API Documentation by *Team Aquascript*

Aquascript Book API Documentation

Overview

The Aquascript Book API is a lightweight JSON-based RESTful API that provides metadata for books, including author, country of origin, language, page count, and more.

API Endpoint

The base URL to fetch the books data is:

```
https://wecoded-dev.github.io/Aquascript/api/books.json
```

File Structure

Each book object follows this format:

```
{
  "author": "Chinua Achebe",
  "country": "Nigeria",
  "imageLink": "assets/ThingsFallApart.jpg",
  "language": "English",
  "link": "https://en.wikipedia.org/wiki/Things_Fall_Apart",
  "pages": 209,
  "title": "Things Fall Apart",
  "year": 1958
}
```

- **author**: Name of the book's author.
- **country**: Country of origin or setting.
- **imageLink**: Path to the book's cover image.
- **language**: Main language of the book.
- **link**: URL for more information (e.g., Wikipedia).
- **pages**: Total number of pages.
- **title**: Book title.
- **year**: Year of publication.

Using the API

Fetch data using JavaScript's

```
fetch()
```

API:

```
fetch('https://wecoded-dev.github.io/Aquascript/api/books.json')
  .then(response => response.json())
  .then(data => {
    console.log(data); // View book data
  });
```

Displaying Books (Example)

HTML placeholder for books:

```
<div id="book-list"></div>
```

JavaScript example to populate the book list:

```
document.addEventListener('DOMContentLoaded', () => {
  const bookList = document.getElementById('book-list');

  fetch('https://wecoded-dev.github.io/Aquascript/api/books.json')
    .then(response => response.json())
    .then(data => {
      data.forEach(book => {
        const bookCard = document.createElement('div');
        bookCard.innerHTML = `
          <h3>${book.title} (${book.year})</h3>
          <p><strong>Author:</strong> ${book.author}</p>
          <p><strong>Country:</strong> ${book.country}</p>
          <p><strong>Language:</strong> ${book.language}</p>
          <p><strong>Pages:</strong> ${book.pages}</p>
          <img src="${book.imageLink}" alt="${book.title}" width="120">
        `;
        bookList.appendChild(bookCard);
      });
    })
    .catch(error => console.error('Error loading books:', error));
});
```

Best Practices

- Always validate external links and inputs.
- Use lazy loading for book images to boost performance.
- Structure content semantically for accessibility.
- Implement fallback/error UI for API fetch failures.

Conclusion

The Aquascript Book API empowers developers to integrate rich book data into their projects. Whether you're building a reading app, digital library, or educational tool, this API provides the structure and accessibility to get started quickly.