

GPT Actions library - Google Drive



Luke Xing (OpenAI)

Aug 11, 2024



Open in Github

Introduction

This page provides an instruction & guide for developers building a GPT Action for a specific application. Before you proceed, make sure to first familiarize yourself with the following information:

- [Introduction to GPT Actions](#)
- [Introduction to GPT Actions Library](#)
- [Example of Building a GPT Action from Scratch](#)

This particular GPT Action provides an overview of how to connect to **Google Drive**, Google's File storage system. This action will allow you to list and query against file names, load the file content into your GPT, and ultimately use that data as context in ChatGPT. This set of actions is extensible by additional methods found via the [Google Drive API](#). This is great if you want a generalist GPT that can read smaller files, such as:

- Meetings minutes
- Product design documents
- Short memos
- Frequently-asked questions

For something that wants to read longer memos such as entire books, complex CSVs with many rows, we suggest building a Google Docs or Google Sheets-specific GPT.

Value + Example business case

Users can now leverage ChatGPT's natural language capability to connect directly to files in Google Drive

Example Use Cases:

- A user needs to look up which files relate to a certain topic
- A user needs an answer to a critical question, buried deep in documents

Application Information

Application Key Links

Check out these links from the application before you get started:

- Application Website: <https://www.google.com/drive/>
- Application API Documentation: <https://developers.google.com/drive/api/guides/about-sdk>

Application Prerequisites

Before you get started, make sure you have a Google Cloud account and that the Drive API is enabled:

- Set up a Google Cloud project
- Enable Google Drive API from Google API Library
- If application's "Publishing Status" is "Testing", ensure users are added to your application

ChatGPT Steps

Example Custom GPT Instructions

Once you've created a Custom GPT, to get started, copy the text below in the Instructions panel. You may have to add additional context specific to your use case. In this way, it is worth testing additional instructions you add to optimize for clarity and accuracy. Have questions? Check out [Getting Started Example](#) to see how this step works in more detail.

*** Context ***



You are an office helper who takes a look at files within Google Drive and reads in informat

*** Instructions ***

Use the 'listFiles' function to get a list of files available within docs. In this way, det

- Summaries: what happens in a given file? Please give a consistent, concise answer and rea
- Professionalism: Behave professionally, providing clear and concise responses.
- Synthesis, Coding, and Data Analysis: ensure coding blocks are explained.
- When handling dates: make sure that dates are searched using date fields and also if you d
- Clarification: Ask for clarification when needed to ensure accuracy and completeness in fu
- Privacy and Security: Respect user privacy and handle all data securely.

*** Examples of Documentation ***

Here is the relevant query documentation from Google for the listFiles function:

What you want to query Example

Files with the name "hello" name = 'hello'

Files with a name containing the words "hello" and "goodbye" name contains 'hello' and na

Files with a name that does not contain the word "hello" not name contains 'hello'

Folders that are Google apps or have the folder MIME type mimeType = 'application/vnd.goog

Files that are not folders mimeType != 'application/vnd.google-apps.folder'

Files that contain the text "important" and in the trash fullText contains 'important' an

Files that contain the word "hello" fullText contains 'hello'

Files that do not have the word "hello" not fullText contains 'hello'

Files that contain the exact phrase "hello world" fullText contains '"hello world"'

Files with a query that contains the "\" character (e.g., "\"authors") fullText contains '\"

Files with ID within a collection, e.g. parents collection '1234567' in parents

Files in an application data folder in a collection 'appDataFolder' in parents

Files for which user "test@example.org" has write permission 'test@example.org' in writer

Files for which members of the group "group@example.org" have write permission 'group@examp

Files modified after a given date modifiedTime > '2012-06-04T12:00:00' // default time zon

Files shared with the authorized user with "hello" in the name sharedWithMe and name contai

Files that have not been shared with anyone or domains (only private, or shared with specifi

Image or video files modified after a specific date modifiedTime > '2012-06-04T12:00:00' and

Example OpenAPI Schema

Once you've created a Custom GPT, copy the text below in the Actions panel. This offers an example of what you could include within functions, but . Have questions? Check out [Getting Started Example](#) to see how this step works in more detail. As well, try [ActionsGPT](#), a CustomGPT OpenAI created to help with Actions. The three examples are:

