
AI-DoxygenCleaner

Ismial Al Shuaybi

Aug 04, 2023

CONTENTS:

1	main package	1
1.1	Subpackages	1
1.1.1	main.DoxygenWarningFixer package	1
1.1.1.1	Submodules	1
1.1.1.2	main.DoxygenWarningFixer.doxygen_warning_fixer module	1
1.1.1.3	Module contents	2
1.1.2	main.ai_doxygen_cleaner package	2
1.1.2.1	Submodules	2
1.1.2.2	main.ai_doxygen_cleaner.ai_doxygen_cleaner module	2
1.1.2.3	Module contents	4
1.1.3	main.ai_language_model package	4
1.1.3.1	Submodules	4
1.1.3.2	main.ai_language_model.ai_language_model module	4
1.1.3.3	Module contents	5
1.1.4	main.config package	5
1.1.4.1	Submodules	5
1.1.4.2	main.config.config_paths module	5
1.1.4.3	main.config.config_variables module	5
1.1.4.4	Module contents	6
1.1.5	main.data package	6
1.1.5.1	Module contents	6
1.1.6	main.db_operations package	6
1.1.6.1	Submodules	6
1.1.6.2	main.db_operations.db_insertion_handler module	6
1.1.6.3	Module contents	11
1.1.7	main.doxy_warning_post_processor package	11
1.1.7.1	Submodules	11
1.1.7.2	main.doxy_warning_post_processor.doxy_warning_post_processor module	11
1.1.7.3	Module contents	11
1.1.8	main.doxygen_management package	11
1.1.8.1	Submodules	11
1.1.8.2	main.doxygen_management.doxy_warning_detector module	11
1.1.8.3	main.doxygen_management.doxyfile_configurator module	14
1.1.8.4	main.doxygen_management.file_handler module	16
1.1.8.5	Module contents	19
1.1.9	main.error_handler package	19
1.1.9.1	Submodules	19
1.1.9.2	main.error_handler.error_handler module	19
1.1.9.3	Module contents	20
1.1.10	main.print_data package	20

1.1.10.1	Submodules	20
1.1.10.2	main.print_data.print_data module	20
1.1.10.3	Module contents	22
1.1.11	main.transport_data package	22
1.1.11.1	Submodules	22
1.1.11.2	main.transport_data.transport_data module	22
1.1.11.3	Module contents	23
1.1.12	main.utils package	23
1.1.12.1	Submodules	23
1.1.12.2	main.utils.runtime_measurement module	23
1.1.12.3	main.utils.utils module	24
1.1.12.4	Module contents	26
1.2	Submodules	26
1.3	main.main module	26
1.4	Module contents	26
2	Indices and tables	27
	Python Module Index	29
	Index	31

MAIN PACKAGE

1.1 Subpackages

1.1.1 main.DoxygenWarningFixer package

1.1.1.1 Submodules

1.1.1.2 main.DoxygenWarningFixer.doxygen_warning_fixer module

class main.DoxygenWarningFixer.doxygen_warning_fixer.**DoxygenWarningFixer**(*prompt_instruction='default_instruction'*,
prompt_input_str='default_instruction',
api_key='default_instruction')

Bases: GPT3_5TurboModel

DoxygenWarningFixer class for fixing Doxygen warnings.

Methods

run(args):

Run the warning fixer based on the provided arguments.

wait():

Print waiting time every second.

print_query(self, pos):

Print the query that is sent to the OpenAI server and its corresponding response.

_concat_warnings_details(self, header_file_name_pre_fix_list, file_content_pre_fix_list, warning_content_pre_fix_llist, warning_line_number_pre_fix_llist):

Concatenate warning details to generate the prompt_input_str_list.

_fix_warnings(self, prompt_input_str_list, prompt_instruction):

Use the GPT model to fix the warnings and return the post-fix file content list.

print_query(pos)

Print the query that is sent to the OpenAI server and its corresponding response.

Parameters

pos

[int] The position of the query in the sequence.

run()

Run the warning fixer based on the provided arguments.

This method loads the warning details from `preprocessed_warnings_data.json`, sends them to the GPT model, gets the fixed content, and saves the fixed content to `postprocessed_warnings_data.json`.

wait()

Print waiting time every second.

Continuously prints elapsed waiting time every second as long as the thread is running.

1.1.1.3 Module contents

1.1.2 main.ai_doxygen_cleaner package

1.1.2.1 Submodules

1.1.2.2 main.ai_doxygen_cleaner.ai_doxygen_cleaner module

class `main.ai_doxygen_cleaner.ai_doxygen_cleaner.AIDoxygenCleaner`

Bases: `WarningPostProcessor`, `DBInsertionHandler`, `DoxygenWarningFixer`

This class is responsible for managing the Doxygen warnings, it consists of several methods for configuring Doxygen, detecting warnings, fixing them, saving the fixed warnings and printing warning details to a file.

...

Methods

doxygen_configuration():

Configures Doxygen.

detect_warnings_pre_fix():

Detects warnings before fixing.

insert_warnings_to_db():

Inserts detected warnings into the database.

fix_doxygen_warning():

Fixes detected Doxygen warnings.

apply_fix():

Applies the fix to the Doxygen warnings.

detect_warnings_post_fix():

Detects any remaining warnings after applying the fix.

print_pre_post_fix_data_to_file():

Prints warning details to a file before and after the fix.

generate_html_diff_page():

Generates an HTML diff page to show differences before and after the fix.

update_warnings_details_post_fix_in_db():

Update warning details in the database after applying fixes.

detect_warnings_post_fix()

Detects Doxygen warnings after applying the fix.

Prints the phase and step of the process. Runs the warning post processor to detect any remaining warnings.

detect_warnings_pre_fix()

This method is responsible for detecting Doxygen warnings before any fixes are applied. It stores the warning data and saves it in a JSON file for further processing.

