

Example 4 — OneDrive document analysis

Analyze PDFs stored in OneDrive with Ask Sage, by sending the file to the

`/server/query_with_file` endpoint from a Power Automate flow.

Prerequisites: the Prerequisites section of the Ask Sage Power Automate docs ·

Auth & secure key storage: the Authentication section of the Ask Sage Power Automate docs.

You can get this flow two ways — **import** the ready-made solution

([Quick start](#)), or **build it step by step**

([the full guide](#)).

Quick start — import the solution

`solution/` holds an importable Power Platform solution (**OneDriveIntegration**) that bundles the flow **and** the Ask Sage custom connector it uses. The flow: a manual trigger → *Get file content* + *Get file metadata* from OneDrive → **Query with File** (Ask Sage) → Compose the answer.

1. **Import** `solution/dist/OneDriveIntegration_1_0_0_2.zip`.
Import mechanics (Solutions → Import solution, roles) are in the Importing a solution section of the Ask Sage Power Automate docs.
2. **Create the two connections** when prompted (or under ... **More** → **Connections**):
 - **OneDrive for Business** — sign in with your Microsoft account.
 - the **Ask Sage custom connector** — paste your **Ask Sage API key** (it goes in the `x-access-tokens` header; see the Authentication section of the Ask Sage Power Automate docs).
3. **Point it at your file.** The flow ships with **placeholder** file IDs. Open the flow and, on **Get file content** and **Get file metadata**, browse to and select the PDF you want to analyze (this replaces `REPLACE_WITH_YOUR_ONEDRIVE_FILE_ID` / `/Documents/YOUR_FOLDER/YOUR_FILE.pdf`).
4. **Check the model.** The flow ships with `model` set to `google-claude-46-sonnet`. If that model isn't enabled in your Ask Sage tenant, open the **Query with File** action and change it to one that is — list available models with `POST /server/get-models`.

5. **Test** → **Manually** → **Run**, enter a prompt (e.g. “Summarize this document and list key risks”), and read the result in the **Compose model analysis** step.

Why the connector is bundled: exporting a flow that uses a custom connector fails unless the connector is part of the same solution (“...requires the custom connector to be added to a dataverse solution”). This solution includes it, so import is self-contained. If you rebuild from scratch, create the custom connector **inside** the solution (or import the OpenAPI from `../connector/`) — a connector made in the standalone *Custom connectors* area can't be added to a solution after the fact.

Reading the response: `Query with File` returns the answer in the `message` field — `body('<action>')['message']`. It begins with a `<file-content>...</file-content>` echo of the extracted text; the **Compose model analysis** step strips it with `split(outputs('Compose_full_response'), '</file-content>')[1]` to keep just the analysis.

Overview

This guide will walk you through creating a Power Automate flow that integrates OneDrive files with Ask Sage. This integration allows you to automatically analyze PDF documents stored in your OneDrive using Ask Sage's AI capabilities. The flow can be triggered manually to process any PDF file and receive AI-generated insights.

Prerequisites

- Microsoft OneDrive for Business account
- Power Automate account with appropriate permissions
- Ask Sage API key (obtain from your Ask Sage account)
- Basic familiarity with Power Automate flows
- PDF documents stored in OneDrive that you want to analyze

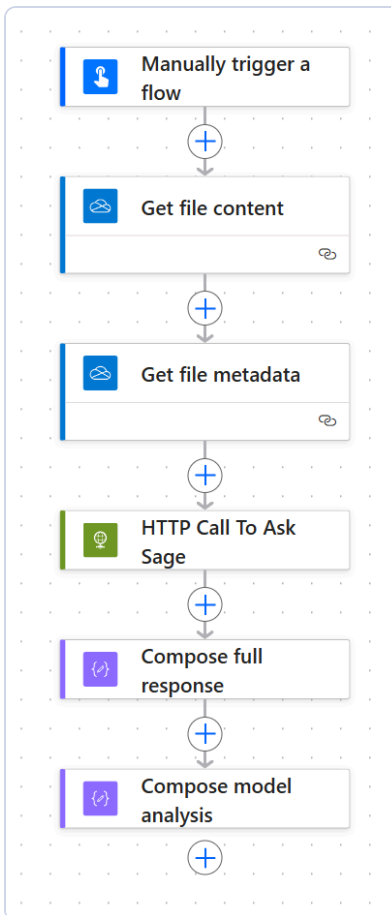
Use Cases

This integration is useful for:

