



Custom Tool

Watch how to use custom tools

Flowise OpenAI function calling



Problem

Function usually takes in structured input data. Let's say you want the LLM to be able to call Airtable Create Record [API](#), the body parameters has to be structured in a specific way. For example:

```
"records": [  
  {  
    "fields": {  
      "Address": "some address",  
      "Name": "some name",  
      "Visited": true  
    }  
  }  
]
```

Ideally, we want LLM to return a proper structured data like this:


```
{  
  "Address": "some address",  
  "Name": "some name",  
  "Visited": true  
}
```

So we can extract the value and parse it into the body needed for API. However, instructing LLM to output the exact pattern is difficult.

With the new [OpenAI Function Calling](#) models, it is now possible. `gpt-4-0613` and `gpt-3.5-turbo-0613` are specifically trained to return structured data. The model will intelligently choose to output a JSON object containing arguments to call those functions.

Tutorial

Goal: Have the agent automatically get the stock price movement, retrieve related stock news, and add a new record to Airtable.

Let's get started 

Create Tools

We need 3 tools to achieve the goal:

- Get Stock Price Movement
- Get Stock News
- Add Airtable Record

Get Stock Price Movement

Create a new Tool with the following details (you can change as you want):

- Name: `get_stock_movers`
- Description: Get the stocks that has biggest price/volume moves, e.g. actives, gainers, losers, etc.

Description is an important piece as ChatGPT is relying on this to decide when to use this tool.

Edit Tool

TOOL NAME *

get_stock_movers

TOOL DESCRIPTION *

Get the stocks that has biggest price/volume moves, e.g. actives, gainers, losers, etc.

OUTPUT SCHEMA ⓘ

+ Add Item

Property	Type	Description	Required
No rows			

Delete

Save

- JavaScript Function: We are going to use [Morning Star](#) `/market/v2/get-movers` API to get data. First you have to click Subscribe to Test if you haven't already, then copy the code and paste it into JavaScript Function.
 - Add `const fetch = require('node-fetch');` at the top to import the library. You can import any built-in NodeJS [modules](#) and [external libraries](#).
 - Return the `result` at the end.

The screenshot shows the RapidAPI interface for the Morning Star API. The 'GET market/v2/get-movers' endpoint is selected. The 'Code Snippets' tab is active, showing a JavaScript fetch example. The example code is highlighted with a red box.

```

const url = 'https://morning-star.p.rapidapi.com/market/v2/get-movers';
const options = {
  method: 'GET',
  headers: {
    'X-RapidAPI-Key': '0f2167913beshbd0a3a345279246p10095c5ub9934dbdaca2',
    'X-RapidAPI-Host': 'morning-star.p.rapidapi.com'
  }
};

try {
  const response = await fetch(url, options);
  const result = await response.text();
  console.log(result);
} catch (error) {
  console.error(error);
}

```

The final code should be:

```

const fetch = require('node-fetch');
const url = 'https://morning-star.p.rapidapi.com/market/v2/get-movers';
const options = {
  method: 'GET',
  headers: {
    'X-RapidAPI-Key': 'replace with your api key',
    'X-RapidAPI-Host': 'morning-star.p.rapidapi.com'
  }
};

try {
  const response = await fetch(url, options);
  const result = await response.text();
  console.log(result);
  return result;
} catch (error) {
  console.error(error);
  return '';
}

```

You can now save it.