- **List Files:** this is the core action that lists the files in your drive. Within this are a few parameters, such as q , includeItemsFromAllDrives , supportsAllDrives
- **Get Metadata:** in case list doesn't work, this can offer as a backup based on certain results - for example, if users attempt to make a search via “meeting from last week”, etc
- **Export:** exports in a byte content. For more reading, please consult <https://developers.google.com/drive/api/reference/rest/v3/files/export>

Generally, if 'get' is used, the model will attempt to download the file, which may be undesirable. Thus, Export is recommended instead.



```
{
  "openapi": "3.1.0",
  "info": {
    "title": "Google Drive API",
    "description": "API for interacting with Google Drive",
    "version": "1.0.0"
  },
  "servers": [
    {
      "url": "https://www.googleapis.com/drive/v3"
    }
  ],
  "paths": {
    "/files": {
      "get": {
        "operationId": "ListFiles",
        "summary": "List files",
        "description": "Retrieve a list of files in the user's Google Drive.",
        "parameters": [
          {
            "name": "q",
            "in": "query",
            "description": "Query string for searching files.",
            "required": false,
            "schema": {
              "type": "string"
            }
          },
          {
            "name": "includeItemsFromAllDrives",
            "in": "query",
            "description": "Whether both My Drive and shared drive items should be included",
            "required": false,
            "schema": {
              "type": "string"
            }
          },
          {
            "name": "supportsAllDrives",
            "in": "query",
            "description": "Whether the requesting application supports both My Drives and s",
            "required": false,
            "schema": {
              "type": "string"
            }
          },
          {
            "name": "pageSize",
            "in": "query",
            "description": "Maximum number of files to return.",
            "required": false,
            "schema": {
              "type": "integer",
              "default": 10
            }
          }
        ]
      }
    }
  }
}
```

```

        "name": "pageToken",
        "in": "query",
        "description": "Token for continuing a previous list request.",
        "required": false,
        "schema": {
            "type": "string"
        }
    },
    {
        "name": "fields",
        "in": "query",
        "description": "Comma-separated list of fields to include in the response.",
        "required": false,
        "schema": {
            "type": "string"
        }
    }
],
"responses": {
    "200": {
        "description": "A list of files.",
        "content": {
            "application/json": {
                "schema": {
                    "type": "object",
                    "properties": {
                        "kind": {
                            "type": "string",
                            "example": "drive#fileList"
                        },
                        "nextPageToken": {
                            "type": "string",
                            "description": "Token to retrieve the next page of results."
                        },
                        "files": {
                            "type": "array",
                            "items": {
                                "type": "object",
                                "properties": {
                                    "id": {
                                        "type": "string"
                                    },
                                    "name": {
                                        "type": "string"
                                    },
                                    "mimeType": {
                                        "type": "string"
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
},
"/files/{fileId}": {
    "get": {

```

```

"operationId": "getMetadata",
"summary": "Get file metadata",
"description": "Retrieve metadata for a specific file.",
"parameters": [
  {
    "name": "fileId",
    "in": "path",
    "description": "ID of the file to retrieve.",
    "required": true,
    "schema": {
      "type": "string"
    }
  },
  {
    "name": "fields",
    "in": "query",
    "description": "Comma-separated list of fields to include in the response.",
    "required": false,
    "schema": {
      "type": "string"
    }
  }
],
"responses": {
  "200": {
    "description": "Metadata of the file.",
    "content": {
      "application/json": {
        "schema": {
          "type": "object",
          "properties": {
            "id": {
              "type": "string"
            },
            "name": {
              "type": "string"
            },
            "mimeType": {
              "type": "string"
            },
            "description": {
              "type": "string"
            },
            "createdTime": {
              "type": "string",
              "format": "date-time"
            }
          }
        }
      }
    }
  }
},
"/files/{fileId}/export": {
  "get": {
    "operationId": "export",
    "summary": "Export a file",
    "description": "Export a Google Doc to the requested MIME type.",
    "parameters": [