- Analyzing business proposals and contracts

- Extracting insights from reports and documents
- Summarizing lengthy PDF files
- Answering questions about document content
- Generating analysis reports from uploaded files

Overall Flow



Step-by-Step Instructions

Step 1: Create a Manual Trigger

1. **Open Power Automate** and create a new instant cloud flow
2. **Search for and select** the trigger: “*Manually trigger a flow*”
3. **Name your flow** something descriptive like “OneDrive PDF Analysis with Ask Sage”
4. This trigger allows you to run the flow on-demand whenever you want to analyze a document

Note: A manual trigger gives you control over when the analysis runs, which is useful for processing specific documents as needed.

Step 2: Get File Content from OneDrive

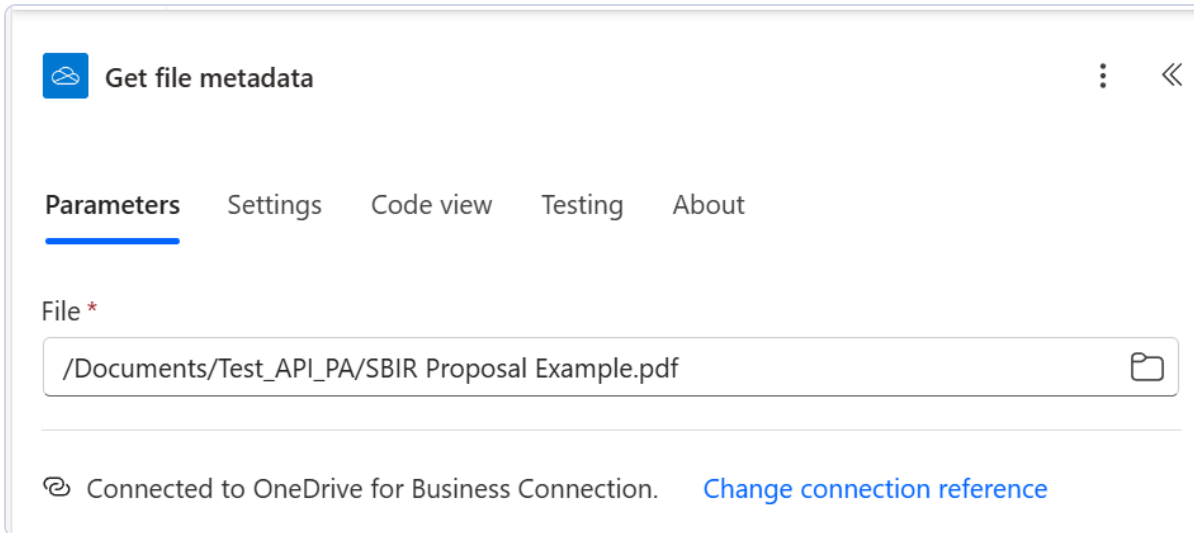
1. **Add a new action** and search for “*Get file content*” (OneDrive for Business connector)
2. **Configure the action:**
 - Click the folder icon to browse your OneDrive
 - Navigate to the location of your PDF file
 - Select the specific file you want to analyze (e.g., `/Documents/YOUR_FOLDER/YOUR_FILE.pdf`)
3. **Advanced parameters:**
 - Set “Infer Content Type” to **Yes** (this helps Power Automate understand it’s a PDF)

Tip: You can modify this later to work with different files or add a file picker input to make it more dynamic.

