API Documentation by *Team Aquascript*

Aquascript Movies API Documentation

Overview

The Aquascript Movies API provides a rich collection of movie data in JSON format, ideal for developers building applications to showcase, filter, or analyze movies. It contains genre lists, metadata, and downloadable links.

API Endpoints

- **Genres & Basic Movie Data:**
  `moviesdata.json`
  - Basic movie information and genres.
- **Extended Movie Listings:**
  `moviesdata++.json`
  - Expanded details, including director, release date, and download links.

File Structure

moviesdata.json contains:

- `genres`
  : Array of movie genres, for categorizing movies.
- `movies`
  : List of movie objects with details like title, release year, director, etc.

moviesdata++.json is an array of objects, where each object contains:

- `name`
  : Movie title.
- `tags`
  : Categories or filters for the movie (e.g., 'action', 'comedy').
- `details`
  : Detailed information, including director, release date, genre, quality, and size.
- `links`
  : Direct download links for the movie.
- `image`
  : URL of the movie's poster image.

Using the API

Use JavaScript's
`fetch()`
to retrieve data from the API:

```
fetch('path/to/moviesdata.json')
  .then(response => response.json()) // Converts response into JSON
```

```
  .then(data => {
    console.log(data); // Logs the data for debugging
  });
```

Displaying Movies (Example)

HTML structure to display movies:

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

JavaScript to populate the movie list:

```
document.addEventListener('DOMContentLoaded', () => {
  const movieList = document.getElementById('movie-list'); // Grabs the container element

  // Fetch movie data from JSON file
  fetch('path/to/moviesdata.json')
    .then(response => response.json()) // Converts data into JSON format
    .then(data => {

      // Iterate through the movies array and create individual movie cards
      data.movies.forEach(movie => {
        const movieCard = document.createElement('div'); // Creates a new div element for each mo
vie

        // Populate the movie card with relevant movie data using template literals
        movieCard.innerHTML = `
          <h3>${movie.title} (${movie.year})</h3>
          <p><strong>Director:</strong> ${movie.director}</p>
          <p><strong>Genres:</strong> ${movie.genres.join(', ')}</p>
          <img src="${movie.posterUrl}" alt="${movie.title}" width="150">
        `;

        // Append each card to the main movie list container
        movieList.appendChild(movieCard);
      });

    })
    .catch(error => console.error('Error loading movie data:', error)); // Handle errors graceful
ly
});
```

Best Practices

- Validate and sanitize external links before use to prevent security issues.
- Lazy load images to improve performance, especially for large datasets.
- Use semantic HTML for accessibility to ensure screen readers can interpret content correctly.
- Gracefully handle API errors, ensuring your application doesn't crash if the API fails.

Conclusion

The Aquascript Movies API is a versatile, developer-friendly way to integrate movie data into your applications. Use it to create genre filters, movie cards, or download features. It’s perfect for developers who want to display rich, dynamic movie content in their projects.