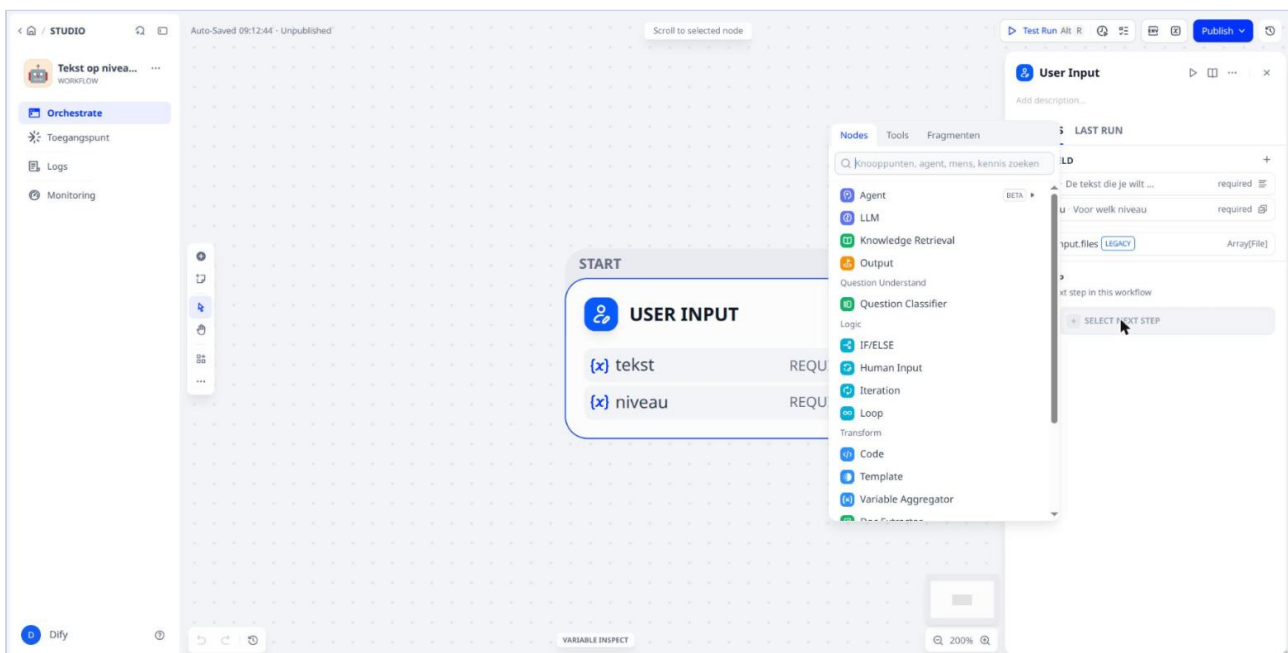
6. Click **Save**. Both fields are now in the node.



Both input fields

Step 4. Add the first thinking step

1. On the right, under **NEXT STEP**, click **SELECT NEXT STEP**. You get the list of nodes.



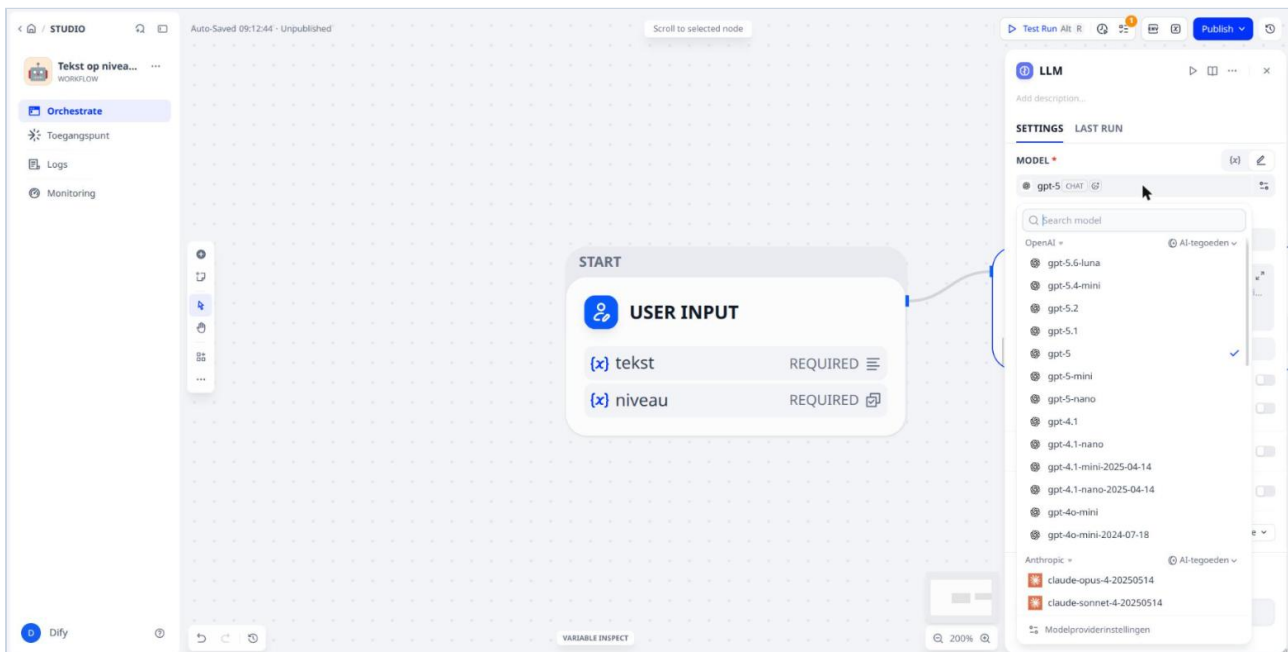
The list of nodes

2. Choose **LLM**.

Set the model right straight away

The node defaults to **gpt-5**. That costs five credits per answer. gpt-5-mini costs one. With two hundred credits that is the difference between forty and two hundred answers.

