# Document Generator Comprehensive Guide

[document-generator]

**Date:** 8/13/2025 04:57:54 PM

## Introduction

The Document Generator is a powerful AI-driven application designed to streamline the process of creating structured documents from user prompts and integrated resource files. Its main intent is to simplify document generation by transforming natural language inputs into comprehensive templates, allowing users to combine various file formats and leverage AI to craft content efficiently. This tool caters to both technical and non-technical users, offering a user-friendly interface and versatile functionalities to meet diverse documentation needs.

With the Document Generator, users can build documents by starting from a simple prompt, manually constructing templates, or utilizing pre-existing ones. It emphasizes flexibility and ease of use, enabling the creation, modification, and delivery of high-quality documents tailored to specific requirements. Ultimately, the Document Generator aspires to enhance productivity and ensure that users can easily generate well-structured and informative documents.

## Vision and Use Cases

The vision behind the Document Generator is to transform the way individuals and organizations create and manage documentation. By leveraging advanced AI technologies, the application aims to provide a seamless user experience that simplifies the document generation process while ensuring high-quality output and robust customization options. The Document Generator facilitates not just efficiency but also creativity in document creation, empowering users to focus on content rather than structure.

### Use Cases

1. **Technical Documentation**: Software developers can use the Document Generator to create comprehensive and structured technical documentation, such as API references or project README files. By inputting technical specifications in natural language, the app generates structured documents that are easy to understand and navigate.
2. **Project Proposals**: Project managers can create detailed project proposals by providing an overview of the project goals and required resources. The Document Generator can help outline the proposal, add necessary sections like budget estimations, timelines, and deliverables, and generate a polished document that can be presented to stakeholders.
3. **Performance Reviews**: HR professionals can utilize the tool to streamline the performance review process. By inputting specific review criteria and employee data, the Document Generator can produce structured performance review documents tailored to each employee, enhancing consistency and objectivity.
4. **Educational Materials**: Educators can generate course materials, syllabi, and lesson plans. By specifying the learning objectives and desired outcomes, teachers can create comprehensive guides that integrate relevant resources and materials, saving time in the curriculum design process.
5. **Meeting Notes and Summaries**: Professionals can automate the generation of meeting notes and summaries by feeding in agenda items and discussion points. The Document Generator can format these inputs into easily readable summaries, ensuring that critical information is captured and shared efficiently after each meeting.

In summary, the Document Generator is designed to support a variety of document creation needs across different sectors by enhancing efficiency, accuracy, and overall productivity.

## How to Use the Document Generator

Using the Document Generator is a straightforward process designed to accommodate users of all expertise levels, from beginners to seasoned developers. This section serves as an introduction to the essential functionalities with a focus on easing your document creation journey. Whether you are looking to generate simple outlines or intricate documents, the Document Generator provides a user-friendly interface to help you achieve your objectives.

For those interested in setting up the Document Generator locally, detailed instructions will be covered in subsequent sections. This setup is typically recommended for developers who want to customize their environment or integrate the tool into other workflows. You will find guidance on how to install, configure, and utilize the Document Generator effectively, ensuring you are well-equipped to harness its full potential.

### Creating and Updating Documents

Creating documents using the Document Generator app is an intuitive process designed to accommodate various user needs, from simple prompts to comprehensive templates. Here’s a detailed overview of how to create, edit, and manage your documents:

1. **Starting the Document Creation**  
   Users can choose one of three approaches to begin the document creation process:
   * **Start with a Prompt**: Type a description of the document you wish to create. For instance, you might input “Draft a project proposal for a new marketing strategy.” Upon clicking “Draft”, the AI will utilize the prompt to generate a structured outline tailored to your needs.
   * **Build Manually**: Users can opt to construct the document template section by section, allowing for complete control over the structure and content. Use the interface to add new sections, subsections, and content directly.
   * **Load a Template**: The app offers pre-built example templates that can be loaded to get started quickly. Users can choose from a variety of common document types, such as performance reviews or technical documentation.
2. **Editing and Customizing**  
   After the initial creation of a document, users can refine their outlines by adding, modifying, or deleting sections according to their evolving requirements. This involves selecting existing sections, adding new ones, and specifying whether AI-generated content or manual text should fill each section.
   * Users can click on **“+ Add section”** to append new sections or use the **“+”** button to insert sections at preferred points in the outline.
   * The structure can be enhanced further by indenting sections to create subsections, which helps in organizing the content hierarchically.
3. **Incorporating Resources**  
   The application allows the uploading of additional relevant resources (such as CSVs, text files, or JSON files) that can complement the document content. Users can update these resources at any point by adding new files or modifying existing ones, ensuring that the document content remains current and relevant.
4. **Saving and Importing Documents**  
   To facilitate continued work on documents, users can save their outlines and resources as **.docpack** files. This not only preserves the current state of a document but also allows for easy sharing and collaboration with other users.
   * **Importing Documents**: Users can later import a .docpack file using the app’s import feature. This functionality is particularly useful for resuming work on previously created documents or for utilizing established outlines as a foundation for new projects.
5. **Regenerating Documents**  
   Once a file is saved and the document is reloaded, users have the option to adjust any sections that need updating. After making necessary edits or adding new content, clicking the **“▷ Generate”** button will prompt the AI to regenerate the document based on the latest inputs and resources. This ensures that users can efficiently refine their documents without starting from scratch each time.