Get Stock news

Create a new Tool with the following details (you can change as you want):

- Name: get_stock_news
- Description: Get latest news for a stock
-

- Input Schema:
 - Property: performanceId
 - Type: string
 - Description: id of the stock, which is referred as performanceID in the API
 - Required: true

Input Schema tells LLM what to return as a JSON object. In this case, we are expecting a JSON object like below:

```
{ "performanceId": "SOME TICKER" }
```

Edit Tool

TOOL DESCRIPTION *

Get latest news for a stock

OUTPUT SCHEMA ⓘ

+ Add Item

Property	Type	Description	Required	
performanceId	string	id of the stock, which is referred as p...	✓	🗑️

Rows per page: 100 ▼

1-1 of 1

< >

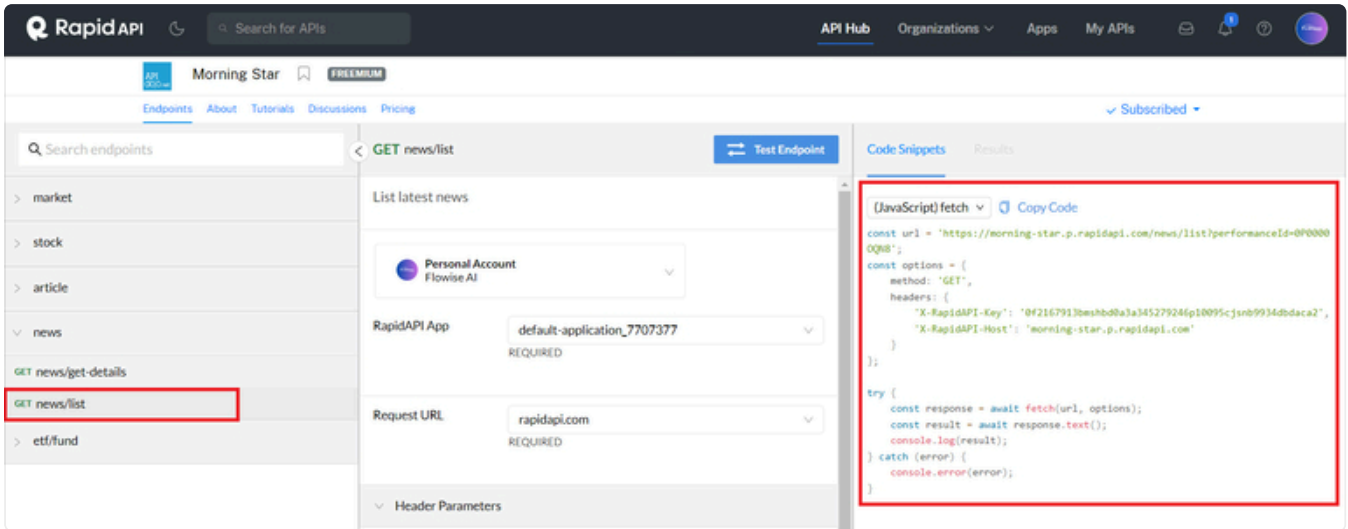
Delete

Save

- JavaScript Function: We are going to use [Morning Star](#) `/news/list` API to get the data. First you have to click Subscribe to Test if you haven't already, then copy the code and paste it into JavaScript Function.
 - Add `const fetch = require('node-fetch');` at the top to import the library. You can import any built-in NodeJS [modules](#) and [external libraries](#).
 - Return the `result` at the end.

Next, replace the hard-coded url query parameter `performanceId: 0P00000QN8` to the property variable specified in Input Schema: `$performanceId`

- You can use any properties specified in Input Schema as variables in the JavaScript Function by appending a prefix `$` at the front of the variable name.



Final code:

```
const fetch = require('node-fetch');
const url = 'https://morning-star.p.rapidapi.com/news/list?performanceId=' + $performanceId;
const options = {
  method: 'GET',
  headers: {
    'X-RapidAPI-Key': 'replace with your api key',
    'X-RapidAPI-Host': 'morning-star.p.rapidapi.com'
  }
};

try {
  const response = await fetch(url, options);
  const result = await response.text();
  console.log(result);
  return result;
} catch (error) {
  console.error(error);
  return '';
}
```

You can now save it.

