```
{
    "openapi": "3.0.0",
    "info": {
        "title": "API",
        "version": "1.0.0"
    },
    "paths": {
        "/api/aichat/history/{cat_slug}": {
            "get": {
                "tags": [
                    "AI Chats (Convesations)"
                ],
                "summary": "Get chatbot conversations",
                "description": "get chatbot conversations by template category slug.",
                "operationId": "getOpenAIChat",
                "parameters": [
                    {
                        "name": "cat_slug",
                        "in": "path",
                        "description": "cat_slug of the OpenAI chat category",
                        "required": true,
                        "schema": {
                            "type": "string",
                            "example": "ai-chat-bot"
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "category": {
                                            "type": "object"
                                        },
                                        "conversations": {
                                            "type": "array",
                                            "items": {
                                                "type": "object"
                                            }
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "404": {
                        "description": "Category not found"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aichat/recent-chats": {
            "get": {
                "tags": [
                    "AI Chats (Convesations)"
                ],
                "summary": "Get recent chatbot conversations",
                "description": "get chatbot conversations by last 20 recent documents.",
                "operationId": "recentChats",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "conversations": {
                                            "type": "array",
                                            "items": {
                                                "type": "object"
                                            }
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aichat/search-recent-chats": {
            "post": {
                "tags": [
                    "AI Chats (Convesations)"
                ],
                "summary": "Search all chatbot conversations and return last 20",
                "description": "Search all chatbot conversations and return last 20 recent.",
                "operationId": "searchRecentChats",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "conversations": {
                                            "type": "array",
                                            "items": {
                                                "type": "object"
                                            }
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aichat/new-chat": {
            "post": {
                "tags": [
                    "AI Chats (Convesations)"
                ],
                "summary": "Create new chat conversation",
                "description": "Create new chat conversation",
                "operationId": "startNewChat",
                "requestBody": {
                    "description": "Request chat template data",
                    "required": true,
                    "content": {
                        "multipart/form-data": {
                            "schema": {
                                "properties": {
                                    "category_id": {
                                        "description": "Category (Template) ID",
                                        "type": "string"
                                    }
                                },
                                "type": "object"
                            }
                        }
                    }
                },
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    },
                    "412": {
                        "description": "Precondition Failed"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aichat/history": {
            "delete": {
                "tags": [
                    "AI Chats (Convesations)"
                ],
                "summary": "Delete a conversation",
                "description": "Delete a conversation and all its messages",
                "operationId": "deleteChat",
                "requestBody": {
                    "description": "Conversation ID to delete",
                    "required": true,
                    "content": {
                        "application/json": {
                            "schema": {
                                "properties": {
                                    "conver_id": {
                                        "type": "string",
                                    "example": "109"
                                }
                            },
                            "type": "object"
                        }
                    }
                }
            },
            "responses": {
                "204": {
                    "description": "Successful operation"
                },
                "401": {
                    "description": "Unauthenticated"
                },
                "403": {
                    "description": "Forbidden"
                },
                "404": {
                    "description": "Chat not found"
                },
                "412": {
                    "description": "Precondition Failed"
                }
            },
            "security": [
                {
                    "passport": []
                }
            ]
        },
        "patch": {
            "tags": [
                "AI Chats (Convesations)"
            ],
            "summary": "Rename a conversation",
            "description": "Rename a user's OpenAI conversation.",
            "operationId": "renameChat",
            "requestBody": {
                "description": "Chat ID and new title",
                "required": true,
                "content": {
                    "application/json": {
                        "schema": {
                            "properties": {
                                "conver_id": {
                                    "type": "string",
                                    "example": "109"
                                },
                                "title": {
                                    "type": "string",
                                    "example": "New Chat Title"
                                }
                            },
                            "type": "object"
                        }
                    }
                }
            },
            "responses": {
                "200": {
                    "description": "Successful operation"
                },
                "401": {
                    "description": "Unauthenticated"
                },
                    "403": {
                        "description": "Forbidden"
                    },
                    "404": {
                        "description": "Chat not found"
                    },
                    "412": {
                        "description": "Precondition Failed"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aichat/search-history": {
            "post": {
                "tags": [
                    "AI Chats (Convesations)"
                ],
                "summary": "Search for chats in template category",
                "description": "Search for user's OpenAI chats based on the provided category and search word.",
                "operationId": "searchChatHistory",
                "requestBody": {
                    "description": "Category ID and search word",
                    "required": true,
                    "content": {
                        "application/json": {
                            "schema": {
                                "properties": {
                                    "category_id": {
                                        "type": "integer",
                                        "example": 1
                                    },
                                    "search_word": {
                                        "type": "string",
                                        "example": "keyword"
                                    }
                                },
                                "type": "object"
                            }
                        }
                    }
                },
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "html": {
                                            "type": "string"
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aichat/chat/chat-send": {
            "get": {
                "tags": [
                    "AI Chats (Convesations)"
                ],
                "summary": "Retrieve chat details",
                "operationId": "chatOutputGet",
                "parameters": [
                    {
                        "name": "conver_id",
                        "in": "query",
                        "description": "ID of the chat",
                        "required": true,
                        "schema": {
                            "type": "integer"
                        }
                    },
                    {
                        "name": "message_id",
                        "in": "query",
                        "description": "ID of the message",
                        "required": true,
                        "schema": {
                            "type": "integer"
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Successful response",
                        "content": {
                            "text/event-stream": {
                                "schema": {
                                    "type": "string",
                                    "example": "data: This is a streamed response\\n\\n"
                                }
                            }
                        }
                    },
                    "404": {
                        "description": "Chat or message not found",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "error": {
                                            "type": "string",
                                            "example": "Chat or message not found"
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            },
            "post": {
                "tags": [
                    "AI Chats (Convesations)"
                ],
                "summary": "Process chat output",
                "operationId": "chatOutputPost",
                "requestBody": {
                    "description": "Request body for chat output",
                    "required": true,
                    "content": {
                        "application/json": {
                            "schema": {
                                "properties": {
                                    "conver_id": {
                                        "type": "integer",
                                        "example": "31"
                                    },
                                    "prompt": {
                                        "type": "string",
                                        "example": "Your chat prompt goes here"
                                    },
                                    "category_id": {
                                        "type": "integer",
                                        "example": "1"
                                    }
                                },
                                "type": "object"
                            }
                        }
                    }
                },
                "responses": {
                    "200": {
                        "description": "Successful response",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "conver_id": {
                                            "type": "integer",
                                            "example": "1"
                                        },
                                        "message_id": {
                                            "type": "integer",
                                            "example": "123"
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "419": {
                        "description": "No credits left",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "errors": {
                                            "type": "array",
                                            "items": {
                                                "type": "string"
                                            },
                                            "example": [
                                                "You have no credits left. Please consider
```

```
upgrading your plan."
