

Documentation and Workflow for Website Chatbot Script

Purpose

The Website Chatbot is a Python-based application that allows users to interact with and extract information from any website using OpenAI's GPT-3.5 model. The chatbot can understand website content and engage in meaningful conversations about it.

Requirements

Dependencies

```
pip install openai selenium beautifulsoup4 webdriver-manager python-dotenv
```

API Keys

- **OpenAI API Key (REQUIRED):**
 - **Not included in code:** For security reasons, the OpenAI API key is not included in the script.
 - **User responsibility:** Users must use their own OpenAI API key.
 - **How to get an API key:**
 1. Sign up at <https://platform.openai.com>.
 2. Navigate to API section.
 3. Generate your API key.
 - **Two ways to add your API key:**
 1. Environment variable: Set `OPENAI_API_KEY` in your system.
 2. Direct input: Enter when prompted by the script.

Other Requirements

- Chrome WebDriver (automatically installed).
-

Workflow

1. Script Initialization

- **Logging Configuration:**
 - Logs critical warnings and errors.
 - Suppresses unnecessary logs from dependencies like `selenium` and `webdriver_manager`.
- **Welcome Message:**
 - Displays an ASCII banner and introduces the chatbot.
- **Environment Variables:**

- Loads the OpenAI API key using `dotenv`.

2. User Input Handling

- Prompts the user for a website URL.
- Validates the URL using `urlparse` to ensure proper format (e.g., `http` or `https` schema).
- Allows user commands:
 - `exit`: Quit the script.
 - `new`: Switch to a different website.

3. Website Content Extraction

- **Selenium WebDriver:**
 - Configures `ChromeDriver` in headless mode for efficient content scraping.
 - Loads the webpage and scrolls to dynamically load all content.
- **HTML Parsing:**
 - Uses `BeautifulSoup` to parse the website content.
 - Strips unnecessary tags (e.g., `script`, `style`, `nav`, etc.).
 - Extracts meaningful content such as headers (`h1`, `h2`, `h3`) and paragraphs (`p`, `div`).
- **Content Cleaning:**
 - Joins extracted headers and paragraphs.
 - Removes excessive whitespace.
 - Truncates content to fit within OpenAI's 4000-character limit.

4. Chatbot Context Creation

- Builds an initial context with the extracted content and website domain.
- Ensures the chatbot:
 - Provides accurate and concise answers.
 - Engages users with a friendly tone and occasional emojis.
 - Redirects irrelevant questions to the website's content.

5. OpenAI Chatbot Interaction

- **Conversation History:**
 - Maintains a history of up to 10 exchanges for context.
- **API Request:**
 - Queries OpenAI's GPT model (`gpt-3.5-turbo`) with the user's question and context.
 - Uses parameters like:
 - `max_tokens`: Limits response length.
 - `temperature`: Balances creativity and consistency.
 - `presence_penalty`: Encourages concise responses.
- **Response Formatting:**
 - Cleans up markdown from API responses.
 - Properly formats numbered or bulleted lists.
 - Adds separators for better readability.

6. Error Handling

- **User Input Errors:**
 - Prompts for re-entry of invalid URLs.
 - **Extraction Failures:**
 - Notifies the user if content extraction fails.
 - **API Errors:**
 - Logs errors and gracefully informs the user.
-

Features

Core Functionality

1. **Website Content Analysis:**
 - Dynamic content loading.
 - Automatic scrolling.
 - Content cleaning and formatting.
2. **Interactive Chat:**
 - Natural language processing.
 - Context-aware responses.
 - Markdown support.
 - Emoji integration.
3. **Navigation Commands:**
 - `new`: Switch to a different website.
 - `exit`: End program.
 - Empty input handling.

Error Handling

- URL validation.
 - Content extraction errors.
 - API response errors.
 - Keyboard interrupts.
 - General exceptions.
-

Usage Guide

Starting the Program

```
python website_chatbot.py
```

Basic Commands

- Enter URL: Input any valid website URL.
- Ask Questions: Type any question about the website.
- Switch Website: Type `new`.

- Exit Program: Type `exit`.

Example Interaction

```

Website Chatbot Assistant 🌐
Chat with any website and learn more!

Please enter your OpenAI API key: Use your API. Hiding API for security reasons.

🔗 Enter website URL (or 'exit' to quit): https://relinns.com/
Loading website content... ⏳
Scanning website content... 🔍

🌟 Ready! Ask me anything about https://relinns.com/
💡 Type 'new' for a different website or 'exit' to quit

You: Tell me baout Relinns!!

🤖 Bot Response:

Relinns is an AI development company that focuses on innovation, reliability, and speed to build AI solutions exceeding expectations. They live by the motto "To Change, To Enhance, To Grow" and offer services like AI development, chatbot solutions, and generative AI. 🤖

They have developed platforms like AppsRhino for AI eCommerce and BotPenguin for generative AI chatbots. Their goal is to empower businesses by transforming ideas into reality using their AI expertise and customization options. 🚀

If you want to know more about how they are impacting industries and changing the game with AI, feel free to ask!

You: What they are specialized in?

🤖 Bot Response:

Relinns specializes in developing innovative AI solutions that can transform ideas into reality. They focus on AI development, chatbot solutions, and generative AI. Their expertise lies in creating customized AI applications that exceed expectations in terms of reliability, innovation, and speed. 🤖

Specifically, they offer services such as building AI eCommerce platforms like AppsRhino and generative AI chatbots like BotPenguin. These solutions aim to empower businesses by providing automation for tasks like lead generation, customer support, and appointment booking. 📅

If you have any more questions about their specialization or the industries they impact, feel free to ask!

```

Best Practices

URL Input

- Always include protocol (`http://` or `https://`).
- Ensure URL is complete and valid.
- Wait for content loading to complete.

Asking Questions

- Be specific about what you want to know.
- Focus on website-related queries.
- Use clear and simple language.

Performance Considerations

- Large websites may take longer to analyze.
- Complex questions may require more tokens.

- Consider conversation history limits.
-

Troubleshooting

Common Issues

1. **API Key Errors:**
 - Check API key validity.
 - Ensure proper environment variable setup.
2. **Website Loading Issues:**
 - Verify internet connection.
 - Check URL accessibility.
 - Allow sufficient loading time.
3. **Response Errors:**
 - Check input formatting.
 - Verify website content extraction.
 - Monitor API rate limits.

Support

For issues related to:

- **OpenAI API:** Visit OpenAI support.
 - **Selenium:** Check Selenium documentation.
 - **General bugs:** Review error messages and logs.
-

Security Notes

API Key Security

- NEVER share your OpenAI API key.
- NEVER commit API keys to version control.
- Each user must use their own API key.
- Consider using environment variables for API key storage.

General Security

- No sensitive data is stored.
 - Website content is processed locally.
 - Secure HTTPS connections preferred.
 - Session data is temporary and not persisted.
-

Thank You

Thank you for taking the time to use and explore the Website Chatbot documentation. I hope you find it helpful and informative!