The entire process is printed in the console for visibility.

doxygen_configuration()

Configures Doxygen.

This method is responsible for configuring Doxygen. It removes any existing Doxyfile in the directory and creates a new one. The new Doxyfile is then configured using provided parameters.

The entire process is printed in the console for visibility.

fix_doxygen_warning()

Fixes detected Doxygen warnings.

Prints the phase and step of the process. Runs the fixer to fix the warnings.

generate_html_diff_page()

Prints warning details to a file before and after the fix.

Prints the phase and step of the process. Loads warning details from JSON files and redirects the print to output files.

insert_warnings_to_db()

Inserts detected warnings into the database.

Prints the phase and step of the process. Insert details of the warnings into the database.

print_pre_post_fix_data_to_file()

Prints warning details to a file before and after the fix.

Prints the phase and step of the process. Loads warning details from JSON files and redirects the print to output files.

update_warnings_details_post_fix_in_db()

Update warning details in the database after applying fixes.

This method handles loading of the post fix data from JSON files, retrieving the pipeline ID and then updating the warning details in the database with the help of the DBInsertionHandler instance.

Prints:

Various status messages with timestamps indicating the start and end of the database update operation.

1.1.2.3 Module contents

1.1.3 main.ai_language_model package

1.1.3.1 Submodules

1.1.3.2 main.ai_language_model.ai_language_model module

```
class main.ai_language_model.ai_language_model.AILanguageModel(prompt_instruction,  
                                                                prompt_input_str)
```

Bases: ABC

AILanguageModel is an abstract class that serves as a base for AI Language Models. This class outlines the fundamental methods and attributes derived AI language models should possess.

...

Attributes

prompt_content

[str] The content of the prompt sent to the AI model.

_response

[any] The raw response returned from the AI model.

response_content

[str] The extracted content from the AI model response.

prompt_instruction

[str] The instructions that guide the AI model's response.

prompt_input_str

[str] The input string that the AI model will respond to.

Methods

_send_query():

Sends a query to the AI model, including creating the prompt, creating the response, and setting and validating the response content.

_get_response_content() -> str:

Returns the content of the response from the AI model.

_get_prompt_content() -> str:

Returns the content of the prompt that was sent to the AI model.

create_prompt_content():

Creates the content of the prompt by concatenating the instruction and the input string.

_create_response(prompt_content):

Abstract method for creating a response, to be implemented by a specific AI model class.

set_response_content():

Abstract method for setting the response content, to be implemented by a specific AI model class.

get_num_tokens_from_response(string: str, encoding_name: str) -> int:

Abstract method for getting the number of tokens from a response, to be implemented by a specific AI model class.

validate_response_content():

Abstract method for validating the response content, to be implemented by a specific AI model class.

create_prompt_content()

Creates the content of the prompt by concatenating the instruction and the input string.

get_num_tokens_from_response(string: str, encoding_name: str) → int

Gets the number of tokens in the response. This is an abstract method that must be implemented in a subclass.

Parameters**string**

[str] String from which to count tokens.

encoding_name

[str] Name of the encoding method to use.

Returns**int**

Number of tokens in the string.

abstract set_response_content()

Sets the processed content of the language model's response. This is an abstract method that must be implemented in a subclass.

validate_response_content()

Method for validating the response content, to be implemented by a specific AI model class.

1.1.3.3 Module contents

1.1.4 main.config package

1.1.4.1 Submodules

1.1.4.2 main.config.config_paths module

1.1.4.3 main.config.config_variables module

This module defines the configuration parameters for Doxygen.

The *dox_config_parameters* variable is a list of tuples. Each tuple contains two elements:

1. The Doxygen configuration parameter as a string, followed by '{' and '='.
2. The number of spaces to insert between the parameter and the '=' sign.

In the actual Doxygen configuration file, each parameter is followed by either ‘YES’ or ‘NO’.

1.1.4.4 Module contents

1.1.5 main.data package

1.1.5.1 Module contents

1.1.6 main.db_operations package

1.1.6.1 Submodules

1.1.6.2 main.db_operations.db_insertion_handler module

class main.db_operations.db_insertion_handler.DBInsertionHandler

Bases: object

Handles database insert operations for doxygen warnings.

This class manages connections to a MySQL database and provides methods for inserting information related to doxygen warnings, such as details about the pipeline, project, and warnings themselves. ...

Attributes

__db_name

[str] Name of the database.

__db_host

[str] Host of the database.

__db_user

[str] Username to connect to the database.

__db_pass

[str] Password to connect to the database.

last_header_file_id

[int] The ID of the last inserted header file.

last_project_id

[int] The ID of the last inserted project.

pipeline_link

[str] The link to the GitLab pipeline.

branch_name

[str] The branch name in the repository.

pipeline_id

[int] The ID of the pipeline.

cursor

[Cursor] A MySQL database cursor to execute SQL commands.

db

[Connection] A MySQL database connection.