3. Click the model name **gpt-5**.
4. Choose **gpt-5-mini**.



The model picker

Write the prompt

5. Click in the **SYSTEM** field.
6. Type the instruction.

TYPE IN

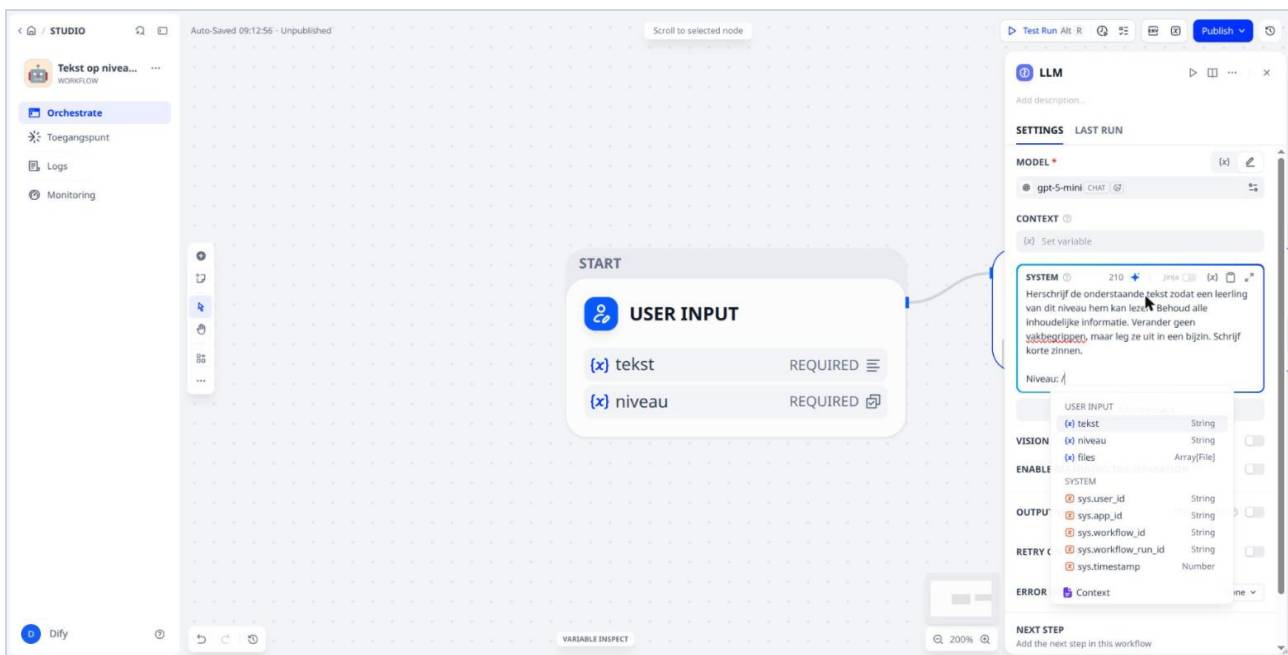
Rewrite the text below so that a student at this level can read it. Keep all the factual content. Do not replace technical terms, but explain them in a subordinate clause. Write short sentences.

7. Press Enter for a new line and type:

TYPE IN

Level:

8. Right after it, type a forward slash /. A list of your input fields appears.
9. Choose **level** from the list. No need to type curly braces; Dify inserts the variable itself.



The variable list after a forward slash

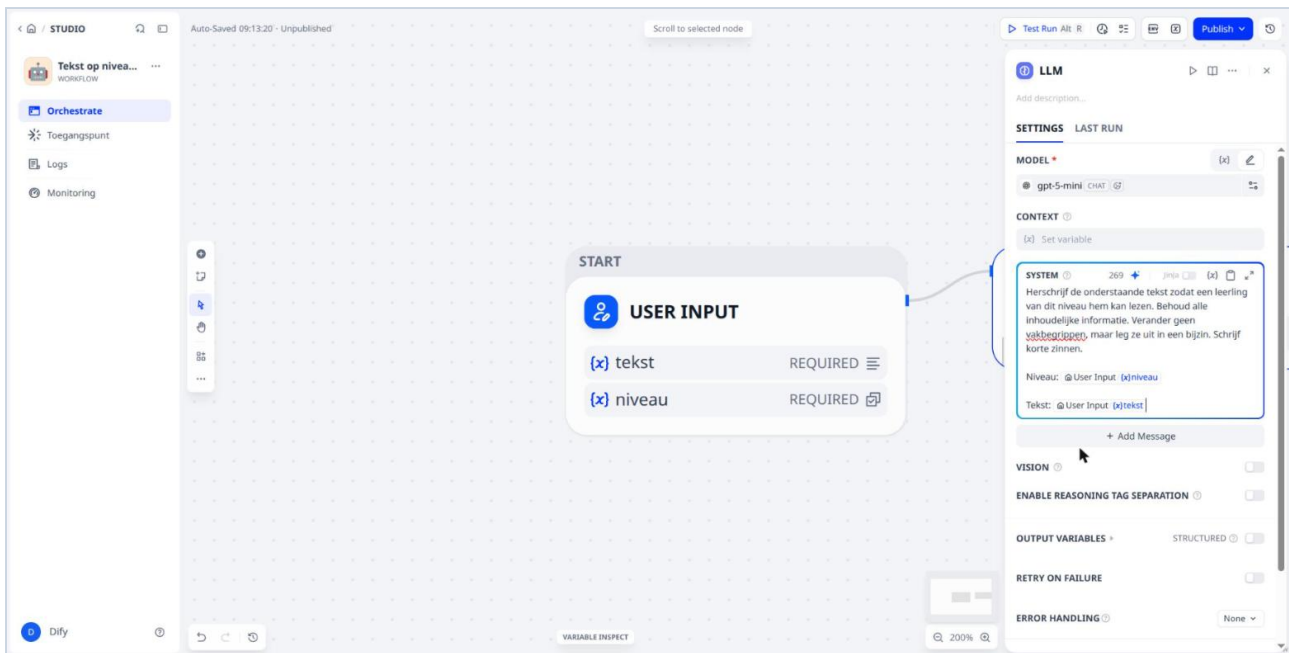
Your choice: the prompt. The instruction above is the heart of your app. Want the text to get shorter too? Add: "Make the text at most half as long." Want difficult words in bold and explained at the bottom? Say so literally. Want the style of your own course book? Paste a paragraph from it as an example, preceded by "Write in this style:". Change one thing at a time and test again (step 7); then you know which change has which effect.

10. Press Enter and type:

TYPE IN

Text:

11. Type a forward slash / again and choose **text**. The prompt is now complete.



The first LLM node, finished

Step 5. Add the second thinking step

1. Scroll down in the right-hand panel.
2. Click **SELECT NEXT STEP** and choose **LLM** again.
3. Click the model name and choose **gpt-5-mini**.

If a message appears saying the temperature parameter was removed because the model does not support it: ignore it, nothing goes wrong.

4. Click in the **SYSTEM** field and type the instruction.

TYPE IN

Write three comprehension questions about the text below. Question 1 is about a fact in the text, question 2 about the main idea, question 3 asks the student to apply something. Do not give answers.

