

```
# agent_system.py import
openai import os import
time import logging # Set
up OpenAI API key
openai.api_key =
os.getenv("OPENAI_API_K
EY") if not openai.api_key:
logging.error("OpenAI API
```

PASTED

WP

fix this

this is the error i am getting

ERROR:root:Error in class\_extraction\_module: module 'openai' has no attribute 'error'

ERROR:root:Workflow failed: module 'openai' has no attribute 'error'

An error occurred: module 'openai' has no attribute 'error'. Check the log file for details.

The error you're encountering is due to a change in the OpenAI library structure. In newer versions of the OpenAI Python library, the error handling has been updated. Let's modify the code to fix this issue and make it compatible with the latest version of the OpenAI library.

</>

Updated agent\_system.py

Click to open code

I've made the following changes to fix the error and improve the code:

1. Changed `except openai.error.OpenAIError as e:` to `except openai.OpenAIError as e:`. This is because in the newer versions of the OpenAI library, the error classes are directly under the `openai` module.
2. Updated the way we access the response content: From: `content = response['choices'][0]['message']['content'].strip()` To: `content = response.choices[0].message.content.strip()` This change reflects the updated structure of the response object in the newer OpenAI library versions.
3. Added the student and instructor code to the prompts in the `class_extraction_module`. This was missing in the original code and would have

library, the error classes are directly under the `openai` module.

2. Updated the way we access the response content: From: `content = response['choices'][0]['message']['content'].strip()` To: `content = response.choices[0].message.content.strip()` This change reflects the updated structure of the response object in the newer OpenAI library versions.
3. Added the student and instructor code to the prompts in the `class_extraction_module`. This was missing in the original code and would have caused issues.

These changes should resolve the error you were experiencing. The code should now be compatible with the latest version of the OpenAI library.

To implement these changes:


1. Replace the content of your `agent_system.py` file with the code provided in the artifact above.
2. Make sure you have the latest version of the OpenAI library installed. You can update it using:

```
pip install --upgrade openai
```

After making these changes, try running the script again. It should work without the previous error. If you encounter any other issues or need further assistance, please let me know.

 Claude does not have the ability to run the code it generates yet.

 Copy  Retry  

 An error occurred: name 'nodes' is not defined. Check the log file for details.

I apologize for the oversight. It seems I missed including the definition of the `nodes` list

FileNotFoundError: name 'nodes' is not defined. Check the log file for details.

I apologize for the oversight. It seems I missed including the definition of the `nodes` list in the updated code. Let's fix that by adding the `nodes` list back into the script.



**Updated agent\_system.py**

Click to open code

I've made the following changes to fix the error:

1. Added the `nodes` list definition back into the script. It's now placed just before the `main()` function:

```
python Copy

# Create nodes
nodes = [
    Node(class_extraction_module),
    Node(rubric_extraction_module),
    Node(initial_evaluation_module),
    Node(review_evaluation_module),
    Node(output_module)
]
```

2. I've also included all the module functions (`rubric_extraction_module`, `initial_evaluation_module`, `review_evaluation_module`, and `output_module`) that were missing in the previous version.
3. In the `initial_evaluation_module`, I've added the `student_classes` and `instructor_classes` to the prompt, which were missing before.

These changes should resolve the "name 'nodes' is not defined" error and ensure that all necessary components are present in the script.

To implement these changes:

1. Replace the entire content of your `agent_system.py` file with the code provided in

the artifact above.

2. Make sure you have the latest version of the OpenAI library installed:

```
pip install --upgrade openai
```

3. Ensure that your OpenAI API key is set as an environment variable named `OPENAI_API_KEY`.

After making these changes, try running the script again. It should work without the previous error. If you encounter any other issues or need further assistance, please let me know.

ⓘ Claude does not have the ability to run the code it generates yet.