Methods

connect_to_database():

Connects to the MySQL database using environment variables.

create_cursor():

Creates a new cursor to execute database commands.

set_pipeline_data():

Sets pipeline data by getting information from environment variables. If no pipeline ID is provided, generates a pseudo-random one.

update_last_project_id():

Updates the last_project_id member variable with the most recent project ID.

update_last_header_file_id():

Updates the last_header_file_id member variable with the most recent header file ID.

execute_sql(sql: str, values: tuple, optional):

Executes a SQL command.

insert_into_pipeline_table():

Inserts pipeline details into the pipeline table.

insert_into_project_table(project_folder_path: str):

Inserts project details into the project table and updates the last project ID.

insert_into_header_file_table(file_name: str, file_content: str, number_of_warnings: int):

Inserts header file details into the header file table and updates the last header file ID.

insert_into_warning_table_pre_fix(warning_content: str, line_number_pre_fix: int, fixed_status: int = FIXED_STATUS_DEFAULT):

Inserts warning details into the warning table.

insert_warnings_details(project_folder_path: str, file_name_list: list, file_content_list: list, number_of_warnings_list: list, warning_content_llist: list, line_number_pre_fix_llist: list):

Wrapper function that inserts pipeline, project, header file, and warning details into their respective tables. Once the data is inserted, it commits the changes and closes the database connection.

update_post_fix_warning_details(self, pipeline_id, file_name_list, file_content_list, line_number_llist, warnings_content_llist): Update the database with the details of post-fix warnings.

connect_to_database()

Establishes a connection to the MySQL database using the details obtained from the environment variables.

Raises

Exception

If there is an error while establishing the database connection.

create_cursor()

Creates a new cursor to execute database commands.

Raises

Exception

If there is an error while creating a cursor.

insert_into_header_file_table(*file_name*, *file_content*, *number_of_warnings*)

Inserts header file details into the header file table and updates the last header file ID.

Parameters

file_name

[str] The name of the header file.

file_content

[str] The content of the header file.

number_of_warnings

[int] The number of warnings in the header file.

Raises

Exception

If there is an error while executing the SQL command.

insert_into_pipeline_table()

Inserts pipeline details into the pipeline table in the database.

Raises

Exception

If there is an error while executing the SQL command.

insert_into_project_table(*project_folder_path*)

Inserts project details into the project table and updates the last project ID.

Parameters

project_folder_path

[str] The path to the project folder.

Raises

Exception

If there is an error while executing the SQL command.

insert_into_warning_table_pre_fix(*warning_content*, *line_number_pre_fix*, *fixed_status=1*)

Inserts the warning details into the warning table before the warnings are resolved by ChatGPT.

Parameters

warning_content

[str] The content of the warning.

line_number_pre_fix

[int] The line number of the warning before fixing by ChatGPT.

fixed_status

[int, optional] The fixed status of the warning (default is FIXED_STATUS_DEFAULT).

Raises

Exception

If there is an error while executing the SQL command.

insert_warnings_details(*project_folder_path*, *file_name_list*, *file_content_list*, *number_of_warnings_list*,
warning_content_llist, *line_number_pre_fix_llist*)

Wrapper function that inserts pipeline, project, header file, and warning details into their respective tables. Once the data is inserted, it commits the changes and closes the database connection.

Parameters

project_folder_path

[str] The path to the project folder.

file_name_list

[list] The list of file names.

file_content_list

[list] The list of file contents.

number_of_warnings_list

[list] The list of the number of warnings per file.

warning_content_llist

[list] The list of warning contents.

line_number_pre_fix_llist

[list] The list of line numbers of warnings before fixing.

Raises

Exception

If there is an error while executing the SQL commands.

set_pipeline_data()

Sets pipeline data by getting information from environment variables. If no pipeline ID is provided, generates a pseudo-random one.

update_last_header_file_id()

Updates the last_header_file_id member variable with the most recent header file ID from the database.

Raises

Exception

If there is an error while executing the SQL command.

update_last_project_id()

Updates the last_project_id member variable with the most recent project ID from the database.

Raises

Exception

If there is an error while executing the SQL command.

update_post_fix_warning_details(*pipeline_id*, *file_name_list*: list, *file_content_list*: list, *line_number_llist*: list, *warnings_content_llist*: list)

Update the database with the details of post-fix warnings.

This method updates the header file content and warning fixed status in the database. It loops through the given list of file names, updates each file's content, and then checks if there are any warnings. If there are, it sets the fixed status to 0 and updates this in the database.

Parameters

pipeline_id

[str] The ID of the current pipeline.

file_name_list

[list] A list of the names of the files to update.

file_content_list

[list] A list containing the content of the files.

line_number_llist

[list] A list of lists, each sublist contains the line numbers where warnings occurred.

warnings_content_llist

[list] A list of lists, each sublist contains the warning content from a file.

Raises

Exception

If there is an error while executing the SQL commands.

1.1.6.3 Module contents

1.1.7 main.doxy_warning_post_processor package

1.1.7.1 Submodules

1.1.7.2 main.doxy_warning_post_processor.doxy_warning_post_processor module

class main.doxy_warning_post_processor.doxy_warning_post_processor.**WarningPostProcessor**

Bases: DoxyWarningDetector

The WarningPostProcessor class handles detection and processing of warnings after fixes have been applied.

Methods

apply_fix():

Applies fixes to the header files and saves changes.

detect_warning():

Detects and updates warning details post-fix.

apply_fix()

Applies fixes to the header files and saves changes.

This method loads pre-warning details, loads post-fix file content, and writes the post-fix file content back to the header files.

detect_warning()

Detects and updates warning details post-fix.

This method loads pre-warning details, detects warnings after fixes, updates warning lists, and finally adds the updated warning details to the postprocessed_warnings_data.json file.

Returns

None

1.1.7.3 Module contents

1.1.8 main.doxygen_management package

1.1.8.1 Submodules

1.1.8.2 main.doxygen_management.doxy_warning_detector module

class main.doxygen_management.doxy_warning_detector.**DoxyWarningDetector**

Bases: DoxyfileConfigurator

A class used to detect warnings in Doxygen documentation generation.

...