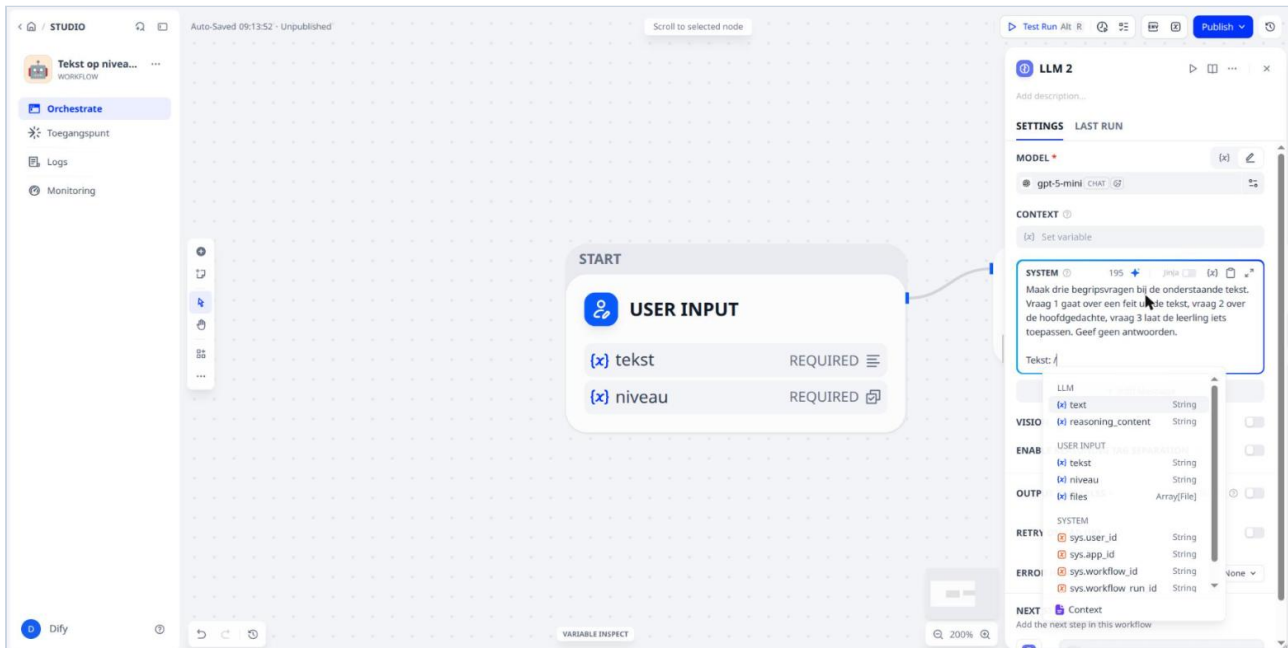
5. Press Enter and type:

TYPE IN

Text:

6. Type a forward slash /. There are now more variables in the list than before: at the top the output of **LLM 2** (the node you are in) and below it that of **LLM** (the previous node).
7. Choose **text** under **LLM**. That is the rewritten text.

If you choose **text** under LLM 2, it writes questions about the original text, and you only notice when testing.



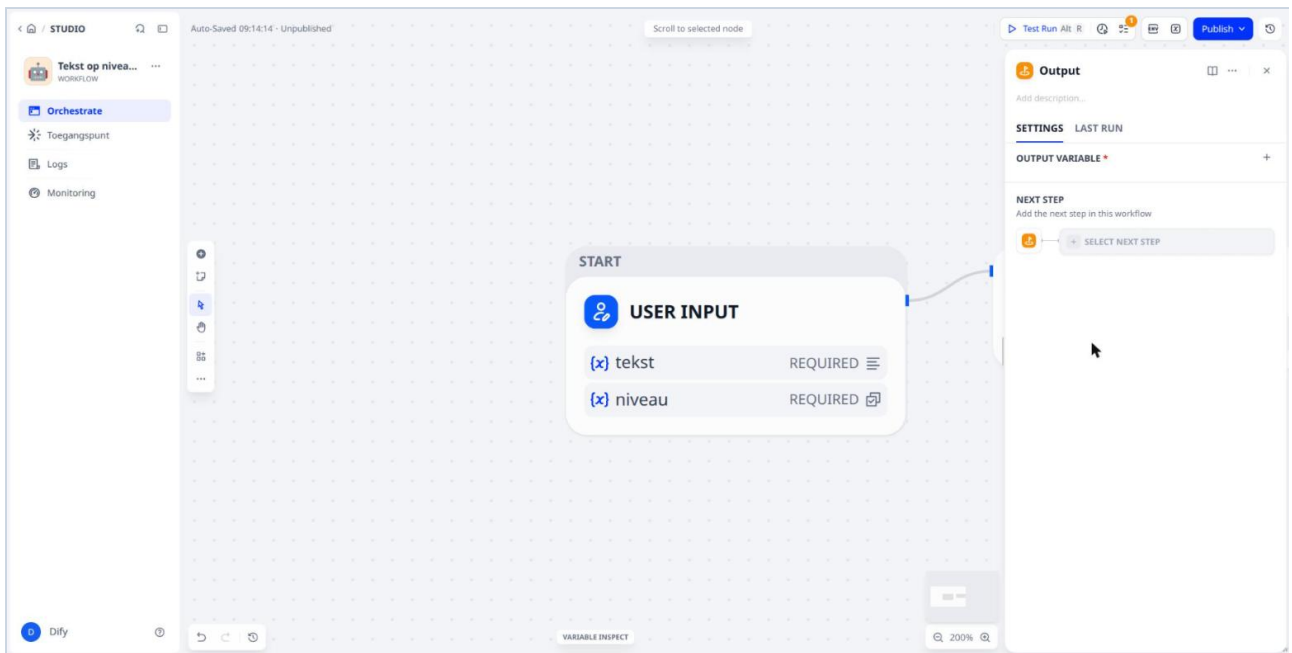
The variable list with LLM and LLM 2

This is what a workflow is all about: the second step works with what the first one produced. You decide that order.

Your choice: the second step. Comprehension questions are one option. The same spot can also produce a word list ("Put the ten hardest words from the text in a table with a one-sentence explanation"), a three-sentence summary, a translation for a newcomer, or a readability check ("Name three sentences that are still too long"). You can also add a third step; its list then contains the output of both previous steps.