The screenshot shows the configuration interface for the 'Get file content' action. At the top, there is a title bar with the action name and a folder icon. Below the title bar, there are tabs for 'Parameters', 'Settings', 'Code view', 'Testing', and 'About'. The 'Parameters' tab is selected. Under 'Parameters', there is a 'File *' field with a text input containing the path '/Documents/Test_API_PA/SBIR Proposal Example.pdf' and a folder icon on the right. Below this, there is an 'Advanced parameters' section with a dropdown menu showing 'Showing 1 of 1' and buttons for 'Show all' and 'Clear all'. Underneath, there is an 'Infer Content Type' dropdown menu set to 'Yes' with a blue 'X' icon on the right. At the bottom, there is a status bar indicating 'Connected to OneDrive for Business Connection.' and a link to 'Change connection reference'.

Step 3: Get File Metadata

1. **Add a new action** and search for “*Get file metadata*” (OneDrive for Business connector)
2. **Configure the action:**
 - Select the same file path as in Step 2
 - This retrieves important information like file name, size, and modification date
3. This metadata will be useful for tracking which file was analyzed



Step 4: Set Up the Ask Sage API Call

This is the core step where your document gets sent to Ask Sage for analysis.

1. **Add an HTTP action** after the Get file metadata step
2. **Configure the HTTP request:**

URI:

```
https://api.asksage.ai/server/query_with_file
```

Method:

```
POST
```

Headers:

- `x-access-tokens` : [YOUR_API_KEY]
- `Content-Type` : `multipart/form-data; boundary=-----WebKitFormBoundary7MA4YWxkTrZu0gW`

Body:

```
-----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="message"

[Insert your question or prompt here – use dynamic content: Text]

-----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="model"

google-claude-45-sonnet

-----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="temperature"

0

-----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="tools"

["read_file"]

-----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="tools_to_execute"

["read_file"]

-----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="file"; filename="[Use dynamic content:
DisplayName]"
Content-Type: application/pdf

[Use dynamic content: File content]

-----WebKitFormBoundary7MA4YWxkTrZu0gW--
```

3. Replace the following:

- [YOUR_API_KEY] : Your actual Ask Sage API key
- [Insert your question or prompt here] : Add dynamic content from the manual trigger (if you added a text input) or write a static prompt like “Analyze this document and provide key insights”
- [Use dynamic content: DisplayName] : Select the file name from the “Get file metadata” step
- [Use dynamic content: File content] : Select the file content from the “Get file content” step

Important: The boundary string (`-----WebKitFormBoundary7MA4YWxkTrZu0gW`) must match exactly in both the Content-Type header and the body. Don't change it unless you change it everywhere.

The screenshot shows an HTTP client interface with the following configuration:

- Title:** HTTP Call To Ask Sage
- Parameters:** Settings, Code view, About
- URI:** `https://api.asksage.ai/server/query_with_file`
- Method:** POST
- Headers:**
 - Key: `x-access-tokens`, Value: `your_api_key`
 - Key: `Content-Type`, Value: `multipart/form-data; boundary=-----WebKitFormBoundary7MA4YWxkTrZu0gW`
- Queries:** Enter key, Enter value
- Body:**

```
-----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="message"

Text X
-----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="model"

google-claude-45-sonnet
-----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="temperature"

0
-----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="tools"

["read_file"]
-----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="tools_to_execute"

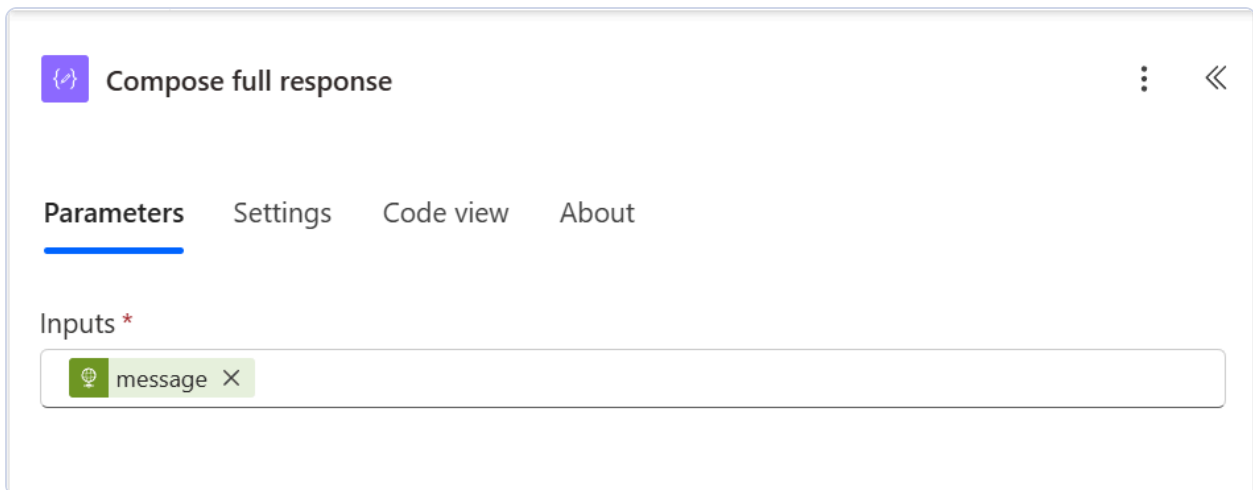
["read_file"]
-----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="file"; filename="DisplayName X"
Content-Type: application/pdf

File content X
-----WebKitFormBoundary7MA4YWxkTrZu0gW--
```