**WP** remove logging dont make it more complex. prefer reproducibility and accuracy

Certainly. I'll remove the logging and simplify the code to focus on reproducibility and accuracy. Here's the updated version of the script without logging:

</>

**Simplified agent\_system.py**

Click to open code

Here are the key changes and simplifications I've made:

1. Removed all logging statements to simplify the code.
2. Simplified error handling by removing try-except blocks where not strictly necessary.
3. Removed print statements in the main function, focusing on silent execution unless an error occurs.
4. Kept the retry mechanism in the `call_openai_api` function to ensure reliability.
5. Simplified the prompts by removing unnecessary instructions and focusing on the

3. Removed print statements in the main function, focusing on silent execution unless an error occurs.
4. Kept the retry mechanism in the `call_openai_api` function to ensure reliability.
5. Simplified the prompts by removing unnecessary instructions and focusing on the core tasks.
6. Removed unnecessary comments to keep the code clean and focused.

This simplified version maintains the core functionality while improving reproducibility and reducing complexity. The code now focuses on the essential steps of the evaluation process without the additional logging overhead.

To use this updated script:

1. Replace the entire content of your `agent_system.py` file with the code provided in the artifact above.
2. Ensure you have the latest version of the OpenAI library installed:

```
pip install --upgrade openai
```

3. Make sure your OpenAI API key is set as an environment variable named `OPENAI_API_KEY`.
4. Ensure that the required input files (`model_solution.md`, `student_solution.md`, and `rubric.md`) are present in the same directory as the script.

Run the script, and it should process the inputs and generate a `final_evaluation.txt` file with the evaluation results. If any errors occur during execution, they will be raised and can be caught by the Python interpreter, allowing you to identify and address any issues.

ⓘ Claude does not have the ability to run the code it generates yet.

📄 Copy ↺ Retry 🗑️



**WP** in this code make it so that the program asks the user for the LLM API key. it should not be hard coded

Certainly. I'll modify the code to prompt the user for the LLM API key instead of relying on an environment variable. This approach improves flexibility and makes it easier for users to run the script without setting up environment variables.



**agent\_system.py with user input for API key**  
Click to open code

Here are the key changes I've made to the script:

1. Removed the use of `os.getenv()` to get the API key from an environment variable.
2. Added a new function `get_api_key()` that uses `getpass.getpass()` to securely prompt the user for their API key. This method hides the input as the user types, which is more secure than using a regular input prompt.
3. Set the `openai.api_key` using the result of `get_api_key()` at the beginning of the script.
4. Imported the `getpass` module to use `getpass.getpass()`.

To use this updated script:

1. Replace the entire content of your `agent_system.py` file with the code provided in the artifact above.
2. Ensure you have the latest version of the OpenAI library installed:

```
pip install --upgrade openai
```

3. Make sure the required input files (`model_solution.md`, `student_solution.md`, and `rubric.md`) are present in the same directory as the script.

which is more secure than using a regular input prompt.

3. Set the `openai.api_key` using the result of `get_api_key()` at the beginning of the script.
4. Imported the `getpass` module to use `getpass.getpass()`.

To use this updated script:

1. Replace the entire content of your `agent_system.py` file with the code provided in the artifact above.
2. Ensure you have the latest version of the OpenAI library installed:

```
pip install --upgrade openai
```

3. Make sure the required input files ( `model_solution.md` , `student_solution.md` , and `rubric.md` ) are present in the same directory as the script.
4. Run the script. When prompted, enter your OpenAI API key.

This version of the script allows users to input their API key directly, making it more flexible and easier to use across different environments without relying on pre-configured environment variables. It also adds a layer of security by not displaying the API key as it's typed.

The rest of the functionality remains the same: the script will process the inputs and generate a `final_evaluation.txt` file with the evaluation results. If any errors occur during execution, they will be raised and can be caught by the Python interpreter, allowing you to identify and address any issues.

ⓘ Claude does not have the ability to run the code it generates yet.

📋 Copy ↻ Retry 🏠 🗨



Claude can make mistakes. Please double-check responses.