Basic template structure for API documentation:

1. Introduction:
   - Overview: Provide a high-level overview of the API, its purpose, and key features.
   - Target Audience: Describe the intended audience for the documentation (developers, partners, etc.).
   - Authentication: Explain the authentication methods required to access the API.

2. Getting Started:
   - Base URL: Specify the base URL for all API endpoints.
   - Authentication: Provide details on how to authenticate requests (API keys, tokens, OAuth, etc.).
   - Request Headers: List any required or optional headers to include in API requests.
   - Response Format: Describe the format of API responses (JSON, XML, etc.).

3. Endpoints: For each API endpoint, include the following details:
   - Endpoint URL: Specify the URL of the endpoint.
   - HTTP Method: Indicate the supported HTTP method (GET, POST, PUT, DELETE, etc.).
   - Parameters: List all query parameters, path parameters, and request body parameters.
   - Request Example: Provide an example of a request with sample data.
   - Response Example: Include an example of a successful response with sample data.
   - Error Handling: Document potential error responses and their corresponding error codes.

4. Resources:
   - Data Models: Describe the data models used in the API, including their properties and types.
   - Pagination: If applicable, explain how pagination is handled in API responses.
   - Rate Limiting: Specify any rate limits or usage restrictions imposed on the API.

5. Best Practices:
   - API Usage Guidelines: Provide recommendations on how to effectively use the API.
   - Performance Optimization: Include tips on optimizing API performance and reducing latency.
   - Error Handling: Offer guidance on handling errors and error response codes.
   - Security: Document security considerations and best practices for using the API securely.

6. Versioning and Changelog:
   - Versioning: Explain how versioning of the API is managed.
   - Changelog: Keep a record of changes made to the API, including new features, bug fixes, and deprecated endpoints.

7. SDKs and Code Samples:
   - SDKs: Provide links or instructions for using SDKs or client libraries available for the API.
   - Code Samples: Include code snippets or examples in various programming languages to demonstrate API usage.

8. Support and Contact:
   - Support Channels: Specify the available support channels (email, forums, etc.).
   - Contact Information: Provide contact details for reporting issues or seeking assistance.