By embracing these functionalities, the Document Generator app empowers users to effortlessly create, update, and manage their documents, resulting in high-quality outputs tailored to specific needs.

### Local Installation and Setup

Setting up the Document Generator locally allows developers to run their own instance of the application, facilitating customization and deeper integration into existing workflows. The following steps outline the process for installing and configuring the Document Generator on a local machine, specifically for Linux environments:

1. **Clone the Repository**  
   Begin by cloning the Document Generator repository from GitHub to your local machine. Open your terminal and enter:

* git clone https://github.com/microsoft/recipe-tool.git  
  cd recipe-tool

1. **Install Dependencies**  
   Next, install the necessary dependencies required for the application. You can do this by running:

* make install
* After the installation, activate the virtual environment:
* source .venv/bin/activate

1. **Configure Environment Variables**  
   Create an environment configuration file by copying the example file provided in the repository. This step is crucial for setting up necessary API keys for the Large Language Model (LLM) provider:

* cp .env.example .env
* Open the .env file with your preferred text editor and fill in the required fields. Be sure to include your LLM provider’s API key. For instance, if you are using OpenAI, update the key as follows:
* RECIPE\_EXECUTOR\_OPENAI\_API\_KEY= your\_openai\_api\_key

1. **Run the Application**  
   To launch the Document Generator app, execute the following command in your terminal:

* document-generator-app
* If you’ve run the deployment step make build, use the following command instead to start the app in development mode:
* document-generator-app --dev
* This option allows you to see live changes without needing to rebuild the application each time.

1. **Access the Web App**  
   Finally, open your web browser and navigate to the local instance of the Document Generator, typically available at http://localhost:7860. From here, you can start utilizing the application by creating documents based on your requirements.

By following these steps, you’ll have a fully functional local installation of the Document Generator, ready for crafting documents and experimenting with its numerous features.

## Understanding Recipes

In the context of the Document Generator, recipes serve as predefined sets of instructions that guide the document generation process. Each recipe outlines a sequence of steps to be executed, utilizing large language models (LLMs) to produce coherent and structured content from given inputs. Recipes are central to the functionality of the Document Generator, allowing users to automate complex workflows and ensure consistency across generated documents.

### Purpose of Recipes

Recipes primarily facilitate the generation and manipulation of documents by encoding essential processes into a structured format. They abstract the underlying complexity, enabling users to generate documents without requiring an in-depth understanding of the technicalities involved. This results in: - **Efficiency**: Recipes streamline the document creation process, reducing the manual effort and time required. - **Reusability**: Users can save recipes for repeated use, enhancing productivity by standardizing processes across multiple documents. - **Customizability**: Recipes can be adjusted or created to cater to specific document generation needs, providing flexibility in how content is produced.

### Application of Recipes

Different types of recipes exist within the Document Generator ecosystem, each designed to tackle specific aspects of document creation. Here are some key examples: - **Document Generation Recipes**: These recipes, such as document\_generator\_recipe.json, define how documents are generated from outlines. They specify inputs like the outline file path, LLM model to be used, and the output directory. They also incorporate context-setting steps that prepare necessary parameters before executing the core document creation logic.

* **Outline Generation Recipes**: Recipes like generate\_outline.json are used to create a structured outline for the document. They leverage LLMs to interpret user descriptions and summarize resources, producing a well-defined document outline in JSON format that guides the eventual content generation.
* **Docpack Generation Recipes**: The generate\_docpack.json recipe exemplifies how to compile all necessary components of a document into a comprehensive package. This recipe focuses on ensuring that all necessary sections, resources, and configurations are collated effectively before the final document generation occurs.
* **Utility Recipes**: Additional recipes may assist in supporting tasks, such as loading resources or validating the structure of the document outline. These utility recipes streamline operations, helping to prevent common issues and enhance the stability of the document generation process.