                                            ]
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "500": {
                        "description": "Error from API call",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "error": {
                                            "type": "string",
                                            "example": "Error from API call. Please try
again. If error persists again, please contact the system administrator."
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aichat/chat/{conver_id}": {
            "get": {
                "tags": [
                    "AI Chats (Messages)"
                ],
                "summary": "Get OpenAI conversation details (relate category, related
conversation, related chat_completions)",
                "description": "Get OpenAI conversation by conver_id.",
                "operationId": "conversations",
                "parameters": [
                    {
                        "name": "conver_id",
                        "in": "path",
                        "description": "conver_id of the OpenAI chat conversations",
                        "required": true,
                        "schema": {
                            "type": "string",
                            "example": "31"
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Successful operation"
                    },
                    "404": {
                        "description": "Category not found"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aichat/chat/{conver_id}/messages/{id}": {
            "get": {
                "tags": [
                    "AI Chats (Messages)"
                ],
                "summary": "Get OpenAI chat details (relate category, related conversation,
related messages, related chat_completions).",
                "description": "Get OpenAI chat details (relate category, related
conversation, related messages, related chat_completions). ID parameter is optional. To get
all messages end path with /messages",
                "operationId": "conversationChats",
                "parameters": [
                    {
                        "name": "conver_id",
                        "in": "path",
                        "description": "conver_id of the OpenAI chat conversations",
                        "required": true,
                        "schema": {
                            "type": "string",
                            "example": "31"
                        }
                    },
                    {
                        "name": "id",
                        "in": "path",
                        "description": "id of the OpenAI chat message. Use to get one message
only",
                        "required": false,
                        "schema": {
                            "type": "string",
                            "example": "0"
                        }
                    },
                    {
                        "name": "page",
                        "in": "query",
                        "description": "Page number",
                        "schema": {
                            "type": "integer",
                            "default": 1
                        }
                    },
                    {
                        "name": "per_page",
                        "in": "query",
                        "description": "Items per page",
                        "schema": {
                            "type": "integer",
                            "default": 10
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Successful operation"
                    },
                    "404": {
                        "description": "Category not found"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/general/recent-documents": {
            "get": {
                "tags": [
                    "General (Helpers)"
                ],
                "summary": "Get the last 6 recent documents",
                "description": "Get the last 6 recent documents for the authenticated user,
excluding documents of type 'image'.",
                "operationId": "getRecentDocuments",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "data": {
                                            "type": "array",
                                            "items": {
                                                "type": "object"
                                            }
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthorized: User not authenticated"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/general/favorite-openai": {
            "get": {
                "tags": [
                    "General (Helpers)"
                ],
                "summary": "Get last 6 favorite OpenAI items",
                "description": "Get the last 6 favorite OpenAI items for the authenticated
user.",
                "operationId": "openAIFavoritesList",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "array",
                                    "items": {
                                        "properties": {
                                            "id": {
                                                "type": "integer",
                                                "example": 1
                                            },
                                            "title": {
                                                "type": "string",
                                                "example": "Post Title Generator"
                                            },
                                            "description": {
                                                "type": "string",
                                                "example": "Get captivating post titles
instantly with our title generator. Boost engagement and save time."
                                            },
                                            "slug": {
                                                "type": "string",
                                                "example": "post_title_generator"
                                            },
                                            "active": {
                                                "type": "integer",
                                                "example": 1
                                            },
                                            "questions": {
                                                "type": "string",
                                                "example": ""
                                            },
                                            "image": {
                                                "type": "string",
                                                "example": ""
                                            },
                                            "premium": {
                                                "type": "integer",
                                                "example": 0
                                            },
                                            "type": {
                                                "type": "string",
                                                "example": "text"
                                            },
                                            "created_at": {
                                                "type": "string",
                                                "example": "2023-03-11T08:26:49.000000Z"
                                            },
                                            "updated_at": {
                                                "type": "string",
                                                "example": "2023-03-11T08:26:49.000000Z"
                                            },
                                            "prompt": {
                                                "type": "string",
                                                "nullable": true
                                            },
                                            "custom_template": {
                                                "type": "integer",
                                                "example": 0
                                            },
                                            "tone_of_voice": {
                                                "type": "integer",
                                                "example": 0
                                            },
                                            "color": {
                                                "type": "string",
                                                "example": "#A3D6C2"
                                            },
                                            "filters": {
                                                "type": "string",
                                                "example": "blog"
                                            },
                                            "package": {
                                                "type": "string",
                                                "nullable": true
                                            },
                                            "pivot": {
                                                "properties": {
                                                    "user_id": {
                                                        "type": "integer",
                                                        "example": 1
                                                    },
                                                    "openai_id": {
                                                        "type": "integer",
                                                        "example": 1
                                                    }
                                                },
                                                "type": "object"
                                            }
                                        },
                                        "type": "object"
                                    }
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthorized: User not authenticated"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/general/search": {
            "post": {
                "tags": [
                    "General (Helpers)"
                ],
                "summary": "Search for templates, workbooks, and AI chats",
                "description": "Search for templates, workbooks, and AI chats based on the
provided keyword.",
                "operationId": "search",
                "requestBody": {
                    "description": "Keyword for the search",
                    "required": true,
                    "content": {
                        "application/json": {
                            "schema": {
                                "properties": {
                                    "search": {
                                        "type": "string",
                                        "example": "keyword"
                                    }
                                },
                                "type": "object"
                            }
                        }
                    }
                },
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "template_search": {
                                            "type": "array",
                                            "items": {
                                                "type": "object"
                                            }
                                        },
                                        "workbook_search": {
                                            "type": "array",
                                            "items": {
                                                "type": "object"
                                            }
                                        },
                                        "ai_chat_search": {
                                            "type": "array",
                                            "items": {
                                                "type": "object"
                                            }
                                        },
                                        "result": {
                                            "type": "string"
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aiwriter/generate": {
            "post": {
                "tags": [
                    "AI Writer"
                ],
                "summary": "Build AI-generated content please see the function there is
custom inputs foreach template",
                "description": "Build AI-generated content based on user input.",
                "operationId": "buildOutputApi",
                "requestBody": {
                    "required": true,
                    "content": {
                        "multipart/form-data": {
                            "schema": {
                                "properties": {
                                    "post_type": {
                                        "type": "string",
                                        "enum": [
                                            "post_title_generator",
                                            "article_generator",
                                            "summarize_text",
                                            "product_description",
                                            "product_name",
                                            "testimonial_review",
                                            "problem_agitate_solution",
                                            "blog_section",
