The formulation of API documentation is imperative for developers to comprehend the manner in which they may engage with your API. A meticulously organized documentation generally encompasses the ensuing sections:

1. **Introduction**
2. **Authentication**
3. **Endpoint Descriptions**
4. **Request and Response Formats**
5. **Examples**
6. **Error Handling**
7. **FAQs**

**Book API:**

**Book API Documentation**

**Introduction**

The Book API allows users to access and manage a collection of books. You can retrieve, add, update, and delete books in the collection.

**Base URL**

Sever

https://api.example.com/v1

**Authentication**

This API uses API keys for authentication. You must include your API key in the header of each request.

**Header Example**

Makefile

Authorization: Bearer YOUR\_API\_KEY

**Endpoints**

**1. Get All Books**

* **Endpoint:** GET /books
* **Description:** Retrieves a list of all books.

**Request**

bash

GET https://api.example.com/v1/books

Authorization: Bearer YOUR\_API\_KEY

**Response**

json

{

"books": [

{

"id": 1,

"title": "1984",

"author": "George Orwell",

"published\_date": "1949-06-08"

},

{

"id": 2,

"title": "To Kill a Mockingbird",

"author": "Harper Lee",

"published\_date": "1960-07-11"

}

]

}

**2. Get Book by ID**

* **Endpoint:** GET /books/{id}
* **Description:** Retrieves a book by its ID.

**Request**

bash

GET https://api.example.com/v1/books/1

Authorization: Bearer YOUR\_API\_KEY

**Response**

json

{

"id": 1,

"title": "1984",

"author": "George Orwell",

"published\_date": "1949-06-08"

}

**3. Create a New Book**

* **Endpoint:** POST /books
* **Description:** Adds a new book to the collection.

**Request**

bash

POST https://api.example.com/v1/books

Authorization: Bearer YOUR\_API\_KEY

Content-Type: application/json

{

"title": "Brave New World",

"author": "Aldous Huxley",

"published\_date": "1932-09-01"

}

**Response**

json

{

"id": 3,

"title": "Brave New World",

"author": "Aldous Huxley",

"published\_date": "1932-09-01"

}

**4. Update a Book**

* **Endpoint:** PUT /books/{id}
* **Description:** Updates an existing book.

**Request**

bash

PUT https://api.example.com/v1/books/1

Authorization: Bearer YOUR\_API\_KEY

Content-Type: application/json

{

"title": "Nineteen Eighty-Four",

"author": "George Orwell",

"published\_date": "1949-06-08"

}

**Response**

json

{

"id": 1,

"title": "Nineteen Eighty-Four",

"author": "George Orwell",

"published\_date": "1949-06-08"

}

**5. Delete a Book**

* **Endpoint:** DELETE /books/{id}
* **Description:** Deletes a book from the collection.

**Request**

bash

DELETE https://api.example.com/v1/books/1

Authorization: Bearer YOUR\_API\_KEY

**Response**

json

{

"message": "Book deleted successfully"

}

**Error Handling**

* **400 Bad Request:** The request was invalid.
* **401 Unauthorized:** The API key is missing or invalid.
* **404 Not Found:** The requested resource was not found.
* **500 Internal Server Error:** An error occurred on the server.

**FAQs**

* **How do I obtain an API key?**
  + Sign up at our website and navigate to the API section.
* **Can I access the API without an API key?**
  + No, an API key is required for all requests.