Step 6. Decide what comes out

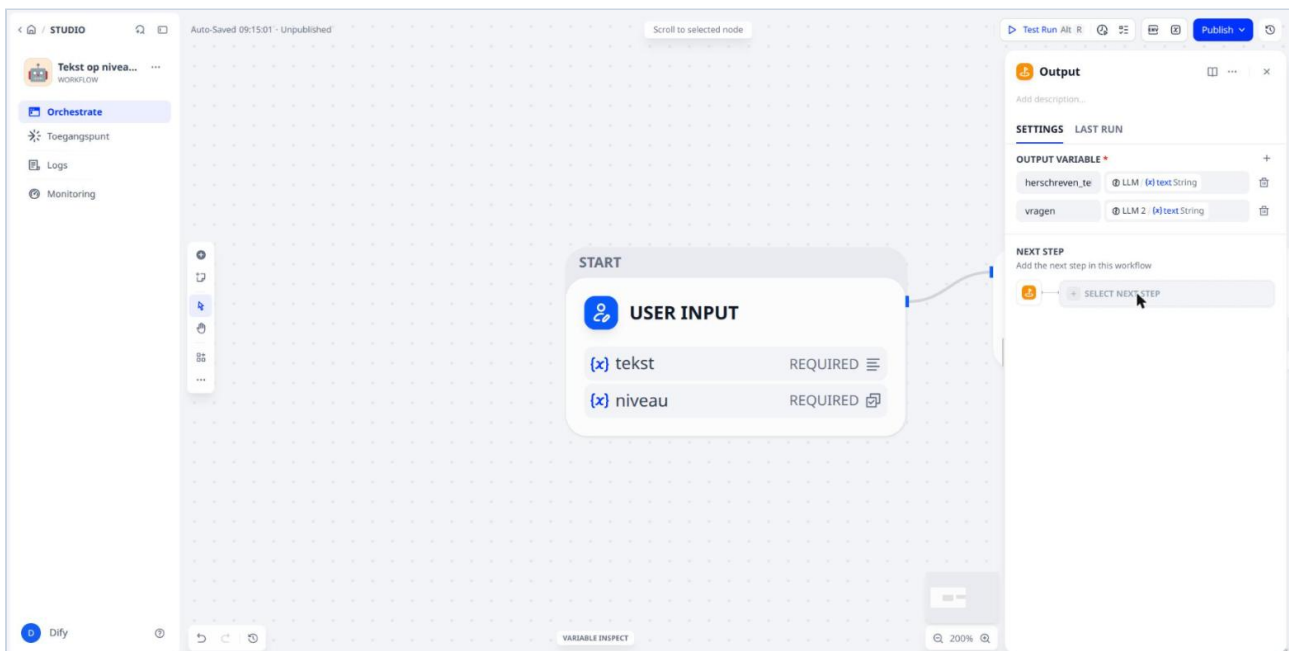
1. Click **SELECT NEXT STEP** and choose **Output**.



The empty Output node

2. Click the plus sign next to **OUTPUT VARIABLE**.
3. Create two output variables. For each one, type the name and choose the source from the list.

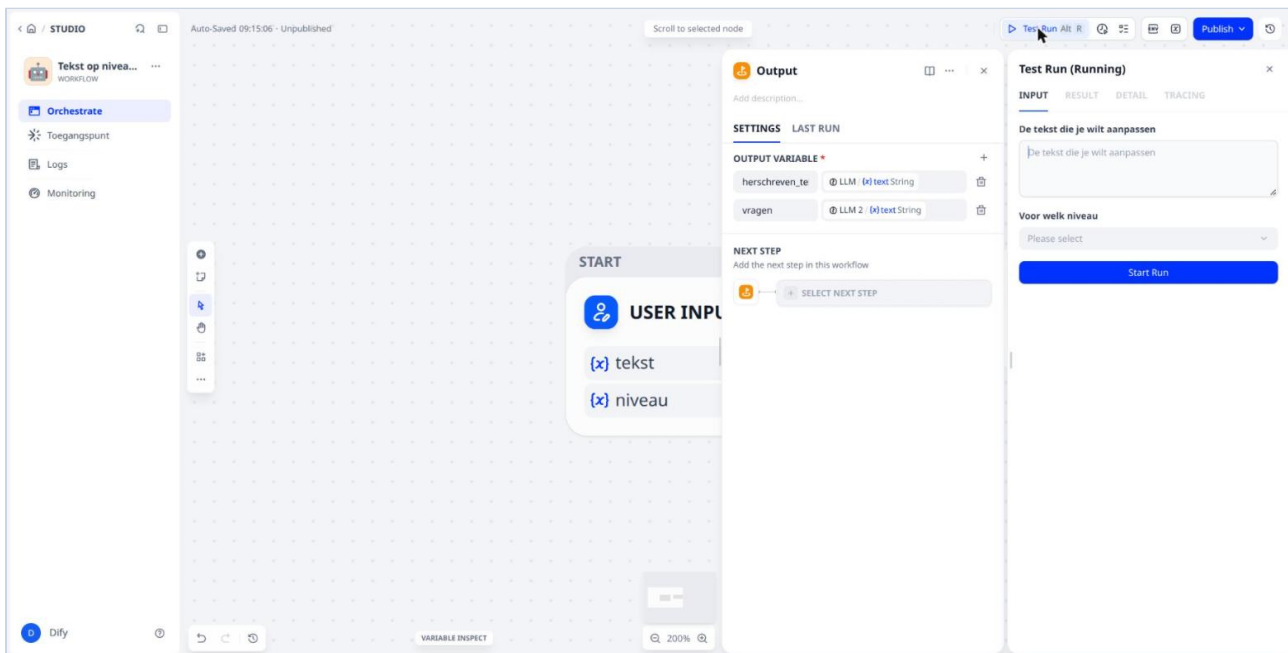
Variable Name (type)	Source (choose)
rewritten_text	LLM / text
questions	LLM 2 / text