                                            "blog_post_ideas",
                                            "blog_intros",
                                            "blog_conclusion",
                                            "facebook_ads",
                                            "youtube_video_description",
                                            "youtube_video_title",
                                            "youtube_video_tag",
                                            "instagram_captions",
                                            "instagram_hashtag",
                                            "social_media_post_tweet",
                                            "social_media_post_business",
                                            "facebook_headlines",
                                            "google_ads_headlines",
                                            "google_ads_description",
                                            "content_rewrite",
                                            "paragraph_generator",
                                            "pros_cons",
                                            "meta_description",
                                            "faq_generator",
                                            "email_generator",
                                            "email_answer_generator",
                                            "newsletter_generator",
                                            "grammar_correction",
                                            "tldr_summarization",
                                            "ai_image_generator",
                                            "ai_code_generator"
                                        ],
                                        "example": "summarize_text"
                                    },
                                    "maximum_length": {
                                        "description": "Maximum length of the generated
text",

                                        "type": "integer",
                                        "example": 200
                                    },
                                    "number_of_results": {
                                        "description": "Number of summary results to
generate",

                                        "type": "integer",
                                        "example": 1
                                    },
                                    "creativity": {
                                        "description": "Creativity level for the generated
content (0 to 1)",

                                        "type": "number",
                                        "example": 0.75
                                    },
                                    "tone_of_voice": {
                                        "description": "Tone of voice for the generated
content",

                                        "type": "string",
                                        "example": "Professional"
                                    },
                                    "language": {
                                        "description": "Language code for the input text",
                                        "type": "string",
                                        "example": "en-US"
                                    },
                                    "text_to_summary": {
                                        "description": "Text to be summarized",
                                        "type": "string",
                                        "example": "dfg"
                                    }
                                },
                                "type": "object"
                            }
                        }
                    }
                },
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "message_id": {
                                            "description": "ID of the generated content
message",
                                            "type": "integer"
                                        },
                                        "workbook": {
                                            "description": "Details about the generated
content workbook",
                                            "type": "object"
                                        },
                                        "creativity": {
                                            "description": "Creativity factor (between 0 and
1)",
                                            "type": "number"
                                        },
                                        "maximum_length": {
                                            "description": "Maximum length of the generated
content",
                                            "type": "integer"
                                        },
                                        "number_of_results": {
                                            "description": "Number of desired results",
                                            "type": "integer"
                                        },
                                        "inputPrompt": {
                                            "description": "User input prompt",
                                            "type": "string"
                                        },
                                        "generated_content": {
                                            "description": "Generated content",
                                            "type": "string"
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "400": {
                        "description": "Bad request",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "errors": {
                                            "description": "Error message",
                                            "type": "string"
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    },
                    "500": {
                        "description": "Server error",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "error": {
                                            "description": "Internal Server Error",
                                            "type": "string"
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aiimage/versions": {
            "get": {
                "tags": [
                    "AI Image Generation"
                ],
                "summary": "Get Model Versions for AI Image Generation (DALL-E, Stable
Diffusion) from settings",
                "description": "Get Model Versions for AI Image Generation",
                "operationId": "versions",
                "responses": {
                    "200": {
                        "description": "Successful operation"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aiimage/check-availability": {
            "get": {
                "tags": [
                    "AI Image Generation"
                ],
                "summary": "Check if image generation is active / available",
                "description": "Check if image generation is active",
                "operationId": "checkActiveGeneration",
                "responses": {
                    "200": {
                        "description": "Image generation is available."
                    },
                    "409": {
                        "description": "Image generation in progress. Please try again
later."
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aiimage/generate-image": {
            "post": {
                "tags": [
                    "AI Image Generation"
                ],
                "summary": "Generate Image (DALL-E / Stable Diffusion parameters required in
request)",
                "description": "Generate Image",
                "operationId": "generateImage",
                "responses": {
                    "200": {
                        "description": "Image generation successful. Image info in json
response [images]."
                    },
                    "409": {
                        "description": "Image generation in progress. Please try again
later."
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aiwriter/generator/{slug}": {
            "get": {
                "tags": [
                    "AI Writer"
                ],
                "summary": "returns the openai writer info and related user generated docs",
                "description": "Retrieve details of an OpenAI generator by slug",
                "operationId": "openAIGeneratorApi",
                "parameters": [
                    {
                        "name": "slug",
                        "in": "path",
                        "description": "Slug of the OpenAI generator",
                        "required": true,
                        "schema": {
                            "type": "string"
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "openai": {
                                            "type": "object"
                                        },
                                        "userOpenai": {
                                            "type": "object"
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "404": {
                        "description": "OpenAI generator not found",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "error": {
                                            "type": "string",
                                            "example": "Resource not found"
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aiwriter/generator/{slug}/workbook": {
            "get": {
                "tags": [
                    "AI Writer"
                ],
                "summary": "returns the openai writer info ",
                "description": "Retrieve details of an OpenAI generator workbook by slug",
                "operationId": "openAIGeneratorWorkbookApi",
                "parameters": [
                    {
                        "name": "slug",
                        "in": "path",
                        "description": "Slug of the OpenAI generator",
                        "required": true,
                        "schema": {
                            "type": "string"
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "openai": {
                                            "type": "object"
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "404": {
                        "description": "OpenAI generator not found",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "error": {
                                            "type": "string",
                                            "example": "Resource not found"
                                        }
                                    },
                                    "type": "object"
                                }
                        }
                    }
                }
            },
            "security": [
                {
                    "passport": []
                }
            ]
        }
    },
    "/api/aiwriter/generate-output": {
        "post": {
            "tags": [
                "AI Writer"
            ],
            "summary": "Streaming Text Output",
            "description": "Handle streamed text output based on specified parameters.",
            "operationId": "77d4ddc47c1c65d83222acc81ec45458",
            "parameters": [
                {
                    "name": "message_id",
                    "in": "query",
                    "description": "ID of the message",
                    "required": true,
                    "schema": {
                        "type": "integer",
                        "example": 1
                    }
                },
                {
                    "name": "creativity",
                    "in": "query",
                    "description": "Creativity level for the generated content (0 to 1)",
                    "required": true,
                    "schema": {
                        "type": "integer",
                        "example": 1
                    }
                },
                {
                    "name": "maximum_length",
                    "in": "query",
                    "description": "Maximum length of the generated text",
                    "required": true,
                    "schema": {
                        "type": "integer",
                        "example": 500
                    }
                },
                {
                    "name": "number_of_results",
                    "in": "query",
                    "description": "Number of summary results to generate",
                    "required": true,
                    "schema": {
                        "type": "integer",
                        "example": 1
                    }
                }
            ],
            "responses": {
                "200": {
                    "description": "Successful response",
                    "content": {
                        "application/json": {
                                "schema": {
                                    "properties": {
                                        "status": {
                                            "type": "string",
                                            "example": "DONE"
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "412": {
                        "description": "Precondition Failed"
                    },
                    "500": {
                        "description": "Error response",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "error": {
                                            "type": "string",
                                            "example": "Error from API call. Please try
again."