Attributes

warnings_list

[list] a list to hold the detected warnings (initialized as an empty list)

Methods

check_project_directory():

Checks the project directory for the existence of a single subfolder.

get_all_header_files(project_folder_path: str):

Gets all the header files in the given project folder path.

run_doxygen_for_file(header_file_name: str, lines: list, input_line: str, project_folder_path: str, line_num: int):

Runs Doxygen for the given header file.

extract_warnings(result: subprocess.CompletedProcess):

Extracts warnings from the result of a Doxygen run.

run_doxygen():

Executes the complete process to run Doxygen for all header files and extract warnings.

check_project_directory()

Checks if there is only one subfolder in the project directory. If there are multiple subfolders, the program exits with an error message. Returns the name of the single subfolder if it exists.

Returns

str

the name of the single subfolder

extract_warnings(result)

Extracts warnings from the result of a Doxygen run. Returns the extracted warnings.

Parameters

result: subprocess.CompletedProcess

The result of the Doxygen run

Returns

list, list, int

Lists of warning contents, warning line numbers and total number of warnings.

get_all_header_files(*project_folder_path*)

Gets all the header files in the project directory. If there are no header files, the program exits with an error message. Returns the list of header files if they exist.

Parameters

project_folder_path: str

The path to the project directory

Returns

list

A list of all header files in the project directory.

run_doxygen()

Executes the complete process to run Doxygen for all header files and extract warnings. The process includes preparing the Doxyfile, running Doxygen, and extracting warnings. Returns a list of results containing the header file name, project folder path, file content, and warnings.

Returns

tuple

The result of running doxygen and extracting warnings. It includes the project folder path, list of header file names, list of file contents, list of warning counts, list of warning contents, and list of warning line numbers.

run_doxygen_for_file(*header_file_name, lines: list, input_line, project_folder_path, line_num*)

Prepares the Doxyfile with the given parameters and runs Doxygen for the header file. Returns the result of the Doxygen run.

Parameters

header_file_name: str

The name of the header file

lines: list

A list of lines in the Doxyfile

input_line: str

The line to add to the Doxyfile

project_folder_path: str

The path to the project directory

line_num: int

The line number in the Doxyfile to modify

Returns

subprocess.CompletedProcess

The result of the Doxygen run

1.1.8.3 main.doxygen_management.doxyfile_configurator module

class main.doxygen_management.doxyfile_configurator.DoxyfileConfigurator

Bases: FileHandler

A class used to configure a Doxyfile.

...

Methods

create_doxyfile(directory_path, cmd):

Tries to create a Doxyfile at the given directory path, using the provided command

add_spaces_to_string(config_parameters):

Adds specific spaces to configuration parameters

get_single_subfolder(directory):

Checks if there is only one subfolder in a given directory

configure_doxyfile(filename, config_parameters):

Configures the Doxyfile with the given parameters

add_spaces_to_string(*config_parameters*)

Method to adds specific spaces to configuration parameters.

Parameters

config_parameters

[list] A list of configuration parameters to which spaces are added

Returns

list

A list of configuration parameters with added spaces

configure_doxyfile(*filename*, *config_parameters*)

Configures the Doxyfile with the given parameters.

Parameters

filename

[str] The name of the file to be configured

config_parameters

[list] A list of configuration parameters

It modifies the lines in the Doxyfile according to the provided configuration parameters.

create_doxyfile(*directory_path*)

Method to create a Doxyfile at the given directory path, using the provided command.

Parameters

directory_path

[str] The path of the directory where the Doxyfile will be created

Raises

subprocess.CalledProcessError

If there is an error while executing the command

FileNotFoundError

If the directory path does not exist

get_header_files(*directory*)

Returns a list of .h and .hpp files in the given directory.

Parameters

directory

[str] The directory to search

Returns

list

A list of .h and .hpp file paths

get_single_subfolder(*directory*)

Checks if there is only one subfolder in a given directory.

Parameters

directory

[str] The directory to check for subfolders

Returns

bool

A boolean value indicating the presence of a single subfolder

str

The name of the subfolder (if any)

remove_doxyfile_if_exists(*directory_path*)

Method to remove a Doxyfile at the given directory path.

Parameters

directory_path

[str] The path of the directory where the Doxyfile will be removed

Raises

subprocess.CalledProcessError

If there is an error while executing the command

FileNotFoundError

If the directory path does not exist

1.1.8.4 main.doxygen_management.file_handler module

class main.doxygen_management.file_handler.FileHandler

Bases: object

A class used to handle file operations.

...

Methods

read_file(filename: str) -> list:

Reads a file and returns its contents as a list of lines.

write_file(filename: str, lines: list):

Writes a list of lines into a file.

has_single_line(list: list) -> bool:

Checks if a list has only a single line.

find_line_numbers(filename: str, searched_text: str) -> int:

Finds the line numbers of all the occurrences of a specific text in a file.

redirect_print_to_file(self, project_folder_path, header_file_name_list,
warning_num_list, warning_content_llist, warning_line_number_llist, file_content_list,
pre_or_post, output_file):

Redirects the output of the *print_project_details_without_color* function to a specified file.

find_line_numbers(filename: str, searched_text: str)

Finds the line numbers of all the occurrences of a specific text in a file.

Parameters

filename

[str] The name of the file to be searched.

searched_text

[str] The text to be found.

Returns

int

The line number of the first occurrence of the search text if there is only one occurrence, else it exits the program.

has_single_line(list: list)

Checks if a list has only a single line.

Parameters

list

[list] The list to check.

Returns

bool

True if the list has only a single line, else it exits the program.

read_file(filename: str)

Reads a file and returns its contents as a list of lines.

Parameters

filename

[str] The name of the file to be read.

Returns

list

A list of lines from the file.

redirect_print_to_file(*project_folder_path*, *header_file_name_list*: list, *warning_num_list*: list, *warning_content_llist*: list, *warning_line_number_llist*: list, *file_content_list*: list, *pre_or_post*, *output_file*)

Redirects the output of the *print_project_details_without_color* function to a specified file.