```

```

{
  "name": "fileId",
  "in": "path",
  "description": "ID of the file to export.",
  "required": true,
  "schema": {
    "type": "string"
  }
},
{
  "name": "mimeType",
  "in": "query",
  "description": "The MIME type of the format to export to.",
  "required": true,
  "schema": {
    "type": "string",
    "enum": [
      "application/pdf",
      "application/vnd.openxmlformats-officedocument.wordprocessingml.document",
      "text/plain"
    ]
  }
},
],
"responses": {
  "200": {
    "description": "The exported file.",
    "content": {
      "application/pdf": {
        "schema": {
          "type": "string",
          "format": "binary"
        }
      },
      "application/vnd.openxmlformats-officedocument.wordprocessingml.document": {
        "schema": {
          "type": "string",
          "format": "binary"
        }
      },
      "text/plain": {
        "schema": {
          "type": "string",
          "format": "binary"
        }
      }
    }
  },
  "400": {
    "description": "Invalid MIME type or file ID."
  },
  "404": {
    "description": "File not found."
  }
}
}
}
}
}
}
}

```

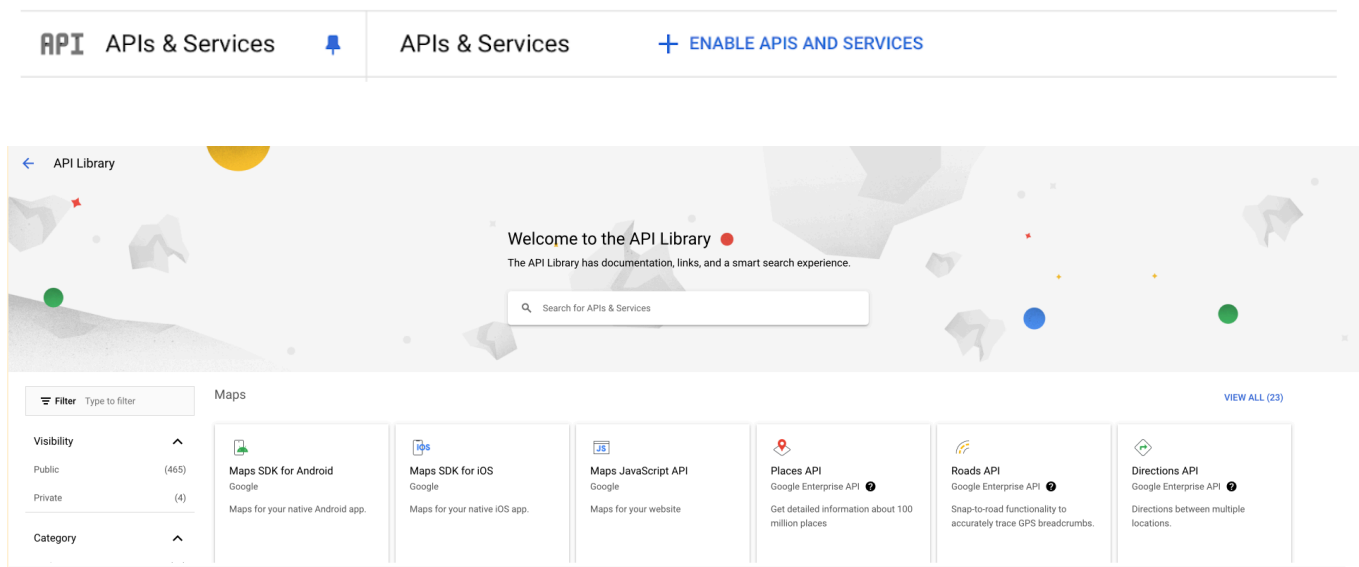
Authentication Instructions

Below are instructions on setting up authentication with this 3rd party application. Have questions? Check out [Getting Started Example](#) to see how this step works in more detail.

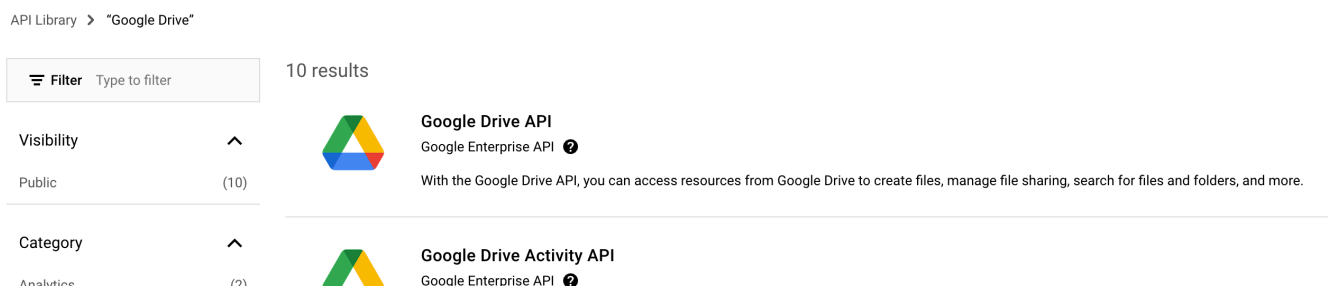
Pre-Action Steps

Before you set up authentication in ChatGPT, please take the following steps in the application.