The Output node with two variables

Step 7. Test

1. Click **Test Run** at the top right. A panel with your own input fields appears on the right.



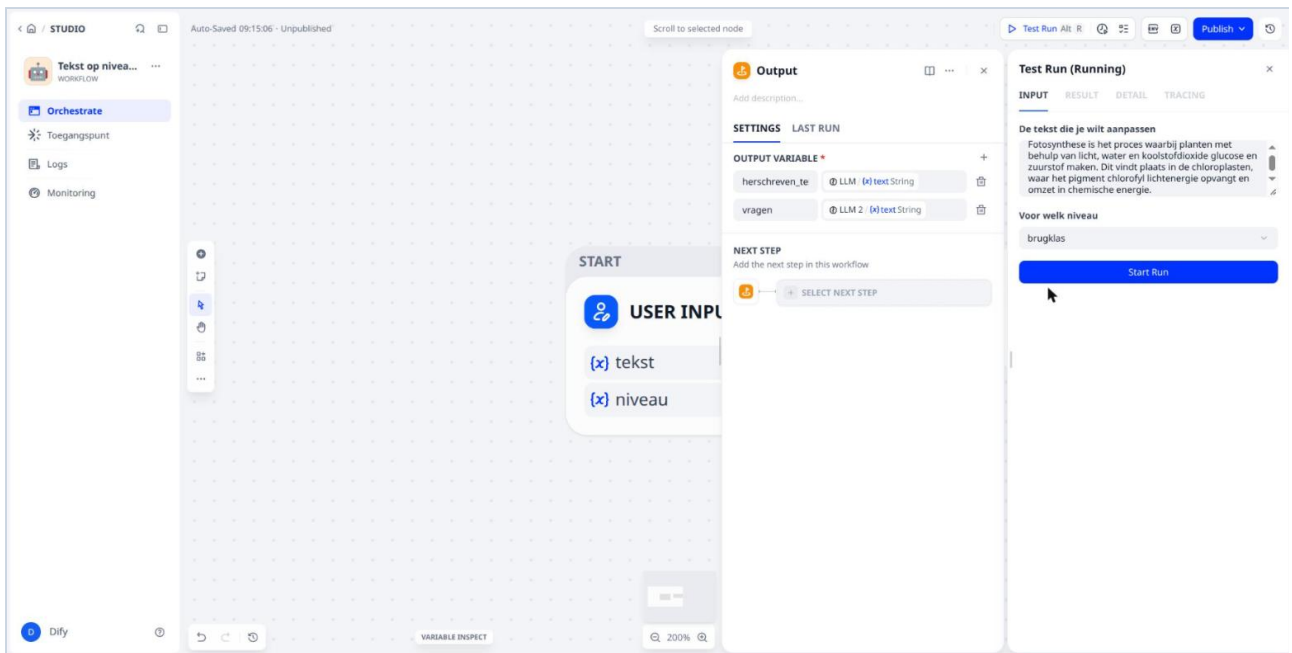
The test panel

2. Paste a piece of text into the field **The text you want to adapt**. If you have nothing to hand, use this:

TYPE IN

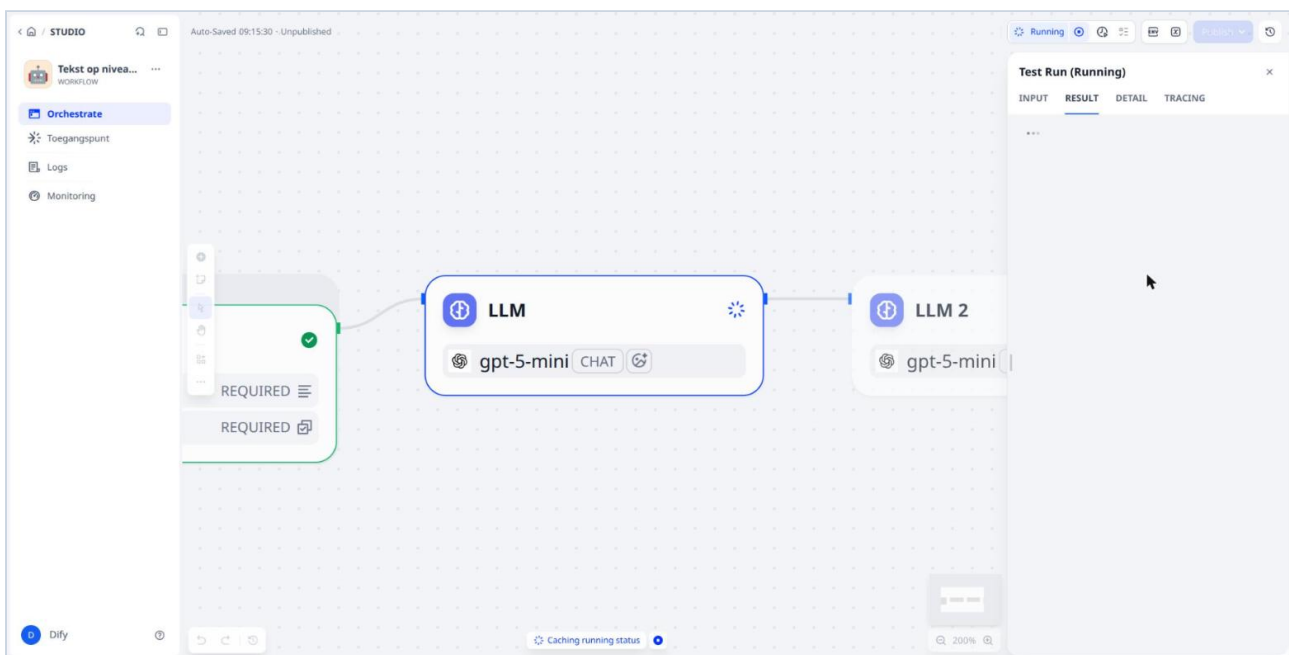
Photosynthesis is the process by which plants use light, water and carbon dioxide to make glucose and oxygen. It takes place in the chloroplasts, where the pigment chlorophyll captures light energy and converts it into chemical energy.