Through the use of recipes, the Document Generator not only simplifies the document creation experience but also empowers users to focus on content development rather than the intricacies of document formatting and assembly.

## Working with the Outline

In the Document Generator, the outline.json file plays a crucial role by facilitating two primary functions: it helps to populate the template editing panel and serves as input for the document\_generator\_recipe.json. Crafting a well-structured outline.json file is essential for ensuring the smooth operation of the Document Generator app. This section provides guidance on how to write these files effectively.

### Structure of outline.json

The outline.json file must follow a specific structure that encompasses key elements required for document generation. Below is the general format:

{  
 "title": "Document Title",  
 "general\_instruction": "Instructions for the document generation process.",  
 "resources": [  
 {  
 "key": "resource\_1",  
 "path": "path/to/resource",  
 "title": "Resource Title",  
 "description": "Description of the resource."  
 }  
 ],  
 "sections": [  
 {  
 "title": "Section Title",  
 "prompt": "Instructions for generating this section's content.",  
 "sections": [], // Nested subsections, if needed  
 "refs": ["resource\_1"] // Array of resource keys  
 }  
 ]  
}

### Key Components

1. **Title**: The title of the document should be clear and descriptive, as it represents the main topic of the generated content.
2. **General Instruction**: This field should contain overall guidance on how the document should be generated, aiding the AI in producing relevant and compliant content.
3. **Resources**: This array holds objects that represent each resource utilized in the document. Each resource must include keys such as key, path, title, and description, with the path accurately reflecting the location of the file.
4. **Sections**: The sections array is critical for organizing the document’s content. Each section must include:
   * **Title**: A descriptive title indicating the focus of that section.
   * **Prompt**: Instructions for what content should be generated within the section.
   * **Sections** (optional): This allows for nested subsections, enhancing the document’s depth and organization.
   * **Refs**: An array of string references that connect sections to corresponding resources from the resources array. This linkage is vital for ensuring the AI knows which resources to pull from when generating content for each section.

### Usage in the Template Editing Panel

When the outline.json file is correctly formatted, it seamlessly populates the template editing panel, providing an intuitive organization of document sections and resources. This organization facilitates easier navigation and editing, ensuring users can modify their documents with confidence.

### Integration with document\_generator\_recipe.json

The outline.json file serves as a direct input for the document\_generator\_recipe.json, streamlining the document creation process. It is imperative that the outline adheres to the defined structure, ensuring compatibility and functionality within the recipe execution workflows. This integration allows for an automated process whereby the structure provided by outline.json translates directly into generated document content, saving time and reducing errors.

By following these guidelines, users can effectively write and utilize outline.json files to harness the full potential of the Document Generator application.

## Future Developments and Ideas

As we look ahead, several exciting features and ideas are poised to expand the capabilities of the Document Generator. These enhancements aim to improve user experience, streamline processes, and leverage advanced technologies to offer more intuitive document creation and management solutions.

### Chat-Driven Recipes

One of the primary features being considered is the implementation of chat-driven recipes. This would allow users to engage with a side chat assistant while working on their documents. Users could initiate commands, request changes, and seek guidance directly within the chat interface, enabling a more interactive and collaborative editing experience.

### API Integration

To facilitate the chat-driven recipes, we are exploring the use of a Model-Client Protocol (MCP) or REST API. These APIs could drive the chat feature, allowing for real-time communication between the Document Generator and the assistant. This integration would make it possible for the chat assistant to access and manipulate document templates dynamically, enhancing the user’s ability to customize and refine their work on the fly.

### Streamlined Resource Integration

We are also aiming to develop more seamless ways to import resources into the document. This includes integrating URL links that can pull in the most recent changes from online documents or storage services. Additionally, there will be a focus on detecting automatic changes in files, allowing users to stay updated without manual intervention. Smart gathering of files could aggregate related documents and resources in a way that is intuitive and user-friendly.

### Intelligent Document Regeneration

Another strategic focus is the smart regeneration of documents. Rather than regenerating the entire document every time an update is made, we plan to implement a system where individual sections or subsections can be regenerated. This would provide users with the option to lock certain sections in place while allowing for flexible updates in others. Such a feature would bolster user confidence and trust in the generated output, as it ensures consistency where desired while maintaining the ability to adapt and modify content quickly.

By implementing these developments, the Document Generator aims to not only enhance the functionality available to users but also to facilitate a more engaging and efficient document creation experience.