Docify: Automated Documentation Generator

1. Project Overview

Docify is a backend service that automatically generates documentation for GitHub repositories. It leverages a large language model (LLM) to analyze the repository's code, README, and other files to produce a comprehensive documentation set. This solves the problem of manually writing and maintaining documentation, saving developers significant time and effort. The target users are software developers and teams who want to improve their project documentation practices. The main purpose is to streamline the documentation process and ensure projects are well-documented.

2. Tech Stack

- **Backend:** Node.js, Express.js
- **Languages:** JavaScript
- **Libraries:** `express`, `cors`, `path`, `url`, `dotenv`, `@google/generative-ai`
- **Database:** None (uses temporary file storage for cloned repositories)
- **API:** Google Gemini API (for generating documentation content)

3. Installation & Setup

Prerequisites:

- Node.js and npm (or yarn)
- A Google Cloud project with the Gemini API enabled and a valid API key. Set the `GEMINI_API_KEY` environment variable.

Steps:

1. Clone the repository: `git clone <repository_url>`
2. Navigate to the project directory: `cd docify`
3. Install dependencies: `npm install`
4. Set up the `.env` file (create one if it doesn't exist) and add your `GEMINI_API_KEY`

   ```
   GEMINI_API_KEY=<your_api_key>
   ```

5. Start the server: `npm start`

4. Usage Guide

Running the Project (Development):

1. Follow the Installation & Setup steps.
2. The server will start listening on port 5000 (or the port defined in your `.env` file).

Running the Project (Production): The project can be deployed to any Node.js compatible hosting service. The `server.js` file contains the server start-up logic.

Example API Usage:

To generate documentation, send a POST request to `/api/docify/generate` with the following JSON payload:

```
{
  "repoUrl": "https://github.com/<username>/<repository>",
  "exportFormat": "md" // Optional, defaults to 'md'.  Supports other formats as implemented in the `exportUtils` (currently only md is implemented)
}
```

A successful response will contain a JSON object with a download URL for the generated documentation.

Downloading the Documentation:

The documentation is available for download via a GET request to the URL returned in the previous step.

5. Features

- Automatic documentation generation from GitHub repositories.
- Support for multiple documentation formats (currently only Markdown).
- Uses Google Gemini API for advanced content generation.
- Handles GitHub repository cloning and cleanup.

6. Codebase Walkthrough

Folder Structure:

- **`app.js`**: Express.js server setup, including middleware and route definitions.
- **`controllers`**: Contains controller logic for handling API requests.
  - **`docifyController.js`**: Contains the core logic for generating and downloading documentation.
- **`routes`**: Defines API routes.
  - **`docifyRoutes.js`**: Defines routes for generating and downloading documentation.
- **`utils`**: Helper functions.
  - **`gitUtils.js`**: Handles Git repository cloning and cleanup (using node-git).
  - **`parseUtils.js`**: Parses the cloned repository to extract relevant information (README, code files).
  - **`aiUtils.js`**: Interacts with the Google Gemini API to generate documentation content.
  - **`exportUtils.js`**: Handles the export of generated documentation in various formats.

- **`exports`**: Directory for storing generated documentation files.
- **`server.js`**: Server entry point.

Important Files:

- **`app.js`**: Sets up the Express.js server, including middleware and routing.
- **`docifyController.js`**: Handles the API request for documentation generation. It coordinates the steps involved: cloning the repo, parsing the repo contents, generating content via AI, exporting the documentation, and cleaning up.
- **`docifyRoutes.js`**: Defines the Express.js routes.
- **`aiUtils.js`**: Contains the logic for interacting with Google's Gemini API. Uses the Gemini API to generate documentation based on the prompt it constructs from the parsed repository content.
- **`server.js`**: The main entry point for starting the server.

7. API Documentation

POST /api/docify/generate

- **Request Body:**

```
{
  "repoUrl": "string", // Required: GitHub repository URL
  "exportFormat": "string" // Optional: Export format ('md' is the only implemented format currently)
}
```

- **Response (Success):**

```
{
  "downloadUrl": "string" // URL to download the generated documentation
}
```

- **Response (Error):**

```
{
  "error": "string" // Error message
}
```

GET /api/docify/download/:filename

- **Parameters:**
    - `filename`: The name of the documentation file to download.
- **Response:** The generated documentation file (e.g., markdown file).

8. Contribution Guide

1. **Fork** the repository.
2. **Create a new branch:** `git checkout -b my-new-feature`
3. **Make your changes** and commit them: `git commit -m "Add some feature"`
4. **Push** to your forked repository: `git push origin my-new-feature`
5. **Create a pull request** from your branch to the main branch of this repository.

Coding Style: The project uses standard JavaScript conventions.

9. License & Credits

This project is currently without a defined license.

Credits:

- Express.js
- Google Gemini API
- Various Node.js packages listed in `package.json`