REST (Representational State Transfer) is an architectural style for building web services. A REST API (Application Programming Interface) is a type of web service that adheres to the principles of REST.

Here are some guidelines for documenting a **REST API**:

1. **Overview:** Provide a high-level description of the API, including its purpose, functionality, and the types of resources it supports.
2. **Endpoint reference**: List the endpoints of the API, including the HTTP methods supported (GET, POST, PUT, DELETE), the URI (Uniform Resource Identifier) of each endpoint, and the parameters that can be used with each method.
3. **Request and response formats**: Specify the format of the request and response messages, including any required headers, body, and media type. It's common to use JSON (JavaScript Object Notation) or XML (Extensible Markup Language) for these messages.
4. **Error handling**: Describe the error responses that the API can return, including the HTTP status code and a human-readable message that explains the problem.
5. **Authentication and security**: Explain the authentication mechanism used by the API, such as OAuth2, Basic Authentication, or API keys. Describe the security measures in place to protect user data.
6. **Examples**: Provide examples of how to use the API, including sample requests and responses.
7. **Best practices**: Include best practices for using the API, such as rate limiting, pagination, and versioning.
8. **Code samples and libraries**: Provide code samples in various programming languages and libraries that developers can use to interact with the API.
9. **Support**: Provide information on how developers can get support, including a contact email or link to a support forum.

Documenting a REST API is essential to ensure its usability and scalability, and it can save time and resources for developers.