3. Under **For which level** choose an option, for example **year 7**.
4. Click **Start Run**.



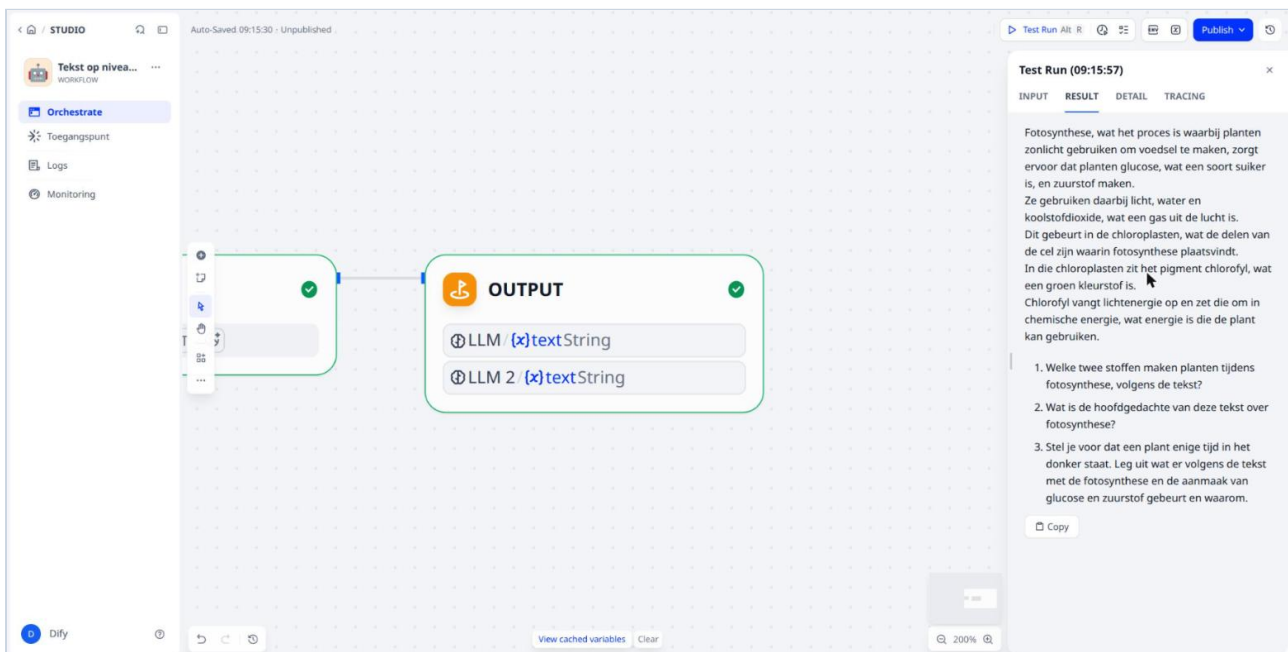
Input filled in

You see the nodes light up one by one as they run.



The workflow running

Under **RESULT** is the outcome: the rewritten text and the three questions. Under **DETAIL** and **TRACING** you see per step exactly what happened and how long it took.

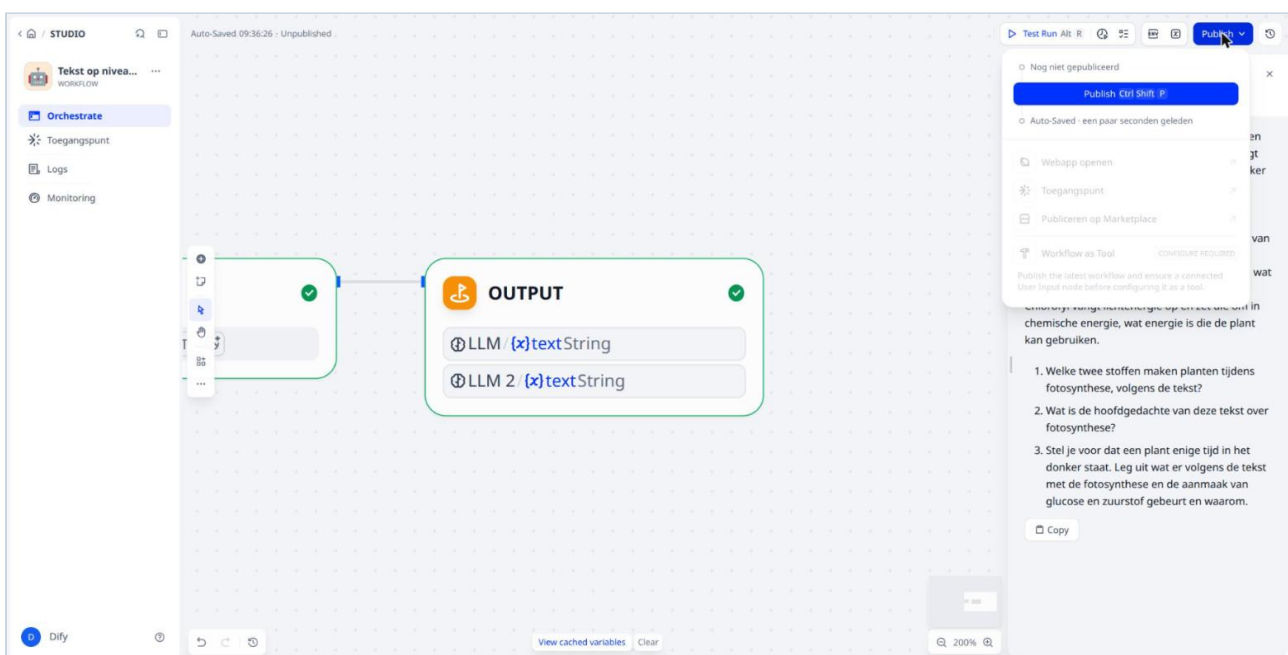


The result

Step 8. Publish

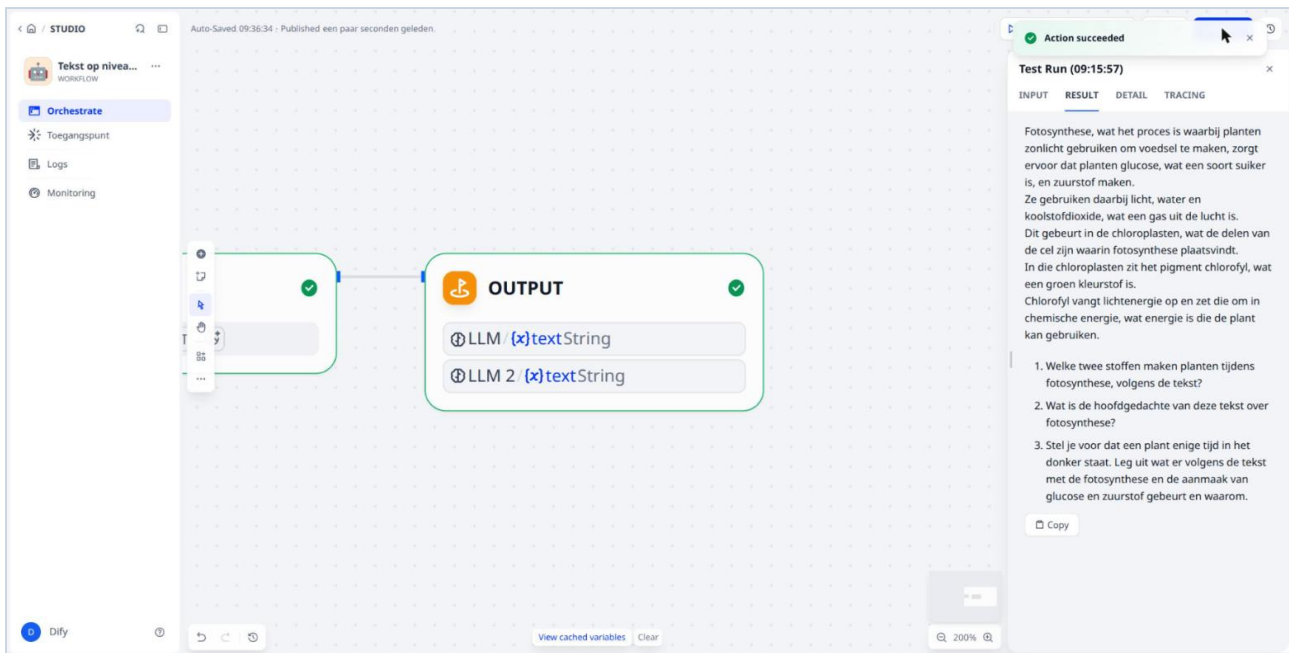
As long as you do not publish, your app exists only for you.

