

Objective:

Using the capabilities of FlowiseAI create a fully automatic & autonomous script/mechanism that is writing content produced here (by Cognosys) and saving it to hdd as files in proper formats and proper location in the local clone of NeuralGPT repository (E:\AI\NeuralGPT\NeuralGPT). It can also run the new/modified scripts 'on the fly' to test their practical functionality. Below are examples of scripts produced by FlowiseAI for agents that reads PDF and can save files to e:\ai

Task:

1. Create a Python script that utilizes the PyPDF2 library to read a PDF file and extract its text content. Save the extracted text content to a text file in the E:\AI\NeuralGPT\NeuralGPT directory with the same name as the original PDF file, but with a .txt extension.

Result:

```
import os
import PyPDF2

# Set the directory path
directory_path = "E:\\AI\\NeuralGPT\\NeuralGPT"

# Get all the PDF files in the directory
pdf_files = [f for f in os.listdir(directory_path) if f.endswith('.pdf')]

# Loop through each PDF file and extract its text content
for pdf_file in pdf_files:
    # Open the PDF file
    with open(os.path.join(directory_path, pdf_file), 'rb') as f:
        pdf_reader = PyPDF2.PdfFileReader(f)
        # Get the text content of the PDF file
        text_content = ""
        for page_num in range(pdf_reader.numPages):
            page_obj = pdf_reader.getPage(page_num)
            text_content += page_obj.extractText()

# Save the extracted text content to a text file with the same name as the original PDF file
output_file = os.path.splitext(pdf_file)[0] + ".txt"
with open(os.path.join(directory_path, output_file), 'w') as f:
    f.write(text_content)
```

This code reads all the PDF files in the specified directory, extracts their text content using PyPDF2 library, and saves the extracted text content to a text file with the same name as the original PDF file, but with a .txt extension. The text files are saved in the same directory as the PDF files.

Task:

2. Modify the script created in task 1 to also save the extracted text content to a JSON file in the E:\AI\NeuralGPT\NeuralGPT directory with the same name as the original PDF file, but with a .json extension. The JSON file should contain a dictionary with the following keys: "filename" (the name of the original PDF file), "text" (the extracted text content), and "timestamp" (the current date and time).

Result:

Here's the modified Python script that saves the extracted text content to both a text file and a JSON file:

```
import os
import json
import datetime
import PyPDF2

# Define the path to the PDF file
pdf_path = "path/to/pdf/file.pdf"

# Open the PDF file in read-binary mode
with open(pdf_path, "rb") as pdf_file:
    # Create a PDF reader object
    pdf_reader = PyPDF2.PdfFileReader(pdf_file)

    # Extract the text content from the PDF file
    text_content = ""
    for page in range(pdf_reader.getNumPages()):
        text_content += pdf_reader.getPage(page).extractText()

# Define the path to the output files
output_dir = "E:/AI/NeuralGPT/NeuralGPT"
output_name = os.path.splitext(os.path.basename(pdf_path))[0]
text_output_path = os.path.join(output_dir, output_name + ".txt")
json_output_path = os.path.join(output_dir, output_name + ".json")

# Save the extracted text content to a text file
with open(text_output_path, "w") as text_file:
    text_file.write(text_content)

# Save the extracted text content to a JSON file
data = {
    "filename": os.path.basename(pdf_path),
    "text": text_content,
    "timestamp": str(datetime.datetime.now())
}
with open(json_output_path, "w") as json_file:
    json.dump(data, json_file)
```

This script opens a PDF file, extracts its text content using PyPDF2, and saves the extracted text to both a text file and a JSON file in the specified directory. The JSON file contains a dictionary with the filename, text content, and current timestamp.