This function modifies the standard output (stdout) to a specified file, calls a function to print data without color, and then restores stdout to its original state.

Parameters

project_folder_path

[str] The path of the project folder.

header_file_name_list

[list] List of header file names.

warning_num_list

[list] List of warning numbers.

warning_content_llist

[list] List of warning content.

warning_line_number_llist

[list] List of warning line numbers.

file_content_list

[list] List of file content.

pre_or_post

[str] String indicating whether it's pre or post data.

output_file

[str] Path to the output file where print statements are to be redirected.

write_file(*filename*: str, *lines*)

Writes a list of lines into a file.

Parameters

filename

[str] The name of the file to write to.

lines

[list] A list containing the lines to be written to the file.

1.1.8.5 Module contents

1.1.9 main.error_handler package

1.1.9.1 Submodules

1.1.9.2 main.error_handler.error_handler module

class `main.error_handler.error_handler.ErrorHandler`

Bases: `object`

A class used to handle common errors that may occur during program execution

...

Static methods

handle_subprocess_error(*e*, *custom_message*):

Handles subprocess errors by displaying the error message in a custom format and terminating the program

handle_file_not_found_error(*e*, *custom_message*):

Handles file not found errors by displaying the error message in a custom format and terminating the program

`execute_sql`(*cursor*, *sql*, *values=None*):

static `execute_sql`(*cursor*, *sql*, *values=None*)

Executes a SQL command.

Parameters

cursor

[Database Cursor] The cursor object from a database connection which is used to execute the SQL query.

sql

[str] The SQL command to be executed.

values

[tuple, optional] The values to be inserted into the SQL command (default is None).

Raises

Exception

If there is an error while executing the SQL command.

static `handle_file_not_found_error`(*e*, *custom_message*)

Handles file not found errors by displaying the error message in a custom format and terminating the program

Parameters

e

[Exception] The exception raised

custom_message

[str] The custom message to be displayed before the exception message

static handle_subprocess_error(*e, custom_message*)

Handles subprocess errors by displaying the error message in a custom format and terminating the program

Parameters

e

[Exception] The exception raised

custom_message

[str] The custom message to be displayed before the exception message

1.1.9.3 Module contents

1.1.10 main.print_data package

1.1.10.1 Submodules

1.1.10.2 main.print_data.print_data module

```
main.print_data.print_data.print_project_details(project_folder_post_fix_path: list,
                                                header_file_name_post_fix_list: list,
                                                warning_num_post_fix_list: list,
                                                warning_content_post_fix_llist: list,
                                                warning_line_number_post_fix_llist: list,
                                                pre_or_post, file_content_post_fix_list=None)
```

Print details of the project, including file contents, Doxygen warnings, and lines where the warnings occurred.

Parameters

project_folder_post_fix_path

[str] The path to the project folder.

header_file_name_post_fix_list

[list] List of the names of the header files.

warning_num_post_fix_list

[list] List of the numbers of the Doxygen warnings detected.

warning_content_post_fix_llist

[list] List of the content of the Doxygen warnings detected.

warning_line_number_post_fix_llist

[list] List of the line numbers where the Doxygen warnings were detected.

pre_or_post

[str] Indicate if the details are before the fix ('pre') or after the fix ('post').

file_content_post_fix_list

[list, optional] List of the contents of the files, by default None.

Returns

None

```
main.print_data.print_data.print_project_details_without_color(project_folder_post_fix_path,  
                                                                header_file_name_post_fix_list,  
                                                                warning_num_post_fix_list,  
                                                                warning_content_post_fix_llist,  
                                                                warn-  
                                                                ing_line_number_post_fix_llist,  
                                                                pre_or_post,  
                                                                file_content_post_fix_list=None)
```

Print details of the project, including file contents, Doxygen warnings, and lines where the warnings occurred.

Parameters**project_folder_post_fix_path**

[str] The path to the project folder.

header_file_name_post_fix_list

[list] List of the names of the header files.

warning_num_post_fix_list

[list] List of the numbers of the Doxygen warnings detected.

warning_content_post_fix_llist

[list] List of the content of the Doxygen warnings detected.

warning_line_number_post_fix_llist

[list] List of the line numbers where the Doxygen warnings were detected.

pre_or_post

[str] Indicate if the details are before the fix ('pre') or after the fix ('post').

file_content_post_fix_list

[list, optional] List of the contents of the files, by default None.

Returns

None

1.1.10.3 Module contents

1.1.11 main.transport_data package

1.1.11.1 Submodules

1.1.11.2 main.transport_data.transport_data module

`main.transport_data.transport_data.add_data_to_json(file_path, new_data)`

Add new data to an existing JSON file.

Parameters

file_path

[str] The path to the JSON file.

new_data

[dict] A dictionary containing the new data to be added to the JSON file.

`main.transport_data.transport_data.load_data_from_json(file_path)`

Load data from a JSON file and return it.

Parameters

file_path

[str] The path to the JSON file.

Returns

dict

A dictionary containing the data loaded from the JSON file.

`main.transport_data.transport_data.load_post_fix_data_from_json()`

Load postprocessed warning data from a JSON file.

The file contains details of the project folder path, header file names, file contents, warning numbers, warning contents, and line numbers of the warnings.

Returns

tuple

A tuple containing all the loaded data from the JSON file.

`main.transport_data.transport_data.load_pre_fix_data_from_json()`

Load preprocessed warning data from a JSON file.

The file contains details of the project folder path, header file names, file contents, warning numbers, warning contents, and line numbers of the warnings.

Returns

tuple

A tuple containing all the loaded data from the JSON file.

`main.transport_data.transport_data.save_data_to_json(file_path, data_dict)`

Save data to a JSON file.

Parameters

file_path

[str] The path to the JSON file.

data_dict

[dict] A dictionary where the key is the name of the data and the value is the list or string to be saved.