1. Click **Publish** at the top right.



The Publish menu, not yet published

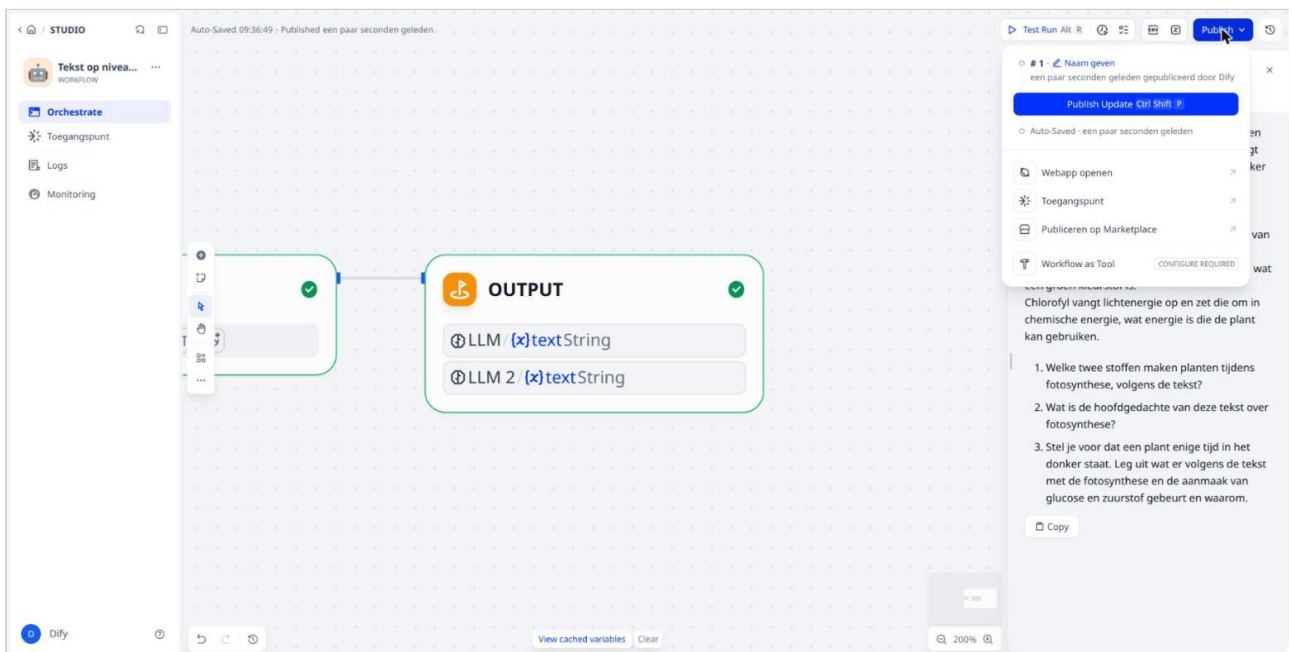
2. Click **Publish** in the menu. At the top the status changes to published.



Published

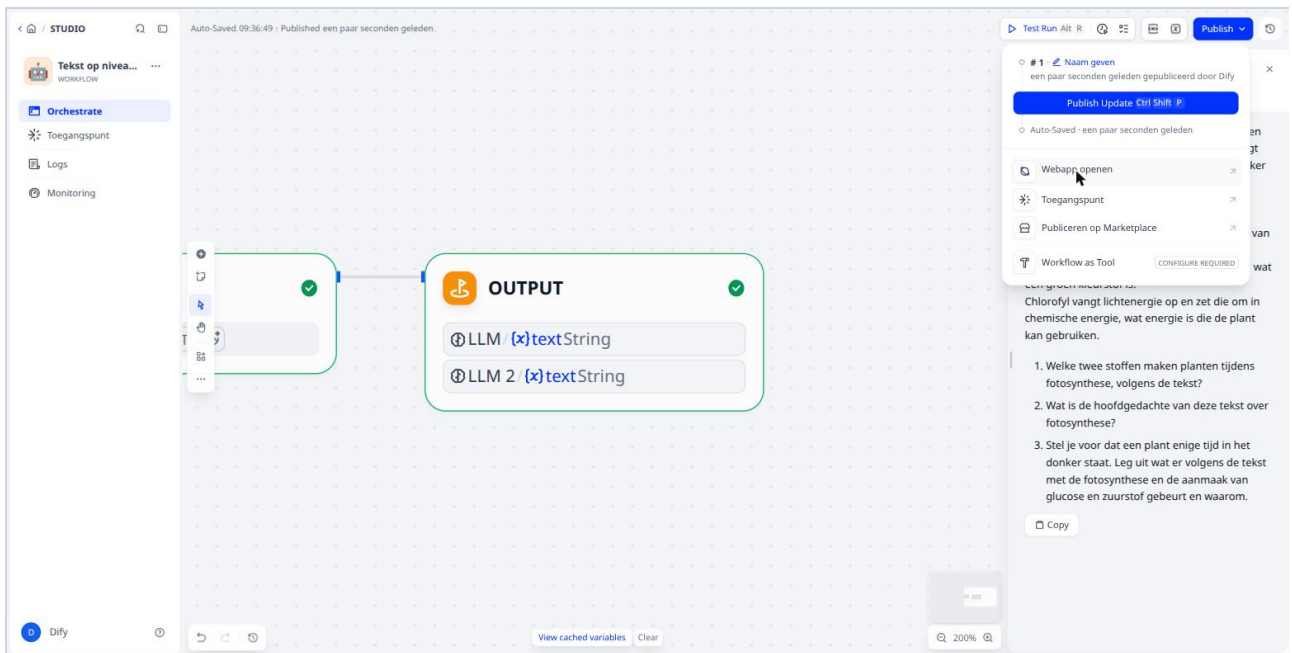
- Open the menu again. It now says **Publish Update** and the grey options have become active.

Every change you make from now on has to be published again with Publish Update. Otherwise the link keeps showing the old version.



The menu after publishing

- Click **Open webapp** (in the Dutch screenshots: Webapp openen).