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aiwriter/generate/lazyload": {
            "get": {
                "tags": [
                    "AI Writer"
                ],
                "summary": "Lazy Load Images",
                "description": "Load images based on the specified parameters.",
                "operationId": "1a8987bdfc501929bf043154e4a447b2",
                "parameters": [
                    {
                        "name": "post_type",
                        "in": "query",
                        "description": "Type of post",
                        "required": true,
                        "schema": {
                            "type": "string",
                            "enum": [
                                "ai_image_generator"
                            ],
                            "example": "ai_image_generator"
                        }
                    },
                    {
                        "name": "offset",
                        "in": "query",
                        "description": "Offset for lazy loading",
                        "required": false,
                        "schema": {
                            "type": "integer",
                            "example": 0
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Successful response",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "images": {
                                            "description": "Array of images",
                                            "type": "array",
                                            "items": {
                                                "type": "string"
                                            }
                                        },
                                        "hasMore": {
                                            "description": "Indicates whether there are more
images",
                                            "type": "boolean"
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "404": {
                        "description": "Post not found",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "error": {
                                            "description": "Error message",
                                            "type": "string"
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aiwriter/generate/save": {
            "post": {
                "tags": [
                    "AI Writer"
                ],
                "summary": "Low Generate Save",
                "description": "Save the generated response and update user information.",
                "operationId": "b75116ced05d0eb8ed50cf07ef763767",
                "requestBody": {
                    "required": true,
                    "content": {
                        "application/json": {
                            "schema": {
                                "properties": {
                                    "message_id": {
                                        "description": "ID of the message",
                                        "type": "integer",
                                        "example": 1
                                    },
                                    "response": {
                                        "description": "Generated response",
                                        "type": "string",
                                        "example": "This is a generated response."
                                    }
                                },
                                "type": "object"
                            }
                        }
                    }
                },
                "responses": {
                    "200": {
                        "description": "Successful response",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "status": {
                                            "type": "string",
                                            "example": "Data saved successfully."
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "500": {
                        "description": "Error response",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "error": {
                                            "type": "string",
                                            "example": "Error saving data."
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aiwriter/openai-list": {
            "get": {
                "tags": [
                    "AI Writer"
                ],
                "summary": "Gets all AI Generators related to text",
                "description": "Gets all AI Generators related to text. Controls user access
to premium generators.",
                "operationId": "getOpenAIWriterList",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "data": {
                                            "type": "array",
                                            "items": {
                                                "type": "object"
                                            }
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthorized: User not authenticated"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aiwriter/favorite-openai-list": {
            "get": {
                "tags": [
                    "AI Writer"
                ],
                "summary": "Gets favorite openai list",
                "description": "Returns favorite openai id list of current user.",
                "operationId": "favoriteOpenaiList",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aiwriter/favorite-openai-list-add": {
            "post": {
                "tags": [
                    "AI Writer"
                ],
                "summary": "Adds openai to favorite openai list",
                "description": "Returns favorite openai id list of current user.",
                "operationId": "addToFavoriteOpenaiList",
                "parameters": [
                    {
                        "name": "openai_id",
                        "in": "query",
                        "description": "OpenAI ID",
                        "required": true,
                        "schema": {
                            "type": "integer",
                            "example": 0
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    },
                    "412": {
                        "description": "Precondition Failed"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aiwriter/favorite-openai-list-remove": {
            "post": {
                "tags": [
                    "AI Writer"
                ],
                "summary": "Removes openai from favorite openai list",
                "description": "Returns favorite openai id list of current user.",
                "operationId": "removeFromFavoriteOpenaiList",
                "parameters": [
                    {
                        "name": "openai_id",
                        "in": "query",
                        "description": "OpenAI ID",
                        "required": true,
                        "schema": {
                            "type": "integer",
                            "example": 0
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    },
                    "404": {
                        "description": "Not found"
                    },
                    "412": {
                        "description": "Precondition Failed"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/affiliates/": {
            "get": {
                "tags": [
                    "Affiliates"
                ],
                "summary": "Get affiliate totals of current user",
                "description": "Get affiliate earnings and withdrawal totals of current
user.",
                "operationId": "affiliates",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/affiliates/withdrawals": {
            "get": {
                "tags": [
                    "Affiliates"
                ],
                "summary": "Get withdrawals of current user",
                "description": "Get affiliate withdrawals list of current user.",
                "operationId": "withdrawals",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/affiliates/request-withdrawal": {
            "post": {
                "tags": [
                    "Affiliates"
                ],
                "summary": "Request withdrawal for current user",
                "description": "Request withdrawal for current user",
                "operationId": "requestWithdrawal",
                "requestBody": {
                    "description": "Request withdrawal data",
                    "required": true,
                    "content": {
                        "multipart/form-data": {
                            "schema": {
                                "properties": {
                                    "affiliate_bank_account": {
                                        "description": "Bank Account info",
                                        "type": "string"
                                    },
                                    "amount": {
                                        "description": "Amount to withdraw",
                                        "type": "number",
                                        "format": "float"
                                    }
                                },
                                "type": "object"
                            }
                        }
                    }
                },
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    },
                    "412": {
                        "description": "Precondition Failed"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/app/email-confirmation-setting": {
            "get": {
                "tags": [
                    "App Settings"
                ],
                "summary": "Get email confirmation setting",
                "description": "Get the email confirmation setting from the application
settings.",
                "operationId": "getEmailConfirmationSetting",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                },
                                "example": {
                                    "emailconfirmation": true
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    }
                },
                "security": [
                    {
                        "bearerAuth": []
                    }
                ]
            }
        },
        "/api/app/get-setting": {
            "get": {
                "tags": [
                    "App Settings"
                ],
                "summary": "Get application settings",
                "description": "Get general application settings.",
                "operationId": "getAppSettings",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    }
                },
                "security": [
                    {
                        "bearerAuth": []
                    }
                ]
            }
        },
        "/api/app/usage-data": {
            "get": {
                "tags": [
                    "App Settings"
                ],
                "summary": "Get usage data of current user",
                "description": "Get usage data and subscription plan details of current
user.",
                "operationId": "getUsageData",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/app/currency/{id?}": {
            "get": {
                "tags": [
                    "App Settings"
                ],
                "summary": "Get default currency",
                "description": "Returns default currency if id is not provided, else returns
currency by id. Use 'all' to get all currencies.",
                "operationId": "getCurrency",
                "parameters": [
                    {
                        "name": "id",
                        "in": "path",
                        "description": "Id of currency or 'all' or null to get default
currency.",
                        "required": false,
                        "schema": {
                            "type": "string"
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    },
                    "404": {
                        "description": "No currency found."