1.1.11.3 Module contents

1.1.12 main.utils package

1.1.12.1 Submodules

1.1.12.2 main.utils.runtime_measurement module

`main.utils.runtime_measurement.measure_runtimes_for_different_lines(text_template: str)`

Measure the runtime of the three functions for texts with varying number of lines.

Parameters: - text_template (str): The template text to be repeated.

Returns: - dict: A dictionary with line counts as keys and runtimes as values.

`main.utils.runtime_measurement.remove_unwanted_line(str)`

Removes unwanted lines from the model's response.

`main.utils.runtime_measurement.remove_unwanted_lines_more_efficient(text: str) → str`

`main.utils.runtime_measurement.remove_unwanted_lines_with_dict(text: str) → str`

```
main.utils.runtime_measurement.runtimes = {10: {'dict_method': 1.8199003534391522e-05,
'regex_method': 4.759000148624182e-06, 'remove_unwanted_line': 1.2368996976874769e-05},
50: {'dict_method': 5.5406009778380394e-05, 'regex_method': 7.30800093151629e-06,
'remove_unwanted_line': 2.2688996978104115e-05}, 100: {'dict_method':
0.00011409299622755498, 'regex_method': 1.3611992471851408e-05, 'remove_unwanted_line':
5.807899287901819e-05}, 500: {'dict_method': 0.0005285770021146163, 'regex_method':
6.570399273186922e-05, 'remove_unwanted_line': 0.00018376500520389527}, 1000:
{'dict_method': 0.001031874999171123, 'regex_method': 9.082800534088165e-05,
'remove_unwanted_line': 0.0003573260037228465}, 5000: {'dict_method':
0.005800165003165603, 'regex_method': 0.0004864369984716177, 'remove_unwanted_line':
0.0022031480039004236}, 10000: {'dict_method': 0.011811809992650524, 'regex_method':
0.0010164989944314584, 'remove_unwanted_line': 0.003925994999008253}, 50000:
{'dict_method': 0.05639708900707774, 'regex_method': 0.005216495002969168,
'remove_unwanted_line': 0.021319448002032004}, 100000: {'dict_method':
0.12259472699952312, 'regex_method': 0.010382110995124094, 'remove_unwanted_line':
0.040552343998569995}, 1000000: {'dict_method': 1.1640679920092225, 'regex_method':
0.10644538400811143, 'remove_unwanted_line': 0.4248778629989829}, 10000000:
{'dict_method': 11.906525145997875, 'regex_method': 1.0933152100042207,
'remove_unwanted_line': 4.281598171000951}}
```

For 10 lines: remove_unwanted_line took 0.00001 seconds. regex_method took 0.00000 seconds. dict_method took 0.00002 seconds. _____ For 50 lines: remove_unwanted_line took 0.00003 seconds. regex_method took 0.00001 seconds. dict_method took 0.00006 seconds. _____ For 100 lines: remove_unwanted_line took 0.00004 seconds. regex_method took 0.00001 seconds. dict_method took 0.00012 seconds. _____ For 500 lines: remove_unwanted_line took 0.00021 seconds. regex_method took 0.00006 seconds. dict_method took 0.00063 seconds. _____ For 1000 lines: remove_unwanted_line took 0.00039 seconds. regex_method took 0.00010 seconds. dict_method took 0.00124 seconds. _____ For 5000 lines: remove_unwanted_line took 0.00193 seconds. regex_method took 0.00050 seconds. dict_method took 0.00588 seconds. _____ For 10000 lines: remove_unwanted_line took 0.00380 seconds. regex_method took 0.00108 seconds. dict_method took 0.01218 seconds. _____ For 50000 lines: remove_unwanted_line took 0.01958 seconds. regex_method took 0.00561 seconds. dict_method took 0.06098 seconds. _____ For 100000 lines: remove_unwanted_line took 0.04078 seconds. regex_method took 0.01022 seconds. dict_method took 0.12191 seconds. _____ For 1000000 lines: remove_unwanted_line took 0.40735 seconds. regex_method took 0.10590 seconds. dict_method took 1.25409 seconds. _____ For 10000000 lines: remove_unwanted_line took 4.08598 seconds. regex_method took 1.07782 seconds. dict_method took 12.47559 seconds. _____

1.1.12.3 main.utils.utils module

```
main.utils.utils.compaine_warn_details(warning_content_llist, warning_line_number_llist,
                                         file_content_list, header_file_name_list)
```

Compile warning details into a list of prompts.

Parameters

warning_content_llist

[list] List of warning content.

warning_line_number_llist

[list] List of warning line numbers.

file_content_list

[list] List of file content.

header_file_name_list

[list] List of header file names.

Returns

list

List of prompts created from warnings and file content.

`main.utils.utils.generate_html_diff()`

Generate an HTML file from a diff text file.

`main.utils.utils.perform_diff_on_files()`

Compare two files and output the difference to a text file.

`main.utils.utils.rename_and_move(source_path: str, target_dir: str) → None`

Move and rename a specific file from source to target directory.

This function takes a file named *diff.html* from the specified source path and moves it to the target directory. In the target directory, the file is renamed following the pattern *test_diff_{n}.html*, where *{n}* is determined by existing files in the target directory.

Parameters

source_path

[str] The path to the directory containing the *diff.html* file.

target_dir

[str] The directory where the file will be moved and renamed.

Raises

FileNotFoundError

If the *diff.html* file is not found in the source path.

Examples

```
>>> rename_and_move('/path/to/source', '/path/to/target')
```

1.1.12.4 Module contents

1.2 Submodules

1.3 main.main module

Main execution script for the Doxygen Warning Manager tool.