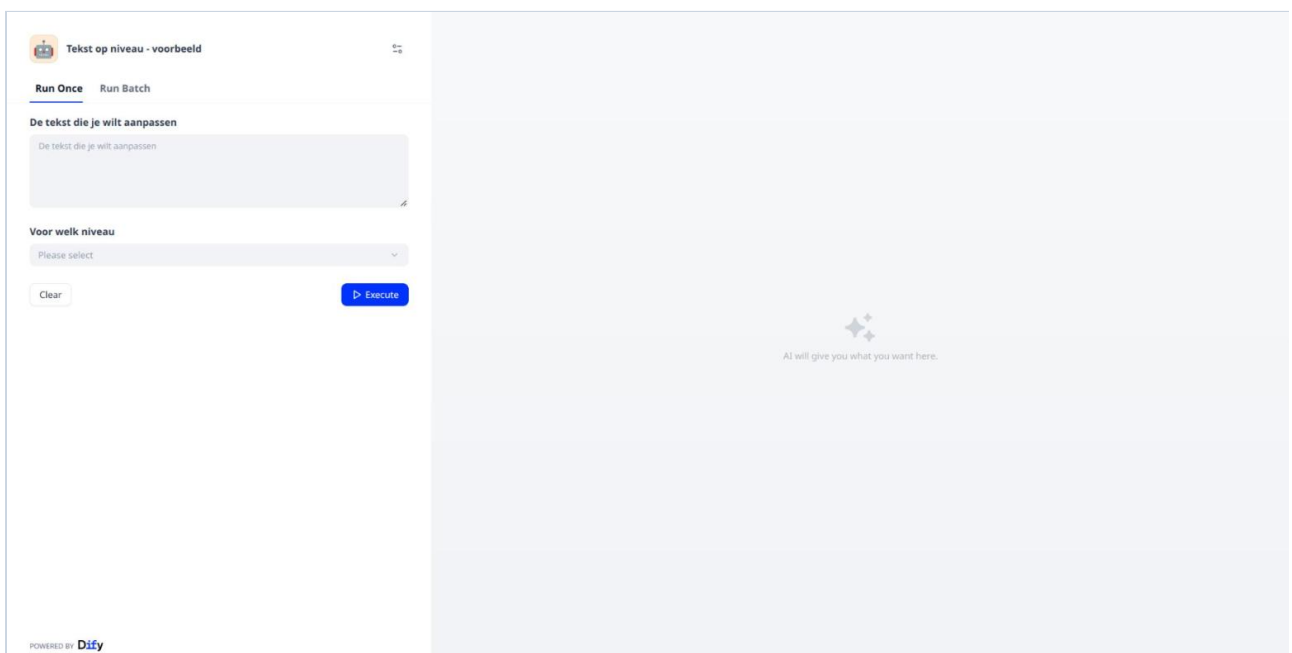


Opening the webapp

Your app opens as an ordinary web page with its own address. That link works without logging in, on a phone too. That is the link you give to your test partner.

The Dutch version of the app in this guide is at

<https://udify.app/workflow/eINp9Jdd6ICodDZF>. Open it to see what you are working towards.



The published webapp

Checking that it worked

- Your app is in Studio with the label WORKFLOW
- The test run gives both a rewritten text and three questions

- The webapp link opens for someone else, without logging in

When things go wrong

What you see	What is going on
Your credits drop fast	The node is still on gpt-5. Set it to gpt-5-mini
The label reads <code>textThe text you...</code>	The label field filled itself. Clear it and type again
The questions are about the original text	In step 5 LLM 2 was chosen instead of LLM
A message about temperature	The model does not support that parameter. Ignore it
An option stays empty	You typed without clicking in the empty box first
The link shows an old version	You did not click Publish Update after your change

Make it yours

Everything above is an example: the levels, the prompts, the second step. This chapter helps you rebuild the app for your subject, your students and your style. Work from top to bottom: the first changes take five minutes, the last ones half a day.

Change	Where	Difficulty	What you get for it
Other levels or audiences	Step 3, field <i>level</i>	easy	Options that describe your students
Other labels on the form	Step 3, <i>Label Name</i>	easy	A form in your words, also for colleagues who did not build the app
Stricter or different writing rules	Step 4, prompt	easy	Texts that resemble what you would write yourself
A sample text in your style in the prompt	Step 4, prompt	medium	Consistent tone, even for long texts
A different second step (word list, summary, translation)	Step 5, prompt	medium	An app for your lesson goal instead of ours
A third step that checks the questions against the level	After step 5, extra LLM	medium	A built-in check
An extra input field (subject, length, language)	Step 3 plus the prompts	medium	More knobs for the user, but also more to explain
Uploading a file instead of pasting text	Step 3, field type <i>Single File</i> , plus a Doc Extractor node	hard	Process worksheets and PDFs directly (see guide 5 for files as input)

Three assignments

1. **Your levels, your text.** Replace the four options with the groups you teach. Paste a paragraph from your own material and run the app for two levels. Put the two results side by side: what is really different? If the difference is too small, state in the prompt what you expect per level ("year 7: sentences of at most twelve words").
2. **Change the second step.** Rebuild step 5 into something you can actually use next week: a word list, a summary for the classroom, three statements for a discussion. Also change the name of the output variable in step 6, so the output gets a sensible heading.
3. **Let a colleague break it.** Give the link to someone who was not in the workshop and ask for three real texts. Look in **Logs** at what came out. Where did it go wrong? The answer is usually one extra sentence in a prompt. Change it, publish again (Publish Update), and let them try once more.

Checklist before you use your own material

- [] The text you paste contains no student names and no test results
- [] You are allowed to use the material (your own work, an open licence, or the publisher's permission)
- [] The model is set to gpt-5-mini and you know roughly what one run costs

- [] There is one owner of the app who maintains the prompts and republishes
- [] You have decided who gets the link: only you, your department, or students too (in that case: talk to your ICT coordinator or data protection officer first)
- [] Someone reads the output before it goes to students; the app produces a draft, not a lesson

And now an agent

You have just built a workflow: you decided every step. An agent works differently. You give it a goal and boundaries, and it chooses its own steps.

1. Click **Create** and choose **Create from Blank**.
2. Expand **MORE BASIC APP TYPES** and choose **Agent**.

The rest of the method you now know: set the model, write the instruction, test, publish.

Licence: CC BY-SA 4.0 (<https://creativecommons.org/licenses/by-sa/4.0/>). You may copy, adapt and reuse this document, including at your own school, provided you credit the source (Guido van Dijk, LeX Consultancy B.V., <https://git.lexconsultancy.nl/guido/dify-guides>) and share your adaptation under the same licence. The LeX Consultancy logo is excluded; product names are trademarks of their owners. Schools, people and documents in the examples are fictional.

Owner and contact: Guido van Dijk, LeX Consultancy B.V. · guido@l-e-x.nl