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/auth/logo": {
            "get": {
                "tags": [
                    "General (Helpers)"
                ],
                "summary": "Get logo path",
                "description": "Returns logo path from settings.",
                "operationId": "getLogo",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    },
                    "404": {
                        "description": "No logo found."
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/auth/register": {
            "post": {
                "tags": [
                    "Authentication"
                ],
                "summary": "Register a new user",
                "description": "Registers a new user with the provided data",
                "operationId": "register",
                "requestBody": {
                    "required": true,
                    "content": {
                        "application/json": {
                            "schema": {
                                "required": [
                                    "name",
                                    "surname",
                                    "email",
                                    "password",
                                    "password_confirmation"
                                ],
                                "properties": {
                                    "name": {
                                        "type": "string",
                                        "example": "John"
                                    },
                                    "surname": {
                                        "type": "string",
                                        "example": "Doe"
                                    },
                                    "email": {
                                        "type": "string",
                                        "format": "email",
                                        "example": "john.doe@example.com"
                                    },
                                    "password": {
                                        "type": "string",
                                        "format": "password",
                                        "example": "password123"
                                    },
                                    "password_confirmation": {
                                        "type": "string",
                                        "format": "password",
                                        "example": "password123"
                                    },
                                    "affiliate_code": {
                                        "type": "string",
                                        "example": "your_affiliate_code",
                                        "nullable": true
                                    }
                                },
                                "type": "object"
                            }
                        }
                    }
                },
                "responses": {
                    "201": {
                        "description": "User registered successfully"
                    },
                    "422": {
                        "description": "Validation error or user already exists"
                    }
                }
            }
        },
        "/api/auth/logout": {
            "post": {
                "tags": [
                    "Authentication"
                ],
                "summary": "Logout the authenticated user",
                "description": "Logs out the authenticated user and revokes the access
token",
                "operationId": "logout",
                "responses": {
                    "200": {
                        "description": "Logout successful"
                    },
                    "401": {
                        "description": "Unauthenticated"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/auth/forgot-password": {
            "post": {
                "tags": [
                    "Authentication"
                ],
                "summary": "Initiate password reset",
                "description": "Initiate the password reset process by sending an email with
a reset link.",
                "operationId": "forgotPassword",
                "requestBody": {
                    "required": true,
                    "content": {
                        "application/json": {
                            "schema": {
                                "required": [
                                    "email"
                                ],
                                "properties": {
                                    "email": {
                                        "type": "string",
                                        "format": "email",
                                        "example": "user@example.com"
                                    }
                                },
                                "type": "object"
                            }
                        }
                    }
                },
                "responses": {
                    "200": {
                        "description": "Password reset link sent successfully"
                    },
                    "422": {
                        "description": "Validation error or user not found"
                    }
                }
            }
        },
        "/api/auth/email/verify": {
            "get": {
                "tags": [
                    "Authentication"
                ],
                "summary": "Verify user's email",
                "description": "Verify the user's email using the provided confirmation
code.",
                "operationId": "verifyEmail",
                "parameters": [
                    {
                        "name": "email_confirmation_code",
                        "in": "query",
                        "description": "Email confirmation code",
                        "required": true,
                        "schema": {
                            "type": "string"
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Email verified successfully"
                    },
                    "422": {
                        "description": "Validation error or user not found"
                    }
                }
            }
        },
        "/api/auth/email/verify/resend": {
            "post": {
                "tags": [
                    "Authentication"
                ],
                "summary": "Resend confirmation email",
                "description": "Resend the confirmation email to the user if not already
verified.",
                "operationId": "resendConfirmationEmail",
                "requestBody": {
                    "required": true,
                    "content": {
                        "application/json": {
                            "schema": {
                                "required": [
                                    "email"
                                ],
                                "properties": {
                                    "email": {
                                        "type": "string",
                                        "format": "email",
                                        "example": "user@example.com"
                                    }
                                },
                                "type": "object"
                            }
                        }
                    }
                },
                "responses": {
                    "200": {
                        "description": "Confirmation email resent successfully",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                },
                                "example": {
                                    "message": "Confirmation email resent successfully"
                                }
                            }
                        }
                    },
                    "403": {
                        "description": "Email already verified",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                },
                                "example": {
                                    "error": "Email already verified"
                                }
                            }
                        }
                    }
                }
            }
        },
        "/api/auth/social-login": {
            "get": {
                "tags": [
                    "Authentication"
                ],
                "summary": "Get supported login methods",
                "description": "Get actively supported login methods as a list.",
                "operationId": "getSupportedLoginMethods",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "array",
                                    "items": {
                                        "type": "string",
                                        "example": "github"
                                    }
                                },
                                "example": [
                                    "github",
                                    "google"
                                ]
                            }
                        }
                    }
                }
            }
        },
        "/api/brandvoice": {
            "get": {
                "tags": [
                    "Brand Voice"
                ],
                "summary": "List of companies",
                "description": "Returns list of companies",
                "operationId": "index",
                "parameters": [
                    {
                        "name": "id",
                        "in": "path",
                        "description": "ID of company to get. If null, returns all
companies",
                        "required": false,
                        "schema": {
                            "type": "string"
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
```

```
                    "description": "Unauthenticated"
                }
            },
            "security": [
                {
                    "passport": []
                }
            ]
        },
        "post": {
            "tags": [
                "Brand Voice"
            ],
            "summary": "Update / Create a company",
            "description": "Update / Create a company",
            "operationId": "store",
            "responses": {
                "200": {
                    "description": "Successful operation",
                    "content": {
                        "application/json": {
                            "schema": {
                                "type": "object"
                            }
                        }
                    }
                },
                "401": {
                    "description": "Unauthenticated"
                }
            },
            "security": [
                {
                    "passport": []
                }
            ]
        },
        "delete": {
            "tags": [
                "Brand Voice"
            ],
            "summary": "Delete a company",
            "description": "Delete a company by id",
            "operationId": "delete",
            "parameters": [
                {
                    "name": "id",
                    "in": "path",
                    "description": "ID of company to delete",
                    "required": true,
                    "schema": {
                        "type": "string"
                    }
                }
            ],
            "responses": {
                "200": {
                    "description": "Successful operation",
                    "content": {
                        "application/json": {
                            "schema": {
                                "type": "object"
                            }
                        }
                    }
                },
                    "401": {
                        "description": "Unauthenticated"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aichat/chat-templates": {
            "get": {
                "tags": [