Step 5: Compose the Full Response

1. **Add a Compose action** after the HTTP call
2. **Configure the inputs:**
 - In the “Inputs” field, select the dynamic content: **message** from the “HTTP Call To Ask Sage” step. The expression is `body('HTTP_Call_To_Ask_Sage')['message']`
 - This extracts the AI-generated response from the API call

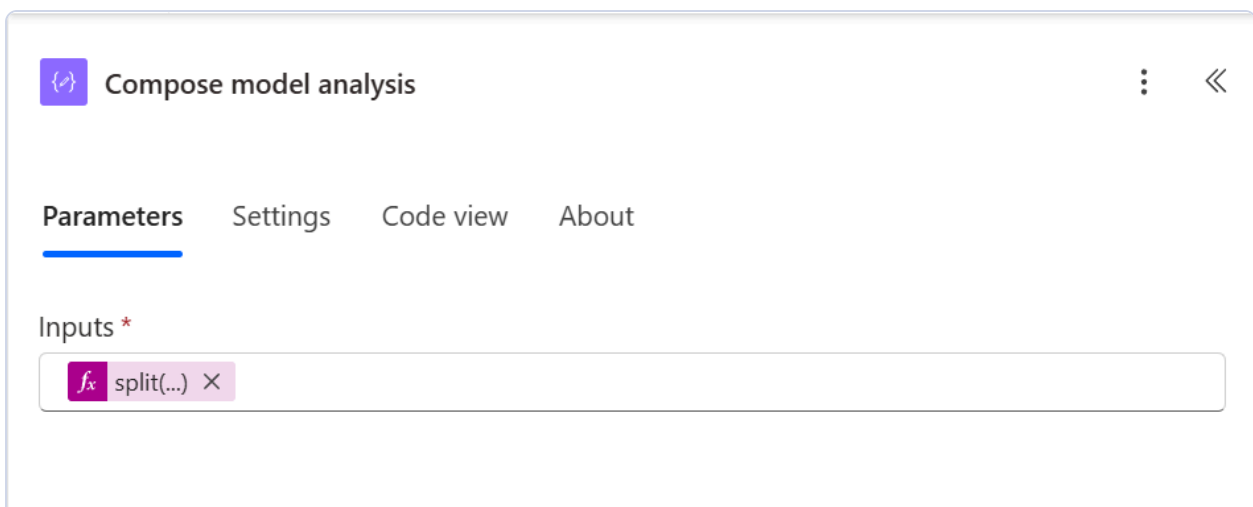
Note: This step makes it easier to work with the response in subsequent actions.



The screenshot shows the configuration for the 'Compose full response' action. The title bar includes a code icon, the text 'Compose full response', and a close icon. Below the title bar are four tabs: 'Parameters' (selected), 'Settings', 'Code view', and 'About'. Under the 'Parameters' tab, there is an 'Inputs *' section with a search bar. A single input is shown: a green pill with a globe icon, the text 'message', and a close 'X' icon.