Add Airtable Record

Create a new Tool with the following details (you can change as you want):

- Name: add_airtable

- Description: Add the stock, news summary & price move to Airtable

- Input Schema:

- Property: stock
- Type: string
- Description: stock ticker
- Required: true
- Property: move
- Type: string
- Description: price move in %
- Required: true
- Property: news_summary
- Type: string
- Description: news summary of the stock
- Required: true

ChatGPT will returns a JSON object like this:

```
{ "stock": "SOME TICKER", "move": "20%", "news_summary": "Some summary" }
```


Edit Tool

TOOL NAME *

add_airtable

TOOL DESCRIPTION *

Add the stock, news summary & price move to Airtable

OUTPUT SCHEMA ⓘ

+ Add Item

Property	Type	Description	Required	
stock	string	stock ticker	✓	🗑
move	string	price move in %	✓	🗑
news_summary	string	news summary of the stock	✓	🗑

Delete

Save

- JavaScript Function: We are going to use [Airtable Create Record API](#) to create a new record to an existing table. You can find the tableId and baseId from [here](#). You'll also need to create a personal access token, find how to do it [here](#).

Final code should look like below. Note how we pass in `$stock`, `$move` and `$news_summary` as variables:

```

const fetch = require('node-fetch');
const baseId = 'your-base-id';
const tableId = 'your-table-id';
const token = 'your-token';

const body = {
  "records": [
    {
      "fields": {
        "stock": $stock,
        "move": $move,
        "news_summary": $news_summary,
      }
    }
  ]
};

const options = {
  method: 'POST',
  headers: {
    'Authorization': `Bearer ${token}`,
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(body)
};

const url = `https://api.airtable.com/v0/${baseId}/${tableId}`

try {
  const response = await fetch(url, options);
  const text = await response.text();
  return text;
} catch (error) {
  console.error(error);
  return '';
}

```

You can now save it.

You should see 3 tools created:

Tools



get_stock_movers

Get the stocks that has biggest price/volume moves, e.g. actives, gainers, losers, etc.