                    "Chat Templates"
                ],
                "summary": "Get a list of chat templates or single one if id passed",
                "description": "Get a list of chat templates ordered by name in ascending
order. If an ID is provided, it will retrieve a single chat template.",
                "operationId": "ChatTemplatesController::index",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            },
            "post": {
                "tags": [
                    "Chat Templates"
                ],
                "summary": "!!PATCH REQUEST!! see payload '_method' Update a chat template -
post used only for test",
                "description": "Update the specified chat template in storage. If
`template_id` is 'undefined' or `null`, a new template will be created.",
                "operationId": "ChatTemplatesController@update",
                "requestBody": {
                    "description": "Chat template data",
                    "required": true,
                    "content": {
                        "multipart/form-data": {
                            "schema": {
                                "properties": {
                                    "_method": {
                                        "description": "ID of the chat template",
                                        "type": "string",
                                        "default": "PATCH"
                                    },
                                    "template_id": {
                                        "description": "ID of the chat template",
                                        "type": "string"
                                    },
                                    "name": {
                                        "description": "Name of the chat template",
                                        "type": "string"
                                    },
                                    "short_name": {
                                        "description": "Short name of the chat template",
                                        "type": "string"
                                    },
                                    "description": {
                                        "description": "Description of the chat template",
                                        "type": "string"
                                    },
                                    "role": {
                                        "description": "Role of the chat template",
                                        "type": "string"
                                    },
                                    "human_name": {
                                        "description": "Human name of the chat template",
                                        "type": "string"
                                    },
                                    "helps_with": {
                                        "description": "How the chat template helps",
                                        "type": "string"
                                    },
                                    "color": {
                                        "description": "Color of the chat template",
                                        "type": "string"
                                    },
                                    "chat_completions": {
                                        "description": "Chat completions",
                                        "type": "string"
                                    },
                                    "avatar": {
                                        "description": "Avatar image file (allowed types:
jpg, jpeg, png, svg, webp; max size: 5MB)",
                                        "type": "string",
                                        "format": "binary"
                                    }
                                },
                                "type": "object"
                            }
                        }
                    }
                },
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "message": {
                                            "type": "string",
                                            "example": "Template edited successfully"
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "419": {
                        "description": "Validation error or unsupported file extension",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "errors": {
                                            "type": "array",
                                            "items": {
                                                "type": "string"
                                            }
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/aichat/chat-templates/{id}": {
            "delete": {
                "tags": [
                    "Chat Templates"
                ],
                "summary": "Remove a chat template",
                "description": "Remove the specified chat template from storage.",
                "operationId": "ChatTemplatesController@destroy",
                "parameters": [
                    {
                        "name": "id",
                        "in": "path",
                        "description": "ID of the chat template",
                        "required": true,
                        "schema": {
                            "type": "string"
                        }
                    }
                ],
                "responses": {
                    "204": {
                        "description": "Chat template deleted successfully"
                    },
                    "404": {
                        "description": "Chat template not found",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "message": {
                                            "type": "string",
                                            "example": "Chat template not found"
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "500": {
                        "description": "Server error",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "message": {
                                            "type": "string",
                                            "example": "Internal Server Error"
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/documents/recent": {
            "get": {
                "tags": [
                    "Documents"
                ],
                "summary": "Get the last 10 recent documents",
                "description": "Get the last 10 recent documents for the authenticated user,
excluding documents of type 'image'.",
                "operationId": "getRecentDocs",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "data": {
                                            "type": "array",
                                            "items": {
                                                "type": "object"
                                            }
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthorized: User not authenticated"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/documents/all-openai": {
            "get": {
                "tags": [
                    "Documents"
                ],
                "summary": "Gets all AI Generators",
                "description": "Gets all AI Generators",
                "operationId": "getOpenAIList",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "data": {
                                            "type": "array",
                                            "items": {
                                                "type": "object"
                                            }
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthorized: User not authenticated"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/documents/openai-filters": {
            "get": {
                "tags": [
                    "Documents"
                ],
                "summary": "Gets all AI Filters",
                "description": "Gets all AI Filters",
                "operationId": "getOpenAIFilters",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "data": {
                                            "type": "array",
                                            "items": {
                                                "type": "object"
                                            }
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthorized: User not authenticated"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/documents/": {
            "get": {
                "tags": [
                    "Documents"
                ],
                "summary": "Gets documents",
                "description": "Gets documents for the authenticated user, excluding
documents of type 'image'.",
                "operationId": "getDocs",
                "parameters": [
                    {
                        "name": "search",
                        "in": "query",
                        "description": "Search string.",
                        "required": false,
                        "schema": {
                            "type": "string",
                            "default": ""
                        }
                    },
                    {
                        "name": "filter",
                        "in": "query",
                        "description": "Filter string defined in the 'name' column of the
'openai_filters' table.",
                        "required": false,
                        "schema": {
                            "type": "string",
                            "default": "all"
                        }
                    },
                    {
                        "name": "sort",
                        "in": "query",
                        "description": "Sort string. Values: 'newest', 'oldest', 'az', 'za'",
                        "required": false,
                        "schema": {
                            "type": "string",
                            "default": "newest"
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "data": {
                                            "type": "array",
                                            "items": {
                                                "type": "object"
                                            }
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthorized: User not authenticated"
                    },
                    "412": {
                        "description": "Precondition Failed"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/documents/doc/{id}": {
            "get": {
                "tags": [
                    "Documents"
                ],
                "summary": "Gets single document",
                "description": "Gets document by id, excluding documents of type 'image'.",
                "operationId": "getDoc",
                "parameters": [
                    {
                        "name": "id",
                        "in": "path",
                        "description": "ID of the document to get.",
                        "required": true,
                        "schema": {
                            "type": "string",
                            "default": ""
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "data": {
                                            "type": "array",
                                            "items": {
                                                "type": "object"
                                            }
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthorized: User not authenticated"
                    },
                    "404": {
                        "description": "Document Not Found"
                    },
                    "412": {
                        "description": "Precondition Failed"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            },
            "post": {
                "tags": [
                    "Documents"
                ],
                "summary": "Updates single document",
                "description": "Updates document by id, excluding documents of type
'image'.",
                "operationId": "saveDoc",
                "parameters": [
                    {
                        "name": "id",
