```
###### VERISON2
import inspect
import os
import threading

from dotenv import load_dotenv

from scripts.auto_tests_docs.docs import DOCUMENTATION_WRITER_SOP
from swarm_models import OpenAIChat
from swarms.structs.majority_voting import MajorityVoting
from swarms.structs.stackoverflow_swarm import StackOverflowSwarm
from swarms.structs.task_queue_base import TaskQueueBase

##########


######################
load_dotenv()

api_key = os.getenv("OPENAI_API_KEY")

model = OpenAIChat(
    openai_api_key=api_key,
    max_tokens=4000,
)
```

```
def process_documentation(cls):
    """
    Process the documentation for a given class using OpenAI model and save it in a Markdown file.
    """
    doc = inspect.getdoc(cls)
    source = inspect.getsource(cls)
    input_content = (
        "Class Name:"
        f" {cls.__name__}\n\nDocumentation:\n{doc}\n\nSource"
        f" Code:\n{source}"
    )

      # Process with OpenAI model (assuming the model's __call__ method takes this input and returns processed content)
    processed_content = model(
        DOCUMENTATION_WRITER_SOP(input_content, "swarms.structs")
    )

    # doc_content = f"# {cls.__name__}\n\n{processed_content}\n"
    doc_content = f"{processed_content}\n"

    # Create the directory if it doesn't exist
    dir_path = "docs/swarms/tokenizers"
    os.makedirs(dir_path, exist_ok=True)
```

```
    # Write the processed documentation to a Markdown file
    file_path = os.path.join(dir_path, f"{cls.__name__.lower()}.md")
    with open(file_path, "w") as file:
        file.write(doc_content)

    print(f"Documentation generated for {cls.__name__}.")


def main():
    classes = [
        MajorityVoting,
        StackOverflowSwarm,
        TaskQueueBase,
    ]
    threads = []
    for cls in classes:
        thread = threading.Thread(
            target=process_documentation, args=(cls,)
        )
        threads.append(thread)
        thread.start()

    # Wait for all threads to complete
    for thread in threads:
        thread.join()
```

```
    print("Documentation generated in 'swarms.structs' directory.")


if __name__ == "__main__":
    main()
```