get_stock_news

Get latest news for a stock



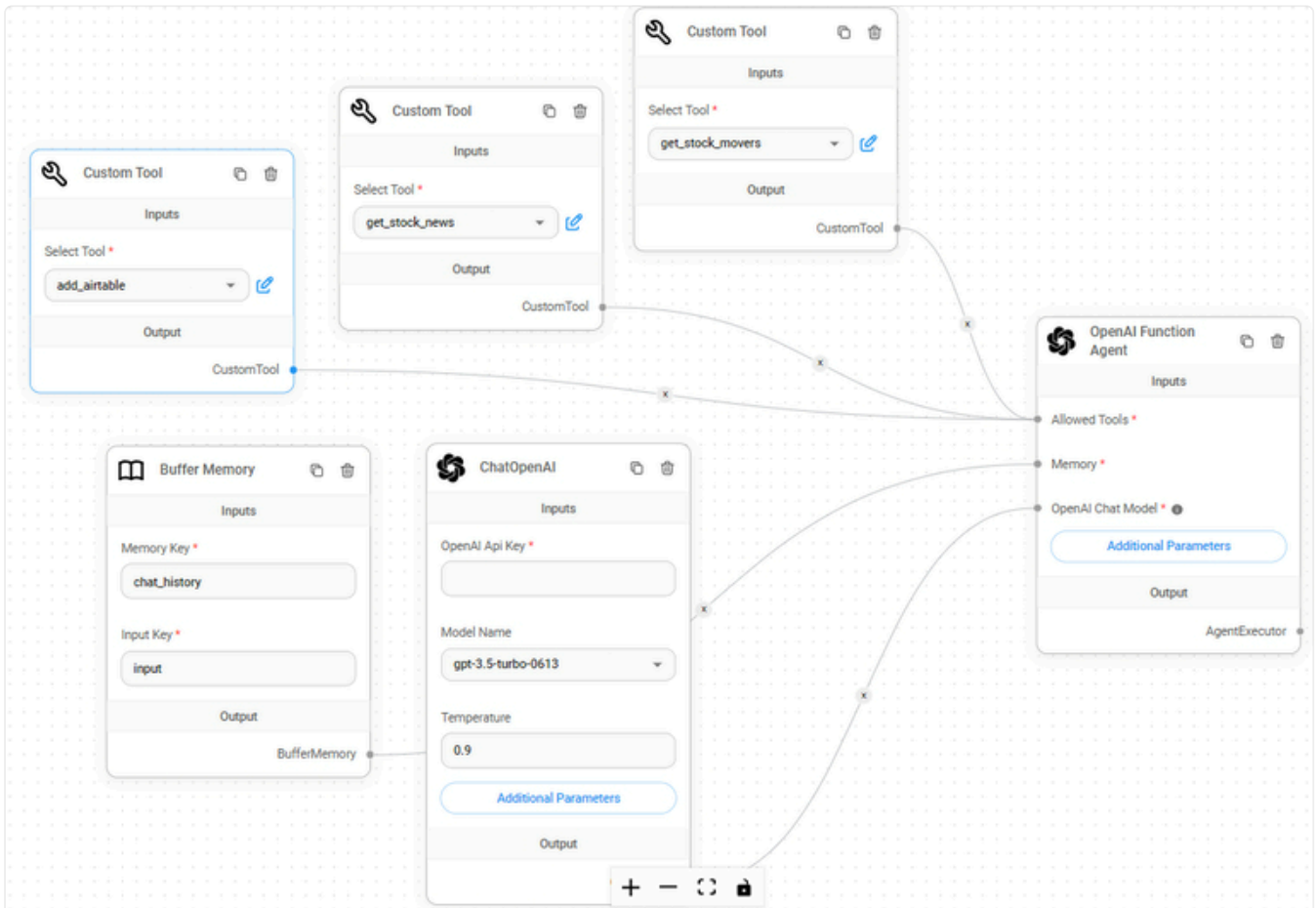
add_airtable

Add the stock, news summary & price move to Airtable

Create Chatflow

You can use the template **OpenAI Function Agent** from marketplace, and replace the tools with **Custom Tool**. Select the tool you have created.

Note: OpenAI Function Agent only supports 0613 models currently.



Save the chatflow and start testing it. For starter, you can try asking:

What is the stock that has the biggest price movement today?

The stock that has the biggest price movement today is Overstock.com (OSTK) with a price move of 17.47%.

You can then follow up with another question to get the news of that particular stock:

What are the latest news about this stock that might cause the price movement?

Here are the latest news about Overstock.com (OSTK) that might have caused the price movement:

1. *Title: "Overstock's shares soar on successful 21.5 million bid for Bed Bath & Beyond assets" Source: MarketWatch Published Date: June 22, 2023 Summary: Overstock.com's shares experienced a significant increase after successfully winning a bid for Bed Bath & Beyond assets worth 21.5 million.*
2. *Title: "Thinking about trading options or stock in Meta Platforms, Overstock.com, Walmart, Home Depot, or United Parcel Service?" Source: PR Newswire Published Date: June 22, 2023 Summary: This news article discusses the potential trading options and stocks, including Overstock.com, that investors may consider.*


Please note that these news articles are for informational purposes only and may not be the sole reason for the price movement. It is always recommended to conduct thorough research and analysis before making any investment decisions.

Lastly, you can ask ChatGPT to add a new record to Airtable:

Can you add a record to Airtable with the stock ticker, price move and news summary?

I have added a record to Airtable with the following details:

Stock Ticker: OSTK Price Move: 17.47% News Summary: Overstock.com's shares experienced a significant increase after successfully winning a bid for Bed Bath & Beyond assets worth \$21.5 million.