                        "in": "path",
                        "description": "ID of the document to get.",
                        "required": true,
                        "schema": {
                            "type": "string",
                            "default": ""
                        }
                    },
                    {
                        "name": "output",
                        "in": "query",
                        "description": "Html Output of the document.",
                        "required": true,
                        "schema": {
                            "type": "string",
                            "default": ""
                        }
                    },
                    {
                        "name": "title",
                        "in": "query",
                        "description": "Title of the document.",
                        "required": true,
                        "schema": {
                            "type": "string",
                            "default": ""
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "data": {
                                            "type": "array",
                                            "items": {
                                                "type": "object"
                                            }
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthorized: User not authenticated"
                    },
                    "404": {
                        "description": "Document Not Found"
                    },
                    "412": {
                        "description": "Precondition Failed"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            },
            "delete": {
                "tags": [
                    "Documents"
                ],
                "summary": "Deletes single document",
                "description": "Deletes document by id, excluding documents of type
'image'.",
                "operationId": "deleteDoc",
                "parameters": [
                    {
                        "name": "id",
                        "in": "path",
                        "description": "ID of the document to delete.",
                        "required": true,
                        "schema": {
                            "type": "string",
                            "default": ""
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "properties": {
                                        "data": {
                                            "type": "array",
                                            "items": {
                                                "type": "object"
                                            }
                                        }
                                    },
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthorized: User not authenticated"
                    },
                    "404": {
                        "description": "Document Not Found"
                    },
                    "412": {
                        "description": "Precondition Failed"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/payment/": {
            "get": {
                "tags": [
                    "Payments"
                ],
                "summary": "Get subscription plan of current user",
                "description": "Get subscription plan details of current user.",
                "operationId": "getCurrentPlan",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    },
                    "404": {
                        "description": "No active subscription found"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/payment/subscriptions/cancel-current": {
            "get": {
                "tags": [
                    "Payments"
                ],
                "summary": "Cancel current subscription plan of current user",
                "description": "Cancel current subscription plan details of current user.",
                "operationId": "cancelActiveSubscription",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/payment/plans/{plan_id?}": {
            "get": {
                "tags": [
                    "Payments"
                ],
                "summary": "Get all plans.",
                "description": "Get all plans.",
                "operationId": "plans",
                "parameters": [
                    {
                        "name": "plan_id",
                        "in": "path",
                        "description": "Id of plan to get details of.",
                        "required": false,
                        "schema": {
                            "type": "string"
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/payment/orders/{order_id?}": {
            "get": {
                "tags": [
                    "Payments"
                ],
                "summary": "Get all orders.",
                "description": "Get all orders. If order_id is provided, then it will return
details of that order.",
                "operationId": "orders",
                "parameters": [
                    {
                        "name": "order_id",
                        "in": "path",
                        "description": "Id of order to get details of.",
                        "required": false,
                        "schema": {
                            "type": "string"
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    },
                    "403": {
                        "description": "Permission Denied"
                    },
                    "404": {
                        "description": "Order not found"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/payment/check-revenue-cat": {
            "get": {
                "tags": [
                    "Payments"
                ],
                "summary": "Trigger RevenueCat to check subscription / token pack status.",
                "description": "Trigger RevenueCat to check subscription / token pack
status.",
                "operationId": "checkRevenueCat",
                "responses": {
                    "200": {
                        "description": "Active subscription found",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    },
                    "404": {
                        "description": "No active subscription found"
                    },
                    "412": {
                        "description": "Error occured. Custom Message displayed"
                    },
                    "500": {
                        "description": "Error occured. Custom Message displayed"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/auth/google-login": {
            "post": {
                "tags": [
                    "Authentication"
                ],
                "summary": "Social Login with Google",
                "description": "Social Login with Google",
                "operationId": "google",
                "requestBody": {
                    "description": "Google token data",
                    "required": true,
                    "content": {
                        "multipart/form-data": {
                            "schema": {
                                "properties": {
                                    "google_token": {
                                        "description": "Google Access Token",
                                        "type": "string"
                                    },
                                    "google_id": {
                                        "description": "Google User ID (OpenID)",
                                        "type": "string"
                                    }
                                },
                                "type": "object"
                            }
                        }
                    }
                },
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    },
                    "412": {
                        "description": "Precondition Failed"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/auth/apple-login": {
            "post": {
                "tags": [
                    "Authentication"
                ],
                "summary": "Social Login with Apple",
                "description": "Social Login with Apple",
                "operationId": "apple",
                "requestBody": {
                    "description": "Apple token data",
                    "required": true,
                    "content": {
                        "multipart/form-data": {
                            "schema": {
                                "properties": {
                                    "apple_token": {
                                        "description": "Apple Access Token",
                                        "type": "string"
                                    },
                                    "apple_id": {
                                        "description": "Apple User ID (OpenID)",
                                        "type": "string"
                                    }
                                },
                                "type": "object"
                            }
                        }
                    }
                },
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    },
                    "412": {
                        "description": "Precondition Failed"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/auth/profile": {
            "get": {
                "tags": [
                    "User Profile"
                ],
                "summary": "Get user profile",
                "description": "Get the profile of the authenticated user.",
                "operationId": "UserController::index",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            },
            "post": {
                "tags": [
                    "User Profile"
                ],
                "summary": "!!PATCH REQUEST!! see payload '_method' Update user profile -
post used only for test",
                "description": "Updates the user's profile information, including name,
surname, phone, country, password, and avatar.",
                "operationId": "UserController::update",
                "requestBody": {
                    "description": "User data",
                    "required": true,
                    "content": {
                        "multipart/form-data": {
                            "schema": {
                                "properties": {
                                    "name": {
                                        "description": "User's first name",
                                        "type": "string"
                                    },
                                    "_method": {
                                        "description": "ID of the chat template",
                                        "type": "string",
                                        "default": "PATCH"
                                    },
                                    "surname": {
                                        "description": "User's last name",
                                        "type": "string"
                                    },
                                    "phone": {
                                        "description": "User's phone number",
                                        "type": "string"
                                    },
                                    "country": {
                                        "description": "User's country",
                                        "type": "string"
                                    },
                                    "old_password": {
                                        "description": "User's old password for
verification",
                                        "type": "string"
                                    },
                                    "new_password": {
                                        "description": "User's new password",
                                        "type": "string"
                                    },
                                    "new_password_confirmation": {