The AI Doxygen Cleaner is a tool that automates several tasks related to Doxygen warning management. This includes configuring Doxygen, detecting warnings, inserting warnings into a database, fixing warnings using an AI language model, applying these fixes, detecting warnings after fixes have been applied, printing warning details to a file, generating a diff page in HTML format, and updating warning data in the database post-fix.

1.4 Module contents

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

m

- main, 26
- main.ai_doxygen_cleaner, 4
- main.ai_doxygen_cleaner.ai_doxygen_cleaner, 2
- main.ai_language_model, 5
- main.ai_language_model.ai_language_model, 4
- main.config, 6
- main.config.config_paths, 5
- main.config.config_variables, 5
- main.data, 6
- main.db_operations, 11
- main.db_operations.db_insertion_handler, 6
- main.doxy_warning_post_processor, 11
- main.doxy_warning_post_processor.doxy_warning_post_processor,
11
- main.doxygen_management, 19
- main.doxygen_management.doxy_warning_detector,
11
- main.doxygen_management.doxyfile_configurator,
14
- main.doxygen_management.file_handler, 16
- main.DoxygenWarningFixer, 2
- main.DoxygenWarningFixer.doxygen_warning_fixer,
1
- main.error_handler, 20
- main.error_handler.error_handler, 19
- main.main, 26
- main.print_data, 22
- main.print_data.print_data, 20
- main.transport_data, 23
- main.transport_data.transport_data, 22
- main.utils, 26
- main.utils.runtime_measurement, 23
- main.utils.utils, 24

INDEX

A

`add_data_to_json()` (in module `main.transport_data.transport_data`), 22

`add_spaces_to_string()` (in module `main.doxygen_management.doxyfile_configurator.DoxyfileConfigurator`), 14

`AIDoxygenCleaner` (class in `main.ai_doxygen_cleaner.ai_doxygen_cleaner`), 2

`AILanguageModel` (class in `main.ai_language_model.ai_language_model`), 4

`apply_fix()` (in module `main.doxy_warning_post_processor.doxy_warning_post_processor.WarningPostProcessor`), 11

C

`check_project_directory()` (in module `main.doxygen_management.doxy_warning_detector.DoxyWarningDetector`), 12

`compaine_warn_details()` (in module `main.utils.utils`), 24

`configure_doxyfile()` (in module `main.doxygen_management.doxyfile_configurator.DoxyfileConfigurator`), 14

`connect_to_database()` (in module `main.db_operations.db_insertion_handler.DBInsertionHandler`), 7

`create_cursor()` (in module `main.db_operations.db_insertion_handler.DBInsertionHandler`), 7

`create_doxyfile()` (in module `main.doxygen_management.doxyfile_configurator.DoxyfileConfigurator`), 15

`create_prompt_content()` (in module `main.ai_language_model.ai_language_model.AILanguageModel`), 5

D

`DBInsertionHandler` (class in `main.db_operations.db_insertion_handler`), 6

`detect_warning()` (in module `main.doxy_warning_post_processor.doxy_warning_post_processor.WarningPostProcessor`), 11

`detect_warnings_post_fix()` (in module `main.ai_doxygen_cleaner.ai_doxygen_cleaner.AIDoxygenCleaner`), 3

`detect_warnings_pre_fix()` (in module `main.ai_doxygen_cleaner.ai_doxygen_cleaner.AIDoxygenCleaner`), 3

`DoxyfileConfigurator` (class in `main.doxygen_management.doxyfile_configurator`), 14

`doxygen_configuration()` (in module `main.ai_doxygen_cleaner.ai_doxygen_cleaner.AIDoxygenCleaner`), 3

`DoxygenWarningFixer` (class in `main.DoxygenWarningFixer.doxygen_warning_fixer`), 1

`DoxyWarningDetector` (class in `main.doxygen_management.doxy_warning_detector`), 11

E

`ErrorHandler` (class in `main.error_handler.error_handler`), 19

`execute_sql()` (in module `main.error_handler.error_handler.ErrorHandler`), 19

`extract_warnings()` (in module `main.doxygen_management.doxy_warning_detector.DoxyWarningDetector`), 12

F

`FileHandler` (class in `main.doxygen_management.file_handler`), 16

`find_line_numbers()` (in module `main.doxygen_management.file_handler.FileHandler`), 17

`fix_doxygen_warning()` (in module `main.ai_doxygen_cleaner.ai_doxygen_cleaner.AIDoxygenCleaner`), 3

G

`generate_html_diff()` (in module `main.utils.utils`), 25

`generate_html_diff_page()` (in module `main.doxy_warning_post_processor.doxy_warning_post_processor.WarningPostProcessor`), 3