Step 6: Compose Model Analysis (Optional)

1. **Add another Compose action** to parse the model response. The expression here is `split(outputs('Compose_full_response'), '</file-content>')[1]`
2. This will extract the model response from the full json



The screenshot shows the configuration for the 'Compose model analysis' action. The title bar includes a code icon, the text 'Compose model analysis', and a close icon. Below the title bar are four tabs: 'Parameters' (selected), 'Settings', 'Code view', and 'About'. Under the 'Parameters' tab, there is an 'Inputs *' section with a search bar. A single input is shown: a purple pill with a code icon, the text 'split(...)', and a close 'X' icon.

Step 7: Save and Test Your Flow

1. **Save your flow** with a descriptive name
2. **Test the flow:**
 - Click the “Test” button in the top-right corner
 - Select “Manually”
 - Click “Test” again
 - The flow will run and process your PDF file
 - Review the outputs in each step to ensure everything worked correctly

Understanding the API Parameters

Model Selection

- **google-claude-45-sonnet:** A powerful model for document analysis
- You can change this to other available Ask Sage models based on your needs

Temperature

- **0:** Provides more deterministic, focused responses
- Higher values (0.1-1.0) provide more creative responses
- For document analysis, keeping it at 0 is recommended for consistency

Tools

- **read_file:** Enables the AI to read and analyze the PDF content
- This must be included for file-based analysis

Customization Options

Making the Flow More Dynamic

Instead of hardcoding a file path, you can:

1. **Add file picker input to manual trigger:**
 - Edit the manual trigger
 - Add an input of type “File”
 - Use this dynamic file input in the “Get file content” step
2. **Add a text input for custom questions:**
 - Edit the manual trigger

- Add an input of type “Text” for your question
- Use this in the message field of the HTTP body

Processing Multiple Files

To analyze multiple files:

1. **Replace the manual trigger** with “*When a file is created*” or “*When a file is modified*” trigger
2. **Add a condition** to check if the file is a PDF
3. The flow will automatically process new files as they’re added to OneDrive

Storing Results

You can extend this flow to:

1. **Save results back to OneDrive:**
 - Add a “Create file” action after the Compose step
 - Save the analysis as a text file or Word document
2. **Send results via email:**
 - Add a “Send an email” action
 - Include the composed response in the email body
3. **Store in SharePoint or a database:**
 - Add actions to save results to SharePoint lists or SQL databases for tracking

Troubleshooting Tips

- **File not found error?** Double-check the file path in OneDrive
- **API authentication error?** Verify your API key is correct and active
- **Invalid / unavailable model?** The flow ships with `google-claude-46-sonnet`. If that model isn’t enabled in your tenant, change the `model` value to one that is — list them with `POST /server/get-models`
- **Empty response?** Ensure the “read_file” tool is included in both the tools and tools_to_execute arrays
- **Boundary errors?** Make sure the boundary string matches exactly in the Content-Type header and body
- **File too large?** Check Ask Sage’s file size limits and consider splitting large documents

Security Best Practices

- Store your API key securely (consider using Power Automate environment variables)

- Limit access to flows that process sensitive documents
- Regularly review and audit flow runs
- Use OneDrive permissions to control which files can be accessed
- Consider implementing approval steps for sensitive document analysis

Example Use Case: Proposal Analysis

Here's a practical example of how to use this flow:

1. **Upload a business proposal PDF** to your OneDrive folder
2. **Run the flow manually** with a prompt like: "Analyze this proposal and provide: 1) Executive summary, 2) Key strengths, 3) Potential concerns, 4) Recommendations"
3. **Review the AI-generated analysis** in the flow run history
4. **Save or email the results** for your team to review

Conclusion

You now have a functional OneDrive integration with Ask Sage! This flow allows you to leverage AI to analyze PDF documents stored in your OneDrive, making it easy to extract insights, generate summaries, and answer questions about your documents without manual review.

For more advanced features and customization options, refer to the Ask Sage API documentation or explore additional Power Automate connectors and actions.