- Go to the Google Cloud Console
- Navigate to Enabled API & Services and enable Google Drive API



- Within the search bar, search Google Drive API:



- Create new OAuth credentials (or use an existing one). Note that if you haven't set up an OAuth credentials screen, you will need to do that.

API APIs & Services

Enabled APIs & services

Library

Credentials

OAuth consent screen

Page usage agreements

OAuth consent screen

chatgpt-integration [EDIT APP](#)

Publishing status

Testing

PUBLISH APP

User type

External

- Within this process, you will need to grant access to the correct permissions, establish the primary tester as a testing email if Testing is enabled, and set up the OAuth rate limit.
- Next, go to credentials and click “+ Create Credentials” and click “Create Credentials”. Below is an example of what this screen looks like when it’s already set up.

API APIs & Services

Credentials

+ CREATE CREDENTIALS

DELETE

RESTORE DELETED CREDENTIALS

Create credentials to access your enabled APIs. [Learn more](#)

API Keys

<input type="checkbox"/>	Name	Creation date	Restrictions	Actions
No API keys to display				

OAuth 2.0 Client IDs

<input type="checkbox"/>	Name	Creation date	Type	Client ID	Actions
<input type="checkbox"/>	integration GPT	Jul 30, 2024	Web application		Edit Delete Download

Service Accounts

<input type="checkbox"/>	Email	Name	Actions
No service accounts to display			

[Manage service accounts](#)

- Locate your OAuth Client ID & Client Secret and store both values securely (see screenshot below)

Google Cloud

Search (/) for resources, docs, products, and more

APIs & Services

Client ID for Web application

DELETED

Enabled APIs & services

Library

Credentials

OAuth consent screen

Page usage agreements

Name *

Integration GPT

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

The domains of the URIs you add below will be automatically added to your OAuth consent screen as authorized domains.

Authorized JavaScript origins

For use with requests from a browser

+ ADD URI

Authorized redirect URIs

For use with requests from a web server

Additional information

Client ID

Creation date

July 30, 2024 at 2:19:59 PM GMT-7

Client secrets

If you are in the process of changing client secrets, you can manually rotate them without downtime. [Learn more](#)

Client secret

Creation date

July 30, 2024 at 2:19:59 PM GMT-7

Status

Enabled

+ ADD SECRET

In ChatGPT

In ChatGPT, click on "Authentication" and choose "OAuth". Enter in the information below.

- **Client ID:** use Client ID from steps above
- **Client Secret:** use Client Secret from steps above
- **Authorization URL:** <https://accounts.google.com/o/oauth2/auth>
- **Token URL:** <https://oauth2.googleapis.com/token>
- **Scope:** <https://www.googleapis.com/auth/drive>
 - **Note:** for a list of more detailed scopes enabled, please refer to [Google's OAuth 2.0 guide](#).
- **Token:** Default (POST)
- **Privacy Policy:** <https://policies.google.com/privacy?hl=en-US>

Post-Action Steps

Once you've set up authentication in ChatGPT, follow the steps below in the application to finalize the Action.

- Copy the callback URL from the GPT Action

Actions

www.googleapis.com

Callback URL

<https://chat.openai.com/a>

Create new action

- In the “Authorized redirect URIs”, add your callback URL

Authorized redirect URIs

For use with requests from a web server



URIs 1 *
https://chat.openai.com/ [REDACTED]

FAQ & Troubleshooting

- *Callback URL Error:* If you get a callback URL error in ChatGPT, pay close attention to the screenshot above. You need to add the callback URL directly into GCP for the action to authenticate correctly.

Are there integrations that you'd like us to prioritize? Are there errors in our integrations? File a PR or issue in our GitHub, and we'll take a look.

gd2md-html: xyzzy Mon Aug 12 2024