Code Documentation Generation

1 Purpose

The script automates the generation of documentation for a provided Python code file. It involves multiple steps: extracting the code, using an agent to generate initial documentation, formatting it in LaTeX, correcting any issues, and finally saving the output as a .tex file.

1.1 Inputs

- `code`: The content of the Python script from which documentation is generated.

1.2 Outputs

- A LaTeX formatted document containing the documentation for the provided code.

2 Agents Used

2.1 Generator Agent

The generator agent is responsible for creating initial English documentation based on the provided code. It follows specific instructions to generate concise descriptions of classes, models, endpoints, functions, and scripts in a structured format.

2.2 Formator Agent

This agent formats the generated documentation into LaTeX structure, ensuring each section has subsections with input, output, and brief explanations.

2.3 Corrector Agent

The corrector agent checks and corrects any errors or AI artifacts in the formatted LaTeX document to ensure it is valid LaTeX without additional text like —latex at the beginning.

3 Steps

3.1 Extracting Code

The script reads the current file's content using `os.path.abspath(file)` and `open(current_file, "r", encoding="utf-8")`.

3.2 Generating Documentation

Using the generator agent with a specific system prompt, it generates documentation for the provided code.

3.3 Formatting Documentation

The formator agent formats the generated documentation into LaTeX structure using sections and subsections as per the instructions.

3.4 Correcting Documentation

The corrector agent checks and corrects any errors or artifacts in the formatted document to ensure it is valid LaTeX.

3.5 Saving Output

Finally, the script saves the corrected LaTeX documentation to a specified output directory with a filename `"documentation_output.tex"`.