get_all_header_files()	main.ai_doxygen_cleaner
(main.doxygen_management.doxy_warning_detector.DoxyWarningDetector	module, 13
method), 13	main.ai_doxygen_cleaner.ai_doxygen_cleaner
get_header_files()	(main.doxygen_management.doxyfile_configurator.DoxyfileConfigurator
method), 15	module, 2
get_num_tokens_from_response()	main.ai_language_model
(main.ai_language_model.ai_language_model.AILanguageModel	module, 5
method), 5	main.ai_language_model.ai_language_model
get_single_subfolder()	module, 4
(main.doxygen_management.doxyfile_configurator.DoxyfileConfigurator	main.config
method), 15	module, 5
	main.config.config_paths
	module, 5
H	main.config.config_variables
handle_file_not_found_error()	module, 5
(main.error_handler.error_handler.ErrorHandler	main.data
static method), 19	module, 6
handle_subprocess_error()	main.db_operations
(main.error_handler.error_handler.ErrorHandler	module, 11
static method), 20	main.db_operations.db_insertion_handler
has_single_line()	(main.doxygen_management.file_handler.FileHandler
method), 17	module, 16
	main.doxy_warning_post_processor
	module, 11
I	main.doxy_warning_post_processor.doxy_warning_post_processor
	module, 11
insert_into_header_file_table()	main.doxygen_management
(main.db_operations.db_insertion_handler.DBInsertionHandler	module, 19
method), 8	main.doxygen_management.doxy_warning_detector
insert_into_pipeline_table()	module, 11
(main.db_operations.db_insertion_handler.DBInsertionHandler	main.doxygen_management.doxyfile_configurator
method), 8	module, 14
insert_into_project_table()	main.doxygen_management.file_handler
(main.db_operations.db_insertion_handler.DBInsertionHandler	module, 16
method), 8	main.DoxygenWarningFixer
insert_into_warning_table_pre_fix()	module, 2
(main.db_operations.db_insertion_handler.DBInsertionHandler	main.DoxygenWarningFixer.doxygen_warning_fixer
method), 9	module, 1
insert_warnings_details()	main.error_handler
(main.db_operations.db_insertion_handler.DBInsertionHandler	module, 20
method), 9	main.error_handler.error_handler
insert_warnings_to_db()	module, 19
(main.ai_doxygen_cleaner.ai_doxygen_cleaner.AI	main.main
method), 3	module, 26
L	main.print_data
	module, 22
load_data_from_json()	(in module
main.transport_data.transport_data), 22	main.print_data.print_data
load_post_fix_data_from_json()	(in module
main.transport_data.transport_data), 22	module, 20
load_pre_fix_data_from_json()	(in module
main.transport_data.transport_data), 22	main.transport_data
	module, 23
	main.transport_data.transport_data
	module, 22
M	main.utils
	module, 26
main	main.utils.runtime_measurement
module, 26	module, 23

main.utils.utils
 module, 24

measure_runtimes_for_different_lines() (in
 module main.utils.runtime_measurement), 23

module

- main, 26
- main.ai_doxygen_cleaner, 4
- main.ai_doxygen_cleaner.ai_doxygen_cleaner,
 2
- main.ai_language_model, 5
- main.ai_language_model.ai_language_model,
 4
- main.config, 6
- main.config.config_paths, 5
- main.config.config_variables, 5
- main.data, 6
- main.db_operations, 11
- main.db_operations.db_insertion_handler,
 6
- main.doxy_warning_post_processor, 11
- main.doxy_warning_post_processor.doxy_warning_post_processor,
 11
- main.doxygen_management, 19
- main.doxygen_management.doxy_warning_detector,
 11
- main.doxygen_management.doxyfile_configurator,
 14
- main.doxygen_management.file_handler, 16
- main.DoxygenWarningFixer, 2
- main.DoxygenWarningFixer.doxygen_warning_fixer,
 1
- main.error_handler, 20
- main.error_handler.error_handler, 19
- main.main, 26
- main.print_data, 22
- main.print_data.print_data, 20
- main.transport_data, 23
- main.transport_data.transport_data, 22
- main.utils, 26
- main.utils.runtime_measurement, 23
- main.utils.utils, 24

P

perform_diff_on_files() (in module
 main.utils.utils), 25

print_pre_post_fix_data_to_file()
 (main.ai_doxygen_cleaner.ai_doxygen_cleaner.AIDoxygenCleaner.
 method), 3

print_project_details() (in module
 main.print_data.print_data), 20

print_project_details_without_color() (in mod-
 ule main.print_data.print_data), 21

print_query() (main.DoxygenWarningFixer.doxygen_warning_fixer.
 method), 1

R

read_file() (main.doxygen_management.file_handler.FileHandler
 method), 17

redirect_print_to_file()
 (main.doxygen_management.file_handler.FileHandler
 method), 18

remove_doxyfile_if_exists()
 (main.doxygen_management.doxyfile_configurator.DoxyfileConfig
 method), 16

remove_unwanted_line() (in module
 main.utils.runtime_measurement), 23

remove_unwanted_lines_more_efficient() (in
 module main.utils.runtime_measurement), 23

remove_unwanted_lines_with_dict() (in module
 main.utils.runtime_measurement), 23

rename_and_move() (in module main.utils.utils), 25

run() (main.DoxygenWarningFixer.doxygen_warning_fixer.DoxygenWarnin
 method), 2

run_doxygen() (main.doxygen_management.doxy_warning_detector.Doxy
 method), 13

run_doxygen_for_file()
 (main.doxygen_management.doxy_warning_detector.DoxyWarnin
 method), 13

runtimes (in module main.utils.runtime_measurement),
 23

S

save_data_to_json() (in module
 main.transport_data.transport_data), 23

set_pipeline_data()
 (main.db_operations.db_insertion_handler.DBInsertionHandler
 method), 10

set_response_content()
 (main.ai_language_model.ai_language_model.AILanguageModel
 method), 5

U

update_last_header_file_id()
 (main.db_operations.db_insertion_handler.DBInsertionHandler
 method), 10

update_last_project_id()
 (main.db_operations.db_insertion_handler.DBInsertionHandler
 method), 10

update_post_fix_warning_details()
 (main.db_operations.db_insertion_handler.DBInsertionHandler
 method), 10

update_warnings_details_post_fix_in_db()
 (main.ai_doxygen_cleaner.ai_doxygen_cleaner.AIDoxygenCleaner.
 method), 3

V

validate_response_content()
 (main.ai_language_model.ai_language_model.AILanguageModel
 method), 5

W

`wait()` (*main.DoxygenWarningFixer.doxygen_warning_fixer.DoxygenWarningFixer*
method), [2](#)

`WarningPostProcessor` (*class* *in*
main.doxy_warning_post_processor.doxy_warning_post_processor),
[11](#)

`write_file()` (*main.doxygen_management.file_handler.FileHandler*
method), [18](#)