                                        "description": "Confirmation of the new password",
                                        "type": "string"
                                    },
                                    "avatar": {
                                        "description": "User's avatar image file (allowed
types: jpg, png; max size: 5MB)",
                                        "type": "string",
                                        "format": "binary"
                                    }
                                },
                                "type": "object"
                            }
                        }
                    }
                },
                "responses": {
                    "400": {
                        "description": "Bad request"
                    },
                    "404": {
                        "description": "Resource Not Found"
                    },
                    "200": {
                        "description": "User settings saved successfully",
                        "content": {
                            "application/json": {}
                        }
                    },
                    "401": {
                        "description": "Unauthorized: User not authenticated"
                    },
                    "419": {
                        "description": "Validation error or unsupported file extension"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            },
            "delete": {
                "tags": [
                    "User Profile"
                ],
                "summary": "Delete user account",
                "description": "Get the profile of the authenticated user and delete
account.",
                "operationId": "UserController::destroy",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/support/": {
            "get": {
                "tags": [
                    "Support"
                ],
                "summary": "Gets all support requests",
                "description": "Gets all support requests",
                "operationId": "supportRequests",
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/support/ticket/{ticket_id}": {
            "get": {
                "tags": [
                    "Support"
                ],
                "summary": "Gets all messages of a support request",
                "description": "Gets all messages of a support request. Use ticket ids like
QZDNGSIFPH not integers.",
                "operationId": "ticket",
                "parameters": [
                    {
                        "name": "ticket_id",
                        "in": "path",
                        "description": "Ticket ID",
                        "required": true,
                        "schema": {
                            "type": "string",
                            "example": "QZDNGSIFPH"
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    },
                    "403": {
                        "description": "Unauthorized"
                    },
                    "404": {
                        "description": "Ticket Not Found"
                    },
                    "412": {
                        "description": "Precondition Failed"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/support/ticket/{ticket_id}/last-message": {
            "get": {
                "tags": [
                    "Support"
                ],
                "summary": "Gets latest message of a support request",
                "description": "Gets latest message of a support request. Use ticket ids like
QZDNGSIFPH not integers.",
                "operationId": "ticketLastMessage",
                "parameters": [
                    {
                        "name": "ticket_id",
                        "in": "path",
                        "description": "Ticket ID",
                        "required": true,
                        "schema": {
                            "type": "string",
                            "example": "QZDNGSIFPH"
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    },
                    "403": {
                        "description": "Unauthorized"
                    },
                    "404": {
                        "description": "Ticket Not Found"
                    },
                    "412": {
                        "description": "Precondition Failed"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/support/new-ticket": {
            "post": {
                "tags": [
                    "Support"
                ],
                "summary": "Create new support request",
                "description": "Create new support request",
                "operationId": "newTicket",
                "requestBody": {
                    "description": "Request support ticket data",
                    "required": true,
                    "content": {
                        "multipart/form-data": {
                            "schema": {
                                "properties": {
                                    "priority": {
                                        "description": "Priority",
                                        "type": "string",
                                        "enum": [
                                            "Low",
                                            "Normal",
                                            "High",
                                            "Critical"
                                        ]
                                    },
                                    "category": {
                                        "description": "Category",
                                        "type": "string",
                                        "enum": [
                                            "General Inquiry",
                                            "Technical Issue",
                                            "Improvement Idea",
                                            "Feedback",
                                            "Other"
                                        ]
                                    },
                                    "subject": {
                                        "description": "Subject of the Message",
                                        "type": "string"
                                    },
                                    "message": {
                                        "description": "Message to send",
                                        "type": "string"
                                    }
                                },
                                "type": "object"
                            }
                        }
                    }
                },
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    },
                    "412": {
                        "description": "Precondition Failed"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/support/send-message": {
            "post": {
                "tags": [
                    "Support"
                ],
                "summary": "Send message to support request",
                "description": "Send message to support request",
                "operationId": "sendMessage",
                "requestBody": {
                    "description": "Request message data",
                    "required": true,
                    "content": {
                        "multipart/form-data": {
                            "schema": {
                                "properties": {
                                    "ticket_id": {
                                        "description": "Ticket ID",
                                        "type": "string",
                                        "example": "QZDNGSIFPH"
                                    },
                                    "message": {
                                        "description": "Message to send",
                                        "type": "string"
                                    }
                                },
                                "type": "object"
                            }
                        }
                    }
                },
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    },
                    "412": {
                        "description": "Precondition Failed"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        },
        "/api/support/user/{ticket_id}": {
            "get": {
                "tags": [
                    "Support"
                ],
                "summary": "Gets information of the user which sent the request",
                "description": "Gets information of the user which sent the request. Use
ticket ids like QZDNGSIFPH not integers.",
                "operationId": "ticketUser",
                "parameters": [
                    {
                        "name": "ticket_id",
                        "in": "path",
                        "description": "Ticket ID",
                        "required": true,
                        "schema": {
                            "type": "string",
                            "example": "QZDNGSIFPH"
                        }
                    }
                ],
                "responses": {
                    "200": {
                        "description": "Successful operation",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "type": "object"
                                }
                            }
                        }
                    },
                    "401": {
                        "description": "Unauthenticated"
                    },
                    "403": {
                        "description": "Unauthorized"
                    },
                    "404": {
                        "description": "Ticket / User Not Found"
                    },
                    "412": {
                        "description": "Precondition Failed"
                    }
                },
                "security": [
                    {
                        "passport": []
                    }
                ]
            }
        }
    },
    "tags": [
        {
            "name": "AI Chats (Convesations)",
            "description": "AI Chats (Convesations)"
        },
        {
            "name": "AI Chats (Messages)",
            "description": "AI Chats (Messages)"
        },
        {
            "name": "General (Helpers)",
            "description": "General (Helpers)"
        },
        {
            "name": "AI Writer",
            "description": "AI Writer"
        },
        {
            "name": "AI Image Generation",
            "description": "AI Image Generation"
        },
        {
            "name": "Affiliates",
            "description": "Affiliates"
        },
        {
            "name": "App Settings",
            "description": "App Settings"
        },
        {
            "name": "Authentication",
            "description": "Authentication"
        },
        {
            "name": "Brand Voice",
            "description": "Brand Voice"
        },
        {
            "name": "Chat Templates",
            "description": "Chat Templates"
        },
        {
            "name": "Documents",
            "description": "Documents"
        },
        {
            "name": "Payments",
            "description": "Payments"
        },
        {
            "name": "User Profile",
            "description": "User Profile"
        },
        {
            "name": "Support",
            "description": "Support"
        }
    ],
    "components": {
        "securitySchemes": {
            "passport": {
                "type": "oauth2",
                "description": "Laravel passport oauth2 security.",
                "in": "header",
                "scheme": "https",
                "flows": {
                    "password": {
                        "authorizationUrl": "http://localhost:8000/oauth/authorize",
                        "tokenUrl": "http://localhost:8000/oauth/token",
                        "refreshUrl": "http://localhost:8000/token/refresh",
                        "scopes": []
                    }
                }
            }
        }
    }
}
```