 **Voila!** That's how you can create your own custom tool and use it with the OpenAI Function Agent!

Additional

Pass Session ID to Function

By default, Function in custom tool has access to the following flow configurations:

```
$flow.sessionId  
$flow.chatId  
$flow.chatflowId  
$flow.input
```

Below is an example of sending the sessionId to Discord webhook:

```
const fetch = require('node-fetch');
const webhookUrl = "https://discord.com/api/webhooks/1124783587267";
const content = $content; // captured from input schema
const sessionId = $flow.sessionId;

const body = {
  "content": `${mycontent} and the sessionid is ${sessionId}`
};

const options = {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(body)
};

const url = `${webhookUrl}?wait=true`

try {
  const response = await fetch(url, options);
  const text = await response.text();
  return text;
} catch (error) {
  console.error(error);
  return '';
}
```

Pass variables to Function

In some cases, you would like to pass variables to custom tool function.

For example, you are creating a chatbot that uses a custom tool. The custom tool is executing a HTTP POST call and API key is needed for successful authenticated request. You can pass it as a variable.

By default, Function in custom tool has access to variables:

```
$vars.<variable-name>
```

Example of how to pass variables in Flowise using API and Embedded:

```
async function query(data) {
  const response = await fetch(
    "http://localhost:3000/api/v1/prediction/<chatflow-id>",
    {
      method: "POST",
      headers: {
        "Content-Type": "application/json"
      },
      body: JSON.stringify(data)
    }
  );
  const result = await response.json();
  return result;
}

query({
  "question": "Hey, how are you?",
  "overrideConfig": {
    "vars": {
      "apiKey": "abc"
    }
  }
}).then((response) => {
  console.log(response);
});
```

Example of how to receive the variables in custom tool:

```

const fetch = require('node-fetch');
const webhookUrl = "https://discord.com/api/webhooks/1124783587267";
const content = $content; // captured from input schema
const sessionId = $flow.sessionId;
const apiKey = $vars.apiKey;

const body = {
  "content": `${mycontent} and the sessionId is ${sessionId}`
};

const options = {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    'Authorization': `Bearer ${apiKey}`
  },
  body: JSON.stringify(body)
};

const url = `${webhookUrl}?wait=true`

try {
  const response = await fetch(url, options);
  const text = await response.text();
  return text;
} catch (error) {
  console.error(error);
  return '';
}

```

Override Custom Tool

Parameters below can be overridden

Parameter	Description
customToolName	tool name
customToolDesc	tool description
customToolSchema	tool schema
customToolFunc	tool function

Example of an API call to override custom tool parameters:

```

async function query(data) {
  const response = await fetch(
    "http://localhost:3000/api/v1/prediction/<chatflow-id>",
    {
      method: "POST",
      headers: {
        "Content-Type": "application/json"
      },
      body: JSON.stringify(data)
    }
  );
  const result = await response.json();
  return result;
}

query({
  "question": "Hey, how are you?",
  "overrideConfig": {
    "customToolName": "example_tool",
    "customToolSchema": "z.object({title: z.string()})"
  }
}).then((response) => {
  console.log(response);
});

```

Import External Dependencies

You can import any built-in NodeJS [modules](#) and supported [external libraries](#) into Function.

1. To import any non-supported libraries, you can easily add the new npm package to `package.json` in `packages/components` folder.

```

cd Flowise && cd packages && cd components
pnpm add <your-library>
cd .. && cd ..
pnpm install
pnpm build

```

2. Then, add the imported libraries to `TOOL_FUNCTION_EXTERNAL_DEP` environment variable. Refer [#builtin-and-external-dependencies](#) for more details.
3. Start the app


```
pnpm start
```

4. You can then use the newly added library in the **JavaScript Function** like so:

```
const axios = require('axios')
```

Watch how to add additional dependencies and import libraries

Supercharge Flowise: How to Add Dependencies and Connect to Cloudta...



[Previous](#)
[Chatflow Tool](#)

[Next](#)
[Exa Search](#)

Last updated 3 months ago

